NTNU
Innovation and Creativity

Universal Design of Electronic Forms for Mobile Phones

Accommodating the Cognitively Disabled

Ragnhild Flaten

Master of Science in Computer Science
Submission date: June 2007
Supervisor: John Krogstie, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

The mobile phone is becoming an important channel for digital services. The web-technologies have matured, and many service providers are now developing strategies for including the mobile channel in the portfolio of relevant service channels.

In this context, one specific area of interest is online forms. Forms are widely used in communication between the citizen and the public sector. Today, many forms are availabe online on the web. A big challenge is to provide these or similar services on the mobile phone platform. The most demanding challenge is probably connected to the presentation of information and interactivity on the limited screen of the mobile phone, still often displaying only small text fonts. At the same time, the complexity of available online services is increasing, and the mobile phone is owned by a continuously larger user population. This population represents the potential users of all public services. Thus, special care has to be taken when designing the services and the user interface of these.

An important research question is how to represent online forms on the mobile phone without missing important content, or without failing to guide the user through the sequence of interactive transactions. To this research question, we connect firstly the general knowledge of forms and online forms (referece to Elmer). Secondly, we include in the thinking the principles of Universal Design or Design for All. Such guidelines must be followed in order to include as many citizens in the user population as possible. It is widely acknowledged that elderly and cognitively disabled users represent a huge challenge within eInclusion.

The research work builds upon a triangulation of themes:
1. mobile platform
2. online forms
3. universal design/design for all (cf. the cognitively disabled users)
The research work including prototyping will be done within a concrete case.

The base for the research work will be connected to the R&D of Tellu. Tellu is a company that develops and executes mobile services. www.tellu.no
Prototyping as mentioned above will be done on Tellu's mobile platform.

Assignment given: 15. January 2007
Supervisor: John Krogstie, IDI

Abstract

Mobile phones are increasingly becoming common property, which we carry with us practically all of the time. Self-services are also becoming more and more common. Along with the self-service society comes the need to access services at any time, any place, which the mobile phone can provide for. Use of a self-service usually requires exchange of information by means of electronic forms. Finding good solutions for representing forms on mobile phones is a demanding challenge. On the one hand, the small displays and keyboards introduce several design challenges. On the other hand, users with special needs, such as users with cognitive impairments, call for solutions that address accessibility and universal design.

This report documents an investigation on how to make electronic forms feasible on mobile phones. The investigation focused on universal design of mobile form user interfaces. Based on the expected potential of multimodality, two target groups were chosen: 'people with cognitive disabilities that lead to moderate text-reading problems', and 'people with cognitive disabilities that lead to moderate number-reading problems'. We investigated the validity of a hypothesis containing two assertions: (1) that specific techniques and form controls can make electronic forms feasible on mobile phones as well as relieve the burden of the users, and (2) that offering visualisation of information as an additional modality in electronic forms on mobile phones can be beneficial for the cognitive processing of content in such forms.

The research method used in the investigation comprised several parts: A literature study and a semi-structured theme interview formed a background for setting success criteria for the feasibility of forms on mobile phones and making design choices. Paper prototyping was conducted by making a paper mock-up of a selected case, kindergarten applications, which illustrated how the design choices could appear. The paper mock-up evolved through several iterations, and served as a design specification for a J2ME prototype, which was made in order for users to better be able to evaluate the design decisions. The design evolved further during the prototyping in J2ME, as necessary improvements were discovered. The final part of the research method was a walkthrough evaluation. Ten users went through the form and were afterwards asked to state their opinion on specific design aspects. The results from the walkthrough evaluations provided material to use when assessing the validity of the hypothesis.

The investigation led to a list of success criteria with suggestions on how to achieve several of them. We attempted to achieve four of these criteria, by making several design decisions. The final user interface of the kindergarten form had aspects we were concerned about, and addressed in the walkthrough evaluations The most important results of the evaluations were: They were all positive towards a technique used for decreasing the information intensity. None of the users claimed to have problems with or dislike the form control 'drop-down box' and a technique used to maintain the form context when drop-down boxes were expanded. 'Choose' commands may be unnecessary. The solution for indicating form size and progress was understood by all. A graphical modality presenting kindergartens in a map, accommodating people with text-reading problems, was preferred by seven of the ten users, while none of the users preferred the graphical modality presenting points of time as clock icons, accommodating people with number-reading problems.

The results led us to conclude that the first assertion seem to be correct, depending on the outcome of further, suggested tests. We assume that professionally made clock icons, combined with a technique suggested by one of the evaluators would have made more people prefer this presentation. We also se no reason why multimodality, acknowledged to be beneficial for the processing of cognitive content, should be less beneficial on mobile phones, and believe that also the second assertion is true. A number of efforts for further investigations are suggested.

Preface

This report is a Masther thesis written in the 10th semester of the Master of Science in Computer Science study at the Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science. The research assignment is given by Tellu AS, a company that develops and executes services for mobile phones.

The assignment was based on Tellu's participation in the UNIMOD-project[1]. The main objective of the UNIMOD-project is to improve the competence of multimodal, personalized user interfaces, and thus to improve the accessibility and use of electronic services. The report documents the investigation on how to make electronic forms feasible on mobile phones and how to implement multimodal content in order to support users with certain cognitive disabilities. The report contains several Computer Science terms, as it is intended to be read by people from this field.

I would like to thank Riitta Hellman at Karde AS, the supervisor of the investigation, for her invaluable guidance and advice. I would also like to thank Geir Melby, Knut Eilif Husa and Robert Bjærum at Tellu AS for providing assistance and feedback, and John Krogstie at the Information Systems Group, NTNU, for providing feedback on the report. Finally, I would like to thank the ten anonymous people who participated in the walkthrough evaluations, particularily the five students who were very busy preparing for their exams.

Trondheim, June, 2007

Ragnhild Flaten

[1] http://www.unimod.no/

Table of Contents

1 Introduction

1.1 Background

Mobile phones have become common in our society. Today, owners and users of mobile phones are among all parts of the population, from children to the elderly. A survey conducted by Statistics Norway showed that 92% of the respondents in a number of Norwegian households owned a mobile phone in 2005 (Statistics Norway, 2006, electronic). The mobile phone is, like wallets and keys, an artefact most of us carry with us practically everywhere at any time. People using mobile phones are today a common sight on busses, in a queue at the supermarket, and in all other places of the public space.

Parallel to the development of the "mobile phone society", we can observe a rapid development of a large variety of self-services. We now expect to find the information we need, purchase products and access services on the Internet. In addition to the Internet, vendors encourage us to use self-service kiosks for purchasing tickets to movies, public transportations and so on, increasingly by-passing the 'middleman'. This is leading to the development of a rather extreme self-service society, as more and more vendors stop providing alternatives to the self-services.

This development is driven by several forces: The users, who like to do their tasks at a time that suits them, rather than taking free from work; the service providers who want to decrease costs as well as provide services that meet the customer needs; technological advances that improve the self-services; and the societal reforms and development programmes. The societal reform programmes often have goals like "broadband access to the Internet for everyone", "information and communication technology should be designed to encompass everyone's abilities" and "information technology shall support the development of the public sector delivering services as good as possible" (The Norwegian Ministry of Government Administration and Reform, 2006; The Norwegian Ministry of Modernisation, 2005).

Along with the self-service society comes the need to access services at any time, and at any place. Mobile phones are increasingly employed for several other purposes than telephony. The phones more and more tend to offer some of the types of functionality and services we are accustomed to from using computers. Many service providers have realised the potential of mobile phones, and are now working to make the mobile phone become one of their regular service channels. These developments together more or less force the *whole population* to become users of self-services; more and more often, there are no other alternatives.

Use of a self-service requires exchange of necessary information between the receiver and the provider. This exchange is usually done by means of an electronic form. For services to be available on mobile phones, it is important to find a good way to represent forms on the mobile phone platform. Finding the best ways of presenting information and interacting with the devices, are demanding challenges on mobile phones. One of the main challenges is, naturally enough, the very limited display and keyboard size compared to screens and keyboards of desktop computers.

Human-Computer Interaction (HCI) is a specialised field of competency which addresses usability issues. Another, related field is Universal Design, or Design for All. These fields have become highly relevant over the last years, also in the context of computing, telecom and digital media. People worldwide are now living several years longer than earlier. In Norway, the expected lifetime of men was 72,87 years in 1986. Nine years later it had risen to 77,72 years. The rise in expected lifetime for women is not quite as drastic, with the corresponding numbers 79,74 and 82,52. (Statistics Norway, 2006, lifetime) The fact that people are living longer, leads to a continuous increase in the number of people with age-related impairments. Since the part of the population with disabilities is growing, more people are now agreeing upon that we need environments, products and services *designed to be usable for all*, rather than making them accessible by customising them for the different types of disabilities. This becomes very necessary in the case of self-services. Several societies and governments have committed themselves to practise Universal Design/Design for All, including the Norwegian government (The Norwegian Ministries of Labour and Social Inclusion and of the Environment, 2004).

There are several solutions and guidelines on how to make services on computers, and a few on mobile phones, accessible and usable by people with disabilities. These solutions mostly address users with visual and hearing impairments. There are relatively few solutions or guidelines for people with physical impairments, as their problems mostly concern physical aspects of the device, usually the visual or tactile ones.

Solutions for the cognitively impaired users also seem to be scarce, while the number of people with low cognitive skills is in fact quite large. The Adult Literacy and Life Skills Survey showed that, depending on country, between one-third and over two-thirds of the population do not attain a suitable minimum skill level in prose and document literacy, numeracy and problem solving (OECD, 2005).

In other words, very many users struggle with reading and understanding simple texts, or solving simple calculations and problems. Moreover, they will be more as the number of elderly increases. For that reason, it is very important to pay attention to the needs of people with cognitive disabilities. In this report, we direct out attention to the cognitively disabled users of self-services accessed by the mobile phone.

1.2 The Investigation

1.2.1 The Research Assignment

The research assignment was to investigate and demonstrate how user interfaces of electronic forms can be successfully represented on the mobile phone, particularly considering the needs of cognitively disabled users. Fundamental ideas behind Universal Design must be incorporated in the solution. Prototyping should be done on a case, in order to illustrate the results. The prototype should be realised on the ServiceFrame-platform of Tellu AS, a spin-off company from Ericsson's Norwegian research centre. Tellu AS develop and execute mobile services, with which the users interact through client applications on their mobile phones.

1.2.2 Research Hypothesis

There are several considerations that need to be made when designing a mobile form user interface for peoply with varying skills and abilities. In addition to the varying skills and abilities of the users, the device introduces limitations because of the small size of the screen. We assert that **specific techniques and form controls can make electronic forms feasible on mobile phones and relieve the burden of the users.**

Incorporating Universal Design in electronic forms on mobile phones should be beneficial. A design that accommodates people with varying degrees and types of disabilities makes the user interface available to more people. Application of multimodality on computer user interfaces is claimed to be helpful to people with cognitive disabilities (Newell et al., 2002; Bohman, 2005; Jiwnani, 2001; WebAIM, 2007, Design Considerations; Burkhart, 2003). We assume that this claim also applies to mobile phones, and therefore assert that **offering visualisation of information as an additional modality in electronic forms on mobile phones can be beneficial for the cognitive processing of content in such forms.**

1.2.3 Target Groups

Since as many as one to two thirds of the populations of countries participating in the Adult Literacy and Life Skills Survey have low literacy and numeracy skills (OECD, 2005), the investigation had two target groups. These groups were: (1) people with cognitive disabilities that lead to moderate *text-reading* problems, and (2) people with cognitive disabilities that lead to moderate *number-reading problems*. 'Moderate' here means that people in both target groups are capable of perceiving the content of respectively text and numbers, if they get enough time and are able to concentrate on the task at hand.

1.3 Structure of the Report

The remainder of the report is organised as follows: Chapter 2 describes the research method used in the investigation. Chapter 3 elaborates on the characteristics of forms, and on electronic forms in particular. This chapter also contains an interview with a form expert, and states several success criteria for forms on mobile phones.

Chapter 4 briefly describes the field of Universal Design and presents its most reputed design principles. The fifth chapter gives a brief description of different disability categories, and elaborates more thoroughly on the 'cognitive disabilities' category. The results from the Adult Literacy and Life Skills Survey (OECD, 2005) are also more thoroughly described, as well as two conditions that can be found among people in the target groups (Section 1.2.3).

Chapter 6 describes multimodality, how it affects the human cognition, and possible benefits and disadvantages from the use of it. The seventh chapter describes the design choices that were made in a kindergarten application to support the hypothesis, by demontrating several of the aspects stated to be a way to achieve the success criteria in Chapter 3. The demonstration of multimodal content comes in under one of these criteria. The chapter presents a paper mock-up which illustrates the design choices.

Chapter 8 describes how the design was relised by prototyping in J2ME on Tellu's ServiceFrame-platform. The chapter also contains screen shots from the prototype as well as reasonings for new and left out design decisions.

The ninth chapter presents a walkthrough evaluation that were executed to get user feedback on the design choices. The chapter describes which of the design considerations that were considered important to have users evaluate, the preparations for the walkthrough evaluation, the results and an analysis of these. Whether or not the assertions of the hypothesis are correct is discussed in Chapter 10 along with other topics, and ideas for future investigations and research are suggested.

The report has three appendixes. Appendix A contains principles and guidelines for design of user interfaces on mobile devices, for design of forms, and for design of user interfaces for people with disabilities and cognitive disabilities. Appendix B contains the Javadoc documentation of the prototype, the source code is in a separate ZIP-file. The last appendix contains the full answers the users gave in the walkthrough evaluations.

2 Research Method

The research method that was followed during this investigation comprised several approaches, depending on the phase of the project. Each of these are presented below.

2.1 Literature Study

First, a literature study on areas that concern the research assignment was performed. This was done in order to educate ourselves on the state-of-the-art of the relevant fields of research, i.e. forms, Universal Design, cognitive disabilities and multimodality, in order to get ideas for making design decisions, and finally, in order to maintain and improve our focus on the target groups.

2.2 Semi-Structured Theme Interview

Connected to the literature study, a semi-structured theme interview with a form expert was executed. The semi-structured interview approach is widely used (Flick, 2006). It was chosen because it, as it takes place as a conversation, enables the interviewer to change the wording of the questions as well as their order according to the answers from the interviewee. It also allows asking the interviewee questions which emerge during the interview. According to (Flick, 2006), the interviewee is also more likely to express viewpoints in a semi-structured interview, as opposed to in a structured interview or questionnaire.

The interview concerned the feasibility of forms on mobile phones, and how to make them feasible, particularly regarding the design of form user interfaces. The interview questions were made in forehand, arranged by themes. The literature study and the answers from the semi-structured theme interview formed a background which helped in setting success criteria for the feasibility of forms on mobile phones and making design choices, considering all people with varying skills and abilities, as well as people in the two target groups.

2.3 Paper Prototyping

Paper prototyping is a widely used technique for designing and testing user interfaces (Snyder, 2003). A paper prototype is a low-fidelity prototype. Low-fidelity prototypes have a short development time. This allows the developers to get results regarding the usability of their design early in the process, thereby making necessary changes cheaper to make and more ideas possible to test (Rettig, 1994). Low-fidelity prototypes, such as paper mock-ups, can be useful when those who have an interest in a design discuss and present ideas (Preece et al., 2002).

Elements that contribute to making forms successful on mobile phones, considering the user interface, was supposed to be demonstrated. However, not all of the success criteria could be realised. In order to effectively decide on design choices without wasting time and resources, paper prototyping was a part of the research method. The paper prototyping was done by making a paper mock-up which illustrated how the user interface of a selected case, a kindergarten application form, could appear. The paper mock-up evolved as it was presented to the supervisor of the investigation, for feedback on the usability of the design, and to Tellu AS, for feedback on what was possible regarding implementation of the design, in several iterations, until the design was found to be satisfactory. The paper mock-up then served as a design specification for the J2ME prototype.

2.4 J2ME Prototyping

The research assignment specified that prototyping of the design should be done on the ServiceFrame-platform of Tellu AS, i.e. in the programming language J2ME. J2ME (Java 2 Platform, Micro Edition) is one of the technologies that Tellu AS uses to create their mobile services. J2ME is more recently known as Java ME (Java™ Platform, Micro Edition). The technology makes developers able to use the Java programming language to make programs for mobile devices like mobile phones, personal digital assistants, TV set-top boxes and printers (Sun Microsystems, 2006).

A J2ME prototype is a high-fidelity prototype. Unlike low-fidelity prototypes, high-fidelity prototypes are interactive and can be operated as if they were a final product. This gives users a feeling of how the product is operated, and enables them to better give feedback regarding improvements of the user interface, as well as to more thoroughly evaluate it. High-fidelity prototypes can also help marketing and selling a product. (Rudd et al., 1996)

As mentioned in the previous chapter, the paper-mock up was the design specification of the J2ME prototype. Issues discovered during the implementation led to some changes, resulting in the final design.

2.5 Walkthrough Evaluation

The final part of the research work was an evaluation, where users first went through the form, presented on the J2ME prototype, and afterwards were asked to state their opinion on specific design aspects, by asking them questions about the design decisions that were made. This qualitative evaluation approach was chosen because the boundaries of this investigation would make us unable to perform a thorough quantitative evaluation, e.g. we would not be able to get a satisfying number of test users, the test users would not be representative for the user group of the tested artefact as a large number of the test users would be students, mostly in their early twenties. For the purpose of this investigation, we evaluated this to be sufficient.

The evaluation procedure used is based on the Heuristic Evaluation method. Heuristic Evaluation is reportedly, in a survey from 2000, the most used usability evaluation method (Rosenbaum et al., 2000). Heuristic Evaluation is a method where evaluators examine the user interface on their own. They can, particularly if they are unfamiliar with the domain, follow a scenario that lists the steps they take to achieve tasks. The evaluators compare elements of the user interface with specified usability heuristics. (Nielsen, 2007) Typically, 3-5 evaluators identify 75-80% of the usability problems (Nielsen, 1994).

We expected that users would have to learn how to interact with the user interface of the J2ME prototype. As there is no help functionality implemented, we decided to have the evaluators go through the form, following a task scenario, with an evaluation leader who provided help when necessary. The evaluation leader took notes during the walkthrough. Rather than having the evaluators consider aspects of the design during the walkthrough, the evaluation leader pointed out and demonstrated the aspects afterwards, asking the evaluators pre-made questions about their opinion towards the usability of these aspects, about their observations, as well as some general questions. The question sessions were semi-structured, using the notes from the walkthrough and allowing the evaluators to come with spontaneous questions and remarks “on the side”. This approach to the Heuristic Evaluation method was used, as mentioned, because of the users’ lack of knowledge on how to interact with this type of user interface. It was also chosen because we wanted to discuss specific aspects of the solution with the evaluators in order to get their full opinion, rather than having them consider the prototype alone against pre-made usability heuristics. We decided to use ten evaluators in two age groups (age 22-50) of five evaluators: the ‘student’ group and the ‘adult’ group. It is important to notice that the evaluators were neither parents of children in the kindergarten age, nor cognitively disabled.

3 Forms

Interaction between citizens and the public authorities, and between customers and suppliers of products and services, is usually performed by using forms for structured information. No matter which service one wishes to access on the Internet, what pops up on the screen is an electronic form of some kind. In other words, case handling in very different contexts is profoundly based on forms. Today's forms are either available only on paper, both on paper and electronically, or only electronically. This chapter outlines the form's characteristics, and elaborates on this in order to cover electronic, interactive forms.

3.1 What is a Form?

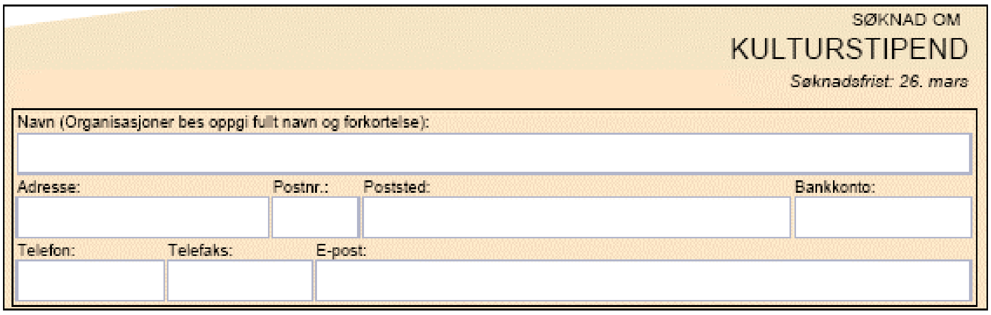
According to Cambridge Advanced Learner's Dictionary, a **form** is

"...a paper or set of papers printed with marked spaces in which answers to questions can be written or information can be recorded in an organized way." (Cambridge University Press, 2007)

Forms are widely used in the electronic world. Narain H. Gehani states in the article "The Potential of Forms in Office Automation" that

"An electronic form is the computer analogue of a paper form. Fields in an electronic form are filled as in a paper form." (Gehani, 1982)

This is still, 25 years later, a reasonable claim. Most electronic forms have clear resemblances with paper forms, although electronic forms can differ from their paper equivalents in several ways. Figure 1 below and Figure 2 on the following page respectively show the paper and electronic versions of a public application form's personal data entry part. Except for the layout of the entry fields, the asterisks and some additional information in the first entry field's label, the samples look alike.



SØKNAD OM
KULTURSTIPEND
Søknadsfrist: 26. mars

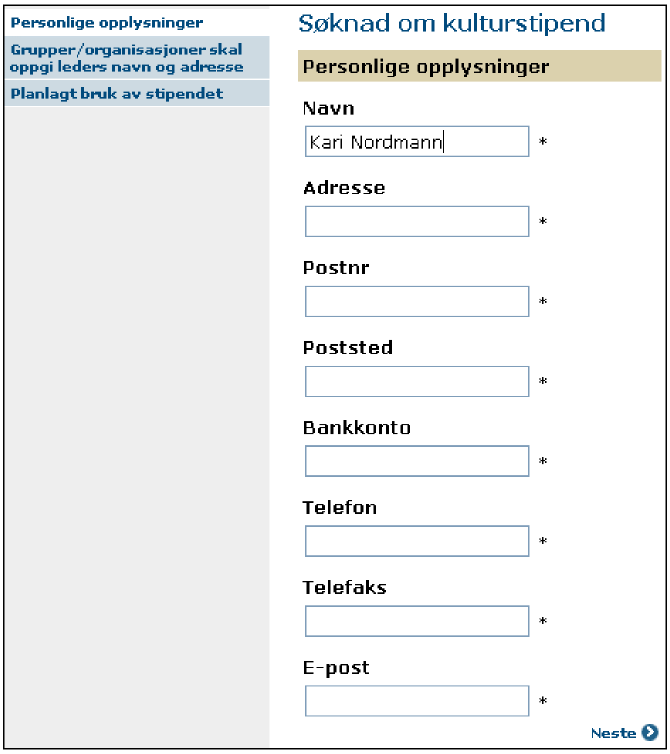
Navn (Organisasjoner bes oppgi fullt navn og forkortelse):

Adresse: Postnr.: Poststed: Bankkonto:

Telefon: Telefaks: E-post:

Figure 1. Part of public paper form: Application for "art scholarship" (The Municipality of Trondheim, 2007).

Generally taken, a form is typically used to collect information from a person, a group of persons or a company. What kind of information the form requests depends on its purpose. There are many types of forms; survey forms, procedure forms, application forms, registration forms, and report forms are some examples. In Statistics Norway's handbook for development of web forms (Haraldsen, 2004), forms are divided into two categories: Distributed forms and forms that are requisitioned by the user. Distributed forms are forms that their provider distributes to certain groups, requesting a response within a time limit. Survey forms clearly belong to this category. Forms that are requisitioned by the user are characterised by that the user initiates the process of finding and filling in the form. Application forms typically belong to this category.



Personlige opplysninger
Grupper/organisasjoner skal oppgi leders navn og adresse
Planlagt bruk av stipendet

Søknad om kulturstipend

Personlige opplysninger

Navn

Kari Nordmann *

Adresse *

Postnr *

Poststed *

Bankkonto *

Telefon *

Telefaks *

E-post *

Figure 2. The "computer analogue" of the part of the form depicted in Figure 1 (The Municipality of Trondheim, 2007).

3.2 Characteristics of Electronic Forms

The term 'electronic form', or other notions that are commonly used (such as 'e-form' and 'Eform'), can apply to both specific services and to simple input forms in connection with the use of an electronic service. Electronic forms provided by public authorities and companies who are specialists on electronic forms, are often referred to as a service in themselves, while those who provide electronic services (selling products over the Internet, organise and manage online communities, etc.) use electronic forms as a means of collecting required information from their customers and clients. Form design principles will be the same for both types. The following sections outline the characteristics of electronic forms.

3.2.1 Types of Electronic Forms

A survey conducted by Alistair Ross at the Department of Information Science at City University of London, revealed that there are four main types of digital forms; the word processor file, the Portable Document Format (PDF) file, the Web (HTML and/or other scripting program) and stand-alone software (Ross, 2000).

The Word Processor File

A word processor file is often a Microsoft Word file or something similar. A digital form in this format can be completed in the corresponding program, or be printed and filled in on paper. If the former alternative is chosen, programs like Microsoft Word can give some benefits in form of for instance macros, which perform certain actions when the necessary information is filled in, and the user has signalled completion of a field by pressing specific keys or navigating to another field.

The Portable Document (PDF) file

PDF-files have to be opened in Adobe's software (Adobe, 2007). Adobe Reader® is a free program which allows users to open PDF-files, but not to manipulate them. Forms on the PDF-format are usually intended only to be printed and filled in on paper, as the software necessary to alter PDF-files with interactive fields cost money, and the Adobe products are not yet so common on people's computers as, for instance, the Microsoft Office products.

The Web (HTML and/or other scripting programs)

Forms in HTML or other scripting programs are interactive, as they have "manipulable" controls in form of input fields, radio buttons, etc., interactive help functions and validation checks. Forms on the Web (in HTML etc.) are intended to be filled in on the Web only.

Stand-alone Software

The stand-alone software package is usually made to support a specific type of information collection. Forms in such packages have the same controls as those in web forms. According to Ross back in the year 2000, these can be the most interactive forms, as they can provide greater opportunities of guiding and directing the user in filling in the forms.

Word processor files and PDF-files are not of interest in this report, as they usually are not interactive, and are rarely intended to be filled in on-screen. Hereafter, the use of the term 'electronic forms' will apply to only the two latter of Ross' categories, i.e. forms on the Web and in stand-alone software.

3.2.2 The Typical Scenario of an Electronic Form

The forms on the Web are either "freely" available (open to use for anyone and no authentication required) or only available after authentication. Forms in the latter group are found on Web-portals for different types of communities and organisations, and in pure form/ reporting systems, such as the Norwegian Altinn (Altinn, 2007). The following is a basic user interaction scenario from (Gouscos et al., 2002), rewritten to apply to forms on Web-sites and stand-alone software that require authentication.

A Typical Interaction Scenario

1. The user authenticates herself and is granted access to the forms and any other services.
2. She chooses the form of interest and fills it in. Basic validation checks, whether or not the entries made are on the correct format etc., are applied on each finished section during the form-filling process. The user must correct any faults in each section before she can proceed to the next.
3. After completing the form, the user submits it to the form provider. An additional validation check is performed, such as cross-checking the given information with other sources. If any faults are found, the user corrects them and submits the form again for a new validation check.
4. After the user has corrected any faults found in the last validation check, she is presented with a receipt, containing the information she gave. She is then given the choice of printing or saving the receipt, logging off or to choose another form or

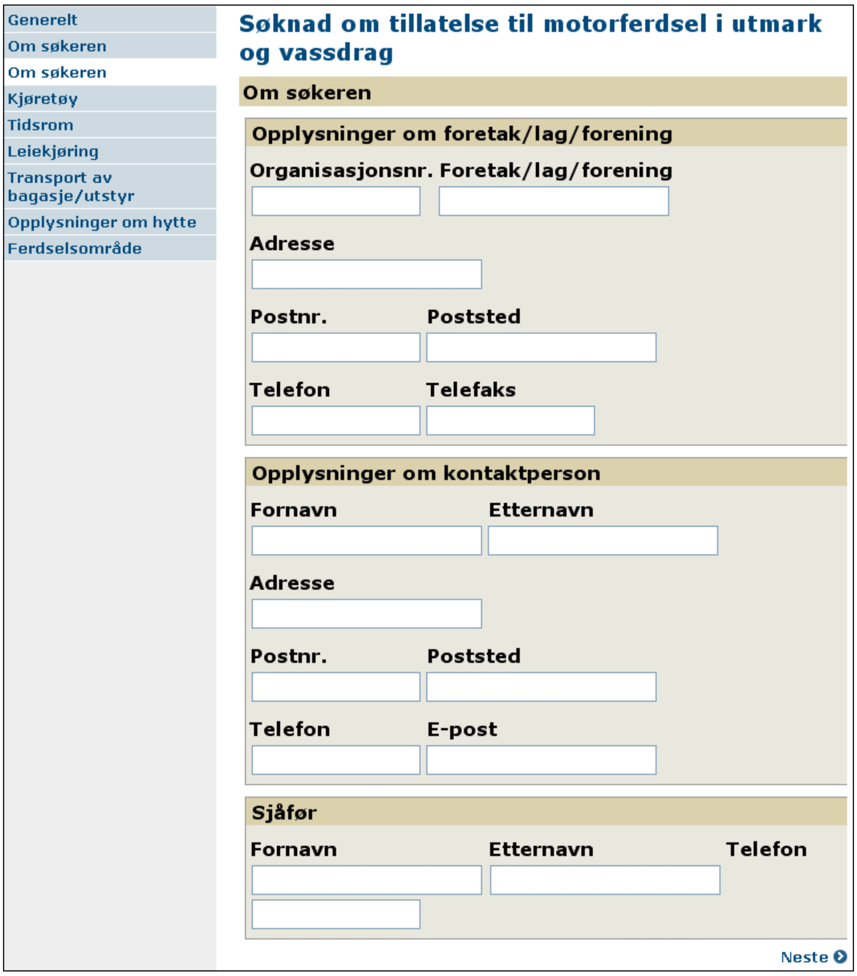
activity. If she chooses one of the first two, she is afterwards asked to do one of the remaining alternatives.

3.2.3 Structure of Forms and Form Systems

Forms

The structure of a form is important, as it can ease the use of the form. Electronic forms usually have a structure consisting of sections, subsections and individual fields, just like forms on paper. These sections and subsections are either all on the same "page", i.e. the whole form is visible on a screen, or they are divided over a number of screens.

Figure 3 illustrates the latter alternative. The users can see which section they are in by looking at the panel on the left side. Subsections in the main part of the screen are separated from each other by frames around them. Structuring forms in this way divides a large task into smaller ones, and can give the users a feeling of accomplishment for each section or subsection, thereby motivating the users to complete the form.



Søknad om tillatelse til motorferdsel i utmark og vassdrag

Om søkeren

Opplysninger om foretak/lag/forening

Organisasjonsnr. Foretak/lag/forening

Adresse

Postnr. Poststed

Telefon Telefaks

Opplysninger om kontaktperson

Fornavn Etternavn

Adresse

Postnr. Poststed

Telefon E-post

Sjåfør

Fornavn Etternavn Telefon

Figure 3. Step 3 in a web application for permission to use motorised vehicles in outlying fields and watercourses (The Municipality of Trondheim, 2007).

(Vassilakis et al., 2005) state that form layout and field placement are important in complex multi-form services in particular, since:

- A document structure helps the users find the fields they have to fill in.
- When conceptually related fields are placed together, it is easier for the users to gather necessary information from relevant documents.

Using the same layout across forms is also beneficial, as "…it allows users to familiarise more quickly with the forms and exploit the knowledge amassed from using one service in the context of other services" (Vassilakis et al., 2005).

Form Systems

Statistics Norway's handbook for development of web forms lists the different types of forms in a web-based reporting system (Haraldsen, 2004, p. 37). These types can be used to describe the structure, not only of web-based reporting systems, but also of other online systems that provide forms and of stand-alone software. The fourth type of form was originally stated to be 'reporting forms', but is here changed to 'forms' in order for the structure to be applicable to other systems than reporting systems.

1.	Log-in form	The users must authenticate themselves to get access to the system.
2.	Administration form	Provides an overview of the different forms (/services) in the system.
3.	Background form	Contains the most frequent questions, such as personal information, address, etc. The other forms can automatically collect this type of information when the user has filled in the background form.
4.	Forms	Different types of forms, found in the administration form.
5.	Submission form	Lets the users submit the information they have given. A submission form is often a part of those in category 4, but with a 'Submit' button.
6.	Receipt form	Shows the information given or a summary. Provides the user a print or/and save alternative.

3.2.4 Interactivity

Electronic forms bring us several advantages in form of opportunities that paper forms do not have. Interactivity may certainly be the greatest advantage when it comes to improving the usability of forms. The term 'Interactivity' can "denote the interactive aspects of an artefact" (Svanæs, 1999). Unlike a paper form, an electronic form can continually adjust its appearance and content, and offer functionality according to the actions of the user, i.e. the form is interactive, and what takes place between the user and the form is an interaction. Below, some beneficial "interactive aspects" that electronic forms may have, are presented.

Better Way of Routing

Forms may have fields or whole sections which are not relevant for some of their users. In paper forms, these typically manifest themselves through messages like "If you answered no in the previous field, jump to section X". This technique of directing users is known as *routing* (Ross, 2000). The electronic forms can continually, based on the information they receive, be customised to not show any redundant sections or lock them for manipulation. This is a clear advantage over paper forms, as it reduces the chance of users filling in wrong fields.

Personalisation

Paper forms are impersonal, as anyone within their target group has to be able to use them. That is why they often have phrases filled with "ors", like in this example:

Is your <u>wife/husband</u> agreeing to <u>the sale of/letting out on</u> your <u>apartment(s)</u>?

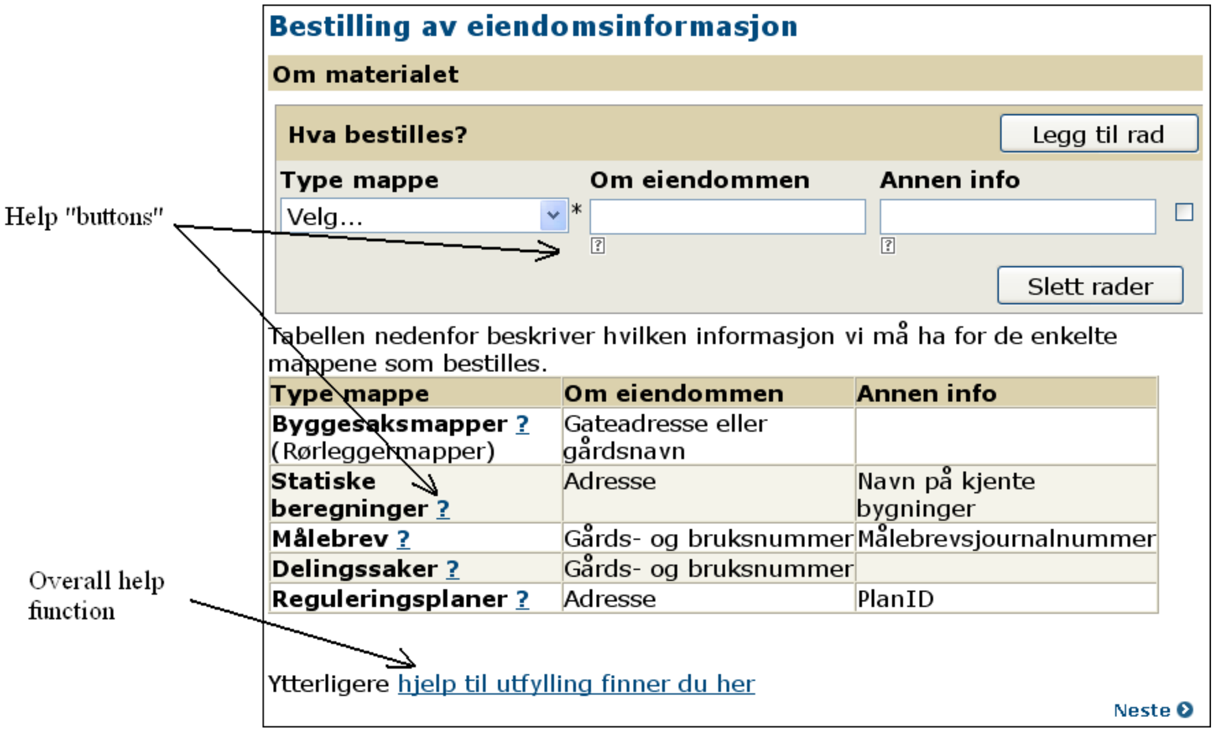
They may also have other impersonal terms, like in this example:

Which three of the following words would you use to describe the working environment of <u>your company</u>?

The text displayed in electronic forms can be customised according to the information given by the users, thereby the "ors" in the first example can be removed. This increases the read-ability of the form, by making it easier to read and more user-friendly (Haraldsen, 2004). The term 'your company' in the second example can be exchanged with the company's name. Using proper names instead of impersonal terms may also increase the readability of the form.

Guidance Functionality

Users may need help and reminders during the process of filling in an electronic form. Producers of electronic forms often place some sort of help buttons near fields they suspect or have experienced can cause confusion among the users. Some choose to have only an overall help function, while others offer an overall help function in addition to the field-help. Figure 4 illustrates some ways to place the guidance functionality. Forms can also initiate help, for instance by measuring the time a user spends on a field. When the time spent has crossed a certain limit, the form can ask the user if he or she needs help, or display a helping text (Ross, 2000). To sum up, there are many ways to implement interactive guidance functionality in a form.



Bestilling av eiendomsinformasjon

Om materialet

Hva bestilles?

Type mappe	Om eiendommen	Annen info	
Velg... *	?	?	□

Tabellen nedenfor beskriver hvilken informasjon vi må ha for de enkelte mappene som bestilles.

Type mappe	Om eiendommen	Annen info
Byggesaksmapper ? (Rørleggermapper)	Gateadresse eller gårdsnavn	
Statiske beregninger ?	Adresse	Navn på kjente bygninger
Målebrev ?	Gårds- og bruksnummer	Målebrevsjournalnummer
Delingssaker ?	Gårds- og bruksnummer	
Reguleringsplaner ?	Adresse	PlanID

Ytterligere hjelp til utfylling finner du her

Figure 4. Order form for information about properties (The Municipality of Trondheim, 2007).

Validation Checks

Validation checks can, as mentioned earlier, be performed during the form-filling and when the users submit their forms. There are many techniques for performing validation checks. In the first case, checks are often performed in connection to a change of screen when the form exceeds a single screen. The content of fields with constraints is checked to ensure that there are no obvious faults. Users are then notified of any faults, and usually can not continue until these are corrected. If validation checks are performed during the process, any checks performed after the users have submitted their forms may involve external operators, like checking if the information given about someone's credit card is correct. Users are also in these cases notified and given the chance to correct their faults.

Controls

Controls are the "manipulable" elements in an electronic form. As Figures 1 and 2 showed, some of the e-form controls look similar to the elements in paper forms. E-forms also have some controls that paper forms cannot have, such as drop-down boxes and radio buttons. The next section presents the common controls of e-forms.

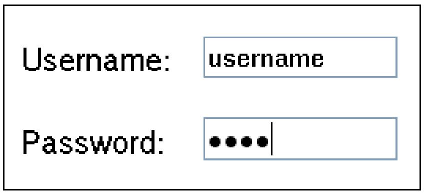
3.2.5 Controls

Controls are representations of information elements, with which the users can interact. They provide interaction mechanisms to let the users fill in their information, and visualise it (controls for blind people may "visualise" the information by reading it, etc.). Input controls are often components of development toolkits, as user interfaces of platforms are, or their creators want them to become, standardised. (Berry, 2000)

The following controls are common and available in many graphical user interface toolkits. The controls may be known by several names, although their functionality is basically the same.

Text Field/Input Field

Text fields are often used when short pieces of information shall be entered. Many toolkits allow developers to put constraints on their text fields, such as numeric input only, password (the field shows the input as asterisks or similar), a minimum of four and a maximum of eight characters, exactly ten characters, etc. Text fields are vital elements in paper forms.



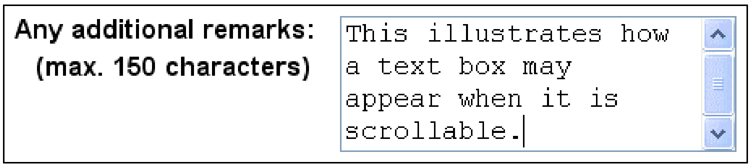
Username: username

Password: ●●●●

Figure 5. Two text fields.

Text Box/Text Area

Text boxes can be used when the users are expected to type in a long text. Text boxes can usually have the same constraints as text fields. The most common way to accommodate this may be to put on a constraint of a maximum number of characters, for layout considerations, transfer considerations, etc. Regarding layout, text boxes from development kits can often become “scrollable” when the users’ input reaches the bottom of the text box, i.e. the input slides up as the user writes. Plain text boxes, i.e. no scrolling, are also used in paper forms.



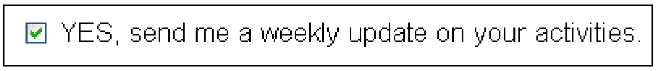
Any additional remarks:
(max. 150 characters)

This illustrates how
a text box may
appear when it is
scrollable.

Figure 6. A scrollable text box.

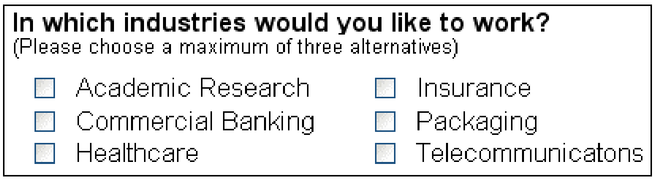
Check Box

Check boxes allow the users to choose information elements. Check boxes are used alone, like in Figure 7, and in groups, see Figure 8. In the latter case, it is common to let the users choose several alternatives. Check boxes are also used in paper forms.



☑ YES, send me a weekly update on your activities.

Figure 7. A single check box.



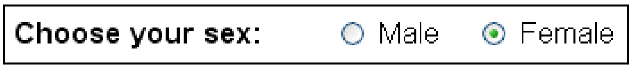
In which industries would you like to work?
(Please choose a maximum of three alternatives)

□ Academic Research □ Insurance
□ Commercial Banking □ Packaging
□ Healthcare □ Telecommunicatons

Figure 8. A group of check boxes.

Radio Button

Radio buttons also allow the users to choose information elements. Unlike check boxes, radio buttons are rarely used alone, and letting the users choose several buttons is not common.

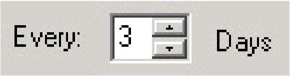


Choose your sex: ○ Male ● Female

Figure 9. Radio buttons.

Spin Button

Spin buttons contain lists or a function for adding/subtracting numbers. Users can only see one element in the list at a time. Changing the list element shown is done by selecting the up or down arrow. Some spin buttons also allow users to enter input.



Every: 3 Days

Figure 10. A spin button (Berry, 2000).

Drop-Down Box

Drop-down boxes contain lists that unfold downwards when the box or its arrow is selected. Only one item in a list can be chosen, as drop-down boxes close upon selection. The users can scroll up and down in the unfolded list if the list of alternatives takes more space than available.

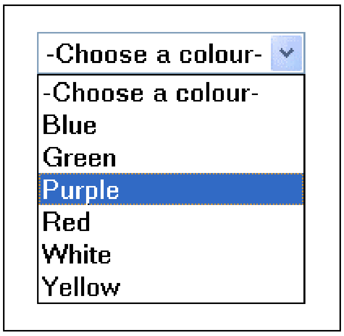


Figure 11. A drop-down box.

List Box

List boxes naturally also contain lists. Like in drop-down boxes, users can scroll through the alternatives in the list, if the size of the list box does not accommodate presentation of all of the list alternatives at a time. List boxes from development kits often automatically become scrollable when the list alternatives exceed the space available. List boxes often allow the users to choose one or more alternatives, e.g. by pressing a specific combination of keys.

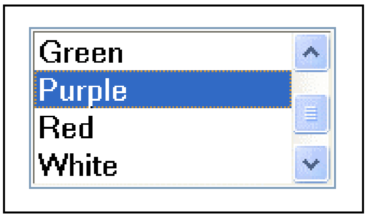


Figure 12. A list box.

Combo Box

Combo boxes can be combinations of text fields and list boxes or drop-down boxes. As the users type, the list box or drop-down box shows all list elements that start with the same characters. The users can finish typing or choose one of the alternatives from the list. This allows the users to save time and/or choose an alternative that is not already in the list.

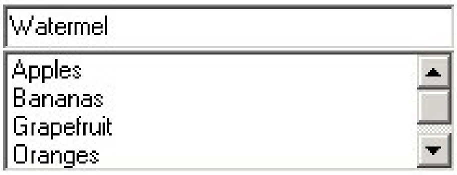


Figure 13. Combo with list box (Berry, 2000).

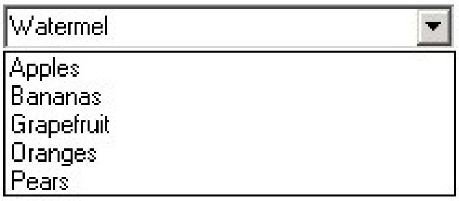


Figure 14. Combo with drop-down box (Berry, 2000).

Push Button/Command Button

Push buttons trigger an action when the users select them. Push buttons in forms are often used to submit the form, go back and forward in the form's screens or dialogues, get help to fill in the form, etc. There are several ways to represent push buttons. Figure 15 shows a rather traditional look:

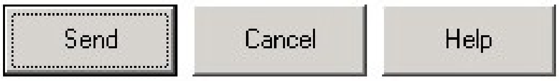


Figure 15. Traditional push button look (Berry, 2000).

Use of other controls, such as the two following, is not quite as common in forms:

Choice Table/Value Set

Choice tables function like radio buttons, by letting the users choose one value from a table. (Berry, 2000)

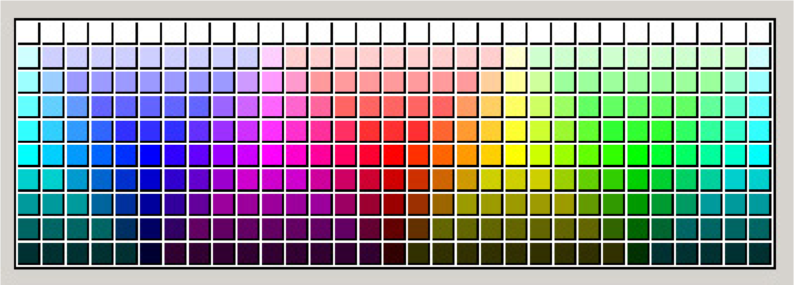
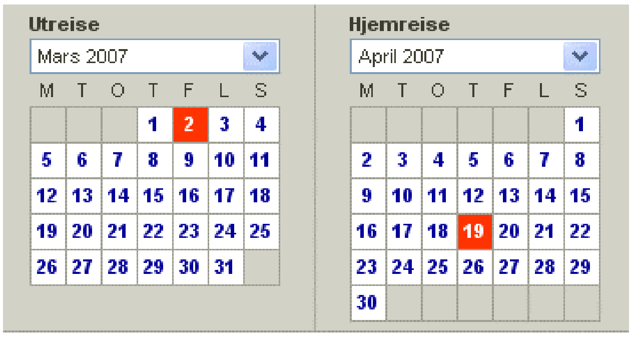


Figure 16. Choice table with colour values (Berry, 2000).

Calendar

Calendars let the users choose a date. Some calendars may also let users choose several dates, although that does not seem to be very common. Calendars are basically, like choice tables, a different way of representing radio buttons or lists of alternatives. Calendar controls usually allow the users to browse through months and a few years ahead.



Utreise

Mars 2007

M	T	O	T	F	L	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Hjemreise

April 2007

M	T	O	T	F	L	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Figure 17. Calendar control for choosing flight dates (SAS Braathens, 2007).

Label/Prompt

Control usually have a label. The label tells users what kind of information the control is there to gather, like "name". Some labels also include directions on how the information should be formatted, its minimum and maximum length, etc. Radio buttons and check boxes in groups have two types of labels; one that describes each button's/box's option and the instruction for the group of controls, like “Choose your three favourite colours”. Some choose to place labels inside the controls, like illustrated in Figure 11.

3.2.6 Pre-filling/Automatic Fill-in

It is quite common that information is automatically collected from registers and filled in forms (Haraldsen, 2004). Sources can be internal and external (not provided by the producer of the form) registers. An example of collection of internal information is the possibility of fetching data saved earlier by the user in a background form, which was mentioned in Section 3.2.3. Another example is a form containing a list of companies. When the user selects a company, fields will be filled with information regarding the specific company. Such information can be held in internal and external registers. Automatic fill-in of forms reduces the work task of the user and, in some cases, also the frustration of for instance having to repeatedly enter personal information. It also eliminates errors and supports users with cognitive challenges like writing, remembering, etc.

3.2.7 Semantics

When designing forms, one has to consider their semantics. Used in connection with forms and electronic forms, the term ‘semantics’ can cover several areas. One has to do with the intended meanings of fields, decided by the form producers (Gouscos et al., 2002). The end users can misunderstand the meanings of the fields, thereby giving the wrong input. Form producers often try to decrease the chance of this, for example by explanatory text in addition to the labels of the fields.

Another kind of semantics is that end users can express the same meaning in different ways (Gouscos et al., 2002). For example, one can state one’s sex as ‘female’ and ‘woman’. This semantics problem may be prevented by using the proper controls of e-forms (like the problem stating which sex one belongs to can be avoided by using radio buttons with the alternatives ‘male’ and ‘female’), tip text (text that appears when moving a pointer over a particular area) or specifying which terminology to use in the fields’ labels and validation checks. Both of these problems with semantics can make it harder to process the results for the form producers, as they will need more time to interpret them.

These two problems with the semantics can apply to both paper and electronic forms. Only electronic forms can exchange information online with external entities. When exchanging information, it is necessary to have common notions of the types of information exchanged. Problems occur when entities express the same elements differently, such as ‘postCode’ and ‘zipCode’. This semantics problem does not influence the forms’ user interfaces, unlike the others.

3.3 Experiencing Forms

Users can perceive a form as easy to answer, difficult to answer, taking a large effort to answer, etc. In the DIADEM-project (DIADEM, 2007), the research team has developed a form complexity indicator. This indicator is composed of several variables, such as number of questions (including sub-questions) in a form; maximum number of possible responses; number of responses requiring additional information; number of drop-down boxes in a form; number of drop-down boxes that require scrolling, etc. Complexity of electronic forms and the consequences of this matter to the users are fields that we yet know little about. However, there is now progress regarding these issues, as projects that investigate services considering the elderly and cognitive disabilities, like the DIADEM-project, address these areas.

Statistics Norway's handbook for development of web forms suggests that the users' perception depends on the form's number of questions, what the questions of the form concern, the structure of the form and its design (Haraldsen, 2004):

The Number of Questions

Counting the number of questions measures the extent of the form. Although, the number of questions might not give a full picture of the extent, as the questions are different. For example, some questions can ask the users to cross of every alternative in a long list that apply to them, while others simply ask the users to choose one of two alternatives. An alternative is to use a technique from interview surveys; to count a sequence of questions that takes thirty seconds to ask and get an answer to, as one question (Haraldsen, 2004).

What the Questions Concern

(Haraldsen, 2004) states that the problems occurring in connection with the questions can be divided into three areas:

- *Language*; the questions are formal and impersonal; forms use terms that need explanation; a rude tone – demanding rather than asking.
- *Problems with the tasks*; the users often need to do calculations to find the answers.
- *Problems with scale and level of detail of the answers*; the scale required in the form is often not used by the respondent; the form requires the answers to be more detailed than possible.

The form producers must plan carefully when making the questions, and try to avoid the three problem areas.

The Structure of the Form

The structure of a form can make the form seem easier to fill in, as mentioned in section 3.2.3. The users' first impression of the form usually marks the respondents' attitude towards answering the rest of the form. Therefore, according to (Haraldsen, 2004), it is important to have a clear relation between the form's title and the first questions (the title should imply what the questions concern) and to not make the first questions too difficult to answer. (Haraldsen, 2004) state that is also important to structure the following questions in a way that seems natural and gives the users the feeling of finishing the form step by step.

The Design of the Form

A form's design can make the form easy and inviting to read, and direct the users' attention to specific areas. There are several established design guidelines for paper forms, but as electronic forms need to have formats according to the constraints of their media, such as limited space for presentation, some of these guidelines can either not be used, or need to be applied differently on e-forms. Forms that use other text and graphics conventions than what the users has grown accustomed to through their operative systems and web browsers, may be difficult to answer for new users. (Haraldsen, 2004)

3.4 Forms in the Future

Statistics Norway's handbook for development of web forms describes the following development of forms (Haraldsen, 2004):

- The forms will have windows, menus and dialogue boxes.
- The questions will be presented by using sound and video, in addition to text.
- The respondents can mark their answers by fields that can be pushed and switches, by moving objects on the screen or by reading the answers into a microphone. This is of course in addition to the existing controls for data entry.
- As the transfer rate improves, more advanced controls of the answers can be made. It will be more common that the forms do consecutive comparisons with information from registers.
- The forms will increasingly simulate the dialogue of interview surveys.

Some forms today have several of these means. For example are menus and dialogue boxes not uncommon to see in e-forms. The development described above is for computers. For other devices, the development may turn out to be otherwise.

3.5 Representing Forms on Mobile Phones

The experiences and research on electronic forms is mostly based on representing forms on computers, with large screens and keyboards with many different keys for input. How about representing forms on mobile phones – is that feasible?

Many forms have to present a lot of additional information necessary for completing the form, and require a lot of input information. Compared to computers, mobile phones have a rather limited display size, and often a simple keypad and no pointing device. They are handheld devices, requiring that the user uses at least one hand to hold and operate the device. Inexperienced users often need to use both hands. Such limitations affect how and what for the devices can be used.

3.5.1 Semi-Structured Theme Interview

More AS (More Optimized Registration Elements AS) is a company that develops and sells software for design and management of electronic forms and surveys. An interview with Mr. Frode Preber Ettesvoll, the CEO of More AS, provided opinions of a form expert and a representative of the business world, on which forms may be feasible on mobile phones, and how this may be accomplished. The company has not focused on developing forms for mobile phones.

What kind of services may be feasible on mobile phones?

We reckoned that not all forms are very usable when presented on mobile phones, and therefore asked for his opinion on which forms may be suited to present. Mr. Ettesvoll started by telling that the company's experience concerning forms on mobile phones, is that the users cannot handle more than 3-4 form pages. Regarding forms larger than this, he therefore thinks that it would be beneficial to be able to start the form-filling process on a mobile phone in some scenarios, like when suddenly discovering that one has forgotten to fill in and deliver a form, but that the process should be completed on a device with a "normal-sized" keyboard and display, and a pointing device.

Mr. Ettesvoll said that feasible services on mobile phones, without using a computer in addition, can be some internal services in organisations, like employees notifying the employer of their sick leave and reports from employees working in the field, who have to deliver short, similar reports on a regular basis. Organisations can also benefit from that their employees can do a task, and report the results while taking on their next task. Midwives are an example of a group that can get more time for patients, if they can use mobile phones or other mobile devices for their form procedures, according to Mr. Ettesvoll. He also thinks reporting unemployment is feasible on mobile phones, as unemployed people have to deliver short reports to the social services every fortnight in order to get unemployment benefit. Reporting sales numbers (related to value added tax) to the authorities and short user surveys are also services that he considers to be feasible on mobile phones.

Which services could be "killer applications" on mobile phones?

We asked Mr. Ettesvoll whether there are any form services that may stand out when considering the usability of their representation on mobile phones. Mr. Ettesvoll mentioned the following examples:

- Notification of sick leave: He thinks that using one's mobile phone for notifications of sick leave would be much more practical than reporting via a computer, because the employee may find starting the computer, logging in, and so on, exhausting or time-consuming when sick, caring for a sick child, etc.
- Unemployment reporting: The reporting can today be done by delivering a paper form, or an electronic form on the Web. Since many own a mobile phone but not quite as many own or have access to a computer, and these are very short forms, he thinks that unemployment reports are particularly suited for mobile phones.
- Reporting sales numbers: The regular reports of sales numbers from companies to the authorities contain a few numbers, which could just as easily be reported using a mobile phone.

Which services are not feasible to offer on mobile phones?

Mr. Ettesvoll agreed to that not all forms are applicable on mobile phones. He said that the public authorities have several forms which he claimed are to complex to represent on mobile phones, because of the phones' limited display size and awkward input possibilities. He gave some examples of such forms: The tax return, several of the applications for financial support, and the application for division of landed property.

Who could be the users of forms on mobile phones, and which forms would they be using?

Mr. Ettesvoll thinks that everyone could be using them, as citizens reporting to or using the services of the authorities, employees to employer, and companies to the authorities. Even pupils to their school, as the schools are starting to require reporting.

In which situations would people be using their mobile phones for form-filling?

One of the examples that Mr. Ettesvoll stated, is when someone is sitting in the sofa, suddenly remembering that he or she has forgotten to fill in a form. The other example is people at work in the field. He cannot imagine anyone sitting on the bus filling in a form, but added that he himself does not like to use the phone when using public transportation, although that refers to using the phone for *phone calls*.

What are critical factors for success of forms on mobile phones?

Mr. Ettesvoll stated three factors for success. The first one is that "the time to load a form's elements must not be long", if it takes more time to fill in a form on a mobile phone than on a computer, people will rather use the latter. The second is that "it is important that the graphical user interface of the forms is well designed", and the third is that "the resolution of the screen of the user's mobile phone must be high", as that makes a huge difference for the representation of a form.

What are design-related criteria for success?

The CEO stated that the look makes the entire difference. In order for people to use forms on mobile phones, and for service-providers to buy forms from companies like More AS, the forms have to have an appealing look, he claimed. They can have the highest usability possible, but without an appealing look, no one will use them.

Are there any form controls or other elements which can be beneficial?

Mr. Ettesvoll mentioned several elements which he thinks may be of advantage in a form user interface on mobile phones. He said that a cursor which can be moved around on the screen is beneficial, but most important is the use of pre-filling of fields and drop-down boxes. Forms on mobile phones should offer selection over text input to the greatest extent possible, to spare the user from writing using the phone's keyboard/keypad.

Are there any form controls or other elements which can have a negative effect?

Mr. Ettesvoll could think of one element which should be avoided: text areas. Making text input is troublesome; therefore the use of text areas should be very restricted.

Can use of multimodality be beneficial in forms on mobile phones?

We asked this question in order to find out how an expert considers use of multimodality on mobile phones, our main area of focus. Mr. Ettesvoll thinks that presenting the same message in different ways is beneficial for the users, regarding the usability perspective. Figures, sound and careful use of colours can better make the users understand the content, as well as occupy less space on the screen. The mobile phone's vibration opportunity can also be exploited in forms by using it to signal wrong input, etc., he added.

What do you think of the future for forms on mobile phones?

He thinks that the future looks promising. Mr. Ettesvoll said that the next, important step is to find good business models, in order to persuade more service providers to make the mobile phone one of their service channels.

These opinions and comments by Mr. Ettesvoll gave valuable input to considering how forms can be made successful on mobile phones. Some of this input was used when making the success criteria stated in the following chapter.

3.5.2 Success Criteria

The expert's opinion was that representation of large forms, particularly forms that require a lot of text input, is likely to not be particularly feasible because of the phones' limitations. The limitations are also the reason to the advantage of the mobile phones; they make them *mobile*, thereby being possible to use any place at any time. This is the main reason to why people would want to use their mobile phone to fill in a form.

The mobile use context has some specific characteristics. Mobile phones are used "on the go", or when we have short idle moments. Whenever we have more time and access to a computer, the computer will be preferred because of its large screen and keyboard, i.e. we "dedicate" time to the computer. The mobile phone, on the other hand, gets our attention when we have a moment available. Disturbances often make our focus change, and we also often have to focus on the mobile phone at the same time as were focusing on something else (Gong et al., 2004). The device itself can suddenly "take us out of" what we are currently doing because of an incoming phone call, message, calendar alert, etc. These characteristics of the mobile use context, in addition to the limitations of the devices, set some criteria for services in order to be successful on mobile phones. Suggested criteria are stated below. Some of them refer to design principles and guidelines which can be found in Appendix A. The research hypothesis in Section 1.2.2 stated that "specific techniques and form controls can make forms feasible on mobile phones and relieve the burden of the users". For each criteria, such techniques and controls are listed. We do by no means state that this list of success criteria is complete, nor the suggested efforts on how to achieve them.

It must be possible to pause the form-filling process

As mentioned above, our use of mobile phones is in short idle moments, and often in environments with disturbances which sometimes require full attention. Sometimes, the device can interrupt what we are doing. In such cases, it must be possible to exit the form without losing any data, and to load it again without having to reenter the information, as stated in Guideline 12 "Design for speed and recovery" (Appendix A, Chapter 1.1). When reloading the form, one should be taken directly to the last edited location, as if a pause had been made. This will spare the user from searching for this location.

A service must not be too large and complex

A form should not be large, both because of the device limitations, as mentioned several times earlier, and also because as the size of the form increases, the complexity increases, which is not in favour in a mobile use context. It must be possible to get an impression of what a form requires and what it is about by casting a short glimpse on it, and it must not take a huge effort to acquaint oneself with the form-filling status, if one is returning to the form after a short or long break. The disturbances in a mobile environment also require that the complexity of a form should be minimal, as one should be able to, to some extent at least, fill in the form with a split attention. From this we can conclude, like Mr. Ettesvoll (cf. Section 3.5.1), that not all forms are feasible on mobile phones. A form that is offered on a mobile phone should suit the mobile use context as well as the device limitations in order to be successful.

There must be a limited intensity of information

Forms which present large blocks of information necessary for filling in the form, are probably not likely to be successful on mobile phones, because of the limited display size. When the screen only contains a block of information (or a large part of it does), the user may easily lose the form context. Large amounts of information on a small screen will also make the user having to focus on reading the information, which again is in defiance of the premises of the mobile use context. Forms with such information intensity should either (1) be regarded as not suitable for mobile phones, or, if possible, (2) one can do as suggested in Guideline 13 "Design for 'top-down' interaction" in Chapter 1.1 in the appendix: Present high levels of information and let the users retrieve the details. In that way, the user can – at least to some extent – see where the information belongs in the form context.

Offer services to appropriate target groups

Large companies have their own accountants who do the reporting to the authorities. Many smaller companies hire accountant firms to keep their accounts and do their reporting. The bookkeeping and reporting of such companies is usually done by using professional systems. These companies should therefore not be considered as a target group for report forms on mobile phones, as professional systems are very complex. It is probably best to consider private individuals and small companies as the target users for such services. However, when offering a company's internal services on mobile phones, like Mr. Ettesvoll suggested (see Section 3.5.1), the target groups may be large companies rather than very small ones.

The service must not stop the form-filling process when loading and sending information

A form client on the mobile phone may have to get its data from a server. Some forms can also fetch information from external sources, i.e. other systems. This takes more time than just fetching the data from the form client's server. If people have to wait long periods of time for data in order to continue the form-filling process, they are likely to not use forms on mobile phones. This can be solved by keeping the amount of data fetched from a server to a minimum, and/or not fetch all data at once. In the latter case, one can fetch data required for a form page first, and while the user fills in this page, the client fetches more data. In addition to requests regarding which information to retrieve, electronic forms can also often, e.g. for every form page, send a lot of information to the form server for verification. A more time-saving approach may be to let the form client do as much of the form fields verification as possible.

The service must have a user interface that accommodates the device limitations

Entering words on a mobile phone can be tiresome, particularly if the words are not in the phone's dictionary; third-party software does not provide a dictionary; there are many words to be entered, etc. Mr. Ettesvoll suggested to relieve the user's task by offering selection over text input, which is also suggested in guideline 10 "Design for small devices", in Chapter 1.1 in the appendix, in Guideline N-2 "Selecting instead of typing" in Chapter 1.2 in the appendix, and also in Statement 5.5.1 "Keep the number of keystrokes to a minimum" in Chapter 1.3 in the appendix. Offering selection can be done by using controls like list and drop-down boxes, radio buttons and check boxes. When use of selection controls is not an alternative, three other techniques can be used: pre-filling of fields, dictionary and a history database. Pre-filling is described in Section 3.2.6. A dictionary suggests words as the user writes letters in a text field, text area or combo box. A form client can have a history database that stores earlier input. Text fields, text areas and combo boxes can use a history database in the same way as they can use a dictionary. These techniques will make entering input easier.

Voice input combined with speech recognition or manual translation of speech to text is also a possible way of inputting information in applications. Although, voice should be an *alternative* input mode (making the form multimodal) rather than the only input mode, as inputting information by voice may be at the expense of the users' privacy. Voice input will also make filling in a form when one's hands are elsewhere occupied, like when driving, possible. Voice as an input mode has at least one issue: Information stated in forms need to be correct. If the form uses speech recognition software, users may worry about whether the conversion really gave the information they stated. If the speech recognition software is on the phone, they can check it by reading the result. In cases where the audio must be sent to a server and converted, as when using third-party speech recognition software that is to resource demanding and too big for phones, sending the audio file, translating it and sending the text back may take too much time for mobile context users. A way to avoid the last problem is to manually translate the speech to text, users can check their input by listening to the recorded audio, and yet be confident about the correctness of the translation. These solutions may not be acceptable to all users, but for users with some types of disabilities it may be of great help.

Some of the form controls presented in Section 3.2.5 may occupy a lot of a mobile phone's screen, which can decrease the user's conception of the form context. This can be solved by using controls like spin-buttons and drop-down boxes. A usability issue with these controls is that the user will not see more than one alternative in a long list, which can have a negative effect. Activating a drop-down box will however lead to the unfolding of a number of alternatives, while a spin-button will never present more than one alternative. Usability of controls versus loss of context must be considered when choosing form controls.

In Chapter 2.2 in Appendix A, it is advised to use check boxes when the number of options is one or five to seven, and to organise them in rows and columns. It may be easier to navigate up and down in a list box than to navigate in rows and columns of check boxes with a mobile phone keypad, but it may also be confusing to be supposed to choose several elements in the list box. Besides, a list box will occupy more screen space (in height) than check boxes, as a list box does not have columns. Here, usability of controls versus loss of context *and* usability of controls must be considered. Radio buttons are also supposedly most suitable when the number of options is five to seven, organised in rows and columns. Regarding the navigation issue, it may be best to use radio buttons when all of the options can be placed in one row, and use a drop down box otherwise. In other words, usability of controls versus loss of context must be considered also here.

Following other principles and guidelines for design of user interfaces on mobile devices will also accommodate device limitations of mobile phones. Chapter 1 of Appendix A contains three different sets of such guidelines. Several of the guidelines have already been mentioned. Very many of the guidelines in the sets are relevant when accommodating device limitations. They mostly concern navigation, input and presentation of content: keep the navigation consistent and uncomplicated, keep the keystrokes needed to a minimum and make the content of the screen easy to perceive and surveyable, i.e. **keep everything simple**. The relevant guidelines are not listed here as they are so many, and their full presentation is easily available in the appendix.

The service's user interface should follow guidelines for design of forms

Using form design principles and guidelines should increase the usability of forms. However, not all of the principles and guidelines may be applicable to mobile phones, because of the phones' user interface. Chapter 2 in Appendix A contains some sets of guidelines and advices on form design, mainly for representation of forms on computers. Most of these are likely to be increasing the usability, as the type of device will have no inflictions on the application of them. These advices and guidelines may not fall into that category (form controls were commented on above this section): Advice 1 "Separate the form from form activities" in Chapter 2.4 and Guideline 1.1 "The Navigation Area" and 1.2 "The Information Area" in Chapter 2.5 in the appendix deal with the same: A form should consist of several areas, where one area contains the form input controls, another tells the users where they are in the form, another contains supplementary information or help text related to the current input controls, other areas should contain other elements. The limited screen size may make the information intensity too high, thereby decreasing the usability.

The service's graphical user interface must be visually pleasing

Mr. Ettesvoll stressed the fact that a mobile form's graphical user interface must be appealing. This statement is supported in Guideline 15 "Design for enjoyment" in Chapter 1.1 in the appendix, which states that aesthetics is influential to mobile application success.

A service must accommodate variations in skills and abilities

People have different preferences, according to their skills and abilities, also stated in Guideline 14 “Allow for personalization” in Chapter 1.1. Making shortcuts for experienced users to make a service easier to use for them is one thing, to accommodate the variations in abilities of users is another, highly important issue. All types of users have to be accommodated in order for all of them to be able to use a service. Of course, that is not possible, and that is where the philosophy of Universal Design comes in. Universal Design is described in the next chapter.

As we chose to focus on how to make user interfaces of forms on mobile phones usable, technical solutions for achieving the success criteria “It must be possible to pause the form-filling process” and “The service must not stop the form-filling process when loading and sending information” are of no interest in this investigation. The success criteria “Offer services to appropriate target groups” and “The service’s graphical user interface must be visually pleasing” were also not further treated. The former because the target groups are highly dependent on the type of form, and the latter because that is a task for aesthetics designers. Efforts to achieve the remaining success criteria were demonstrated on a form service, illustrated in Chapters 7 and 8, and evaluated by users in Chapter 9.

4 Universal Design

The field of accessibility design, often labelled as 'Universal Design' in USA, and 'Design for All' in Europe, has gained a lot of attention over the last years. The field arose in the wake of a shift of focus: It became important to *include* disabled people in the community, rather than institutionalise them and tailor solutions for them. Universal Design/Design for All is also called "Universal Usability", "Inclusive Design" and several other terms. This report uses the term Universal Design, as it is increasingly used by the Norwegian government (Brynn et al., 2006). This chapter gives a brief introduction to the field. A more detailed elucidation on the field, also including an overview of a number of efforts done in the context of mobile phones, can be found in (Flaten, 2006).

4.1 What is Universal Design?

Ron Mace can be regarded as the inventor of the term 'Universal Design'. He defined it as follows (The Center for Universal Design, 2007):

"…the design of products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design."

The Center for Universal Design further adds:

"The intent of universal design is to simplify life for everyone by making products, communications, and the built environment more usable by as many people as possible at little or no extra cost. Universal design benefits people of all ages and abilities."

Like the definition says, universal design is not about making special solutions for the disabled. It is to consider what makes products, services and environments unusable to people with all kinds of abilities, and then try to design them for as many users as possible ("to the greatest extent"). As there will always be someone with a combination of disabilities that makes them unable to use a specific product or service, or function "normally" in a certain environment, the reservation of considering as many as possible must be made. The reservation is also made because of the usability perspective: Although a product theoretically can be made accessible for everyone, the usability of the product may not be satisfying for absolutely all. Return on investment is also an issue, as the cost of accommodating very many may not be covered in the earnings from the product or service, which makes accommodating a smaller number of disabilities necessary.

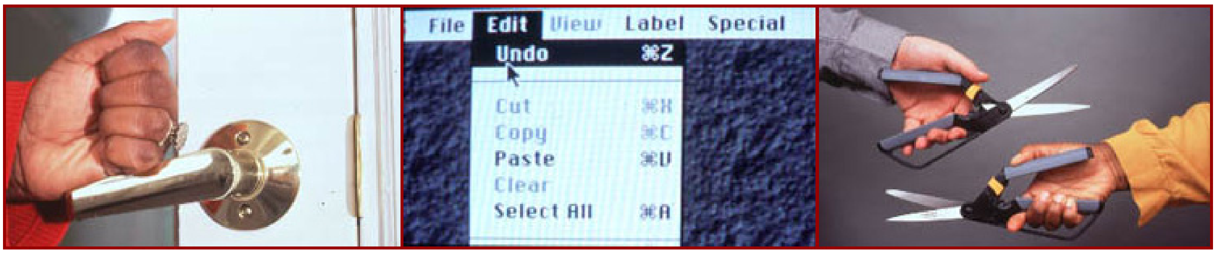


Figure 18. Applications of Universal Design (The Center for Universal Design, 1997).

4.2 Why Design Universally?

There are several motivations for designing universally. The most universal of these can be found in different policy documents. The first article of "The Charter of Fundamental Rights of the European Union" states that

Human dignity is inviolable. It must be respected and protected. (Official Journal of the European Communities, 2000)

The United Nations' "Declaration on the Rights of Disabled Persons" from 1975 has a similar statement for people with disabilities:

Disabled persons have the inherent right to respect for their human dignity. Disabled persons, whatever the origin, nature and seriousness of their handicaps and disabilities, have the same fundamental rights as their fellow-citizens of the same age, which implies first and foremost the right to enjoy a decent life, as normal and full as possible. (Office of the United Nations High Commissioner for Human Rights, 1975)

Dignity is stated to be very important, and should therefore be a strong motivation. People with disabilities may not want to be referred to special solutions, or even not at all have access to the same products, services and environments as everyone else.

There are also several other motivations for Universal Design. One motivation for the authorities is that the amount of money spent on welfare can be less, as designs make people with disabilities more independent and integrated. A motivation for the vendors is that a product or service that accommodates people with different abilities has more potential buyers and can thus remove or decrease other costs, such as resources spent on help-desks, etc. Another motivation for them is that efforts to achieve usability also for people with disabilities can raise the public opinion of a company and make its brand name more acknowledged. (Telematics Applications Programme, 1998)

All users can benefit from Universal Design, regardless of abilities. For example, people in very loud environments may benefit from a design that also accommodates people with hearing impairments. Another example is people who are in a dark environment or someone who has to focus on a particular task or point (like when driving). These may benefit from design considering people with visual impairments.

4.3 The Principles of Universal Design

'The Principles of Universal Design' developed at The Center for Universal Design at North Carolina State University, seem to be the most quoted and acknowledged Universal Design principles. The principles can be used to "evaluate existing designs, guide design processes and educate designers and customers about the characteristics of more usable products and environments" (The Center for Universal Design, 1997). In addition to the principles, the authors have provided definitions and guidelines for each of them[2]:

[2] Copyright © 1997 NC State University, The Center for Universal Design. The Center for Universal Design does not take any responsibility for an individual's or organisation's use or application of the principles in any form.

1: Principle One: Equitable Use
The design is useful and marketable to people with diverse abilities

GUIDELINES

- Provide the same means of use for all users: identical whenever possible; equivalent when not.
- Avoid segregating or stigmatizing any users.
- Provisions for privacy, security, and safety should be equally available to all users.
- Make the design appealing to all users.

2: Principle Two: Flexibility in Use
The design accommodates a wide range of individual preferences and abilities.

GUIDELINES

- Provide choice in methods of use.
- Accommodate right- or left-handed access and use.
- Facilitate the user's accuracy and precision.
- Provide adaptability to the user's pace.

3: Principle Three: Simple and Intuitive
Use of the design is easy to understand, regardless of the user's experience, knowledge, language skills, or current concentration level.

GUIDELINES

- Eliminate unnecessary complexity.
- Be consistent with user expectations and intuition.
- Accommodate a wide range of literacy and language skills.
- Arrange information consistent with its importance.
- Provide effective prompting and feedback during and after task completion.

4: Principle Four: Perceptible Information
The design communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities.

GUIDELINES

- Use different modes (pictorial, verbal, tactile) for redundant presentation of essential information.
- Provide adequate contrast between essential information and its surroundings.
- Maximize "legibility" of essential information.
- Differentiate elements in ways that can be described (i.e., make it easy to give instructions or directions).
- Provide compatibility with a variety of techniques or devices used by people with sensory limitations.

5: Principle Five: Tolerance for Error
The design minimizes hazards and the adverse consequences of accidental or unintended actions.

GUIDELINES

- Arrange elements to minimize hazards and errors: most used elements, most accessible; hazardous elements eliminated, isolated, or shielded.
- Provide warnings of hazards and errors.
- Provide fail safe features.
- Discourage unconscious action in tasks that require vigilance.

6: Principle Six: Low Physical Effort
The design can be used efficiently and comfortably and with a minimum of fatigue.

GUIDELINES

- Allow user to maintain a neutral body position.
- Use reasonable operating forces.
- Minimize repetitive actions.
- Minimize sustained physical effort.

7: Principle Seven: Size and Space for Approach and Use
Appropriate size and space is provided for approach, reach, manipulation, and use regardless of user's body size, posture, or mobility.

GUIDELINES

- Provide a clear line of sight to important elements for any seated or standing user.
- Make reach to all components comfortable for any seated or standing user.
- Accommodate variations in hand and grip size.
- Provide adequate space for the use of assistive devices or personal assistance.

In connection with this investigation, these principles have been treated as the background theory for design considerations and decisions documented in Chapters 7 and 8.

4.4 Universally Designed Forms

Chapter 3.5.2 stated that a success criterion for form services on mobile phones should be to accommodate variations in skills and abilities, thereby designing forms to have a Universal Design. As mentioned in Chapter 1.1, people are now living longer, which is resulting in more people getting age-related diseases. In other words, the number of people with disabilities is increasing. Depending on how profound their disability or disabilities are, these people are a part of the modern society equally to everyone else, and their needs should be provided for in products and services offered to the population without hurting their dignity by treating them differently.

One can not accommodate the needs of absolutely everyone, as mentioned in Chapter 4.1. There will always be someone with a combination of disabilities which makes it impossible to provide for their needs, or, if that was possible, meeting everyone’s needs could decrease the overall usability for a large number of the users. Return on investment is also an important issue for producers and providers of products and services, which cannot be ignored.

This implies that, in order to achieve the success criterion “A service must accommodate variations in skills and abilities” (Section 3.5.2), mobile form services should implement the principles of Universal Design in a way that accommodates the needs of as many as possible. Finding 'as many as possible' here means that one first has to calculate the cost (in form of decrease in usability, loss of income and increased expenses) of accommodating "everyone", then decide who to provide for. When designing public e-forms, only decrease in usability and increased expenses should be relevant, as the authorities do not have to earn money from the services they offer.

4.5 Universal Design for Cognition

This investigation focused in particular on accommodating people with cognitive disabilities which affect their literacy and numeracy skills, as stated in Section 1.2.3. The book "The Universal Design File, Designing for People of All Ages and Abilities" published by The Center for Universal Design dedicates a section to Universal Design and variation in cognition abilities. About designing for cognition, the authors state (Story et al., 1998):

Universal design for cognition means considering the variety of human abilities in receiving, comprehending, interpreting, remembering, or acting on information. This includes:

- self-starting; initiating tasks without prompting
- reacting to stimuli; response time
- paying attention; concentration
- comprehending visual information
- comprehending auditory information
- understanding or expressing language
- sequencing; doing things in proper order
- keeping things organized
- remembering things, either short- or long-term
- problem-solving; decision-making
- creative thinking; doing things in a new way
- learning new things

That is, to consider all of the problems people with cognitive disabilities can have with a product, service or environment. In order to evaluate whether a design is accommodating variations in cognition abilities, the authors suggest answering the following questions (Story et al., 1998):

Is the design still as usable and safe if you...

- are using it for the first time without help or instructions?
- cannot read?
- perform steps out of order?
- try to use it much faster or slower than intended?
- make a mistake and want to correct it or start over?
- are distracted or interrupted while using it?

In our case, two of the bullet points in the upper section stand out when designing mobile services and accommodating the focus groups. **To consider the ability of comprehending visual information, and the ability of understanding language**. Although, considering the rest of the points may also affect the usability for the target groups. For instance, the ability to solve problems may affect one's ability to put words together into sentences that make sense.

None of the points in the lower section are really appropriate to use to evaluate whether a form's design is accommodating the target groups. Evaluating the design considering people who can not read is a bit drastic, as people in the literacy target group experiencing problems when reading, are not unable to read. One should simply evaluate whether the design accommodates the problems people in the target groups have, to see if the design is usable and safe.

5 Cognitive Disabilities

This chapter briefly describes three main types of disabilities and more thoroughly the 'cognitive disabilities' category, the results from the Adult Literacy and Life Skills Survey (OECD, 2005), and two conditions, one from each of the target groups. The last part of this chapter sums up the principles and guidelines in Chapter 3 of Appendix A that are relevant for our target groups.

5.1 Types of Disabilities

There are several types of disabilities, in addition to the cognitive. This section briefly renders the disability categories presented in (Flaten, 2006). The categories are from the report "Accessible Design of Consumer Products: Guidelines for the Design of Consumer Products to Increase Their Accessibility to People with Disabilities or Who Are Aging" by Katherine and Gregg Vanderheiden (Vanderheidens, 1991).

Visual Impairments

People with visual impairments have problems in situations where information is displayed visually. Visual impairments comprise everything from "…very poor vision, to people who can see light but no shapes, to people who have no perception of light at all" (Vanderheidens, 1991). 3% of the Norwegian population were registered as having "Reduced eyesight" in 2002 (Statistics Norway, 2006, StatBank), while the European percentage was 2,8 with "Difficulties in seeing" in 2002 (Eurostat, 2003).

Hearing Impairments

The hearing-impaired experience problems when important information is auditory. The Vanderheidens separate between having a hearing impairment and being deaf; "Hearing impairment means any degree and type of auditory disorder, while deafness means an extreme inability to discriminate conversational speech through the ear" (Vanderheidens, 1991). That is, people with hearing impairments can use their hearing when communicating and interacting in the society, whereas the deaf can not. In Norway, 3% percent of the population was registered with "Reduced hearing" in 2002 (Statistics Norway, 2006, StatBank), while 2% of the European was registered with "Difficulties in hearing" (Eurostat, 2003).

Physical Impairments

People with physical impairments have limited or no ability to manipulate their body, or part(s) of it. According to (Vanderheidens, 1991), the category include people with "…poor muscle control, weakness and fatigue, difficulty walking, talking, seeing, speaking, sensing or grasping (due to pain or weakness), difficulty reaching things, and difficulty doing complex or compound manipulations (push and turn)". Statistics do not focus simply on the category "physical impairments", but on types of physical impairments, making it difficult to get a correct number or percentage – as many have several physical impairments and thereby can be counted in multiple categories.

Cognitive/Language Impairments

Those who have a cognitive/language impairment may have difficulties in recognizing and retrieving information, solving problems, comprehending written and/or spoken language, and/or expressing themselves. Cognitive impairments comprise many types of disabilities, such as severe retardation, inability to remember, and absence or impairment of specific cognitive functions (usually language) (Vanderheidens, 1991). As for physical disabilities, the categories here are so many and so complex, making it hard to acquire a somewhat correct percentage of the population by adding them.

For the sake of simplicity, this report will not use the notion “cognitive/language impairments”, but rather use “cognitive impairments”, “cognitive disabilities”, etc. The next section further introduces cognitive impairments.

5.2 Cognitive Impairments

As stated above, cognitive impairments comprise problems with comprehension, recognition, problem solving and expressing oneself, i.e. to be cognitively impaired means that something in one’s brain does not function “normally”, or slightly out of the ordinary. Although the cognitively impaired have such problems, they can be of average or above average intelligence (Seeman, 2002).

Cognitive impairments can be present at birth, such as for those who have Down’s syndrome. People can also become cognitively impaired because of a head injury or illness: Aphasia, speech and language disorders and amnesia can be acquired in this manner. People can also be genetically disposed to acquire cognitive impairments when they are aging. (McGrenere et al., 2006) Alzheimer’s disease is a very serious and not so common example of such impairments, while less severe problems with one’s memory are more common, particularly among the elderly. People with profound cognitive disabilities need assistance with practically everything in their daily life, while others may be as little impaired, as to function so well that the disability is never discovered or diagnosed (WebAIM, 2007, Cognitive Disabilities).

Learning disabilities is often referred to as an own disability category, or as an own group within the cognitive disability category. Learning disabilities comprise reading and writing problems, such as the conditions dyslexia and dysgraphia, and computational disabilities in form of problems with numbers, such as dyscalculia, and other areas relating to numerical manipulation. Learning disabilities are sometimes diagnosed together with Attention Deficit Disorder and Hyperactivity (ADD or ADHD). (Kolatch, 2000)

5.3 The Adult Literacy and Life Skills Survey

The results from the Adult Literacy and Life Skills Survey showed, as mentioned in the introduction, that between one-third and over two-thirds of the population, depending on country, aged between 16-65, do not attain a skill level considered by experts as a suitable minimum level in the areas investigated. These areas are prose and document literacy, numeracy and problem solving. (OECD, 2005)

Prose Literacy

Prose literacy is defined as “*the knowledge and skills needed to understand and use information from texts including editorials, news stories, brochures and instruction manuals*”. This area is divided into five skill levels, where the first two are considered to be under the suitable minimum level, see descriptions of the two levels below. The percentage of the Norwegian population at these two levels is 7,9 at level one and 26,2 at level two, i.e. a total of 34,1% (OECD, 2005). This is a remarkable figure – in other words, over one third of the Norwegian population suffers from inadequate literacy skills.

LEVEL 1
Most of the tasks in this level require the respondent to read relatively short text to locate a single piece of information which is identical to or synonymous with the information given in the question or directive. If plausible but incorrect information is present in the text, it tends not to be located near the correct information.

LEVEL 2
Some tasks in this level require respondents to locate a single piece of information in the text; however, several distractors or plausible but incorrect pieces of information may be present, or low-level inferences may be required. Other tasks require the respondent to integrate two or more pieces of information or to compare and contrast easily identifiable information based on a criterion provided in the question or directive.

(OECD, 2005)

Document Literacy

Document literacy is defined as "*the knowledge and skills required to locate and use information contained in various formats, including job applications, payroll forms, transportation schedules, maps, tables and charts*". This area is also divided into five skill levels, see descriptions of the two levels under the suitable minimum level below. The percentage of the Norwegian population in these two levels is 8,9 at level one and 23,5 at level two, i.e. a total of 32,4% (OECD, 2005). Also here, wee see that more than one third of the Norwegian population suffers from inadequate literacy skills.

LEVEL 1
Tasks in this level tend to require the respondent either to locate a piece of information based on a literal match or to enter information from personal knowledge onto a document. Little, if any, distracting information is present.

LEVEL 2
Tasks in this level are more varied than those in Level 1. Some require the respondents to match a single piece of information; however, several distractors may be present, or the match may require low-level inferences. Tasks in this level may also ask the respondent to cycle through information in a document or to integrate information from various parts of a document.

(OECD, 2005)

Numeracy

Numeracy is defined as "*the knowledge and skills required to effectively manage the mathematical demands of diverse situations*". Numeracy is also divided into five skill levels, the descriptions of the two levels under the suitable minimum level are rendered below. The percentage of the Norwegian population in these two levels is 10,6 at level one and 29,6 at level two, i.e. a total of 40,2% (OECD, 2005). This figure also calls for attention. A majority of all self-services are based on numerical log-in procedures, e.g requiring users to enter PIN codes, numeric passwords, birth date and/or personal identification number. This may represent a major accessibility problem.

LEVEL 1
Tasks in this level require the respondent to show an understanding of basic numerical ideas by completing simple tasks in concrete, familiar contexts where the mathematical content is explicit with little text. Tasks consist of simple, one-step operations such as counting, sorting dates, performing simple arithmetic operations or understanding common and simple percents such as 50%.

LEVEL 2
Tasks in this level are fairly simple and relate to identifying and understanding basic mathematical concepts embedded in a range of familiar contexts where the mathematical content is quite explicit and visual with few distractors. Tasks tend to include one-step or two-step processes and estimations involving whole numbers, benchmark percents and fractions, interpreting simple graphical or spatial representations, and performing simple measurements.

(OECD, 2005)

Problem Solving

Problem solving is defined to involve "*...goal-directed thinking and action in situations for which no routine solution procedure is available. The problem solver has a more or less well defined goal, but does not immediately know how to reach it. The incongruence of goals and admissible operators constitutes a problem. The understanding of the problem situation and its step-by-step transformation, based on planning and reasoning, constitute the process of problem solving*". Problem solving is divided into four skill levels, where the two first are under the suitable minimum level, see descriptions rendered below. The Norwegian percentage in these two levels is 23,3 at level one and 37,5 at level two, i.e. a total of 60,8% (OECD, 2005). This is the most remarkable figure, more than half of the Norwegian population supposedly have inadequate problem solving skills.

LEVEL 1
Tasks in this level typically require the respondent to make simple inferences, based on limited information stemming from a familiar context. Tasks in this level are rather concrete with a limited scope of reasoning. They require the respondent to make simple connections, without having to check systematically any constraints. The respondent has to draw direct consequences, based on the information given and on his/her previous knowledge about a familiar context.

LEVEL 2
Tasks in this level often require the respondent to evaluate certain alternatives with regard to welldefined, transparent, explicitly stated criteria. The reasoning however may be done step by step, in a linear process, without loops or backtracking. Successful problem solving may require to combine information from different sources, as e.g. from the question section and the information section of the test booklet.

(OECD, 2005)

As the OECD report shows, the percentages of people with low skills in several cognitive areas important in the daily life, are quite high. Norway is the country with best average scores in all areas except for numeracy skills. Most of the people with unsatisfying scores in the areas have probably not been diagnosed to have a specific cognitive impairment as doing the survey on a large number of people with such conditions would give unrepresentative statistics, they may just have general low cognitive abilities. Low cognitive abilities can be due to a lack of education, age, etc. Some of the results also indicated that gender matters in numeracy skills (OECD, 2005).

5.4 Two Conditions

The areas investigated in the OECD report span several specific conditions; this section gives a brief introduction to the literacy condition dyslexia and the numeracy condition dyscalculia, which both in turn affects problem solving.

5.4.1 Dyslexia

The term ‘dyslexia’ comes from the Greek and means ‘difficulty with words’ (The British Dyslexia Association, 2007). The condition was earlier called ‘congenital word blindness’. Encyclopedia of Neurological Disorders provides this definition:

Dyslexia is an unexpected impairment in reading and spelling despite a normal intellect. (Encyclopedia of Neurological Disorders, 2007)

Dyslexia results from impairment of brain processes that convert letters into words. It reduces the ability to develop an awareness of spoken and written words, and to segment such smaller units of sound, which are essential in alphabetic languages like English. The dyslexics have a reduced ability to link and map printed letters to sound. (Encyclopedia of Neurological Disorders, 2007) Five to ten percent of the population have severe cases of dyslexia (The Norwegian Dyslexia Association, 2007).

The condition has several subtypes which manifest themselves differently. People with **semantic reading dyslexia** can only assemble a word’s pronunciation when they have assessed its meaning, they also make visual errors when reading. People with **non-semantic reading dyslexia** have problems with reading words that are not commonly used or that are not similar to common words. The attention control of people with **attentional dyslexia** is impaired, causing intrusion of letters from one word into another when several words are visible to them at once, thereby making their ability to read words in sentences weakened. Those with **left neglect dyslexia** have problems with reading initial letters of words. Letters may be substituted, omitted or added. **Right neglect dyslexia** causes letter errors when reading the end of words. People with **letter-by-letter reading dyslexia** see words as random letter strings. (Encyclopedia of Neurological Disorders, 2007)

Dyslexia can not be cured, although the state of the condition can be improved. Early identification is important for improvement. Intervention programs teach how letters are linked to sounds and spelling. (Encyclopedia of Neurological Disorders, 2007) Software and gadgets can help dyslexic people in their daily life. Scanners can be used to scan text into a computer, where a text-to-speech program can read the text, or the user can change the text’s font and size to make it easier to read. Reading pens, see Figure 19, can also scan text and present the results on their display and/or send it to a computer with text-to-speech software. Reading pens are best suited for long and difficult words, short texts and explanations to pictures. Audio recorded books are also considered valuable to those who find reading text impossible or difficult. (The Norwegian Dyslexia Association, 2007, Remedies)

Dictionaries and spelling checks in text writing programs help dyslexic people write better. Lingdys is a spelling check program that knows the usual misspellings done by people with dyslexia, and that reads correction suggestions to the user. It also contains a dictionary with word explanations. Prediction programs can be used together with a text writing program, they try to predict what the user wants to write. When the user writes, they suggest several words based on the input. (The Norwegian Dyslexia Association, 2007, Remedies)

Figure 19. C-Pen, a pocket-sized scanner (The Norwegian Dyslexia Association, 2007, Remedies).

The content of text itself can be made easier to comprehend by presenting it using flowcharts, graphs, and diagrams (Murphy, 2005). The type of font used also affects readability for people with dyslexia. The British Dyslexia Association states that “fonts should be rounded, allow for space between letters, reflect ordinary cursive writing and be ‘easy on the eye’”. Such fonts are ‘sans serif’ fonts[3] like Arial and Comic Sans. They also suggest using Verdana, Helvetica, Tahoma, Trebuchet and Sassoon. (The British Dyslexia Association, 2007, Style Guide)

People with moderate dyslexia belong to one of the target groups we are considering when designing a form user interface with multimodality, presented in Chapters 7 and 8.

5.4.2 Mathematics Disorder (Dyscalculia)

Mathematics disorder was formerly called dyscalculia, development arithmetic disorder and developmental acalculia. Encyclopedia of Mental Disorders provides this definition:

Mathematics disorder ... is a learning disorder in which a person's mathematical ability is substantially below the level normally expected based on his or her age, intelligence, life experiences, educational background, and physical impairments. This disability affects the ability to do calculations as well as the ability to understand word problems and mathematical concepts. (Encyclopedia of Mental Disorders, 2007)

According to (Encyclopedia of Mental Disorders, 2007) the condition comprises several distinct types of mathematical deficiencies, such as:

- Difficulty reading and writing numbers
- Difficulty aligning numbers in order to do calculations
- Inability to perform calculations
- Inability to comprehend word problems

All causes of mathematics disorder are not known, different manifestations of the disorder may have different causes. Turner syndrome and fragile X syndrome are two genetic disorders that affect girls, and which are both associated with low numeracy skills. Injury to certain parts of the brain can also cause difficulties in performing calculations. (Encyclopedia of Mental Disorders, 2007)

[3] ‘Sans serif’, i.e. fonts without details (“serifs”) on the ends of the strokes that make up letters (Wikipedia, 2007)

Symptoms of mathematical disorder can be divided into four categories. The **language symptoms** category comprises trouble with naming mathematical terms, understanding word problems and mathematical concepts like “greater than” and “less than”. **Recognition symptoms** comprise difficulty reading numbers and operational signs (such as plus and minus), and aligning numbers in order to perform calculations. People with **mathematical symptoms** can have problems with counting, memorising arithmetical data (like multiplication tables) and following a sequence of steps in problem solving. Those with **attention symptoms** seem to have a lack of understanding of factors or operations involved in solving a problem, causing them to ignore operational signs and fail in copying numbers. (Encyclopedia of Mental Disorders, 2007)

Some of the people with mathematics disorder can be taught to overcome their disability, while others struggle with the condition throughout their lives. Improvements depend on the difficulties they have, resources available and their motivation to work through it. Learning programs for helping them can include emphasising basic mathematical concepts, teaching problem-solving and how to eliminate distractions and extraneous information, depending on the type of problems each individual has. (Encyclopedia of Mental Disorders, 2007) Such learning programs can use software developed for this purpose. Software and gadgets like text-to speech and scanners should also be of help to those who have trouble reading numbers.

People with moderate dyscalculia belong to the other of the two target groups we are accommodating when designing a form user interface with multimodality, presented in Chapters 7 and 8.

5.5 Designing User Interfaces for the Target Groups

To best meet the need of the target groups, people with moderate problems with reading text and numbers, using design principles and guidelines for disabled people may help. Chapter 3.1 of Appendix A contains guidelines for design of Web content for people with disabilities, while Chapter 3.2 contains guidelines for people with cognitive disabilities. Most of these should also be applicable to user interfaces of software for mobile phones. The following sections list the guidelines that seem relevant for the target groups.

Guidelines in Chapter 3.1 in Appendix A:

- Guideline 1 “Provide equivalent alternatives to auditory and visual content” is very relevant, if one implement it as providing equivalent alternatives to visual content. It is also relevant for visually impaired people using screen readers and hearing impaired, as all of the guideline’s checkpoints deal with using redundant text.
- Guideline 2 “Don’t rely on color alone” is somewhat relevant, as the use of colour on text and background will affect the readability of a text or numbers. One should therefore make sure that the contrast between text and background is at least “sufficient” as stated in the guideline’s second checkpoint.
- Guideline 3 “Use markup and style sheets and do so properly” affects a text’s layout, thereby its readability.
- Guideline 4 “Clarify natural language usage” is relevant for the same reason as number 3.
- Guideline 10 “Use interim solutions” is relevant for the people in the target groups who use screen readers, although they can manage without.

- Guidelines 12 “Provide context and orientation information” will enhance the comprehensibility of the user interface, which will affect the usability for people in the target groups.
- Guideline 13 “Provide clear navigation mechanisms” is relevant for the same reasons as number 12.
- Guideline 14 “Ensure that documents are clear and simple” is also relevant for the same reasons as number 12.

Guidelines in Chapter 3.2 in Appendix A:

- In guideline 1 “Create transformable, rich, multi-modal content” in 3.2 in the appendix, everything but point 1.2.2 is very relevant, as people who have trouble reading and understanding text and numbers, will not need a text version of a video.
- Guideline 2 “Focus the attention of the user” is also very important, as it will make the user interface facilitate understanding of the content.
- All of the remaining guidelines in Section 3.2 are relevant or somewhat relevant for the target groups.

These guidelines facilitate understanding of content by putting restrictions on the presentation of user interfaces with text, and by advising to represent the content in an alternative mode, or making it possible for the users to use screen readers. A well arranged layout is beneficial also for the healthy users, but may not help all of the users from the target groups, and to depend on using screen readers is not fortunate because of the mobile context with loss of privacy – although someone have to. For these reasons, and because multimodal user interfaces are acknowledged to be helpful to the cognitively disabled, as stated in section 1.2.2, *multimodality* is presented in the next chapter.

6 Multimodal Presentations

This chapter describes what multimodality is, how it affects cognition, and benefits and disadvantages of multimodal user interfaces. In the last part of this chapter, how to accommodate reading and writing by using multimodality, is discussed.

6.1 Multimodality and Multimedia

The terms 'multimodality' and 'multimedia' are closely connected. The ETSI guide containing multimodal interaction, communication and navigation guidelines, defines **multimodality** as

...property of a user interface in which:
a) more than one sensory modality is available for the channel (e.g. output can be visual or auditory); or
b) within a channel, a particular piece of information is represented in more than one sensory modality (e.g. the command to open a file can be spoken or typed). (ETSI, 2003)

Schwartz and Beichner defines **multimedia** as

…the use of multiple forms of media in a presentation. (Schwartz et al., 1999)

Both of the definitions state that multimodality/multimedia involves the use of more than one modality/medium, where multimodality concerns input and output while multimedia concerns modalities in output only. Regarding ICT equipment, such modalities are text, sound, graphics, animation, video, touch (mostly for the blind), and probably many more, i.e. all possible ways to present something or enter input.

The problem with information presented or gathered using only one modality, is that it excludes or makes things harder for the people with impairments in the sensors necessary to take in the information. This can severely affect their life, making even the simplest, daily tasks impossible to do. For instance, we are often required to type in a username and password when using a computer. People with some form of dyslexia or dyscalculia will struggle with this task, which will have a large impact on their daily life. Sensory impairments may restrict the life of persons in their education, job, social life and otherwise (taking the bus, walking on the street, in the supermarket, etc.). Multimodality/multimedia can to some degree compensate for this, by using several modalities when possible.

6.2 Multimodality and Cognition

That multimodality can help people who are blind or deaf is common sense, but multimodality may also improve learning and living for others, with and without impairments, by helping the cognition to process information. Wicken's Multiple Resource Theory proposes that the human brain has different cognitive resources which each process different modalities of stimuli (Wickens, 2002). Cognitive overload can become a problem when a large amount of information is presented by means of only one modality. Using multiple modalities enables multiple processing of information, thereby decreasing the cognitive overload (Wickens, 2002).

Mayer has developed a similar theory, the Cognitive Theory of Multimedia Learning (Mayer, 2001). His definition of multimedia is restricted to apply to the verbal and pictorial channel only. The theory, illustrated in Figure 21, is based upon three primary assumptions:

1. There are separate channels for the processing of verbal and pictorial information.
2. There is a limit to how much information these channels can process at a time.
3. The processing of the information, i.e. the learning, consists of three processes: (a) Selecting words and images for processing in their verbal and visual working memory, (b) Organising the words and images into a verbal and pictorial mental model, and (c) Integrating the verbal and visual representations and prior knowledge.

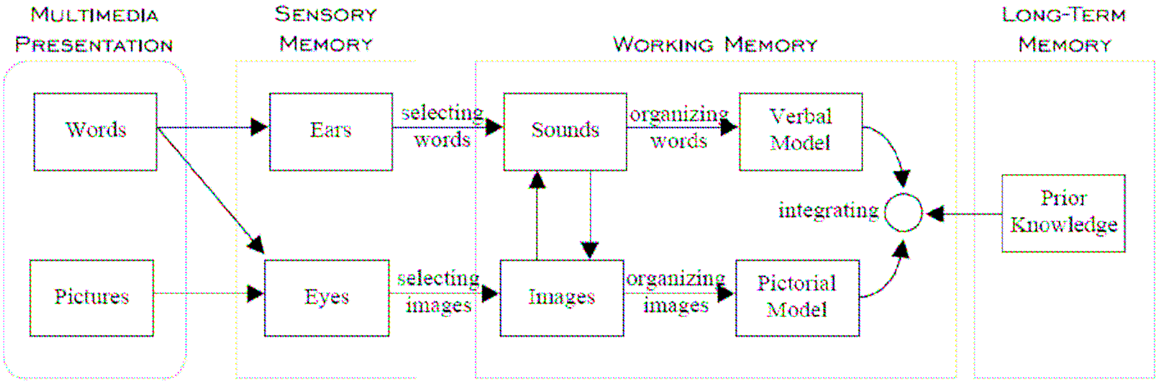


Figure 20. Cognitive theory of multimedia learning model (Doolittle, 2001).

The model led to several experiments, which resulted in five principles on how to use multimedia to better make students understand scientific explanations (Mayer et al., 1998):

Multiple Representation Principle
It is better to present an explanation in words and pictures than solely in words.

Contiguity Principle
When giving a multimedia explanation, present corresponding words and pictures contiguously rather than separately.

Split-Attention Principle
When giving a multimedia explanation, present words as auditory narration rather than as visual on-screen text.

Individual Differences Principle
The foregoing principles are more important for low-knowledge than high-knowledge learners, and for high-spatial rather than low-spatial learners.

Coherence Principle
When giving a multimedia explanation, use few rather than many extraneous words and pictures.

It is reasonable to believe that the principles can be applied to make also other types of explanations/information easier to perceive, rather than only scientific explanations.

According to (ETSI, 2003), in addition to having different abilities and needs, people also have individual ways of interpreting data, i.e. they have different cognitive styles. For instance, some interpret data best when presented in tables, others when presented in images and others again, when presented auditorially, regardless of health condition.

Use of multimodality is acknowledged to be of great help to people with cognitive impairments. (Newell et al., 2002) states that multimedia and multimodal systems "…may be particularly advantageous for users with cognitive dysfunction." (Jiwnani, 2001) advices Web page designers to provide multisensory presentation of feedback information, to increase the accessibility of Web pages for these users. (Bohman, 2005) states that presentation of Web content through many different sensory modalities increases the chance of the users understanding the content. (WebAIM, 2007, Design Considerations) states that *supplemental media*, like illustrations, icons, video audio, can enhance accessibility of web content to people with cognitive disabilities. (Burkhart, 2003) found that multimodal input and output supports autistic children's learning.

6.3 Benefits and Disadvantages of Multimodality

Multimodal interfaces may have benefits and disadvantages for people who are using them, depending on their implementation and the abilities and preferences of these people. This chapter summarises some of the benefits and disadvantages for everyone, and some of the benefits found for people with cognitive disabilities.

Benefits for everyone

- According to the theories of Wickens and Mayer rendered above, multimodal presentations improve the ability to process information (Wickens, 2002; Mayer, 2001).
- Combining diagrams and audiotapes has been proven to support expansion of working memory and problem solution in geometry tasks better than using only one medium (Mousavi et al., 1995).
- According to (Oviatt, 1997), when people had the ability to choose input mode (speaking and/or writing) they completed spatial map tasks with 50 percent fewer disfluencies and briefer and simpler linguistic constructions, and 36 percent fewer task-critical errors.
- Multimodal presentation of information lets users with different preferences and abilities interpret data in their preferred way (ETSI, 2003) thereby extending the range of potential users and uses (Reeves et al., 2004).
- Multimodality is also beneficial in situations where we lose one of our sensors, or are otherwise occupied, like when we are driving.

Benefits for people with cognitive impairments

- Multimodal interfaces are acknowledged to increase accessibility and understanding for people with cognitive impairments, as stated in Chapter 6.2.
- People who have problems with perception and processing and memory benefit from pairing of icons or graphics with text (WebAIM, 2007, Conceptualizing Design Considerations).
- People who have perception and processing, problem solving or attention problems, benefit from demonstrations and audio descriptions (WebAIM, 2007, Conceptualizing Design Considerations).

- Users with a limited textual comprehension, and others, benefit from making information as multimodal as possible, as it enhances understanding (Bartlett, 1999).
- Graphic representations and point and click interfaces remove the need of reading and writing skills (Kolatch, 2000).
- Flowcharts, graphs, diagrams, multimedia presentations and other modalities for presentation of information have proven to aid people with cognitive impairments (Murphy, 2005).

Possible disadvantages for everyone

- High quality media may be difficult to make. Using poor quality media may make Web content more confusing, and decrease the accessibility for people with cognitive disabilities. (WebAIM, 2007, Design Considerations)
- Extensive use of graphics and use of animations can distract and increase the cognitive load (Mariger, 2006; Bartlett, 1999; Jiwnani, 2001).
- If users must simultaneously attend to different modalities to comprehend the information presented, the cognitive load may be increased at the cost of learning the information (Reeves et al., 2004).
- We can also ask if making the users choose between many modalities is really helpful, as it may be confusing the users, particularly the users with cognitive impairments.

All of these disadvantages are likely to affect people with cognitive impairments more severely than the people without such impairments, as the disadvantages are related to the human cognition.

Another disadvantage, which does not affect the users (other than possibly a higher cost of the product), is that design and development of a multimodal product will demand more resources from the producers, making the product more expensive to create. However, as mentioned in Chapter 4.2, products that accommodate the needs of several user groups may gain more customers, possibly giving return on the investment. And as the number of elderly people increases, the producers may not afford *not* accommodating these groups, if they are to make a profit.

6.4 Using Multimodality to Accommodate Reading and Writing

Since the target groups are having problems with text and numbers, presenting content in any other modality than the written one, should benefit those who purely have reading problems. Some of the dyslectics in the reading problem category are having problems also with segmenting sound units, i.e. these may not benefit a hundred percent from modalities including sound, although they would understand more than when only representing information using the written modality. For the two target groups, illustrating information graphically (pictures, charts, diagrams, etc.) should therefore be the modality that will help the most, however it will not support writing on a mobile phone, nor, in many cases, enable people to check their input. The modalities will also help people without disabilities, whose cognitive style is interpreting and remembering illustrations better than text, for instance.

Offering sound as a modality is also bound to help very many in the focus groups. They will benefit from being able to hear the information presented and the information they have entered, as they can perceive the information, check if they have written what they intended to, and if not, correct the input. For those more severely affected by conditions like dyscalculia, who have huge problems reading numbers, it is especially important to hear what they have entered, as they have to make wild guesses when choosing numbers to input. Writing numbers wrong are in many situations more critical than misspelling a word, as guessing how a word is supposed to be spelled is possible, but in many cases it is not possible to do with numbers.

Users of mobile services who do not have an earpiece may restrict their use of the sound modality according to which context they are in because of the privacy issue or, of course, because not all contexts allow noise. However, users who are dependent on sound as a modality in order to perceive information are likely to make sure to get an earpiece. Apart from the issue of sound anyone can hear (which is not the case if they do not have to speak input, and have an earpiece), sound for presenting information is very useful in a mobile context. It better enables users to use mobile services with a split attention, i.e. when doing several tasks simultaneously. It also benefits users whose cognitive style is to best perceive and comprehend sound.

When using illustrations and audio on mobile phones, several issues must be considered. As far as illustrations are concerned, the issue of poor quality media which could make content more confusing (WebAIM, 2007, Design Considerations) becomes even more relevant on mobile phones. Mobile phones have different screen sizes and different screen resolutions. Software producers can accommodate this either by making pictures tailored for each phone size and resolution, or they can make software that resizes pictures according to the available screen space. The former alternative requires more time to create software for a number of phones, but the producers can be certain that the pictures are shown as they are supposed to be. The latter alternative makes the development time shorter, but the presentation of pictures will vary from phone to phone. Particularly pictures that shall be presented together with a lot of other elements are small. When the software is used on mobile phones with varying screen types, the picture will sometimes be resized to be even smaller, thereby loosing details. This can make the picture incomprehensible, especially for people with cognitive disabilities.

Audio files can be stored on phones. However, if the number and size of these are very high, few may want to use the software as it will occupy a lot of storing space on a mobile phone. It is therefore necessary to consider the time it takes to download files with a relatively large size, like audio files. As mentioned in Section 3.5.2, there are cases when it will be necessary to upload audio input to a server for conversion, i.e. upload rate needs to be considered when implementing this modality.

In the UMTS network of Telenor in Norway, the download transmission rate is now up to 384 kbit/s, and the upload transmission rate is up to 64 kbit/s (Telenor, 2007). Using audio as input together with stepwise verification may therefore be a far fetch. However, audio output and input may work if one uses the actions suggested in the success criterion “The service must not stop the form-filling process when loading information” in Section 3.5.2 regarding audio output, and continually sends input audio files as they are “spoken” to the server and verifies the input when all input have been made and sent. In the latter case, the user could close the application when he/she is finished, but the application continues to run until it has done its tasks. The user could later open the application, or be notified that the verification process has finished, and check the results.

This chapter has shown that multimodality is beneficial to people with cognitive disabilities, and will benefit not only the target groups, but also many other people. If several issues are considered and attended, multimodality will also be beneficial for people using mobile phones.

7 Designing a Mobile Phone Form Service

The introduction of this report stated that multimodality should be beneficial for cognitive processing of content also on mobile phones. This chapter describes a design of a form that attempts to illustrate the use of multimodality in a way that supports this assertion for the two target groups, and also incorporates several other design considerations asserted to be beneficial when designing forms on mobile phones. The design was specified by a paper mock-up in order to get an idea of how the design would appear as well as being able to get feedback from stakeholders early in the investigation. The paper mock-up evolved through iterations to the design presented in this chapter, as we got an idea of how the design would appear and stakeholders contributed with their knowledge and opinions.

7.1 Choosing Multimodal Content and Case for Demonstration

The two target groups of the investigation are people with **moderate text-reading problems** and people with **moderate number-reading problems** (cf. Chapter 1.2.3). The groups are assumed to *be able* to perceive respectively text and numbers correctly, although they may need to concentrate very hard on their task in order to fulfil it. With respect to the first group we wanted to demonstrate that multimodality can be used to relieve their task by replacing textual lists containing information about locations, with the locations placed on a map segment. Regarding the other group we wanted to achieve the same, by replacing list information containing numbers on a similar format, like time, date, etc. with graphic illustrations, i.e. decrease the issue of visually indistinctive choices in a list, mentioned in Chapter 2.2 in Appendix A.

A realistic example service containing lists of textual information and numeric, similar-looking information may be a kindergarten application, where kindergarten addresses are presented geographically in a map and time information about a kindergarten's activities are represented as icons. Kindergarten applications are used by all parts of the population within a certain age group; hence should Universal Design be applied. Kindergarten applications are therefore an appropriate example, and were chosen to be the case for the demonstration.

7.2 The Design of the Service Content

The content of the demonstrated kindergarten application form, as well as how it is divided on the form pages is based on two electronic forms available on the Web. They can be found at (More, 2007) and (The Municipality of Bergen, 2007). The point of the demonstration is not to show how a kindergarten application form could be realised on mobile phones, but first and foremost to demonstrate the multiple modalities in the context of electronic forms, as well as design decisions that attempt to achieve a selection of the success criteria for making forms feasible on mobile phones, Section 3.5.2. The form demonstrated does therefore lack some of the (requested) information in (More, 2007) and/or (The Municipality of Bergen, 2007). We have also added some elements which are not included in the real service interface[4]. (The Municipality of Bergen, 2007) offers additional information about each kindergarten, although not the multimodal information demonstrated in this section.

[4] According to the CEO of More AS, Mr. Ettesvoll, the company is currently working on how to design representation of kindergartens on a map in their kindergarten application forms.

The demonstration of the application is based on several assumptions:

- The authorities provide software for mobile phones containing several forms, i.e. the software has a “Choose form” screen. The kindergarten application form is one of these forms.
- The owner of the mobile phone, which is assumed to be the user of the application, has to fill in a separate form containing personal information. The other forms retrieve the information registered in this form, which corresponds to the background form in a form system, mentioned in Section 3.2.3. Then the other forms will not have to ask the user to fill in the same information over and over again.
- When the user goes out of the form, either temporarily by pressing the phone’s menu key when she suddenly needs to do something else, or by choosing the form’s “Exit” command, the information entered is sent to the form server for storing.
- The mobile software provided by the authorities is a supplement to their electronic forms on the Web. The electronic forms on the Web are available on a public form portal. The users will have to authenticate themselves when opening the mobile software and when entering the form portal, in order to get access to their profile. Information users have entered in a form on a mobile phone can be fetched from the form server and be presented in their Web equivalent, and vice versa.
- The mobile software has a “Settings” screen where the user can change the font sizes used in the form according to needs and preferences.
- The kindergarten application form can fetch all types of information about kindergartens, from map coordinates to a daily routine plan, from a server.

7.2.1 Design Considerations Applied to All Form Pages

The paper mock-up is designed to follow a selection of the guidelines and principles in Appendix A of this report, some of them were mentioned specifically in the success criteria in Section 3.5.2. These were chosen because they were considered to be important for demonstrating how forms can be made more usable on mobile phones, and for making forms more usable to people with moderate cognitive disabilities, also for those not in the target groups. The design also follows several guidelines in Appendix A which were not consciously chosen, but were applied as a result from both using common sense and being accustomed to user interfaces where these are applied. For instance, Guideline 2 in Chapter 2.1 states that each control should have a label that indicates what the users should do with the control, which may be evident for most developers of user interfaces.

The form offers selection controls over typing controls where possible, as suggested in Guideline 10 in Chapter 1.1, Guideline N-2 in Chapter 1.2 and Statement 5.5.1 in Chapter 1.3 in Appendix A. Selection over typing was stated to be a way to achieve the success criterion “The service must have a user interface that accommodates the device limitations” in Section 3.5.2.

Drop-down boxes were given as an example of a control that offers selection over typing. It was also mentioned that one should consider which would be best to use of drop-down boxes and radio buttons. The paper mock-up uses drop-down boxes in cases where the user has to choose one of several alternatives, except for cases where the number of alternatives is two. Drop-down boxes were considered to be the best alternative because a drop-down box is space-saving compared to the number of radio buttons needed to present the alternatives. Navigating between alternatives may also be easier in a drop-down box where one only needs to press the up and down keys, rather than navigating between radio buttons situated in rows and columns. When the number of alternatives is two, radio buttons is assumed to be a more usable form control, as the users can immediately see that they have to state one of two alternatives, and they only need to navigate between the two.

We believe that having a history and dictionary database associated with form controls that take text as input, as a way to achieve the same success criterion, is a very good solution for forms on mobile phones. However, database functionality is naturally not demonstrated in the paper mock-up.

A third way to contribute to achieving the success criterion is pre-filling of fields, as stated in Section 3.5.2. Where possible, the mock-up demonstrates pre-filling of fields. The information pre-filled is assumed to be fetched from the “Personal information” form.

One success criterion stated that the user interface of a service should follow guidelines for design of forms. The mock-up partly follows Advice 1-2 in Chapter 2.4 in Appendix A, of always keeping a ‘Next’ button[5] as the first choice available, as “Next” is the command users are likely to be looking for when they have no problems filling in a page and want to continue to the next step. There are cases where the ‘Next’ button will be replaced by other alternatives: ‘Choose kindergarten’ when the users choose kindergartens in a map, ‘See daily routine’ when the “See daily routine with numbers or clock icons” element is selected, and ‘Add sibling’ when the element for setting name, birth year and kindergarten for the sibling is selected, in Figures 22, 23 and 28.

Since the design has a ‘Next’ command, it naturally also has a ‘Back’ command, situated on the opposite button of the ‘Next’ command as long as it has not been replaced by other alternatives. Navigating from page to page can also be done by using the keys for moving right and left. Navigating in the latter way is assumed to be easier for the users physically: On many mobile phones, the left/right/up/down keys are placed together. In this mock-up users will be using the up and down key to navigate on a page. Somewhat experienced mobile phone users use one hand to operate their mobile phone, i.e. they use their thumb to press buttons and the four other fingers to lean the phone against. It will be easier to move the thumb from e.g. the down key or the numeric keypad, which most mobile phones have and which is usually located right below the direction keys, to the left key, than to a soft button. When a soft button contains several commands, choosing a command will require several actions. Going back and forth with the keys for moving left and right buttons is therefore easier. However, as this functionality may not be apparent, particularly for less experienced users and users with varying cognitive abilities, it seems important to have the design incorporate ‘Next’ and ‘Back’ commands on the soft buttons.

A part of Advice 1-3 in the same section and of Direction 2 in Chapter 2.3 is also followed in the mock-up: The second blue rectangle from the top in all screens but one tells the users where they are in the form-filling process and how much is left. The HCI-package of Tellu offers the possibility of creating form pages with “portfolio tabs”, one tab for each form page.

[5] Buttons known as “push buttons” on user interfaces for computers can on mobile phones be represented by actual buttons. These buttons are often situated at the top left and top right of the phone’s keypad, and are known as “soft buttons”. The phone’s (or third-party) software can draw a menu line with text containing the names of the button’s commands, like “Next”. Because phones usually only have two soft buttons, a first push on a button with a “Menu” text will show the commands available on that button. Users will have to select a command by moving up/down and choose it by pushing the soft button again or the button known as the “fire button”, which is the button usually used to select with in a phone’s interface. There are also other ways to implement this functionality.

The idea is to use these portfolio tabs, with numbers on, to represent the form page numbers and the steps of the form process. Because portfolio tabs were too difficult to draw in the paper mock-up, it does not demonstrate this idea. The third blue rectangle contains a describing name of the current step. One can often see all step names in e-forms on computers, and the steps are often not numbered. However, as the limited space on the portfolio tabs is a hindrance to presenting all step names, the solution of presenting all but the current with numbers was chosen. Telling the users where they are and how much is left was applied because of the importance of giving the users a sense of how much "work" is needed to fill in a form, which may be even more important on mobile phones.

The paper mock-up follows the usability expert Jakob Nielsen's advice of not having reset buttons for resetting the form fields, rendered in Chapter 2.2 of Appendix A, as he claims that these are unnecessary and can be clicked by mistake, which can also be applicable to mobile phones.

Several guidelines were applied to make the form usable to the people in the target groups. The design of the form attempts to achieve the success criterion "A service must accommodate variations in skills and abilities" by incorporating several principles/guidelines. Guideline 14 in Chapter 1.1 in Appendix A, Guideline "Provide choice in methods of use" of Universal Design principle 2 "Flexibility in Use" and Guideline "Use different modes (pictorial, verbal, tactile) for redundant presentation of essential information" of Universal Design principle 4 "Perceptible Information", rendered in Chapter 4.3, are incorporated by offering enlargement of text and letting users choose presentation modality in two places, described in the next section. Enlargement of text is a recommendation when designing for people with cognitive disabilities, stated in Recommendation 1.1.1 in Chapter 3.2 in Appendix A. The possibility of enlarging text is also beneficial to people with moderate visual disabilities and to people who for some reason can not focus on reading, as reading large letters is easier than reading small. Multimodality is, as mentioned in Chapter 6.2, acknowledged to be beneficial for people with cognitive disabilities, which is also stated in Recommendation 1.2.1 in Chapter 3.2 in Appendix A. Multimodality is also beneficial to those who better interpret a particular mode of presentation better than other modes, as stated in Chapter 6.3.

We can also say that Guideline "Accommodate a wide range of literacy and language skills" of Universal Design principle 3 "Simple and Intuitive" is followed when offering alternative presentation modes, although our range of literacy and language skills is rather narrow, considering the chosen target groups.

The form does not present text in italics and all capital letters, as Recommendation 2.1.3 in Chapter 3.2 in Appendix A states that this should be avoided in order to improve readability for the cognitively disabled. The form applies Recommendation 2.2.4 in the same section: text is emphasised by using bold font and/or larger text size. Information text like "The search gave 5 results" is not considered important to emphasise, and is shown with a relatively small text size. Information that is more important for the user's form-filling process, like labels to form controls, has a larger text size. Headings are emphasised by using bold font.

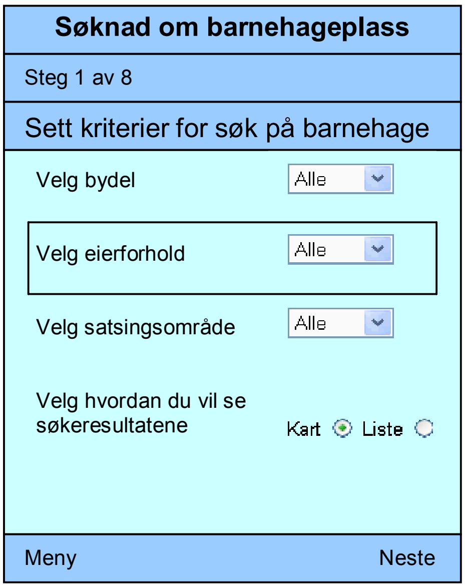
The form controls and their labels were placed in a grid layout to improve the readability and comprehensibility of the form. Section 5.4.1 mentions several fonts that increase readability for people with dyslexia. Because these fonts are likely to improve readability also for other people, the font Arial is used in the paper mock-up. All of these efforts should increase readability and comprehensibility for all people, not only for the target groups and people with other cognitive disabilities.

Finally, a very important aspect, the guidelines under Universal Design principle 1 “Equitable Use” (with the exception of the guideline regarding privacy, security and safety) are incorporated in the design. The suggested design does not point out or address specific user groups but treats everyone as equals, thereby should no one feel that they loose their dignity when interacting with the form.

7.2.2 The Design of Each Form Page

As mentioned earlier, navigating from control to control is done by using the up and down buttons, navigating from form page to form page can be done by using the left and right keys, as well as the soft buttons ‘Next’ and ‘Back’. Associated radio buttons and check boxes are here considered to be one control. In order to be able to operate controls which require navigation within them, e.g. a drop-down box require up and down navigation when selecting a different element in it, the fire button must be pressed to activate the control, and also to deactivate it.

In the first step of the form filling process, shown in Figure 21, the users have to set the criteria district (which part of town), type of ownership (public or private) and what the kindergartens’ focus area should be (outdoor life, religion, pedagogy, etc.). These criteria will be used when searching for kindergartens. The users can also choose which modality the search results should be presented in: *Graphical* in form of a map, or *textual* in form of a list. The form control currently selected is marked by the black rectangular line. The lowest part of Figure 21 is the menu line. Commands on the menu line should be ‘Next’, situated on the right side, which takes the user to the next form page, ‘Help’, which presents a new screen with a help text related to the current element selected in the form page, in addition to instructions regarding the form page, ‘Enlarge’ which enlarges the text of the current element selected in the form page and ‘Exit’, which takes the user back to the “Choose form” screen.



Søknad om barnehageplass

Steg 1 av 8

Sett kriterier for søk på barnehage

Velg bydel Alle

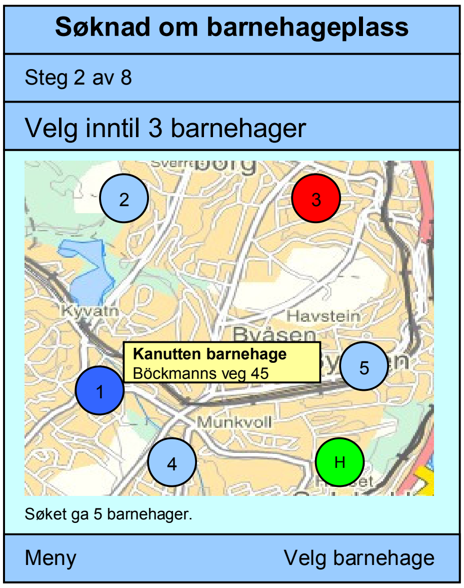
Velg eierforhold Alle

Velg satsingsområde Alle

Velg hvordan du vil se søkeresultatene Kart ● Liste ○

Meny Neste

Figure 21. Setting criteria for finding kindergartens.



Søknad om barnehageplass

Steg 2 av 8

Velg inntil 3 barnehager

Søket ga 5 barnehager.

Meny Velg barnehage

Figure 22. First round of choosing kindergartens.

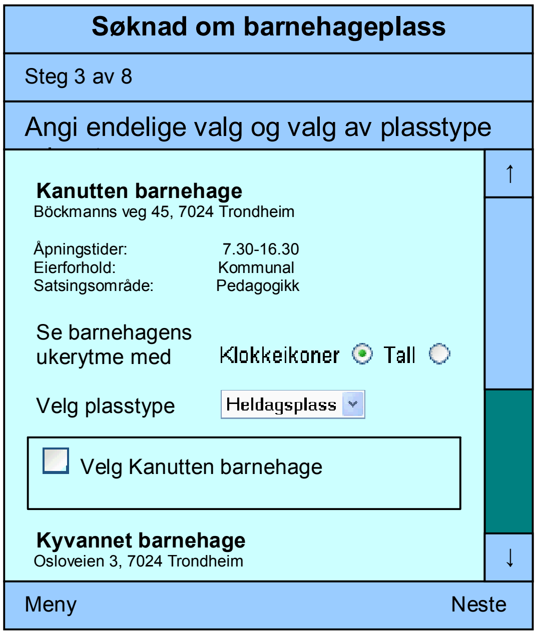
Figure 22 demonstrates how a map with kindergartens can be an alternative modality to a list with textual alternatives. The map segment shows the district the user chose in the previous step. The green circle with an 'H' represents the user's house in this paper mock-up, but the idea is to use a house icon in an actual form, which will better communicate what it represents. The circles with numbers represent kindergartens. Which kindergartens are shown depends on the other criteria the user set in the previous step. Each kindergarten's number is the number the user must press to select it. Kindergarten 1 is currently selected in the figure, illustrated by a darker colour than the others, and by a small "popup" rectangle connected to it, showing the name and address of the kindergarten. The circle of kindergarten 3 is red in order to illustrate that the user has chosen it, i.e. she wants to bring it to the next step (in order to see more information and decide whether or not to apply to it). Menu-line commands on the right soft button should be 'Choose kindergarten' when the user has selected but not chosen a kindergarten, and 'Cancel choice' when the user has both selected and chosen a kindergarten. On the left soft button, 'Menu', commands should be the same as in the previous page, although now 'Back' and 'Next' will also have to be placed here. Users should be able to choose and regret choice of kindergartens also by pressing the fire button.

The solution of replacing the 'Next' command with commands for choosing or cancelling choice of kindergartens, was chosen in order to make the users easily spot these new commands. An alternative could have been to draw a traditional, selectable push button on the screen, below the map, but keeping commands for doing other actions than form-filling (in terms of choosing or entering values) out of the form page seemed to be a better solution. However, the appearing/disappearing buttons may be confusing to the users. The design is now based on that users understand or learn that, for anything other than filling in form fields, they should first cast a glimpse at the right soft button, and then check the menu on the left soft button if they are looking for a particular command.

Functionality like navigating in the map with a phone's left/right and up/down keys, and changing the level of detail in the map presented may also have been beneficial in this case. For instance, for some parents it might have been nice to be able to see where a kindergarten is situated in relation to public transportation. As this report focuses on demonstrating multimodality, *not mechanisms of electronic maps*, the map element of the user interface is relatively simple in our case.

People with moderate reading problems will now see each kindergarten's position, and if they want, they can use the knowledge they have about the geography of their district to quicker interpret the address text. People who have trouble reading numbers can see where in a street a kindergarten is situated. Those, whose cognitive style is to perceive and interpret illustrations better than text, will benefit. This way of presenting kindergartens should make it easier for all users to decide which ones are most relevant for them, as they have the possibility to see where the kindergartens are situated in proportion to their own home, work place, or other important locations entered in the personal information form, provided they choose the district they are situated in. The multimodal representation of kindergartens in a map is relieving the task not only for the target group it was made for, but also for others.

The next step for the user is to make the final choices of which kindergartens to apply to. For each kindergarten chosen in the map, the form page presents the kindergarten's address, opening hours, type of ownership, focus area and a possibility of seeing its weekly routine, see Figure 23. Users who want to see a kindergarten's weekly routine can choose presentation modality by deciding whether to see it with clock icons or numbers. When one of the radio buttons is chosen, the right soft button shows the command ‘See weekly routine’. Commands that suddenly appear/disappear will be an issue also here. The applicant can also see which place types (e.g. full day, half day, 28 hours/week, 16 hours/week, etc.) the kindergarten offers. If the applicant decides to apply to the kindergarten, a place type must be selected and a “Choose” check box must be checked. If the applicant finds any of the kindergartens uninteresting, he/she can go back to the map and unselect them by using the ‘Cancel choice’ command for each chosen kindergarten, and choose new ones until the maximum number of possible to apply to is reached.



Søknad om barnehageplass

Steg 3 av 8

Angi endelige valg og valg av plasstype

Kanutten barnehage
Böckmanns veg 45, 7024 Trondheim

Åpningstider:	7.30-16.30
Eierforhold:	Kommunal
Satsingsområde:	Pedagogikk

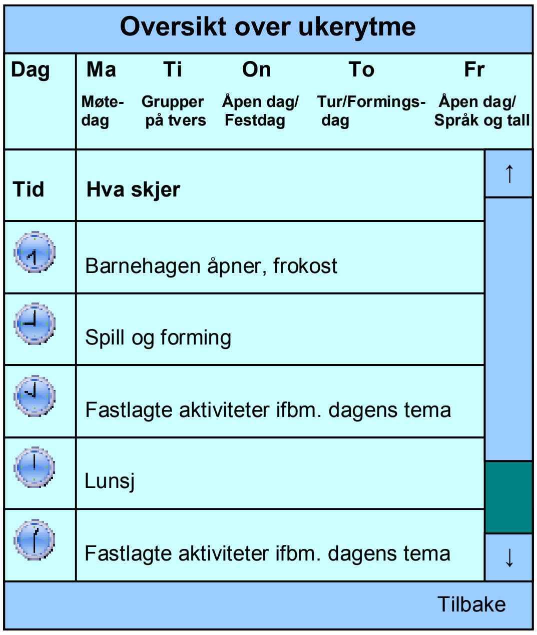
Se barnehagens ukerytme med Klokkeikoner ● Tall ○

Velg plasstype Heldagsplass

□ Velg Kanutten barnehage

Kyvannet barnehage
Osloveien 3, 7024 Trondheim

Figure 23. Deciding whether or not to choose specific kindergartens.



Oversikt over ukerytme

Dag	Ma Møte-dag	Ti Grupper på tvers	On Åpen dag/ Festdag	To Tur/Formings-dag	Fr Åpen dag/ Språk og tall

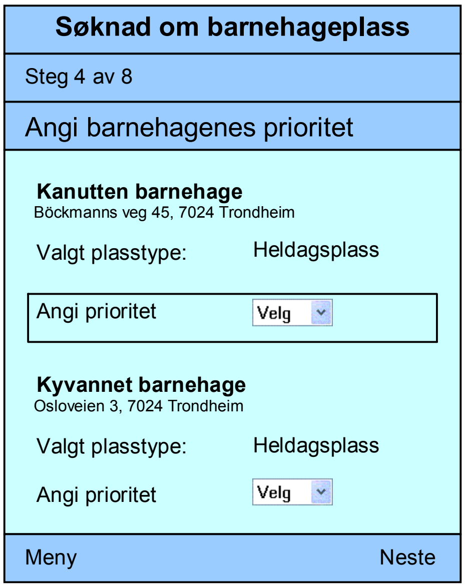
Tid	Hva skjer
◷	Barnehagen åpner, frokost
◷	Spill og forming
◷	Fastlagte aktiviteter ifbm. dagens tema
◷	Lunsj
◷	Fastlagte aktiviteter ifbm. dagens tema

Figure 24. A kindergarten's weekly routine.

Figure 24 has a somewhat different design than the other form pages: it lacks an indicator of how far the user has come in the form-filling process, it does not have an instructing text, and the menu line only has one command: ‘Back’. This is meant to signalise that this page is not a part of the form, it is purely informing the user of the theme for each day in a week in the kindergarten, as well as the daily routine. The figure shows how the page appears when presentation with clock icons has been selected.

Presenting clock icons in a list rather than points of time in a list that are of a format like hh:mm, should eliminate the problems people with moderate number-reading problems would have with this page. It will also benefit all whose cognitive style is to perceive and interpret illustrations better than numbers. The presentation does not follow the “Contiguity Principle” rendered in Chapter 6.2; to present corresponding words and pictures contiguously rather than separately. In the map presentation, the kindergartens had a popup with their name and address. We consider the redundant, numeric information to not be necessary in this page, as users who want to see points of time as numbers would choose numeric presentation. In the map however, redundant, textual information is necessary for knowing the exact address of the kindergarten.

The next step in the form, shown in Figure 25, is to set the priority to first or second (or third, if three had been chosen) for each chosen kindergarten. Drop-down boxes on previous pages had a pre-selected value, while these have the text “Choose” to initiate the user to make a choice, as they contain the same values. Presenting both with the first value, which is “1”, seemed strange, since the user can not give the kindergartens the same priority. Unlike other informative text, the place type chosen in the previous step is emphasised with a medium font size, to make the user notice it and react if he/she chose the wrong type.



Søknad om barnehageplass

Steg 4 av 8

Angi barnehagenes prioritet

Kanutten barnehage
Böckmanns veg 45, 7024 Trondheim

Valgt plasstype: Heldagsplass

Angi prioritet: Velg

Kyvannet barnehage
Osloveien 3, 7024 Trondheim

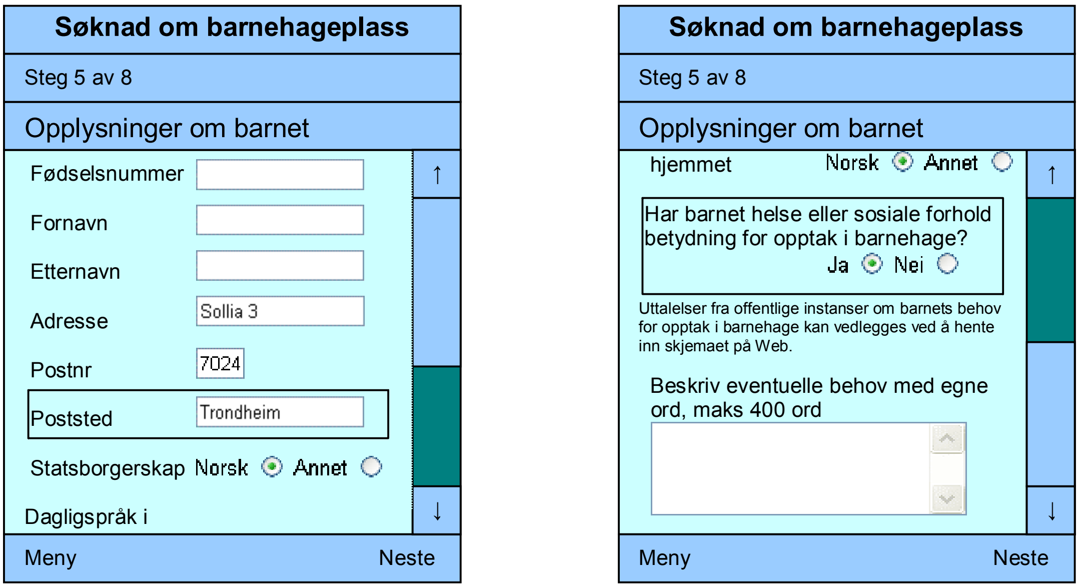
Valgt plasstype: Heldagsplass

Angi prioritet: Velg

Figure 25. Setting priorities.

Figure 26 shows the form page for stating the child’s personal information. This is where the pre-filling technique is first demonstrated. The form here makes the assumption that the form-filler is the child’s guardian and that the child lives with the form-filler, and fetches relevant information that the form-filler has entered in the personal information form. If the assumptions turn out to be incorrect, the form-filler can erase the content of the fields and enter the correct information. Otherwise, he/she can simply move down until a form control that needs to be edited is selected. Text fields requiring input where the length of the input is known in advance, like year of birth, post code and phone number, have a different length than the other fields. According to (Fichter, 2003), rendered in Chapter 2.2 of Appendix A, the length of text fields should imply what kind of answer is expected. This advice is only applied to fields with the type of input mentioned above, as applying it to every text field may make the grid layout messy and confusing. It is also applied to make these fields stand out from the others, as they require a different (i.e. fixed length and numeric) type of input. In Figure 26, the post code field is shorter than the others. As the kindergarten form is a supplement to its Web equivalent, the applicant is instructed to enclose documents relevant for the application on the Web.

Another technique commonly used in e-forms, is to fade fields that become irrelevant according to earlier input made, i.e. a type of “routing”, described in Section 3.2.4. This technique could be applied according to the setting of the last radio buttons in Figure 26, “Does the child’s health or social conditions impact the child’s need for kindergarten?”. If “No” is selected, the rest of the form page should be locked for manipulation by fading it.



Søknad om barnehageplass

Steg 5 av 8

Opplysninger om barnet

Fødselsnummer

Fornavn

Etternavn

Adresse: Sollia 3

Postnr: 7024

Poststed: Trondheim

Statsborgerskap Norsk ● Annet ○

Dagligspråk i

Meny — Neste

Søknad om barnehageplass

Steg 5 av 8

Opplysninger om barnet

hjemmet Norsk ● Annet ○

Har barnet helse eller sosiale forhold betydning for opptak i barnehage? Ja ● Nei ○

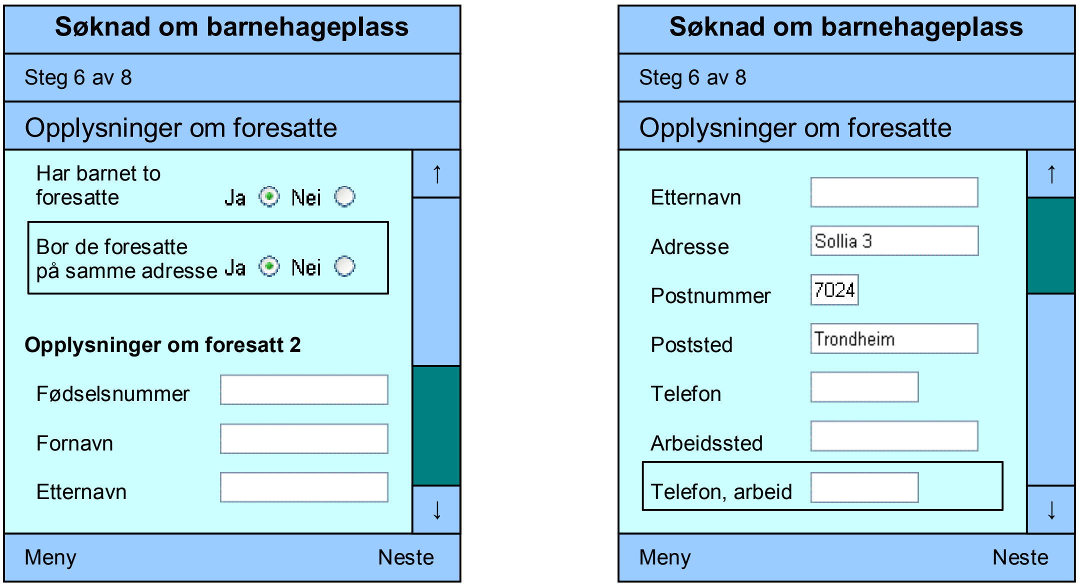
Uttalelser fra offentlige instanser om barnets behov for opptak i barnehage kan vedlegges ved å hente inn skjemaet på Web.

Beskriv eventuelle behov med egne ord, maks 400 ord

Meny — Neste

Figure 26. Stating information about the child.

The next step can also use pre-filling and fading of irrelevant fields, see Figure 27. The user is first asked to state whether the child has two guardians or not. If the answer is “No”, the rest of the form can be faded. If “Yes”, the user has to state whether or not the guardians live at the same address. The remaining part of the form page asks the user to enter the other guardian’s personal information. If the guardians live at the same address, the address fields can be pre-filled in the same way as for the child. Phone number is also a candidate for pre-filling, however, as more people now tend to want to be contacted by their personal mobile phone, it may be better to leave it empty. In the figure, post code and phone number text fields have a shorter length than the other fields.



Søknad om barnehageplass

Steg 6 av 8

Opplysninger om foresatte

Har barnet to foresatte Ja ● Nei ○

Bor de foresatte på samme adresse Ja ● Nei ○

Opplysninger om foresatt 2

Fødselsnummer

Fornavn

Etternavn

Meny — Neste

Søknad om barnehageplass

Steg 6 av 8

Opplysninger om foresatte

Etternavn

Adresse: Sollia 3

Postnummer: 7024

Poststed: Trondheim

Telefon

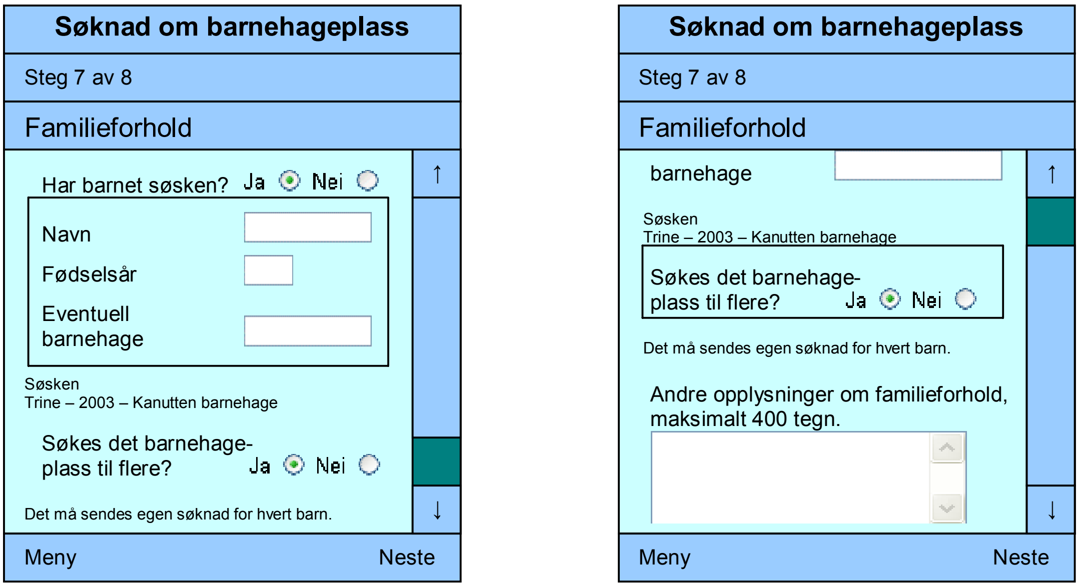
Arbeidssted

Telefon, arbeid

Meny — Neste

Figure 27. Guardian information.

In step 7, the user enters information about the family that is relevant for the application, see Figure 28. If the child does not have any siblings, the three text fields after the first radio button should be faded. The three text fields are connected. When one of these is selected, the right soft button should display the command ‘Add sibling’, which also should be faded as long as not all of the three fields are filled in, to let the user know that this is not an option yet. If the user adds a sibling, the information entered is presented beneath the input fields. Also here, commands that suddenly appear/disappear will be an issue.



Søknad om barnehageplass

Steg 7 av 8

Familieforhold

Har barnet søsken? Ja ● Nei ○

Navn

Fødselsår

Eventuell barnehage

Søsken
Trine – 2003 – Kanutten barnehage

Søkes det barnehageplass til flere? Ja ● Nei ○

Det må sendes egen søknad for hvert barn.

Meny Neste

Søknad om barnehageplass

Steg 7 av 8

Familieforhold

barnehage

Søsken
Trine – 2003 – Kanutten barnehage

Søkes det barnehageplass til flere? Ja ● Nei ○

Det må sendes egen søknad for hvert barn.

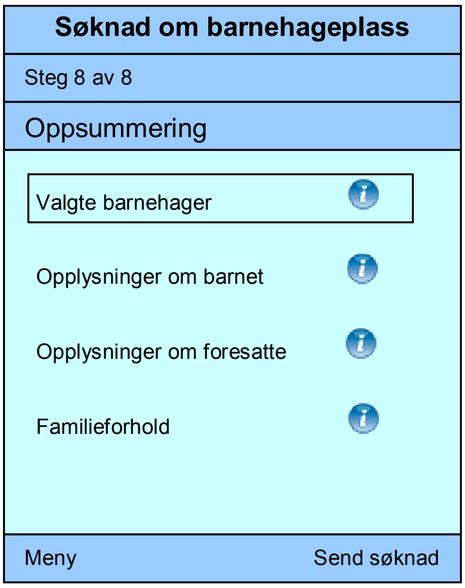
Andre opplysninger om familieforhold, maksimalt 400 tegn.

Meny Neste

Figure 28. Information about family situation.

In the last step, the form summarisation technique presented in Chapter 2.5 in Appendix A is applied, see Figures 29 and 30. The technique correlates with Guideline 13, "Design for top-down interaction" in Chapter 1.1 in Appendix A, i.e. it can be used to achieve the success criterion "There must be a limited intensity of information", stated in Section 3.5.2. Unlike the demonstration in the appendix, which used the technique on labels and their respective text fields, Figures 29 and 30 demonstrate how it can be used to hide relatively large blocks of information and show the blocks one by one, thereby reducing the information intensity. Hiding each form control may have been sensible on the mobile phones of 2002 and back, but on the screens of many of today's phones, this is likely to only be tedious. That is why nothing but pure information elements use this technique in the proposed design.

Figure 29 shows a summary of the form, consisting of a list containing the names of the main areas of the form and information icons. The information icons are meant to signal that the list element has more to tell the user if it is manipulated somehow (chosen by pressing the fire button). Figure 30 demonstrates that the form presents the information the user entered when choosing kindergartens. The page is not editable, so if the user has to change anything, he/she will have to go back to that page. The right soft button has a 'Send application' command for sending the information to the server.



Søknad om barnehageplass

Steg 8 av 8

Oppsummering

Valgte barnehager

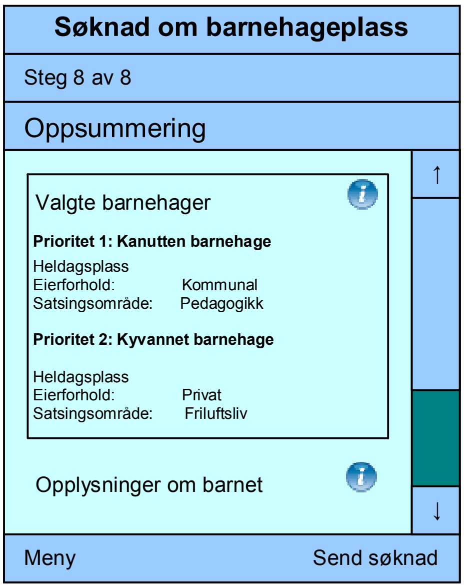
Opplysninger om barnet

Opplysninger om foresatte

Familieforhold

Meny | Send søknad

Figure 29. Form summarisation.



Søknad om barnehageplass

Steg 8 av 8

Oppsummering

Valgte barnehager

Prioritet 1: Kanutten barnehage

Heldagsplass

Eierforhold:	Kommunal
Satsingsområde:	Pedagogikk

Prioritet 2: Kyvannet barnehage

Heldagsplass

Eierforhold:	Privat
Satsingsområde:	Friluftsliv

Opplysninger om barnet

↑ ↓

Meny | Send søknad

Figure 30. Form summarisation, expanded.

Simple validation, like checking that all fields requiring input have input, should be executed. As all of the fields that are not faded require input, the paper mock-up, like the form at (The Municipality of Bergen, 2007), does not show which of the fields require input, e.g. by placing an asterisk next to them. The paper mock-up does also not demonstrate how to present the errors from validation of form fields, as form validation is not what we are focusing on in this report. One way of doing this can be the same as for many electronic forms on computer user interfaces: by marking fields that need to be corrected, in a way that makes them stand out.

This chapter has demonstrated two presentation modes for making forms more usable to people in the target groups. In addition, several design elements that are asserted to be beneficial in forms on mobile phones, generally and for people with cognitive disabilities, were applied to the demonstration. Voice input and output would also be beneficial modes for people in the target groups, as well as people with other cognitive disabilities, people in other disability categories and healthy people in a mobile context (people who are driving, etc.), as mentioned in Chapter 6.4 and Section 3.5.2. However, this is not demonstrated as we chose to focus on a graphical mode in this report, since this was considered to be the mode that would be beneficial to most people, stated in Chapter 6.4.

8 Realising the Design in a J2ME Application

We wanted to get a better view of how the form design presented in the previous chapter can appear on a mobile phone. We also wanted to give those who would evaluate elements of the design a better physical representation, in order for them to feel how the interaction with the form on an actual phone will appear. By prototyping the specified design in J2ME, described in Chapter 2.4, representation of the form on mobile phones is enabled. During the implementation, some necessary design improvements were discovered. Chapter 8.1 presents screen shots from the form's final user interface, Chapter 8.2 describes how the design was implemented.

8.1 Screen Shots from the Prototype

The following figures present screen shots of the prototype from a phone emulator. Different emulated phones give different presentations of the screens, as the display sizes vary and the hardware producers implement the J2ME application program interface differently. Seeing the service on the physical phones may also give another impression than on the corresponding emulated ones. The emulated phone used when taking the screen shots is the SonyEricsson JP7 with a screen size of 240x320 pixels.

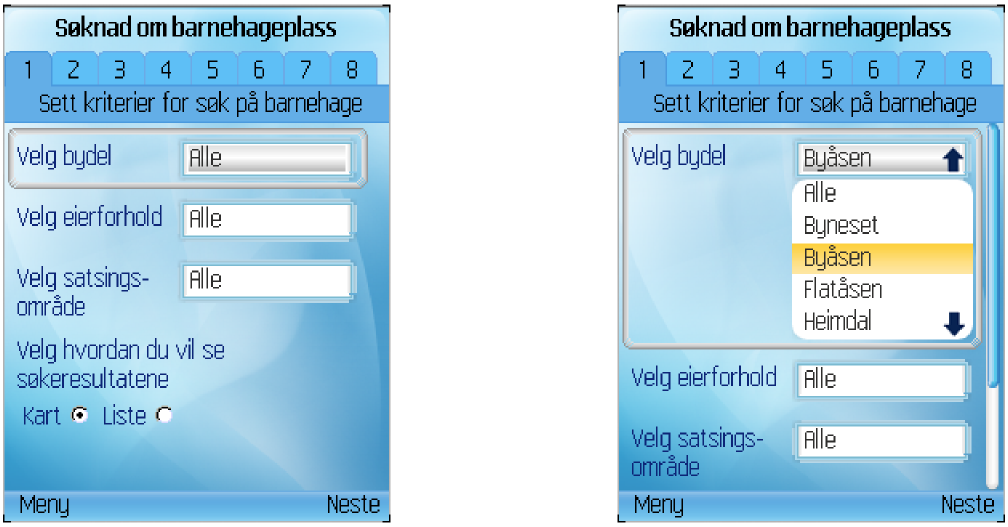
Several design aspects stated to be desirable in the previous chapter, are not demonstrated in the prototype. Text input fields connected to a history and dictionary database were neither demonstrated in the paper mock-up, nor in the prototype, as the research assignment as well as the hypothesis led us to focus on other elements of the design. The possibility of enlarging text was not implemented, as this was demonstrated in (Flaten, 2006). Pre-filling of fields are illustrated, but was done by setting the values directly into the relevant fields in the source code. Fading of irrelevant text was not considered necessary to demonstrate. The prototype uses the emulated phone's built-in font. In order to use a font more like Arial or any other of the recommended screen fonts, the prototype would have had to retrieve such fonts from a server.

Figure 31 shows the first form page of the prototype. The portfolio tabs are labelled with numbers. The darker blue colour on the first tab shows that the user currently is at step 1 of 8: "Set the kindergarten search criteria". The tabs should clearly represent the steps of the form process as well as the page numbers.

The figure also shows a special feature of Tellu's HCI-package: When elements are expanded, in this case a drop-down box, the expanded element is not drawn over the elements below. The layout of the form page changes, as the elements below are moved downwards. This feature may be very beneficial on mobile phones, as the users have the possibility of still seeing the elements that they need to fill in after the current selected, thereby not loosing the form context. On the screen of e.g. a desktop computer, this would not be such an issue, as the users would still be able to see many of the elements below the expanded drop-down box. However, the change of layout may be confusing or disturbing, i.e. the opposite of beneficial.

The three first grouped elements demonstrate the grid layout. Many of the radio buttons' labels, and in some cases also the alternative label associated to a single radio button, are quite long and can require wrapping the text over several lines. Presenting radio buttons in a grid layout therefore turned out to be confusing. The design was therefore changed to let radio buttons with labels occupy the entire width of the screen, also when there is enough room for them on one line.

The demonstrated design in the paper mock-up also placed, when a label needed to be wrapped, the associated form control next to the bottom line of the label. The prototype places the form control next to the top line, as that seemed more reasonable. In cases where the width of the screen is relatively small or the label is very long, this may not be a good solution, as the user will have to read the label maybe many lines downwards, and then search upwards for the form control. However, a mobile phone screen may be too small for this to be an issue, as long as the label does not present itself over more than a screen.


Søknad om barnehageplass

Sett kriterier for søk på barnehage

Velg bydel: Alle

Velg eierforhold: Alle

Velg satsings-område: Alle

Velg hvordan du vil se søkeresultatene

Kart ● Liste ○

Søknad om barnehageplass

Sett kriterier for søk på barnehage

Velg bydel: Byåsen

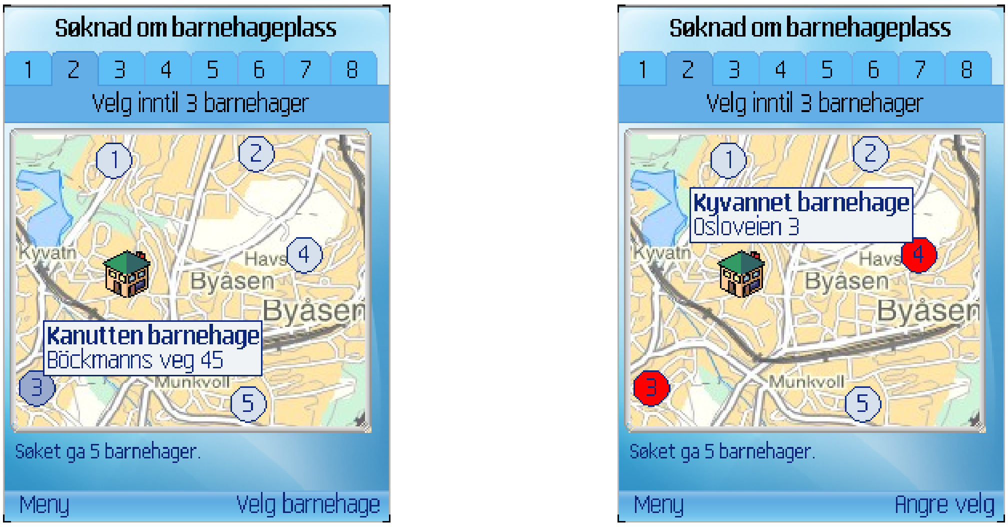
- Alle
- Byneset
- Byåsen
- Flatåsen
- Heimdal

Velg eierforhold: Alle

Velg satsings-område: Alle

Figure 31. Page 1 of the kindergarten application form.

The next figure presents the second form page when the user has chosen to see the search results in a map. The user's home is now represented by a house icon. The figure shows that a kindergarten which only is selected is drawn with a darker blue colour and a popup, while a kindergarten that the user has chosen to take to the next step is drawn with a red colour. It also demonstrates that the right soft button displays the 'Choose' command when a kindergarten is selected, and 'Cancel' when a kindergarten is selected and have been chosen.


Søknad om barnehageplass

Velg inntil 3 barnehager

Kanutten barnehage
Böckmanns veg 45

Søket ga 5 barnehager.

Søknad om barnehageplass

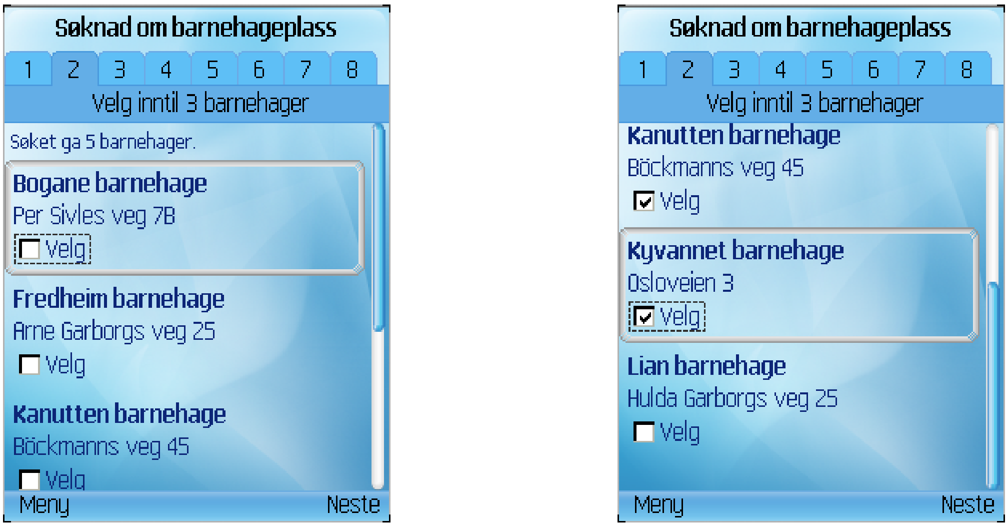
Velg inntil 3 barnehager

Kyvannet barnehage
Osloveien 3

Søket ga 5 barnehager.

Figure 32. Page 2 of the kindergarten application form, map representation.

Figure 33 shows the second page of the form when the user has chosen to see the search results in a list. This is a list where making multiple choices is allowed. It deviates from the traditional list boxes as the alternatives are not in a box, and each alternative in the list can be chosen by checking a check box, not by pressing some "mysterious" combination of keys. Two of the kindergartens have been chosen, illustrated by the checked check boxes.

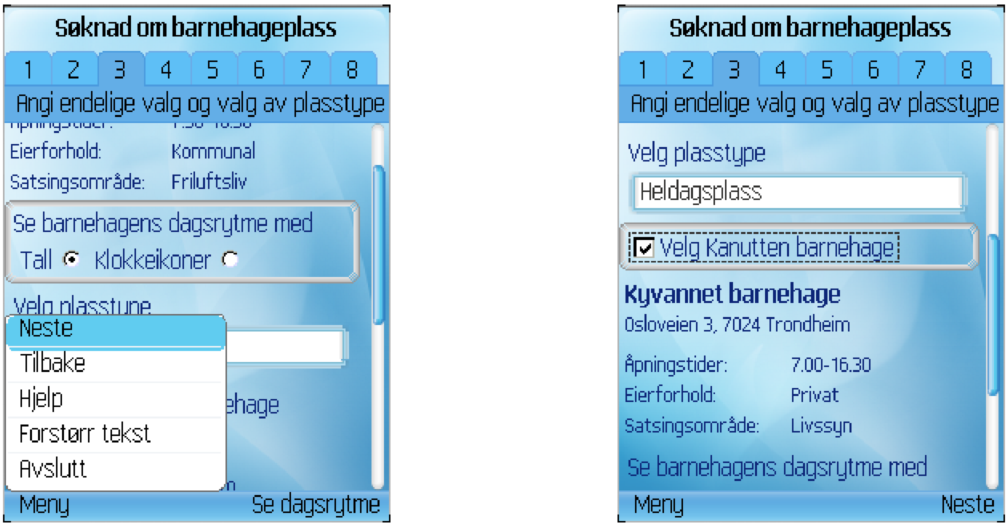


Søknad om barnehageplass
1 2 3 4 5 6 7 8
Velg inntil 3 barnehager
Søket ga 5 barnehager.
Bogane barnehage
Per Sivles veg 7B
□ Velg
Fredheim barnehage
Arne Garborgs veg 25
□ Velg
Kanutten barnehage
Böckmanns veg 45
□ Velg
Meny Neste

Søknad om barnehageplass
1 2 3 4 5 6 7 8
Velg inntil 3 barnehager
Kanutten barnehage
Böckmanns veg 45
☑ Velg
Kyvannet barnehage
Osloveien 3
☑ Velg
Lian barnehage
Hulda Garborgs veg 25
□ Velg
Meny Neste

Figure 33. Page 2 of the kindergarten application form, list representation.

The following figure shows the third page of the form, where the user decides whether or not to apply to the two kindergartens chosen in the previous step. The figure also demonstrates that the 'See daily routine' command is displayed when the group element "See daily routine of the kindergarten with numbers or clock icons" is selected, and that the 'Next' command is moved to the left soft button when it is replaced by another command at the right soft button.



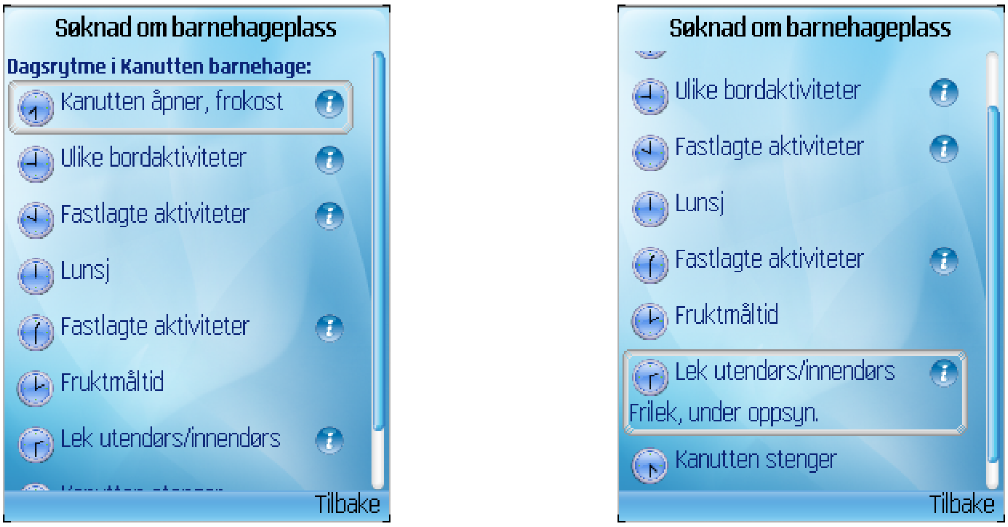
Søknad om barnehageplass
1 2 3 4 5 6 7 8
Angi endelige valg og valg av plasstype
Eierforhold: Kommunal
Satsingsområde: Friluftsliv
Se barnehagens dagsrytme med
Tall ● Klokkeikoner ○
Velg plasstype
Neste
Tilbake
Hjelp
Forstørr tekst
Avslutt
Meny Se dagsrytme

Søknad om barnehageplass
1 2 3 4 5 6 7 8
Angi endelige valg og valg av plasstype
Velg plasstype
Heldagsplass
☑ Velg Kanutten barnehage
Kyvannet barnehage
Osloveien 3, 7024 Trondheim
Åpningstider: 7.00-16.30
Eierforhold: Privat
Satsingsområde: Livssyn
Se barnehagens dagsrytme med
Meny Neste

Figure 34. Page 3 of the kindergarten application form.

Figure 35 shows a kindergarten's daily routine, where each activity's point of time is represented by clock icons. The paper mock-up's corresponding screen presented the themes of each day as well as the daily routine. However, that design was rejected because it turned out to require horizontal scrolling, also on quite large screens. Horizontal scrolling may be confusing to relate to in addition to the vertical scrolling on devices with such limited screen sizes. Guideline 5.3.3 of the Mobile Web Best Practices (Chapter 1.3, Appendix A) also advises against using scrolling in more than one direction. It was therefore decided to only have the daily routine in the design, which was all that was needed anyway, in order to demonstrate how clock icons can relieve the reading of points of time in a list.

It was also decided to let each activity that does not have a self-descriptive name, have "hidden" detailed information about itself, like the elements in the summary list described in the paper mock-up. An activity signals that it has more details by an information icon attached to it. Then it is up to the user to decide whether or not to retrieve the details, which squares with the guideline "Design for top-down interaction" (Guideline 13, Chapter 1.1, Appendix A). The clock icons displayed in the figure are "home made", which makes, apparently, the issue of poor quality media that can be more confusing, mentioned in Chapter 6.3, valid here. However, professionally made clock icons are likely to clearly communicate the points of time.



Søknad om barnehageplass

Dagsrytme i Kanutten barnehage:

Kanutten åpner, frokost

Ulike bordaktiviteter

Fastlagte aktiviteter

Lunsj

Fastlagte aktiviteter

Fruktmåltid

Lek utendørs/innendørs

Tilbake

Søknad om barnehageplass

Ulike bordaktiviteter

Fastlagte aktiviteter

Lunsj

Fastlagte aktiviteter

Fruktmåltid

Lek utendørs/innendørs

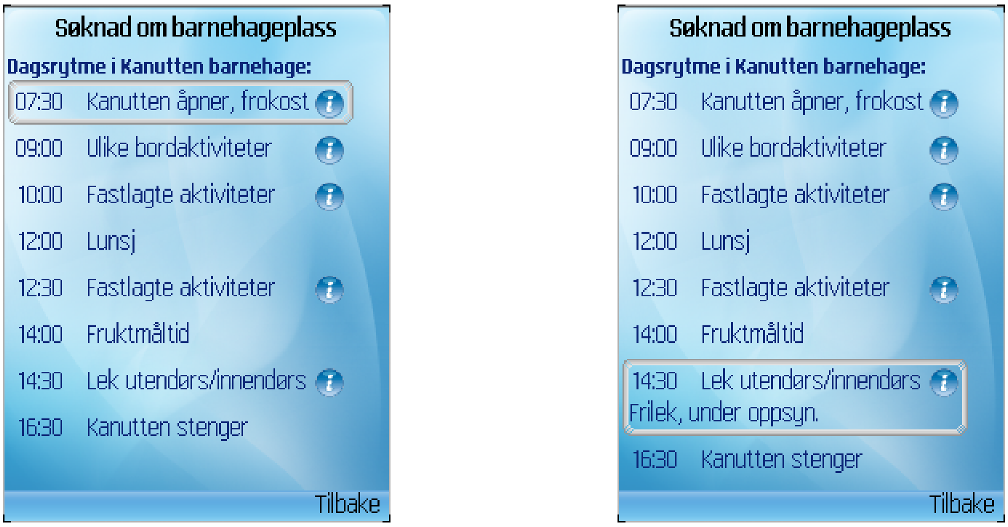
Frilek, under oppsyn.

Kanutten stenger

Tilbake

Figure 35. Daily routine with clock icons.

The next figure shows exactly the same as Figure 35, except for that the points of time now are represented by numbers.



Søknad om barnehageplass

Dagsrytme i Kanutten barnehage:

07:30 Kanutten åpner, frokost
09:00 Ulike bordaktiviteter
10:00 Fastlagte aktiviteter
12:00 Lunsj
12:30 Fastlagte aktiviteter
14:00 Fruktmåltid
14:30 Lek utendørs/innendørs
16:30 Kanutten stenger

Tilbake

Søknad om barnehageplass

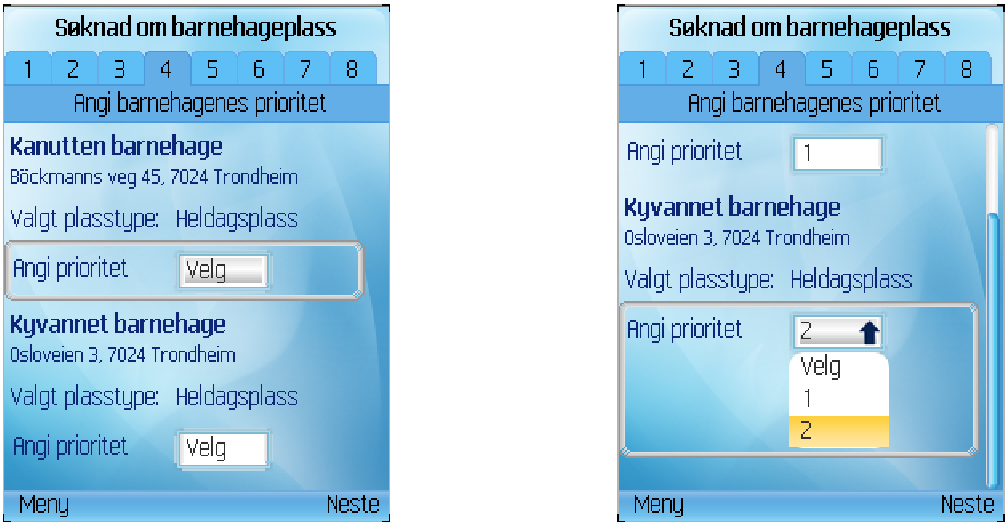
Dagsrytme i Kanutten barnehage:

07:30 Kanutten åpner, frokost
09:00 Ulike bordaktiviteter
10:00 Fastlagte aktiviteter
12:00 Lunsj
12:30 Fastlagte aktiviteter
14:00 Fruktmåltid
14:30 Lek utendørs/innendørs
Frilek, under oppsyn.
16:30 Kanutten stenger

Tilbake

Figure 36. Daily routine with numbers.

Figure 37 presents the fourth form page, where the user sets the application priority for the two chosen kindergartens.



Søknad om barnehageplass

1 2 3 4 5 6 7 8

Angi barnehagenes prioritet

Kanutten barnehage
Böckmanns veg 45, 7024 Trondheim

Valgt plasstype: Heldagsplass

Angi prioritet Velg

Kyvannet barnehage
Osloveien 3, 7024 Trondheim

Valgt plasstype: Heldagsplass

Angi prioritet Velg

Meny Neste

Søknad om barnehageplass

1 2 3 4 5 6 7 8

Angi barnehagenes prioritet

Angi prioritet 1

Kyvannet barnehage
Osloveien 3, 7024 Trondheim

Valgt plasstype: Heldagsplass

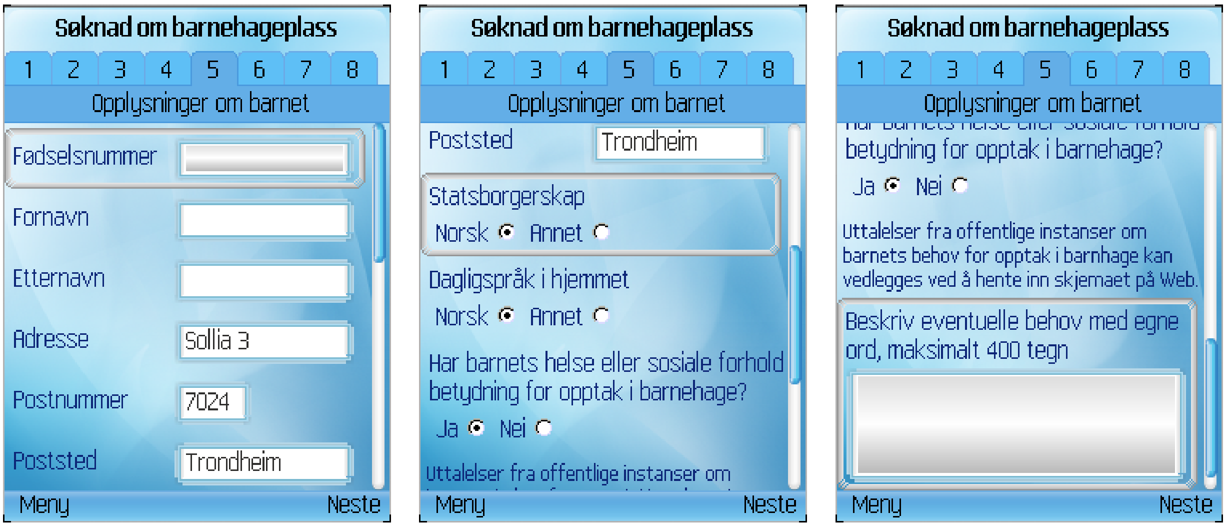
Angi prioritet 2

Velg
1
2

Meny Neste

Figure 37. Page 4 of the kindergarten application form.

Figure 38 presents the fifth step of the application process, where the user enters the child's personal information.



Søknad om barnehageplass
1 2 3 4 5 6 7 8
Opplysninger om barnet

Fødselsnummer
Fornavn
Etternavn
Adresse Sollia 3
Postnummer 7024
Poststed Trondheim

Meny Neste

Søknad om barnehageplass
1 2 3 4 5 6 7 8
Opplysninger om barnet

Poststed Trondheim
Statsborgerskap
Norsk ● Annet ○
Dagligspråk i hjemmet
Norsk ● Annet ○
Har barnets helse eller sosiale forhold betydning for opptak i barnehage?
Ja ● Nei ○
Uttalelser fra offentlige instanser om

Meny Neste

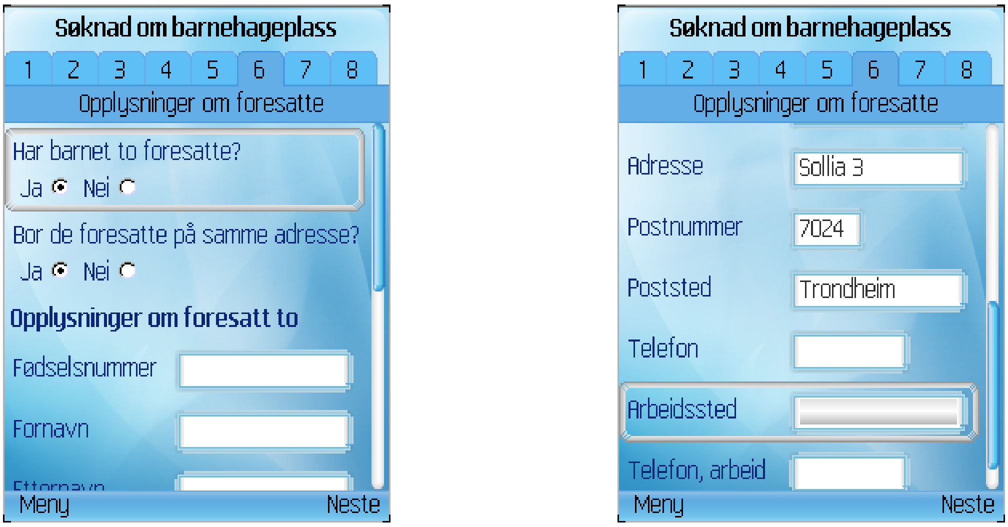
Søknad om barnehageplass
1 2 3 4 5 6 7 8
Opplysninger om barnet

betydning for opptak i barnehage?
Ja ● Nei ○
Uttalelser fra offentlige instanser om barnets behov for opptak i barnhage kan vedlegges ved å hente inn skjemaet på Web.
Beskriv eventuelle behov med egne ord, maksimalt 400 tegn

Meny Neste

Figure 38. Page 5 of the kindergarten application form.

The following figure shows the sixth form page, which requests the user to enter guardian information.



Søknad om barnehageplass
1 2 3 4 5 6 7 8
Opplysninger om foresatte

Har barnet to foresatte?
Ja ● Nei ○
Bor de foresatte på samme adresse?
Ja ● Nei ○
Opplysninger om foresatt to
Fødselsnummer
Fornavn
Etternavn

Meny Neste

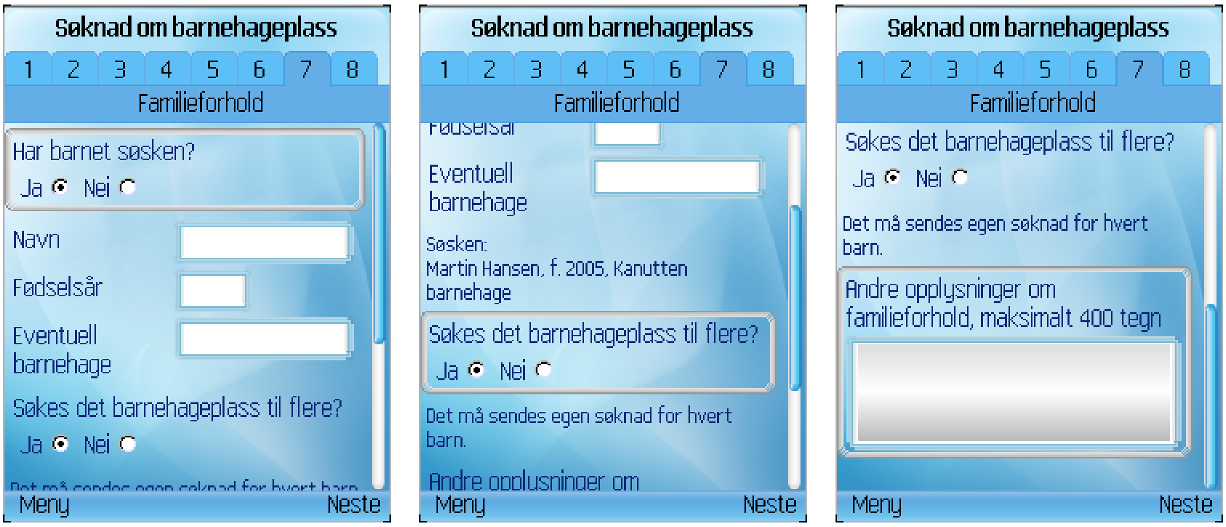
Søknad om barnehageplass
1 2 3 4 5 6 7 8
Opplysninger om foresatte

Adresse Sollia 3
Postnummer 7024
Poststed Trondheim
Telefon
Arbeidssted
Telefon, arbeid

Meny Neste

Figure 39. Page 6 of the kindergarten application form.

Figure 40 shows the seventh step in the application process, where the user states family information. The middle part of the figure shows that a sibling has been added to the application.


Søknad om barnehageplass

1 2 3 4 5 6 7 8

Familieforhold

Har barnet søsken?

Ja ● Nei ○

Navn

Fødselsår

Eventuell barnehage

Søkes det barnehageplass til flere?

Ja ● Nei ○

Det må sendes egen søknad for hvert barn

Meny Neste

Søknad om barnehageplass

1 2 3 4 5 6 7 8

Familieforhold

Fødselsår

Eventuell barnehage

Søsken:
Martin Hansen, f. 2005, Kanutten barnehage

Søkes det barnehageplass til flere?

Ja ● Nei ○

Det må sendes egen søknad for hvert barn.

Andre opplysninger om

Meny Neste

Søknad om barnehageplass

1 2 3 4 5 6 7 8

Familieforhold

Søkes det barnehageplass til flere?

Ja ● Nei ○

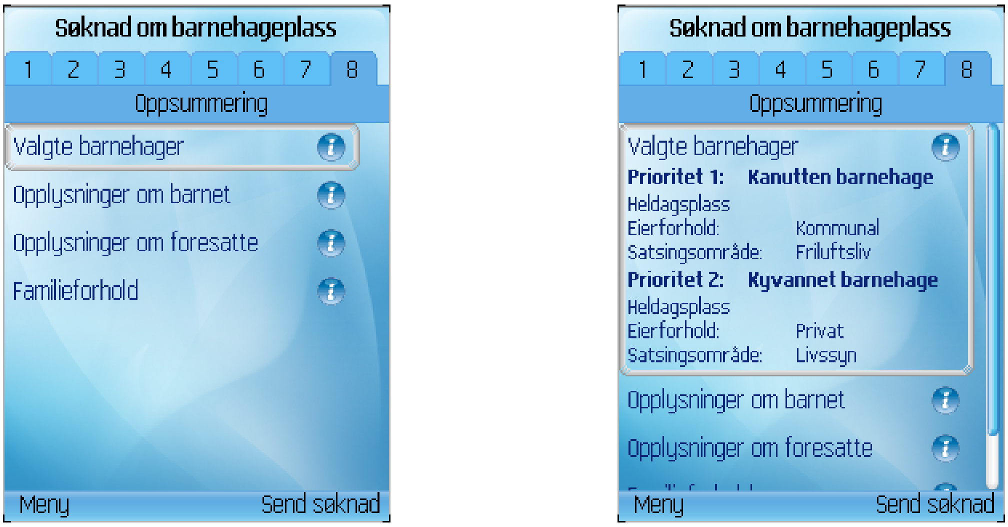
Det må sendes egen søknad for hvert barn.

Andre opplysninger om familieforhold, maksimalt 400 tegn

Meny Neste

Figure 40. Page 7 of the kindergarten application form.

The final form page is presented in Figure 41. The figure illustrates how the page is presented before the user chooses to look closer at a summary element, and how it is presented when the user has chosen to look closer at the top summary element. The right soft button has a ‘Send application’ command.


Søknad om barnehageplass

1 2 3 4 5 6 7 8

Oppsummering

Valgte barnehager

Opplysninger om barnet

Opplysninger om foresatte

Familieforhold

Meny Send søknad

Søknad om barnehageplass

1 2 3 4 5 6 7 8

Oppsummering

Valgte barnehager

Prioritet 1: **Kanutten barnehage**

Heldagsplass

Eierforhold: Kommunal

Satsingsområde: Friluftsliv

Prioritet 2: **Kyvannet barnehage**

Heldagsplass

Eierforhold: Privat

Satsingsområde: Livssyn

Opplysninger om barnet

Opplysninger om foresatte

Meny Send søknad

Figure 41. Part 8 of the kindergarten application form.

8.2 Prototyping the Design

The prototype uses Tellu’s framework and HCI-package. Tellu and Ericsson NorARC have as part of the ARTS research project (ARTS, 2006) developed the Java framework ServiceFrame. The framework enables model driven development by using UML 2.0. It was created to make development of internet and telecom services more rapid, by separating the services from the underlying dependencies of the system: ServiceFrame is layered on top of two other frameworks, ActorFrame and JavaFrame, where ActorFrame is on top of JavaFrame. This distinction makes the developers able to focus on the service they are developing, rather than technicalities (Bræk et al., 2002).

Tellu is in the process of developing their own HCI-package for J2ME applications which is integrated with ActorFrame, but it can also be used to make applications without the extensive framework, as illustrated in Figure 42. The HCI-package contains elements needed to build a user interface, such as drawing of a scroll bar, panels, text, input fields and drop-down boxes. The HCI-package lets the developer construct a user interface by creating a window, to which a panel can be added. The developer can add text elements, input fields, etc. as well as other panels to this panel. The HCI-package also offers adding such elements into groups, which can be added to a panel. Elements in a group have, by default, a frame surrounding them in the graphical user interface, to show that they belong together.

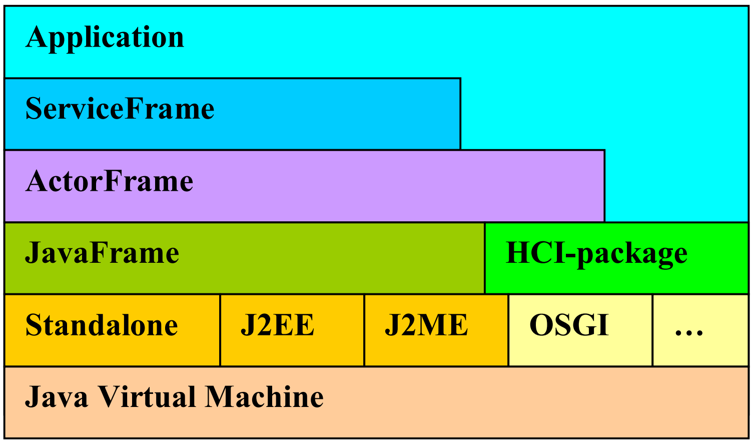


Figure 42. Architecture of frameworks with HCI-package.

When creating the prototype, the classes were divided into two packages: ‘framework’ and ‘kindergartenform’, see Appendix B for the Javadoc documentation. The source code of the prototype is in a separate ZIP-file. The package ‘framework’ contains classes that are generic, which Tellu can use in their framework and HCI-package. The classes ‘MapObject’ and ‘POIObject’ are copied from the package ‘map’ in ServiceFrame, as Tellu wanted the prototype to use local versions of classes in the framework that needed to be changed, rather than changing them directly. Code redundant for our needs was removed from the two classes.

MapObject

‘MapObject’ got a constructor which takes an image of a map as a parameter, because the map presented in the prototype is a static image. There was no need to fetch a map segment from a map server for this demonstration of multimodality.

POIObject

The classes used in the map package operate with UTM coordinates, as a map segment is supposed to be retrieved from a Norwegian map server. Forms will place points of interest (POIs) on maps by their coordinates. ‘POIObject’ therefore needed to have methods for finding a point of interest’s x and y coordinates within a given area in pixels, according to its and the area’s easting and northing coordinates. ‘POIObject’ also got the variables ‘id’, ‘selected’, for determining whether the point of interest is selected, ‘selectable’, to specify whether it can be selected, and ‘info’, for giving the POI an information object which contains information that is not coordinate-related.

The other classes in the ‘framework’ package are new.

POIInformation
The variable ‘info’ mentioned above is of the type ‘POIInformation’ in ‘framework’, a class that contains a point of interest’s name, street address, post code, post address, and the variable ‘extraPopupInfo’, which can be set if one wants the POI to be drawn with additional information “popping up” when selected.

MapElement
The class ‘MapElement’ resizes the map image to the requested size and handles the drawing of the image.

POIElement
‘POIElement’ handles the drawing of the points of interest, selection of them, drawing a selected point of interest’s popup information, keeping control of which ones have been chosen to bring along for further “operations”, as well as cancelling such choices.

MapLayout
‘MapLayout’ was created to calculate the layout of a panel containing map-related elements. The prototype has a panel containing objects of ‘MapElement’ and ‘POIElement’, and a layout manager of the type ‘MapLayout’. Each time the application draws the panel with its content, the layout manager first calculates the content’s positions and the panel’s width and height.

CheckBoxElement
The class ‘CheckBoxElement’ draws a check box with its associated alternative text. It handles whether it is selected or not. If it is selected, a dotted rectangle is drawn around it. It also handles and whether it has been checked or not. If it has been checked, the check box is drawn as checked and vice versa.

RadioButtonElement
The class ‘RadioButtonElement’ draws a number of radio buttons with their associated alternative text, and has the same functionality as ‘CheckBoxElement’ in addition to functionality for moving from radio button to radio button using the left and right keys.

The package ‘kindergartenform’ contains classes that are specific to the form we are prototyping.

KindergartenFormMIDlet
‘KindergartenFormMIDlet’ extends the MID Profile application class javax.microedition. midlet.MIDlet in order to be able to start the application, and initiates the class ‘FormWindow’.

FormWindow
‘FormWindow’ extends the ‘GuiWindow’ class of the HCI-package, and initiates and controls everything the screen contains. Since there is no server communication, the kindergarten information is created here, by making objects of ‘KindergartenPOIInfo’. The class creates an object of ‘FormTabbedElement’, to which it adds all form pages. It also adds commands to the soft buttons and reacts when they have been pushed.

KindergartenPOIInfo

The class ‘KindergartenPOIInfo’ extends ‘POIInformation’ described earlier, in order to get kindergarten objects containing information specific for kindergartens. Such information is the district the kindergartens lie in, the kindergarten’s ownership, area of focus, opening hours and types of places they offer, e.g. full day, half day etc. One can also add kindergarten activities by stating time, name and extended information of the activity. The activity information is used to create an ‘Activity’ object, which is added to a vector.

Activity

‘Activity’ contains the information stated above.

FormTabbedElement

This class extends ‘TabbedElement’ of the HCI-package because the kindergarten form needs to have soft buttons with commands for ‘Next’ and ‘Back’, which ‘TabbedElement’ does not support. ‘TabbedElement’ draws and handles “portfolio tabs” in a window, one for each form page added to it.

FormPageOne

In ‘FormPageOne’ the content of the first form page is created. All form pages extend ‘GuiPanel’ of the HCI-package, to get the desired behaviour and functionality. The layout of form pages are typically consisting of a number of ‘FormGroupelement’ objects. Each such object usually has a label and a form control. The label and form control have a frame surroundding them, unless the opposite is specified. ‘FormPageOne’ also has a method for checking whether the user has requested to see kindergartens in a map or in a list.

FormGroupElement

The class ‘FormGroupElement’ extends the ‘GroupElement’ class of the HCI-package. ‘GroupElement’ handles the drawing, layout and navigation of grouped elements. For the kindergarten form, we wanted to have a grid layout on most elements within a group, i.e. the elements are presented in two columns, see the screen shots from the prototype in the next chapter. We therefore needed to create a method which overrides the elementLayoutManager() method of ‘GroupElement’. The paint() method overrides paint() in ‘GroupElement’ because it was necessary to mark only one element as selected in cases where a ‘FormGroupElement’ contains several selectable elements. Finally, moveLeft() and moveRight() overrides the corresponding methods in ‘GroupElement’ because radio buttons require navigating between them with the left and right keys, which ‘GroupElement’ does not support.

FormPageTwoMap

In the second form page, we assume that the search performed by the criteria set in form page one gave 5 results, like the paper mock-up did. When the user chooses to see kindergartens in a map, ‘FormPageTwoMap’ is shown. The class has a separate panel containing the object of ‘MapElement’ and ‘POIElement’. Objects of ‘POIObject’, ‘MapObject’ and ‘MapBoundingBox’ are also created here, since they are not needed in any of the other classes. ‘MapBoundingBox’ is a class in the package ‘map’ in ServiceFrame, which contains a map’s right-most, left-most, bottom-most and top-most coordinates and coordinate-related logic. ‘FormPageTwoMap’ also handles adding and removing of commands on the right soft button by overriding the paint() method of ‘GuiPanel’: A kindergarten can be selected by pressing the number it is represented by. When a kindergarten is selected, the right soft button shall show the ‘Choose’ command. If the kindergarten is selected, the button shows the ‘Cancel’ command. When the maximum number of kindergartens to choose, which is 3, has been reached and the user selects a kindergarten which has not been chosen, the button shall show the ‘Next’ command.

FormPageTwoList
‘FormPageTwoList’ presents the kindergartens in a list. Each element in the list is an object of ‘GroupElement’, which contains the kindergarten’s name, address and a “Choose” check box.

FormPageThree
In ‘FormPageThree’ we assume that the user has chosen to see more information about two of the kindergartens. This class creates objects of ‘GroupElement’ and ‘FormGroupElement’ containing information specific to each of the two kindergartens. It also overrides the paint() method of ‘GuiPanel’ in order to add a ‘See daily routine’ command on the right soft button when one of the objects of ‘FormGroupElement’ containing the “See daily routine of the kindergarten with numbers or clock icons” is selected. ‘FormWindow’ has the logic for reacting to this command, which initiates ‘DailyRoutineWindow’.

DailyRoutineWindow
‘DailyRoutineWindow’ extends ‘GuiWindow’ of Tellu’s HCI-package. The class creates a ‘KindergartenPOIInfo’ object, to which a kindergarten’s daily activities are added. The class also has an object of ‘GuiPanel’, which contains ‘ActivityElement’ objects presented as a kind of list. The list elements are all presented with their key information, except for when the fire button has been pressed, in which case the current list element, i.e. ‘ActivityElement’, selected is expanded, and more detailed information is shown. This only applies if the element has more information to show. Each object of ‘ActivityElement’ is created by fetching an ‘Activity’ object from the ‘KindergartenPOIInfo’ object. ‘DailyRoutineWindow’ handles whether the activities should be presented with numbers or clock icons, according to what the user chose in the third form page. When the user chooses the ‘Back’ command, the class takes the user back to the third form page.

ActivityElement
This class extends ’GroupElement’ of the HCI-package and overrides several of the methods in the super class, in order to get the desired functionality. The class checks if an activity has a more detailed description of itself, in which case it draws an information icon inside the element. It also handles presentation of the detailed description, so that it is only presented when the user presses the fire button, and hidden again the next time fire is pressed.

FormPageFour
This class assumes that the user chose to apply to both of the kindergartens from the previous step, and that the place type “Full day” was chosen on both. The form page presents key information and objects of ‘FormGroupElement’ which request that the user prioritises the kindergarten applications.

FormPageFive
‘FormPageFive’ contains form controls for entering the child’s personal information. Information in fields which could have been pre-filled is set directly, since there is no separate personal information form.

FormPageSix
This class contains form controls for entering whether the child has two guardians, whether they live at the same address and information about the other guardian. Also here, information in fields which could have been pre-filled is set directly, since there is no separate personal information form. Fading of parts of the form that becomes irrelevant according to the user's input is not implemented, as this is not considered to be necessary to demonstrate the benefits of.

FormPageSeven
'FormPageSeven' contains form controls for entering family information. Fading of irrelevant fields is not implemented here either. The class overrides the paint() method of 'GuiPanel' in order to add an 'Add Sibling' command if the object of 'FormGroupElement' containing text fields for entering name, birth year and kindergarten of a sibling is selected. The class has a method for adding a group element containing sibling information, which can be called in 'FormWindow' when the right soft button is pressed. The sibling information is set directly in this class, as we only considered it necessary to demonstrate how it appears on the page.

FormPageEight
'FormPageEight' presents all the information which should have been entered in the form: As we wanted to spare the evaluators of entering information in each and every control, but yet wanted them to evaluate the 'look and feel' of the summary page, the summary information is created directly in this class. The summary information is added to four objects of 'SummaryElement', which are added to a 'GuiPanel' object. Like the panel in 'DailyRoutineWindow', this form page presents a list with high levels of information. If the user presses the fire button, the current list element is expanded and its detailed information is shown.

SummaryElement
This class functions like 'ActivityElement', except for that it does not check if there is any detailed information, as a summary element always has information which shall be shown when the user presses the fire button.

We tried to run the J2ME application on four different phones, two SonyEriccson and two Nokia. However, for unknown reasons, it did not work. The two SonyEriccson phones turned themselves off when the application was run, and the two Nokia phones refused to install it. The user evaluations were therefore executed using the phone emulator. The evaluations are presented in the next chapter.

9 Walkthrough Evaluation

In order to get feedback from users on aspects of the user interface of the electronic form, a walkthrough evaluation was executed. Ten users from two age groups, the ‘student’ group and the ‘adult’ group, participated. This chapter describes what was considered important to have users evaluate, the resources made in the evaluation preparations, the results of the walkthrough evaluation and an analysis of the results.

9.1 Elements That Should Be Evaluated

The first part of the research hypothesis of this investigation, as presented in Section 1.2.2, stated that certain design considerations can make forms feasible on mobile phones, particularily considering the device limitations and the burden of the users. In the previous chapters, several usability issues that need to be considered when making services on mobile phones, also considering groups of disabled users, have been pointed out. When designing the prototype, it was necessary to take a stand to several of these issues, described in the two previous chapters. In order to get an indication of whether or not the right design choices were made, we wanted to ask users of the form to evaluate the aspects described below.

General questions

We wanted to investigate the users’ general opinion to the feasibility of forms on mobile phones, by asking some general questions.

A service must not be too large and complex

In the demonstrated form, the company More’s experience on people not being able to handle more than 3-4 form pages on mobile phones, was ignored, except for presupposing that the kindergarten application form has a web equivalent. We therefore wanted to know how people found the size of the form when presented on a mobile phone. The form’s complexity was not to be evaluated, as it is assumed to depend on the size of the form, and because most of the form contains the same information and formulations as the forms (The Municipality of Bergen, 2007) and (More, 2007).

There must be a limited intensity of information

Only two pages of the demonstrated form are consciously designed to *keep the intensity of information low*: The screen showing a kindergarten’s daily routine and the last form page, containing a summary of the input information. Both of these allow the users to retrieve details if and when they want to, in order to decrease the information intensity. We wanted to know the users’ opinion on this functionality. The rest of the form probably needs some empty spaces here and there. However we were not interested in this form’s information intensity as such, only if the technique used was a good way of decreasing it.

The service must have a user interface that accommodates the device limitations

When one of several alternatives should be selected, the demonstrated form uses drop-down boxes and radio buttons. Radio buttons are only used when the number of alternatives are two, as navigating between radio buttons situated in rows and columns were considered to be less usable than navigating upwards and downwards in a list. In this success criterion, users not being able to see more than one alternative in a list was stated to be a usability issue. When using drop-down boxes, the users can see more than one alternative when they have activated the form control, and the list unfolds. Forms can have many information elements where drop-down boxes are appropriate form controls to use. We wanted to know how the users found the use of drop-down boxes on a mobile phone, as they can only see one alternative when the form control is not activated. We also wanted to know how they found the layout change when Tellu's drop-down boxes were opened, as that may be confusing or annoying.

The service's user interface should follow guidelines for design of forms

A 'Next' command was positioned on the right soft button most of the time, partly following Advice 1-2 in Chapter 2.4 in Appendix A. There were cases when this command was moved to the other commands under the 'Menu' choice on the left soft button, and the right soft button had other commands instead, as described in the two previous chapters. We wanted to investigate how the users found the appearing and disappearing commands, if they even did notice them. We also wanted to know whether they managed to figure out where the 'Next' command had been moved to, or if they thought that it did not exist at those places.

The demonstrated form has portfolio tabs with numbers on, in order to tell the user which step she is at in the form-filling process, and how much is left (Direction 2 in Chapter 2.3 and Advice 1-3 in Chapter 2.4 in Appendix A). We wanted to know how the users perceived these tabs and their numbers, in order to get an indication of whether it was a good or bad way to represent the steps of the process.

A service must accommodate variations in skills and abilities

The form is designed to accommodate variations in some skills and abilities, by following several guidelines. Basically, these should be just as applicable on mobile phones as on computers. The second part of the hypothesis of this investigation was that multimodality, in our case represented by offering visualisation of information as a modality, can be beneficial for the cognitive processing of content in electronic forms on mobile phones. The investigation had two target groups; 'people with moderate text-reading problems' and 'people with moderate number-reading problems'. A quantitative or qualitative evaluation regarding the truthfulness of this hypothesis is almost impossible, as we do not have access to people from the two target groups. However, as some people have the cognitive style to *interpret graphics better than text*, we wanted to investigate whether some preferred the multimodal representations over the lists with text.

There were two things in the demonstrated form that could prevent us from achieving a positive outcome:

- In the map, the users may not know the demonstrated area well or at all.
- In the daily routine screen, the clock icons are evidently "home made", which may decrease the user's ability to perceive the screen content, as mentioned earlier.

9.2 Evaluation Preparations

A scenario and a step by step assignment ('usage scenario' cf. Nielsen, 2007) were prepared for the users who were to evaluate the elements of the design mentioned above. In order for the users to get an impression of how the form user interface functions, they were asked to go through the form before they evaluated. As half of them, the evaluators in the student group were preparing for their exams, were very busy, it was decided to allow all users to enter "dummy" values rather than entering proper values in text fields, as that would have taken more time than the students had available.

Since the users needed to learn how to use the form controls, like pressing the fire button to open drop-down boxes, and help functionality was not implemented, it was also decided to let the users have a dialogue with the evaluation leader during their interaction with the form.

9.2.1 Scenario

In order to introduce the test situations, a scenario was presented to the users.

> *"You live at Byåsen in Trondheim and have two children, of which one is in a kindergarten and the other is not. You are living together with the mother/father of the children. You want to use your mobile phone to fill in a kindergarten application for the child not already in a kindergarten. The child needs a full-time place, and has a condition which favours the kindergarten application."*

9.2.2 Assignment

In order to observe which design elements could be confusing when the users first went through the form (such as the disappearance of the 'Next' command), the instructions in the assignment does not tell the users down to the smallest detail how to accomplish it.

Step 1: Set criteria for the kindergarten search

- Choose your part of town
- Continue to the next step

Step 2: Choose up to three kindergartens

- Check the results of the search
- Choose the kindergartens Bogane and Kanutten
- Cancel the choice of Bogane
- Choose Kyvannet
- Continue to the next step

Step 3: Enter final kindergarten choices and choice of type of place

- Check Kanutten's daily routine, shown with clock icons
- Read the detailed information in the elements "Different activities at the table" and "Indoors/outdoors play"
- Go back to the form page you came from
- Choose the two kindergartens
- Continue to the next step

Step 4: Set the priority of the kindergartens

- Set Kanutten to have the priority '1'
- Set Kyvannet to have the priority '2'
- Continue to the next step

Step 5: Information about the child

- Enter values in all text fields
- Continue to the next step

Step 6: Information about guardians

- Enter information about the other guardian of the children
- Continue to the next step

Step 7: Family information

- Add a sibling to the application
- Set 'No' on "Are you applying for several children?"
- Continue to the next step

Step 8: Summary

- Read through each summary element.

9.2.3 How to start the evaluation

In order to make the evaluation go smoothly, particularly for the users, a list regarding how to "kick off" the evaluation was made:

1. Properly introduce yourself.
2. Explain to the user that we need help to evaluate parts of the design. Make it clear that the design is the test object, not the user.
3. Make sure they know that they can abort or pause the evaluation at any time.
4. Tell them that you will be there to guide them through the form-filling process, and that they can ask questions at any time.
5. Tell them that after they have finished the form-filling, we will ask some questions and also go back to several pages in the form.
6. Ask them to, and teach them how to, think aloud.
7. Tell them that they can fill in "dummy" values in text fields.
8. Tell them that is important to read all information in the form.
9. Describe the scenario and the assignment. Let them have it on paper, so they can see it as well as hear it.
10. Ask if they have any questions, and start the evaluation.

9.2.4 Evaluation questions

Since our assignment was not to check the *usability* of the form *as a whole*, but to investigate the aspects described in the previous chapter, the users had to look more carefully at the design aspects after having finished the form-filling, and were being asked some questions related to these as well as some general questions. The questions that were prepared for the evaluation are presented below.

Age:
Time and place:
Experience with mobile phones: Calling, messages / Other experience (specify)

- "Do you think you would use your mobile phone for filling in applications, registration forms and reports?"

If the answer is no:

  - "Why not?"

If the answer is yes:

    - "In which situations do you think you would be using your phone for such tasks, rather than using a computer or the paper versions?"

- "What do you think of the size of this form?"

If the answer is that the form is too big or similar:

    - "At what point would you say that you had had enough?"
    - "Why do you think you felt that it was too big?"

Go to the last form page.

- "When you activated one of these elements, a lot of information appeared. Did you like that you had the possibility of hiding and revealing information, or would you rather have had all of the information presented at once?"

If the answer is all information at once:

    - "Why would you rather have all information at once?"

Show the user the first form page, which has three drop-down boxes.

- "Did you dislike that you could not see more than one alternative at first, or think it was confusing?"

If the answer is yes:

    - "Can you describe why you have this feeling?"

Ask the user to look at the page, and then activate one of the drop-downs.

- "Would you say that you became confused or annoyed when the layout of the form suddenly changed, and elements that you earlier saw were gone?"

From the first form page, go to the second with map presentation and select a kindergarten.

- "Did it take you long to notice that the right button had a different command when you first saw this?"
- "What did you think had happened to the 'Next' command?"

Go to the third form page and select the top "See daily routine with numbers or clock icons" group element. Ask the user to look at the page, then choose the group element below.

- "Did you see that the 'Next' command came back, when you first filled in the form?"
- "Did you think that the 'See daily routine' command was connected only to the "See daily routine with numbers or clock icons" group elements?"

- "What do the portfolio tabs mean to you?"

Go back to the second form page with map, and ask the user to study the page. Then present the second form page with kindergartens in a list, and ask the user to explore it.

- "Which of the two representations did you like best, and why?"

Go to the daily routine window with clock icons, and ask the user to study it. Then present the daily routine with points of time represented as numbers.

- "Which of the two representations did you like best, and why?"

- "Do you have anything you would like to remark on?"

9.3 Evaluation Results

The evaluation leader observed the users' actions during their walkthrough of the form, in addition to answering their questions. Interesting aspects were noted and used in the question sessions, in connection with the pre-made questions. The essence of the answers from the evaluation questions are presented in the tables below. Appendix C contains the full answers.

The users had little experience with use of mobile phones for other tasks than calling and sending messages. Only one states that he uses several programmes on his phone, while others use typical built-in functionality, like a calendar, see Table 1.

Age	Experience with mobile phones: Calling, messages / Other experience (specify)
25	Other: Radio, MP3, calendar.
28	Other: Calendar.
26	Other: Several programmes, calendar, text editing programmes.
24	Other: Radio, MP3, calendar.
22	Other: Calendar.
38	Calling, messages.
39	Calling, messages.
45	Calling, messages.
50	Other: MP3.
49	Other: Calendar.

Table 1. Experience with mobile phones.

Six of the users could not imagine themselves using a mobile phone for filling in forms, while two of the users could, given that the forms are not too large. As much as four of the five eldest users gave a negative response, see Table 2.

Age	Do you think you would use your mobile phone for filling in applications, registration forms and reports?
25	Yes, if they are not too large.
28	Yes, if they are not too large.
26	No.
24	No.
22	Yes.
38	No.
39	No.
45	No.
50	No.
49	Yes.

Table 2. General question regarding feasability of forms on mobile phones.

Table 3 shows that all but one of the users who gave a negative response stated device limitations, like limited size of the screen, to be the reason.

Age	Follow-up question: If the answer is no: Why not?
26	Too hard to enter input.
24	Too small screen.
38	Too small screen.
39	Too small screen, and device.

45	Too small screen, too hard to enter input.
50	Does not want to learn how to interact with mobile phone forms.

Table 3. Follow-up question, why not feasible.

Most of the users who responded positively think they would use their mobile phone for form-filling in situations where they do not have access to a computer. Two of the users also mentioned other circumstances, see Table 4.

Age.	**Follow-up question: If the answer is yes: In which situations do you think you would be using your phone for such tasks, rather than using a computer or the paper versions?**
25	When without access to a computer. In any situation, if the form information is not very critical.
28	When without access to a computer.
22	When without access to a computer, or when remembering a form that needs to be filled-in and delivered, in the last minute.
49	Does not know.

Table 4. Follow-up question, in which situations.

As Table 5 shows, two of the users found the size of the form too large.

Age	**What do you think of the size of this form?**
25	OK.
28	Too large.
26	OK.
24	OK.
22	OK.
38	OK.
39	Too large.
45	OK.
50	OK.
49	OK.

Table 5. How they found the size of the form.

The two users who found the size of the form too large were asked some follow-up questions. One of them suggested more use of letting the user choose which detailed information to see, in order to make the size of the form seem smaller, see Table 6.

Age	**Follow-up questions:**
	At what point would you say that you had had enough?
28	Does not know.
39	Would only use the phone to confirm "Yes"/"No" in forms.
	Why do you think you felt that it was too big?
28	More comfortable on a computer screen. More use of the hiding/revealing information technique would be beneficial.
39	Would only use the phone to confirm "Yes"/"No" in forms.

Table 6. Follow-up questions regarding the size of the form.

Table 7 shows that all of the users were positive to the technique that lets them choose which detailed information to see. Two of the users also stated that it helped them keep focus on what they were reading.

Age	Go to the last form page. When you activated one of these elements, a lot of information appeared. Did you like that you had the possibility of hiding and revealing information, or would you rather have had all of the information presented at once?
25	Yes.
28	Yes.
26	Yes.
24	Yes.
22	Yes, helps you keep focus.
38	Yes, helps you keep focus.
39	Yes.
45	Yes.
50	Yes.
49	Yes.

Table 7. Hiding and revealing information technique.

Table 8 shows that none of the users claimed to dislike or find confusing that drop-down boxes only show one alternative when they have not been opened.

Age	Show the user the first form page, which has three drop-down boxes. Did you dislike that you could not see more than one alternative at first, or think it was confusing?
25	No.
28	No.
26	No.
24	No.
22	No.
38	No.
39	No.
45	No.
50	No.
49	No.

Table 8. One alternative visible in drop-down boxes.

Neither did any of the users claim to be confused or annoyed when elements below opened drop-down boxes were moved downwards, rather than drawing the opened list over them, as Table 9 shows.

Age	Ask the user to look at the page, and then activate one of the drop-downs. Would you say that you became confused or annoyed when the layout of the form suddenly changed, and elements that you earlier saw were gone?
25	No, helps you keep focus.
28	No.
26	No, better than drawing over them.
24	No, but prefers drawing over.
22	No.
38	No.
39	No, but prefers that the screen shows only the alternatives of the current drop-down.
45	No.
50	No.
49	No.

Table 9. Layout change in drop-down boxes.

All but one of the users instinctively used the fire button to choose kindergartens. Five of these first noticed that the right soft button had a different command when they were going to cancel the choice of Bogane kindergarten, one of them noticed while choosing by using the fire button. Another first noticed when she had made the final choices of kindergartens by using the fire button to both choose and cancel choices, and were going to go to the next form page. Two of the users did not notice the command changes at all, as they used the fire button for choosing and cancelling, and the keys for moving right and left, for navigating between form pages. Only one of the users noticed the command change right away, and used only the right soft button to choose and cancel choices, see table 10.

Age	From the first form page, go to the second with map presentation and select a kindergarten. Did it take you long to notice that the right button had a different command when you first saw this?
25	Noticed first when cancelling choice. Used the fire button to choose.
28	Noticed first when cancelling choice. Used the fire button to choose.
26	Did not notice. Used the fire button to choose and cancel.
24	Noticed first when cancelling choice. Used the fire button to choose.
22	Noticed first when cancelling choice. Used the fire button to choose.
38	Noticed right away.
39	Noticed while using the fire button to choose.
45	Noticed first when she was going to go to the next step. Used the fire button to choose and cancel choices.
50	Did not notice. Used the fire button to choose and cancel.
49	Noticed first when cancelling choice. Used the fire button to choose.

Table 10. Change of commands on the right soft button.

Several of the users instinctively used the right and left keys to navigate between form pages. Only one of these consistently navigated with those keys. The others alternated between using the right and left keys and the soft buttons. On the form page with the map, seven of the users navigated to the next form page by using the right key. Three of these users did not think that the map page has a ‘Next’ command, while two of them stated to not know or to not have considered it. Further questions on what these two now thought gave no results. Five of the ten users thought, correctly, that the command could be found in the menu on the left soft button, and three of these used it. Table 11 shows the answers according to the users’ age. The three users who never used the right and left keys when navigating, were among the eldest.

Age	What did you think had happened to the ‘Next’ command?
25	Did not think it existed on this page. Assumed that she could navigate with right and left keys.
28	Did not think it existed on this page. Had not really considered it, as he navigated with the right and left keys.
26	Does not know, navigated with the right and left keys through all of the form.
24	Had not really considered it, as he navigated with the right and left keys.
22	Navigated with right and left keys, but assumed it could be found in the menu.
38	Navigated with right and left keys, saw no reason for there to be a ‘Next’ command.
39	That it could be found in the menu.
45	That it could be found in the menu.
50	Did not notice it as he navigated with right and left keys, but assumed that it could be found in the menu.
49	That it could be found in the menu.

Table 11. What happened to the disappeared ‘Next’ command.

On the form page where two identical group elements have a 'See daily routine' command attached to them, one user noticed that the right soft button's command changed when another group element was selected. Six of the users noticed that the button's command had changed to 'Next' first when they reached the bottom of the page and were going to go to the next form page. The three remaining of the ten users did not notice the command change at all, see Table 12.

Age	Go to the third form page and select the top "See daily routine with numbers or clock icons" group element. Ask the user to look at the page, then choose the group element below. Did you see that the 'Next' command came back, when you first filled in the form?
25	Did not notice it until she needed it.
28	Yes.
26	No, did not look much at the menu line as he navigated with right and left keys.
24	No (probably for the same reason as the 26 year-old).
22	Did not notice it until she needed it.
38	Did not notice it until she needed it.
39	Did not notice it until he needed it.
45	Did not notice it until she needed it.
50	No, said he did not look at the commands on that button as long as he found everything he needed in the form.
49	Did not notice it until she needed it.

Table 12. The appearing 'Next' command.

Six of the users claimed to understand that the 'See daily routine' command is attached to the "See daily routine with numbers or clock icons" group elements, while the rest said they had not really thought about it, see Table 13.

Age	Did you think that the 'See daily routine' command was connected only to the "See daily routine with numbers or clock icons" group elements?
25	Yes.
28	Yes.
26	Did not consider it.
24	Did not consider it.
22	Yes.
38	Yes.
39	Yes.
45	Did not consider it, but said it seemed reasonable.
50	Yes.
49	Did not consider it.

Table 13. Command connected to a specific element on the screen.

As Table 14 shows, all of the users perceived the portfolio tabs as steps in the form-filling process or form pages.

Age	What do the portfolio tabs mean to you?
25	Steps in the form-filling process.
28	Steps in the form-filling process.
26	Steps in the form-filling process, particularly because of the numbers.
24	Steps in the form-filling process.
22	Steps in the form-filling process.
38	Steps in the form-filling process.
39	Steps in the form-filling process.

45	Steps in the form-filling process.
50	Steps in the form pages, progression.
49	Page numbers of the form.

Table 14. Portfolio tabs.

Table 15 shows that seven of the users liked the form page with the map better than the form page with kindergartens in a list, because of the possibility of seeing where the kindergartens are situated. Further questions on whether they felt that it was easier to perceive the information in the map than in the list, gave no results.

Age	**Go back to the second form page with map, and ask the user to study the page. Then present the second form page with kindergartens in a list, and ask the user to explore it. Which of the two representations did you like best, and why?**
25	Wants a combination of the two, with kindergartens in a textual list and possibility of seeing each kindergarten's position in a map.
28	Map, beneficial to get an overview of the kindergartens' positions.
26	List, but the map is beneficial for orientation.
24	Map, beneficial to get an overview of the kindergartens' positions.
22	Map, beneficial to get an overview of the kindergartens' positions.
38	Map, beneficial to get an overview of the kindergartens' positions.
39	Map, beneficial to get an overview of the kindergartens' positions.
45	Map, beneficial to get an overview of the kindergartens' positions.
50	Map, beneficial to get an overview of the kindergartens' positions.
49	No favourite. Liked the map, as it was beneficial to get an overview of the kindergartens' positions, and the list as she could se the addresses (in text) of several kindergartens at the same time.

Table 15. Map and list representation of kindergartens.

All of the users preferred points of time represented by numbers, as Table 16 shows. The general complaint was that it was too hard too perceive the content of the clock icons, and they were too small. One of the users suggested enlarging the selected activity's clock icon.

Age	**Go to the daily routine window with clock icons, and ask the user to study it. Then present the daily routine with points of time represented as numbers. Which of the two representations did you like best, and why?**
25	Numbers, most surveyable. It was a bit hard to perceive the content of the clock icons.
28	Numbers, easiest to perceive the meaning of. Suggests enlargement of the selected activity's clock icon, in order to make it stand out and make it easier to interpret.
26	Numbers, easiest to perceive the meaning of.
24	Numbers. It was hard to perceive the content of the clock icons. Did not think of the icons as something else than pictures.
22	Numbers. It was hard to perceive the content of the clock icons, they were too small.
38	Numbers. It was hard to perceive the content of the clock icons, they were too small.
39	Numbers. It was hard to perceive the content of the clock icons, they were too small and dim.
45	Numbers. They were easier to perceive.
50	Numbers. They were easier to perceive, the clock icons were too dim. Says he generally perceives point in time represented by numbers better than by clock icons.
49	Numbers. The clock icons were too dim.

Table 16. Points of time represented graphically and with numbers.

Table 17 shows the users' comments on the user interface of the form.

Age	Do you have anything you would like to remark on?
25	Wants more information about the kindergartens.
28	Wants possibility of changing the detail level in the map and navigating with right/left/up/down keys.
26	Wants possibility of navigating with the right and left keys in the map. Form page 3 has too much information and is not very surveyable. Suggests use of the hide/reveal information technique used in two other places.
24	No.
22	Pre-filling is beneficial.
38	Wants to use only the right and left keys for navigating. Beneficial to have the possibility of filling in forms anywhere, at any time.
39	No.
45	Nice to have the possibility of choosing presentation modality.
50	The selected group element at any time should be made more perceivable, e.g. by a different background colour.
49	No.

Table 17. Remarks.

The users in the age 22-28 filled in the form with few hesitations during the process, which lasted for an absolute maximum of ten minutes. Users in the age 38-50 spent more time considering which actions to take next, using no less than ten minutes, and up to twenty minutes. However, in both age groups, some of the time was spent on commenting and discussing the user interface during the form-filling process.

9.4 Analysis of the Evaluation Results

General questions

When asked a general question concerning the test users' attitude towards using their mobile phones for form-filling, as much as six of the ten users responded negatively, see Table 2. It was no surprise that four of the five eldest were among these, as it is well known that young people tend to be quicker at embracing new technologies, trends and innovations. The users' negativity or acceptance may also be related to how little or much experience they have with mobile phones, although the user who had most experience with using his phone for other purposes than calling and sending messages, had a negative attitude. The three people who only used their phones for calling and sending messages were all negative. Also not surprisingly, the main reason for the users' negativity is the device limitations "small screen" and "awkward to enter input", see Table 3. Maybe, if people gain more experience with using their phones for other than the most basic functionality, the other tasks would be easier to fulfil, and their attitude towards form-filling could change.

The users who responded positively were asked a follow-up question in order to find out in which situations they could see themselves using their phone for form-filling, see Table 4. The intention of the question was to investigate whether they would do this in the mobile context "any time, any place", but the answer was mainly "when they did not have access to a computer", i.e. large screen and keyboard. Only one of them mentioned the mobile context, provided that the information of the form is not critical. We are without access to a computer part of or much of the time in a mobile context, but these users may also have been thinking of their home situation where the environment is calm and the disturbances are fewer than e.g. on the street.

A service must not be too large and complex

As mentioned earlier, we wanted to investigate if the size of the demonstrated form, a kindergarten application, was too large for the users, i.e. whether the cognitive load was too heavy. Only two responded that they considered the size to be too large, see Table 5. In order to find out if anyone supported the experience of the company More, that people could not handle more than 3-4 form pages, we asked the two users to point out where they were "fed up". We also asked if they could state a reason for their opinion, to find out whether it was too much information for them to take in on the small screen, or if there were any other factors that led them to feel this, see Table 6. These questions did not give us any answers of a quality detailed enough. However, one of the users suggested more use of the hiding/revealing information technique demonstrated in the daily routine window and the last form page, the sum-ary page, in order to make the form pages more surveyable and seem shorter. As none of the users had any problems with these elements, i.e. they figured out that they had to press the fire button to see the detailed information, it may be beneficial to use the technique more extensively. Further user tests would show whether users find more use of the technique annoying, as they would have to constantly use the fire button to see information and form controls, confusing, as they will not see the full content of a form page when they enter it, or beneficial.

There must be a limited intensity of information

We tried to decrease the information intensity by using the hiding/revealing information technique, as mentioned above. All of the users claimed to like this technique. Two of them even stated that it helped them keep focus on what they were reading, see Table 7. However, the page shown when asking this question was the summary page, which contains a lot of information. Had the question been asked when showing them the daily routine window, the response may have been otherwise, as the hidden information was quite short. We may conclude that on form pages where the information intensity is very high, the possibility of choosing which information to see is likely to be beneficial. Also here, further user tests in order to see how large amounts of information make the technique necessary and beneficial, is needed.

The service must have a user interface that accommodates the device limitations

The users were asked to state whether they disliked seeing only one alternative in unopened drop-down boxes or if they found it confusing. All the users answered 'no' to this question, see Table 8. However, the answers may be misleading. Users may be afraid to appear as dumb, they may not be aware of that they were confused, etc. In order to get a dependable answer to whether they were confused or not, quantitative user tests of drop-down boxes versus lists where all alternatives, or as many as the size of the screen allows, are always visible, should be executed. The tests should measure the time it takes to make the choices, as well as the number of wrong input. People are today accustomed to use drop-down boxes in e-forms, which also several of our users expressed. That none stated to dislike not seeing more than one alternative at first is therefore not surprising, as they may not have considered the alternative.

When a drop-down box is opened, the elements below it are moved downwards to avoid drawing the expanded list over them, which normally is done. We were worried that the users may become confused or annoyed when the form's layout suddenly changes. None of the users confessed to get confused or annoyed, see Table 9. However, also here the users may be afraid to appear as dumb or be unaware of their confusion. Two of the users claimed to prefer other ways of presenting the expanded list, while two others meant that the layout change was beneficial. It is therefore hard to conclude that this way is the best[6]. Although, we claim to have an indication of that it is satisfactory, as none of the users reacted in body language or through their statements, when the layout of the pages with drop-down boxes changed during the form-filling.

The service's user interface should follow guidelines for design of forms

We wanted to investigate how the users found, if they even noticed, the command changes, and what they thought had happened to the command that had been replaced. That as many as nine of the ten users instinctively used the fire button to choose kindergartens was a surprise, the possibility of choosing kindergartens with this button was implemented considering experienced users. The reason to why so many used it, is probably that the fire button is used for making choices on many phones. In the first case we asked them about, see Table 10, most of the users did not notice the command changes until they had been interacting with the form page for a while. Two of the users did not even notice at all.

One reason to why they did not notice, may be that as the users found what they needed in the form page and instinctively used appropriate buttons, there was no need to check the commands available on the menu line on the bottom of the screen.

Another reason may be that the form-filling was executed on a phone emulator, as the application for unknown reasons could not be installed on real mobile phones. Many of the users repeatedly tried to manipulate form controls by using the mouse cursor directly on the emulated phone's screen, they had problems relating to the emulated phone as a real mobile phone. If they had tried the application on a real phone, they may have associated the commands on the soft buttons better with the buttons on the phone, and been more aware of the commands. Using an emulator on a computer when evaluating has some benefits, however. The users can easier see, thereby easier judge what they see, and the observer of the users can easily see their actions in the form user interface, which would be quite difficult if they were to use an actual mobile phone. Observing the user actions on an actual phone might also have made the users feel uncomfortable, as one practically would have had to lean up against them because of the small-sized display.

However, almost *instinctive use* of the fire button is a welcome surprise. Further, quantitative tests could show whether the 'Choose kindergarten'/'Cancel choice' commands on the right soft button could be left out entirely.

Several of the users also instinctively used the keys for moving right and left to navigate between form pages, although not consistently. Asking the users what they thought had happened to the 'Next' command on the second form page when they first filled in the form, see Table 11, was probably not very helpful for getting any indications of the usability of moving it, as they did not have any preferences. They needed to learn that every page has such a command, which they still had not after completing the form's eight pages. Of the five users who thought the command could be found under 'Menu', four were among the five eldest users. Three of these were the only users who did not navigate with the right key to the next form page. The eldest may have been more observant than the youngest, and reflecting more over the possibilities they had. As mentioned earlier, the youngest spent shorter time on completing the form-filling process, and may have been quicker to continue to the next step

[6] Fairly enough, these walkthrough evaluations do not prove any design to be correct or superior. They only provide the investigation with indications of how to design electronic forms in a good manner, or how definitely not to do it.

when they were finished with the current. That so many used the right and left keys when navigating was also a surprise. However, as they alternated between keys and commands, we would not consider removing the 'Next' and 'Back' commands on the soft buttons.

When we asked the users whether they saw that the 'Next' command reappeared when they chose another element on the third form page, only one did, see Table 12. That six of the users first noticed it when they reached the bottom of the form page, indicates what was mentioned earlier, that as long as the users found what they needed in the form page, there was no need to check the soft button's commands. The users' interaction with the "See daily routine with numbers or clock icons" elements supports this indication, when their task was to look at the daily routine with clock icons, they looked on the screen until they saw the 'See daily routine' command, and pressed the right soft button. The problems with relating to the emulated phone as a real phone may also be relevant here.

In order to find out whether the users could connect a specific, "unordinary" soft button command to only one of the elements on the screen, they were asked to state whether they understood that the 'See daily routine' command is attached to the "See daily routine with numbers or clock icons" group elements, see Table 13. Four of the users said they had not considered it, and one of these added that it seemed reasonable. Again, some of the six users who answered positively may have been afraid to appear as stupid or similar. A different wording of the question may have given other results. However, as all of the users found and used the command when they were going to check the kindergarten's daily routine, it successfully filled its purpose. Since several navigated to the next page by using the 'Next' command when they had reached the bottom of the page and the 'Next' command had replaced the 'See daily routine' command, and none of these found the command change confusing or unnatural, it seems like a good solution.

The portfolio tabs were a success, all of the users perceived them as steps in the form-filling process or form pages, see Table 14. We have a clear indication of that representing the steps of the process in this manner is a well functioning solution on mobile phones.

A service must accommodate variations in skills and abilities

Seven of the users preferred the kindergartens presented in a map over the kindergartens in a list, see Table 15. They stated the possibility of getting an overview of the kindergartens' positions to be the reason. Lack of knowledge about the demonstrated area did not seem to be a hindrance. As mentioned earlier, further questions on whether they felt that the information in the map was easier to perceive, did not give any useful answers. The users did not seem sure what to answer to these questions, and usually ended up talking of other, irrelevant aspects. However, presenting selectable points of interest in a map on mobile forms seem to be a good solution in cases where location is one of the criteria for the form-fillers.

The points of time represented by numbers were preferred by all the ten users, see Table 16. The clock icons were stated to be too small and dim, and therefore hard to perceive the content of. That the low quality of the clock icons could decrease the users' ability to perceive the content of the screen, was stated to be an issue before the user evaluations. An interesting idea was suggested by one of the users: To enlarge the current selected activity's clock icon. A combination of this, as well as professionally made icons, may have given quite different answers.

The additional comments made by the evaluators, see Table 17, were mostly irrelevant for our investigation. That one commented that she found pre-filling beneficial, and another that it was nice to be able to choose presentation modality, was although good to hear.

The results from the walktrough evaluation provided feedback on the usability of the design decisions that were made, taken from the success criteria, and could help us determine what should be left out, which ideas were good, and which ideas were interesting to investigate further in order to find specific techniques and controls that will make forms feasible and usable on mobile phones. The most important results were these:

- 60% of the evaluators were negative towards using their mobil phones for form-filling, mainly because of the device limitations.
- 80% did not find the size of the demonstrated form too large. A more extensive use of the technique for hiding/revealing information is suggested by a user that finds the form too large.
- All of the evaluators find the hiding/revealing information technique beneficial on a page with high information intensity.
- None of the users claimed to have any problems with the drop-down boxes nor the layout change when they were opened.
- 90% instinctively used the fire button to choose kindergartens, several also instinctively used the keys for moving left and right when navigating between pages.
- Command changes seem to be unnoticed until they can not find what they need in the form itself, but going through the form on actual phones may have given different observations.
- 50% assumed that the command ‘Next’ had been moved to the menu, while 30% did not think it existed on the form page. The evaluators had not learned that every page has a ‘Next’ command, regardless of position.
- The representation of progress in the form was understood by all.
- 70% preferred the presentation of kindergartens in a map, for the possibility of getting an overview of the kindergartens’ positions. Asking whether they perceived the map representation better than the list did not give any results.
- None preferred the points of time represented by clock icons, mainly because they were too small and dim and hard to perceive. An evaluator suggested that the current activity’s clock icon should be enlarged.

10 Discussion and Conclusions

This report has documented an investigation on how to make electronic forms feasible on mobile phones. The investigation addressed the design of mobile form user interfaces. Great design challenges are introduced by the different, smaller-sized user interface than the one of a typical desktop computer. The constraints of two target groups within the 'cognitive disabilities' category have been accommodated: people with moderate *text-reading* problems and people with moderate *number-reading* problems. Since the number of people with low literacy and numeracy skills is quite high (OECD, 2005), and the solutions for people with cognitive disabilities are scarce, users with these challenges were considered important to provide assistance to.

The research hypothesis (Section 1.2.2) for the investigation contained two assertions: (1) that specific techniques and form controls can make forms feasible on mobile phones and relieve the user burden, and (2) that offering visualisation of information as an additional modality in forms on mobile phones can be beneficial for the cognitive processing of content in such forms. The research work that was executed in order to come to a conclusion regarding the correctness of the assertions in the hypothesis, led to answers as well as aspects to be treated further.

As far as the first assertion is concerned, the investigation led to a list of success criteria with suggestions on how to achieve several of them (Section 3.5.2 and Chapter 4.4). The list is not claimed to be a complete list on how to make forms feasible on mobile phones. Nevertheless, we assert that these criteria are essential in universal design of the HCI on mobile phones. The suggested success criteria were:

- It must be possible to pause the form-filling process
- A service must not be too large and complex
- There must be a limited intensity of information
- Offer services to appropriate target groups
- The service must not stop the form-filling process when loading information
- The service must have a user interface that accommodates the device limitations
- The service's user interface should follow guidelines for design of forms
- The service's graphical user interface must be visually pleasing
- A service must accommodate variations in skills and abilities

Efforts to achieve the success criteria that were relevant for the focus of our investigation, i.e. criteria connected to the usability of the user interface, were demonstrated by a paper mock-up (Chapter 7) and a J2ME prototype (Chapter 8). Following design decisions evolved during this phase of the research work:

There must be a limited intensity of information
The form demonstrated a technique for allowing the users decide which information to reveal on form pages where the information intensity can be perceived as high.

The service must have a user interface that accommodates the device limitations
The prototypes demonstrated several of the suggested efforts to achieve this criterion. The form offered selection controls over typing controls where possible, by using lists, drop-down boxes, radio buttons and check boxes. The map can also be seen as a

selection control. We believe that having a history and dictionary database associated with form controls that take input, is a very beneficial effort to decrease the burden of the user when making text input. However, this was not demonstrated. Pre-filling of fields was also illustrated by the prototypes. Another technique commonly used is fading of fields. Used in forms on mobile phones, the technique could relieve the burden of the users regarding to the device limitations. Fading of fields was not demonstrated either.

The service's user interface should follow guidelines for design of forms
The prototypes demonstrated several guidelines for the design of forms. A 'Next' command is the first choice available, except for cases where it is very important to have other commands as the first available choice. There is also an indicator of how large the form is, and how far the user has progressed in it. There are no commands for resetting the form fields.

A service must accommodate variations in skills and abilities
In order to achieve this criterion, as well as test the second part of our hypothesis, we offered a graphical modality in addition to the textual, of kindergartens' geographical information and points of time regarding the kindergartens' activities in their daily routine. Offering the possibility of enlarging text was also suggested, but not implemented as it was demonstrated in (Flaten, 2006). The form does not emphasise words by using italics or all capital letters, as this is advised against. Instead, bold fonts and larger text sizes were used as this is often recommended for people with cognitive impairments. The form had a grid layout to make the content easier to perceive. The paper mock-up used the Arial font, as it is one of the recommended sans serif fonts for people with dyslexia. The J2ME prototype used the emulated phone's built-in fonts, as a connection to a server with additional fonts was not implemented. These efforts to accommodate variations in skills and abilities were incorporated in the design of the demonstrated form without pointing out or addressing specific user groups, following the Universal Design principle "Equitable Use" which is very important considering the dignity perspective.

Walkthrough evaluations were executed to get feedback on specific design aspects, as well as investigating the general opinion to the feasibility of forms on mobile phones (Chapter 9). Our concerns were:

A service must not be too large and complex
The experience of the company More was that people could not handle more than 3-4 form pages (cf. Section 3.5.1). If that is correct, the selected case would be one of very many forms that are not feasible to represent on mobile phones.

There must be a limited intensity of information
We wanted feedback on whether the suggested technique for allowing the user to hide/reveal information was a good way of decreasing the information intensity.

The service must have a user interface that accommodates the device limitations
We wanted to investigate whether the users became confused by the use of drop-down boxes, as they only present one alternative when the control is not activated, and as expanding a drop-down box causes the elements below to move downwards, rather than drawing over them.

> **The service's user interface should follow guidelines for design of forms**
> The 'Next' command was sometimes moved as other, more important commands needed to be presented at the right soft button. Although we based our design on the assumption that users understand or learn that they have to check their command alternatives when they need a specific command, we were worried that this replacement could be confusing. As a consequence, they might not understand that the 'Next' command was not removed entirely from the form. We also wanted to know how they perceived the indicator of form size and progress.
>
> **A service must accommodate variations in skills and abilities**
> Since we did not have the possibility of testing the graphical modalities on people in the *real* target groups, we wanted to investigate whether any of the evaluators preferred the graphical modality, as some of them could have the cognitive style to interpret graphics better than text.

The results from the evaluations provided us with feedback to use when discussing the addressed aspects.

General opinion to feasibility of forms on mobile phones
Although six of the ten evaluators were negative when being asked whether they would use their mobile phone for form-filling, we think that this view is likely to change over the next years, particularily among the youngest users. As the extent of the self-service society increases, more vendors and authorities adopt the mobile phone as one of their service channels. Along with this development, forms on mobile phones become more common, and people will get accustomed to the thought of using their mobile phones for this purpose. The main reason for their negativity was the physical limitations of the device itself. These are likely to change with the technological advances, thereby decreasing peoples' remonstrances.

A service must not be too large and complex
Only two of the ten evaluators considered the demonstrated form to be too large. One of these was extremely negative towards mobile forms, stating that he would use them if he only had to do yes/no confirmations. That the remaining eight did not find the form too large was positive, as it indicates that a large number of forms may be feasible to represent on mobile phones. However, asking users who followed an assignment that told them what to enter on several places, may not be a good basis to claim that forms of this size are feasible. Nor is a test situation where users complete a form they have no interest in. Offering a number of forms on mobile phones and collecting the opinions of real users may give more trustable answers to base the decision of which forms are too large, on. A general conclusion is certainly that profound tests in a scientific setting are required in order to really evaluate a design and find proof for hypotheses of the kind that are the basis of our work.

Cognitively impaired users may have completely different constraints regarding form size. Since the constraints of these users vary depending on their condition and how impaired they are by it, it will be impossible to find an absolute maximum size that all users can handle. People in the target groups of this investigation will manage to achieve a task if they get enough time and are in an environment that enable them to concentrate on achieving it, i.e. they are assumed to be able to fill-in a form with the size of the demonstrated one. A task is obviously easier to execute when it is small. Those who offer form services will have to consider which forms they should provide on mobile phones, and reflect upon how many users they want to accommodate. Offering only small-sized forms lets also a large number of cognitively impaired people use them, as well as make them easier to complete for people with other disabilities. Offering forms of bigger sizes lets a larger amount of people *without* cognitive or other impairments use forms, while the cognitively impaired may become irritated and feel helpless when they discover that they struggle with, or can not, complete them. Putting warning signs on large forms may segregate and stigmatise users, and discourage the people without impairments. However, providing forms with an indicator of their size and complexity without directing specific user groups seem to be the best alternative. Users may also get "braver" after trying the small forms first, and making success.

There must be a limited intensity of information
The evaluators responded positively to the possibility of choosing which information to see on a page with high information intensity. We therefore conclude that the technique used can be beneficial. However, how extensively the technique can be used should be investigated in further tests. On pages with small blocks of pure information, having to continuously reveal these blocks by pressing a button may be annoying. Statement 5.5.1 in Chapter 1.3 in Appendix A states that the number of keystrokes should be kept to a minimum. The use of the technique demonstrated in (Buyukkokten et al., 2002), where every element of the form needs to be "opened", seem to be too extensive. One of the evaluators who found the size of the form too large, suggested more use of the technique to make it seem smaller. Something between what was demonstrated in (Buyukkokten et al., 2002) and our approach may be better. Form elements that are closely related on a page can be grouped, e.g. on pages that present information about several kindergartens, each kindergarten could be a group. Allowing the users to hide and reveal all elements of each group may decrease the information intensity as well as make the form seem smaller.

Hiding and revealing information changes the form layout. This may be confusing to people with some types of cognitive disabilities. Our target groups should not be affected by this to any large extent. Form providers will need to consider how many users are able to use the forms if the information intensity is reduced, versus how many are able to use the forms if the layout changes.

The service must have a user interface that accommodates the device limitations
None of the users stated to dislike or be confused by the drop-down boxes, which present only one of several alternatives when they are not activated. Since, as mentioned earlier, most people are well accustomed to these controls from electronic forms, we think that drop-down boxes are a very suitable form control in forms on mobile phones. They can decrease the information intensity as well as help the user maintain the context when there is enough room for the previous and following form elements.
In our opinion, quantitative user tests of drop-down boxes versus lists showing all alternatives (or as many as the screen size allows) should be performed. These would show whether the cost of the cognition problem with seeing one alternative at first is higher than the cost of the cognition problem with high information intensity and loss of form context. We believe the latter to be most severe and most likely to occur. Users with some types of cognitive impairments may be harder affected by these issues. Again, form producers will need to consider which solution leads to most users.

The HCI-package of Tellu has an unusual feature, which moves elements below an expanded form control downwards, rather than drawing the expanded control over them. We thought that this feature could be particularily beneficial on mobile phones, since the users can see the content of form elements that are to be filled in immediately after the expanded one, helping them maintain the context. On the other hand, it could also be disturbing, as the content of the form suddenly moves downwards, and upwards when the form control is closed. None of the evaluators seemed to find this feature confusing or annoying, nor did they in any way react to it when they filled in the form. We therefore claimed to have an indication of that our concerns were in vain. Further tests would show whether maintaining the form context in this way is beneficial or not. The earlier discussion on users with some types of cognitive disabilities and layout changes will apply here, too.

The service's user interface should follow guidelines for design of forms
The investigation of whether users were confused by command changes or not, showed that most of the users did not notice them at all. This is probably so because they did not have any need for them at the time they changed, and also because several of them navigated with the keys for moving right and left. This was originally implemented as a feature for the experienced users. First when they could not find what they needed in the form page, they looked at the menu line. However, had the evaluations been executed on actual phones, their attention may have been drawn to changes happening within their field of vision. We think that always keeping the most important command on the right soft button and other commands in the menu on the left soft button is a good principle. This is yet another design feature which requires further tests, in order to measure whether the cost of confusion is small enough to be ignored because of measured benefits. Again, people with some types of cognitive disabilities may suffer greatly from anything that may cause confusion, and one needs to consider this when deciding to apply such a principle. Also for these users, any benefits of the principle may turn out to be valuable enough to upweigh the cost of confusion.

We expected that the users quickly learned that every page has a 'Next' command, except for the pages where the command seemed to disappear. However, this turned out not to be the case. Particularily the users who navigated with the keys for moving right and left, although not consequently, probably did not always observe the command available when they reached the bottom of a page, as there was no need to. In order to find out whether anyone thought the command had been removed entirely when the right soft button had a different command, we asked the evaluators what they thought had happened to it. Five of ten evaluators correctly expected it to be found in the menu, some of these used it when filling in the form. Of the remaining five, who all navigated with keys for moving right and left, three answered that they did not think there was such a command. Had the form not offered navigating with the keys for moving right and left, we expect that they eventually would have discovered that the command was not removed from the form page. Again, for the people with some types of cognitive disabilities, this may be a problem. However, if they check the alternatives, they will find the command. We therefore consider this to be a non-issue as long as any benefits of the principle of always keeping the most important command on the right soft button, upweigh the cost of confusion.

In addition to commands for choosing and cancelling choice of kindergartens on the right soft button, we implemented the possibility of choosing/cancelling choice of kindergartens with the fire button as a feature for more experienced users. Since so many used the fire button when choosing in this case, we recommend quantitative tests on whether 'Choose' commands can be left out of form interfaces on mobile phones. People are probably accustomed to use the fire button for choosing, as this is incorporated in many mobile phone user interfaces.

Since representing the form size and progress with portfolio tabs in the demonstrated manner was a success, we conclude that this solution fills its purpose. However, had the number of form pages been larger than eight, we would have had to make the tabs smaller or show as many as possible and somehow indicate how many pages the form has. Making the tabs smaller is probably not a good idea, as that would make the content of the tabs harder to perceive and the design less appealing. It is therefore necessary to find a good solution of indicating that a form has more pages than the number of tabs shown.

A service must accommodate variations in skills and abilities
Checking whether users with a cognitive style that makes them interpret graphics better than text benefited from the graphical modality was not really a success, as (1) we did not know whether any of the evaluators had this cognitive style, and (2) in the case of the map representation, most of the seven users who preferred it, may have been "blended" by the benefit of getting a geographical overview of the kindergartens' positions. Asking follow-up questions like whether they felt that the information in the map was easier to perceive than the list information, was resultless. That none of the users preferred the representation of points of time as clock icons was disappointing and a bit surprising. We expected at least a few to find them appealing. From this we conclude that presenting objects which have a location, geographically, is beneficial regarding the possibility of seeing where they are situated related to each other and one's home. It should be particularily beneficial on mobile phones, where going through a list of names and addresses on a small screen can be tedious. We also expect professionally made clock icons, combined with a technique suggested by one of the evaluators; to enlarge the current selected activity's clock icon, to make more users prefer the representation with clock icons. We recommend further tests in order to find out whether users benefit from these graphical representations.

When looking at the first assertion of the hypothesis – that specific techniques and form controls can make electronic forms feasible on mobile phones and relieve the burden of the users – our results indicate that we have suggested techniques and form controls that may fulfill this assertion. Nevertheless, claiming that the assertion is completely fulfilled depends on the outcome of further tests. Our suggested techniques and form controls are:

- The hiding/revealing information technique, called "Form summarisation" by (Buyukkokten et al., 2002), to decrease the information intensity.
- Drop-down boxes, to make a list with alternatives take less space, thereby helping the users to keep the form context.
- Moving form elements downwards when the form control above is expanded, to help the users keep the form context.
- Always keeping the most important command, for different situations, at the right soft button while other commands are kept in a menu at the left soft button, so the users can easily find and use the important command.
- Indicating form size and progress by using portfolio tabs with numbers on them, the tab with the number of the current form page should have a different colour than the others. The portfolio tabs should have a field, with the same colour, below them with the description of the content/name of the current form page. This tells the users how large the form is, where they are, and how much is left.
- Representing objects with a location geographically, so the users will not have to go through lists of names and addresses.

We also believe that our other suggestions which were not demonstrated and/or evaluated will fulfill this assertion.

The second assertion of our hypothesis, i.e. that offering visualisation of information as an additional modality in forms on mobile phones can be beneficial for the cognitive processing of content, was not completely supported by the empirical evauations. We cannot conclude from the evaluations that anyone perceived the graphical content better than the textual. Since we did not have access to people in the two target groups, we could not test whether the demonstrated solutions accommodated their constraints either. However, we can see *no reason* to why multimodal content is less beneficial on mobile phones than when presented on e.g desktop computers, and therefore firmly believe that the assertion is true. Moreover, users with moderate reading problems can see where the kindergartens are positioned. This may help them interpret the address text, rather than struggle with trying to separate between words and letters in a list of names and addresses. People with moderate number-reading problems will not have to struggle with perceiving numeric points of time in a list. People whose cognitive style is to interpret graphic representations better than text and numbers should also benefit. Further tests of the demonstrated multimodal content on people in the target groups, would hopefully give results that can support our assumption regarding the truthfulness of the assertion.

Very many state that multimodality is beneficial for everyone, and people with cognitive disabilities in particular. However, there are also some disadvantages, or "mine traps", with the use of multimodality. Some of these were listed in Chapter 6.3.
The first of these has been mentioned several times in this report, i.e. that poor quality media can be more confusing than helpful (WebAIM, 2007, Design Considerations). We have practically proved this to be true, with our clock icons.
The second one concerns that extensive use of graphics and animations may distract the user and increase the cognitive load (Mariger, 2006; Bartlett, 1999; Jiwnani, 2001). We are certain that this is no issue in the demonstrated form, as the only graphic presentations are the map, clock icons and the information icons on the two pages that use the hiding/revealing information technique.
The third trap appears in a situation in which the user has to attend to different modalities simultaneously. This may decrease the learning of the information (Reeves et al., 2004). In our case, this is , however, not an issue.
The fourth one which concerns users having to choose between modalities and get confused by it, is not an issue, as (1) there are only two modalities, and (2) users in our target groups have not other cognitive challenges than those related to number and text *reading*. During the walkthrough evaluations, we did not get any indications of, that the users were confused by having to choose between the two modalities. Users with cognitive disabilities which affect their concentration or ability to abstract thinking, are more likely to suffer from having to choose between a number of modalities. Probably, the worse the state of their condition is, the smaller is the number of modalities they can manage to choose between.

The assertion considered the perception of information, but the fact that people in the target groups will also struggle with entering input is not accommodated. The best solution we can think of, is using the microphone of the phone and make forms take audio as an additional modality, which was mentioned in Section 3.5.2 and Chapter 6.4, provided that speech-to-text converters give accurate results and that the time it takes to convert, including the upload/ download time, is not unreasonably long. An alternative solution, or combined with, is to offer the users to hear the information they have entered. If they have "written" information, they can hear the result of their input and try to correct it, over and over until it is correct. As the technology advances, speech-to-text converters become nearly 100% accurate, phones get larger storing space and the network transmission rates increase. The first solution or a combination of the two may become common. However, privacy regarding voice input and using the mobile phone to fill in forms anywhere, anytime will still be an issue. Who would want to state sensitive information about themselves on e.g. a train?

The research assignment stated that "fundamental ideas behind Universal Design must be incorporated in the solution". We believe to have done this in several ways. The philosophy of considering "as many as possible" have been followed, which in our case was to have a strict focus on the two target groups. Four of the principles of Universal Design were followed. We chose several other principles and guidelines which accommodate the needs of the target groups, people with cognitive disabilities and people with other disabilities, as described in Section 7.2.1.

The dilemma of deciding who "as many as possible" are, may be hard for form service providers who want to design universally. For instance, as we described earlier, accommodating people with some types of cognitive disabilities may imply to decrease the number of forms to offer on mobile phones, to only the ones with a small size. This would accommodate the needs of more people, but the number of forms represented on mobile phones would not be large. Solutions for accommodating more people will also have a higher cost. For instance, multimodal forms will cost more to make than forms with only one modality. However, as mentioned in Chapter 6.3, as the number of elderly people increases, form providers may have to accommodate them if they are to make a profit.

To conclude, we claim to have completed the research assignment. The first main result is, of course, the investigation itself, including the prototype and the discussions concerning multimodality. There is no doubt about that there are both "pros" and "cons" connected to almost any design decision. It is also obvious that all guidelines allow many different design decisions which often may go in opposite directions. What is required from the designer is a good proportion of common sense, and a profound knowledge of the general aims and ambitions of Universal Design. Finally, as one goes concrete, and real designs and functional prototypes are being shaped, new challenges appear.

The second main result is exactly the need for more knowledge. We have suggested a number of efforts for further investigations. Tests should be performed on actual phones with different screen sizes and resolutions. We suggested several criteria for the feasibility of forms on mobile phones, and ways to achieve some of them. More detailed design and usability criteria need to be found. Regarding people with cognitive disabilities, research on their interaction with mobile phones can provide useful information for form service providers and developers who want their forms to accommodate the needs and preferences of as many as possible. In other words, detailed, scientific studies are called for in order to accumulate the knowledge base of universally designed user interfaces on mobile phones.

List of Figures

List of Tables

References

Adobe, 2007
Adobe Systems Incorporated (2007): *Adobe*. http://www.adobe.com/, accessed 12 February, 2007.

Altinn, 2007
Altinn: *Altinn*, http://www.altinn.no/. accessed 13 February, 2007.

ARTS, 2006
Department of Telematics, NTNU (2006): *ARTS - Arena for Research on Advanced Telecom Services*, Program for Advanced Telecom Services.
http://www.item.ntnu.no/avantel/projects/ARTS/arts.html, accessed 17 November, 2006.

Bartlett, 1999
Bartlett, Kynn: *Designing Pages Accessible to Limited Textual Comprehension Users*, AWARE Center, HTML Write, 1999. http://aware.hwg.org/tips/ltcu.html, accessed 12 March, 2007.

Berry, 2000
Berry, Dick: *The User Experience, Part 2*, IBM developerWorks, Web Development, 01 Dec 2000. http://www-128.ibm.com/developerworks/library/w-berry3/, accessed 23 January, 2007.

Bohman, 2005
Bohman, Paul, R.: *Cognitive Disabilities Part 1: We Still Know Too Little, and We Do Even Less*, WebAIM: Web Accessibility in Mind, Articles.
http://www.webaim.org/articles/cognitive/cognitive_too_little/, accessed 9 February, 2007.

Bollaert, 2002
Bollaert, Jodi: *Using Web Widgets Wisely, Part 1*, *Common Usability Problems and Solutions,* IBM developerWorks, Web Architecture, 01 Jun 2002. http://www-128.ibm.com/developerworks/library/us-widget/, accessed 23 January, 2007.

Brynn et al., 2006
Brynn, R., Bergerud Buene, T. – Deltasenteret: *Universell utforming og tilgjengelighet – politikk og lovgivning i inn- og utland*, Sosial- og helsedirektoratet, januar 2006.

Bræk et al., 2002
Bræk, R., Husa, K., E., Melby, G.: *ServiceFrame Whitepaper*, Ericsson NorARC, May 2002.

Burkhart, 2003
Burkhart, Linda J.: *Multi-Modality Supports for Children on the Autism Spectrum*, In online Proceedings of Technology and Persons with Disabilities Conference, Los Angeles, CA, 2003. http://www.csun.edu/cod/conf/2003/proceedings/csun03.htm, accessed 8 March, 2007.

Buyukkokten et al., 2002
Buyukkokten, Orkut; Kaljuvee, Oliver; Garcia-Molina, Hector; Paepcke, Andreas; Winograd, Terry: *Efficient Web Browsing on Handheld Devices Using Page and Form Summarization*, ACM Transactions on Information Systems, Vol.2, No. 1, page 82-115, January 2002.

Cambridge University Press, 2007
Cambridge University Press (2007): *Cambridge Dictionaries Online*, Cambridge Advanced Learner's Dictionary. http://dictionary.cambridge.org/, accessed 31 January, 2007.

Chisholm et al., 1999
Chisholm, Wendy; Vanderheiden, Gregg; Jacobs, Ian (Eds.): *Web Content Accessibility Guidelines 1.0*, WC3 Recommendation 5-May-1999. http://www.w3.org/TR/WAI-WEBCONTENT/, accessed 21 February, 2007.

DIADEM, 2007
European Union Project DIADEM (2006-7): *DIADEM, Delivering Inclusive Access for Disabled or Elderly Members of the community*. http://www.project-diadem.eu/, accessed 8 May, 2007.

Doolittle, 2001
Doolittle, Peter E.: *Multimedia Learning: Empirical Results and Practical Applications*, 2001. http://www.ipfw.edu/as/tohe/2001/Papers/doo.htm, accessed 5 March, 2007.

ELMER 2, 2006
Norwegian Ministry of Trade and Industry: *Simplified Forms for the Private Sector: ELMER 2*, *User Interface Guidelines for Governmental Forms on the Internet*, October 2006. http://www.elmer.no/retningslinjer/pdf/elmer2-english.pdf, accessed 3 February, 2007.

Encyclopedia of Mental Disorders, 2007
Encyclopedia of Mental Disorders: *Mathematics Disorder*, Tish Davidson, A.M. http://health.enotes.com/mental-disorders-encyclopedia, accessed 5 March, 2007.

Encyclopedia of Neurological Disorders, 2007
Encyclopedia of Neurological Disorders: *Dyslexia*, Laith Farid Gulli, MD; Nicole Mallory, MS, PA−C; Robert Ramirez, DO. http://health.enotes.com/neurological-disorders-encyclopedia, accessed 5 March, 2007.

ETSI, 2003
ETSI: *Human Factors (HF); Multimodal interaction, communication and navigation guidelines*, ETSI EG 202 191 V1.1.1 (2003-08).

Eurostat, 2003
Eurostat: Statistics in focus, Population and social conditions, Theme 3 – 26/2003, Employment of disabled people in Europe in 2002, table 2, *Distribution (%) of population with long-standing health problem or disability (LSHPD)*.
http://epp.eurostat.ec.europa.eu/portal/page?_pageid=1073,46587259&_dad=portal&_schema=PORTAL&p_product_code=KS-NK-03-026, accessed 1 March, 2007.

Fichter, 2003
Fichter, Darlene: *Designing Web Forms Using Boxes and Arrows*, ONLINE Magazine, Vol. 27, No. 2, page 59-61, Mar/Apr 2003.

Flaten, 2006
Flaten, Ragnhild: *Designing Inclusive Mobile Services*, Project report in TDT4730 Information Systems, Depth Study, Autumn 2006.

Flick, 2006
Flick, Uwe: *An Introduction to Qualitative Research*, 3rd Edition, Sage Publications Ltd, London, 2006.

Gehani, 1982
Gehani, Narain H.: *The Potential of Forms in Office Automation*, IEEE Transactions on Communications, Vol. Com-30, No. 1, page 120-125, January 1982.

Gong et al., 2004
Gong, Jun, Tarasewich, Peter: *Guidelines for Handheld Device Interface Design*, In Proceedings of the 35th Annual Meeting of the Decision Science Institute, Boston, Massachussets, page 82-88, 2004.

Gouscos et al., 2002
Gouscos, Dimitris; Rouvas, Stathis; Vassilakis, Costas; Georgiadis, Panagiotis: *An Object-Oriented Approach for Designing Administrative E-forms and Transactional E-services*, In Proceedings of the 8th International Conference on Object-Oriented Information Systems, Montpellier, France, Lecture Notes in Computer Science Vol. 2425, page19-30, 2002.

Haraldsen, 2004
Haraldsen, Gustav: *Råd for utvikling og utforming av webskjema,* Versjon 1.1, Håndbok 81, Statistics Norway, 2004.

Jiwnani, 2001
Jiwnani, Kanta: *Designing for Users with Cognitive Disabilities*, 2001. http://www.otal.umd.edu/uupractice/cognition/, accessed 12 March, 2007.

Kolatch, 2000
Kolatch, Erica: *Designing for Users with Cognitive Disabilities*, Department of Computer Science, University of Maryland, College Park, MD 20742 USA, April 2000. http://www.otal.umd.edu/UUGuide/, accessed 2 March, 2007.

Mariger, 2006
Mariger, Heather: *Cognitive Disabilities and the Web: Where Accessibility and Usability Meet?,* 2006. http://ncdae.org/tools/cognitive/, accessed 12 March, 2007.

Mayer, 2001
Mayer, Richard E.: *Multimedia Learning*, Cambridge, UK, Cambridge University Press, 2001.

Mayer et al., 1998
Mayer, R. E.; Moreno, R.: *A Cognitive Theory of Multimedia Learning: Implications for Design Principles*, Paper presented at the CHI-98 Workshop on Hyped-Media to Hyper-Media, Los Angeles, CA, 1998.

McGrenere et al., 2006
McGrenere, Joanna; Sullivan, Jim; Baecker, Ronald M.: *Designing Technology for People with Cognitive Disabilities*, CHI-2006 Workshop, April 22-27, Montreal, Canada. http://www.cs.ubc.ca/~joanna/CHI2006Workshop_CognitiveTechnologies/ExtendedAbstract.pdf, accessed 2 March, 2007.

Miller et al., 2001
Miller, Sarah; Jarrett, Caroline: *Should I use a drop-down? Four steps for choosing form elements on the Web*, Effortmark Ltd 2001. http://www.formsthatwork.com/, accessed 26 January, 2007.

More, 2007
More AS (2007): *Barnehagesøknad 2007*, Herøy kommune. https://www.more.no/more/wizard/wizard.jsp?wizardid=702, accessed 24 April, 2007.

Mousavi et al., 1995
Mousavi, S. Y.; Low, R.; Sweller, J.: *Reducing cognitive load by mixing auditory and visual presentation modes*, Journal of Educational Psychology, Vol. 87, No. 2, page 319-334, 1995.

Murphy, 2005
Murphy, Steve: *Accessibility of Graphics in Technical Documentation for the Cognitive and Visually Impaired*, Proceedings of the 23rd International Conference on Design of Communication, page 12-17, September 21-23, 2005.

Newell et al., 2002
Newell, A. F.; Carmichael, A.; Gregor, P.; Alm, N.: *Information technology for cognitive support*, The Human-Computer Interaction Handbook: Fundamentals, Evolving technologies and Emerging Applications, Lawrence Erlbaum Publishers, page 464-481, 2002.

Nielsen, 1994
Nielsen, Jakob: *Enhancing the Explanatory Power of Usability Heuristics*, Proceedings of Conference on Human Factors in Computing Systems, ACM Press, page 152-158, 1994.

Nielsen, 2000
Nielsen, Jakob: *Reset and Cancel Buttons*, Jakob Nielsen's Alertbox, April 16, 2000. http://www.useit.com/alertbox/20000416.html, accessed 23 January, 2007.

Nielsen, 2007
Nielsen, Jakob: *How to Conduct a Heuristic Evaluation*. http://www.useit.com/papers/heuristic/heuristic_evaluation.html, accessed 6 June, 2007.

Nikkanen, 2003
Nikkanen, Mikko: *One-handed use as a design driver: enabling efficient multi-channel delivery of mobile applications*, In Crestani, F., Dunlop, M. & Mizzaro, S. (Eds.), Mobile and Ubiquitous Information Access: Mobile HCI 2003 International Workshop, Udine, Italy, September 8, 2003, Revised and Invited Papers, Lecture Notes in Computer Science Vol. 2954, Springer, page 28-41, 2003.

OECD, 2005
Organisation for Economic Co-operation and Development, Statistics Canada: *Learning a Living*, First Results of the Adult Literacy and Life Skills Survey, Ottawa and Paris: Statistics Canada and OECD, 2005.

Official Journal of the European Communities, 2000
Official Journal of the European Communities: *The Charter of Fundamental Rights of the European Union*, (2000/C 364/01), 18 December, 2000.

Oviatt, 1997
Oviatt, Sharon L.: *Multimodal interactive maps: Designing for human performance*, Human Computer Interaction, Vol. 12, No. 1 & 2, page 93-129, 1997

Preece et al., 2002
Preece, J.; Sharp, H.; Rogers, Y.: *Interaction Design: Beyond Human-Computer Interaction*, John Wiley & Sons, Ltd, New York, 2002.

Rabin et al., 2006
Rabin, Jo; McCathieNevile, Charles (Eds.): *Mobile Web Best Practices 1.0, Basic Guidelines*, WC3 Proposed Recommendation 2 November 2006. http://www.w3.org/TR/mobile-bp/, accessed 21 February, 2007.

Reeves et al., 2004
Reeves, Leah M.; Lai, Jennifer; Larson, James A.; Oviatt, Sharon; Balaji, T. S.; Buisine, Stéphanie; Collings, Penny; Cohen, Phil; Kraal, Ben; Martin, Jean-Claude; McTear, Michael; Raman, TV; Stanney, Kay M.; Su, Hui; Wang, Qian Ying: *Guidelines for Multimodal User Interface Design*, Communications of the ACM, Vol. 47, No. 1, page 57-59, January 2004.

Rettig, 1994
Rettig, Marc: Prototyping for Tiny Fingers, Communications of the ACM, Vol. 37, No. 4, page 21-27, April 1994.

Rosenbaum et al., 2000
Rosenbaum, S.; Rohn, J.; Humburg, J.: *A toolkit for strategic usability; Results from workshops, panels and surveys,* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, The Hague, Netherlands, page 337-344, April 01-06, 2000.

Ross, 2000
Ross, Alistair: *Screen-based Information Design*, Aslib Proceedings Vol. 52, No. 9, page 331-342, October 2000.

Rudd et al., 1996
Rudd, Jim; Stern, Ken; Isensee, Scott: *Low vs. high-fidelity prototyping debate*, Interactions, Vol. 3, No. 1, page 76-85, January 1996.

SAS Braathens, 2007
SAS Braathens (2007): *SAS Braathens*. http://www.sasbraathens.no/, accessed 20 February, 2007.

Schwartz et al., 1999
Schwartz, James E.; Beichner, Robert J.: *Essentials of Educational Technology*, Allyn & Bacon, Boston, 1999.

Seeman, 2002
Seeman, Lisa: *Inclusion of Cognitive Disabilities in the Web Accessibility Movement*. In Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii, USA, 2002. http://www2002.org/CDROM/alternate/689/, accessed 26 February, 2007.

Shneiderman, 1998
Shneiderman, Ben: *Designing the User Interface*, 3rd Edition, Addison Wesley, Boston, MA, 1998.

Snyder, 2003
Snyder, Carolyn A.: *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, Morgan Kaufmann, San Fransisco, 2003.

Statistics Norway, 2006, electronic
Statistics Norway (2006): *Andelen som hadde tilgang til ulike elektroniske tilbud i hjemmet.* 1994-2005. Prosent. http://www.ssb.no/samfunnsspeilet/utg/200602/04/tab-2006-05-18-04.html, accessed 20 November, 2006.

Statistics Norway, 2006, lifetime
Statistics Norway (2006): Table 05375, *Expectation of lifetime, by sex and ages*. http://statbank.ssb.no/statistikkbanken/, accessed 31 January, 2007.

Statistics Norway, 2006, StatBank
Statistics Norway, StatBank Norway (2006): Table 04226, *Level of functioning, by sex and age (per cent)*. http://statbank.ssb.no/statistikkbanken/, accessed 28 February, 2006.

Story et al., 1998
Story, Molly Follette; Mueller, James L.; Mace, Ronald L.: *The Universal Design File, Designing for People of All Ages and Abilities*, The Center for Universal Design, NC State University, 1998. http://www.design.ncsu.edu/cud/pubs_p/pudfiletoc.htm, accessed 21 February, 2007.

Sun Microsystems, 2006
Sun Microsystems, Inc (2006): *The Java ME Platform — the Most Ubiquitous Application Platform for Mobile Devices*. http://java.sun.com/javame/index.jsp, accessed 28 November, 2006.

Svanæs, 1999
Svanæs, Dag: *Understanding Interactivity, Steps to a Phenomenology of Human-Computer Interaction*, 1999. http://www.idi.ntnu.no/~dags/interactivity.pdf, accessed 8 May, 2007.

Telematics Applications Programme, 1998
Telematics Applications Programme: *Design for All and ICT Business Practice: Addressing the Barriers; Examples of Best Practice*, Disabled and Elderly sector, Seminar on "Design-for-All" for an Inclusive Information Society, Brussels, 8 December, 1998.

Telenor, 2007
Telenor ASA (2007): *Mer om UMTS og EDGE*, Tjenester, 3G. http://telenormobil.no/tjenester/3g/merom.do, accessed 22 April, 2007.

The British Dyslexia Association, 2007
The British Dyslexia Association (2007): *What is Dyslexia?*, All About Dyslexia. http://www.bdadyslexia.org.uk/whatisdyslexia.html, accessed 5 March, 2007.

The British Dyslexia Association, 2007, Style Guide
The British Dyslexia Association (2007): *Dyslexia Style Guide*, About Dyslexia. http://www.bdadyslexia.org.uk/extra352.html, accessed 19 April, 2007.

The Center for Universal Design, 1997
Bettye Rose Connell, Mike Jones, Ron Mace, Jim Mueller, Abir Mullick, Elaine Ostroff, Jon Sanford, Ed Steinfeld, Molly Story, & Gregg Vanderheiden – The Center for Universal Design (1997): *The Principles of Universal Design*, Version 2.0. Raleigh, NC: North Carolina State University. http://www.design.ncsu.edu/cud/about_ud/udprinciples.htm, accessed 20 February, 2007.

The Center for Universal Design, 2007
The Center for Universal Design (2007): *About Universal Design*, North Carolina State University. http://www.design.ncsu.edu/cud/about_ud/about_ud.htm, accessed 19 February, 2007.

The ELMER Project, 2007
ELMER: *ELMER - Simplified Forms for the Private Sector*. http://www.elmer.no/english/, accessed 27 February, 2007.

The Municipality of Bergen, 2007
Bergen kommune (2007): *Søk barnehageplass*, Alt om barnehage. http://www.bergen.kommune.no/portal/page/portal/bkinternet/barnehagesok?id=T863887&m2_mode=&re1_mode=venstre&m3_mode=spot1&sectionId=289, accessed 24 April, 2007.

The Municipality of Trondheim, 2007
Trondheim kommune (2007): *Trondheim kommune*, Selvbetjening, Skjema. http://www.trondheim.kommune.no/, accessed during February, 2007.

The Norwegian Dyslexia Association, 2007
Dysleksiforbundet i Norge (2007): *30 spørsmål om dysleksi*, Om Dysleksi. http://www.dysleksiforbundet.no/--/?/=19&//=49&ID=&C=, accessed 5 March, 2007.

The Norwegian Dyslexia Association, 2007, Remedies
Dysleksiforbundet i Norge (2007): *Hjelpemidler*. http://www.dysleksiden.no/hjelpemidler.html, accessed 19 April, 2007.

The Norwegian Ministry of Government Administration and Reform, 2006
Det kongelege fornyings- og administrasjonsdepartement: *St.meld. nr. 17 (2006-2007), Eit informasjonssamfunn for alle.* http://www.regjeringen.no/Rpub/STM/20062007/017/PDFS/STM200620070017000DDDPDFS.pdf, accessed 8 May, 2007.

The Norwegian Ministries of Labour and Social Inclusion and of the Environment, 2004
Arbeids- og sosialdepartementet, Miljøverndepartementet: *Regjeringens handlingsplan for økt tilgjengelighet for personer med nedsatt funksjonsevne, Plan for universell utforming innen viktige samfunsområder*, 2004. http://www.universell-utforming.miljo.no/file_upload/hptuu.pdf, accessed 8 May, 2007.

The Norwegian Ministry of Modernisation, 2005
Moderniseringsdepartementet: *eNorge 2009 – det digitale spranget*, 2005. http://www.regjeringen.no/upload/FAD/Vedlegg/IKT-politikk/enorge_2009_komplett.pdf. accessed 8 May, 2007.

Office of the United Nations High Commissioner for Human Rights, 1975
Office of the United Nations High Commissioner for Human Rights: *Declaration on the Rights of Disabled Persons*, Proclaimed by General Assembly resolution 3447 (XXX) of 9 December 1975. http://www.ohchr.org/english/law/res3447.htm, accessed 1 March, 2007.

Vanderheidens, 1991
Vanderheiden, G., Vanderheiden, K., R.: *Accessible Design of Consumer Products: Guidelines for the Design of Consumer Products to Increase Their Accessibility to People with Disabilities or Who Are Aging*, 1991, Trace R & D Center, University of Wisconsin – Madison, http://trace.wisc.edu/docs/consumer_product_guidelines/consumer.htm, accessed 28 February, 2007.

Vassilakis et al., 2005
Vassilakis, Costas; Lepouras, George; Rouvas, Stathis; Georgiadis, Panagiotis: *Exploiting Form Semantics and Validation Checks to Improve E-form Layout*, International Journal of Web Engineering and Technology, Vol. 2, No. 1, page 81-96, 2005.

WebAIM, 2007, Cognitive Disabilities
WebAIM (1999-2007): *Cognitive Disabilities*. http://www.webaim.org/articles/cognitive/, accessed 2 March, 2007.

WebAIM, 2007, Conceptualizing Design Considerations
WebAIM(1999-2007): *Cognitive Disabilities Part 2: Conceptualizing Design Considerations*. http://webaim.org/articles/cognitive/conceptualize/, accessed 5 March, 2007.

WebAIM, 2007, Design Considerations
WebAIM (1999-2007): *Cognitive Disabilities, Design Considerations*. http://www.webaim.org/articles/cognitive/design.php, accessed 8 March, 2007.

Wickens, 2002
Wickens, Cristopher D.: *Multiple resources and performance prediction*, Theoretical Issues in Ergonomics Science, Vol.3, No.2, page159-177, 2002.

Wikipedia, 2007
Wikipedia, The Free Encyclopedia (2007): Sans-serif. http://en.wikipedia.ord/wiki/Sans-serif, accessed 22 May, 2007.

Appendix A: Design Principles and Guidelines

1 Design of User Interfaces on Mobile Devices

Guidelines for designing user interfaces on mobile devices are yet scarce, compared to the amount of guidelines for the Web and user interfaces on computers. This chapter renders some guidelines for design of user interfaces on mobile devices. Some of these are also directed mostly towards Web content, and some are somewhat outdated considering the mobile phone technology progress.

1.1 Guidelines for Handheld Mobile Device Interface Design

The guidelines of Jun Gong and Peter Tarasewich are based on Shneiderman's eight golden rules (Shneiderman, 1998), with some additional guidelines (Gong et al., 2004). They claim that the following four of Shneiderman's rules apply directly to mobile devices:
Supplementary comments by Shneiderman.

1 Enable Frequent Users to Use Shortcuts
As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

2 Offer Informative Feedback
For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

3 Design Dialogs to Yield Closure
Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment.

4 Support Internal Locus of Control
Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

The four remaining rules have been changed to be applicable on mobile devices:
Supplementary comments by Gong/Tarasewich.

5 Consistency
Consistency takes on an additional dimension with mobile applications: the consistency across multiple platforms and devices for the same application. Users of mobile devices may need to switch between their desktop machines and different mobile devices frequently.

- The "look and feel" should be the same across multiple platforms and devices.

- Elements of mobile interfaces such as names, color schemes, and dialog appearances should be the same as their desktop counterpart.
- Create input/output methodologies that are device independent - avoid using methods specific to mobile platforms where possible.

6 Reversal of actions
Allowing easy reversal of actions may be more difficult for mobile devices because of a lack of available resources and computing power.

- Mobile applications should rely on network connectivity as little as possible.

7 Error prevention and simple error handling
Preventing and handling errors on mobile interfaces are similar to those for desktop interfaces, although the need becomes more critical due to the more rapid pace of events in the mobile environment. Error prevention also needs to take the physical design of mobile devices into account.

- Nothing potentially harmful should be triggered by too simple an operation (e.g., power on/off).

8 Reduce short-term memory load
Given the limitations of a user's short-term memory, interfaces should be designed such that very little memorization is required during the performance of tasks

- Rely on recognition of function choices instead of memorization of commands.
- Use modalities such as sound to convey information where appropriate.

They have proposed seven additional principles focusing on the interface design:
Supplementary comments by Gong/Tarasewich.

9 Design for multiple and dynamic contexts
The contexts of computer applications used in the office, home, or similar settings are relatively stable. On the other hand, with mobile applications, there can be a significant number of additional people, objects, and activities vying for a user's attention aside from the application or computer itself. Environmental conditions (e.g., brightness, noise levels, weather) can change depending on location, time of day, and season. The usability or appropriateness of an application can change based on these different context factors.

- Allow users to configure output to their needs and preferences (e.g., text size, brightness).
- Allow for single- or no-handed operation.
- Have the application adapt itself automatically to the user's current environment.

10 Design for small devices
As technology continues to advance, mobile platforms will continue to shrink in size and include items such as bracelets, rings, earrings, buttons, and key chains. New or modified interaction techniques may be necessary to overcome the physical limitations. Speech input is a viable alternative for devices too small for buttons. Sound can also be used for output, taking the place of text or graphics.

- Provide word selection instead of requiring text input.

11 Design for limited and split attention
Users of mobile devices often need to focus on more than one task. A mobile application may not be the focal point of the user's current activities. Mobile devices that demand too much

attention may distract users from more important tasks. Interfaces for mobile devices need to be designed to require as little attention as possible.

- Provide sound and tactile output options.

12 Design for speed and recovery
For mobile devices and applications, time constraints need to be taken into account in initial application availability and recovery speed. When time is critical, waiting a few minutes for an application to start may not be in the user's best interest. Given the different contexts under which mobile devices are used, users may need to quickly change or access functions or applications. When such situations rise, a user would need to quickly and securely save any work already performed and resume it later without any loss.

- Allow applications to be stopped, started, and resumed with little or no effort.
- Application should be up and running quickly.

13 Design for "top-down" interaction
Mobile devices with small screens have limitations on the amount of information that they can present at one time. Reading large amounts of information from such devices can require large amounts of scrolling and focused concentration.

- Present high levels of information and let users decide whether or not to retrieve details.

14 Allow for personalization
Different users have different usage patterns, preferences, and skill levels.

- Provide users the ability to change settings to their needs or liking.

15 Design for enjoyment
While functionality and usability are keys to mobile application success, other factors are also influential. Aesthetics is also part of designing an overall enjoyable user experience with mobile devices. … If functionality and usability are equal, an application or device will stand out if it is attractive in some way.

- Applications should be visually pleasing and fun as well as usable.

1.2 Guidelines for Mobile Application and Service Design

Mikko Nikkanen at Nokia Enterprise Solutions presents in the article "One-handed use as a design driver: enabling efficient multi-channel delivery of mobile applications" a collection of guidelines acquired from multiple sources. Some of these guidelines were originally intended for a specific technology, such as search engines, but may also be suited for applications and services. The guidelines are divided into three types; general, content and navigation.

General Design Guidelines for Mobile Devices (Nikkanen, 2003):
The supplementary comments are Nikkanen's.

G-1 Design for users on the go
The design for mobile devices must include context and forgiveness, and provide time-critical information.

G-2 Enable fast use
Two major considerations for the users of a mobile service are the cost of access and the speed of downloading content. Many users are paying for mobile services by the minute, so if

they cannot get the information they are looking for within a short period of time they will stop using the service.

G-3 Keep it simple
The old adages about keeping a system simple stupid and about “less being more” certainly apply for mobile devices and services. For instance, the most successful PDA devices do not attempt to replace the PC, but to complement the PC use, and the use of some other traditional tools.

G-4 Provide feedback and navigation cues
It should be obvious what the application is, and how one can navigate from the page.

G-5 Include self-recovering capabilities
Even if the network goes down, the service or application need not. There should be means to restore the values or written text, or to have them restored automatically.

Content Design Guidelines for Mobile Devices (Nikkanen, 2003):
The supplementary comments are Nikkanen’s.

C-1 Present the most important content first
The most important content should appear at the top of the page.

C-2 Keep content compact
It is recommended to keep the pages short.

C-3 Don’t make the page layout complicated
It is recommended to keep pages simple and task-oriented, possibly text only, and to avoid elements that don’t add direct value to the content.

C-4 Use simple text elements and styles
The elements used in text layout should be clear and simple.

C-5 Pay attention to page titles
It is important that the page title elements are descriptive, since they enable bookmarking and knowing where one is. The titles should however be short, preferably less than 15 characters.

C-6 Keep documents small
Because there are various memory restrictions in mobile devices, the documents should be kept as small as possible.

C-7 Use compact link names
Long linked text can make a page difficult to read and time consuming to scroll. It is recommended to use only one or two words as the title of the link.

C-8 Design clear forms
Forms should not be too long. A clear way to cancel the form filling and for going back should be provided, but attention should be paid to form resets, since on small devices, forms are laborious to refill if all values are reset by accident.

C-9 Use smart graphics
If graphics are used at all on small devices, they should be made informative, small and simple.

Navigation Design Guidelines for Mobile Devices (Nikkanen, 2003):
The supplementary comments are Nikkanen's.

N-1 Minimize steps in navigation
With small screen devices, it is very important to design for economy of navigation. Users will be frustrated by scrolling through long lists of options, filling out complex search forms, and seeing needless pages along the navigation path.

N-2 Selecting instead of typing
It is recommended to consider whether it is possible to ask the user to choose from a default list using select lists, check boxes or radio buttons rather than typing in a selection. Alternatively one can offer a default list together with an input box.

N-3 Keep the navigation consistent throughout the service
The way in which a user makes his or her way through the pages that constitute a service, interacting via links, menus and data input should be kept consistent throughout the service.

N-4 Design flat menus
It is recommended to keep menus flat, because it is often difficult to form an overview of a service containing too many layers, and because a deep hierarchy makes the use more difficult.

N-5 Cross link
The Back functionality is the most important way to go back. However, when users need to go back several levels, links to the starting page and subsection main pages are useful. A simple tree design is efficient, but the deeper the navigational hierarchy gets, the more necessary it becomes to get back to the starting point, and also to other pages.

N-6 Provide confirmations for important actions
Confirmations must be there for actions like changing important values or deleting items. Even though the user needs to click OK on the confirmation page, that requires much less effort than e.g. returning to a list to check if an item was really removed.

N-7 Searching should be intuitive
Searching should be a step-by-step, logical process. Once the search is performed, the results must be easy to scan, and the information should enable making good, informed choices within the results.

1.3 Mobile Web Best Practices 1.0

The Best Practices contains basic guidelines on how to deliver Web content to mobile devices in a way that improves the user experience (Rabin et al., 2006). The Mobile Web Best Practices 1.0 document is currently a proposed W3C recommendation. The Best Practices concern many aspects relating to delivery of Web content, this section will only render the statements and some of their explanations concerning usability of interaction with user interfaces on mobile devices, and which are not relevant only to Web browsers. See (Rabin et al., 2006) for all statements, full explanations, descriptions of how to implement them, and much more.

Best Practices (Rabin et al., 2006):

The statements and parts of their explanations are directly rendered from (Rabin et al., 2006).

Navigation and Links

ID	Statement	Explanation
5.2.2	Provide only minimal navigation at the top of the page.	Provide basic navigation, which should be placed on the top of the page. Any other secondary navigational element may be placed at the bottom of the page if really needed. It is important the users should be able to see page content once the page has loaded without scrolling.
5.2.4	Provide consistent navigation mechanisms.	Using the same navigation mechanisms across a service helps users orient themselves and allows them to identify navigation mechanisms more easily.
5.2.5	Assign access keys to links in navigational menus and frequently accessed functionality.	Where there is no pointing device, assigning an access key (keyboard short cut) to a link can provide a convenient way for users to access the link and avoid navigating to the link by repeated pressing of the navigation key. Provide the same access key for links that are repeated across pages such as links to the home page.
5.2.8	Do not cause pop-ups or other windows to appear and do not change the current window without informing the user. Do not create periodically auto-refreshing pages, unless you have informed the user and provided a means of stopping it.	Each of these activities is likely to cause the user confusion, or add cost and delay to their interaction. Some mobile devices use a separate window for input; this section does not refer to such windows. Auto-refreshing pages are widely recognized as presenting accessibility problems. In a mobile environment they may expose the user to undue cost as a result of such a page being left open or put unnoticed into the background. If an auto-refreshing page is demanded by the application, always provide a means of ceasing the refresh and always inform the user that the page will refresh and may expose them to higher usage costs.

Table A1. Navigation and links statements (Rabin et al., 2006).

Page Layout and Content

ID	Statement	Explanation
5.3.1	Ensure that content is suitable for use in a mobile context. Use clear and simple language.	Users in a mobile context are often looking for specific pieces of information, rather than browsing. Content providers should consider the likely context of use of information and, while providing the option to access all information, should offer appropriate information first. The general prescription to use clear language is of particular importance for mobile delivery, where brevity and directness are generally more desirable than a discursive style.

5.3.3	Limit scrolling to one direction, unless secondary scrolling cannot be avoided.	The page should lay out so that simple repeated scrolling in the same direction (axis) allows the user to experience all its content. However some content (such as maps and other images) cannot be displayed without secondary scrolling.
5.3.4	Ensure that material that is central to the meaning of the page precedes material that is not.	Because it is important for the user to gain an idea of the content of the page on initial view, there should be a minimum amount of clutter preceding this - including navigation, decorative images, advertising and other material that is not central to the user's experience of the page. The user should not have to scroll significantly to find the primary content of the page.
5.3.6	Ensure that information conveyed with color is also available without color. Ensure that foreground and background color combinations provide sufficient contrast.	Mobile devices often do not have good color contrast and are often used in less-than-ideal lighting conditions. Hence information highlighted in color may not be visible to users. If color is used to indicate a feature then that feature should generally also be indicated in a way that is not color dependent. In particular, do not use blue or purple text, as this may be confused with hyperlinks, especially on devices that do not underline links.
5.3.7	When using background images make sure that content remains readable on the device.	Images that are used indiscriminately can lead to content that is hard to view, particularly with the limited contrast often found on mobile devices and in the hostile viewing conditions in which mobile devices are frequently used.

Table A2. Page layout and content statements (Rabin et al., 2006).

Page Definition

ID	Statement	Explanation
5.4.1	Provide a short but descriptive page title.	Provide a descriptive title for the page to allow easy identification. Keep the title short to reduce page weight, and bear in mind that it may be truncated.
5.4.4	Do not use tables unless the device is known to support them. Do not use nested tables. Do not use tables for layout.	Tables do not work well on limited size screens and may result in the user having to scroll horizontally to read them. Putting navigational links into tables may result in the user having both to scroll horizontally and vertically to see possible navigational choices.
5.4.13	Provide informative error messages and a means of navigating away from an error message back to useful information.	…where errors are within the control of the content provider the user should be provided with clear information regarding the fault they have experienced. This should help them to understand whether the fault was temporary or permanent, whether they should retry the attempt to access the content and how they may be able to escalate the problem. It should also be possible for the user to escape from the error condition. They should either be able to return to the page they were on prior to the error, or to be able to move onwards to a convenient part of the service from where

		they can retry or alter the transaction they were attempting.

Table A3. Page definition statements (Rabin et al., 2006).

User Input

ID	Statement	Explanation
5.5.1	Keep the number of keystrokes to a minimum. Avoid free text entry where possible. Provide pre-selected default values where possible. Specify a default text entry mode, language and/or input format, if the device is known to support it.	Given the typical input limitations of a mobile device, the interface must as far as possible minimize user input. Where possible, use selection lists, radio buttons and other controls that do not require typing.
5.5.2	Create a logical order through links, form controls and objects.	It is important that as the user navigates through the page the various fields and objects are presented in a logical order, especially as many of them will not be visible at the same time as the focus item.

Table A4. User input statements (Rabin et al., 2006).

2 Designing Form User Interfaces

This chapter provides some principles, guidelines and advices on how to design forms. Several of these were made considering forms on the Internet. Except for one known exception, Chapter 2.5, probably all of them were made considering typical computer screen sizes, pointing devices and computer keyboards, i.e. they are made for a considerable different user interface than what many mobile phones have.

2.1 General Guidelines on Use of Form Controls

In the article "The User Experience, Part 2", Dick Berry at the IBM Ease of Use Architecture and Design team writes about using controls in forms (Berry, 2000). The article presents general guidelines on use of controls, as well as specific guidelines for each type of control. This section summarises the general guidelines.

General Guidelines for Form Controls (Berry, 2000):

1 Use controls consistently
Using controls consistently means to use the same type of control for the same types of information. This accommodates transfer of the users' knowledge from one form to another; they will more quickly recognize what information is requested, and will know how to interact with the control. This also minimises errors.

2 Provide a prompt for each control or group of related controls
Controls should be identified, so that users know what they represent and what they can do. This is usually done by placing a label adjacent to each control. Use a prompt adjacent to the control rather than within the control, as the latter does not let you have a default value, it complicates visual scanning, the fields without an adjacent prompt are easier to miss for the users, there is not enough space for additional useful information and, if the user returns to the field, the prompt will not be there to identify its purpose.

3 Use controls that allow users to choose rather than type
If all or some of the possible values for a field is known, it is usually easier to choose an option, rather than typing it. It also reduces the chances of incorrect input.

4 Avoid using selection controls for actions
Consistently represent actions with push buttons. When using selection controls for actions, it is not clear when the action is performed; whether it is when the action is selected or if one needs to select a push button (such as 'OK') afterwards.

5 Choose the correct control for the situation
Some of Berry's guidelines for specific controls are presented in the next section.

2.2 Choosing and Using Form Elements

This section provides some of the guidelines and advices of several form- and usability-experts on how to use some of the most common controls and labels.

Text Fields
Use text fields when it is more natural to type the answer rather than select it. Personal details are easier and more natural for us to type in directly, as we know them very well. If the answer is easily mis-typed, use one of the other form elements. (Miller et al., 2001)

Text fields are great for entering short pieces of unique information. Try to set the length of the field to the standard entry size, as the length implies what kind of answer you expect. Ensure that the field is clearly labelled, with the label placed adjacent to the field. (Fichter, 2003)

Some entries need to be made on a specific format. In order to avoid frustrated users from poor or lacking instructions regarding format and minimise the potential for errors, you can simply pre-format a text field (Bollaert, 2002). For example can birth dates be entered in three text fields: One for month, one for day and one for year.

Text Boxes
Text boxes are great for long free-form answers. Ensure that the text box wraps the text, as the users reach the text box's end. The size of a text box also implies, like for text fields, what kind of (or how detailed) an answer you are expecting. (Fichter, 2003)

Check Boxes
Use check boxes (or radio buttons) if the users need to review the options to understand the question, and/or if the number of options is four or less, as available space not will be an issue. Otherwise, use a drop-down box. (Miller et al., 2001)

Check boxes work best when users can choose one or more options, and there are five to seven options in total. Make sure the users understand from the labelling whether they can choose one or many. Provide an open-ended option: "Other: _______ " in case you have failed to foresee which options would be appropriate for the users. (Fichter, 2003)

Use check boxes when users are allowed to select several options. The number of options should be five to seven, to make scanning and selection easy. The options should be static. Check boxes that belong to a group should be organised in rows and columns, also to make scanning and selection easy. Single check boxes can be used when offering choices that can be answered with "yes/no", "true/false, etc. In such cases, make sure that users understand both options by reading the label. (Berry, 2000)

Radio Buttons

Use radio buttons (or check boxes) if the users need to review the options to understand the question, and/or if the number of options is four or less, as available space not will be an issue. Otherwise, use a drop-down box. Also use radio buttons with a button for 'any' when you do not want to write instructions about possibility of multiple choices with check boxes, or you suspect that the users will experience trouble with them. (Miller et al., 2001)

Use radio buttons when the users have two or more options to choose from. Industry-standard practice is to have a default selection, where the default is the most likely choice or the first of the options. Offer a button with the value "None" if the users may not like any of the options, and a button with "Any" if any value may be acceptable to them. Users may have problems making a choice if the buttons have unclear labels. Keep the labels short and succinct. Make sure the users can tell from the layout of labels and buttons, which label goes with which button. Vertical layouts are the easiest to scan. (Bollaert, 2002)

Use radio buttons when only one option must be selected. Keep the number of options to five to seven and organise the buttons in rows and columns, to make scanning and selection easy. The options should be static. Be careful with use of group boxes, as they contribute to visual noise. Make an option that is likely to be chosen, the default, to speed up the form process for some users. (Berry, 2000)

Spin Buttons

Use spin buttons to present lists of options when the available space is minimal. Keep the number of options to fewer than 20, to avoid that the users find scrolling the spin button tedious. The options should not be longer than about 20 characters, to facilitate recognition of options when users scan through them. It is important to keep the values in a logical order, so users can predict what the next option is, as they can only see one option at a time. It reduces the chance of scrolling past the desired option. (Berry, 2000)

Drop-Down Boxes

Use a drop-down box if the options are visually distinctive. Otherwise, consider varying or abbreviating the options, or using a text field. Options that look similar are hard to scan through in a list. (Miller et al., 2001)

Drop-down boxes are great for picking one option from a list. Put the options in the drop-down in an order that is logical to the users. Place the box's label adjacent to the box, not in it. (Fichter, 2003)

Drop-downs are space-saving alternatives to radio-buttons. Avoid using drop-downs when it is easier for the users to type information in a text field. If you want to use drop-downs for navigation, pair the drop-down box with a 'Go!' button. Otherwise, the users may be confused when a selection in a drop-down box suddenly takes them to a new screen. The default option when the drop-down is opened should be the option most likely to be selected. If there is no such option, set the default to be a null value that makes the users choose, like "Select One". (Bollaert, 2002)

List Boxes
List boxes work great with from a few options, up to hundreds. Show a reasonable number of options in the display area; give directions on how to select more than one option, if possible; present the options in a logical order and avoid starting several options with the same word(s), as that makes it hard for the users to scan through the list. (Fichter, 2003)

Use a list box to display many options. Provide instructions on the top of the box that indicate whether or not multiple selections can be made in the list, and if they can be made, how. The options in the list should be similar in length, to make scanning through the list easier. Make the order of the options logical to the users. As for drop-down boxes, default the most likely choice to save users an interaction, or make the default a null value. (Bollaert, 2002)

List boxes can be used to display a number of options, from a few up to hundreds, even thousands. Use words that distinguish an option from the next in the list. Lists should provide cursor movement and scrolling shortcuts, like moving the cursor to, and showing the options in, the lists that start with a character typed in by the users. Give directions on how to make multiple selections, if possible. (Berry, 2000)

Combo Boxes
Use combo boxes when the option lists are long, to help users close in on the option they want by displaying the options starting with the typed characters, or/and when users can choose options which are not in the lists. Use combo boxes consisting of text fields and drop-down boxes when space is an issue, spin buttons with text fields when the available space is minimal. (Berry, 2000)

Push Buttons
Be consistent with the behaviour of buttons. For instance, one behaviour in a form can be navigation, while a different behaviour can be to trigger events on the current screen. Too many button behaviours can be confusing. Also be consistent with the placement of buttons, users should not need to relearn the layout for every screen. Place 'Submit' buttons at the bottom of the screen, and 'Reset' buttons in a place where the users will not accidentally click them. (Bollaert, 2002)

Do not use reset buttons in forms, as no one needs to use them. It is usually quicker to edit erroneous fields than it is to erase the form's fields and start over. A reset button can also clutter up the interface and be clicked by mistake, erasing all the entered data. (Nielsen, 2000)

Choice Tables
Use choice tables as a more visually pleasing alternative to radio buttons for small groups of options (less than 20). If the options are very similar and can have small representations, you can display a few hundred options in a table. It is recommended to show the chosen option as "pushed in" in 3-D in the table, like push buttons show their selected state. (Berry, 2000)

Labels
End all labels with a colon, as the colon signals where the label ends and the control begins, also for those who use screen readers. Do not put the label of a control inside it, as the label will not be visible when the users have typed in the control. Line up all fields on a left margin, or right, if the audience reads right to left, to make scanning through a screen easier. (Bollaert, 2002)

2.3 Directions on Design of Web Forms

A bad form design makes form-filling troublesome for the users, and can make them feel inadequate (Fichter, 2003). Darlene Fichter in Northern Lights Internet Solutions, Ltd. provides in the article "Designing Web Forms Using Boxes and Arrows" directions on how to design forms to avoid frustration among the users. Some of her directions on use of controls were presented in the previous section.

<u>Directions on Design of Web Forms (Fichter, 2003):</u>

1 Tell the user upfront the purpose of the form and what it will deliver
Users worry whether the form will be a waste of time. The first area the users look at is the middle area (content area) of the page. Place a title that is clear and descriptive for the purpose of the form in the beginning of this area.

2 Tell the users how long it takes to complete a form
Users worry how long a form will take to complete. Avoid this concern by indicating the number of screens, or if it is a survey, how long it usually takes to complete. Also, in long forms, let users know upfront whether they have to finish it in one sessions, or if they can stop and continue later. When a form is long, try to ensure that users can stop filling it in and continue later.

3 Keep forms short and focused on one activity, if possible
Since long, complicated forms discourage users, try to chop long forms into two shorter ones if you can find a logical division point.

4 Tell users what happens to the information they enter
Users want to know what happens to the information they give. They might feel that their privacy is at stake, and give bogus names, addresses, etc. This may be resolved by telling them why you are asking for the information, and what happens to it when they hit 'Submit'.

5 Make questions clear and unambiguous, and use appropriate selection options
Users may worry about filling in the forms incorrectly. Help them by clear and unambiguous questions and appropriate selection options.

6 Use form validation to identify incomplete or incorrect entries
Form validation is a critical aspect for usable forms. Messages to the users should be in context, preferably placed above the field that needs to be changed. The font style of the messages should stand out and be easy to read. Also use a polite tone.

7 Limit the use of required fields and "close-ended" questions
Users do not like to answer questions, where they can not see any relationship to the task at hand. They "lie" by randomly choosing options in lists, radio buttons and check boxes, just to complete compulsory questions before they can submit the form. You should rather add a text box where users type other options, in addition to select from a list. Give the users a polite message about which fields they have not completed when they hit 'Submit', and ask if they want to complete them or submit the form.

2.4 Statistics Norway's Advices on Design of Web Forms

Statistics Norway's handbook for development of Web forms provides several advices on how to design Web forms. This section renders some of these advices as points.

Advices on Design of Web Forms (Haraldsen, 2004):

1 Separate the form from form activities
It might be best to keep the form's main purpose, collection of information, and the activities connected to the filling of the form separate, by placing them in different windows. These windows should be: The *form field*, containing the information collection elements; a window with *action buttons*; a *form map*, for showing where the users are in the form; and an *information and communication box*, containing help texts and other support functions.

1-1 The form field
The form field can be made easy to spot by making it the largest window. Certain combinations of colours can also make it stand out. The form field should be lighter than the other windows, as light fields on a dark background catch the attention of the users.

1-2 Action buttons
A 'Next' button should be the first choice/available. Research has shown that visibility is more important than the distance from the last field to the buttons. A web form should have four additional buttons: A 'Back' button; a 'Pause' button, a button for going "fast forward"; and a button for activating help functionality. Particularly the pause button is important, as users sometimes need to stop and continue the form filling later.

1-3 Form map
A form map should indicate how large the form is, and how far the users have come in it. It might be best to let the map be possible to open and close as needed by the users. Rather than dividing the map into pages, it might be best to divide it into themes.

1-4 Information and communication box
Choosing the button for activation of help functionality or discovery of an error should trigger a dialogue box popping up. When help functionality has not been activated, there is no point in having an empty window occupy screen space. That a dialogue box pops up is also more striking.

2 Use simple means
Different font styles, numbering, graphics, contrasts and distances between form elements used carefully and consistently can signal the importance of things, what to read first and last and what belongs together or not. Use of different font types, strong colours and colour combinations that the colour blind struggle with, such as red and green, should be avoided.

3 Regard the eyes of the users
Make the form easier to read for the eyes by considering common reading custom, well-organised screens and short distance between pieces of information that should be read in connection to each other. Keep the first elements that should be seen by the users in the upper left side of the screen, as that is where they will look for the start of the form.

4 Beware of means that can affect the answers
Means can ease the burden of filling in a form, but they may also affect how the users perceive the questions. A question containing an illustration will be differently perceived than the same question without the illustration. A question with a number of options, and one of the options as default, makes that option more likely to be chosen. Several means can affect which answers the users enter.

5 Make means work together
Do not mix means that will be competing or in conflict with each other. For instance, if questions are emphasised by making their text bold, do not make other types of form elements bold. The screen should consist of means that give clear and unambiguous signals of what is important, and how the form should be read.

2.5 Web Form Summarisation on Handheld Devices

In the article "Efficient Web Browsing on Handheld Devices Using Page and Form Summarization", a technique for displaying forms on handheld devices is presented. The authors refer to the technique as 'Form summarisation'.

<u>Form Summarisation (Buyukkokten et al., 2002):</u>

As it sometimes may be difficult for users to gain overview of forms on handheld devices, the authors suggest to display only minimal textual prompts for the form fields, illustrated in Figure 19. This will let the users easier scan through the form and understand which information it demands. When the users want to enter information, selecting and activating the desired textual prompt makes the form display the associated control(s), as illustrated in Figure 20.

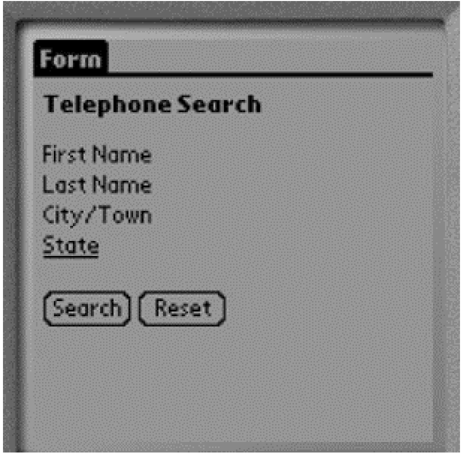


Figure A1. Minimal textual prompts (Buyukkokten et al., 2002).

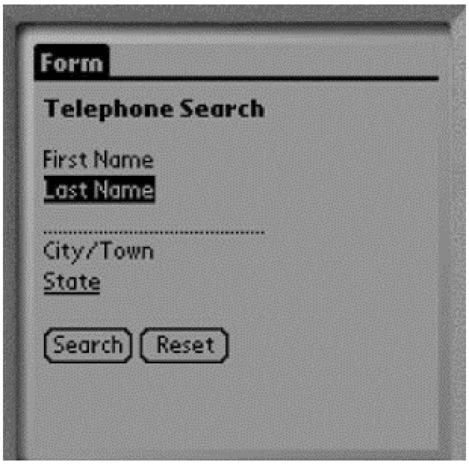


Figure A2. Displaying the associated control (Buyukkokten et al., 2002).

2.6 ELMER 2

The Norwegian Ministry of Trade and Industry adopted ELMER 2, the result of the ELMER project (The ELMER Project, 2007), as "…the common guidelines for user interfaces in public forms for enterprises on the Internet". ELMER 2 is a set of principles and specifications for the design of forms on the Internet intended to be used by small enterprises reporting to the authorities, where employees relate to forms in the same way as private individuals do. (ELMER 2, 2006)

ELMER 2 consists of a number of requirements in several groups, which are divided into five main areas. ELMER 2 also states allowed alternatives for the requirements, when any. This section only renders the main principle for each group. See (ELMER 2, 2006) for the requirements, alternatives, explanatory examples and more.

User Interface Guidelines for Governmental Forms on the Internet (ELMER 2, 2006):
The content of this section is directly rendered from ELMER 2.

1: Components of the form page

1.1: The Navigation Area
A separate navigation menu shall help the form filler orient himself about the main topics of the form, show where he is, and enable him to move between the different form pages. Carefully considered and standardised use of navigation elements shall ensure easy recognition between forms from different inquirers.

1.2: The Information Area
A separate area must be set aside for providing the user with optional additional information in addition to the information given by questions, labels, and messages relating to completion-related errors or ambiguities. User-requested help makes it possible to provide new, desired knowledge to unskilled users without having to overburden the professional user with unwanted information. Carefully considered and standardised use of information elements shall ensure recognition and familiarity between forms issued by different inquirers.

1.3: Other elements
In the world of paper, the actual interaction between the inquirer and the form filler is supported by guidance material and a support apparatus. These functions must also be safeguarded in the electronic forms, above and beyond the use of help texts, and in a way that is easily recognisable from form to form. Also navigation between different form pages, saving, validation and the actual submission process must have a common, well thought through methodology.

2: Structure and Order

2.1: Track options and response-dependent questions
The various questions presented in a form have a varying degree of relevance to different user groups. Helping large user groups to avoid questions that are not relevant to their situation, is perhaps the most important simplification measure which can be achieved by introducing electronic forms. But this requires extensive knowledge as to which fields are relevant to which user.

2.2: Paging and Page order
When the scope and order of form pages is adjusted to fit the medium and the form filler's situation, it may increase the form filler's understanding of the task at hand, the navigation between different pages, and give him or her a better understanding of how the different parts of the form completion are interlinked.

2.3: Page Structure

If the structure and order of questions are adjusted to fit the web medium and the form filler's situation, it may serve to increase overview and understanding of the separate components of the task. This may require great deviation from any paper version of the form.

3: Form elements

3.1: Identification and labels

The label constitutes the primary description or definition of the data to enter in each individual field. In addition, an identifying name or number, as well as the label, may serve as a reference in the communication with the user with regard to the relevant field, topic or form. It is a challenge to find short and concise definitions that are meaningful, and easy to grasp, and to use concepts which the user can be expected to understand, still managing to convey the complete meaning of the content.

3.2: Tables

Many forms contain extensive information sets which have to be entered into a table. Traditionally, this involves particular pedagogical and layout-related challenges with regard to electronic forms as well as for their counterparts on paper.

3.3: Figure Processing in Forms

Many questions used in forms are meant to serve as an educational tool, guiding the form filler toward the correct answer to complicated calculations based on more familiar figures. A uniform notation and processing of figures will increase the form filler's understanding of an area that is of critical importance to data quality.

3.4: Conventions and Symbol Use

In time, many experienced users have become familiar with various standard symbols and descriptions on the Internet. Sticking to the conventions will ease the user's understanding of the functionality and usage of the web form.

4: Help and Feedback Messages

4.1: Prefilling

Applied correctly, prefilled data can ease the workload for the form filler. But checking prefilled data can also be a time consuming task. And prefilled information which cannot be corrected in the same web form in which it appears, may be more confusing than helpful. Appropriate use of prefilled data requires some general rules of play.

4.2: Help texts

In paper-based reporting, providing sufficient information to the unskilled user, without overburdening the expert with unnecessary details, has always been a problem. Correct use of optional (clickable) help texts, in conscious combination with information which all users are exposed to, is a good solution to this dilemma. Optional, clickable information elements at different stages of the completion process, ensure that the information comes at the right time, and only to those who need it.

4.3: Error Messages and Warnings

In web forms it is possible to give feedback on errors and mistakes before submission. This saves time and effort for both the declarant and the inquirer. Where ever the form filler is halted by automatic validations, it is important to provide clear information on how to correct it.

5: The Form Environment

5.1: Concluding Messages
People are often more unsure about whether reports submitted electronically will reach the intended recipient than when they use regular mail. It is therefore important that the recipient returns a clear confirmation that the form has been received, and provide clear information with regard to the further activities – if any – that are expected between the parties.

5.2: Other External Functions related to the Form
Electronic forms are to a varying degree supported by surrounding systems. They may be part of a dedicated form portal and interact closely with this, or they may be free-standing applications. Another difference is that some forms can be submitted openly on the web while others require specific login/authentification mechanisms. For the user's sake, the access to relevant support functions that are not part of the actual form completion, should appear and behave as uniformly as possible, irrespective of these variations.

3 Designing Considering Disabilities

This section first presents the highly acknowledged Web Content Accessibility Guidelines, which were made to ensure access to Web content for people with disabilities. The last section presents some recommendations on how to make Web content accessible to people with cognitive disabilities

3.1 Web Content Accessibility Guidelines 1.0

The Web Accessibility Initiative (WAI) was established by the World Wide Web Consortium (W3C), as they realised there was a need of ensuring that people with disabilities can access web content. The WAI formed a work group to create guidelines for how to achieve accessibility for disabled people. The result is known as the Web Content Accessibility Guidelines (WCAG). Version 1.0 of the guidelines is a W3C recommendation, version 2.0 of the guidelines is still a working draft.

Each guideline has one or several checkpoints. The checkpoints are assigned the priority of 1, 2 or 3, where 1 is a checkpoint that must be satisfied in order for all user groups to be able to access the content, 2 should be satisfied, as one or several groups otherwise will have difficulties, and 3 may be satisfied for groups to avoid slight difficulties with accessing content (Chisholm et al., 1999)

This section will only present the guidelines and shortened descriptions of their checkpoints. The Web Content Accessibility Guidelines 1.0 document also contains the guidelines' reasonings, their priorities, examples and more.

Web Content Accessibility Guidelines 1.0 (Chisholm et al., 1999):

Guideline 1. Provide equivalent alternatives to auditory and visual content.
Provide content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.

No.	Checkpoints
1.1	Provide a text equivalent for every non-text element.
1.2	Provide redundant text links for each active region of a server-side image map.
1.3	Until user agents can automatically read aloud the text equivalent of a visual track, provide an

	auditory description of the important information of the visual track of a multimedia presentation.
1.4	For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation.
1.5	Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.

Table A5. Checkpoints for WCAG guideline 1 (Chisholm et al., 1999).

Guideline 2. Don't rely on color alone.

Ensure that text and graphics are understandable when viewed without color.

No.	Checkpoints
2.1	Ensure that all information conveyed with color is also available without color, for example from context or markup.
2.2	Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.

Table A6. Checkpoints for WCAG guideline 2 (Chisholm et al., 1999).

Guideline 3. Use markup and style sheets and do so properly.

Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.

No.	Checkpoints
3.1	When an appropriate markup language exists, use markup rather than images to convey information.
3.2	Create documents that validate to published formal grammars.
3.3	Use style sheets to control layout and presentation.
3.4	Use relative rather than absolute units in markup language attribute values and style sheet property values.
3.5	Use header elements to convey document structure and use them according to specification.
3.6	Mark up lists and list items properly.
3.7	Mark up quotations. Do not use quotation markup for formatting effects such as indentation.

Table A7. Checkpoints for WCAG guideline 3 (Chisholm et al., 1999).

Guideline 4. Clarify natural language usage

Use markup that facilitates pronunciation or interpretation of abbreviated or foreign text.

No.	Checkpoints
4.1	Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions).
4.2	Specify the expansion of each abbreviation or acronym in a document where it first occurs.
4.3	Identify the primary natural language of a document.

Table A8. Checkpoints for WCAG guideline 4 (Chisholm et al., 1999).

Guideline 5. Create tables that transform gracefully.

Ensure that tables have necessary markup to be transformed by accessible browsers and other user agents.

No.	Checkpoints
5.1	For data tables, identify row and column headers.
5.2	For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.
5.3	Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version.
5.4	If a table is used for layout, do not use any structural markup for the purpose of visual formatting.
5.5	Provide summaries for tables.
5.6	Provide abbreviations for header labels.
Refer also to checkpoint 10.3.	

Table A9. Checkpoints for WCAG guideline 5 (Chisholm et al., 1999).

Guideline 6. Ensure that pages featuring new technologies transform gracefully.
Ensure that pages are accessible even when newer technologies are not supported or are turned off.

No.	Checkpoints
6.1	Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document.
6.2	Ensure that equivalents for dynamic content are updated when the dynamic content changes.
6.3	Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.
6.4	For scripts and applets, ensure that event handlers are input device-independent.
6.5	Ensure that dynamic content is accessible or provide an alternative presentation or page.
Refer also to checkpoint 11.4.	

Table A10. Checkpoints for WCAG guideline 6 (Chisholm et al., 1999).

Guideline 7. Ensure user control of time-sensitive content changes.
Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped.

No.	Checkpoints
7.1	Until user agents allow users to control flickering, avoid causing the screen to flicker.
7.2	Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off).
7.3	Until user agents allow users to freeze moving content, avoid movement in pages.
7.4	Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.
7.5	Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.
Refer also to checkpoint 11.4.	

Table A11. Checkpoints for WCAG guideline 7 (Chisholm et al., 1999).

Guideline 8. Ensure direct accessibility of embedded user interfaces.
Ensure that the user interface follows principles of accessible design: device-independent access to functionality, keyboard operability, self-voicing, etc.

No.	Checkpoints
8.1	Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies.

Table A12. Checkpoints for WCAG guideline 8 (Chisholm et al., 1999).

Guideline 9. Design for device-independence.
Use features that enable activation of page elements via a variety of input devices.

No.	Checkpoints
9.1	Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.
9.2	Ensure that any element that has its own interface can be operated in a device-independent manner.
9.3	For scripts, specify logical event handlers rather than device-dependent event handlers.
9.4	Create a logical tab order through links, form controls, and objects.
9.5	Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.

Table A13. Checkpoints for WCAG guideline 9 (Chisholm et al., 1999).

Guideline 10. Use interim solutions.
Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.

No.	Checkpoints
10.1	Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.
10.2	Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.
10.3	Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.
10.4	Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.
10.5	Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.

Table A14. Checkpoints for WCAG guideline 10 (Chisholm et al., 1999).

Guideline 11. Use W3C technologies and guidelines.
Use W3C technologies (according to specification) and follow accessibility guidelines. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, provide an alternative version of the content that is accessible.

No.	Checkpoints
11.1	Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.
11.2	Avoid deprecated features of W3C technologies.
11.3	Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.).
11.4	If, after best efforts , you cannot create an accessible page, provide a link to an alternative page

	that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.

Table A15. Checkpoints for WCAG guideline 11 (Chisholm et al., 1999).

Guideline 12. Provide context and orientation information.

Provide context and orientation information to help users understand complex pages or elements.

No.	Checkpoints
12.1	Title each frame to facilitate frame identification and navigation.
12.2	Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.
12.3	Divide large blocks of information into more manageable groups where natural and appropriate.
12.4	Associate labels explicitly with their controls.

Table A16. Checkpoints for WCAG guideline 12 (Chisholm et al., 1999).

Guideline 13. Provide clear navigation mechanisms.

Provide clear and consistent navigation mechanisms – orientation information, navigation bars, a site map, etc. – to increase the likelihood that a person will find what they are looking for at a site.

No.	Checkpoints
13.1	Clearly identify the target of each link.
13.2	Provide metadata to add semantic information to pages and sites.
13.3	Provide information about the general layout of a site (e.g., a site map or table of contents).
13.4	Use navigation mechanisms in a consistent manner.
13.5	Provide navigation bars to highlight and give access to the navigation mechanism.
13.6	Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.
13.7	If search functions are provided, enable different types of searches for different skill levels and preferences.
13.8	Place distinguishing information at the beginning of headings, paragraphs, lists, etc.
13.9	Provide information about document collections (i.e., documents comprising multiple pages.).
13.10	Provide a means to skip over multi-line ASCII art.

Table A17. Checkpoints for WCAG guideline 13 (Chisholm et al., 1999).

Guideline 14. Ensure that documents are clear and simple.

Ensure that documents are clear and simple so they may be more easily understood.

No.	Checkpoints
14.1	Use the clearest and simplest language appropriate for a site's content.
14.2	Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.
14.3	Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.).
14.4	Create a style of presentation that is consistent across pages.

Table A18. Checkpoints for WCAG guideline 14 (Chisholm et al., 1999).

3.2 Making Web Content Accessible to the Cognitively Disabled

In the article "Cognitive Disabilities Part 1: We Still Know Too Little, and We Do Even Less", published at WebAIM (Web Accessibility in Mind), Paul Bohman presents recommendations for making Web content accessible to the cognitively disabled. The recommendations are based on existing research, commonly-assumed best practices and Bohman's own ideas. (Bohman, 2005)

Recommendations for Making Web Content Accessible to People with Cognitive Disabilities (Bohman, 2005):
Directly rendered from (Bohman, 2005).

1 Create transformable, rich, multi-modal content

1.1 Transformable

1.1.1	**Allow fonts to be enlarged.** The ability to enlarge fonts is dependent upon the capabilities of the user agent (browser), but relative units are recommended, rather than absolute units. For example, use "em" or "%" rather than "pt" or "cm".
1.1.2	**Use real text or vector-based text, rather than text within raster-based images, to allow for higher quality enlargement, without pixilation.** Real text is always the best, most transformable method of conveying text content. When text is used within graphics or rich media, vector-based formats (flash, SVG) enlarge better than raster-based formats (e.g. jpg, gif, bmp).
1.1.3	**Provide all content in a text format so that it can be read aloud by text-to-speech synthesizers.** Content can be in plain text, HTML, in alt text for images, or in any other format with true text that can be accessed by assistive technologies. It may be appropriate in some circumstances to provide the text version separate from the rich media version (e.g. text transcripts for videos).

Table A19. Recommendations for accessible Web content, 1.1 Transformable (Bohman, 2005).

1.2 Multi-modal

1.2.1	**Illustrate concepts with drawings, diagrams, photos, audio files, video clips, animations, and other non-textual media.** Communicate with the user through as many different sensory modalities and input modes as possible (sight, hearing, interaction, reading, etc.) to increase the chance that content will be understood.
1.2.2	**Provide synchronized captions and transcripts for the audio portion of time-based media.** Add captions to video files (e.g. using SMIL or SAMI) and provide a link to a text transcript.
1.2.3	**Provide audio descriptions of visual events in time-based media.** Narrate the visual actions in videos so that the video can be understood by listening to it, without watching it.

Table A20. Recommendations for accessible Web content, 1.2 Multi-modal (Bohman, 2005).

2 Focus the attention of the user

2.1 Sensory focus

2.1.1	**Use softer colors (e.g. pastels) for graphical elements, rather than sharply contrasting colors.** (Note: this is not widely accepted) When using background colors to differentiate sections of the same page, choose softer

	colors, rather than high contrast colors.
2.1.2	**Limit the types of font faces in a document**. Use only one font, or a very small number of fonts in any single document.
2.1.3	**Limit or eliminate the use of italics or ALL CAPS.** Avoid italics and all caps to the extent possible, to improve readability.
2.1.4	**Avoid background sounds that distract the user's attention (e.g. background music).** Allow the user to focus on the main content without audio distractions.
2.1.5	**Use sounds to focus the user's attention (e.g. give instructions, alert the user to errors, etc.).** Provide audible cues that help the user to focus on the main content.
2.1.6	**Include "white space" – non-content space – around the content, between paragraphs, and between headings.** Don't crowd the design visually.
2.1.7	**Avoid complex or "busy" visual backgrounds.** Don't create extraneous visual information that distracts from the main content.

Table A21. Recommendations for accessible Web content, 2.1 Sensory focus (Bohman, 2005).

2.2 Content focus

2.2.1	**Place the important parts of a paragraph (key points) in the first sentence.** Don't hide important points in the middle of paragraphs.
2.2.2	**Organize content into well-defined groups or chunks, using headings, bulleted lists, and other visual-semantic organizing schemes.** Make the document's structure as obvious as possible.
2.2.3	**Highlight text as it's being read out loud (or allow users to activate this option).** This recommendation is most applicable to rich media, such as Flash and SVG, that does not have native captioning capabilities, and where captions are added through programming or scripting.
2.2.4	**Emphasize important text – or the headings to sections of text – with bold font faces or larger text size.** Use bold and/or large text to visually emphasize important text. Note: the default style of HTML headings is bold and large, so there is no need to use extra tags or markup to achieve this affect in HTML.

Table A22. Recommendations for accessible Web content, 2.2 Content focus (Bohman, 2005).

2.3 Interaction focus

2.3.1	**Provide multi-modal navigational cues (e.g. text + graphical/visual highlight + auditory instructions + animated demonstration).** Help users know what to do and how to interact with the content (e.g. create an audible voice that says "click the 'next' button to go to the next page", or a sound to accompany error messages, or visually highlight the 'next' button, etc.)
2.3.2	**Give feedback on a user's actions (e.g. confirm correct choices, alert users to errors or possible errors).**
2.3.3	**Provide instructions for unfamiliar interfaces.**

Table A23. Recommendations for accessible Web content, 2.3 Interaction focus (Bohman, 2005).

3 Design a consistent environment

3.1 Ensure that similar interface elements and similar interactions produce predictably similar results

3.2 Create a navigational scheme that is consistent across pages within a site or within related sections of a site

4 Create simple, concise content

4.1 Use clear and simple language
This recommendation is difficult to evaluate, but important.

4.2 Avoid tangential information
Stick to the main topic.

4.3 Use correct grammar and spelling
Use a spell-checker. Write well.

5 Allow the user sufficient time to access and interact with content

5.1 Don't set short "expiration" times on content
To the extent possible, avoid time-dependent JavaScript, HTML auto-refreshing, and other types of timed redirects.

5.2 If expiration times are necessary allow the user to request more time
Allow users to set preferences and/or alert users when time is about to expire and give them the option to extend the timeline.

6 Allow users to recover from accidental and erroneous interactions

6.1 Ask users to confirm choices

6.2 Use shorter, multi-step forms for complex interactions, rather than lengthy, all-in-one forms

Appendix B: Javadoc of the Prototype

The source code of the prototype can be found in a separate ZIP-file.

Hierarchy For All Packages

Package Hierarchies:
framework, kindergartenform

Class Hierarchy

- java.lang.Object
    - kindergartenform.**Activity**
    - no.tellu.cdlc.gui.GuiElement (implements no.tellu.cdlc.gui.ElementListner)
        - framework.**CheckBoxElement**
        - no.tellu.cdlc.gui.GroupElement
            - kindergartenform.**ActivityElement**
            - kindergartenform.**FormGroupElement**
            - kindergartenform.**SummaryElement**
        - no.tellu.cdlc.gui.GuiPanel
            - kindergartenform.**FormPageEight**
            - kindergartenform.**FormPageFive**
            - kindergartenform.**FormPageFour**
            - kindergartenform.**FormPageOne**
            - kindergartenform.**FormPageSeven**
            - kindergartenform.**FormPageSix**
            - kindergartenform.**FormPageThree**
            - kindergartenform.**FormPageTwoList**
            - kindergartenform.**FormPageTwoMap**
            - no.tellu.cdlc.gui.TabbedElement (implements no.tellu.cdlc.gui.ElementListner)
                - kindergartenform.**FormTabbedElement**
        - framework.**MapElement**
        - framework.**POIElement**
        - framework.**RadioButtonElement**
    - no.tellu.cdlc.gui.GuiWindow (implements no.tellu.cdlc.gui.ElementListner)
        - kindergartenform.**DailyRoutineWindow**
        - kindergartenform.**FormWindow**
    - framework.**MapLayout** (implements no.tellu.cdlc.gui.LayoutManager)
    - framework.**MapObject**
    - javax.microedition.midlet.MIDlet
        - kindergartenform.**KindergartenFormMIDlet**
    - framework.**POIInformation**
        - kindergartenform.**KindergartenPOIInfo**
    - framework.**POIObject**

Package framework

Class Summary	
CheckBoxElement	CheckBoxelement.java Draws (a) check box(es).
MapElement	MapElement.java Draws a map with the given height and the max width of the element.
MapLayout	MapLayout.java Handles the layout of a panel containing a map and POIs.
MapObject	MapObject.java Contains all information in a map, including its points of interest.
POIElement	POIElement.java Draws the points of interest in a map.
POIInformation	POIInformation.java Contains a POI's key information that is not related to coordinates.
POIObject	POIObject.java Contains all POI information.
RadioButtonElement	RadioButtonelement.java Draws radio buttons The radio button alternatives are set in the choices vector by calling setChoices(Vector choices).

---

framework

Class CheckBoxElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └framework.CheckBoxElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class CheckBoxElement
extends no.tellu.cdlc.gui.GuiElement
```

CheckBoxelement.java

Draws (a) check box(es).

The check box alternatives are set in the choices vector by calling setChoices(Vector choices).

Pressing fire checks/unchecks the selected check box.

Constructor Summary

`CheckBoxElement``()` Constructor CheckBoxElement

Method Summary

`void`	**`elementLayoutManager`**`()` Controls the height check box element.
`boolean`	**`fire`**`(int key)` Sets the check box to be chosen if it is not already, and unchooses it if it is chosen.
`void`	**`paint`**`(javax.microedition.lcdui.Graphics g, boolean selected)` Paints the check box, its text and a dotted rectangle around it if it is selected.
`void`	**`setChoices`**`(java.util.Vector choices)` Sets the check box alternative(s).

Constructor Detail

CheckBoxElement

```
public CheckBoxElement()
```

Constructor CheckBoxElement

Method Detail

elementLayoutManager

```
public void elementLayoutManager()
```

Controls the height of the check box element.

Overrides:

`elementLayoutManager` in class `no.tellu.cdlc.gui.GuiElement`

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the check box, its text and a dotted rectangle around it if it is selected.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiElement`

setChoices

```
public void setChoices(java.util.Vector choices)
```

Sets the check box alternative(s).

Parameters:

`choices` - a vector containing string alternatives

---

fire

```
public boolean fire(int key)
```

Sets the check box to be chosen if it is not already, and unchooses it if it is chosen.

Overrides:

`fire` in class `no.tellu.cdlc.gui.GuiElement`

---

framework

Class MapElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └framework.MapElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class MapElement
extends no.tellu.cdlc.gui.GuiElement
```

MapElement.java

Draws a map with the given height and the max width of the element. The map is not selectable.

Constructor Summary
MapElement(java.lang.String elementId, int mapWidth, int mapHeight, MapObject map) Constructor MapElement

Method Summary	
void	**paint**(javax.microedition.lcdui.Graphics g, boolean selected) Paints the map image and a border around it.

Constructor Detail

MapElement

```
public MapElement(java.lang.String elementId,
                  int mapWidth,
                  int mapHeight,
                  MapObject map)
```

Constructor MapElement

Parameters:

`elementId` - The id of the object of this class

`mapHeight` - The map height to be set

`map` - The object containing the map information

Method Detail

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the map image and a border around it.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiElement`

---

framework

Class MapLayout

```
java.lang.Object
  └ framework.MapLayout
```

All Implemented Interfaces:

no.tellu.cdlc.gui.LayoutManager

```
public class MapLayout
extends java.lang.Object
implements no.tellu.cdlc.gui.LayoutManager
```

MapLayout.java

Handles the layout of a panel containing a map and POIs.

Constructor Summary

MapLayout()

Method Summary

void	**addLayoutComponent**(no.tellu.cdlc.gui.GuiElement comp, int constraint) Empty.
int	**getConstraint**(no.tellu.cdlc.gui.GuiElement comp) Get the constraint applied when the component was added.

`int`	`getHorizontalGap()` Get horizontal spacing between layed out elements.
`int`	`getVerticalGap()` Get vertical spacing between layed out elements.
`void`	`layoutContainer(no.tellu.cdlc.gui.GuiPanel parent)` Sets the x and y coordinates of the MapElement and POIElement.
`no.tellu.cdlc.gui.Dimension`	`minimumLayoutSize(no.tellu.cdlc.gui.GuiPanel parent)` Calculates the size dimensions for the specified container, given the components it contains.
`no.tellu.cdlc.gui.Dimension`	`preferredLayoutSize(no.tellu.cdlc.gui.GuiPanel parent)` Calculates the preferred size dimensions for the specified container, given the components it contains.
`void`	`removeLayoutComponent(no.tellu.cdlc.gui.GuiElement comp)` Empty.
`void`	`setHorizontalGap(int horizontalSpacing)` Empty.
`void`	`setVerticalGap(int verticalSpacing)` Empty.

Constructor Detail

MapLayout

```
public MapLayout()
```

Method Detail

addLayoutComponent

```
public void addLayoutComponent(no.tellu.cdlc.gui.GuiElement comp,
                               int constraint)
```

Empty. MapLayout must implement the inherited abstract method LayoutManager.addLayoutComponent(GuiElement, int).

Specified by:

`addLayoutComponent` in interface `no.tellu.cdlc.gui.LayoutManager`

---

removeLayoutComponent

```
public void removeLayoutComponent(no.tellu.cdlc.gui.GuiElement comp)
```

Empty. MapLayout must implement the inherited abstract method LayoutManager.removeLayoutComponent(GuiElement).

Specified by:

`removeLayoutComponent` in interface `no.tellu.cdlc.gui.LayoutManager`

---

minimumLayoutSize

```
public no.tellu.cdlc.gui.Dimension
minimumLayoutSize(no.tellu.cdlc.gui.GuiPanel parent)
```

Calculates the size dimensions for the specified container, given the components it contains.

Specified by:

`minimumLayoutSize` in interface `no.tellu.cdlc.gui.LayoutManager`

Parameters:

`parent` - the container to be laid out

Returns:

the layout size to be paintend

---

preferredLayoutSize

```
public no.tellu.cdlc.gui.Dimension
preferredLayoutSize(no.tellu.cdlc.gui.GuiPanel parent)
```

Calculates the preferred size dimensions for the specified container, given the components it contains.

Specified by:

`preferredLayoutSize` in interface `no.tellu.cdlc.gui.LayoutManager`

Parameters:

`parent` - the container to be laid out

Returns:

the layout size to be paintend

---

layoutContainer

```
public void layoutContainer(no.tellu.cdlc.gui.GuiPanel parent)
```

Sets the x and y coordinates of the MapElement and POIElement.

Specified by:

`layoutContainer` in interface `no.tellu.cdlc.gui.LayoutManager`

Parameters:

`parent` - the specified component being laid out

---

setVerticalGap

```
public void setVerticalGap(int verticalSpacing)
```

Empty. MapLayout must implement the inherited abstract method LayoutManager.setVerticalGap(int).

Specified by:

`setVerticalGap` in interface `no.tellu.cdlc.gui.LayoutManager`

---

getVerticalGap

```
public int getVerticalGap()
```

Get vertical spacing between layed out elements.

Specified by:

`getVerticalGap` in interface `no.tellu.cdlc.gui.LayoutManager`

Returns:

0

---

setHorizontalGap

```
public void setHorizontalGap(int horizontalSpacing)
```

Empty. MapLayout must implement the inherited abstract method LayoutManager.setHorizontalGap(int).

Specified by:

`setHorizontalGap` in interface `no.tellu.cdlc.gui.LayoutManager`

---

getHorizontalGap

```
public int getHorizontalGap()
```

Get horizontal spacing between layed out elements.

Specified by:

`getHorizontalGap` in interface `no.tellu.cdlc.gui.LayoutManager`

Returns:

0

---

getConstraint

```
public int getConstraint(no.tellu.cdlc.gui.GuiElement comp)
```

Get the constraint applied when the component was added.

Specified by:

`getConstraint` in interface `no.tellu.cdlc.gui.LayoutManager`

Parameters:

`comp` - component

Returns:

constraint return -1 if not available

---

framework

Class MapObject

```
java.lang.Object
  └ framework.MapObject
```

```
public class MapObject
extends java.lang.Object
```

MapObject.java

Contains all information in a map, including its points of interest.

This is originally MapObject.java in no.tellu.commonplatform.serviceframe.common.map, slightly changed, to be able to use a map image. Redundant code from no.tellu.commonplatform.serviceframe.common.map.MapObject has been removed from this file.

Constructor Summary
`MapObject``(javax.microedition.lcdui.Image mapImage, java.util.Vector POIs, no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox mapBoundingBox)` Constructor MapObject

Method Summary

no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox	**getMapBoundingBox**()
javax.microedition.lcdui.Image	**getMapImage**()
java.util.Vector	**getPOIs**()

Constructor Detail

MapObject

```
public MapObject(javax.microedition.lcdui.Image mapImage,
                 java.util.Vector POIs,

no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox mapBoundingBox)
```

Constructor MapObject

Parameters:

`mapImage` - An image containing a map

`POIs` - A vector with POIObject objects (points of interest)

`mapBoundingBox` - An object containing map coordinates

Method Detail

getMapImage

```
public javax.microedition.lcdui.Image getMapImage()
```

Returns:

the image of the map

---

getPOIs

```
public java.util.Vector getPOIs()
```

Returns:

the vector with points of interest

---

getMapBoundingBox

```
public no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox
getMapBoundingBox()
```

Returns:

the object with map coordinates

---

framework

Class POIElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └framework.POIElement
```

All Implemented Interfaces:
no.tellu.cdlc.gui.ElementListner

```
public class POIElement
extends no.tellu.cdlc.gui.GuiElement
```

POIElement.java

Draws the points of interest in a map. The POIs can be selected by pressing the number (1-9) they are represented by. When a POI is selected, a popup with information is drawn. Choosing POIs is done by pressing the fire button.

The POIElement is used together with MapElement and MapLayout in a panel.

Constructor Summary
`POIElement``(java.lang.String elementId, int mapWidth, int mapHeight, MapObject map, int maxNumberOfPOIsToSave)` Constructor POIElement

Method Summary	
`void`	**`deleteSavedPOI`**`()` Deletes a saved POI
`boolean`	**`fire`**`(int arg0)` Saves the selected POI if it is not saved, deletes it otherwise.
`int`	**`getMaxNumberOfPOIsToSave`**`()`
`int`	**`getPOIIndexInSavedVector`**`(POIObject poiObjectOnMap)` Finds the index number of a POI in the saved POIs vector
`java.util.Vector`	**`getSavedPOIs`**`()`
`POIObject`	**`getSelectedPOI`**`()`
`boolean`	**`numberPressed`**`(int key)` Keeps control of which kindergarten is selected by the user.
`void`	**`paint`**`(javax.microedition.lcdui.Graphics g, boolean selected)` Paints the selectable and not selectable points of interest.
`boolean`	**`savePOI`**`()` Saves a POI in the saved POIs vector
`void`	**`setMaxNumberOfPOIsToSave`**`(int maxNumberOfPOIsToChoose)` Sets the highest number of kindergartens possible to save.
`void`	**`setSavedPOIs`**`(java.util.Vector savedPOIs)` Sets the saved POIs.
`void`	**`setSelectedPOI`**`(POIObject selectedPOI)`

	Sets the selected point of interest.

Constructor Detail

POIElement

```
public POIElement(java.lang.String elementId,
                  int mapWidth,
                  int mapHeight,
                  MapObject map,
                  int maxNumberOfPOIsToSave)
```

Constructor POIElement

Parameters:

`elementId` - The id of the object of this class

`mapHeight` - The map height to be set

`map` - The object containing the map information

`maxNumberOfPOIsToSave` - The maximum number of POIS that can be saved

Method Detail

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the selectable and not selectable points of interest.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiElement`

---

savePOI

```
public boolean savePOI()
```

Saves a POI in the saved POIs vector

Returns:

false if the POI already has been saved, true otherwise

---

deleteSavedPOI

```
public void deleteSavedPOI()
```

Deletes a saved POI

---

getPOIIndexInSavedVector

```
public int getPOIIndexInSavedVector(POIObject poiObjectOnMap)
```

Finds the index number of a POI in the saved POIs vector

Parameters:

`poiObjectOnMap` - the POI to be searched for

Returns:

the index number or -1 if the POI isn't in the vector

---

getSavedPOIs

```
public java.util.Vector getSavedPOIs()
```

Returns:

the vector with saved points of interest

---

setSavedPOIs

```
public void setSavedPOIs(java.util.Vector savedPOIs)
```

Sets the saved POIs.

Parameters:

`savedPOIs` - the vector with saved POIs

---

numberPressed

```
public boolean numberPressed(int key)
```

Keeps control of which kindergarten is selected by the user.

Overrides:

`numberPressed` in class `no.tellu.cdlc.gui.GuiElement`

---

fire

```
public boolean fire(int arg0)
```

Saves the selected POI if it is not saved, deletes it otherwise.

Overrides:

`fire` in class `no.tellu.cdlc.gui.GuiElement`

---

getMaxNumberOfPOIsToSave

```
public int getMaxNumberOfPOIsToSave()
```

Returns:

the highest number of kindergartens possible to save.

---

setMaxNumberOfPOIsToSave

```
public void setMaxNumberOfPOIsToSave(int maxNumberOfPOIsToChoose)
```

Sets the highest number of kindergartens possible to save.

Parameters:

`maxNumberOfPOIsToChoose` - the highest number of kindergartens possible to save

---

getSelectedPOI

```
public POIObject getSelectedPOI()
```

Returns:

the selected point of interest

---

setSelectedPOI

```
public void setSelectedPOI(POIObject selectedPOI)
```

Sets the selected point of interest.

Parameters:

`selectedPOI` - the point of interest to be set as selected

---

framework

Class POIInformation

```
java.lang.Object
  └ framework.POIInformation
```

Direct Known Subclasses:

KindergartenPOIInfo

```
public class POIInformation
```

```
extends java.lang.Object
```

POIInformation.java

Contains a POI's key information that is not related to coordinates.

Use setExtraPopupInfo() to set any extra infomation to be drawn in the popup of a POI in a map

Constructor Summary

`POIInformation``(java.lang.String name, java.lang.String streetAddress, java.lang.String postCode, java.lang.String postAddress)`
Constructor POIInformation

Method Summary

`java.lang.String`	**`getExtraPopupInfo`**`()` Gets extra info about the POI, to be drawn in a popup.
`java.lang.String`	**`getName`**`()`
`java.lang.String`	**`getPostAddress`**`()`
`java.lang.String`	**`getPostCode`**`()`
`java.lang.String`	**`getStreetAddress`**`()`
`void`	**`setExtraPopupInfo`**`(java.lang.String extraPopupInfo)` Sets the extra POI info to be drawn in a popup.
`void`	**`setName`**`(java.lang.String name)` Sets the name of the POI.
`void`	**`setPostAddress`**`(java.lang.String postAddress)` Sets the post address of the POI.
`void`	**`setPostCode`**`(java.lang.String postCode)` Sets the post code of the POI.
`void`	**`setStreetAddress`**`(java.lang.String streetAddress)` Sets the street address of the POI.

Constructor Detail

POIInformation

```
public POIInformation(java.lang.String name,
                      java.lang.String streetAddress,
                      java.lang.String postCode,
                      java.lang.String postAddress)
```

Constructor POIInformation

Parameters:
`name` - Name of the POI
`streetAddress` - The street address of the POI
`postCode` - The POI's post-/zipcode
`postAddress` - The POI's post address

Method Detail

getName

```
public java.lang.String getName()
```

Returns:
name of the POI

---

setName

```
public void setName(java.lang.String name)
```

Sets the name of the POI.
Parameters:
`name` - name of the POI

---

getStreetAddress

```
public java.lang.String getStreetAddress()
```

Returns:
the street address of the POI

---

setStreetAddress

```
public void setStreetAddress(java.lang.String streetAddress)
```

Sets the street address of the POI.
Parameters:
`streetAddress` - the street address of the POI

---

getPostAddress

```
public java.lang.String getPostAddress()
```

Returns:
the post address of the POI

---

setPostAddress

```
public void setPostAddress(java.lang.String postAddress)
```

Sets the post address of the POI.
Parameters:
`postAddress` - the post address of the POI

---

getPostCode

```
public java.lang.String getPostCode()
```

Returns:
the postcode of the POI

---

setPostCode

```
public void setPostCode(java.lang.String postCode)
```

Sets the post code of the POI.
Parameters:

`postCode` - the post code of the POI

getExtraPopupInfo

```
public java.lang.String getExtraPopupInfo()
```

Gets extra info about the POI, to be drawn in a popup.

Returns:

the string with extra info about the POI

setExtraPopupInfo

```
public void setExtraPopupInfo(java.lang.String extraPopupInfo)
```

Sets the extra POI info to be drawn in a popup.

Parameters:

`extraPopupInfo` - the extra info about the POI

framework

Class POIObject

```
java.lang.Object
  └ framework.POIObject
```

```
public class POIObject
extends java.lang.Object
```

POIObject.java

Contains all POI information.

A POI is selectable by default. Use setNotSelectable() if the POI is not supposed to be selectable.

This is originally POIObject.java in no.tellu.commonplatform.serviceframe.common.map, Methods for finding x and y coordinates to a POI in a map, variables 'id', 'selected', 'info', and 'selectable' have been added and the variables 'longitude' and 'latitude' have been changed to 'easting' and 'northing'. Redundant code from no.tellu.commonplatform.serviceframe.common.map.POIObject has been removed from this file.

Constructor Summary
`POIObject``(java.lang.String easting, java.lang.String northing, java.lang.String id, POIInformation info)` Constructor POIObject

Method Summary	
`java.lang.String`	**`getEasting`**`()`

`java.lang.String`	**`getId`**`()`
`POIInformation`	**`getInfo`**`()` Gets the POI information not coordinate related.
`java.lang.String`	**`getNorthing`**`()`
`java.lang.String`	**`getPoiImage`**`()`
`int`	**`getXCoordinate`**`(int startX, int width, no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox mbb)` Finds the x coordinate of a POI within a given area
`int`	**`getYCoordinate`**`(int startY, int height, no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox mbb)` Finds the y coordinate of a POI within a given area
`boolean`	**`isSelectable`**`()`
`boolean`	**`isSelected`**`()`
`void`	**`setEasting`**`(java.lang.String easting)` Sets the easting coordinate of the POI.
`void`	**`setId`**`(java.lang.String id)` Sets the ID of the POI.
`void`	**`setInfo`**`(POIInformation info)` Sets the POI information not coordinate related.
`void`	**`setNorthing`**`(java.lang.String northing)` Sets the northing coordinate of the POI.
`void`	**`setNotSelectable`**`()` Sets the POI to not be possible to select.
`void`	**`setPoiImage`**`(java.lang.String poiImage)` Sets the POI icon.
`void`	**`setSelected`**`(boolean selected)` Sets the POI to be selected or unselects it.

Constructor Detail

POIObject

```
public POIObject(java.lang.String easting,
                 java.lang.String northing,
                 java.lang.String id,
                 POIInformation info)
```

Constructor POIObject

Parameters:

`easting` - The easting coordinate

`northing` - The northing coordinate

`id` - ID of the POI

`info` - A POIInformation object containing general location information not related to coordinates

Method Detail

getEasting

```
public java.lang.String getEasting()
```

Returns:

the easting coordinate of the POI

---

setEasting

```
public void setEasting(java.lang.String easting)
```

Sets the easting coordinate of the POI.

Parameters:

`easting` - the easting coordinate of the POI

---

getNorthing

```
public java.lang.String getNorthing()
```

Returns:

the northing coordinate of the POI

---

setNorthing

```
public void setNorthing(java.lang.String northing)
```

Sets the northing coordinate of the POI.

Parameters:

`northing` - the northing coordinate of the POI

---

getInfo

```
public POIInformation getInfo()
```

Gets the POI information not coordinate related.

Returns:

information about the POI not coordinate related

---

setInfo

```
public void setInfo(POIInformation info)
```

Sets the POI information not coordinate related.

Parameters:

`info` - the object with POI information

---

getPoiImage

```
public java.lang.String getPoiImage()
```

Returns:

the POI icon

---

setPoiImage

```
public void setPoiImage(java.lang.String poiImage)
```

Sets the POI icon.

Parameters:

`poiImage` - the icon ofthe POI

setSelected

```
public void setSelected(boolean selected)
```

Sets the POI to be selected or unselects it.

Parameters:

`selected` - true if the POI shoud be selected, false if not

isSelected

```
public boolean isSelected()
```

Returns:

true if the POI is selected, false if not

setNotSelectable

```
public void setNotSelectable()
```

Sets the POI to not be possible to select.

isSelectable

```
public boolean isSelectable()
```

Returns:

true if the POI is selectable, false if not

getXCoordinate

```
public int getXCoordinate(int startX,
                          int width,

no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox mbb)
```

Finds the x coordinate of a POI within a given area

Parameters:

`startX` - The area's leftmost x coordinate

`width` - The width of the area

`mbb` - The area's coordinates

Returns:

The x coordinate

getYCoordinate

```
public int getYCoordinate(int startY,
                          int height,

no.tellu.commonplatform.serviceframe.common.map.MapBoundingBox mbb)
```

Finds the y coordinate of a POI within a given area

Parameters:

`startY` - The area's upper coordinate

`height` - The area's height

`mbb` - The area's coordinates

Returns:

The y coordinate

getId

```
public java.lang.String getId()
```

Returns:

the ID of the POI

setId

```
public void setId(java.lang.String id)
```

Sets the ID of the POI.

Parameters:

`id` - the ID of the POI

---

framework

Class RadioButtonElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └framework.RadioButtonElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class RadioButtonElement
extends no.tellu.cdlc.gui.GuiElement
```

RadioButtonelement.java

Draws radio buttons

The radio button alternatives are set in the choices vector by calling setChoices(Vector choices).

Pressing fire activates the radio button element. Selecting a different button is then done by pressing right/left.

Constructor Summary
`RadioButtonElement``()` Constructor RadioButtonElement
`RadioButtonElement``(java.lang.String elementId)` Constructor RadioButtonElement

Method Summary	
`void`	**`elementLayoutManager`**`()` Updates the width and the height of the RadioButtonElement
`boolean`	**`fire`**`(int key)` Activates the radio buttons if they're not, deactivates them otherwise.
`int`	**`getSelectedIndex`**`()`

java.lang.String	**getStringValue**()
boolean	**moveLeft**(int x) Handles the selection of radio buttons when the key for moving left is pressed.
boolean	**moveRight**(int x) Handles the selection of radio buttons when the key for moving right is pressed.
void	**paint**(javax.microedition.lcdui.Graphics g, boolean selected) Paints the radio buttons, their text and a dotted rectangle around the selected radio button.
void	**setChoices**(java.util.Vector choices) Sets the radio buttons' alternatives.

Constructor Detail

RadioButtonElement

```
public RadioButtonElement()
```

Constructor RadioButtonElement

RadioButtonElement

```
public RadioButtonElement(java.lang.String elementId)
```

Constructor RadioButtonElement

Parameters:

`elementId` - The id of the object of this class

Method Detail

elementLayoutManager

```
public void elementLayoutManager()
```

Updates the width and the height of the RadioButtonElement

Overrides:

`elementLayoutManager` in class `no.tellu.cdlc.gui.GuiElement`

---

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the radio buttons, their text and a dotted rectangle around the selected radio button.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiElement`

---

setChoices

```
public void setChoices(java.util.Vector choices)
```

Sets the radio buttons' alternatives.

Parameters:

`choices` - a vector containing string alternatives

getStringValue

```
public java.lang.String getStringValue()
```

Returns:

the alternative text of the selected radio button

getSelectedIndex

```
public int getSelectedIndex()
```

Returns:

the index of the selected radio button

moveLeft

```
public boolean moveLeft(int x)
```

Handles the selection of radio buttons when the key for moving left is pressed.

Overrides:

`moveLeft` in class `no.tellu.cdlc.gui.GuiElement`

moveRight

```
public boolean moveRight(int x)
```

Handles the selection of radio buttons when the key for moving right is pressed.

Overrides:

`moveRight` in class `no.tellu.cdlc.gui.GuiElement`

fire

```
public boolean fire(int key)
```

Activates the radio buttons if they're not, deactivates them otherwise.

Overrides:

`fire` in class `no.tellu.cdlc.gui.GuiElement`

Package kindergartenform

Class Summary	
Activity	Activity.java Contains the time, clock image, name and extended info of an activity.
ActivityElement	ActivityElement.java Draws an activity element in a list of activities (ActivityPanelElement).
DailyRoutineWindow	DailyRoutineWindow.java

	Contains the information shown in the daily routine window.
FormGroupElement	FormGroupElement.java Created to support functionality that GroupElement doesn't.
FormPageEight	FormPageEight.java Contains the content of the eight page of the form.
FormPageFive	FormPageFive.java Contains the content of the fifth page of the form.
FormPageFour	FormPageFour.java Contains the content of the fourth page of the form.
FormPageOne	FormPageOne.java Contains the content of the first page of the form.
FormPageSeven	FormPageSeven.java Contains the content of the seventh page of the form.
FormPageSix	FormPageSix.java Contains the content of the sixth page of the form.
FormPageThree	FormPageThree.java Contains the content of the third page of the form.
FormPageTwoList	FormPageTwoList.java Contains the content of the second page of the form, presented as a list.
FormPageTwoMap	FormPageTwoMap.java Contains the content of the second page of the form, presented in a map.
FormTabbedElement	FormTabbedElement.java Contains methods for changing, adding and removing form pages (panels)
FormWindow	FormWindow.java Contains the data and pages of the kindergarten form application
KindergartenFormMIDlet	KindergartenFormMIDlet.java The MIDlet class enables creating, starting and stopping the application.

KindergartenPOIInfo	KindergartenPOIInfo.java Contains kindergarten specificinformation.
SummaryElement	SummaryElement.java Draws a summary element in a list of main topics.

kindergartenform

Class Activity

```
java.lang.Object
  └─kindergartenform.Activity
```

```
public class Activity
extends java.lang.Object
```

Activity.java

Contains the time, clock image, name and extended info of an activity.

Constructor Summary
Activity(java.lang.String timeString, java.lang.String timeImage, java.lang.String activity, java.lang.String extendedInfo) Constructor Activity

Method Summary	
java.lang.String	**getActivity**()
java.lang.String	**getExtendedInfo**()
java.lang.String	**getTimeImage**()
java.lang.String	**getTimeString**()

Constructor Detail

Activity

```
public Activity(java.lang.String timeString,
                java.lang.String timeImage,
                java.lang.String activity,
                java.lang.String extendedInfo)
```

Constructor Activity

Parameters:

`timeString` - Starting time of an activity

`timeImage` - Starting time icon

`activity` - Name of the activity

`extendedInfo` - Description of the activity

Method Detail

getTimeString

```
public java.lang.String getTimeString()
```

Returns:

the time of the activity, in text

---

getTimeImage

```
public java.lang.String getTimeImage()
```

Returns:

the time of the activity, as an icon

---

getActivity

```
public java.lang.String getActivity()
```

Returns:

the name of the activity

---

getExtendedInfo

```
public java.lang.String getExtendedInfo()
```

Returns:

the information about the activity

---

kindergartenform

Class ActivityElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GroupElement
          └kindergartenform.ActivityElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class ActivityElement
extends no.tellu.cdlc.gui.GroupElement
```

ActivityElement.java

Draws an activity element in a list of activities (ActivityPanelElement). The element consists of a clock icon or time string, name of the activity and, if the activity has a description, an information icon.

If the activity has extended information, pressing fire displays a description of the activity.

Constructor Summary

ActivityElement() Constructor ActivityElement

Method Summary

void	**elementLayoutManager**() Updates the layout of the extended information object
boolean	**fire**(int key) Sets the extended information to be shown if it is not, hides it if it is already shown.
boolean	**moveDown**(int x) Leaves the activity element and selects the element below.
boolean	**moveUp**(int x) Leaves the activity element and selects the element above.
void	**paint**(javax.microedition.lcdui.Graphics g, boolean selected) Paints the activity element

Constructor Detail

ActivityElement

```
public ActivityElement()
```

Constructor ActivityElement

Method Detail

elementLayoutManager

```
public void elementLayoutManager()
```

Updates the layout of the extended information object

Overrides:

`elementLayoutManager` in class `no.tellu.cdlc.gui.GroupElement`

---

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the activity element

Overrides:

`paint` in class `no.tellu.cdlc.gui.GroupElement`

---

fire

```
public boolean fire(int key)
```

Sets the extended information to be shown if it is not, hides it if it is already shown.

Overrides:

`fire` in class `no.tellu.cdlc.gui.GroupElement`

moveDown

```
public boolean moveDown(int x)
```

Leaves the activity element and selects the element below.

Overrides:

`moveDown` in class `no.tellu.cdlc.gui.GroupElement`

moveUp

```
public boolean moveUp(int x)
```

Leaves the activity element and selects the element above.

Overrides:

`moveUp` in class `no.tellu.cdlc.gui.GroupElement`

kindergartenform

Class DailyRoutineWindow

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiWindow
      └kindergartenform.DailyRoutineWindow
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class DailyRoutineWindow
extends no.tellu.cdlc.gui.GuiWindow
```

DailyRoutineWindow.java

Contains the information shown in the daily routine window.

Constructor Summary

DailyRoutineWindow(java.lang.String windowId, no.tellu.cdlc.gui.GuiListener controllerApp, java.util.Vector kindergartensInfo, boolean showIcons, FormWindow fw) Constructor DailyRoutineWindow

Method Summary

void	**callbackAction**(java.lang.String elementId, int type, java.lang.Object value) Shows the main window of the form when the 'Back' soft button is pressed.

Constructor Detail

DailyRoutineWindow

```
public DailyRoutineWindow(java.lang.String windowId,
                                  no.tellu.cdlc.gui.GuiListener controllerApp,
                                  java.util.Vector kindergartensInfo,
                                  boolean showIcons,
                                  FormWindow fw)
```

Constructor DailyRoutineWindow

Parameters:

`windowId` - the ID of the window

`controllerApp` - the controller application

`kindergartensInfo` - the information about the kindergartens

`showIcons` - whether or not to show the window with icons

`fw` - the form's main window

Method Detail

callbackAction

```
public void callbackAction(java.lang.String elementId,
                           int type,
                           java.lang.Object value)
```

Shows the main window of the form when the 'Back' soft button is pressed.

Specified by:

`callbackAction` in interface `no.tellu.cdlc.gui.ElementListner`

Overrides:

`callbackAction` in class `no.tellu.cdlc.gui.GuiWindow`

---

kindergartenform

Class FormGroupElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GroupElement
          └kindergartenform.FormGroupElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormGroupElement
extends no.tellu.cdlc.gui.GroupElement
```

FormGroupElement.java

Created to support functionality that GroupElement doesn't. This functionality is moving left and right within a RadioButtonElement, drawing elements in a grid layout, and, if a GroupElement has several selectable GUI elements, only marking one of them as selected.

Constructor Summary

FormGroupElement()   Constructor FormGroupElement
FormGroupElement(java.lang.String elementId)

Constructor FormGroupElement

Method Summary

`void`	`elementLayoutManager()` Handles the layout of the FormGroupElement.
`boolean`	`moveLeft(int x)` Allows moving left within a FormGroupElement.
`boolean`	`moveRight(int x)` Allows moving right within a FormGroupElement.
`void`	`paint(javax.microedition.lcdui.Graphics g, boolean selected)` Paints the group element.

Constructor Detail

FormGroupElement

```
public FormGroupElement()
```

Constructor FormGroupElement

FormGroupElement

```
public FormGroupElement(java.lang.String elementId)
```

Constructor FormGroupElement

Parameters:

`elementId` - the ID of the object of this class

Method Detail

elementLayoutManager

```
public void elementLayoutManager()
```

Handles the layout of the FormGroupElement.

Overrides:

`elementLayoutManager` in class `no.tellu.cdlc.gui.GroupElement`

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the group element.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GroupElement`

moveLeft

```
public boolean moveLeft(int x)
```

Allows moving left within a FormGroupElement.

Overrides:

`moveLeft` in class `no.tellu.cdlc.gui.GroupElement`

moveRight

```
public boolean moveRight(int x)
```

Allows moving right within a FormGroupElement.

Overrides:

`moveRight` in class `no.tellu.cdlc.gui.GroupElement`

---

kindergartenform

Class FormPageEight

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageEight
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageEight
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageEight.java

Contains the content of the eight page of the form.

Constructor Summary

`FormPageEight``(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr, FormWindow fw)` Constructor FormPageEight initialises the content to be shown.

Method Summary

`void`	**`leavePanel`**`()` Removes the 'Send' command when the user leaves the form page.
`void`	**`paint`**`(javax.microedition.lcdui.Graphics g, boolean selected)` Handles the commands to be shown on the menu line.

Constructor Detail

FormPageEight

```
public FormPageEight(java.lang.String elementId,
                     no.tellu.cdlc.gui.LayoutManager layoutMgr,
                     FormWindow fw)
```

Constructor FormPageEight initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class `layoutMgr` - the layout manager used in this form page

Method Detail

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Handles the commands to be shown on the menu line.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiPanel`

---

leavePanel

```
public void leavePanel()
```

Removes the 'Send' command when the user leaves the form page.

Overrides:

`leavePanel` in class `no.tellu.cdlc.gui.GuiPanel`

---

kindergartenform

Class FormPageFive

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageFive
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageFive
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageFive.java

Contains the content of the fifth page of the form.

Constructor Summary

`FormPageFive(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr)` Constructor FormPageFive initialises the content to be shown.

Constructor Detail

FormPageFive

```
public FormPageFive(java.lang.String elementId,
                    no.tellu.cdlc.gui.LayoutManager layoutMgr)
```

Constructor FormPageFive initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class
`layoutMgr` - the layout manager used in this form page

---

kindergartenform

Class FormPageFour

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageFour
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageFour
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageFour.java

Contains the content of the fourth page of the form.

Constructor Summary

FormPageFour(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr, java.util.Vector kindergartens) Constructor FormPageFour initialises the content to be shown.

Constructor Detail

FormPageFour

```
public FormPageFour(java.lang.String elementId,
                    no.tellu.cdlc.gui.LayoutManager layoutMgr,
                    java.util.Vector kindergartens)
```

Constructor FormPageFour initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class
`layoutMgr` - the layout manager used in this form page
`kindergartens` - the kindergartens of the form

---

kindergartenform

Class FormPageOne

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageOne
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageOne
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageOne.java

Contains the content of the first page of the form.

Constructor Summary

FormPageOne(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr) Constructor FormPageOne initialises the content to be shown.

Method Summary

boolean	**mapPresentation**() Find the selected presentation

Constructor Detail

FormPageOne

```
public FormPageOne(java.lang.String elementId,
                   no.tellu.cdlc.gui.LayoutManager layoutMgr)
```

Constructor FormPageOne initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class

`layoutMgr` - the layout manager used in this form page

Method Detail

mapPresentation

```
public boolean mapPresentation()
```

Finds the selected presentation

Returns:

true if map is selected, false if list is selected

kindergartenform

Class FormPageSeven

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
```

```
                  └ kindergartenform.FormPageSeven
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageSeven
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageSeven.java

Contains the content of the seventh page of the form.

Constructor Summary
FormPageSeven(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr, FormWindow fw) Constructor FormPageSeven initialises the content to be shown.

Method Summary	
void	**addSibling**() Adds a sibling to the form page.
void	**leavePanel**() Removes the command 'Add sibling' when the user leaves the form page.
void	**paint**(javax.microedition.lcdui.Graphics g, boolean selected) Handles the commands to be shown on the menu line.

Constructor Detail

FormPageSeven

```
public FormPageSeven(java.lang.String elementId,
                     no.tellu.cdlc.gui.LayoutManager layoutMgr,
                     FormWindow fw)
```

Constructor FormPageSeven initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class

`layoutMgr` - the layout manager used in this form page

Method Detail

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Handles the commands to be shown on the menu line.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiPanel`

leavePanel

```
public void leavePanel()
```

Removes the command 'Add sibling' when the user leaves the form page.

Overrides:

`leavePanel` in class `no.tellu.cdlc.gui.GuiPanel`

---

addSibling

```
public void addSibling()
```

Adds a sibling to the form page.

---

kindergartenform

Class FormPageSix

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageSix
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageSix
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageSix.java

Contains the content of the sixth page of the form.

Constructor Summary

FormPageSix(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr) Constructor FormPageSix Initialises the content to be shown.

Constructor Detail

FormPageSix

```
public FormPageSix(java.lang.String elementId,
                   no.tellu.cdlc.gui.LayoutManager layoutMgr)
```

Constructor FormPageSix Initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class

`layoutMgr` - the layout manager used in this form page

---

kindergartenform

Class FormPageThree

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageThree
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageThree
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageThree.java

Contains the content of the third page of the form.

Constructor Summary

FormPageThree(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr, java.util.Vector kindergartens, FormWindow fw) Constructor FormPageThree initialises the content to be shown.

Method Summary

boolean	**isIconSelected**() Checks if the user has chosen to see the daily routine with icons or numbers.
void	**leavePanel**() Removes the command 'See daily activities' when the user leaves the form page.
void	**paint**(javax.microedition.lcdui.Graphics g, boolean selected) Handles the commands to be shown on the menu line.

Constructor Detail

FormPageThree

```
public FormPageThree(java.lang.String elementId,
                     no.tellu.cdlc.gui.LayoutManager layoutMgr,
                     java.util.Vector kindergartens,
                     FormWindow fw)
```

Constructor FormPageThree initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class

`layoutMgr` - the layout manager used in this form page

`kindergartens` - the kindergartens of the form

Method Detail

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Handles the commands to be shown on the menu line.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiPanel`

---

leavePanel

```
public void leavePanel()
```

Removes the command 'See daily activities' when the user leaves the form page.

Overrides:

`leavePanel` in class `no.tellu.cdlc.gui.GuiPanel`

---

isIconSelected

```
public boolean isIconSelected()
```

Checks if the user has chosen to see the daily routine with icons or numbers.

Returns:

true if the user chose icons, false if not

---

kindergartenform

Class FormPageTwoList

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └kindergartenform.FormPageTwoList
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageTwoList
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageTwoList.java

Contains the content of the second page of the form, presented as a list.

Constructor Summary

`FormPageTwoList``(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr, java.util.Vector kindergartens)` Constructor FormPageTwoList initialises the content to be shown.

Constructor Detail

FormPageTwoList

```
public FormPageTwoList(java.lang.String elementId,
                       no.tellu.cdlc.gui.LayoutManager layoutMgr,
                       java.util.Vector kindergartens)
```

Constructor FormPageTwoList initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class

`layoutMgr` - the layout manager used in this form page

`kindergartens` - the kindergartens of the form

---

kindergartenform

Class FormPageTwoMap

```
java.lang.Object
  └ no.tellu.cdlc.gui.GuiElement
      └ no.tellu.cdlc.gui.GuiPanel
          └ kindergartenform.FormPageTwoMap
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormPageTwoMap
extends no.tellu.cdlc.gui.GuiPanel
```

FormPageTwoMap.java

Contains the content of the second page of the form, presented in a map.

Constructor Summary

`FormPageTwoMap``(java.lang.String elementId, no.tellu.cdlc.gui.LayoutManager layoutMgr, java.util.Vector kindergartens, FormWindow fw)` Constructor FormPageTwoMap Initialises the content to be shown.

Method Summary

`POIElement`	**`getPOIElement`**`()`
`void`	**`leavePanel`**`()` Removes the commands for choosing and regretting choice of kindergarten when the user leaves the form page.
`void`	**`paint`**`(javax.microedition.lcdui.Graphics g, boolean selected)` Handles the commands to be shown on the menu line.

Constructor Detail

FormPageTwoMap

```
public FormPageTwoMap(java.lang.String elementId,
                      no.tellu.cdlc.gui.LayoutManager layoutMgr,
                      java.util.Vector kindergartens,
                      FormWindow fw)
```

Constructor FormPageTwoMap Initialises the content to be shown.

Parameters:

`elementId` - the ID of the object of this class

`layoutMgr` - the layout manager used in this form page

`kindergartens` - the kindergartens of the form

Method Detail

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Handles the commands to be shown on the menu line.

Overrides:

`paint` in class `no.tellu.cdlc.gui.GuiPanel`

---

getPOIElement

```
public POIElement getPOIElement()
```

Returns:

the POIElement in the form page

---

leavePanel

```
public void leavePanel()
```

Removes the commands for choosing and regretting choice of kindergarten when the user leaves the form page.

Overrides:

`leavePanel` in class `no.tellu.cdlc.gui.GuiPanel`

---

kindergartenform

Class FormTabbedElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GuiPanel
          └no.tellu.cdlc.gui.TabbedElement
              └kindergartenform.FormTabbedElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormTabbedElement
extends no.tellu.cdlc.gui.TabbedElement
```

FormTabbedElement.java

Contains methods for changing, adding and removing form pages (panels)

Constructor Summary

FormTabbedElement`()` Constructor FormTabbedElement
FormTabbedElement`(java.lang.String elementId)` Constructor FormTabbedElement

Method Summary

`void`	**addTabElement**`(no.tellu.cdlc.gui.GuiElement element, int i)` Adds an element to a certain index of the panel vector.
`boolean`	**removeTabElement**`(int i)` Removes an element at a given index in the panel vector.
`void`	**setNextTab**`()` Sets the next form page to be shown.
`void`	**setPreviousTab**`()` Sets the previous form page to be shown.

Constructor Detail

FormTabbedElement

`public` **FormTabbedElement**`(java.lang.String elementId)`

Constructor FormTabbedElement

Parameters:

`elementId` - the ID of the object of this class

FormTabbedElement

`public` **FormTabbedElement**`()`

Constructor FormTabbedElement

Method Detail

setNextTab

`public void` **setNextTab**`()`

Sets the next form page to be shown.

setPreviousTab

`public void` **setPreviousTab**`()`

Sets the previous form page to be shown.

addTabElement

```
public void addTabElement(no.tellu.cdlc.gui.GuiElement element,
                          int i)
```

Adds an element to a certain index of the panel vector.

Parameters:

`element` - the element to be added

`i` - the index the element should be added at

removeTabElement

```
public boolean removeTabElement(int i)
```

Removes an element at a given index in the panel vector.

Parameters:

`i` - the index of the element that is to be removed

Returns:

true

kindergartenform

Class FormWindow

```
java.lang.Object
  └─no.tellu.cdlc.gui.GuiWindow
      └─kindergartenform.FormWindow
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class FormWindow
extends no.tellu.cdlc.gui.GuiWindow
```

FormWindow.java

Contains the data and pages of the kindergarten form application

Field Summary	
no.tellu.cdlc.serviceframe.messages.guicommands.GuiCmd	**activities**
no.tellu.cdlc.serviceframe.messages.guicommands.GuiCmd	**addSibling**
no.tellu.cdlc.serviceframe.messages.guicommands.GuiCmd	**choose**
no.tellu.cdlc.serviceframe.messages.guicommands.GuiCmd	**regret**
no.tellu.cdlc.serviceframe.messages.guicommands.GuiCmd	**send**

Constructor Summary
`FormWindow``(java.lang.String windowId, no.tellu.cdlc.gui.GuiListener controllerApp, KindergartenFormMIDlet kfm)` Constructor FormWindow

Method Summary	
`void`	**`callbackAction`**`(java.lang.String elementId, int type, java.lang.Object value)` Handles which actions to take according to the pressed soft button.

Constructor Detail

FormWindow

```
public FormWindow(java.lang.String windowId,
                  no.tellu.cdlc.gui.GuiListener controllerApp,
                  KindergartenFormMIDlet kfm)
```

Constructor FormWindow

Parameters:

`windowId` - the ID of the window object

`controllerApp` - the controller application

Method Detail

callbackAction

```
public void callbackAction(java.lang.String elementId,
                           int type,
                           java.lang.Object value)
```

Handles which actions to take according to the pressed soft button.

Specified by:

`callbackAction` in interface `no.tellu.cdlc.gui.ElementListner`

Overrides:

`callbackAction` in class `no.tellu.cdlc.gui.GuiWindow`

---

kindergartenform

Class KindergartenFormMIDlet

```
java.lang.Object
  └ javax.microedition.midlet.MIDlet
      └ kindergartenform.KindergartenFormMIDlet
```

```
public class KindergartenFormMIDlet
extends javax.microedition.midlet.MIDlet
```

KindergartenFormMIDlet.java The MIDlet class enables creating, starting and stopping the application.

Constructor Summary

`KindergartenFormMIDlet``()` Constructor KindergartenFormMIDlet

Constructor Detail

KindergartenFormMIDlet

```
public KindergartenFormMIDlet()
```

Constructor KindergartenFormMIDlet

---

kindergartenform

Class KindergartenPOIInfo

```
java.lang.Object
  └ framework.POIInformation
      └ kindergartenform.KindergartenPOIInfo
```

```
public class KindergartenPOIInfo
extends POIInformation
```

KindergartenPOIInfo.java

Contains kindergarten specificinformation. Activities have to be added by using addActivity(String timeString, String timeImage, String activity, String extendedInfo).

Constructor Summary

`KindergartenPOIInfo``(java.lang.String name, java.lang.String streetAddress, java.lang.String postCode, java.lang.String postAddress, java.lang.String district, java.lang.String owner, java.lang.String focusArea, java.lang.String openingHours, java.util.Vector placeTypes)` Constructor KindergartenPOIInfo

Method Summary

`void`	**`addActivity`**`(java.lang.String timeString, java.lang.String timeImage, java.lang.String activity, java.lang.String extendedInfo)` Adds an activity to the kindergarten.
`java.util.Vector`	**`getActivities`**`()`
`java.lang.String`	**`getDistrict`**`()`
`java.lang.String`	**`getFocusArea`**`()`

`java.lang.String`	**`getOpeningHours`**`()`
`java.lang.String`	**`getOwner`**`()`
`java.util.Vector`	**`getPlaceTypes`**`()`
`void`	**`setDistrict`**`(java.lang.String district)` Sets the district of the kindergarten.
`void`	**`setFocusArea`**`(java.lang.String focusArea)` Sets the focus area of the kindergarten.
`void`	**`setOpeningHours`**`(java.lang.String openingHours)` Sets the opening hours of the kindergarten.
`void`	**`setOwner`**`(java.lang.String owner)` Sets the ownership of the kindergarten.
`void`	**`setPlaceTypes`**`(java.util.Vector placeTypes)` Sets the place types of the kindergarten.

Methods inherited from class framework.POIInformation
`getExtraPopupInfo, getName, getPostAddress, getPostCode, getStreetAddress, setExtraPopupInfo, setName, setPostAddress, setPostCode, setStreetAddress`

Constructor Detail

KindergartenPOIInfo

```
public KindergartenPOIInfo(java.lang.String name,
                           java.lang.String streetAddress,
                           java.lang.String postCode,
                           java.lang.String postAddress,
                           java.lang.String district,
                           java.lang.String owner,
                           java.lang.String focusArea,
                           java.lang.String openingHours,
                           java.util.Vector placeTypes)
```

Constructor KindergartenPOIInfo

Parameters:

`name` - name of the kindergarten

`streetAddress` - street address of the kindergarten

`postCode` - post code of the kindergarten

`postAddress` - post address of the kindergarten

`district` - the district the kindergarten is situated in

`owner` - the owner of the kindergarten

`focusArea` - the focus area of the kindergarten

`openingHours` - the opening hours of the kindergarten

`placeTypes` - the place types of the kindergarten

Method Detail

getDistrict

```
public java.lang.String getDistrict()
```

Returns:

the district of the kindergarten

setDistrict

```
public void setDistrict(java.lang.String district)
```

Sets the district of the kindergarten.

Parameters:

`district` - the district of the kindergarten

getFocusArea

```
public java.lang.String getFocusArea()
```

Returns:

the focus area of the kindergarten

setFocusArea

```
public void setFocusArea(java.lang.String focusArea)
```

Sets the focus area of the kindergarten.

Parameters:

`focusArea` - the focus area of the kindergarten

getOpeningHours

```
public java.lang.String getOpeningHours()
```

Returns:

the opening hours of the kindergarten

setOpeningHours

```
public void setOpeningHours(java.lang.String openingHours)
```

Sets the opening hours of the kindergarten.

Parameters:

`openingHours` - the opening hours of the kindergarten

getOwner

```
public java.lang.String getOwner()
```

Returns:

the ownership of the kindergarten

setOwner

```
public void setOwner(java.lang.String owner)
```

Sets the ownership of the kindergarten.

Parameters:

`owner` - the ownership of the kindergarten

getPlaceTypes

```
public java.util.Vector getPlaceTypes()
```

Returns:

the place types of the kindergarten

setPlaceTypes

```
public void setPlaceTypes(java.util.Vector placeTypes)
```

Sets the place types of the kindergarten.

Parameters:

`placeTypes` - the place types of the kindergarten

addActivity

```
public void addActivity(java.lang.String timeString,
                        java.lang.String timeImage,
                        java.lang.String activity,
                        java.lang.String extendedInfo)
```

Adds an activity to the kindergarten.

Parameters:

`timeString` - the time represented by a string

`timeImage` - the time represented by an icon

`activity` - the name of the activity

`extendedInfo` - the description of the activity

getActivities

```
public java.util.Vector getActivities()
```

Returns:

the activities of the kindergarten

kindergartenform

Class SummaryElement

```
java.lang.Object
  └no.tellu.cdlc.gui.GuiElement
      └no.tellu.cdlc.gui.GroupElement
          └kindergartenform.SummaryElement
```

All Implemented Interfaces:

no.tellu.cdlc.gui.ElementListner

```
public class SummaryElement
extends no.tellu.cdlc.gui.GroupElement
```

SummaryElement.java

Draws a summary element in a list of main topics. The element consists of the name of the main topic and an information icon.

Pressing fire displays the summarisation of the information given in the main topic.

Constructor Summary
SummaryElement() Constructor SummaryElement

Method Summary

void	**elementLayoutManager**() Updates the summary vector
boolean	**fire**(int key) Sets the extended information to be shown if it is not, hidden otherwise.
boolean	**moveDown**(int x) Leaves the summary element and selects the element below.
boolean	**moveUp**(int x) Leaves the summary element and selects the element above.
void	**paint**(javax.microedition.lcdui.Graphics g, boolean selected) Paints the summary element

Constructor Detail

SummaryElement

```
public SummaryElement()
```

Constructor SummaryElement

Method Detail

elementLayoutManager

```
public void elementLayoutManager()
```

Updates the summary vector

Overrides:

`elementLayoutManager` in class `no.tellu.cdlc.gui.GroupElement`

---

paint

```
public void paint(javax.microedition.lcdui.Graphics g,
                  boolean selected)
```

Paints the summary element

Overrides:

`paint` in class `no.tellu.cdlc.gui.GroupElement`

---

fire

```
public boolean fire(int key)
```

Sets the extended information to be shown if it is not, hidden otherwise.

Overrides:

`fire` in class `no.tellu.cdlc.gui.GroupElement`

---

moveDown

```
public boolean moveDown(int x)
```

Leaves the summary element and selects the element below.

Overrides:

`moveDown` in class `no.tellu.cdlc.gui.GroupElement`

---

moveUp

```
public boolean moveUp(int x)
```

Leaves the summary element and selects the element above.

Overrides:

`moveUp` in class `no.tellu.cdlc.gui.GroupElement`

Appendix C: Evaluation Results

Age:									
User 1	**User 2**	**User 3**	**User 4**	**User 5**	**User 6**	**User 7**	**User 8**	**User 9**	**User 10**
25	28	26	24	22	38	39	45	50	49

Table C1. Age of the users

Time and place:	
User 1	15 May, 2007, Gløshaugen
User 2	15 May, 2007, Gløshaugen
User 3	15 May, 2007, Gløshaugen
User 4	15 May, 2007, Gløshaugen
User 5	15 May, 2007, Gløshaugen
User 6	15 May, 2007, Brøttem
User 7	15 May, 2007, Brøttem
User 8	16 May, 2007, Flatåsen
User 9	16 May, 2007, Enge
User 10	16 May, 2007, Enge

Table C2. Time and place of the evaluations.

Experience with mobile phones: Calling, messages / Other experience (specify)	
User 1	Radio, MP3, calendar.
User 2	Calendar.
User 3	Use of several programmes, calendar, text editing programmes.
User 4	Radio, MP3, calendar.
User 5	Calendar.
User 6	Calling, messages.
User 7	Calling, messages.
User 8	Calling, messages.
User 9	MP3
User 10	Calendar

Table C3. Experience with mobile phones.

Do you think you would use your mobile phone for filling in applications, registration forms and reports?	
User 1	She does not think she would use it for filling in the tax return, but thinks it can be beneficial for small forms.
User 2	Yes, but they must not be too large.
User 3	Generally not, prefers to use computers.
User 4	No, he thinks it is too awkward, prefers to use a computer.
User 5	Yes, she thinks it is ”genious”.
User 6	Probably not, depends very much on screen size. She thinks she may have been more positive if she had been accustomed to use her mobile phone for other tasks than calling and sending messages.
User 7	No.
User 8	No, or at least not when she has access to a computer.
User 9	No.
User 10	Yes.

Table C4. General question regarding feasability of forms on mobile phones.

If the answer is no: Why not?	
User 3	It is too hard to enter input.
User 4	They have too small screens.
User 6	The screen of her own phone is too small.
User 7	It is too awkward because of the size of the device, particularly because of the small screen.
User 8	They have too small screens, and it is too awkward to enter input.
User 9	Thinks he is too old or old fashioned to learn how to interact with these user interfaces. He prefers computers.

Table C5. Follow-up question, why not feasible.

If the answer is yes: In which situations do you think you would be using your phone for such tasks, rather than using a computer or the paper versions?	
User 1	When she does not have access to a computer with Internet. If it is not an important form, she states that she could fill it in at any place.
User 2	When he does not have access to a computer with Internet.
User 5	When she does not have access to a computer, or when she realises that she has forgotten to fill in a form in the last minute.
User 10	Does not know.

Table C6. Follow-up question, in which situations.

What do you think of the size of this form?	
User 1	It is okay, the form contains everything necessary, and it was no problem to complete it.
User 2	It was a bit too large
User 3	It is okay.
User 4	It is okay.
User 5	It was okay, should be that much in order to contain all of the necessary content.
User 6	It is fine, the form was surveyable and good.
User 7	It has too many questions, he would not have used his mobile phone for filling in this form.
User 8	It is okay. Says it would probably feel a lot shorter if she had been used to use her mobile phone for other things than calling and sending messages, since she would have used less time on figuring out how the user interface works.
User 9	It has an okay size, not too big.
User 10	Thinks it is great, but is worried about how the form would appear on her own phone, which has a much smaller screen.

Table C7. How they found the size of the form.

If the answer is that the form is too big or similar: At what point would you say that you had had enough?	
User 2	Does not know.
User 7	He would only have used mobile phones for filling in forms where he need to confirm "Yes"/"No".

Table C8. Follow-up questions regarding the size of the form.

If the answer is that the form is too big or similar: Why do you think you felt that it was too big?	
User 2	It would be more comfortable to fill in the form on a computer. The size would have felt better if the technique of allowing the users to choose which information they want to see, had been used more.
User 7	He would only have used mobile phones for filling in forms where he need to confirm "Yes"/"No".

Table C9. Follow-up questions regarding the size of the form.

Go to the last form page. When you activated one of these elements, a lot of information appeared. Did you like that you had the possibility of hiding and revealing information, or would you rather have had all of the information presented at once?	
User 1	She likes best that she can choose what she wants to see.
User 2	Yes, he really liked this possibility.
User 3	He liked it, he claims that it was a very neat, tidy way of doing it.
User 4	He thinks it is a quite okay solution.
User 5	Yes, it is good to be able to choose, plus it helped her keep focus.
User 6	Yes, it was easier to focus on what she was reading.
User 7	Yes, it was good to have that possibility.
User 8	Yes, it was best to be able to choose.
User 9	Yes, it is nice to be able to choose in this way, particularly as it reduces the scrolling.
User 10	Yes.

Table C10. Hiding and revealing information technique.

Show the user the first form page, which has three drop-down boxes. Did you dislike that you could not see more than one alternative at first, or think it was confusing?	
User 1	No, that was okay. That is what is common, is it not?
User 2	No, it was okay.
User 3	No, it was just as usual.
User 4	No, there was no problem.
User 5	No, it was quite alright.
User 6	No. The drop-downs must be very beneficial on the small screens of mobile phones, rather than showing a list with all alternatives – they make the page more surveyable, she said.
User 7	No, it was okay.
User 8	No, it was quite alright. She understood that she could see more alternatives by opening the list.
User 9	No, these were quite alright.
User 10	No, they were very easy to use.

Table C11. One alternative visible in drop-down boxes.

Ask the user to look at the page, and then activate one of the drop-downs. Would you say that you became confused or annoyed when the layout of the form suddenly changed, and elements that you earlier saw were gone?	
User 1	No, she thinks it helped her focus on what she was doing.
User 2	No, it was no problem.
User 3	No, the frame surrounding the drop-down made it very neat. He thinks it better that the elements below move, than that the drop-down is drawn over them.
User 4	No, but he prefers that drop-down boxes are drawn over the other screen elements.
User 5	No, she thinks it was good that they were moved downwards.
User 6	No.
User 7	No, but he would have preferred seeing only the content of the current selected drop-down box.
User 8	No, that was no problem.
User 9	No, no problems there.
User 10	No.

Table C12. Layout change in drop-down boxes.

From the first form page, go to the second with map presentation and select a kindergarten. Did it take you long to notice that the right button had a different command when you first saw this?	
User 1	She did not notice until she had regretted the choice of a kindergarten. Before that, she used the fire button.
User 2	He used the fire button to choose kindergartens, and did not notice the command change

	until he was going to cancel a choice.
User 3	He did not notice it, he used the fire button to choose and cancel choices.
User 4	He did not notice until he was going to cancel. He was using the fire button for choosing kindergartens.
User 5	She used the fire button, but noticed when she were going to cancel a choice.
User 6	She noticed immediately, and used the right soft button to choose and cancel choice of kindergartens.
User 7	He used the fire button to choose the first kindergarten, but then discovered the command on the menu line.
User 8	She did not notice until she was going to use the 'Next' command. She used the fire button both to choose and cancel kindergartens.
User 9	He did not notice, he used the fire button both for choosing and cancelling.
User 10	She used the fire button first, but noticed when she was going to cancel.

Table C13. Change of commands on the right soft button.

What did you think had happened to the 'Next' command?	
User 1	She did not think this page had that command. She said that, as this was only the second form page, she had no reason to assume that every page had a 'Next' command. She figured that she could navigate between pages using the right/left keys.
User 2	He did not really consider it, because he changed pages with the right/left keys. He finally concluded that the command did not exist on this page.
User 3	Does not know. He consistently navigated with the right/left keys between form pages.
User 4	Says he never got to consider it, as he used the right/left keys for navigating between form pages.
User 5	She navigated with right/left keys, but assumed it could be found in the menu.
User 6	She saw no reason for the page to have that command, and navigated with the right/left keys.
User 7	That it could be found in the menu.
User 8	That it could be found on the menu.
User 9	He changed form pages with the right/left keys. He had not noticed the missing command, but assumed it could be found in the menu.
User 10	That it could be found in the menu.

Table C14. What happened to the disappeared 'Next' command.

Go to the third form page and select the top "See daily routine with numbers or clock icons" group element. Ask the user to look at the page, then choose the group element below. Did you see that the 'Next' command came back, when you first filled in the form?	
User 1	She did not notice this command until she needed it, she was focused on the task at hand.
User 2	Yes.
User 3	No, says he did not look much at the menu line. He navigated with right/left keys.
User 4	He did not notice.
User 5	No, not until she needed it.
User 6	No, she did not notice until she reached the bottom of the page.
User 7	No, not until he needed it.
User 8	No, not until she needed it. Says she looked at those commands only when she did not find what she needed in the form, or when she was going to the next step.
User 9	Did not notice. Said he was focused on the task at hand at any time, and that he did not look at that command if he found what he needed in the form.
User 10	Did not notice it until she needed it.

Table C15. The appearing 'Next' command.

Did you think that the 'See daily routine' command was connected only to the "See daily routine with numbers or clock icons" group elements?	
User 1	Yes.

User 2	Yes.
User 3	He did not consider it.
User 4	He did not consider it.
User 5	Yes.
User 6	Yes.
User 7	Yes.
User 8	She had not considered it, but it seemed reasonable.
User 9	Yes.
User 10	She did not consider it.

Table C16. Command connected to a specific element on the screen.

What do the portfolio tabs mean to you?	
User 1	Steps in the form-filling process.
User 2	Steps in the form-filling process.
User 3	Steps in the form-filling process, particularly because of the numbers.
User 4	Steps in the form-filling process.
User 5	Steps in the form-filling process.
User 6	Steps in the form-filling process.
User 7	Steps in the form-filling process.
User 8	Steps in the form-filling process.
User 9	Steps of the form pages, progression.
User 10	The pages of the form.

Table C17. Portfolio tabs.

Go back to the second form page with map, and ask the user to study the page. Then present the second form page with kindergartens in a list, and ask the user to explore it. Which of the two representations did you like best, and why?	
User 1	She would have had a combination of the two, with kindergartens in a list and the possibility of seeing each kindergarten's position in a map.
User 2	The map, it was great to get an overview of the positions of the kindergartens.
User 3	Had he really been going to fill in a kindergarten application, he would have gotten information about the kindergartens before starting the form-filling process. He states this to be the reason to why he prefers the list. He also thinks that the map is beneficial for orientation.
User 4	The map, because it was easier to see where the kindergartens are for those who do not know every street name and number.
User 5	The map, because it was good to get an overview of the kindergartens.
User 6	The map, it was most practical, as one could get an overview of the positions of the kindergartens.
User 7	The map, provided that it is not dim. It gives an overviw ofthe kindergartens' positions.
User 8	The map, because it gave an overview over the kindergarten's positions.
User 9	The map, because it gives an overview of the kindergartens' positions.
User 10	She has no favourite. She liked the map because of the possibility of seeing the kindergartens' positions, and the lis because of the possibility of seeing several kindergartens' adresses in text at the same time.

Table C18. Map and list representation of kindergartens.

Go to the daily routine window with clock icons, and ask the user to study it. Then present the daily routine with points of time represented as numbers. Which of the two representations did you like best, and why?	
User 1	The numbers, they were most surveyable. She found the clock icons a bit hard to perceive the content of.
User 2	The numbers, they were easiest to perceive the meaning of. He said that the clock icons could be easier to interpret as actual points of time, if the clock icon of the selected activity had been enlarged. This could increase the readability of the icon and make it stand out.

User 3	The numbers, they were easiest to understand.
User 4	The numbers, the clock icons were too dim. Says that at first, he did not understand that the icons actually meant something, he just thought of them as decorations.
User 5	The numbers, because it was easiest to see the points of time with them. The clock icons were too small.
User 6	The numbers, they were easier to read. The clock icons were too small.
User 7	Numbers, those can not be misunderstood. The clock icons were too small and dim.
User 8	The numbers, because they were easier to perceive the content of.
User 9	The numbers, they were easier to perceive. The clock icons were too dim. He also states that he perceive numbers better than icons, regarding points of time.
User 10	The numbers. The clock icons were too dim.

Table C19. Points of time represented graphically and with numbers.

Do you have anything you would like to remark on?	
User 1	There should have been more information about the kindergartens, for instance placed in the window with the daily routine.
User 2	He thinks it should be possible to change the level of detail in the map and to navigate in it with the keys for moving right/left/up/down.
User 3	He expected to be able to navigate with right/left/up/down keys in the map. Form page 3 was not as surveyable as the others, there were a bit too much information. He thinks this page would have benefited from use of the hiding/revealing information technique.
User 4	No.
User 5	It was beneficial that the form had pre-filling.
User 6	She thinks it is easiest to navigate between form pages only with the right/left keys. It is nice to have the opportunity of filling in forms any place, any time.
User 7	No.
User 8	It was nice being able to choose how kindergartens and clock icons should be presented.
User 9	The selected group element at any time should be made more perceivable, e.g. by a different background colour.
User 10	No.

Table C20. Remarks.