



Norwegian University of
Science and Technology

Design and Evaluation of a User- Centric Information System

Enhancing Student Life with Mobile Computing

Sindre Paulsrud Moe

Master of Science in Computer Science

Submission date: June 2009

Supervisor: John Krogstie, IDI

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

Problem Description

To get timely information to students at the Norwegian University of Science and Technology (NTNU), the project is to develop and evaluate a mobile system providing user-centric information services. Although expected to be of most use to new students, it should also contain services believed to be useful for more senior students.

A fully functional prototype, named "MSIS", was developed last autumn. The services offered by MSIS include a search tool for rooms and buildings around campus, with integrated map and geographical positioning, a dynamic schedule service providing up-to-date information of weekly lectures, and a news and announcements service. This project should perform a user-driven evaluation of the existing prototype for getting ideas for a functional service. Based on the feedback from the evaluation, a new version of the system should be made, and evaluated. In addition to the user-evaluation, the service should be evaluated relative to other existing location and context-oriented services.

The project should be carried out in accordance with the design-science research model, and should be related to the work of the Wireless Trondheim Living Lab, a member of ENOLL (<http://www.openlivinglabs.eu/>). The report is expected to be written in English.

Assignment given: 15. January 2009
Supervisor: John Krogstie, IDI

Abstract

This project is a continuation of the work carried out autumn 2008 by Moe [56], which reviewed the digital communication channels currently used for distribution of student information at the Norwegian University of Science and Technology (NTNU), and defined the key design decisions for a mobile service called MSIS. The project proposes a new mobile computer system (MSIS) intended to make user-centric information more easily available to students at NTNU. The system is designed using a Service Oriented Architecture (SOA), providing a number of services which offers functionality such as a dynamic course schedule, and a location search tool. Furthermore, MSIS makes use of context-awareness and elements of mobile computing, in order to provide a service that dynamically adapts to the situation of the user. A geographical positioning module based on Wi-Fi location fingerprinting technology is described, which makes it possible to determine the position of a handheld device within existing wireless network infrastructure. The project has been carried out in accordance with the design-science research model over a number of implementation and evaluation iterations.

A user-driven evaluation of the MSIS service has been conducted among a group of NTNU students. The utility and usability of the system were evaluated by applying observational and empirical evaluation methods in a real-world environment on campus. The user tests identified numerous issues with the initial design, and suggested ideas for enhancements which have been implemented in the final version of the system. The Mobile Service Acceptance Model (MSAM) has been used to examine the factors that are influential for user adoption of mobile services in light of our project. The MSAM instrument measures different facets of a mobile information service, such as the perceived usefulness, ease of use, and usage intention. Our findings confirm that the utility of the MSIS system is perceived as very high, and students would likely benefit from such a system. There is no doubt great potential for a service like MSIS, and it is believed to be a useful addition to existing systems.

Preface

This report is a documentation of the project work performed as part of the Master's thesis in Computer Science spring 2009 by Sindre Paulsrud Moe. The project is carried out in the 10th semester, and marks the end of the 5-year long Master of Science education in Computer Science at The Norwegian University of Science and Technology (NTNU). The scope of the project amounts to 30 units.

The project has been defined in consultation with my supervisor, professor John Krogstie at the Department of Computer and Information Science (IDI). The project describes an information system comprehending geographical positioning, context-awareness, and mobile computing in an effort to enhance daily life for students at NTNU. In spite of much hard work, and a few intermediate challenges, it has been an extremely instructive and rewarding process, both academically and for me personally. The fact that the project has received quite a lot of positive attention, and addresses a genuine problem has been a great motivational factor.

I would like to express my deepest gratitude to my supervisor, John Krogstie, for invaluable feedback and advice during the course of this project. I would also like to thank Shang Gao at the Department of Computer and Information Science for his assistance, and all the students who were kind enough to volunteer in our research study. Lastly, I would like to thank my parents for their continuous support and encouragement throughout the years as a student in Trondheim.

Trondheim, June 9, 2009

Sindre Paulsrud Moe

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Project scope	2
1.3	Project description and context	2
1.4	Problem definition	2
1.5	Report outline	3
2	Background	5
2.1	Preliminary study	5
2.1.1	Summary of the survey	6
2.1.2	Technology criteria	8
2.2	Description of the MSIS system	9
2.2.1	Introduction	9
2.2.2	Functionality	10
2.2.3	Architecture and implementation	11
2.2.4	Screenshots	13
2.3	Similar work	14
3	Research approach	17
3.1	Research questions	17
3.2	Research method	17
3.2.1	The design-science paradigm	18
3.3	Usability and user acceptance testing	20
3.3.1	What is usability?	20
3.3.2	Principles of usability design	22
3.3.3	How to measure usability?	24
3.3.4	Addressing usability in mobile applications	26
3.3.5	Technology acceptance models and theories	28
3.3.6	The measurement constructs	32
4	Evaluation	35
4.1	Introduction	35
4.2	The usability test	36
4.3	The user acceptance survey	37

4.4	Presentation of the test results	39
4.4.1	Usability test	39
4.4.2	Descriptive analysis of survey feedback	46
4.4.3	Statistical analysis of the user acceptance survey	53
4.5	Validity of the results	56
4.6	Summary	57
4.6.1	Directions for next iteration	58
5	Presentation of the final system	61
5.1	New functionality	61
5.1.1	Contextual location search	61
5.1.2	Improved schedule service	64
5.1.3	Desktop version of location search service	65
5.2	Better exception handling	66
5.3	Visual enhancements	67
5.4	Other enhancements	67
5.5	Evaluation of the final system	67
5.6	Video demonstration of the system	68
6	Design and implementation details	69
6.1	Design goals	69
6.2	The web services	70
6.2.1	UserService	70
6.2.2	LocationService	71
6.2.3	ScheduleService	71
6.2.4	AnnouncementService	72
6.3	Geographical positioning of users	72
6.3.1	General	72
6.3.2	The offline phase	73
6.3.3	The online phase	73
6.3.4	Advantages	75
6.3.5	Limitations	75
6.4	Implementation example	75
6.4.1	Log in	76
6.4.2	Search for locations	76
6.5	Database design - revised	78
6.6	Third-party components	80
6.6.1	OpenNETCF Smart Device Framework	80
6.6.2	ASP.NET AJAX Control Toolkit	80
7	Conclusions and further work	81
7.1	Contributions	81
7.2	Main findings	82
7.2.1	Research questions: retrospection	83
7.3	Further work	83

7.3.1	Limitations	85
A	Usability scenarios (in Norwegian)	1
B	User acceptance survey	1
C	Deployment instructions	1
C.1	The backend servers	1
C.1.1	Database server	1
C.1.2	Application server	2
C.2	Client application	4
C.3	Development environment	5
D	Digital attachments	1
D.1	Source code	1
D.2	Database	1
D.3	MSIS CAB installer	1
D.4	Video demonstration	1

List of Figures

2.1	Students' satisfaction with current communication channels	7
2.2	Mobile services deemed useful by the respondents	8
2.3	Functional overview	11
2.4	The MSIS architecture	12
2.5	Screenshots of MSIS	13
3.1	The general methodology of design-science	18
3.2	The IS Diffusion Variance Model [3, 21, 22]	29
3.3	The Technology Acceptance Model from Davis et al. [24]	30
3.4	The Unified Theory of Acceptance and Use of Technology [80]	31
3.5	The Mobile Services Acceptance Model [32]	32
4.1	Test results - location search service	40
4.2	Test results - geographical positioning module	41
4.3	Ajax-powered MSIS Web Administration Interface	43
4.4	Suggestions for new features to the schedule service	44
4.5	Perceived Usefulness vs. Intention to Use	51
4.6	Perceived Ease of Use vs. Intention to Use	52
4.7	Junior vs. senior students	53
4.8	Computer science students vs. non-comp. science students	53
4.9	The structural measurement model	55
5.1	Contextual Location Search	62
5.2	Contextual Location Search - Map	63
5.3	Support for courses outside study program	64
5.4	The improved Schedule Service	65
5.5	Web-based Location Search Service	66
5.6	New main menu and configuration screen	68
6.1	Data representation of the location fingerprints	73
6.2	Location fingerprinting - online phase	74
6.3	Old Database ER Diagram	78
6.4	Revised Database ER Diagram	79
A.1	Kart	2

C.1	Enable TCP/IP for the database server	2
C.2	Create new Virtual Directory	3
C.3	Application configuration file	4

List of Tables

3.1	Table from [41]: Design-Science Research Guidelines	19
4.1	Frequency Results - Perceived Usefulness	47
4.2	Frequency Results - Perceived Ease of Use	48
4.3	Frequency Results - Trust	48
4.4	Frequency Results - Personal Initiatives & Characteristics	49
4.5	Frequency Results - Context	50
4.6	Frequency Results - Intention To Use	51
4.7	Correlations between constructs	54
4.8	ICR analysis - reliability coefficients	57

Chapter 1

Introduction

“There is nothing more difficult to take in hand, more perilous to conduct or more uncertain in its success than to take the lead in the introduction of a new order of things.”

–Niccolo Machiavelli

1.1 Motivation

Examining the results from the project carried out last autumn [56], there is a clear indication that the students at NTNU are positive to a mobile user-centric information service, such as the MSIS system described in [56]. Thus, we feel there is a need to conduct further research on this subject.

The fact that the project addresses a genuine problem is a great motivational factor for developing such a system, in an effort to bring timely information to the students by making use of mobile technology and wireless networking. By using a mobile phone or PDA, students can easily keep track of their lecture schedule, or get a map description of a particular room or building. While the concept was proven in the first prototype, we still needed to conduct a thorough usability study among the prospective target group to assess the applicability, perceived usefulness, and ease of use of the system. This will allow us to draw conclusions regarding potential user adoption of the mobile service. A new version of the system will be developed, incorporating user feedback and findings from the evaluation, ultimately leading to a better service.

1.2 Project scope

In the project last autumn, an extensive prestudy was conducted to explore the need for a user-centric mobile service. In addition, a number of similar solutions were reviewed. A fully functional prototype was proposed, based on ideas gathered from the students. The work described in this project builds further upon this foundation. As such, the two projects should be seen as a whole, both contributing towards a final goal; to research the need for a user-centric mobile information system, evaluate the prototype in accordance with a set of usability metrics, and finally apply the findings from the evaluation in order to improve the system.

When dealing with location based services and geographical positioning of users, one has to be careful not to infringe upon the privacy of the users. However, this is not as important in this project because the data is not shared between the users. We will not touch on these subjects in this project, other than state the importance of taking these considerations before the system is put into production.

1.3 Project description and context

The purpose of this project is to examine whether the assumptions made regarding the viability of a new mobile student information service are correct, by testing a prototype of the system on a group of potential users. This evaluation will establish whether or not our system is capable of fulfilling its intended purpose. The Mobile Services Acceptance Model [32] will be used to measure metrics such as perceived usefulness, perceived ease of use, and intention to use. Furthermore, issues pertaining to trust, context, and personal initiatives will be considered as well.

In the concluding chapters of the previous report [56], we gave suggestions for further work and also noted which functional requirements remain to be implemented. We will use this information, together with feedback from the user tests, and the requirements specification previously derived to further develop the system.

1.4 Problem definition

The goal of this project is to design and evaluate an information system for enhancing daily life of students at the Norwegian University of Science and Technology. We will investigate to which extent user-centric mobile services combined with context-awareness can improve upon existing information systems, and finally, explore the students' attitudes toward such a system in order to establish a measure of user acceptance.

1.5 Report outline

Below is a short summary of the different chapters that constitute this document.

Chapter 1 Introduction gives an introduction to the project by presenting the task at hand, the motivation for choosing the project, and a description of how we will approach the problem.

Chapter 2 Background presents the main findings of the preliminary study which formed the basis for this project, and describes the initial prototype of the MSIS system.

Chapter 3 Research approach states the questions we seek to answer during the work with this project, and describes the research methods and models used.

Chapter 4 Evaluation describes the usability test and user acceptance survey conducted as part of the evaluation of the system. The findings from the evaluation are presented, together with suggestions for improvements of the service.

Chapter 5 Presentation of the final system contains a description of the new features implemented in the revised version of MSIS, how they have been realized, and to which extent the enhancements were successful in addressing the issues identified in the evaluation phase.

Chapter 6 Design and implementation details describes the technical details of the implementation, how the various services operate, and explains the technology behind the system. The chapter also gives an example of how the web services can be applied to develop a location search tool.

Chapter 7 Conclusions and further work marks the end of the report by giving some concluding thoughts about the outcome of the project and the achievements made. We also discuss whether the work with this project has provided sufficient results in order to answer the research questions.

Chapter 2

Background

“I find that a great part of the information I have was acquired by looking up something and finding something else on the way.”

–Franklin P. Jones

This chapter constitutes the theoretical fundament that this project is built upon. It provides insight into state-of-the-art subjects, theories, and models that are further elaborated throughout the report. Since an extensive preliminary study was conducted as part of the project last autumn, some parts of this chapter are based on the material from the other report. As mentioned in section 1.2, the two projects should be seen as a whole, and the previous work is therefore inevitable in that regard. The main findings from our preliminary study are presented in section 2.1. Section 2.2 gives a brief description of the initial MSIS prototype from a technical and functional perspective, while previous research work related to our project is discussed in section 2.3.

2.1 Preliminary study

In autumn 2008, this project was started as part of the author’s depth study. The goal of the project was to evaluate the current channels for communication of information to students at NTNU, and the availability of such information. A survey was conducted to identify problems and flaws with the current information services based on feedback gathered from students. The findings from the survey showed that the majority of the respondents were somewhat dissatisfied with the current information services. In an attempt to solve these problems a new information system called the Mobile Student Information System (MSIS) was proposed. A requirements analysis was conducted followed by the implementation of a working prototype. In brief, MSIS provides a service which allow users to search for rooms and buildings on NTNU campus, and a schedule service which can be used to keep track of weekly lectures.

2.1.1 Summary of the survey

An electronic questionnaire was created to enable rapid distribution and computer aided analysis of the data gathered from the respondents. The target group consisted of 89 students majoring in engineering. The questions concerned how information currently is organized and communicated to the students, and what services they would like to see in a mobile student information system. We also looked at how the results were dispersed by semester and area of study. The decision of which services to include in MSIS was largely influenced by the feedback gathered from the survey, which also depicted the functional requirements of the system.

The general perception is that there is a great need for new information services and improvements in this area. Many students feel that it is hard to find the information they are looking for, because information is spread across different systems, websites, and portals, and often the information is too generic instead of being tailor to better accommodate the individual needs of each student. Therefore, it is a desire to consolidate the information from the multitude of channels into a single digital information service. Furthermore, the application should run on handheld devices to make it easily accessible. The application should be user-centered, meaning that it should adapt dynamically to the situation of the student, by taking into account aspects such as the current semester of the student, which courses and area of study the student is attending, and so forth. This would allow students to more easily obtain what they are looking for without having to search through a wealth of information. A user profile should be assigned to each student to capture their personal preferences. The user profile should be password protected to ensure that the security is maintained.

In addition to information specified in the user profile, the system should also consider contextual information. Such systems are often referred to as context-aware systems or applications. As people more and more turn into mobile workers, it becomes increasingly important to have software that can adapt dynamically to changes in the environment. Context-aware applications try to make assumptions about the user's current situation, and provide functionality and information that comprehend the immediate surroundings. Common aspects of context are location, actions (i.e. what the user is doing), and which people you are with [71]. Abowd et al. [2] defines context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*".

The MSIS service provides a geographical positioning module which can be used to add *location-awareness* to the application. For example, by applying knowledge about the user's current whereabouts, the application may offer suggestions for nearby locations of interest, such as a computer lab, a meeting room, or a canteen. The prototype offers a tool that denotes the user's current position on a map, together with a description of the location where the user is situated. *Time* is another element of context which we make use of. The schedule service combines information in the user's profile (courses and study program) with time to generate a time table for the present week. The idea of combining contextual information from various sources is known as context-fusion.

Figure 2.1 shows the students' satisfaction with the current information services, sorted according to year/semester. One interesting thing to note is that the senior students seem to be less satisfied with the communication channels than first and second grade students. A possible reason for this finding could be that senior students have been around longer, and therefore have

a better understanding of the information systems in use. It could also be that the situation has improved since the senior students first started, and efforts have since been made to solve some of the issues.

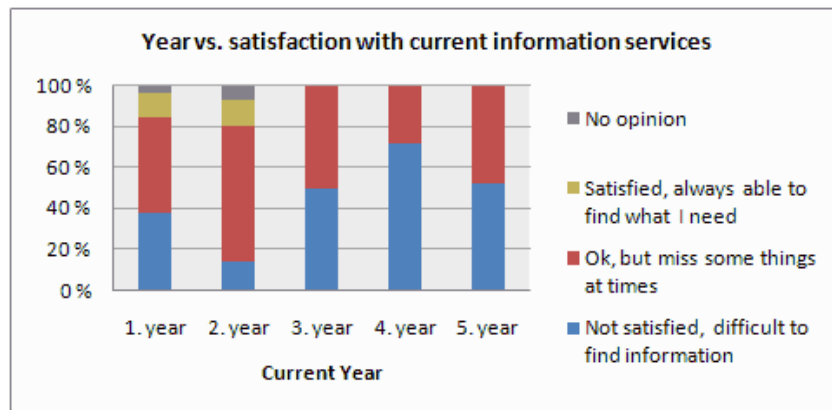


Figure 2.1: Students' satisfaction with current communication channels

During the work with the survey it was revealed that close to 30% of the respondents claimed they had at least once missed a deadline due to lack of information. This is quite an alarming signal, although we do not have scientific proof that the current information services are contributing to these numbers. To quote one of the respondents:

“The information is inaccurate and often comes too late, in the form of last minute emails or meetings without the proper announcements upfront.”

Another aspect we wanted to look into was whether people would benefit from a mobile location service that would allow them to search for rooms and buildings on campus. Our assumption is that the campus area can be overwhelming, and often it can be challenging to find a particular room number or lecture hall – especially for junior students. Although, even long-time employees have expressed their interest in such a service, so the area of application might be quite large. About half of the respondents were very positive to a location search service for mobile devices, while close to 40% would maybe use a service like this. The big advantage of mobile services is that they are available anytime and anywhere where there is wireless network coverage. Currently, the coverage on campus is close to 100% indoors, and large outdoor areas are covered as well. Traditionally, to look up information you would first have to locate a computer to find what you are searching for, or alternatively contact the student help desk. By offering access to these useful services on a mobile phone, you never have to look further than to your own pocket. Our prior research [56] confirms that most of the students today are familiar with using the mobile phone for other activities besides making phone calls, so we believe they could easily adapt a new mobile service. This will be tested as part of this project.

The rest of the survey concerned other potential services which should be supported by MSIS. As Figure 2.2 depicts, the most requested services were a course lecture planner and a notification service. These two services, in addition to the location service, were included in the initial design of the prototype. One important design consideration was that the system should be based on a

flexible framework. Thus, facilitating development of client applications for different software platforms, and allowing extensions to the system in the future.

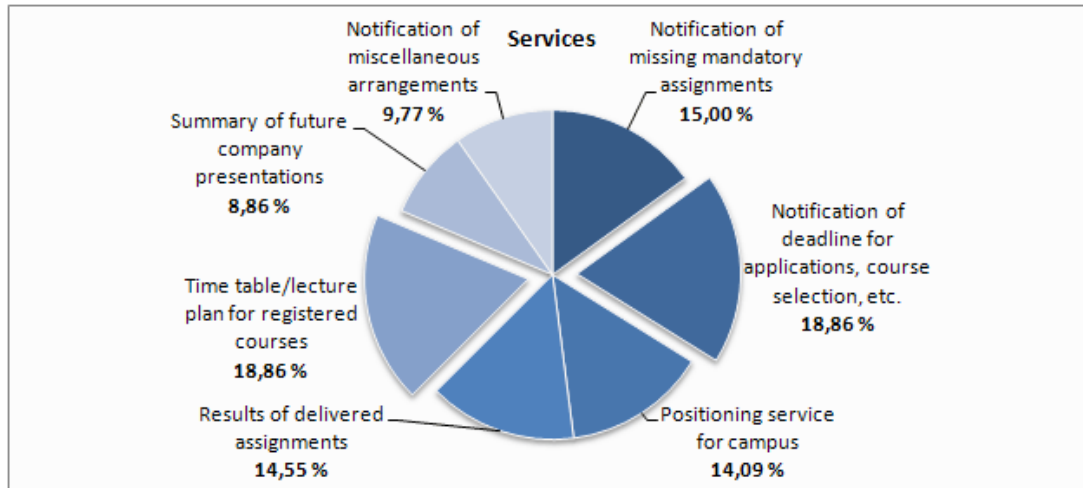


Figure 2.2: Mobile services deemed useful by the respondents

2.1.2 Technology criteria

Once it was clear that we were going to develop a mobile application, we started to research available technologies and programming frameworks suitable for the task at hand. It was important to decide on a technology and platform that would allow us to fulfill the functional requirements that were set forth. A set of potential technologies were evaluated according to eight different technology criteria:

- TC1** The technology should be platform independent and run on a large number of devices.
- TC2** How mature is the technology? It should be supported by more than just a few devices.
- TC3** The technology should be stable and well documented, with available APIs and other resources.
- TC4** The technology should make it possible to create interactive applications with graphical user interfaces.
- TC5** Since the application will run on hand-held devices, the technology should be lightweight and require only moderate resources.
- TC6** The technology should support the use of IDEs (integrated development environments) and other tools that can aid in development and debugging.
- TC7** The technology should preferably be available for free.

TC8 Previous experience with the technology.

Because the field of mobile computing still is relatively new, there were not that many potential technologies to choose from which could offer the kind of functionality needed for this project. In general, only two real candidates were considered: Java ME¹ and the Microsoft .NET Compact Framework² for Windows Mobile. Java ME might be the more versatile of the two in terms of device availability, however, the .NET Framework is more modern and has all of the required functionality built in. In addition, Microsoft offers a proven development platform with a range of products and developer tools, which simplifies system maintenance and facilitates seamless integration. This was one of the main reasons for choosing .NET as a platform, because it would allow us to quickly develop a prototype with as much functionality as possible, considering the time allotted to the project. MSIS makes use of the Microsoft SQL Server database engine and ASP.NET technology, as well as the built-in support for web services, which is an indispensable part of our client-server architecture.

To implement the geographical position service, several technologies and techniques were considered. One of our main goals was to keep the implementation as self-contained as possible, and not to dependent on other systems or products. The idea was to use the existing wireless network infrastructure at NTNU to create a positioning service based on the Wi-Fi technology. Contrary to satellite based systems, which only work under certain conditions (i.e. a clear vision to the satellite is required), Wi-Fi networks are usually intended for indoor use, and the signals can traverse walls and other obstacles. This makes it well suited for our task. We ended up using a technique referred to as *location fingerprinting*, which is based on a location-dependent characteristic like the signal strength of a radio signal to produce a digital “fingerprint”, which uniquely identifies the location. The fingerprint is recorded by measuring the received signal strength indications (RSSI) from a number of nearby Access Points at the given position. These measures are then linked to a descriptive entity like a room number or building, and stored in a database for later retrieval. The location fingerprinting technique is described in more detail in section 6.3.

2.2 Description of the MSIS system

In the course of last autumn’s project, a fully functional prototype called the Mobile Student Information System (MSIS) was developed. In this section, the main features of the MSIS system and its technical architecture are presented.

2.2.1 Introduction

The main purpose of the MSIS system is to enhance daily life at the Norwegian University of Technology (NTNU), and assist students in their daily activities through the use of various mobile services. The system makes use of contextual information such as location, time, and personal preferences to provide the user with relevant and timely information. A prototype of MSIS has been deployed and evaluated within the infrastructure of the wireless network at NTNU.

¹<http://java.sun.com/javame/>

²<http://www.microsoft.com/.NET/>

2.2.2 Functionality

MSIS consists of three main parts: a lightweight client application for deployment on handheld devices, a web-based administration tool, and a backend server which provides database storage services, business logic, and a number of public web services. An account profile is associated with each user, which is used for authentication and storing of personal preferences. The user profile contains information about the student's situation, such as which courses they are attending and how far they have progressed in their study program. MSIS includes a location module which supports positioning of mobile devices within the wireless network available at the NTNU campus. The geographical positioning service does not depend on any third-party providers, and requires no upgrades to the existing infrastructure. The technology can be used to implement location-aware services that adapt their behavior to the current location of the user. One example of such a service is a location-aware search tool, which allows users to look up particular "points of interest" nearby. A point of interest is a specific location that the user is looking for. In terms of our project, a point of interest can for instance be a lecture hall or a computer lab, or a object like a coffee machine. By introducing a variable of context into the application, in this case location, we believe the user experience is greatly enhanced and the users will be able to carry out their tasks more efficiently.

The MSIS application consists of three main services:

- **Location Search:** allows users to search for different "points of interest" throughout campus, for example lecture rooms, computer labs, canteens, etc. It provides a short description of the location, as well as a visual indication of the position of the location on a map. The map has zoom capabilities, supporting a potential unlimited number of zoom levels. It is also possible to obtain your own location based on the position of the mobile device in a similar fashion to GPS technology. The Wi-Fi based positioning technology eliminates the need for other positioning systems such as GPS.
- **Lecture Schedule Planner:** allows users to view appointments scheduled for the current week. At present, the service provides information about course lectures and exercise guidance hours. The schedule service retrieves data about the courses from a database, and up-to-date timetables are created dynamically based on the courses the student is attending the current semester.
- **Announcements:** this service is intended to provide news, notifications, and other relevant information to the user. The announcements are presented in a list sortable by time, category, or priority.

Figure 2.3 shows the functional overview of the first MSIS system. Details about each component are discussed in the following section.

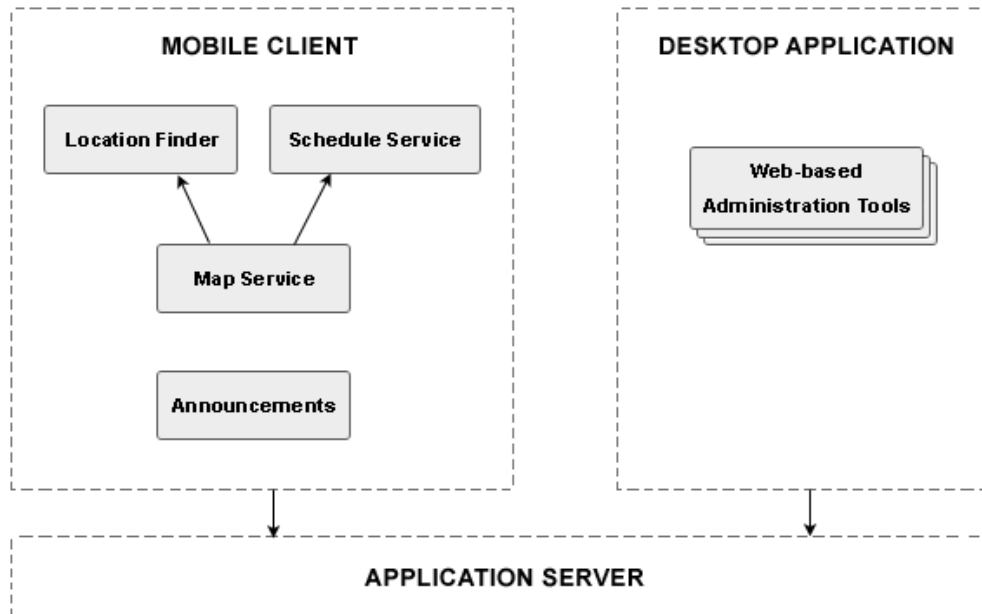


Figure 2.3: Functional overview

2.2.3 Architecture and implementation

At the topmost level, the architecture of MSIS consists of three disjunctive layers: the presentation layer, the business/data access layer, and the data layer. A visual representation of the architecture is illustrated in Figure 2.4. The multi-tier architecture makes the system design flexible and extensible, because of a clear separation of the user interface from the business logic and underlying data models. It also facilitates deployment on heterogeneous platforms, and offers the ability to leverage processing power on multiple nodes. On a more detailed level, the MSIS architecture comprises the following main components:

MSIS Web Services

This module consists of a number of web services, which is responsible for most of the processing and calculations done by the system. The web services are implemented on a Microsoft .NET application server using ASP.NET and C# business objects. The web services make use of several helper classes, such as the Data Access Layer (DAL) and Business Logic Layer (BLL). The DAL encapsulates the raw operations on data using the ADO.NET data access components included as part of the Microsoft .NET Framework. ADO.NET provides a set of classes that can be used by developers to access and modify data stored in a data source, in our case a relational database system powered by Microsoft SQL Server. The BLL encapsulate various objects available for manipulation by MSIS, such as appointments, news announcements, and locations. Furthermore, the BLL provides Wi-Fi positioning functionality. Support for web services is included in many development frameworks and programming languages, thus it is possible to develop clients for different platforms consuming the same web service.

MSIS Client Application

The MSIS Client is an application developed for handheld devices based on the Windows Mobile operating system. It offers a graphical user interface to all the functions available in MSIS: the dynamic schedule service, a tool for locating rooms and buildings on campus with support for maps and real-time geographical positioning of the mobile device, and a news/announcements service. The client application was designed to be light-weight and easy to use. All resource intensive functions are offloaded to the external application server to minimize the need for computational power on the mobile device, a concept known as *cyberforaging*³. All external communication is done through the web services.

MSIS Administration Interface

The administration interface is a set of web-based tools for managing MSIS. It is accessible through a standard web browser, and allows users of MSIS to manage their own profile and personal preferences. In the future it could also be extended to provide administrators access to various parts of the system, for example allow course supervisors to edit their lecture schedule, or simplify the addition of new locations. The administration interface is implemented with ASP.NET technology, and makes use of the web services for database access.

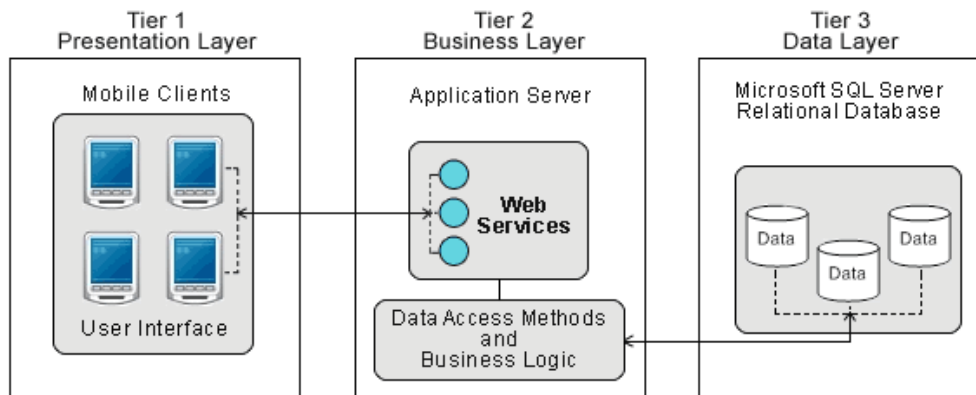


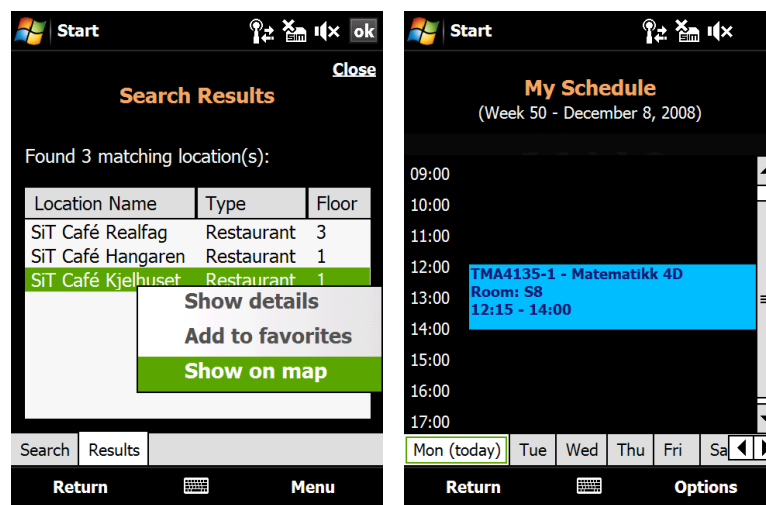
Figure 2.4: The MSIS architecture

For a more detailed description of the modular decomposition of the MSIS architecture, refer to Moe [56], Chapter 5.4.

³Cyberforaging refers to the temporary extension of the computing resources of a mobile device with remote computing resources on a wired machine.

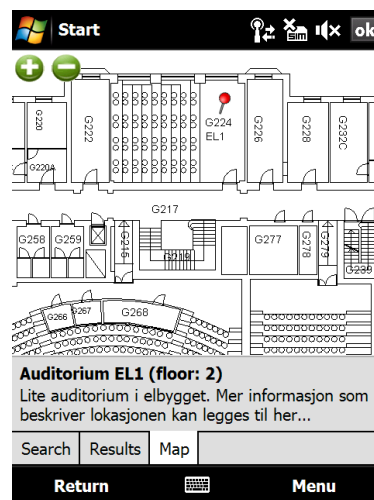
2.2.4 Screenshots

Below is a collection of screenshots, which illustrate some of the MSIS services in the first prototype as they appear on a mobile device. Figure 2.5(a) shows the result list after doing a location search. Figure 2.5(b) shows how a typical lecture is displayed in the Schedule service. Finally, the map functionality is illustrated in Figure 2.5(c).



(a) Location Search

(b) Lecture Schedule



(c) The Map Service

Figure 2.5: Screenshots of MSIS

2.3 Similar work

As part of our preliminary study we reviewed similar solutions and work done by others in the field of location-based services and mobile computing. A total of five solutions were examined: myCampus [58], i:Byen [38], Wireless Historical City Guide [43], Cyberguide [1], Mobile Turismo [55]. These systems have certain similarities with our project, mainly in that they use wireless networking for communication, and include an element of context-awareness.

myCampus is an environment for context-aware mobile services aimed at enhancing everyday campus life at Carnegie Mellon University (CMU). It makes use of location-awareness, personal preferences, and semantic web technology⁴ to provide various services to its users. The i:Byen project utilizes geographical positioning within Wireless Trondheim [8] to provide a cultural guide to citizens of Trondheim. The Historical City Guide is a similar service, with the purpose to make the history of Trondheim more easily accessible by combining information from the libraries and museums in Trondheim with location-awareness. The Cyberguide project, introduced in 1997, was one of the first to make use of context-awareness and mobile computing to provide a handheld tourist guide. Mobile Turismo is a platform that can be used to build mobile services for tourists. It uses various wireless technologies to support geographical positioning and communication.

As far as our understanding goes, none of these systems have yet to be set into production. They are mostly research projects and proof of concepts. One of the reasons for this might be that wireless networks and mobile computing are still not ubiquitous, and there are often factors besides technical constraints that impede the widespread deployment of such services. For example, political issues, lack of standardization, and security and privacy issues are among the factors that may defer the progress.

Our positioning module is inspired by some of the ideas presented by Gjøl [35] in their description of the “Fumble” project. Fumble relies on location fingerprinting to determine the position of the user. The work of Gjøl [35] proves that existing infrastructure can be utilized to position mobile users within a wireless network. Hence, we decided to employ this technique in our system as well. During the work with Fumble, they discovered that a key success factor for such a system is that it needs to be closely integrated with existing information systems to take advantage of the existing data. The usefulness of the Fumble project suffered because of this. We have tried to prepare for this possibility in our system. The modular design and the clear separation between the data layer and the presentation layer makes it easy to replace a data source, or create interfaces for simplified information sharing between our system and legacy systems. For example, instead of maintaining our own database of course information, the schedule service can be integrated with the existing system that is managed by the student administration department, thereby reducing duplication of data. At the present time, unfortunately, it is not possible to get access to the existing systems, as the database is using a proprietary format. However, we have been in contact with the person responsible for the database, and they intend to make it available at a later point. It should then be an easy task to connect the two systems together.

In a project by Haugvik [39] from 2006, a survey was conducted among students at NTNU concerning a location-aware system. The purpose of the survey was to gather feedback regarding what information and functionality a system for location-based services should support. The

⁴The semantic web is a term used to describe web content that is meaningful to computers.

primary focus of their work concerned context-sensitive information, which would allow users to “attach” notes to specific areas, such as upcoming events or small memos, but that are not necessary related to student information. This is different from the system described in our project, in that we focus more on information pertaining to the student life on campus. Nevertheless, some of the findings from their research still apply to our project. Some interesting observations can be made from their conclusions:

- The test persons expressed a need for a service which would list relevant locations or services in close proximity, e.g. the nearest computer lab or printer.
- New students felt a need for information about where and when they had lectures.
- Many of the test persons would find the map service useful, but would like to see a better separation of buildings, rooms, and other objects. Zoom functionality was also requested.
- A guiding service that would direct the user from their current position to a certain location on the map was suggested.
- All the test persons felt that such a system would make it easier to become acquainted with the campus area, and it would be particularly useful for new students.

These findings correlate very well with the feedback we got from our survey. Our service offers a location search tool which allows students to search for rooms or buildings, and also an option to denote the position of the location on a map. The maps have been designed to emphasize the buildings and rooms, and do not show other objects like roads, vegetation, etc. We have employed a local coordinate system instead of world coordinates to make it easier to highlight the relevant information. In the first version, the system did not automatically list locations or services in order of distance from the user, but this may be implemented using data from the positioning module. For example, when a user submits a search for “Computer labs”, the system takes into account where the user is located, and returns a list of computer labs in close proximity.

Chapter 3

Research approach

“Generally, the theories we believe we call facts, and the facts we disbelieve we call theories.”

–Felix Cohen

This chapter describes the research approach chosen for this project. In section 3.1, the research questions are presented. The chapter continues with a discussion of the selected research models and methodologies in section 3.2. A discussion of usability design principles and technology acceptance models is given in section 3.3.

3.1 Research questions

In the course of this project, we seek to find answers to the following questions:

- 1) *Can user-centric mobile information services enhance the everyday campus life of students at NTNU?*
- 2) *How does users' attitudes towards new technology influence their acceptance of mobile services?*
- 3) *Will the introduction of context-awareness add to the utility of a mobile student information system?*

3.2 Research method

When conducting scientific research, the work should be organized in pursuance of well established research models and methodologies. Models on how to conduct information system (IS) research have been the subject of much discussion over the past few years [46, 67, 81]. This project employs the following research methods: (a) the design-science paradigm, (b) survey research, and (c) usability testing. In the following sections, we will describe these methods in more detail.

3.2.1 The design-science paradigm

Design-science is fundamentally a problem solving paradigm, and involves the creation, analysis and evaluation of design artifacts to gain knowledge about a problem domain and propose a solution. In computer science research, these artifacts include – but are not limited to – algorithms, mathematical proofs, human-computer interfaces, or prototypes. Design research can be especially fruitful when exploring the need for a new computer system. By presenting scenarios and design artifacts to the stakeholders during the process, it is easier to identify potential issues early in the design/development phase.

Figure 3.1 illustrates the general methodology for all design research. The figure is the result of work done by Takeda et al. [76], who has analyzed the reasoning that occurs in the course of a normal design cycle.

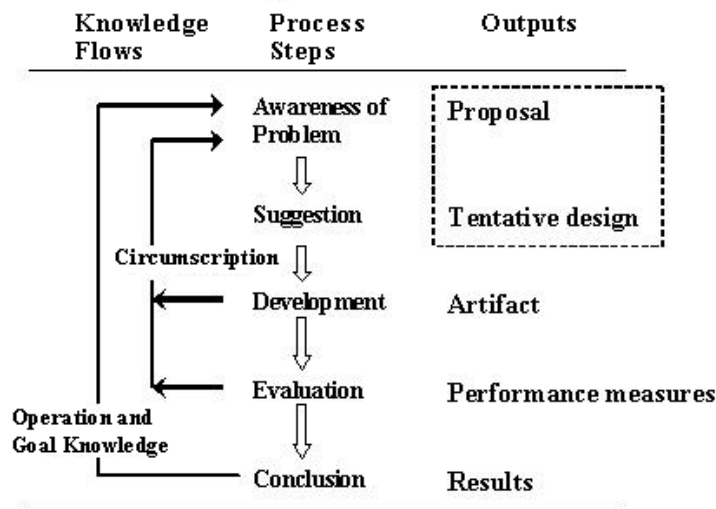


Figure 3.1: The general methodology of design-science

All design projects arise from a problem description, i.e. the awareness of a problem. Existing knowledge and theories are then applied to *suggest* solutions to the problem, for example by using techniques such as rapid prototyping. In the following phases, the solution is implemented and evaluated according to the criteria set forth in the *Awareness of Problem* phase. Usually, these steps are performed iteratively until a final implementation is achieved. The *Circumscription* arrow on Figure 3.1 denotes the basis of the iteration; the flow from partial completion of the cycle back to awareness of the problem. The *Conclusion* marks the end of the design project. The *Circumscription* and *Operation and Goal Knowledge* can be thought of as knowledge generating processes, in that it generates understanding that could only be gained from the specific act of construction [79]. This is at the heart of understanding design-science as a valid research model.

Hevner et al. [41] argues that IS research involves two complementary paradigms. Contrary to the design-science approach, the behavioral-science paradigm seeks to investigate the non-technical aspects of information systems. That is, to predict and study the human and organizational impacts of information system design and implementation, which are of vital importance when it comes to achieving the stated goals and final acceptance of the system. The goal of

behavioral-science research is truth. It seeks, according to Hevner et al. [41], “*justification of theories that explain or predict phenomena related to the identified business need*”. The goal of design-science research is utility. Truth informs design and utility informs theory. Thus, both aspects should be considered when conducting IS research. The danger of a design-science research approach is putting too much emphasis on the design artifacts, without considering the usefulness of the artifact in real organizational settings. Consequently, it becomes equally important to align the theoretical foundations with technological capabilities and constraints.

Hevner et al. [41] defines seven guidelines that should be addressed during the course of a design-science project. The guidelines are summarized in Table 3.1

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 3.1: Table from [41]: Design-Science Research Guidelines

Guideline 1 states that the outcome of the design process should be a viable artifact. For the purpose of this project, the artifact constitutes the MSIS service. The evaluation of the prototype will generate feedback and suggestions to the first design, which will be used to improve upon

the MSIS service in the next iteration. During the first iteration, a number of user scenarios were generated to exemplify actual use of the system, and elicit system requirements. Based on the requirement specification, a suitable architecture was proposed, and a prototype was developed using the selected technology. Because of the time constraints affecting the project, all the requirements were not implemented in the first version of the prototype. Chapter 5 presents the results of the second iteration. The design artifact is still the MSIS service, and the findings from the evaluation phase influence the next implementation iteration.

The use of mobile computing and positioning services relate to Guideline 2: application of state-of-the-art technology to solve a known problem. Guideline 3 states that the outcome of the design process must be evaluated to assess the usefulness and quality of the design artifact. In Chapter 4, the system is thoroughly evaluated in terms of utility and user acceptance. Guideline 4 suggests that the design must be underpinned by theoretical foundations and facts. This is covered by the preliminary study and student survey conducted last autumn. Guideline 5 states that design-science research should be carried out using sound and well-founded methods. This chapter describes the research methods and design principles which we employ in this project. Guideline 6 argues that the research must be seen as an iterative process, where you continuously search for an effective solution to the problem. That is, we start out with a description of the problem, and arrive at the solution by making informed design decisions. Subsequently, the solution must be presented to the project stakeholders, including both management and technology-minded parties. The project carried out last autumn constitutes the first iteration of the design research process. The outcome of the first iteration was a fully functional prototype. In this project, we evaluate the service with regard to usability and user acceptance. The findings from this evaluation forms the basis for the next step in the process; identifies new requirements and shortcomings with the first design, then provide directions for implementation of the next version of the system. The last iteration will be concluded with a short evaluation of the final system after the suggested enhancements have been implemented.

3.3 Usability and user acceptance testing

An important part of this project is to perform a thorough usability and user acceptance study of the proposed system. This is needed in order to assess how well the system fulfills its intended purpose in a real-world environment, and is formalized as item 3 in the Design-Science Research Guidelines. The evaluation serves to reveal possible shortcomings with the initial design and potential areas for improvement. This section will establish a theoretical fundament for the methods and models that are to be used in the evaluation phase.

3.3.1 What is usability?

It is the goal of any application developer to make useful and easy to use computer systems. But before we continue our discussion of usability testing, we will take a moment to describe what *usability* is, and which design principles and methodologies that are available to aid developers in creating better applications. In human-computer interaction or computer science, usability is often explained as the user-friendliness or ease of use of a given computer system. That is one aspect of it, but usability involves more than with what ease a computer system can be

operated. According to usability consultant Jakob Nielsen and computer science professor Ben Shneiderman, the usability of a system predicts five factors of good system design [61, 73]:

- Learnability: How easy is it for users to learn the system, and accomplish basic tasks the first time they encounter the system?
- Efficiency: Once users have learned to operate the system, how quickly can they perform their tasks? If the system is designed to replace a legacy system, how does the new system affect task efficiency?
- Memorability: After a period of non-use, how easy is it for users to return to the system and resume their tasks?
- Errors: How likely are errors to occur within the system, user-generated or system-generated? How severe are these errors, and how easily can users recover from the errors?
- Satisfaction: How pleasant is it to use the system? The satisfaction of a user is often directly correlated with other concepts of usability (learnability, efficiency, memorability, and error handling).

To evaluate the usability of a system, both the user interface and functionality must be considered. An intuitive user interface can lower the learning curve of the system, and increase the efficiency of typical tasks, but it must offer sufficient functionality to remain useful. User interface engineering has been subject to extensive research, and many guidelines and principles have been proposed to improve the quality of user interface design. In his book *The Humane Interface*, Raskin [65] suggests two paramount laws of user interface design:

First Law A computer shall not harm your work or, through inactivity, allow your work to come to harm.

Second Law A computer shall not waste your time or require you to do more work than is strictly necessary.

In other words, you should be able to perform your tasks in an efficient manner without any interruption. The computer system should work with you – not against you.

Larry Constantine and Lucy Lockwood suggests that user interface design should be directed by 6 principles, referred to as the Principles of User Interface Design [19, 20]:

- *The structure principle*: Design should organize the user interface purposefully, in meaningful and useful ways based on clear, consistent models that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. The structure principle is concerned with overall user interface architecture.
- *The simplicity principle*: The design should make simple, common tasks easy, communicating clearly and simply in the user’s own language, and providing good shortcuts that are meaningfully related to longer procedures.

- *The visibility principle:* The design should make all needed options and materials for a given task visible without distracting the user with extraneous or redundant information. Good designs do not overwhelm users with alternatives or confuse with unneeded information.
- *The feedback principle:* The design should keep users informed of actions or interpretations, changes of state or condition, and errors or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users.
- *The tolerance principle:* The design should be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing, while also preventing errors wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions.
- *The reuse principle:* The design should reuse internal and external components and behaviors, maintaining consistency with purpose rather than merely arbitrary consistency, thus reducing the need for users to rethink and remember.

In an effort to encourage software developers and usability professionals to understand user needs and tasks, the National Institute of Standards and Technology (NIST) has collaborated with industry to develop the Common Industry Specification for Usability - Requirements (CISU-R) [37]. The CISU-R sets standards for specifying usability requirements, which include information such as the context of use (the intended users, their goals and tasks, etc.), measures of usability for the product, and how to test whether the usability requirements have been met. The underlying goal of the standard is to aid in the creation of useful and usable products that allow users to complete their tasks efficiently, effectively, and with satisfaction. The CISU-R specifies three levels of compliance, allowing the usability requirements to be developed over time, with increasing detail and precision.

Other related standards for usability design and evaluation exist; including the ISO/IEC 25062 - “Common industry format for usability test reports”, ISO 13407 - “Human centred design processes for interactive systems”, and ISO/TR 16982:2002 - “Ergonomics of human-system interaction – Usability methods supporting human-centered design”.

3.3.2 Principles of usability design

Early Focus on Users and Tasks

As a designer, there are various considerations that will help ensure conformance with usability requirements. Gould and Lewis [36] claims that usability engineering should be user driven. In order to produce usable and user-friendly systems, designers must first understand the potential users. This understanding is achieved through observation and study of their cognitive, behavioral, and attitudinal characteristics. Furthermore, the nature of the tasks that the system is expected to carry out must be studied. This early focus on users and tasks is the first of three principles proposed by Gould and Lewis [36] in their research. It helps to rule out any misunderstandings or flaws at an early stage, and makes sure the direction of the design is in line with the needs of the users. According to Gould and Lewis [36], the most effective user interaction is

achieved by making end users part of the design team, facilitating direct contact between designers and potential users. Through personal interviews and observations of users, the designers can learn a lot about their users and the nature of the problems that the system is intended to solve. The user characteristics assessment should be conducted at the very earliest stages of the process, prior to system design. Once implementation has begun, it is usually very hard to re-arrange the usability requirements without substantial work. An alternative to having a panel of real users at the designers' disposal, the less expensive concept of personas can be used. Personas are made-up characters intended to resemble the characteristics of representative users.

Empirical Measurement

The second principle of design, according to Gould and Lewis [36], is Empirical Measurement. Intended users should test the system early in the design process, using simulators and prototypes, and their performance and attitudes towards the system should be evaluated according to empirical methods. The authors stress that the usability tests should measure how easily *people* can learn and use the prototype, not how the *system* conforms to functional requirements, which usually is the objective of a system test. They further claim that “reviewing or demonstrating a prototype system for typical users and getting their reaction to it can result in misleading conclusions.” People who have developed a system think differently about its use, do not make the same mistakes, and use it differently from novices. Users should be given simple tasks to carry out, and their performance, thoughts, and attitudes should be recorded and analyzed.” [36]

Iterative Design

When problems or issues are encountered during user testing, they must be identified, rectified, and tested again. This results in a cyclic process of prototyping, testing, analyzing, and refining a product or process. At each *iteration*, new improvements are added, and results of the behavioral testing are incorporated into the next version of the system – ultimately moving towards the final design. In iterative design, interaction with the designed system is used as a form of research for informing and evolving a project, as successive versions, or iterations of a design are implemented. Gould and Lewis [36] argue that software projects will benefit from an iterative development phase, however, it will only be feasible if an implementation strategy that permits early testing of design features and cheap modification of the evolving implementation has been planned. Such a strategy has to include fast flexible prototyping and highly modular implementation. After Gould and Lewis described the iterative design methodology back in 1985, it has since gained widespread recognition within the software development community. Nielsen [60] showed how iterative user interface design used in four case studies improved the overall usability from the first to the last iteration with 165%, while the median improvement per iteration was 38%. The design-science research method used in this project supports iterative design, and the .NET technology facilitates rapid development.

During more recent years, other design methodologies have emerged, such as the ones belonging to the agile [74, 49] group of development methodologies: Extreme Programming (XP) [12], Scrum [72], and the Rational Unified Process (RUP). However, the discussion of these topics falls outside the scope of this project.

3.3.3 How to measure usability?

We now know that usability is an expression of how easy a computer system is to learn and use, but how do we measure the usability of a system? There are a variety of methods currently in use to evaluate usability. Some methods make use of empirical data gathered from a group of potential users, while others rely on usability experts. Different evaluation methods apply to different stages of the design and development cycle. Some methods are intended to assess certain usability characteristics early in the process by evaluating the product specification, while other methods should be used to evaluate the final system. In addition, cost and time constraints are factors that should be considered when choosing the appropriate usability evaluation method. The evaluation methods can be classified into the following categories: (1) Cognitive modeling methods, (2) Inspection methods, (3) Inquiry methods, (4) Prototyping methods, (5) Testing methods, and (6) Other methods.

Cognitive modeling

Cognitive modeling involves creating a computation model to simulate or predict human behavior or performance on a given task [62]. Parameters which are measured can be such as key strokes and hand movement, for example the time it takes to select a menu item. This information can be used to improve user interfaces or predict problems during the design process. GOMS [15] is an example of a cognitive modeling method.

Inspection evaluation: cognitive walkthrough

Inspection evaluation methods involve observation of users while they carry out a specific task. Often these tests are carried out in a controlled environment allowing quantitative data to be recorded. An example of inspection evaluation is cognitive walkthrough [84], where a team of evaluators walk through the application discussing usability issues through the use of a paper prototype or a working prototype. This method is good at refining requirements and does not require a fully functional prototype, but it does not address user satisfaction or efficiency. Also, the designers may not behave as the average user when using the application. Cognitive walkthrough can be applied at all stages of the development process.

Focus groups and interviews

The Inquiry evaluation methods involve collecting qualitative data from users. Data is usually gathered through focus groups or personal interviews. In a focus group a moderator guides a discussion with a group of users of the application. Focus groups can help improve customer relations, and produce useful ideas from the potential users. However, it can be expensive and time consuming to schedule the meetings with the participants. One should be aware that this method may provide inaccurate results, because the environment is not natural to the user. Focus groups are best suited for testing and evaluation of a working application or prototype.

Interviews can be conducted among potential users or other shareholders with interests in the project to find out about their expectations and experience. It requires fewer participants than with focus groups, and detailed information can be obtained. However, it requires personal

attendance. Interviews can be conducted at different stages of the process to gather opinions from different people.

Prototyping

Rapid prototyping is a method that can be used in early stages of development to validate and refine the usability of a system. It is often very difficult for designers to conduct usability tests with the exact system being designed. Cost constraints, size, and design constraints usually lead the designer to creating a prototype of the system. For example, paper prototypes provide a cheap and simple mean to evaluate user interface designs without the need for an expensive working model. Rapid prototyping is well suited for identifying flaws with the design, or clarify misunderstandings before actual implementation begins. The disadvantage of this technique is that it can be hard for users to visualize how the final system will behave based on a simple prototype. A set of paper prototypes were created prior to the first implementation to help visualize the system and graphical user interface.

Remote usability testing and the “think-aloud protocol”

A commonly used method for conducting usability testing is the “think-aloud protocol” based on the work of Ericsson and Simon [28, 29]. Users are asked to express their thoughts (what they think, do, or feel) about the application while they are performing a set of specified tasks. The tests are often recorded on video or audio tapes, and observers usually take notes as the users go about their tasks. This gives usability engineers information about how the user interface matches the natural human way of thinking and acting, and highlights the features and processes to be improved. Results from such tests are usually quite reliable and close to what would be experienced by users in a real-world environment, however, the unnatural environment might be perceived as disturbing to the user. This method is useful in capturing users’ attitude towards the system, and is often successful in pinpointing specific problems with the design. Often the system is modified to (transparently) record some quantitative measures as well, such as the time to complete a specific task or the number of errors encountered during the test.

A similar method is remote usability testing, where the experimenter does not directly observe the users while they use the system. Instead, the participants provide feedback through a survey (e.g. online questionnaire), which provides the ability to gather data from a large sample of test persons. Remote testing has the advantage that the tests can be carried out in the user’s own environment rather than in usability labs, however it can be hard to control the participants and provide assistance during the tests. Therefore, remote testing should only be used when the system (or the particular module to be tested) is fairly functional, and free from obvious bugs.

Other methods

Various other methods are also used to evaluate the usability of a system. Benchmarking is a method which can be used to establish standardized threshold values for four key characteristics of a particular type of design: time to perform the task, time to fix errors, time to learn the application, and the functionality. Other designs can then be compared to the benchmark to

determine the usability of the system. Meta-Analysis [85] is another method, which combines the results of several studies to reach a more informed conclusion.

Metrics in quantitative evaluation methods

While some usability tests intend to elicit cognitive information from the subjects, such as their feelings, thoughts, and attitudes towards a system, other tests require the use of more quantitative metrics to assess the usability of a system. These metrics depend on the goals of the project, its quality requirements, and the current stage of the system. To test typical usability characteristics such as the effectiveness and efficiency of the system, one can measure how long it takes to complete the tasks, success/failure ratio, etc. More general usability factors such as user satisfaction and intention to use can be measured using rating scales of satisfaction, by measuring the percentage of users that completes the task, or by recording how many times the user expressed frustration with the system [48]. MSAM and TAM are instruments that measure usability metrics such as perceived usefulness, perceived ease of use, and intention to use by collecting empirical data from a sample of target users.

3.3.4 Addressing usability in mobile applications

The overall principles of usability discussed above apply to mobile services as well. However, the form factor and technical specifications of mobile devices may influence the usability of such applications in ways that differ from usability in general. Handheld devices such as cell phones and PDAs are much smaller in size than desktop computers, and they often come with low resolution displays in different sizes and proportions. In addition, mobile devices may have limited means for controlling the system, e.g. small and few buttons. This imposes several challenges with regard to the design of mobile applications:

- How should information be presented in a clear way on a screen with limited space?
- How should the user interface adapt to different screen sizes, resolutions, and layouts (portrait vs. landscape)?
- How should the system facilitate easy and intuitive navigation? Input devices may differ between devices, so the application might need to support more than one input device. For example: stylus, keypad, and/or voice control.
- It is often cumbersome to enter text on a mobile device, in which case it is better to allow the user to make selections instead.
- Users may have less experience of using mobile services than they have of using their computer. Because of this, the system must be very easy to learn and use.
- Mobile devices are based on different platforms which often use proprietary programming languages and application programming interfaces (APIs). This makes it hard to develop cross-platform mobile services.

- Mobile devices are often used in highly dynamic environments, and factors like lighting, noise, wind conditions, and temperature can affect the usability of the application. On the contrary, desktop computers have more predictable environments.
- Limited processing power and memory put constraints on the complexity of the application. If the application is sluggish and navigation is slow, the usability of the system is depreciated.

We can relate many of these challenges to the Principles of User Interface Design described by Constantine and Lockwood [19, 20] in their research on usability. By organizing the user interface in meaningful ways, important information and frequently used functions can be made easily accessible. Moreover, by keeping the number of steps necessary to carry out a task small, the efficiency increases, and the likelihood for errors is reduced. This is in line with Nielsen's and Shneiderman's five factors for good system design which were discussed in section 3.3.1. The tolerance principle states that the design should be flexible and tolerant. To achieve this, the system can be designed to allow multiple input devices to be used for controlling the system (e.g. both a stylus on touch screens and a keypad on devices without this possibility), the user interface can automatically stretch and align according to the screen properties, etc. By following these guidelines, the usability of the system will improve, and so will the user satisfaction.

Laboratory vs. real-world testing

When evaluating mobile services, the process is often a bit more involved because of the portable nature of handheld devices. Mobile services are designed to be used in everyday situations, often under rapidly changing environments. Therefore, laboratory testing is not always feasible. A typical usability test laboratory consists of a living room or office-like area connected to a monitoring area with a one-way mirror. Traditional inspection evaluation methods may not be used as it is difficult to observe what the user is doing within the application because of the small screen. Additional technology or software may be used to facilitate information capturing during the tests. A common approach is to attach a small camera to the device, in order for the designers to later analyze how the user interacted with the system. However, these devices may be inconvenient for the user, and can potentially influence the results of the test. Thus, the test setup itself may actually impede the usability of the system. Another option is to install special screen capturing software on the mobile device, which makes a video recording of the screen while the test person is operating the system. This approach is unobtrusive to the user, but on the other hand it does not capture emotional response or physical interaction (e.g. click of a hardware button). Another implication of screen capturing software is that it usually requires the device to be connected to a computer through a cable. This is likely to affect the mobility of the test scenario. A literature study by Kjeldskov and Graham [45] revealed that most (71%) mobile usability evaluations were conducted in laboratory settings. This may be due to data collection techniques such as think aloud, video recording or observations being difficult in the field. In 2005, Kaikkonen et al. [44] conducted a comparative study exploring the effect of the environment in usability tests. Their results were somewhat surprising: the cost of conducting a time-consuming field test is not worthwhile when searching user interface flaws to improve user

interaction. They further conclude that “*usability testing a mobile application in the laboratory seems to be sufficient when studying user interface and navigation issues.*”

We believe that the need for a field test largely depends on the nature of the mobile service that is to be tested, and the kind of problems it is designed to address. It is reasons to believe some applications can only be tested thoroughly in a real-world environment. For example, a digital journaling service for doctors should account for the typical workflow of the doctor, how he interacts with the patients, etc. It would be hard to discover all potential flaws with the design in a laboratory test. We believe the usability of such a system will be better evaluated after a period of regular use in familiar surroundings, because the context may affect the usage and performance. Mobile applications may also utilize the portability of the user, for example by adapting their behavior according to the location, in which case a field test may be the only option. This is also in keeping with the conclusion of Kaikkonen et al. [44]. On the other hand, in a laboratory test the moderators have much more control over the environment, and it is easier to prevent external factors and unexpected interruptions from interfering with the test. A pilot study should be performed initially to check the feasibility and to discover any inadequacies with the design, and then it can be accompanied with a field study at a later point. To quote Kaikkonen et al. [44]:

“In a field test, running a pre-test or a pilot is critical: there are so many details that can go wrong, and you really need to check that everything is working correctly.”

3.3.5 Technology acceptance models and theories

Mobile computing and mobile application services is a relatively new field of research. When introducing new information technology, it is critical to study the factors that influence the users’ intention to adopt the new services. Developers and vendors can apply this knowledge throughout the design and implementation process to create a better service. Various technology acceptance models and theories have been suggested to assist developers in the evaluation of new software applications. In this section, we will have a closer look at some of the major reference models. While some of the models do not directly apply to our research approach, the models that we use are largely derived from them. We have therefore included them for completeness.

Theory of Reasoned Action (TRA)

The Theory of Reasoned Action [6, 31] is derived from the field of social psychology. The theory posits that a person’s volitional behavior is a function of an individual’s attitude towards the behavior and subjective norms surrounding the performance of the behavior. In other words, if a person intends to do a behavior, then it is likely that the person will do it. According to Miller [54], a person’s attitude is defined as the sum of beliefs about a particular behavior; the individual’s positive or negative feelings about the behavior, and consequences arising from the behavior. The subjective norms concern how other people important to the individual perceive the behavior. For example, if the individual feel that a new application will be beneficial (e.g. increase task efficiency), and at the same time his co-workers encourage him to use the system, it is likely that the individual intend to use the new system. One should note, though, that attitudes and norms are not weighted equally. Some people may care less for what people think and put

more weight to their own attitudes, and vice versa. This is especially true concerning behavior that results in significant personal gain. The TRA has formed the basis for many subsequent theories and models.

Theory of Planned Behavior (TPB)

The Theory of Planned Behavior [5] is an extension of the Theory of Reasoned Action. Ajzen [5] felt the TRA was lacking one important factor. He claimed that individual behavior, in addition to attitude and subjective norms, also is influenced by the ease of which the behavior can be performed which he referred to as perceived behavioral control. The behavioral control is defined as one's perception of the difficulty of performing a behavior, including required effort, resources, etc.

Innovation Diffusion Theory (IDT)

The diffusion of innovations theory, proposed by Rogers [70], has been widely adopted in IS research over the past years. The theory seeks to capture how, why, and at what rate new ideas and technology spread throughout cultures. According to Rogers [70], people are seen as possessing different degrees of willingness to adopt innovations (e.g. new technology), and thus it is generally observed that the portion of the population adopting an innovation is approximately normally distributed over time. Breaking this normal distribution into segments leads to the segregation of individuals into the following five categories of individual innovativeness (from earliest to latest adopters): innovators, early adopters, early majority, late majority, laggards [70]. Rogers [70] suggested that the rate of adoption of innovations is impacted by five factors: relative advantage, compatibility, trialability, observability, and complexity. This list was later expanded in the context of IS research to include voluntariness, image, ease of use, result demonstrability, and visibility [57]. A more general model for the adoption of information technology is illustrated in Figure 3.2.

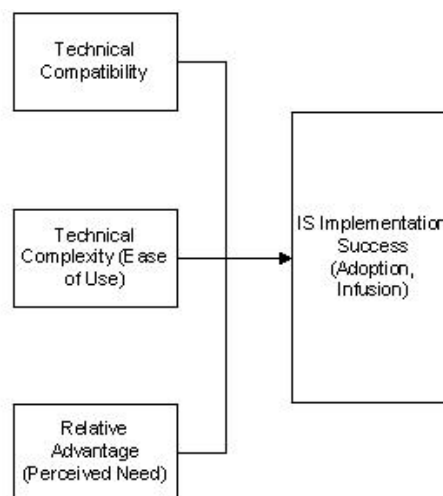


Figure 3.2: The IS Diffusion Variance Model [3, 21, 22]

Technology Acceptance Model (TAM)

The Technology Acceptance Model is an extension of the TRA, with the purpose to model how users approach and perceive new information systems and technology. The TAM suggests that a person's intention to use a system is determined by two main factors: perceived usefulness (PU) and perceived ease of use (PEoU). Davis [23] defines PU as the degree to which a person believes that using a particular system would enhance his or her task performance. Furthermore, PEoU is defined as the degree to which a person believes that using a particular system would be free from effort [23].

Nevertheless, a strong indication of intention to use does not implicitly stipulate actual use. The model implies that when someone forms an intention to act, they will be free to act without limitation. In practice constraints such as limited ability, time, environmental or organizational limits, and unconscious habits will limit the freedom to act. Thus, actual usage may not be a direct or immediate consequence of people's attitudes and intentions [9]. TAM is one of the most widely accepted models, and has been successfully applied to predict user acceptance of various new information technologies [32]. The correlation between the various factors of TAM is illustrated in Figure 3.3.

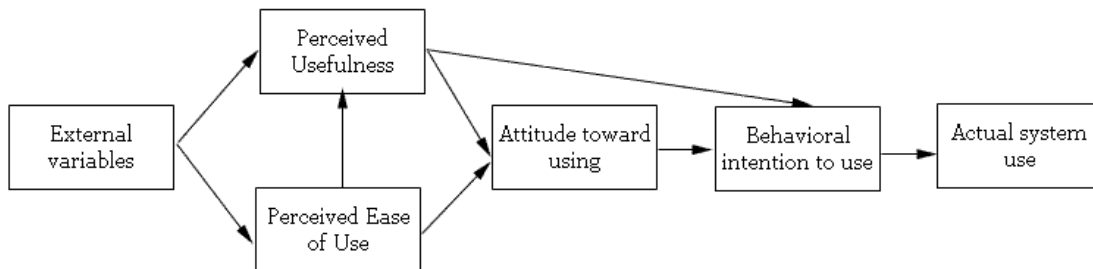


Figure 3.3: The Technology Acceptance Model from Davis et al. [24]

Unified Theory of Acceptance and Use of Technology (UTAUT)

The Unified Theory of Acceptance and Use of Technology (UTAUT) proposed by Venkatesh et al. [80] extends TAM to take into account four new constructs (Performance Expectancy, Effort Expectancy, Social Influence, and Facilitating Conditions) that bear significant influence on behavioral intention and ultimately usage of technologies. The variables of gender, age, experience and voluntariness of use are posited to mediate the impact of the four key constructs on usage intention and behavior as shown on Figure 3.4. The theory was developed through a review and consolidation of the constructs of eight models that earlier research had employed to explain IS usage behavior: the Theory of Reasoned Action [6, 31]; the Technology Acceptance Model [23]; the Motivational Model [25]; the Theory of planned behavior [5]; a combined TP-B/TAM [77]; the Model of PC Utilization [78]; the Innovation Diffusion Theory [70]; and Social Cognitive Theory [10, 16, 17, 18]). Subsequent validation of UTAUT in a longitudinal study found it to account for 70% of the variance in usage intention [80].

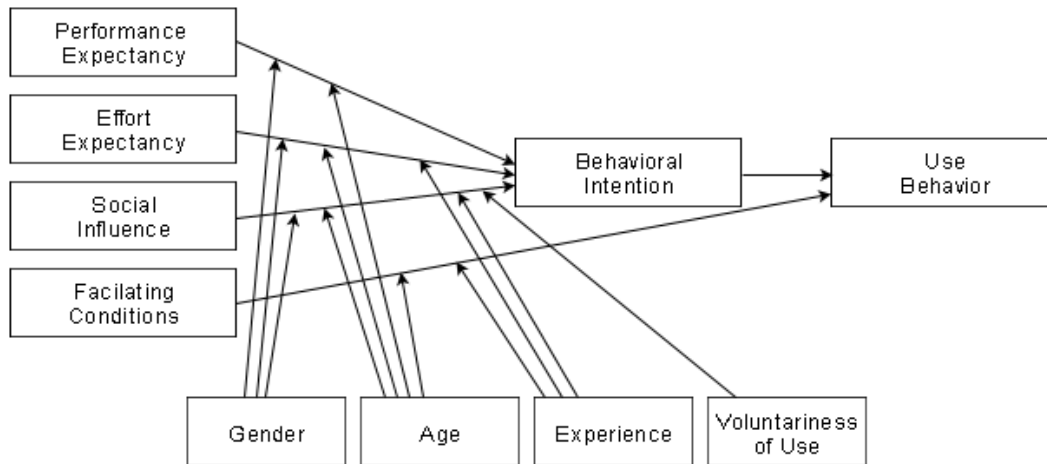


Figure 3.4: The Unified Theory of Acceptance and Use of Technology [80]

Mobile Services Acceptance Model (MSAM)

As previous research has shown, there exist well established models and theories for assessing the user acceptance of new ICT and IS. However, none of the above models were developed with mobile information systems in mind. Therefore, it is important to study the factors that influence the user adoption of mobile applications in particular. Gao et al. [32] propose an extended technology acceptance model, called the Mobile Services Acceptance Model (MSAM). The model is based on TAM, but also considers the influence of trust, context, and personal initiatives and characteristics on user adoption of mobile information systems.

The constructs Perceived Usefulness and Perceived Ease of Use were adopted from TAM. Figure 3.5 depicts that context (i.e. location, identity, resources, etc.) has a direct impact on the perceived usefulness and perceived ease of use of the system. For example, a mobile application may have higher perceived usefulness when no desktop computer is available. The personal initiatives can be defined as the user's willingness to try out new applications. Some people are more susceptible to new technology and innovations than others, which may be explained by the diffusion of innovations theory. The personal characteristics include the following elements: age, gender, educational background, knowledge and skills, culture, and preference. The Trust construct captures users' beliefs or faiths in a mobile system in terms of potential security and privacy threats. Trust is of concern in any software application, but the heterogeneous, ubiquitous nature of mobile computing presents additional challenges to the developers of such systems. For example, wireless communication requires data encryption schemes to protect against eavesdropping [56].

Many factors may influence people's trust in mobile applications; such as the ability to control privacy settings and personal preferences. A reputable software provider will also engender trust and integrity in the application. As illustrated in Figure 3.5, trust is assumed to directly influence the intention to use mobile applications. Intention to use refers to people's motives to adopt a particular piece of software. According to Gao et al. [32], the perceived usefulness and perceived ease of use are positively correlated with usage intention. Likewise, previous research

work [24] has confirmed that perceived ease of use has a positive effect on perceived usefulness.

In a recent case study [32], the MSAM was applied to a location based service named Find-MyFriends. The aim of the system was to allow people to locate each other during the student festival UKA-07. The study demonstrates the applicability of the MSAM, and suggests that it is a suitable model for evaluating user adoption of mobile services.

In our study, we are using a revised version of the Mobile Services Acceptance Model to evaluate the MSIS system. We have kept the six main constructs; PU, PEOU, Trust, Personal Initiatives and Characteristics, Context, and Intention to Use, but the model has been adapted to better capture the distinctive characteristics of the MSIS system. The MSAM was initially developed for mobile e-commerce applications, thus some of the measurement items were not relevant to our project.

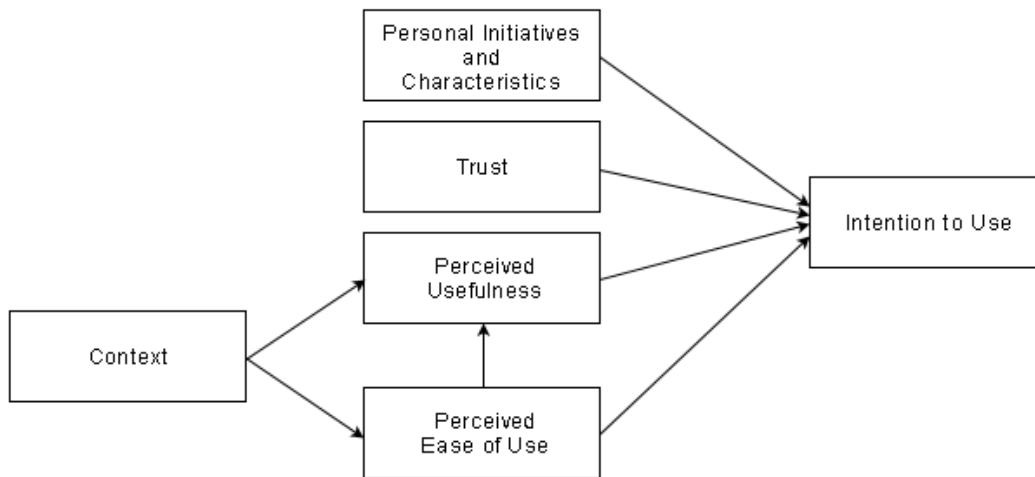


Figure 3.5: The Mobile Services Acceptance Model [32]

3.3.6 The measurement constructs

Before deciding which models and techniques to employ in our research, we needed to have a clear understanding of which parameters and aspects of the system that were to be measured. In a traditional usability study, the focus is mainly on the functional aspects of the system. Usability testing measures the ease of use of a particular system or part of a system, and the aim is to discover errors and areas of improvements by observing people as they interact with the system. Usability tests can be executed under highly controlled conditions in a laboratory, or under conditions that closely mimic the user's own work environment. Usability testing generally involves measuring how well test subjects respond in four areas: efficiency, accuracy, recall, and emotional response [61], which relate to Nielsen and Shneiderman's principles of good system design.

- Efficiency: measures how much time is required for people to complete a task using the system, e.g. the number of steps involved.

- Accuracy: how many mistakes did people make? Did the system provide useful error messages and information, or were the errors fatal and non-recoverable?
- Recall: how much does the person remember afterwards or after periods of non-use?
- Emotional response: what feelings does the person express towards the system? Is the person confident completing the task, or do they find the situation stressful?

The evaluation in this project has been conducted in accordance with these principles, but we have also considered some additional factors introduced by MSAM. Of particular interest are the impact of context, and the notion of trust on user adoption of mobile services. For each of the constructs defined by the MSAM, we composed a number of measurement items. A 7-point Likert scale was used to measure the responses to each of the items. The usefulness and perceived ease of use measures to what extent people will benefit from using the system, in terms of increased task efficiency and lower error rate. The trust, personal initiatives and characteristics, and intention to use constructs concern the users' emotional response towards the system. Due to the nature of mobile services, the notion of context becomes an important factor. Context in this regard concerns characteristics about the user, such as their location, past experience/history, social factors, and so forth. These are some of the factors that may influence how likely the user is to adopt the system.

Chapter 4

Evaluation

“It is a fine thing to be honest, but it is also very important to be right.”

–Winston Churchill

4.1 Introduction

To get an understanding of how well our system works in a real environment, we decided to carry out a usability test and user acceptance survey among our intended users. A group of 25 students were invited to participate in the study. The test group consisted of students from various study programs, including people with both technical and non-technical background. Most of the survey participants had at least one mobile device and had some previous experience with mobile applications. Students at all levels participated, ranging from first year students to graduates. This diversity among the test persons is expected to produce a more balanced view. The study constitutes an important part of our final evaluation of this project. The results from the study will either confirm or reject our hypothesis that a system like MSIS is capable of conveying user-centric information to students at NTNU, and thus improve upon the current information services available. Findings from our preliminary study last autumn confirmed that the students were not particularly satisfied with the current systems.

The evaluation process is twofold. One part involves a qualitative study, which in many ways is similar to a usability test, where the focus is on the usability of the system and its services, and how it is perceived by its users. The other part of the study consists of a quantitative survey based on an empirical model for estimating user acceptance of mobile services. The purpose of the quantitative survey is to investigate whether the target group intends to use the system and how they see it fit their needs. The qualitative study makes use of a number of scenarios intended to illustrate the various range of uses. Feedback was gathered by observing the users during the test, and the participants were asked to write down notes while they carried out the tasks specified in the scenarios.

Before commencing the test, the participants were given a brief introduction to the MSIS system and the services it provides. A mobile device with the application pre-installed was handed out. The participants were also informed that the data being collected were part of a research study.

4.2 The usability test

The usability tests were carried out in accordance with the inspection evaluation method, in an environment resembling a real-world situation for the students. The “think-aloud protocol” was applied to elicit cognitive feedback from the participants during the tests. Each test lasted for approximately 45 minutes. Two scenarios were proposed to illustrate the use of the application. The scenarios consisted of a number of tasks that students are likely to encounter during their stay at the university. The first scenario concerned the location finder service and map service. The test person was first asked to log on to the system using the provided username and password. Then, using the touch screen navigation, they were instructed to open the location search tool and locate a computer lab in the IT building. The task included paper prints of the relevant maps on which the position should be indicated. We used this as a mean to later check that the correct positioning data was returned. It could always be a chance for some students would mark the location on the correct position on the paper map, without actually using the application to find the location. To avoid this, we picked a less known laboratory room which we believe few students would be able to pinpoint without using the application.

In the last part of the scenario we tested the Wi-Fi positioning module. The positioning module provides GPS-like functionality within the coverage area of any wireless network. In contrast to the traditional GPS system, our positioning service works equally well indoors as outdoors. Because GPS relies on an unobstructed view to the satellites it works poor inside buildings. Our positioning service allows users to query their current position and the name of the location from any room or building throughout campus. The test persons were asked to walk to a lecture hall nearby, then activate the “Where am I” function. The system would then denote their current position on a map, and show a description of the location. We also walked around a bit to test the service on other locations to get a better understanding of how the service is supposed to help people get acquainted with the campus area. It is natural to think this service is of more importance to junior students than senior students.

In the second scenario, the schedule service was put to the test. The scenario consisted of three sub tasks. The first task was to log on to the web-based administration interface using a computer. The administration interface allows users to manage their user profile, change their account password, and add or remove courses through a regular web browser. The course list is used to generate a time table showing lectures and exercise guidance hours for the present week. The test persons were each assigned a user profile which contained information about the student’s courses and study program. We allowed for users to enter arbitrary course codes and study programs to test the flexibility of the system. A set of example data was provided for those who did not attend any classes this semester.

Next, the testers were asked to start the schedule service in the MSIS client application. The schedule shows information about upcoming lecture hours and exercise guidance hours for the courses supplied in the previous task. The start and ending time, room number, and teacher is

displayed for each appointment. To verify the correctness of the data, the test persons were asked to note down the information on a piece of paper. This allowed us to review the results later in order to verify the validity of the data. In the last task, the integration between the schedule service and the map service was tested. It is possible to click on an appointment in the schedule to have the location indicated on a map.

During the tests we observed the participants and how they interacted with the system. To measure how intuitive and easy to learn the user interface is for new users, the participants were offered as little assistance as possible during the tests. Gould and Lewis [36] discuss the importance of empirical measurement with respect to usability design; potential users do seldom perceive the system in the same way as the developers. They argue that one is more likely to obtain a correct measure of a system's usability by observing users while they carry out basic tasks, instead of just demonstrating the prototype. We wanted to observe the participants' emotional responses while operating the system. This proved to be difficult when running many tests simultaneously; therefore we decided to perform no more than two tests at any given time. It was more time consuming, but it had several advantages. First of all, it became easier to find a suitable time slot for each participant. It is important to keep in mind that our test persons were occupied most of the day, and participated on a voluntary basis during their spare time. A few of the test persons were interviewed after completing the test, to elicit detailed feedback regarding particular parts of the scenarios. As such, several usability evaluation methods were applied to gather as much information as possible from the tests. To be able to offer some incentive while recruiting testers, we arranged a prize draw among the participants. The original usability test scenarios in Norwegian are included in Appendix A.

4.3 The user acceptance survey

The user acceptance survey is an empirical study based on the Mobile Services Acceptance Model (MSAM) developed by Gao et al. [32].

The MSAM is an instrument which is used to estimate the usefulness and usability of mobile applications. It base its assumption on a set of quantifiable data gathered from a group of users, which are likely to be among the intended target group. The survey respondents were the same that also participated in the usability test. The respondents were asked to evaluate the system according to a number of constructs which are described in section 3.3.5. To measure the impact of these constructs on mobile service adoption, the respondents were asked to specify their degree of agreement with a total of 33 statements (measurement items). A 7-point Likert scale, with 1 being the negative end of the scale (*Strongly Disagree*) and 7 being the positive end of the scale (*Strongly Agree*), was used to measure participants' responses to items in the questionnaire. These ratings were then analyzed to estimate the individual's intention to use the application. The instrument also describes how factors such as context, personal initiatives, and trust influence the user adoption of the mobile service. The survey seeks to address the behavioral aspects of information system research; human and organizational impacts of IS design and implementation, which according to Hevner et al. [41] are an important determinant for final acceptance of the system.

A 7-point scale was selected for the instrument because it has been used in previous re-

search [32], and the model has shown to give the best results with a 7-point scale. A recent study conducted by Dawes [26] examines how using Likert scales with 5-point, 7-point or 10-point format affects the resultant data in terms of mean scores, and measures of dispersion and shape. Most researchers will agree that more than seven points on a scale are too much. People are not able to place their point of view on a scale greater than seven as it becomes too dispersed. On the other hand, a small scale may be too narrow and not allow for sufficient nuances in the responses. According to a study by Alwin and Krosnick [7], scales of 5 - 7 points seem to be the most accurate and produce the most reliable results. It is common practice to choose an odd-numbered scale in surveys where respondents may remain neutral to some statements. In studies where the respondents should be “forced” to make a statement, an even-numbered scale may be used (e.g. 2-, 4-, or 6-point scale). Findings show that a 5- or 7-point scale may produce slightly higher mean scores relative to the highest possible attainable score, compared to those produced from a 10-point scale [26]. Regarding the order of the scale, we have not been able to find any strong academic reference that favor one over the other. Though, the general perception by looking at example surveys in academic literature seems to be that it is better to go from the negative to the positive (left to right).

MSAM propose seven hypotheses, relating user’s adoption of mobile service to the correlations between the measurement constructs [33]:

- H1** Context has a direct positive effect on Perceived Usefulness.
- H2** Context has a direct positive effect on Perceived Ease of Use.
- H3** Perceived Ease of Use has a direct positive effect on Perceived Usefulness.
- H4** Personal Initiatives and Characteristics has a direct positive effect Intention to Use.
- H5** Trust has a direct positive effect on Intention to Use.
- H6** Perceived Usefulness has a direct positive effect on Intention to Use.
- H7** Perceived Ease of Use has a direct positive effect on Intention to Use.

As this study is one of the first to make use of the Mobile Services Acceptance Model in the evaluation of a mobile service, it also serves to assess the validity of the hypotheses by analyzing actual user responses. The findings may be used to further improve upon or make enhancements to the research model in future studies.

The complete user acceptance survey is included in Appendix B.

4.4 Presentation of the test results

In this section we will present the results and findings from the usability and user acceptance tests. We will start with a discussion of the qualitative data gathered from the usability tests. In the next section, the data from the mobile services user acceptance survey will be analyzed. As mentioned above, the usability tests were based on common scenarios that illustrate the use of the various services provided by the system. At the end of each task the users were encouraged to give comments and feedback with regard to usability of the graphical user interface, response time, and task difficulty. The following observations are based on comments obtained from the participants, as well as feedback collected during discussions with a selection of the test subjects. The results are presented in the same order as the tasks were performed.

4.4.1 Usability test

Test Scenario 1

Logging on to the system

The first step is for the user to log on to the system using the provided username and password. From an educational and psychological perspective, we wanted to start the test with a simple task because it helps boost the confidence of the participant. A usability test can be a distressing situation for the participants, and as such a “flying start” may contribute to a pleasant experience.

Locating a room

In the next task the test person were asked to find a computer lab using the location search tool. During the first two tests there were some comments about the shape and wording of the actual task. Since most of the test persons had never used a Windows Mobile handheld device previously, there was some confusion and misunderstandings with regard to whether certain parts and controls were part of our system or the operating system. Likewise, the on-screen keyboard was a bit awkward compared to the keys on a traditional mobile phone. Considering that almost no one had experience with Windows Mobile devices we expected this to become a challenge; the test persons may be inclined to comment on issues pertaining to the device or operating system rather than our system. This did not turn out to be a problem. Some adjustments were made to rule out any confusion.

An issue which occurred a few times was due to instabilities of the network. The device occasionally lost connection to the wireless network, which affected the operation of the system. This is reflected in Figure 4.1 which shows the success rate of the “location search” test. As depicted by the chart, one test failed on the first attempt due to a network problem. After the connection was restored, the test completed successfully. These problems can however not be contributed to our system, as it was caused by a weakness in the network component of the device, or a problem with the Wi-Fi infrastructure (i.e. poor signal reception). Strangely enough, these issues were only present on the first test day, which lead us to believe the problem was somehow related to an issue with the network that particular day. After this incident we modified the system to alert the user if the network connection unexpectedly is lost. We also implemented a Wi-Fi network status indicator, informing the user of the signal reception.

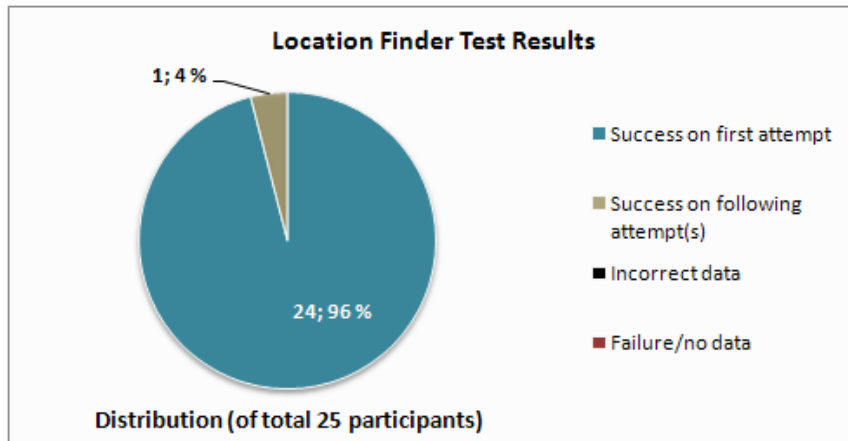


Figure 4.1: Test results - location search service

All the test persons were able to find the correct location without any intervention from us. The map service also worked flawlessly. One user felt the zoom buttons were at a bit too small. It is easy to increase the size of the buttons, but it is also a trade-off with visible map area on the limited screen size. At least one respondent expected the zoom to work by also clicking directly on the location pin. This will likely not be implemented. First of all, it would be difficult to differentiate between the zoom-in and zoom-out functionality, and secondly, it would interfere with the drag-and-drop functionality for scrolling the map. A few minor bugs were reported and fixed. A suggestion was given by several participants that the map should be automatically centered on the target location. This suggestion has been implemented in the final version. Another suggestion concerning the possibility for viewing adjacent floors within the same building was brought up when working with the map. It might be a good idea and is definitely something we will consider. We also got feedback from a few that the cardinal points should be indicated on the map for easier orientation.

The geographical positioning functionality

The geographical positioning module of the location service was tested by walking with the test person to a particular location, a large auditorium in the IT building, then activating the system. The current position of the user should then be indicated on the corresponding map, along with a textual description of the location. In this case, the description includes the name of the auditorium, the floor number, and some information about the auditorium. The initial test scenario did only include one location; however, we decided to try out the service in other nearby locations as well. One of the reasons for this was to demonstrate that the data was indeed returned from the Wi-Fi positioning module based on the location fingerprint, and not just fake data hard-coded into the source code of the application. As Figure 4.2 shows, in 88% (22/25) of the tests the correct location and position was returned at the first attempt. In two tests the system at first did not return a location at all, but at the second attempt it worked correctly. We believe that this was due to network interference which affected the wireless signals, or that the test person's

orientation resulted in a discrepancy between the observed fingerprint and the fingerprint stored in the location database. Especially if the signal reception is poor, the characteristics of the observed signals can vary, and therefore in some cases no match is found in the database. When this happens, it is usually enough to change the orientation of the device or move a few feet to get a proper reading. In one test the network connection was lost which obviously prevented us from completing the test. This problem, however, was caused by the phone or wireless network and not our system. After restarting the phone the problem had disappeared, so likely it was some networking component in Windows Mobile that had failed. It is worth mentioning, that at no point were incorrect data returned by the system, such as a wrong location or a map error. It is our opinion that no data is preferred over incorrect data.

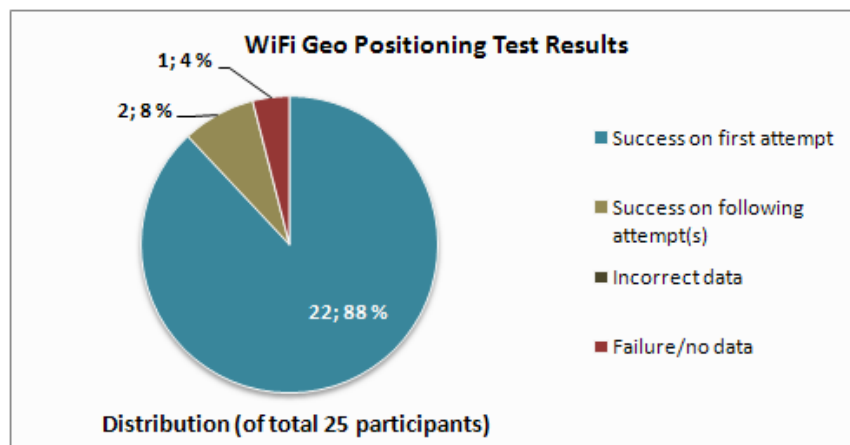


Figure 4.2: Test results - geographical positioning module

The response time was perceived as either “very good” or “acceptable”, which we are pleased with. Especially, considering that the application server was running on low-end hardware over a home ADSL Internet connection, and most of the data is retrieved in real-time via the network. Each operation was completed in 1-3 seconds. This is well within the upper bound of 5 seconds defined in the requirements specification for the system (see [56], section 5.3.2). In a production environment, a more reliable network connection together with dedicated servers will likely improve the response time even further. Similar to all applications utilizing networking, temporary congestion and network latency will of course affect the user experience of our system as well. However, to minimize the impact of occasional network timeouts or temporary connection problems, the system has been designed to offer limited functionality while in offline mode. For example, if a map has already been downloaded it is possible to navigate around the map. Similarly, it is possible to view the weekly schedule without an Internet connection if the schedule has been previously loaded. This is at least better than to just lock down the application if the connection is lost.

In general, the test persons were very pleased with the location search service. The user interface appeared clear and intuitive, despite the fact that they had little previous experience

with mobile services and Windows Mobile devices in general. Especially the “drag-and-drop” feature with which you interact with the map was well-received. Some of the participants even expressed excitement while operating the system, smiling and clearly enjoying working with the system. Based on the feedback received, there are indications that the location search service is perceived as more usefulness by first year students, and decreases as the student is becoming acquainted with the campus. A more detailed analysis of the perceived usefulness is given in the descriptive analysis of the survey in section 4.4.2.

Test Scenario 2

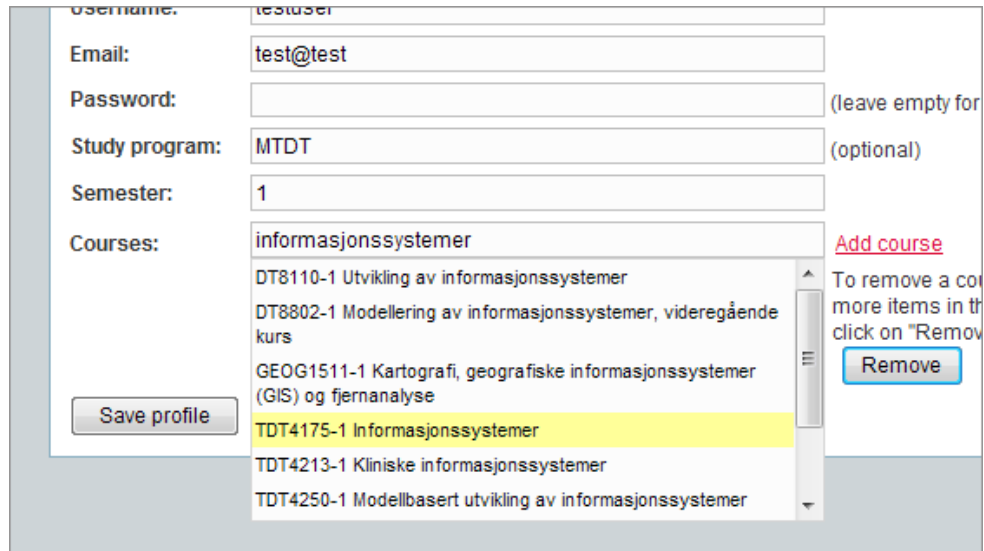
The second scenario involved the other main component of the MSIS system: the schedule service. The scenario is based on a typical situation where a student wants to check the upcoming lectures for the current week. As with the previous scenario, this scenario was also divided into tasks. Each task was designed to test a particular functionality within the system.

The administration web interface

In Task 1, the test person was asked to log on to the web based administration interface and fill out their user profile, including the courses they are attending and their study program code. For students that did not attend any classes this semester, or for other reasons did not want to use their own courses for the test, we provided a set of example data. Based on the feedback from the test users, no serious issues were discovered with the web interface. Everything seemed to work very well, and several participants pointed out that the interactive Ajax-based input fields counted for a much improved user experience. Ajax (Asynchronous JavaScript and XML) is a group of interrelated web development techniques used to create interactive web applications, improving the responsiveness and interactivity of web pages. Instead of having to remember each course code, the system provides matching suggestions as soon as the user starts typing a course name or course code into the text field. This makes the web page seem more responsive and user-friendly as certain sections of the page can be updated without the need for a complete page refresh. See Figure 4.3 for a screenshot illustrating the Ajax input fields.

We got a few suggestions regarding the web administration tool. To require less effort from the user, one participant suggested that the system could automatically fill out the course list based on the semester and line of study applying to the student. This way, the courses would not have to be added manually. Although a good idea, it would complicate things quite a lot because often students get to pick courses from different study programs or departments, and these courses would require manual entry anyway. Therefore, we are in doubt whether the benefit of such a feature would outweigh the implementation costs and implications. Currently, we do not have access to a reliable source to pull this information from, such as which courses are compulsory for each line of study. In the future, however, the schedule service could be integrated with existing information systems at NTNU to automatically obtain the course information for each student.

During the tests, it became evident that the functionality of the first prototype was lacking in one area. In order to explain this, we will first briefly explain how the courses are organized at NTNU and how this aligns with our system. Usually, a course is bound to one or more study programs. The course is only intended for these study programs, and the lectures are scheduled



The screenshot displays a web form for user profile management. The fields are as follows:

- Username:** testuser
- Email:** test@test
- Password:** (empty) (leave empty for...)
- Study program:** MTDT (optional)
- Semester:** 1
- Courses:** A dropdown menu showing a list of courses. The selected course is "informasjonssystemer". Other visible courses include:
 - DT8110-1 Utvikling av informasjonssystemer
 - DT8802-1 Modellering av informasjonssystemer, videregående kurs
 - GEOG1511-1 Kartografi, geografiske informasjonssystemer (GIS) og fjernanalyse
 - TDT4175-1 Informasjonssystemer** (highlighted)
 - TDT4213-1 Kliniske informasjonssystemer
 - TDT4250-1 Modellbasert utvikling av informasjonssystemer

Additional elements include a "Save profile" button, an "Add course" link, and a "Remove" button with a tooltip that says "To remove a course, click on 'Remove'".

Figure 4.3: Ajax-powered MSIS Web Administration Interface

accordingly. For example, the course “Object-oriented Programming” is planned for Computer Science and Communication Technology (among others). We make use of this relationship when the lecture schedule is generated for a particular student. A computer science student enters their respective study program code in their profile, and the system returns matching lecture hours for the course planned for Computer Science. Similarly, a potentially different set of lecture hours are returned for the communication technology student. However, from time to time, the student may sign up for courses which belong to a different study program and this is when the problem occurs. For example, if a computer science student wants to attend a course in psychology, these lecture hours would not show up in the schedule in the first version of MSIS, because they are planned for a different line of study. There are two ways to solve this problem. Obviously, if we remove the internal check that matches the study program code defined in the student profile against the applicable program codes for each course, the problem would be solved. But then, every lecture arranged for that course will be listed in the schedule regardless of the line of study the student is pursuing. This is not particularly useful; especially considering some courses can have in excess of five duplicate lectures each week. This would create a big mess on the small mobile screen. Similarly, if the user chooses not to enter a study program in their profile, the courses will show up as well, but this is not the intended behavior. A more elegant way to solve this would be to allow individual courses to be excluded from the study program check. For example, the web administration interface can offer a checkbox next to the course name, which let the user turn on or off the study program check for that particular course. This way, courses that belong to the student’s line of study would be handled as before, while elective courses are processed according to the new rules. This is a quite important extension as almost every student will attend a course from a different line of study at some point. The final version of the system supports this feature.

The mobile schedule service

After the user profile had been filled out in the previous task, the tester returned to the mobile device. In this part of the scenario, the user starts the schedule service in the MSIS client application, and a time table is automatically generated for the courses that have been assigned to the user profile.

The schedule service worked very well. The lecture information for each course appeared on the schedule with the correct date/time, room number and teacher. Initially, we wanted to integrate our service directly with the database from the administration department, which contains information about all the courses at NTNU. However, after being in contact with one of the responsible for the database it became clear that no such interface currently exists. It was simply not possible to extract the data from a third-party system. Therefore, we decided to use a sample dataset (“snapshot”) from the database instead. After all, it was not crucial for the purpose of our test to work with “live” data.

We got some suggestions for new features and improvements to the schedule service. These suggestions are summarized in Figure 4.4. As the chart in Figure 4.4 shows, we received most feedback regarding the lack of support for courses outside of the student’s study program. Three of the participants specifically requested this feature, however, we believe a lot more users would welcome this addition. The reason why not a larger proportion of the test group commented on this functionality could be that they simply did not attend any courses from a different study program at the time of the test, and therefore the aforementioned shortcoming was not apparent to them at this point.

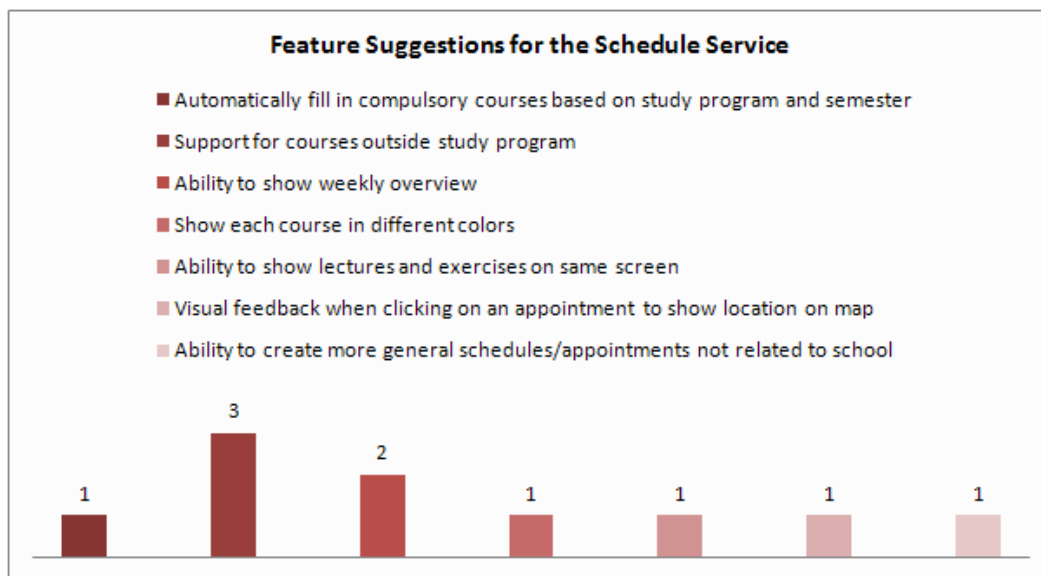


Figure 4.4: Suggestions for new features to the schedule service

Next, two test persons would like the ability to show a weekly overview in addition to the daily schedule. Of course, with the limited screen space it would not be possible to show much information about each appointment for an entire week, but nevertheless, it would be useful to get a quick overview of the weekly appointments without having to click through each day tab. One way to accomplish this would be to indicate on the schedule when there is a lecture, for instance, only showing the room number for the lecture. It should be possible to click on the appointment to show more information. We will consider implementing this feature in the final version of the application.

Regarding the ability to show both lectures and exercise guidance at the same time, we believe this will not work particularly well because of the limited screen estate. With only one or two courses, it could work, but with more courses it would quickly become messy. Instead, the user can switch between a Lectures View and Exercises View. The last suggestion in the chart, the ability to create schedules for non-school related appointments, is not a bad idea and one that could easily be implemented. However, it is outside the scope of this project. The rest of the suggestions are purely visual remarks or things pertaining to the graphical user interface. These issues did not receive high priorities in the early stages of the system, however, they will be considered in the final implementation of the system for enhanced usability.

Discussion

All in all, the feedback we got was very positive. Both the schedule service and the location service worked well considering the application tested was a prototype. The test participants could easily recognize the need for such a service. As expected, we observed that junior students rated the usefulness of the system higher than some of the senior students. This applies in particular to the location service, considering senior students are more familiar with the campus area than new students. They could easily see how the availability of a service like this would have been beneficial when they first started at NTNU, although they might not need it as much today. The schedule service, however, was perceived as useful for both juniors and seniors. During the work with the system, we have also received positive feedback from several staff members as well. With the large amount of meeting rooms at NTNU, it is convenient to have a location search service easily accessible.

As mentioned initially, the test group consisted of people with different backgrounds. We were curious to find out whether a mobile service like this would only appeal to technology savvy users, and if users with less computer skills would hesitate to adopt the service. Our results do not indicate this is the case. Virtually all the participants were positive to the service regardless of their profession. What we did notice, however, was that the technology savvy users gave more comments regarding the technical aspects of the service. They were for instance more interested in the implementation and technical details behind the service than the rest of the test persons. They also paid more attention to small technical subtleties which on the other hand was insignificant for the non-technical users. For example, one computer science student instantly expressed their satisfaction with the auto-complete feature in the administration panel. Others regarded it just as a matter of course.

4.4.2 Descriptive analysis of survey feedback

In this section, the results from the user acceptance survey will be analyzed and discussed. Because the Mobile Services Acceptance Model is still relatively new, this survey also serves to assess the applicability and validity of the research model.

Responses to a Likert scale item are strictly speaking ordinal data, and as such, one cannot assume that respondents perceive the difference between adjacent levels as equidistant. The data have an inherent order or sequence, but one cannot assume that the respondent means that the difference between agreeing and strongly agreeing is the same as between agreeing and being undecided. Because of this, many researchers dissuade from using the *mean* as a measure of item response. For example, it makes little sense to add a response of “Agree” (coded as 5) to a response of “Undecided/Neutral” (coded as 4) to get a mean of 4.5. A more appropriate measure is the median or mode, because they are less sensitive to extreme observations. On the other hand, researchers are sometimes inclined to interpret Likert responses as interval (or “pseudo-interval”) data, which allows more powerful statistical methods to be used [14].

We believe that the computed mean can provide some useful information, in that it gives an indication of where the majority of the responses are centered. In some cases, the median or mode can produce an incorrect picture. For example, in a sample of 15, say 13 respondents strongly agree (coded as 7) with a given statement, one disagree (coded as 1), and one is neutral (coded as 4). This would produce a median of 4 (neutral). If this item e.g. is a measure of the perceived usability of a system, the median statistic does not capture the majority of the sample. Taking the mode of the responses yields 7 (strongly agree) as the “average” response, which is closer to the reality. However, the use of mode introduces other implications. The mode is not necessarily unique, since the same maximum frequency may be attained at different values. This causes ambiguity in the result set.

In the following we will analyze the data obtained from the survey and discuss the results. The tables below use the same labeling for the measurement items as in the questionnaire (see Appendix B). The tables show the percentage distribution of the responses categorized by each construct. The last two rows show the summarized total of the measurement items. Total¹ denotes the average response to that particular construct, based on the mode calculated from each participants’ set of responses. To alleviate the impact of duplicate modes, the summarized mean is included as well (Total²). The last column denotes the average score (mean).

Perceived Usefulness

Table 4.1 presents the results of the measurement items concerning the usefulness of the MSIS application. According to the results, the location search tool was perceived as the most useful service (item PU2), with an average score of 6.68. 96% (24) of the respondents agreed that the system would be useful for them as students, while one respondent did not know if it would be useful. The respondents only partially agreed that the system would better help them to schedule their time. The average score for this statement was 5.52, which still is on the right side of the center (neutral) point. One possible reason that this particular item did not get a higher score might be due to all the other factors that affect the schedule of a student. The MSIS schedule service is not sufficient to get a complete overview of the weekly schedule. After all, that is not within the scope of the current MSIS system. For now, we have only integrated lecture hours and

exercise guidance hours into the schedule service. However, the system may be extended in the future to include other appointments as well.

All in all, the majority of the users perceived the system as easy to learn and interact with. Based on the results, we believe new users would easily be able to adopt the system. The findings from the user acceptance survey confirm the tendency which we described in the discussion of the usability test; junior students rate the usefulness of the system higher than senior students. This became apparent when analyzing the responses to item PU5: “The system would be useful for me as a student”.

Item	Response: (1) Strongly Disagree – (7) Strongly Agree												Avg.		
	(1)		(2)		(3)		(4)		(5)		(6)			(7)	
PU1	0.0%	(0)	0.0%	(0)	0.0%	(0)	8.0%	(2)	24.0%	(6)	44.0%	(11)	24.0%	(6)	5.84
PU2	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	4.0%	(1)	24.0%	(6)	72.0%	(18)	6.68
PU3	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	24.0%	(6)	40.0%	(10)	36.0%	(9)	6.12
PU4	0.0%	(0)	0.0%	(0)	8.0%	(2)	8.0%	(2)	24.0%	(6)	44.0%	(11)	16.0%	(4)	5.52
PU5	0.0%	(0)	0.0%	(0)	0.0%	(0)	4.0%	(1)	8.0%	(2)	24.0%	(6)	64.0%	(16)	6.48
Total ¹	0.0%	(0)	0.0%	(0)	0.0%	(0)	8.0%	(2)	16.0%	(4)	28.0%	(7)	48.0%	(12)	6.16
Total ²	0.0%	(0)	0.0%	(0)	1.6%	(2)	4.0%	(5)	16.8%	(21)	35.2%	(44)	42.4%	(53)	6.13

Table 4.1: Frequency Results - Perceived Usefulness

Perceived Ease of Use

The responses to the items pertaining to the perceived ease of use of the service is presented in Table 4.2. It is interesting to note that no one thought they would have any problems with learning to use the MSIS application. The majority of the respondents (52%) strongly agreed that it would be easy for them to learn how to operate the system (item PEU1). All but one respondent thought the system was user-friendly (PEU5). The user interface was also perceived as clear and intuitive (PEU3); the average response to this statement was 6.24. The most frequent score was “Strongly agree” (coded as 7). Item PEU4 got the lowest average score. Although many users thought the system was flexible to work with, the results indicate that there might be room for some improvement in this area. Another reason could be that respondents were uncertain how to interpret the word ‘flexible’ with regard to the system. During the tests, some of the respondents specifically asked us to elaborate this statement. We did not find any evidence in dispersion of the responses which could indicate that the year of the student had an influence on the perceived ease of use.

Importance of Trust

As previously discussed, the notion of trust is believed to be an important factor for the adoption of mobile services. We proposed seven measurement items related to trust in our user acceptance survey. Table 4.3 shows the results of the Trust construct.

The single most important factor according to the results concerns the reliability of the data provided by the system. 76% of the respondents gave this item the maximum importance, while

Response: (1) Strongly Disagree – (7) Strongly Agree															
Item	(1)		(2)		(3)		(4)		(5)		(6)		(7)	Avg.	
PEU1	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	12.0%	(3)	36.0%	(9)	52.0%	(13)	6.40
PEU2	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	16.0%	(4)	56.0%	(14)	28.0%	(7)	6.12
PEU3	0.0%	(0)	0.0%	(0)	0.0%	(0)	4.0%	(1)	16.0%	(4)	32.0%	(8)	48.0%	(12)	6.24
PEU4	0.0%	(0)	0.0%	(0)	4.0%	(1)	12.0%	(3)	24.0%	(6)	48.0%	(12)	12.0%	(3)	5.52
PEU5	0.0%	(0)	0.0%	(0)	4.0%	(1)	0.0%	(0)	16.0%	(4)	28.0%	(7)	52.0%	(13)	6.24
Total ¹	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	16.0%	(4)	48.0%	(12)	36.0%	(9)	6.20
Total ²	0.0%	(0)	0.0%	(0)	1.6%	(2)	3.2%	(4)	16.8%	(21)	40.0%	(50)	38.4%	(48)	6.10

Table 4.2: Frequency Results - Perceived Ease of Use

Response: (1) Not Important – (7) Very Important															
Item	(1)		(2)		(3)		(4)		(5)		(6)		(7)	Avg.	
TU1	4.0%	(1)	0.0%	(0)	4.0%	(1)	12.0%	(3)	12.0%	(3)	40.0%	(10)	28.0%	(7)	5.60
TU2	8.0%	(2)	4.0%	(1)	8.0%	(2)	16.0%	(4)	32.0%	(8)	28.0%	(7)	4.0%	(1)	4.60
TU3	4.0%	(1)	4.0%	(1)	0.0%	(0)	4.0%	(1)	8.0%	(2)	40.0%	(10)	40.0%	(10)	5.88
TU4	0.0%	(0)	4.0%	(1)	0.0%	(0)	4.0%	(1)	28.0%	(7)	36.0%	(9)	28.0%	(7)	5.76
TU5	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	0.0%	(0)	24.0%	(6)	76.0%	(19)	6.76
TU6	0.0%	(0)	4.0%	(1)	0.0%	(0)	4.0%	(1)	4.0%	(1)	44.0%	(11)	44.0%	(11)	6.16
TU7	0.0%	(0)	0.0%	(0)	4.0%	(1)	8.0%	(2)	0.0%	(0)	32.0%	(8)	56.0%	(14)	6.28
Total ¹	4.0%	(1)	0.0%	(0)	0.0%	(0)	4.0%	(1)	8.0%	(2)	32.0%	(8)	52.0%	(13)	6.16
Total ²	2.3%	(4)	2.3%	(4)	2.3%	(4)	6.9%	(12)	12.0%	(21)	34.9%	(61)	39.4%	(69)	5.86

Table 4.3: Frequency Results - Trust

24% rated it as important (score of 6). These findings align well with our own assumptions; the validity of the data is vital to the success of the system. The respondents were less concerned about the provider of the system. Only one felt that the reputation of the software provider would be a deciding factor for their intention to use the system, however, some respondents agreed to the statement to some extent. The average score obtained for this item were 4.60. Generally, the respondents were split in their opinions on this subject. On matters concerning privacy, the majority of the respondents (80%) rated this as important (40%) or very important (40%).

To sum up, our findings indicate that there is a correlation between trust and the individual's intention to use a mobile service, and thus it confirms the hypothesis proposed by the Mobile Services Acceptance Model. However, there are different aspects of trust which may apply to different type of applications. For example, in our study, the most important factor proved to be the reliability of data. This may have turned out differently if the application subject to evaluation was for instance a service for mobile commerce, for which vendor reputation and recognition is likely to be of greater concern (users are generally more cautious to adopt services which involve financial transactions).

Personal Initiatives & Characteristics

Table 4.4 presents the findings from the construct Personal Initiatives & Characteristics. By looking at the responses to item PIC1 (“I am capable of using the system”), it shows that none of the participants faced any problems which caused them not to be able to use the system. Worth noting is that people in general hesitate to admit their own faults. I.e. they *want* to be capable of using the system. Therefore, one should be careful to draw any final conclusions on the basis of such data alone. However, in our case the responses to the other measurement items in the survey were consistent with these results, so we have no reason to question the numbers.

Item	Response: (1) Strongly Disagree – (7) Strongly Agree							Avg.
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	
PIC1	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	4.0% (1)	44.0% (11)	52.0% (13)	6.48
PIC2	0.0% (0)	0.0% (0)	0.0% (0)	12.0% (3)	12.0% (3)	40.0% (10)	36.0% (9)	6.00
PIC3	0.0% (0)	12.0% (3)	4.0% (1)	12.0% (3)	24.0% (6)	8.0% (2)	40.0% (10)	5.32
PIC4	0.0% (0)	0.0% (0)	0.0% (0)	4.0% (1)	28.0% (7)	36.0% (9)	32.0% (8)	5.96
PIC5	0.0% (0)	0.0% (0)	0.0% (0)	4.0% (1)	12.0% (3)	20.0% (5)	64.0% (16)	6.44
PIC6	0.0% (0)	0.0% (0)	0.0% (0)	8.0% (2)	36.0% (9)	28.0% (7)	28.0% (7)	5.76
PIC7	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	24.0% (6)	76.0% (19)	6.76
Total ¹	0.0% (0)	0.0% (0)	0.0% (0)	8.0% (2)	8.0% (2)	24.0% (6)	60.0% (15)	6.36
Total ²	0.0% (0)	1.7% (3)	0.6% (1)	5.7% (10)	16.6% (29)	28.6% (50)	46.9% (82)	6.10

Table 4.4: Frequency Results - Personal Initiatives & Characteristics

Furthermore, many of the respondents thought the system was somewhat amusing to use. 88% thought it was fun to use the system (score 5-7), while 12% did not have an opinion. These findings can be related to the theory put forth in the previous section, suggesting that the intention to use a service may not depend entirely upon its perceived usefulness; the entertainment factor must be considered as well.

It is also apparent by the results that students would only use the system if it was provided for free (PIC5). Thus, one can argue that the system is a convenient service to offer students, but not as important that people are willing to pay for it. That was never the intention either.

If we look at the two measurement items PIC1 and PIC7 combined, this gives a good indication of both the usability and usefulness of the application. All the respondents felt it was a good idea to use the service (PIC7), and likewise, the test persons were capable of using the system (PIC1). We observe that the respondents do not agree whether they want to be of the first to make use of the service. 16% did not want to be among the first to use the system, 72% was likely to be among the first users of the system, while 12% remained undecided. What we noticed is those that were most eager to use the system early on were technically skilled persons, primarily senior students. This is understandable because people with a technical background are generally more open-minded towards new technology.

Significance of Context

The adoption of mobile services is likely to be more affected by context than traditional desktop applications. As expected, our findings show that students are more likely to use the system if they are in a situation where they do not have access to a desktop computer or a laptop. This is depicted in Table 4.5, which presents the responses to the measurement items related to context.

Item	Response: (1) Strongly Disagree – (7) Strongly Agree									Avg.
	(1)	(2)	(3)	(4)	(5)	(6)	(7)			
CT1	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	12.0% (3)	32.0% (8)	56.0% (14)			6.44
CT2	4.0% (1)	4.0% (1)	24.0% (6)	32.0% (8)	20.0% (5)	12.0% (3)	4.0% (1)			4.12
CT3	0.0% (0)	12.0% (3)	24.0% (6)	12.0% (3)	28.0% (7)	20.0% (5)	4.0% (1)			4.32
CT4	0.0% (0)	0.0% (0)	12.0% (3)	20.0% (5)	28.0% (7)	32.0% (8)	8.0% (2)			5.04
CT5	0.0% (0)	0.0% (0)	4.0% (1)	0.0% (0)	4.0% (1)	48.0% (12)	44.0% (11)			6.28
CT6	0.0% (0)	0.0% (0)	4.0% (1)	0.0% (0)	4.0% (1)	44.0% (11)	48.0% (12)			6.32
CT7	0.0% (0)	0.0% (0)	4.0% (1)	4.0% (1)	8.0% (2)	24.0% (6)	60.0% (15)			6.32
Total ¹	0.0% (0)	0.0% (0)	4.0% (1)	4.0% (1)	16.0% (4)	32.0% (8)	44.0% (11)			6.08
Total ²	0.6% (1)	2.3% (4)	10.3% (18)	9.7% (17)	14.9% (26)	30.3% (53)	32.0% (56)			5.55

Table 4.5: Frequency Results - Context

According to the responses to item CT1, all the participants would use the system if they were out of their office or home. Furthermore, we note that most of the respondents regarded it as important that the system should be easy to obtain and install (CT5). An interesting observation can be made from item CT3, which concerned the users' previous experience with mobile services. 36% of the respondents did not regard this to be a critical factor. 12% were neutral to this matter, while 52% agreed that they would more likely use the system if they previously had had a nice experience with mobile services. The test persons would also more likely use the service if it would be meaningful in the current situation and help increase task efficiency. The least important contextual determinant for user adoption, according to our results, is to which degree other people are using the service (CT2). This shows that people are generally not affected by others' decisions to use a mobile service or not, provided that the service has a value for them.

Intention to Use

In Table 4.6 the responses to the last construct in the survey, Intention to Use, are presented. We have omitted the modular scores from this table as it makes no sense to compute the mode of only two variables. The first row (IU1) shows the proportion of respondents that intend to use the system, assuming they have access to it. 48% strongly agrees that they would use the system, 28% would most likely use it, while 16% thought they might use it. The remaining 8% (2 respondents) was either neutral or unlikely to use it. The last item (IU2) gives an indication of the proportion of users that predict they would use the system provided they had access to it. The results resemble the frequency distribution of IU1, but with a slightly higher average score. The reason for this is hard to say, but it may be that people generally are inclined to *predict* the probability of an outcome (use the system) to be higher than what is the actual probability

for that event to take place. We also got reports during the tests that the participants had some problems to distinguish between the two items, so the discrepancy might just be a matter of mere coincidence.

Response: (1) Strongly Disagree – (7) Strongly Agree								
Item	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Avg.
IU1	0.0% (0)	4.0% (1)	0.0% (0)	4.0% (1)	16.0% (4)	28.0% (7)	48.0% (12)	6.08
IU2	0.0% (0)	0.0% (0)	0.0% (0)	4.0% (1)	12.0% (3)	20.0% (5)	64.0% (16)	6.44
Total ²	0.0% (0)	2.0% (1)	0.0% (0)	4.0% (2)	14.0% (7)	24.3% (12)	56.0% (28)	6.26
Total ³	0.0% (0)	4.0% (1)	0.0% (0)	4.0% (1)	16.0% (4)	32.0% (8)	44.0% (11)	6.04

Table 4.6: Frequency Results - Intention To Use

To get an overview of the percentage of respondents that intend to use the system, we have taken the minimum value of the two responses from each respondent, and then computed the frequency distribution which is depicted in the row labeled Total³. If we consolidate the responses which have a score of 5, 6, or 7, the total percentage of respondents that intend to use the system equates to 92%. One respondent did not intend to use the system, while one was undecided. We chose to use the minimum of the two item responses in order to get the most conservative estimate of user adoption. As expected, our findings indicate a higher intention to use among junior students than senior students.

Figure 4.5 shows how the perceived usefulness of the application relates to intention to use. The graph shows that it might not always be a positive correlation between the perceived usefulness and the intention to use a mobile service. Possible reasons for these findings are discussed in the statistical analysis of the survey results on page 55.

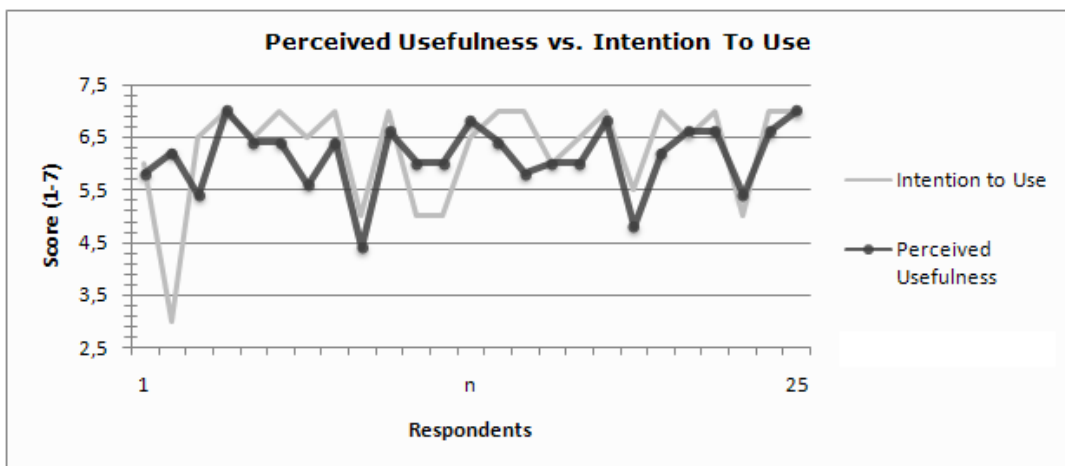


Figure 4.5: Perceived Usefulness vs. Intention to Use

The relationship between the perceived ease of use and intention to use is illustrated in Figure 4.6. It is a clear indication that the ease of use of the application affects the intention to use. The respondents, who rate the ease of use low, do so for the intention to use as well.

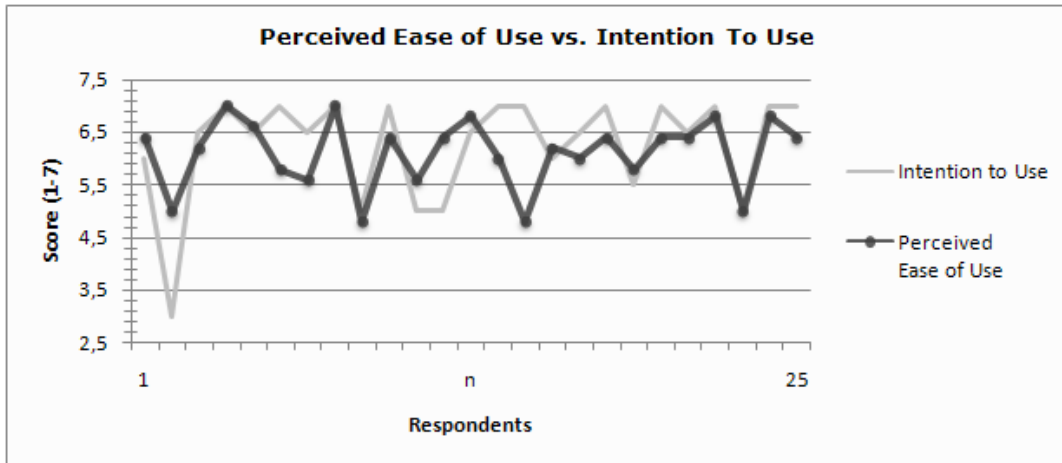


Figure 4.6: Perceived Ease of Use vs. Intention to Use

The chart in Figure 4.7 depicts how the responses to the constructs Perceived Usefulness, Intention to Use, and Trust are distributed across the groups of students. The average score for each group (Year) was computed by (1) taking the sum of all item responses within a construct per respondent, (2) extracting the mean score for each group according to year, and translating the numeric values to a percentage of the maximum attainable score. The figure shows that both the perceived usefulness and intention to use were rated higher by junior students than senior students. The importance of trust does not seem to have a direct correlation with the year of the student. It could seem that the first year students place a little more emphasis on trust than senior students, but the sample size is too small to draw any conclusions for certain. It is more likely that each individual regards the importance of trust in computer applications differently from each other.

Figure 4.8 illustrates how students with a computer science background responds to the measurement items, compared to the responses of non-computer science students. The percentage values are calculated in the same matter as above (Figure 4.7). Our findings indicate that computer science students rate the perceived usefulness and ease of use of the service slightly higher than respondents without a computer science background. This is also confirmed by the results from the Personal Initiatives and Characters construct. The figure depicts that non-computer science students place slightly more emphasis on trust than computer science students. The influence of context is equal between the two groups. A bit surprising is that the intention to use is exactly equal for both groups of students. Hence, it seems that the small difference observed in perceived usefulness and ease of use is not significant.

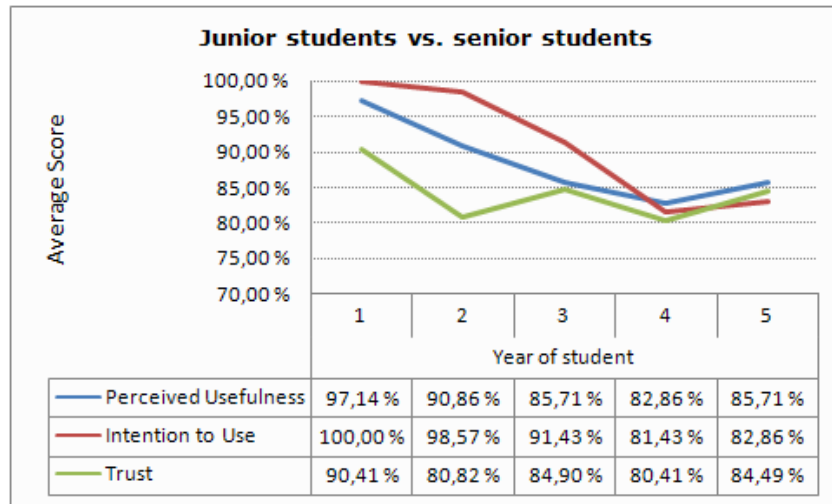


Figure 4.7: Junior vs. senior students

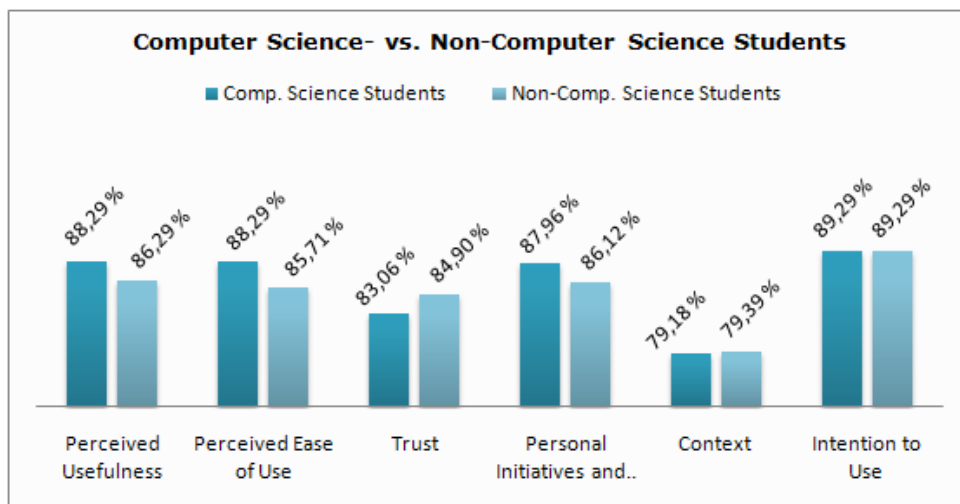


Figure 4.8: Computer science students vs. non-comp. science students

4.4.3 Statistical analysis of the user acceptance survey

This section describes the statistical analysis of the user acceptance survey, based on the Mobile Services Acceptance Model. The measurement result model was obtained by Partial Least Squares (PLS) analysis. PLS was chosen for this study because it fits both exploratory and confirmatory research, places less restriction on the data distribution, and requires a smaller sample size than some other statistical methods (e.g. LISREL) [34]. PLS is a regression-based technique which can be used to examine correlations between measurement items of different constructs.

The data collected in this study was analyzed using the statistical software Smart PLS and SPSS. The PLS analysis seeks to examine the correlations between the different survey constructs. Table 4.7 presents the correlations between the constructs obtained from the correlation analysis. Correlation analysis was conducted to look at the correlation coefficients for all constructs as a means to check for multicollinearity. To avoid problems caused by multicollinearity, correlations among the predictors should be within the range of plus/minus 1, since a weight close to 1 indicates that the two variables are close to being identical [66].

Constructs	Intercorrelations					
	1	2	3	4	5	6
1	1.000					
2	0.671**	1.000				
3	0.358	0.464	1.000			
4	0.831**	0.697**	0.295	1.000		
5	0.191	0.179	0.132	0.138	1.000	
6	0.512**	0.578**	0.329	0.424*	0.205	1.000

1 = Perceived Usefulness, 2 = Perceived Ease of Use, 3 = Trust,
4 = Personal Initiatives and Characteristics, 5 = Context, 6 = Intention to Use
** Correlation is significant at the 0.01 level (2-tailed)
* Correlation is significant at the 0.05 level (2-tailed)

Table 4.7: Correlations between constructs

As Table 4.7 depicts, it was as expected a strong correlation between the perceived ease of use and perceived usefulness. A system which is easy to learn and use is naturally perceived as more useful. Davis [23] suggested that perceived ease of use has a direct influence on the perceived usefulness, which is confirmed by our results. Furthermore, significant correlations were identified between personal initiatives and characteristics and perceived usefulness, and between personal initiatives and characteristics and perceived ease of use.

The structural measurement model generated by the PLS algorithm is presented in Figure 4.9. This is a graphical illustration of the MSAM research model, where each path corresponds to one of the hypotheses in section 4.3. The arrows on the figure indicates a relationship between two constructs, i.e. to what extent the response of one construct is likely to affect the response of the other. The path coefficients as obtained through PLS regression are noted on each arrow. The number within the circles denotes the coefficient of determination, R^2 . It provides a measure of how well future outcomes are likely to be predicted by the model, the amount of variability of a given construct. R^2 is a statistic that gives some information about the *goodness of fit*¹ of the model. In our PLS analysis, the coefficient of determination is a statistical measure of how well the regression coefficients approximates the real data points. An R^2 of 1.0 indicates a perfect fit. The amount of variance in intention to use explained by the model was 56.7%. The explained variance of the Perceived Usefulness and Perceived Ease of Use constructs were 53.8% and 18.6%, respectively.

¹The goodness of fit of a statistical model describes how well it fits a set of observations.

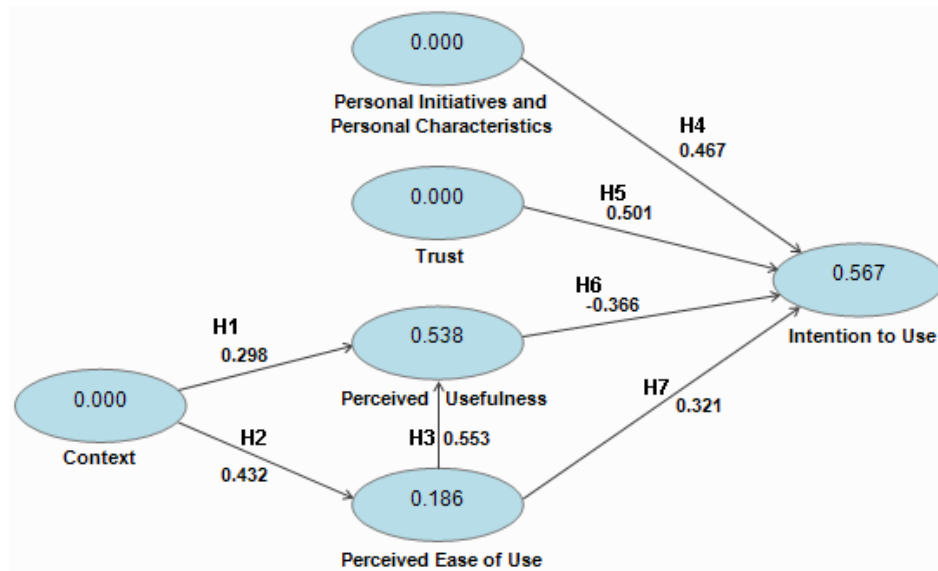


Figure 4.9: The structural measurement model

According to the results, Personal Initiatives and Characteristics, Perceived Ease of Use, and Trust, are direct determinants of Intention to Use. Among all constructs, trust has the strongest impact on intention to use. Previous research work [75, 32] also confirms that the notion of trust is becoming increasingly important for gaining customer confidence in the field of mobile computing. Our results confirm six of the seven proposed hypotheses. Some initial observations indicated that hypothesis H6 was to be rejected, which called for further research. According to our results, the perceived usefulness in some cases seemed to affect the intention to use negatively, which is contrary to previous research findings of similar technology acceptance studies. However, further analysis revealed that this result was not statistically significant; hence we cannot draw any final conclusion with respect to this hypothesis. Despite of this, we can think of a few plausible factors which may result in such behavior:

- The sample size of this pilot study is quite small. This could potentially accentuate irregularities in the responses which otherwise would not have been as prominent with a larger sample size. However, as the primary focus of this pilot study is to conduct a usability and user acceptance test of the MSIS application, and not to produce scientific proof for the validity of the MSAM instrument, we do not feel this will be a problem.
- The respondents consisted of students majoring in engineering, which naturally have an interest in new technology and digital services. The usefulness of the application may not be as determinative for the user's adoption of the system as one might anticipate. Other factors might be more influential on the intention to use. As described in the previous section, several of the participants expressed excitement and joy while using the application. Perhaps the entertainment factor is more important to the user than actual usefulness. However, this does not mean that the MSIS application is not useful, as descriptive results show quite the opposite. It is merely an indication that the perceived usefulness may not

be deterministic to user adoption for the specific respondents in our study.

- The year of the student: senior students may appreciate the need for the application, but to a lesser extent see that this is any longer useful for them (but would have been useful when they were new students). Hence, they rate the usefulness of the system high, while they might not have any intention to use the system themselves. Junior students may not appreciate all the functionality, but would be more interested in using the system since it had sufficiently with functionality to be useful according to their needs. Personal interviews with some of the survey participants confirmed these assumptions.
- Lastly, for the purpose of this pilot study, the version of the MSIS application that was tested run on Windows Mobile-based high-end devices (HTC Touch), which quite few students own or would consider buying because of the relatively high costs involved. Thus, even if the service as tested in the experiment was perceived to be useful, some might have regarded it to be unlikely that they would be able to use it themselves.

Findings by Robinson et al. [69] also indicates that perceived usefulness may not be directly related to intention to use, in a cross-sectional survey among sales organization members.

4.5 Validity of the results

When doing usability testing with mobile devices it is hard to monitor the participants and system responses due to the limited screen size and portability issues. Kaikkonen et al. [44] discusses mobile video recording systems that can be attached to the screen to facilitate field and laboratory usability testing of mobile applications. However, we felt that this was not necessary in our study as such devices can also be an impediment on the use of the system. Instead, we included a few control questions in our tasks for each scenario in order to measure the validity of the data returned from the system. For example, to verify that the positioning module returned the correct location, the participants were asked to note down a code that was only returned as part of the correct location. This way we would avoid “false positive” feedback that could potentially occur if the test persons were just after “picking the right answer”.

Although a sample size of 25 might be a bit small from a statistical point of view, the main objective of this user test was to reveal usability issues with the system and capture qualitative feedback from potential users. According to a paper by Faulkner [30], a sample size of 20 users was sufficient to find 95% of all usability problems.

Lastly, the respondents were from the engineering departments of the university, which means that the results do not represent views from other departments. Hence, the generalization of the results to other potential users remains to be determined. The students also used the application for a limited period of time. A longitudinal study with a larger sample size may reveal other aspects such as cultural relationships, or changes in user behavior over time, which this study did not expose.

Reliability analysis of measurement items

To measure the reliability and validity of each construct in the mobile service acceptance model, the Internal Consistency Reliability (ICR) of each construct was tested with Cronbach’s Alpha

coefficient. The ICR is a method used to measure the reliability of different survey items intended to measure the same characteristic. For each respondent the smaller the internal variability within the same construct, the greater the internal consistency reliability of the survey instrument. According to previous research work [68], a reliability coefficient of 0.6 is marked as the lowest acceptable limit for Cronbach's Alpha for exploratory research. Table 4.8 shows the reliability coefficients for each of the constructs in our measurement model.

Constructs	Number of items	Cronbach's α
Context	7	0.622
Personal Initiatives and Characteristics	7	0.628
Trust	7	0.689
Perceived Ease of Use	5	0.829
Perceived Usefulness	5	0.829
Intention to Use	2	0.906

Table 4.8: Reliability analysis of constructs

As the table shows, all of the constructs have a reliability coefficient above the defined threshold value of 0.6. Thus, the accuracy of the model is deemed adequate. A relatively large number of measurement items were used for some of the constructs to allow sufficient room for item refinement in future research. The lower reliabilities for the constructs (Context, Personal Initiatives and Characteristics, and Trust) can be partly attributed to larger number of items in the scale. To further improve the reliability of the model, some of the measurement items may be revised or eliminated in future studies.

Sources of distortion

Likert scales may be subject to distortion from several sources, which may influence the results of the study. Respondents may avoid using extreme response categories (central tendency bias); agree with statements as presented (acquiescence response bias); or try to portray themselves or their group in a more favorable light (social desirability bias). Designing a scale with balanced keying (an equal number of positive and negative statements) can obviate the problem of acquiescence bias, since acquiescence on positively keyed items will balance acquiescence on negatively keyed items, but central tendency and social desirability are somewhat more problematic. We did observe central tendency bias to some extent, but a remarkable large portion of the responses were to the right-most, i.e. a score of 7 (Strongly Agree). It is possible that this is a result of social desirability bias: the respondents wanted to agree with statements in order to please the experimenters. Or, in the case of the Ease of Use construct, they wanted to give the impression of being capable of using the system.

4.6 Summary

The evaluation was successful in identifying problems and issues with the prototype. We got lots of useful feedback from the participants regarding the service, and the general perception is that the functionality provided by MSIS is of great value to the students at NTNU. The findings from

the user acceptance survey indicate that the perceived usefulness of the application is rated higher by junior students (1-2 year) than senior students. The results also indicate that junior students are more likely to adopt the service than senior students. The senior students, however, may acknowledge the need for the service, but do not see it as likely to use it themselves considering they have been students for a while, and therefore are more familiar with the NTNU campus. They could see the value of such a service for new students, though. This may also be one of the reasons for the large amount of variance in Perceived Usefulness (53.8%) and Intention to Use (56.7%) explained by the structural measurement model (Figure 4.9).

Looking back at the survey results from last year, about 40% believed they would use the service when first presented with the idea of a mobile application like MSIS. The results of our evaluation suggest a considerably higher portion of users willing to adopt the system. Hence, it could seem that the students are more likely to adopt an information system when they have a clear understanding of its functionality, and get a chance to try it out for themselves. According to the TAM [23], a strong indication of intention to use does not implicitly stipulate actual use. Hence, the actual user adoption of the system might be higher or lower than what our measurements indicates. However, our results should be regarded as a good indication of users' willingness to use the system. The MSAM addresses some of the limitations of the TAM by also accounting for environmental, personal and organizational aspects, which might constrain the user adoption of information technology.

In our analysis of the correlations between the different survey constructs, we note that there is a strong correlation between the Perceived Ease of Use and Perceived Usefulness of the application, which confirms the findings of Davis [23]. Furthermore, Personal Initiatives and Characteristics, Perceived Ease of Use, and Trust, are direct determinants of Intention to Use. The results from the PLS analysis depicts the notion of Trust as being of strongest influence on Intention to Use. We did not find a positive correlation between Perceived Usefulness and Intention to Use. Possible explanations for this were suggested in the previous sections. When comparing the responses according to the respondents' professional occupation, our findings indicate a slightly higher perceived usefulness and ease of use among computer science students than non-computer science students. No dispersion was found in intention to use across the two groups.

4.6.1 Directions for next iteration

During the evaluation phase, a few shortcomings with the initial design were identified. We also received several suggestions for new functionality. This section lists the new features and enhancements that will be implemented in the final iteration of the MSIS design cycle. The decision of which features to include in the revised version was made by prioritizing the items according to the number of requests received. Some of the additions are also a result of our own ideas. The final version of MSIS is described in Chapter 5.

Contextual search

The location search service should be extended to incorporate contextual information. The current position of the user should be taken into account when querying the database for matching locations. For example, if the user submits a search for nearby computer labs, the system should

transparently calculate the position of the user, and return the X nearest computer labs. We believe this will enhance the user experience and increase task efficiency.

Improvements to the schedule service/calendar

The schedule service should be extended to allow a weekly overview of appointments to be generated. The new view should indicate when and where lecture hours or exercise guidance hours take place for the current week, without the need to tab through the weekdays. The final version of the service should also have support for courses outside of the user's study program. Some general visual improvements should also be made.

Desktop version of location search service

A desktop version of the location search service should be developed. The application should be web-based, and offer the same functionality as the mobile version, but customized to the desktop platform. This will allow users of MSIS to access the service from an ordinary computer. The purpose of this is also to explore how clients on different platforms can make use of the existing data sources and shared web services. It gives an indication of the flexibility of the system.

Improved error handling and bug fixes

All areas of the system should cope with errors gracefully, and no known exceptions/issues should cause fatal errors or failures which result in abnormal termination of the application. Furthermore, any bugs discovered during the evaluation phase should be rectified.

Visual enhancements

The look and feel of the client application and the web-based administration interface should be improved. This includes usability elements and simplified functionality. This will, however, have lower priority than the functional enhancements described above if time is running short.

Chapter 5

Presentation of the final system

“Software and cathedrals are much the same – first we build them, then we pray.”

–Anonymous

This chapter addresses the weaknesses discovered during the evaluation of the system, and presents the final version of MSIS. In addition, a number of screenshots are given illustrating the progress since the initial prototype which was described in section 2.2. The evaluation described in the previous chapter concerns the first major iteration of the implementation, while this chapter presents the outcome of the final iteration. In accordance with the design-science paradigm, the suggestions which were received during the evaluation have been applied to develop a better service. Section 5.5 concerns the evaluation of the final iteration.

5.1 New functionality

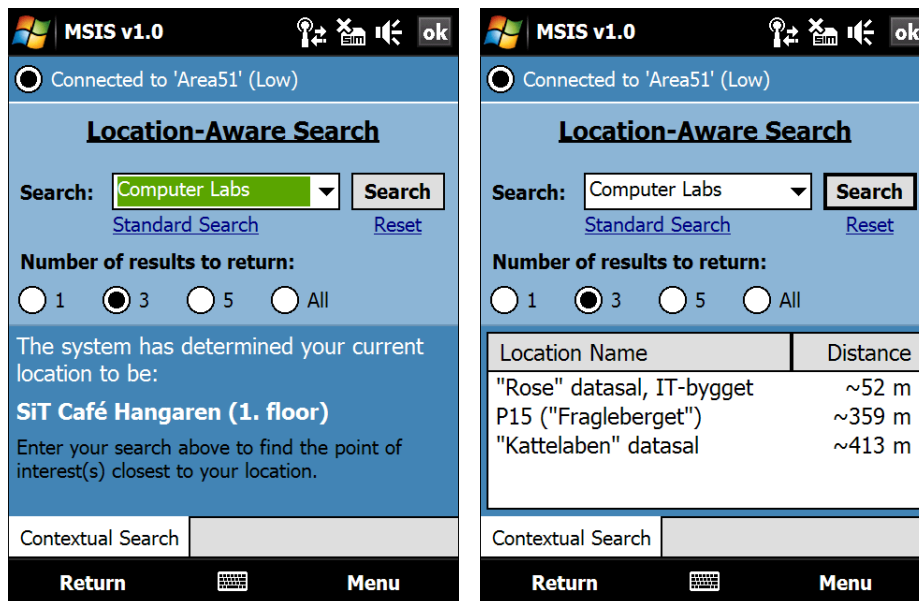
The prototype developed last autumn had basic functionality for location search and schedule planning, but were also lacking in several areas as discovered as part of our evaluation phase. This section describes the new features of the final system, and discusses how issues and problems have been addressed.

5.1.1 Contextual location search

A new service called “Contextual location search” has been developed. The service is an extension of the existing location search service, which additionally makes use of context in order to provide more informed search results to the user. The contextual information that has been incorporated into the system is *location*. The idea is that this will allow the user to find relevant locations in close proximity more quickly. The survey conducted by Haugvik [39] in 2006 showed that students would like to have such a service, and this was also confirmed by our preliminary study.

How it works

When the user starts the location search service, they are presented with a choice between the standard location search tool and contextual search. When the contextual search is selected, the system attempts to determine the position of the user by using the built-in Wi-Fi positioning module. The user may then search for different “point of interests” such as canteens, meeting rooms, or computer labs as depicted in Figure 5.1(a). The search preferences are submitted to the application server, together with the user’s location. The application server then consults the location database for matching locations, and sorts the results in order of distance from the user. This is done by computing the Euclidean distance between the user’s current position and the particular location. The distance from the user to the location is returned as part of the results, providing a useful indication of how far away the points of interest are. In order to accommodate this, we had to extend the map module to take into account the scale of the map. Figure 5.1(b) shows how the results are presented to the user. Because our application supports multiple zoom levels, a scale is associated with each map sheet. The distance vector is scaled according to the zoom factor to give a more correct proportion between the distance values. It is worth mentioning that the sample data we used for the testing did not have accurate measures and map scales. However, the functionality and technology is in place.



(a) Location-aware search

(b) The search results

Figure 5.1: Contextual Location Search

In a similar fashion to the standard location search, it is possible to show the locations on a map in the contextual search as well. The map module were extended to display two pins, denoting the user’s current position and target location connected with a line (see Figure 5.2). By clicking on one of the pins, it is possible to zoom in on that location for a more detailed view. If both locations (current and target) are in the same building or floor section (i.e. visible on same map),

the pin positions of both locations update automatically as the zoom level change. When the map is first loaded, the zoom level is set to the maximum allowable value while both locations are still visible on the same map. For example, if the user searches for a computer lab, and the nearest one is next door, it is needless to open the map in overview mode.

Together with the new contextual search functionality, we also added a network status indicator to the application. During the user tests, we discovered that the system did not cope very well with intermittent network disconnections. The system now periodically checks the status of the Wi-Fi network connection, and in case it is lost it informs the user. The new version also offers significantly improved exception handling to allow the system to recover from sudden network problems. This is a quite frequent issue in mobile computing, so it was important to minimize the impact of such events.

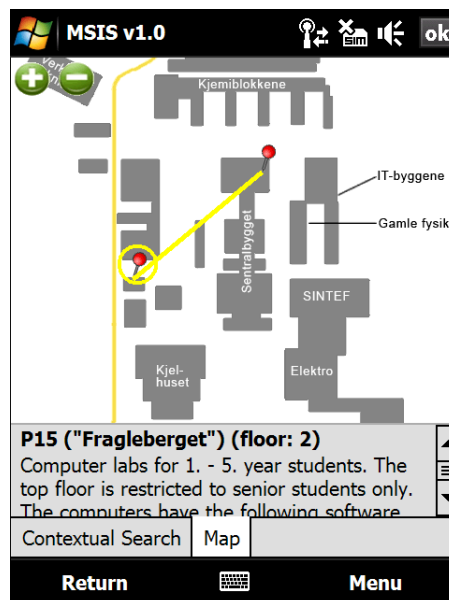


Figure 5.2: Contextual Location Search - Map

Limitations

In order for the system to be able to determine the distance from the user's current location/position to the requested point of interest, the two locations must share at least one common map sheet. If the locations does not both have a coordinate position on a sufficiently high-level map it is impossible to tell how far they are apart. This is not an issue within the borders of the campus, though, because all locations should have coordinate positions relative to a campus overview map at their initial zoom level.

Secondly, if the user's location cannot be determined (for example, they are outside of the Wi-Fi signal coverage area, or the position has not been mapped to a location in the location database), contextual search is not available. This is obvious because there is no way the system

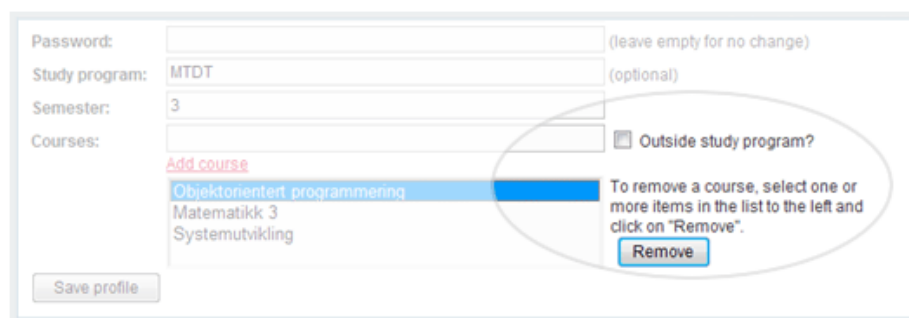
is able to calculate the position of the user relative to other locations if the current position is not known.

5.1.2 Improved schedule service

During the usability- and user acceptance tests, we identified some issues with the schedule service as it appeared in the prototype. This had mainly to do with the support, or rather lack of support for courses outside the study program which the student is part of. Often, students attend courses which either does not belong to a particular study program, or courses belonging to a study program other than their own. Therefore, it is crucial to the success of the service that this issue is resolved.

How it works

To accommodate the aforementioned situation, the solution we came up with is one that allows students to exclude certain courses from the study program conditional check. Hence, courses that are scheduled for the student's study program are retrieved in the usual matter, while the lecture hours for the "excluded" courses will show up in the schedule regardless of the study program settings. A new Boolean field has been associated with the course list and Appointment class, which indicates whether a particular course is part of the student's study program or not. It was important to implement this in such a way that requires minimal effort from the user. Figure 5.3 shows how it has been implemented in the administration interface. When a user adds a new course to their profile, they get the option to specify that the course is outside their study program. On the other hand, if they only have mandatory courses, the course entry requires no additional effort.



The screenshot displays a user profile management interface. It includes fields for Password, Study program (MTDT), Semester (3), and Courses. A checkbox labeled "Outside study program?" is checked. Below the courses list, there is a "Remove" button. A "Save profile" button is at the bottom left. A red "Add course" link is visible above the course list. The course list contains "Objektorientert programmering", "Matematikk 3", and "Systemutvikling". A blue highlight is under "Objektorientert programmering". A tooltip on the right says: "To remove a course, select one or more items in the list to the left and click on 'Remove'." A red oval highlights the "Outside study program?" checkbox and the "Remove" button.

Figure 5.3: Support for courses outside study program

Another enhancement made to the schedule service is the introduction of a weekly summary view. Often, users are only concerned about whether or not they have an appointment at a given time slot, and therefore it is more efficient to be able to view all lectures and exercise guidance hours for the entire week on a single screen. The challenge in this regard is the limited screen space available on a mobile device. In order to display as much information as possible in a clear

and compact way, we created the interface which is depicted in Figure 5.4(a). Only the type of appointment (Lecture *L* / Exercise *E*) is displayed together with the room number. By clicking on an appointment, it is possible to view the details of the appointment as illustrated in Figure 5.4(b).

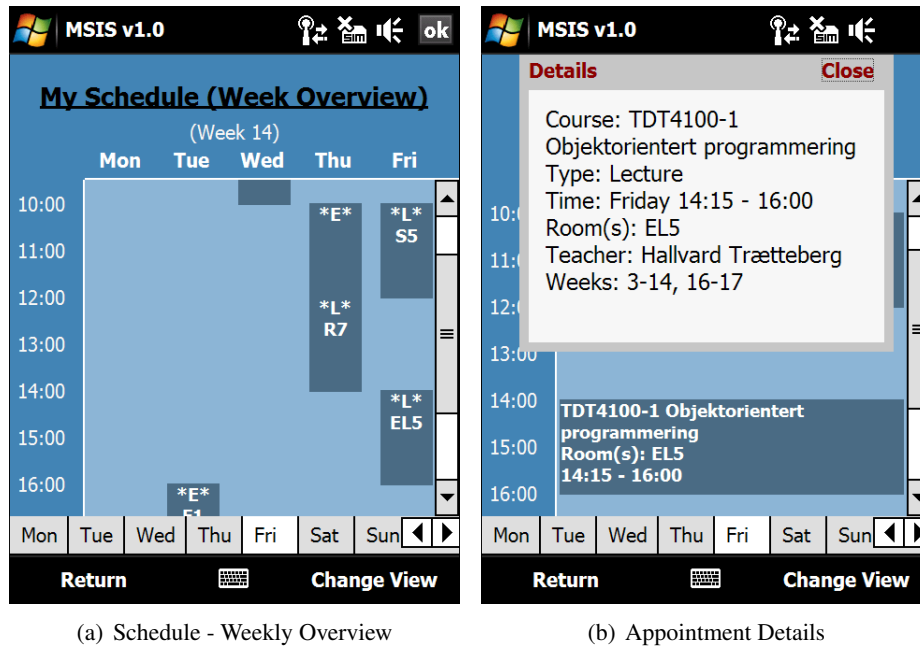


Figure 5.4: The improved Schedule Service

One minor issue which was also discovered during the evaluation is that some courses may not have lectures planned for certain weeks. The new version of the schedule service addresses this by taking the current week number into account when looking up the lecture hours. In addition to the improvements mentioned above, the look and feel of the graphical user interface has been given a substantial make over in order to be more streamlined with the rest of the application.

5.1.3 Desktop version of location search service

In order to explore the flexibility of the MSIS architecture, we have implemented a desktop client application for the location search service. The application provides the same functionality as the mobile location search service, but is accessible through a standard web browser. Because of the flexibility made possible by the web services technology, we were able to create a new client for a completely different platform with relative ease. Much of the existing methods and objects could be reused, keeping the amount of duplicate code to a minimum. Figure 5.5 shows a screenshot of the application. The web application only supports standard textual search and not the context-aware search functionality. Standard browser/web technology does not allow direct access to the low-level network APIs of the operating system. Hence, it is not possible to examine the wireless signals in order to construct a “fingerprint” for the location. Besides, desktop computers are usually stationary or at least not as portable as a mobile phone, which partly contradicts the

premise of the contextual search feature. The location fingerprinting technology is described in section 6.3.

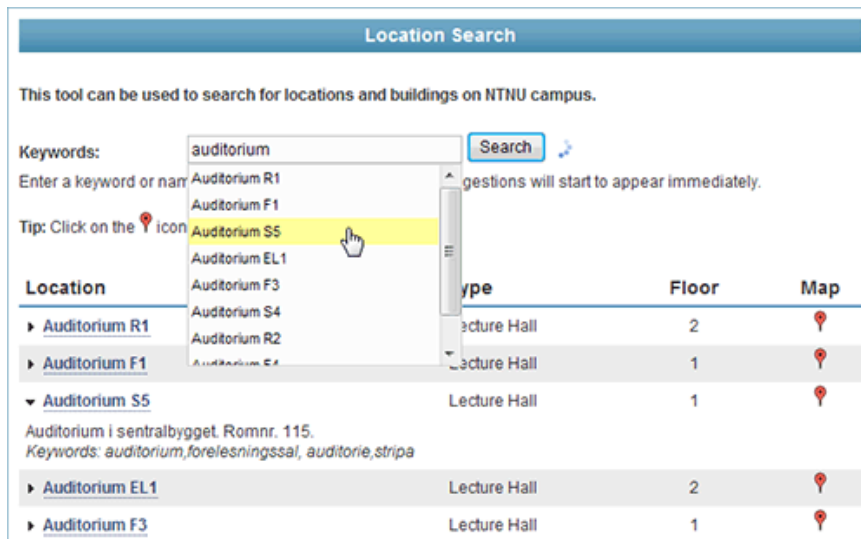


Figure 5.5: Web-based Location Search Service

5.2 Better exception handling

The prototype of MSIS that was installed on the test devices had very limited error recovery mechanisms. Since the wireless transmission used in mobile computing is more exposed to interruptions and disconnections, the application should provide exception handling logic in order to deal more gracefully with unexpected issues that may occur during the operation of the system. In case of the first prototype, a network problem would generate an exception in the communication protocol, which would further propagate to the rest of the system ultimately leading to a fatal error and termination of the application. By accounting for such issues beforehand, we can instruct the application to take necessary actions to limit the impact of the error.

The solution we have implemented is to continuously monitor the network status, and if a problem is discovered, the functionality relying upon network communication is disabled until the network connection is restored. Often, a disconnection in wireless networks only lasts for a few seconds, for example, if the user moves outside the coverage area of a transmitter or the signal is intercepted in some way. Thus, with the improved exception handling support such problems are less severe. A status indicator was also implemented, which gives the user information about which network they are connected to, and the current signal strength. If a drop in signal strength is noticed, they can move to a place with better reception. Figure 5.1 shows the network status indicator in the top section of the screen. In this case, the device is connected to the network “Area51” and the signal strength is “low”.

The application server has also been modified to catch any exceptions before they propagate to the client application. Exceptions may arise from database connectivity issues or incorrectly formatted data.

5.3 Visual enhancements

In the first version of the application, the main focus was to develop a working prototype, and we did not spend much time on the graphical user interface. We have made some adjustments to the user interface in the new version in order to incorporate some of the feedback received during the evaluation phase. This concerns things such as font sizes, shapes and color combinations. The initial user interface consisted of a solid black background, which caused our application to blend in nicely with the operating system interface and exterior of the HTC Touch mobile device. However, we got some comments that it was hard to differentiate between what was part of our application, and what was part of the core operating system (buttons and other elements).

The screenshots in the previous section show the new, improved user interface. The structural layout and main design elements have been kept from the initial prototype, but the interface has been given a new look and feel. The usability test revealed that the interface was perceived as intuitive and easy to understand, so we did not want to make any radical changes. A new graphical design for the main menu has also been created. These improvements have been implemented in order to better comply with the 6 principles of user interface design proposed by Constantine and Lockwood [19, 20]. Thus, the perceived usability of the system should increase as discussed in section 3.3. Figure 5.6(a) shows the new main menu GUI.

5.4 Other enhancements

During the analysis of the user acceptance survey, we noted that the majority of the respondents agreed that they were more likely to use the system if it was easy to install and configure. A new “Configuration” menu has been added to the system, which makes it possible to specify various settings such as the web services URL and other options at runtime. When the application is started for the first time, the user will now be prompted to specify the URL to the application server to ensure the application is correctly set up. The URL parameters were previously hard-coded into the application, which is rigid and hard to change in case the application server is moved to a different address. The new configuration screen is illustrated in Figure 5.6(b). The application has also been packaged into a CAB auto-installer for convenient distribution and installation. We believe these enhancements should increase user adoption of the system.

5.5 Evaluation of the final system

After the last implementation iteration, a small evaluation of the final system was conducted among some of the people that previously had participated in the usability test and user acceptance survey. The final evaluation serves to assess whether the new features and enhancements successfully address the issues described in Chapter 4: Evaluation.

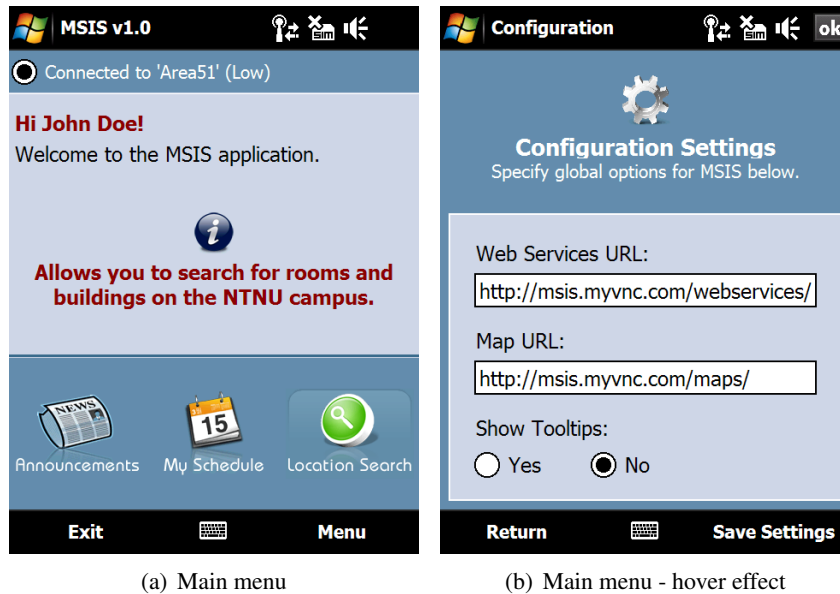


Figure 5.6: New main menu and configuration screen

The evaluation was an informal one; we asked a few of the test users, which previously had expressed concerns regarding parts of the system, to test the new version with emphasis on the new features. The revised version of the MSIS application was installed on the device, and handed out to the users. The same device was used, and the tests were conducted in the same environment as before, to limit the likelihood of new factors influencing the results. The participants all agreed that there was a clear improvement over the first version of the system. Especially, some of the students that were attending courses outside their study program perceived the new schedule service as more useful, since all their lectures now would show up in the calendar. The new weekly summary view was also appreciated by the test persons. This particular feature was implemented as a direct result of the user feedback, and the students appreciated that their opinions were taken into consideration when formalizing the final feature set.

The project was also presented to a few members of staff and fellow students at NTNU, in order to gather valuable feedback regarding the feasibility of the system and potential issues. This relates to item 7 of the Design-Science Research Guidelines described in Chapter 3: Research Approach, which states that IS research should be communicated to both technology-oriented and management-oriented audiences.

5.6 Video demonstration of the system

A video demonstration of the services offered by the MSIS application is available in Appendix D.4. The video has been created using the VH PocketPC screen capturing software [42], and shows the application in use when operated by a user.

Chapter 6

Design and implementation details

“Art has to move you and design does not, unless it’s a good design for a bus.”

–David Hockney

A description of the MSIS architecture and the different modules comprising the system were given in section 2.2. The overall architectural design is carried on in the revised version. In this chapter we will discuss the web services in more detail and the functionality they provide. This is important in order to understand how the client application interacts with the server, and can be used as a reference for developing clients for different platforms, or to extend the system in other ways. We will also explain with an example how the services can be used to create a working application.

6.1 Design goals

The MSIS system is divided into modules, which each are responsible for providing a set of services and features. The core functionality (business objects and logic) and data storage (relational database) constitute the backend of the system. Functions are encapsulated into business objects, which are published as ASP.NET Web Services. This makes it possible to develop client applications for different platforms without having to duplicate the business logic.

When designing the system we have adopted the notion of loose coupling, which was introduced by Karl Weick as part of his organizational studies [82, 83]. In computer science, coupling is a software quality metrics which refers to the degree of dependency between different software modules. Modules with loose (or low) coupling make their requirements explicit through a set of clearly specified interfaces, without making assumptions about the other application modules. Thus, integration between modules can be made with minimum risk that a change in one module will force a change in interconnected modules. A loosely coupled system is easy to maintain, because the program modules does not make detailed assumptions about the internal implementation of the other modules. The concept of loose coupling is often correlated with semantic coherence. Semantic coherence refers to the relationships among responsibilities in a module.

The goal is to ensure that all of these responsibilities work together without excessive reliance on other modules [11], referred to as *high cohesion*. A system with high cohesion tends to have more self-contained modules, with a clear separation of functionality between modules. This has several advantages such as increased robustness, reliability, reusability. Systems with high cohesion are also easier to understand and maintain than complex systems where functionality often is heavily dispersed.

MSIS is designed in accordance with a Service Oriented Architecture (SOA) approach, which aims at separating functions into loosely coupled services [13]. The services are implemented using the Web Services standard, and communicate with each other by passing data from one service to another, or between one or more service and its consumers. Newcomer and Lomow [59] describes how web services can be used to realize a service oriented architecture. The SOA services expose their functionality via publicly available interfaces that other applications or services can understand in order to utilize those services. The service provider and service consumer/requester communicate with each other using a well defined protocol such as SOAP, Jini or CORBA [27]. The web services implemented in MSIS use SOAP for structured message exchange with XML as its message format. SOAP implementations are available for many different programming languages and platforms, which makes it a very versatile protocol. SOAP has become a de-facto standard for web service communication, and it is also recommended by the W3C¹ international standards organization. The web services are implemented on top of other application layer- and transport layer protocols (HTTP over TCP/IP in MSIS) for lower-level message negotiation and transmission.

6.2 The web services

In total there are four web services, which each are responsible for a certain set of functionality. Functionality is divided in such a way that things that share a common element are kept together, to keep in accordance with the goal of *high cohesion and low coupling*, which are attributes of good software design [64, 40].

The naming convention used for the web services is **Service.asmx*, and **Service.cs* for the code-behind, respectively. The following web services are available: *UserService.asmx*, *LocationService.asmx*, *ScheduleService.asmx*, and *AnnouncementService.asmx*.

6.2.1 UserService

The *UserService* web service is responsible for account management and login authentication functionality. The service provides the following methods:

Login(string username, string password) : The method accepts a username and a SHA1-encrypted hash sum of the password, and performs login authentication against the database of known users. If the user is successfully authenticated, a *UserProfile* object is returned to the requester containing the details about the user. If the login fails, an empty *UserProfile* object is returned with the property 'IsAuthenticated' set to *False*. The SHA1 value must be

¹World Wide Web Consortium

generated before calling the web service to prevent transmission of the password in plain text.

UpdateUser(*UserProfile user*, *string newPassword*) : The method may be used to modify the profile of an existing user. The method accepts a *UserProfile* object and optionally a new password (encrypted using the SHA1 algorithm) for the account. If the password should remain unchanged, just pass an empty string. The method returns *True* upon success and *False* if the update failed.

6.2.2 LocationService

The LocationService web service provides the core functionality necessary for the location search service. The web service declares two data structures - **Signal** and **Fingerprint** - which are used throughout the application. The Signal structure contains the SSID (Service Set Identifier), BSSID (Basic Service Set Identifier), and RSSI (Received Signal Strength Indication) of a wireless access point. The Fingerprint structure contains a Signal vector and a location ID. The ID refers to the unique identifier of a location in the database.

The following are the main methods provided by the web service:

GetLocationByName(*string searchString*) : This method accepts a search string of arbitrary length and returns a list of locations matching the search string. The 'name', 'description', and 'keywords' fields associated with the location are included in the query.

GetLocationByFingerprint(*Signal signalVector*) : This method queries the database for locations with fingerprints matching the characteristics of a given signal vector (usually the observed signal), in order to add location-awareness to the application.

AddLocation(*Location location*, *Signal signalVector*) : This method can be used to add location fingerprints to the database. The method accepts a *Location* object with information about the location, and the corresponding signal vector which constitutes the fingerprint. The method returns *True* if the location was successfully added, or *False* if an error occurred.

GetNearbyLocations(*LocationTypes locationType*, *Signal signalVector*, *int locationsToReturn*) : This method returns a list of locations in close proximity to the position where a given signal vector was observed. The *locationType* parameter specifies the type of locations to search for (must be a member of the *LocationTypes* enum declared in the *Location* class), and the last parameter specifies the number of locations that should be returned. The list of locations is sorted in descending order according to their distance from the signal vector.

6.2.3 ScheduleService

The ScheduleService provides functionality for the schedule service. Currently, it provides methods for getting lecture hours and exercise guidance hours for students' courses. However, in the future it can be extended to also support other type of appointments.

The following methods are available:

GetStudyProgramList(string *prefixText*, int *count*) : This method returns a list of study programs. If a prefix is provided, only study programs matching the prefix are included in the list. If the *count* value is specified, the list is limited to *count* number of elements. This method is *Script-enabled*, which allows it to be invoked from a script using ASP.NET Ajax.

GetCourseList(string *prefixText*, int *count*) : This method returns a list of available courses. If a prefix is provided, only courses matching the prefix are included in the list. If the *count* value is specified, the list is limited to *count* number of elements. This method is *Script-enabled*, which allows it to be invoked from a script using ASP.NET Ajax.

CourseExists(string *courseCode*) : This method can be used to check whether a course with a given course code exists. The method returns *True* if a matching course is found, or *False* otherwise.

GetLectures(string *username*) : This method returns all the lectures and exercises for courses the given user is attending. The method accepts a username, and returns a list of *Appointment* objects.

6.2.4 AnnouncementService

The idea of the announcements service is to deliver relevant information regarding the study to the students. Initially, we thought of implementing a service incorporating important deadlines, information about upcoming events, and other information pertaining to the student. However, it was decided not to pursue this idea any further as it would require tight integration with existing information systems (e.g. “Innsida”) to be of any real value. Instead, we have focused on the other two services: the location search service and the schedule service.

For more details about the classes, class members, and objects, please refer to the application source code in Appendix D.

6.3 Geographical positioning of users

The MSIS system can be used to estimate the geographical position of a mobile device within a wireless Wi-Fi network. In this section we will describe the internal workings of the positioning module, and how new locations are mapped to coordinates.

6.3.1 General

The positioning technology is based on a concept known as *location fingerprinting*. The Wi-Fi location fingerprinting approach consists of two parts: a data collection task in which the geographical positions are mapped to wireless signal strength vectors in order to generate a database of fingerprints. This is sometimes referred to as the offline phase [47]. When a sufficiently large

fingerprint database covering the target area has been generated, the actual positioning is done during what is referred to as the online or real-time phase [47].

6.3.2 The offline phase

In order to map a position to its geographical coordinates and a descriptive location, we need to scan the network for signal strength values from nearby access points, and record the samples in a database together with the BSSID (MAC address) which uniquely identifies the access point. Our system stores a maximum of three distinct signal strength values for each location for better accuracy. This implies that there must be at least three different wireless access points within the reach of the device, which can be used as basis for the measurements. If less than three access points are available, the signal vector will be reduced to one or two signal measurements. The device does not have to be physically connected to these networks, as long as it is able to scan the signals. The signal vector is then linked to an entity which describes the location.

Figure 6.1 shows an illustration of how the fingerprint-location relationship is represented in the database. The table in the background shows the fingerprint for the location with ID 52 (highlighted). This fingerprint consists of three signal strength indications (rssi column), which are measured from the access points with IDs 106, 107, 108, respectively. The WIFI_APS table contains records of all access points which are part of a fingerprint and their unique BSSID value. The map coordinates are stored together with the rest of the location information in the Locations table.

id	loc_id	ap_id	rssi
111	51	105	-64
112	52	106	-70
113	52	107	-77
114	52	108	-

id	name	description
50	Infohjernet	Informasjon
51	Auditorium EL1	Lite auditori
52	"Kattelaben" datasal	Maskinene krev...

id	bssid
101	00:1D:68:DD:15:F0
102	00:0C:F6:2F:F9:4B
103	00:14:6C:90:65:1C
104	00:19:5B:01:66:0F
105	00:1E:E5:98:CB:CA
106	00:21:1B:EB:77:E0
107	00:21:1B:EB:77:E1
108	00:21:1B:EB:77:E2

Figure 6.1: Data representation of the location fingerprints

6.3.3 The online phase

During the online phase of the location fingerprinting technique, a mobile device is positioned within the coverage area using the data in the location database. The following steps explain how the location of a user is determined:

1. Using the network interface on the device, a scan of “hearable” access points is performed.
2. The received signal strength indications (RSSI) are sorted, and the three strongest indications are included in the signal vector for the fingerprint, which are then sent to the application server as input to the *GetLocationByFingerprint()* web service method.
3. At the server, the observed fingerprint is compared against the collection of known fingerprints using a pattern-matching algorithm. The algorithm computes the Euclidean distance between the measured RSSI vector and each fingerprint in the database, and returns the fingerprint whose signal characteristic closest resembles the observed signals. The first step is to extract the fingerprints that consist of one or more access points with BSSID values equal to that of the observed ones, followed by a comparison of signal strength values.
4. If a match is found, the location associated with that particular fingerprint is returned to the client.

Figure 6.2 illustrates the online phase of the location fingerprinting technique. The code snippet is not an excerpt of the actual algorithm. It is just to illustrate where in the process the fingerprint pattern-matching takes place. The complete pattern-matching algorithm can be found in the source code for the *LocationService* class.

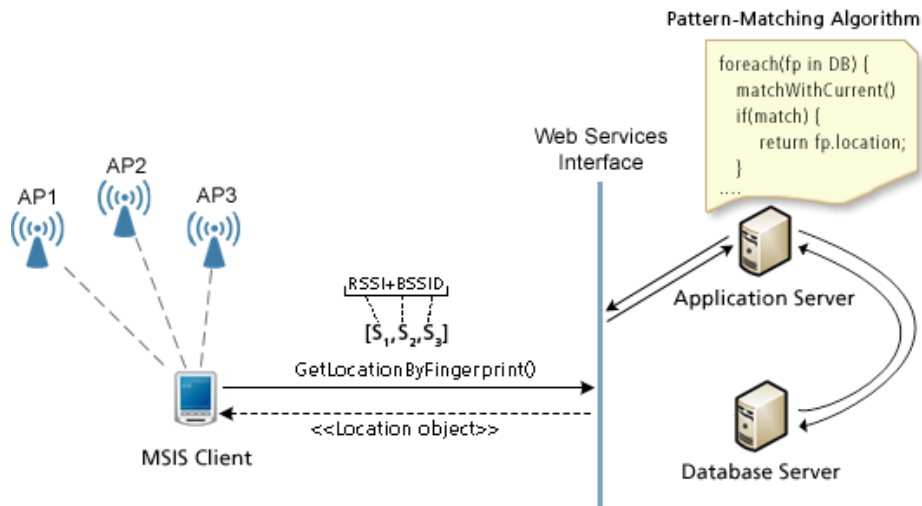


Figure 6.2: Location fingerprinting - online phase

6.3.4 Advantages

The Wi-Fi positioning module has several advantages:

- Works indoors: satellite-based navigation systems such as GPS require clear vision to the satellite, making them highly susceptible to signal interference and practically useless indoors.
- Implementation costs are low: it can be deployed within existing Wi-Fi infrastructure without a need for third-party components.
- The availability has high potential: the number of Wi-Fi compatible devices are increasing rapidly, and so are the availability of wireless hotspots. The Wi-Fi technology is also platform independent and part of the IEEE 802.11 standard.
- It offers quite good accuracy and high resolution, making it possible to position users with 8-10 meters precision (according to our tests).

6.3.5 Limitations

The technology also has a few constraints and limitations:

- The technique requires a fairly stable and well established Wi-Fi network. Since the technique rely on the previously recorded signal strength values and BSSID's to construct the fingerprint, a change in these parameters would affect the system. By basing the location fingerprint on more than one reference AP, the risk is reduced.
- The initial generation of the fingerprint database is time consuming, and must be done manually to some extent.
- Due to Wi-Fi networks mainly operating within buildings, the coverage area is limited outdoors. This, however, is not a limitation with the positioning technology itself, but rather a consequence of how Wi-Fi networks generally are deployed. By adding wireless access points outside, this problem is easily overcome.
- The reliability and accuracy of the system is proportional to the number of access points in the coverage area. More access points make it easier to separate between different locations.

6.4 Implementation example

The example below briefly explains how to connect to the web services, and how to apply them practically in an application. Only the code related to the web service methods and objects will be shown for the sake of brevity and conciseness. An actual application would also include code for the presentation layer to display the data to the user.

6.4.1 Log in

In our example, the service should only be available to authenticated users. To allow users to log in, we make use of the UserService web service. Listing 6.1 illustrates how the username and password is sent to the application server for authentication. Of course, the application would provide some means for the user to enter their login details. The `SHA1()` function generates a SHA1 hash value from the password string.

```
1 UserService service = new UserService();
2 UserProfile user = service.Login('myUsername',
    SHA1('myPassword'));
3
4 if(user.IsAuthenticated)
5 {
6     ShowSecureArea(); // Login successful, proceed to next screen
7 }
8 else
9 {
10    Message('Incorrect username or password'); // Login failed
11 }
```

Listing 6.1: User Authentication

6.4.2 Search for locations

Once the user is successfully logged in, we can provide an option to search for locations. First, a reference to the LocationService web service is obtained, then the `GetLocationByName()` method provides the necessary functionality. Listing 6.2 shows how this can be implemented.

```
1 LocationService service = new LocationService();
2
3 Locations[] foundLocations; // will store the matching locations
4 try
5 {
6     // Lookup locations matching search string
7     foundLocations = service.GetLocationByName('search string');
8 }
9 catch (Exception) // something bad happened, eg. network issue
10 {
11    Message('Unable to retrieve locations');
12 }
13
14 if(foundLocations not null)
15 {
16    foreach (Location in foundLocations)
17    {
18        // Display location information to user
19        Print(Location.Name, Location.Description, ...)
20    }
21 }
```

```
21 }
22 else
23 {
24     Message('No locations were found');
25 }
```

Listing 6.2: Search locations by name

To determine the current geographical position of the device, the application needs to obtain the raw signal strength data from the network interface APIs of the operating system. This procedure varies greatly between platforms, but in the following we will assume the information is available through a class `WiFiHelper`. Listing 6.3 gives an example of how the location service can be used to obtain the current location.

```
1  /* Retrieve the signal strength values of nearby access points */
2  Signal[] signalVector = WiFiHelper.GetSignalVector()
3
4  if(signalVector not empty)
5  {
6      currentLocation =
7          service.GetLocationByFingerprint(signalVector);
8  }
9  Message('You are here:' + currentLocation.Name);
10
11 /* We can also use this to offer location-aware search.
12  * First, provide some means for the user to select type of
13  * location/point of interest...
14  */
15 LocationTypes typeOfLocation = RequestInfoFromUser();
16 // ... Then we get nearby locations
17 Locations[] nearbyLocations = service.GetNearbyLocations(
18     typeOfLocation, signalVector, numberOfLocationsToReturn);
19
20 foreach (Location in nearbyLocations)
21 {
22     DisplayLocationToUser()
23     // Show distance to location
24     Print(Location.DistanceFromCurrent);
25 }
```

Listing 6.3: Determine geographical position

As the above example illustrates, very little code is needed to implement the basic services offered by MSIS. This clearly underlines the benefits of the Web Services technology and the SOA architecture. The majority of the processing is done on the external server, as part of code which is reusable between multiple client applications. The `ScheduleService` works in the same way, and the code will be very similar to the above. Basically, you provide the web service methods with some input data, and presents the results in a way tailored to the given client platform.

6.5 Database design - revised

A description of the MSIS database structural design is given in [56]. The main entities of the database has not changed since the first version, however, a few new tables have been added to accommodate the new features and improvements introduced in the final version. Figure 6.3 shows the old database ER diagram, and the revised database design is depicted in Figure 6.4. In the initial version of the system, the course and lecture information was extracted from the NTNU website. This proved to be a highly unreliable source to base our data on, as a slight change in the HTML markup on the page could interfere with our data extraction methods. Therefore, we decided to replace the data source with a database-based solution.

Three new tables have been added for this purpose: *Courses*, *Program_Codes*, and *Timetable*. The *Courses* table contains definitions about each course available, the different study program codes are stored in *Program_Codes*, and *Timetable* contains information about each lecture and exercise. The time table has one record for each appointment, and contains a reference to a course code, and zero or more study programs.

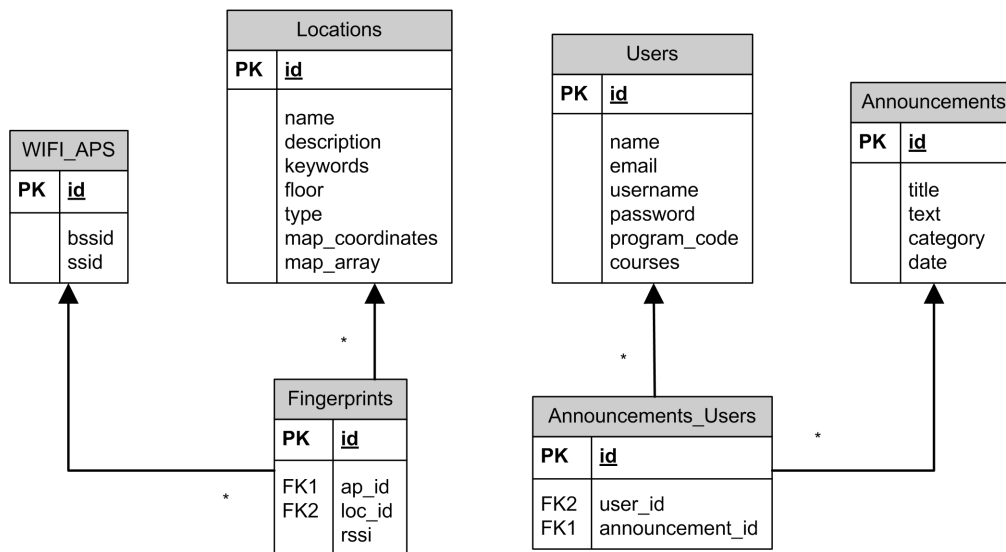


Figure 6.3: Old Database ER Diagram

In the future, we recommend that the service is to be integrated directly with the existing course information database at NTNU to avoid duplicate entries and stale data. This can easily be done by modifying the data access methods in the web service, and would simplify management considerably.

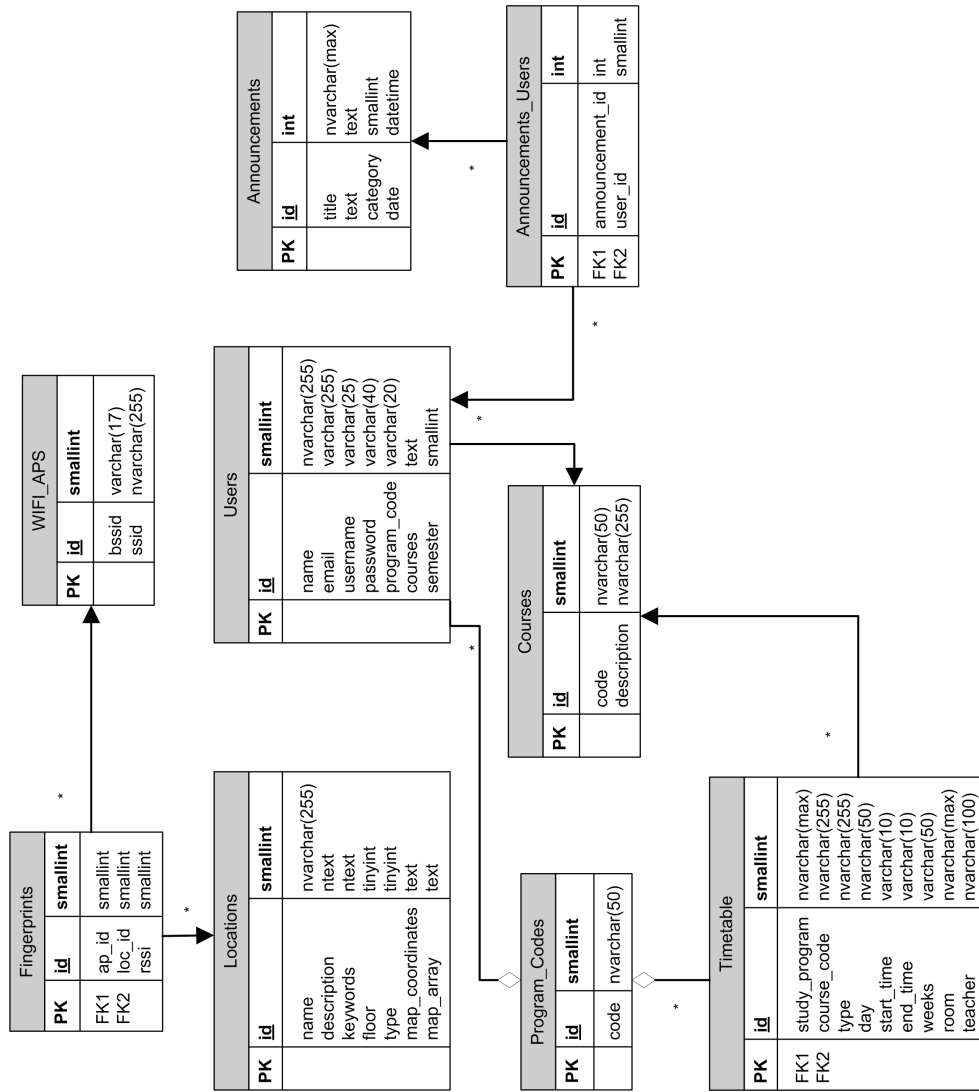


Figure 6.4: Revised Database ER Diagram

6.6 Third-party components

In addition to the Microsoft .NET Framework, we have made use of two other components which provides additional functionality on top of the existing classes.

6.6.1 OpenNETCF Smart Device Framework

The OpenNETCF Smart Device Framework (SDF) [63] is a library of classes and application programming interfaces (APIs) which facilitates interaction with low-level functions part of the Windows Mobile operating system. Classes provided by the OpenNETCF.Net package are utilized in our implementation of the positioning module. The classes provide methods for network layer access and manipulation of the wireless communication facilities of the device. The MSIS client application makes use of the OpenNETCF SDF Community Edition.

6.6.2 ASP.NET AJAX Control Toolkit

The AJAX Control Toolkit [4] is a free collection of ASP.NET controls and extenders which aids developers in creating rich Internet applications. Some of the controls offered by the toolkit have been used for the web-based administration interface and the desktop version of the location search application.

Chapter 7

Conclusions and further work

“It’s more fun to arrive at a conclusion than to justify it.”

–Malcolm Forbes

This chapter recapitulates the main findings of our research and concludes the report. Some suggestions for further work on the MSIS system are proposed, together with a discussion of possible limitations and challenges.

7.1 Contributions

This work is based on a preliminary study carried out autumn 2008, which examined the need for a mobile student information system at NTNU. A survey was conducted among students at the Department of Computer and Information Science, reviewing existing information services and communication channels with respect to applicability and user-friendliness. In light of these findings, suggestions for the design of a new information service were proposed.

During the course of this project, we have designed and developed a mobile service for students at NTNU, which makes user-centric information easily accessible through the use of mobile computing and wireless networking. An extensive evaluation of the initial prototype has been conducted, and the findings were analyzed in accordance with scientific models and methodologies. A group of potential users were selected to be part of a usability test and user acceptance survey. The main purpose of the usability test was to elicit feedback from a representative sample of users, to assess the utility and usability of the proposed service. The user acceptance survey is based on the Mobile Services Acceptance Model (MSAM); an instrument intended to explore various empirical metrics of mobile services, such as the perceived usefulness and likelihood of user adoption. A paper [33] presenting some of the key findings of the MSAM evaluation was submitted for journal publication in June, 2009.

This project has been carried out in accordance with the design-science research method. The implementation of the system has been carried out in multiple iterations, where each iteration contributes towards a final solution by applying knowledge acquired at the previous steps. The

outcome of the evaluation in Chapter 4 has been largely influential on the directions for the final iteration of implementation in our design cycle.

7.2 Main findings

The preliminary study revealed that there is a high demand for improved information systems among the students at NTNU. In particular, it seemed to be a need for a more consistent set of information channels. In an effort to improve the current situation, a service called the Mobile Student Information System (MSIS) has been developed. The system design consolidates the main findings from the preliminary survey and evaluation phase, and the following services have been implemented: a location search tool, which allows users to search for *points of interest* on campus (e.g. a lecture hall or meeting room). The location search is location-aware, which makes it possible to sort the search results in order of distance from the current position of the user. The second service incorporates a lecture schedule planner. A time table is generated dynamically based on the courses the student have signed up for and is downloaded to the mobile application. A Wi-Fi positioning technology based on the location fingerprinting technique has been implemented for use in MSIS. The positioning module makes it possible to determine the position of a handheld device within a wireless network.

The evaluation of the mobile service has confirmed that MSIS is both useful and easy to learn. The location search functionality works well, and is believed to be a valuable asset for new students in order to become acquainted with the campus area. Our findings indicate that there are no significant differences between technology savvy users and users without a technical background, when it comes to the perceived usefulness, perceived ease of use, and intention to use the application. The course schedule service was also appreciated by the students. According to our results, the majority of the participants agreed that a mobile version of the time tables are extremely useful, as many students today often have to create their own time tables on paper, which requires both time and effort.

During the analysis of the user acceptance survey, it became evident that the notion of trust has significant influence on the adoption of a mobile service like MSIS. According to our findings, students at NTNU are most concerned with the reliability of the data provided by the system. 76% percent of the respondents consider this an important factor for their intention to use the system. Issues of privacy were also regarded as important. Furthermore, we note that context is likely to affect the usage of the mobile service. The service is perceived to be very useful in situations where it is impossible to use a standard computer, for example when students are travelling between lectures, spending time in the cafeteria, etc. All in all, of the students participating in the survey, 92% said they would use the system now, or probably would have used it when they were junior students. Hence, we conclude that users are likely to adopt the system if it is made available.

The validity of the survey instrument has been tested using Internal Consistency Reliability analysis. Results show that all measurement constructs have a reliability coefficient above the minimum threshold value defined by Robinson et al. [68], thus the research model (MSAM) is considered well suitable for estimating user adoption of mobile services.

A new iteration was started right after the evaluation, with the purpose to address the issues that had been discovered. The suggested enhancements were implemented in the revised version

of MSIS in order to arrive at a final solution in accordance with our design-science approach. Finally, a new evaluation, although on a much smaller scale, was conducted to ascertain whether the improvements were successful in addressing the previously identified shortcomings. It was consensus among the participants that the final version includes a number of new features and issue resolutions that contribute to a better user experience, and ultimately, adoption of the service.

7.2.1 Research questions: retrospection

The overall goal of this project has been to answer the three research questions defined in Chapter 3. We asked: *Can user-centric mobile information services enhance the everyday campus life of students at NTNU?* We believe it can. The findings of our evaluation confirm that the students perceive MSIS as a very useful service, and the majority of the test persons believe they would use the system if it was available. However, some of the functionality offered by the system is expected to gain more support among junior students than senior students, such as the location search service. In order for the service to be successful, a robust and scalable infrastructure must be maintained. The servers should allow for a high number of simultaneous users, and a reliable wireless network must be in place. The modular service oriented architecture of MSIS facilitates flexibility and scalability.

With regard to the second research question, we did not find strong evidence that the students' attitudes towards new technology are of significant influence for the acceptance of the MSIS service. Given that the service is perceived as useful and easy to learn, the students does not let their previous experience with mobile services affect their intention to use the application. Computer science students did rate the usefulness and ease of use slightly higher than students not majoring in computer science. We note that the students in the survey consisted mainly of students majoring in engineering subjects; hence, they are probably somewhat more interested in technology than the average population. The results might have turned out differently if the sample comprised a more diverse group of people. Nonetheless, the target audience for the MSIS service is university students; hence, it makes no sense to test the application on, say, nurses and construction workers. Thus, we conclude that students' attitudes does not play an important role with regard to system adoption, but this does not imply that the same must be true for all other mobile applications or user groups.

In the last research question, we asked whether the utility of a mobile student service would increase by adding context-awareness to the application. The contextual search functionality confirms this hypothesis. The final evaluation acknowledged several advantages of a location-aware search tool: it provides more relevant search results to the user, and increases task efficiency considerably. Other aspects of context, such as time and personal preferences are also important factors in terms of this discussion. It adds knowledge, allowing the system to exert informed behavior adapting to the current situation of the user.

7.3 Further work

This section suggests directions for further work on the MSIS project, and reflects on some of the limitations with the present design.

Better integration

The MSIS service should be integrated with existing information systems in order to leverage the existing sources of information at NTNU, and avoid duplication of data. For example, the schedule service can be integrated with the global “StudWeb” database, which contains information about each student and their study program. Hence, instead of the user having to add the courses manually, the information could be extracted directly from the database. The same can be done for the location search service as well. There might already exist databases of information about the various rooms and buildings on campus, which can be used to populate the location database.

Set up a proper infrastructure

If the system is to be set in production, a dedicated server should be configured within the NTNU IT department in order to ensure consistent network speed and reliable operation.

Clients for other platforms

Currently, Windows Mobile devices are not in widespread use among students, and therefore, one should consider developing mobile clients for alternative software platforms, such as Java or Symbian. This would allow a greater portion of the students to take advantage of the system. The flexible service oriented architecture will simplify this task.

Expand location database

The database of location fingerprints must be expanded to cover the entire campus area. This is basically a data entry process, which involves mapping Wi-Fi fingerprints to physical locations according to the scheme described in section 6.3.

Evaluation on a larger scale

In order to examine how user behavior evolves over time, a new evaluation may be conducted as part of a longitudinal study of the MSIS service. The sample size may be increased to represent students from other departments and nationalities. This may reveal differences in usage pattern due to discrepancies between cultures and ethnicities.

Combine Wi-Fi positioning with other technologies

The existing Wi-Fi based positioning system could be combined with for example GPS positioning, to provide a more ubiquitous positioning service. Wi-Fi positioning could be used indoors, while a switch to GPS would occur if the user travelled outside the Wi-Fi coverage area. This could for instance make the contextual search work in larger areas, and may thereby to some extent obviate the limitations discussed in section 5.1.1.

7.3.1 Limitations

The location fingerprinting technology requires that the network topology and signal characteristics of the wireless network remain fairly persistent over a period of time. Otherwise, the fingerprints may become obsolete, and consequently, the location database must be updated frequently. During almost 10 months of testing we have not experienced any changes to the network or other issues, so we have no reasons to believe this would be a problem in the future, but it is something one needs to be aware of if the system is to be deployed on a large scale. Thus, network maintenance such as hardware replacements or other procedures should be planned carefully beforehand.

Some wireless access points are capable of dynamically adjusting their signal strength / power ratio according to the network topology, traffic patterns, and other variables. Normally, this would make a positioning system based on Wi-Fi signal characteristics highly unreliable. To circumvent this problem, our solution is to do fingerprint pattern matching according to unique BSSIDs before examining the signal characteristics. Hence, in most cases the system would be unaffected by changes to the signal strength indications as long as the physical address (BSSID) of the access point does not change (usually only happens if the access point itself is replaced). Only if the fingerprints of two locations comprise the same (potentially three) access points, the signal characteristics are used to distinguish one position from another. This is in fact quite unlikely to occur on the NTNU campus, because virtually every room or point of interest has its own access point. Thus, in most cases all three access points associated with a location have to be replaced in order to invalidate the fingerprint.

Appendix A

Usability scenarios (in Norwegian)

The usability test scenarios which was used for the evaluation of the system are included below.

SCENARIO 1 – Karttjeneste

Ola er førsteårsstudent ved NTNU Gløshaugen, og som så mange andre, ikke kjent med alle rom og bygninger på campus. Han har en avtale i laboratoriet i sokkeletasjen på IT-bygget.

Oppgave 1: Logg på applikasjonen med følgende brukernavn/passord: testuser / test

Oppgave 2: Start lokasjonstjenesten (“Find a Location”) i MSIS og søk etter “Datalab itbygget”. Klikk på lokasjonen i resultatet for å vise detaljer eller åpne karttjenesten. Marker posisjonen på kartene under (Figur A.1). Prøv å zoome inn og ut på kartet for å vise etasjeplanen.

Oppgave 3: Gå nå (fysisk) til auditorium F1 (bruk eventuelt kartsøket for å finne fram), og velg så menyvalget “Where am I?” på menyen nede til høyre inne i lokasjonssøkermenyen. Systemet vil returnere lokasjonen din sammen med en kode. Noter koden under.

Kode: _____

Oppgave 4: Returner til hovedmenyen.

Kommentarer (Hva fungerte, hva fungerte ikke? Responstid? Noe ulogisk/vanskelig?):



Figure A.1: Kart

SCENARIO 2 – Forelesningsplanlegger

Jorunn har litt ledig tid mellom to forelesninger og ønsker å sjekke hvordan forelesningsplanen ser ut for resten av uken. Hun har på forhånd oppdatert sin MSIS brukerprofil med de emnene hun tar dette semesteret, og slipper derfor å huske de individuelle fagkodene (Oppgave 1).

Oppgave 1: Logg på det webbaserte administrasjonsverktøyet på følgende adresse: <http://msis.myvnc.com/msis/> (bruk samme brukernavn/passord som i Scenario 1). Fyll inn felter for fagkoder og studieprogramkode for dine timeplanfestede emner dette semester. Har du ikke timeplanfestet undervisning, bruk følgende eksempeldata:

Fagkoder: TDT4175-1, TFY4125-1, TDT4215-3

Studieprogramkode: MTDT

Oppgave 2: Start forelesningsplanleggeren (“My Schedule”) i MSIS. Du vil få en oversikt over ukens forelesninger og øvinger i fagene du la inn i forrige oppgave. (Du kan velge å vise forelesninger eller øvinger via menyen nede til høyre.)

For ukedagene mandag-fredag, noter under fagkoder, tid og sted for forelesninger slik det fremgår av planen.

	Man.	Tir.	Ons.	Tor.	Fre.
Kode(r)					
Tid					
Sted					

Oppgave 3: Klikk på en av forelesningene i planen. Den geografiske posisjonen til auditoriet/rommet vil indikeres på kartet. Klikk “zoom inn” knappen på kartet for å se etasjeplan. Zoom ut for å returnere til oversiktskartet.

Kommentarer (Hva fungerte, hva fungerte ikke? Responstid? Noe ulogisk/vanskelig?):

Appendix B

User acceptance survey

Mobile Student Information System

User Acceptance Survey

Please use a few minutes to answer the following questions pertaining to the utility, perceived usefulness, usability and general conception of the MSIS service. All respondents remain anonymous.

Perceived Usefulness

	Strongly disagree				Strongly agree			
PU1. Using the system would increase the efficiency of my daily work.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU2. The system would allow me to find rooms and buildings at NTNU.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU3. The system would make it easier to keep track of my weekly tasks.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU4. The system would allow me to better schedule my time.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU5. The system would be useful for me as a student.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Perceived Ease of Use

	Strongly disagree				Strongly agree			
PEU1. Learning to operate the system would easy for me.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEU2. I would easily find the information I am looking for using the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEU3. I would find the user interface of the system clear and intuitive.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEU4. I would find the system to be flexible to interact with.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEU5. I would find the system to easy to use (user-friendly).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Trust

I could use the system...

	Not important				Very important			
TU1. if I have a clear conception of the functionality of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TU2. if the system provider is widely acknowledged (e.g. NTNU)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TU3. if the system protects the privacy of its users .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TU4. if I feel confident that I can keep the system under control.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TU5. if I feel confident that the data returned by the system is reliable.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TU6. If I believe it is risk-free to use the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TU7. if it is safe to use the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Personal Initiatives and Characteristics

	Strongly disagree				Strongly agree			
PIC1. I am capable of using the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIC2. I have fun using the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIC3. I prefer to be the first one using the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIC4. Using the system gives me an advantage over those who don't.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIC5. I would only use the system if it was available for free.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIC6. I find it rewarding to use the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIC7. Using the system is a good idea.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Appendix C

Deployment instructions

This chapter describes how to install the MSIS system, and the prerequisites that must be met in order to deploy the application in a production environment.

C.1 The backend servers

The MSIS application relies on an application server and database server to provide its services. The application server and database server may reside on the same computer, or a dedicated computer may be used for each server.

C.1.1 Database server

The MSIS database runs on a Microsoft SQL Server. We recommend the following setup:

- Operating System: Microsoft Windows Server 2003 or later
- Microsoft SQL Server 2005 (or Express Edition) or later.

The database is named “msis.mdf” and is included in Appendix D.2. To install it, just attach the database to an SQL instance on the database server. This can be done from Microsoft SQL Server Management Studio. If the application server and database server are deployed on separate computers, the database server should be configured to allow connections over TCP/IP. Exactly how this is specified varies between versions.

Figure C.1 shows a screenshot of the SQL Server Configuration Manager which is included with Microsoft SQL Server 2008. Make sure to note down the Account Name and Password for the database instance, as this will be needed later.

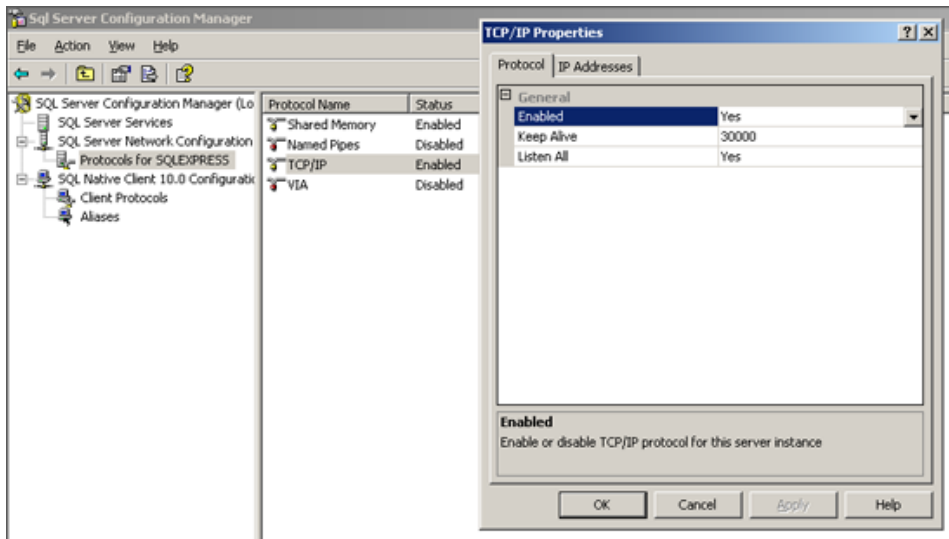


Figure C.1: Enable TCP/IP for the database server

C.1.2 Application server

The following software should be installed on the computer prior to deploying the application server:

- Recommended Operating System: Microsoft Windows Server 2003 or later
- Microsoft Internet Information Services (IIS) 6.0 or later
- Microsoft .NET Framework 3.5 [52] + Service Pack 1 [53]

Once the prerequisites have been met, the application can be deployed on the server. Appendix D.1 contains the files for the application server (Web Services and administration interface). Copy the contents of the **Application_Server** to a folder on the computer. The first step is to add a new application for the MSIS Web Services in IIS.

1. Create a new Virtual Directory and Web Site using the IIS Manager (see Figure C.2). The document root should be set to the folder on the computer where the Web Services are located.
2. Turn the Virtual Directory into an Application by clicking on the **Create** button in the Application Settings section of the Virtual Directory Properties pane. The application can be added to an existing Application Pool, or a new one may be created.
3. Make sure to enable ASP.NET 2.0+ for the application under the ASP.NET tab.

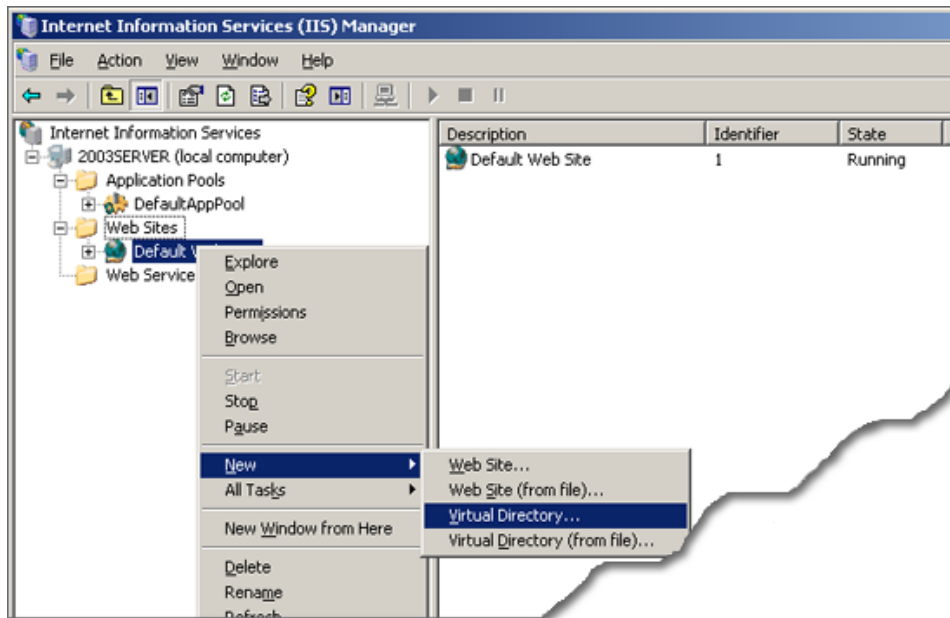


Figure C.2: Create new Virtual Directory

Now, the web services should be accessible through a browser using the hostname of the server. The next step is to configure the database connection parameters. The settings are stored in the *App_Code\SQLConn.cs* class as shown in Listing C.1. The **Data Source** parameter specifies the hostname/IP and port number of the database server (as configured in the previous section). **User ID** and **Password** should be changed to match the login credentials of the database user.

```
1 static string connectionString = 'Data Source=127.0.0.1,1433;'  
2                               + 'Initial Catalog=msis;'  
3                               + 'User ID=sa;Password=msis;'  
4                               + 'Connection Timeout=5';
```

Listing C.1: Database connection details

Note: By default, the administration tool is assumed to reside on the same computer as the application server. If it is moved to a different server, the URL reference to the web services must be changed to match the correct server address in the *web.config* file under the **<appSettings>** section.

The installation of the application server is now complete!

C.2 Client application

In order to use the MSIS application from a mobile device/PocketPC, the client application must first be installed on the device. The application is based on the Microsoft .NET Compact Framework 3.5 [51], and runs on Windows Mobile compatible devices. If the latest .NET CF common language runtime libraries are not pre-installed on the device, they must first be installed. There are two ways to install the .NET Compact Framework:

1. Connect the device to a computer which has ActiveSync [50] installed. Then run the .NET Compact Framework Installation Wizard¹.

OR

2. Run the CAB files on the device that correspond to the device-specific CAB files found in the install folder of the downloaded file.

Once the .NET CF is installed, the MSIS application can be deployed on the device. The application is distributed as a CAB file, and is included in Appendix D.3. A CAB file is the standard file format for distribution of Windows Mobile/PocketPC applications, and can be installed using the **ActiveSync Application Manager** on a desktop computer, or by copying the CAB file to the device and then invoking it from the device. A shortcut will be added to the Programs folder.

When starting the MSIS application for the first time, you will be prompted to provide the URL to the Web Service application server and the map URL. Type in the URLs as configured during the setup of the the application server. A configuration file (*AppConfig.xml*) will be created in the same directory as the main executable. If you later need to update the URLs, the settings can be changed by using the “Options” menu item in the main menu, or by manually editing the *AppConfig.xml* file (see Figure C.3).

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <SettingsFile>
3   <SettingsGroups>
4     <group name='Configuration' />
5   </SettingsGroups>
6   <Configuration>
7     <WebServiceUrl>
8       <Type>System.String</Type>
9       <Value>http://url-to-server/webservices/</Value>
10    </WebServiceUrl>
11    <MapUrl>
12      <Type>System.String</Type>
13      <Value>http://url-to-server/maps/</Value>
14    </MapUrl>
15  </Configuration>
16 </SettingsFile>
```

Figure C.3: Application configuration file

The installation is now complete!

¹Available from <http://www.microsoft.com/downloads/>

C.3 Development environment

The MSIS client application and the web services have been developed in Microsoft Visual Studio 2008 using the latest version of the .NET Framework (3.5). The client application is based on a Windows Mobile 6 Professional Smart Device project, while the web services are ASP.NET Web Service applications.

The Visual Studio solution files and application source code are included in Appendix D. In order to build the projects, the Microsoft .NET Framework 3.5 must first be installed. The client application contains references to two third-party libraries; the `OpenNETCF.AppSettings` and `OpenNETCF.Net` libraries, which are further described in section 6.6. These libraries are included with the solution.

The database is a standard `.mdf` file which can be attached to any Microsoft SQL Server instance.

Appendix D

Digital attachments

This report is accompanied with A ZIP archive containing source code, program binaries, and other deliverables for the project.

D.1 Source code

The source code for the system is included in two folders:

Client_Application: Contains the source code for the MSIS mobile client application.

Application_Server: Contains the source code for the ASP.NET web services, and the web-based applications (administration interface, etc).

D.2 Database

The file **msis.mdf** contains the MSIS SQL database populated with test data. A test user with username “testuser” and password “test” has been created.

D.3 MSIS CAB installer

The CAB installer should be used to deploy the client application on a mobile device. The file is named **msis.cab**.

D.4 Video demonstration

A video demonstration presenting the functionality of MSIS is included with this report. The demo has the file name **msis_demo.mpg**.

References

- [1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wireless Networks*, 3(5):421–433, October 1997.
- [2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *HUC '99: Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer-Verlag, 1999.
- [3] R. Agarwal and J. Prasad. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, 9(2):204–215, 1998.
- [4] AJAX Control Toolkit. ASP.NET AJAX Control Toolkit, 2009. URL <http://www.ajaxcontroltoolkit.com/>. Retrieved on: 2009-05-27.
- [5] I. Ajzen. From intentions to actions: A theory of planned behavior. In J. Kuhl and J. Beckmann, editors, *Action control: From cognition to behavior*, pages 11–39. Springer-Verlag, 1985.
- [6] I. Ajzen and M. Fishbein. *Understanding Attitudes and Predicting Social Behavior*. Prentice Hall, Englewood Cliffs, NJ, 1980.
- [7] D. F. Alwin and J. A. Krosnick. The reliability of survey attitude measurement: The influence of question and respondent attributes. *Sociological Methods & Research*, 20(1): 139–181, 1991.
- [8] S. H. Andresen, J. Krogstie, and T. Jelle. Lab and research activities at wireless trondheim. In *ISWCS '07: Proceedings of the 4th IEEE International Symposium on Wireless Communication Systems*, pages 385–389, 2007.
- [9] R. P. Bagozzi, F. D. Davis, and P. R. Warshaw. Development and test of a theory of technological learning and usage. *Human Relations*, 45(7):659–686, July 1992.
- [10] A. Bandura. *Social Foundations of Thought and Action: A Social Cognitive Theory*. Prentice Hall, Englewood Cliffs, NJ, 1986.
- [11] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 2nd edition, 2007.

-
- [12] K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2nd edition, 2004.
- [13] M. Bell. *Service-Oriented Modeling: Service Analysis, Design, and Architecture*, chapter Introduction to Service-Oriented Modeling, page 3. Wiley & Sons, 2008.
- [14] G. M. Breakwell, S. Hammond, J. A. Smith, and C. Fife-Schaw. *Research Methods in Psychology*. SAGE, 2006.
- [15] S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983.
- [16] D. R. Compeau and C. A. Higgins. Application of social cognitive theory to training for computer skills. *Information Systems Research*, 6(2):118–143, 1995a.
- [17] D. R. Compeau and C. A. Higgins. Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 19(2):189–211, 1995b.
- [18] D. R. Compeau, C. A. Higgins, and S. Huff. Social cognitive theory and individual reactions to computing technology: a longitudinal study. *MIS Quarterly*, 23(2):145–158, 1999.
- [19] L. L. Constantine and L. A. D. Lockwood. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley Professional, New York, NY, USA, 1999.
- [20] L. L. Constantine and L. A. D. Lockwood. Usage-centered software engineering: New models, methods, and metrics. In *SEEP '96: Proceedings of the 1996 International Conference on Software Engineering: Education and Practice*, pages 2–9, Los Alamitos, CA, USA, 1996. IEEE Computer Society.
- [21] R. B. Cooper and R. W. Zmud. Information technology implementation research: A technological diffusion approach. *Management Science*, 36(2):123–139, 1990.
- [22] M. R. Crum, G. Premkumar, and K. Ramamurthy. An assessment of motor carrier adoption, use, and satisfaction with edi. *Transportation Journal*, 35(4):44–57, 1996.
- [23] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.
- [24] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, 35(8):982–1003, 1989.
- [25] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology*, 22(14):1111–1132, 1992.
- [26] J. Dawes. Do data characteristics change according to the number of scale points used? An experiment using 5-point, 7-point and 10-point scales. *International Journal of Market Research*, 50(1):61–104, 2008.

- [27] J. Dollimore, T. Kindberg, and G. Coulouris. *Distributed Systems: Concepts and Design*. Addison-Wesley, 4th edition, 2005.
- [28] A. K. Ericsson and H. A. Simon. Verbal reports as data. *Psychological Review*, 87(3): 215–251, May 1980.
- [29] A. K. Ericsson and H. A. Simon. *Protocol Analysis: Verbal Reports as Data*. Bradford Books / MIT Press, Cambridge, MA, USA, 1984.
- [30] L. Faulkner. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments and Computers*, 35(3), 2003.
- [31] M. Fishbein and I. Ajzen. *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research*. Addison-Wesley, Reading, MA, 1975.
- [32] S. Gao, J. Krogstie, and P. A. Gransæther. Mobile services acceptance model. In *ICHIT '08: Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology*, pages 446–453, Washington, DC, USA, 2008. IEEE Computer Society.
- [33] S. Gao, S. P. Moe, and J. Krogstie. An empirical test of mobile services acceptance model. Submitted for publication, June 2009.
- [34] D. Gefen, D. W. Straub, and M.-C. Boudreau. Structural equation modeling and regression: Guidelines for research practice. *Communications of the AIS*, 4(7):1–77, 2000.
- [35] G. M. Gjul. Fumble. Master’s thesis, Norwegian University of Science and Technology, 2006.
- [36] J. D. Gould and C. Lewis. Designing for usability: Key principles and what designers think. *Communications of the ACM*, 28(3):300–311, 1985.
- [37] I. U. R. P. W. Group. Common industry specification for usability - requirements (cisur), June 2007. URL <http://zing.ncsl.nist.gov/iusr/documents/CISU-R-IR7432.pdf>. Retrieved on: 2009-04-27.
- [38] T. S. Hansen. A cultural guide for mobile devices. Master’s thesis, Norwegian University of Science and Technology, 2007.
- [39] B. E. Haugvik. CULBIS - context-user-location based information structure. Master’s thesis, Norwegian University of Science and Technology, 2006.
- [40] J. Henry and D. Gotterbarn. Coupling and cohesion in object-oriented design and coding. In *CSC '96: Proceedings of the 1996 ACM 24th Annual Conference on Computer Science*, page 149, New York, NY, USA, 1996. ACM.
- [41] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.
- [42] V. Hmelyoff. Vh pocketpc capture, 2006. URL <http://www.hmelyoff.com/index.php?section=18>. Retrieved on: 2009-05-29.

- [43] M. A. Ibrahim. Utvikling og evaluering av ulike mobile, lokasjonsbaserte tjenester. Master's thesis, Norwegian University of Science and Technology, 2008.
- [44] A. Kaikkonen, T. Kallio, A. Kekäläinen, A. Kankainen, and M. Cankar. Usability testing of mobile applications: A comparison between laboratory and field testing. *Journal of Usability Studies*, 1(1):4–16, 2005.
- [45] J. Kjeldskov and C. Graham. A review of mobile hci research methods. In *Proceedings of the 5th International Mobile HCI 2003 Conference*, pages 317–335, Udine, Italy, 2003. Springer-Verlag.
- [46] H. K. Klein and M. D. Myers. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23(1):67–93, March 1999.
- [47] K. W. Kolodziej and J. Hjelm. *Local Positioning Systems: LBS applications and services*. CRC Press, 2006.
- [48] M. Kuniavsky. *Observing the User Experience: A Practitioner's Guide to User Research*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.
- [49] C. Larman. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, 2003.
- [50] Microsoft. ActiveSync 4.5, 2007. URL <http://www.microsoft.com/windowsmobile/en-us/help/synchronize/activesync-download.mspix>. Retrieved on: 2009-06-02.
- [51] Microsoft. .NET Compact Framework 3.5 Redistributable, 2008. URL <http://www.microsoft.com/downloads/details.aspx?FamilyID=e3821449-3c6b-42f1-9fd9-0041345b3385>. Retrieved on: 2009-06-02.
- [52] Microsoft. .NET Framework 3.5, 2007. URL <http://www.microsoft.com/downloads/details.aspx?FamilyId=333325FD-AE52-4E35-B531-508D977D32A6>. Retrieved on: 2009-06-02.
- [53] Microsoft. .NET Framework 3.5 Service Pack 1, 2008. URL <http://www.microsoft.com/downloads/details.aspx?familyid=AB99342F-5D1A-413D-8319-81DA479AB0D7>. Retrieved on: 2009-06-02.
- [54] K. Miller. *Communications theories: perspectives, processes, and contexts*. McGraw-Hill, New York, 2005.
- [55] Mobile Turismo. Mobile Turismo, 2004. URL <http://www.mobileturismo.com>. Retrieved on: 2008-10-20.
- [56] S. P. Moe. Mobile student information system (MSIS). Norwegian University of Science and Technology, 2008. Depth Study.
- [57] G. C. Moore and I. Benbasat. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, 2(3): 192–222, 1991.

- [58] myCampus. myCampus, 2005. URL <http://www.cs.cmu.edu/~sadeh/mycampus.htm>. Retrieved on: 2009-04-21.
- [59] E. Newcomer and G. Lomow. *Understanding SOA with Web Services*. Addison-Wesley Professional, 2004.
- [60] J. Nielsen. Iterative user-interface design. *IEEE Computer*, 26(11):32–41, 1993.
- [61] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, USA, 1994.
- [62] J. R. Olson and G. M. Olson. The growth of cognitive modeling in human-computer interaction since goms. In R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, editors, *Human-Computer Interaction: Toward The Year 2000*, pages 603–625. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.
- [63] OpenNETCF. OpenNETCF Smart Device Framework, 2009. URL <http://www.opennetcf.com/>. Retrieved on: 2009-05-27.
- [64] M. Page-Jones. *Practical Guide to Structured Systems Design*. Prentice Hall PTR, 2nd edition, 1988.
- [65] J. Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Professional, 2000.
- [66] E. E. Rigdon. A necessary and sufficient identification rule for structural models estimated in practice. *Multivariate Behavioral Research*, 30(3), 1995.
- [67] D. Robey. Research commentary: Diversity in information systems research: Threat, promise, and responsibility. *Information Systems Research*, 7(4):400–408, December 1996.
- [68] J. P. Robinson, P. R. Shaver, and L. S. Wrightsman. Criteria for scale selection and evaluation. In *Measures of Personality and Social Psychological Attitudes*, pages 1–16. Academic Press, San Diego, CA, USA, 1991.
- [69] L. J. Robinson, G. W. Marshall, and M. B. Stamps. Sales force use of technology: Antecedents to technology acceptance. *Journal of Business Research*, 58(12):1623–1631, 2005.
- [70] E. M. Rogers. *Diffusion of Innovations*. Free Press, New York, 4th edition, 1995.
- [71] B. N. Schilit, N. I. Adams, and R. Want. Context-aware computing applications. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, December 1994.
- [72] K. Schwaber. *Agile Project Management With Scrum*. Microsoft Press, Redmond, WA, USA, 2004.
- [73] B. Shneiderman. *Software Psychology: Human Factors in Computer and Information Systems*. Winthrop Publishers, 1980.

- [74] J. Shore and S. Warden. *The Art of Agile Development*. O'Reilly Media, Inc., 2007.
- [75] K. Siau and Z. Shen. Building customer trust in mobile commerce. *Commun. ACM*, 46(4): 91–94, 2003.
- [76] H. Takeda, P. Veerkamp, T. Tomiyama, and H. Yoshikawa. Modeling design processes. *AI Mag.*, 11(4):37–48, 1990.
- [77] S. Taylor and P. A. Todd. Understanding information technology usage: A test of competing models. *Information Systems Research*, 6(4):144–176, 1995.
- [78] R. L. Thompson, C. A. Higgins, and J. M. Howell. Personal computing: Toward a conceptual model of utilization. *MIS Quarterly*, 15(1):125–143, 1991.
- [79] V. Vaishnavi and B. Kuechler. Design research in information systems, 2007. URL <http://www.isworld.org/Researchdesign/drisISworld.htm>. Retrieved on: 2008-10-21.
- [80] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3):425–478, 2003.
- [81] R. Weber. Editor's comments: Still desperately seeking the it artifact. *MIS Quarterly*, 27(2):iii–xi, June 2003.
- [82] K. E. Weick. The management of organizational change among loosely coupled elements. In P. Goodman, editor, *Change in Organisation*, pages 375–408. Jossey-Bass, San Francisco, CA, USA, 1982.
- [83] K. E. Weick. Educational organization as loosely coupled systems. *Educational organization as loosely coupled systems*, 21(1):1–19, 1976.
- [84] C. Wharton, J. Rieman, C. Lewis, and P. Polson. The cognitive walkthrough method: A practitioner's guide. In J. Nielsen and R. L. Mack, editors, *Usability Inspection Methods*, pages 105–140. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [85] F. M. Wolf. *Meta-Analysis: Quantitative Methods for Research Synthesis*. Sage Publications, Inc, Beverly Hills, CA, USA, 1986.