NTNU
Norwegian University of
Science and Technology

Design and Evaluation of a Personalized Mobile Tourist System

Per Christian Røine

Master of Science in Computer Science
Submission date: June 2011
Supervisor: John Krogstie, IDI

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

Problem Description

Mobile applications supporting tourists with travel information can make use of information about the user's location, time and personal preferences to provide personalized recommendations. This could address the problem of displaying information and navigating on small mobile devices, as it allow tourists to receive information that fit very well with their current situation and needs.

Over the last year a prototype of a mobile tourist system using recommender technology has been developed. This project shall develop this prototype further based on

1. Existing identified needs for functionality
2. Evaluation of user acceptance, utilizing MSAM - Mobile System Acceptance Model

The research should be framed in a design science methodology, and the report should be written in English

Assignment given: January 14, 2011
Supervisor: John Krogstie, IDI

Abstract

Smartphones are rapidly evolving, making them powerful devices with many features. Location awareness is one of the hot topics, aiding applications to provide better services to users. A challenge is to combine the large amount of tourist information with the limited display sizes of smartphones. Also, tourists spend a lot of time finding information with little knowledge of their probable enjoyment of these tourist relevant locations. Recommender systems attempt to solve this by using information about users and points of interest. We will investigate several studies that discuss tourist applications.

This project presents the Mobile Tourist Service Recommender which is a personalized tourist application introduced by [Wium, 2010]. We have further developed this system, performed thorough usability testing, applied the Mobile Services Acceptance Model, and analyzed the results.

The results indicate that the system has potential and is encouraged by the positive feedback from many users. Users especially found the system to be beneficial to them as tourists, and that they could use the system during vacations. Unfortunately, the achieved responses were not completely satisfactory to our goals, but further iterations with the suggested improvements implemented will raise the user experience.

Preface

This report is the documentation of the project work as part of the Master's thesis in Computer Science spring 2011 by Per Christian Røine. This project marks the end of the 5-year long Master of Science education in Computer Science at the Norwegian University of Science and Technology (NTNU), and constitutes 30 units.

This project is an extension of the personalized mobile tourist application proposed by Magnar Wium, and is defined to further develop and evaluate this application. The fact that the system addresses actual problems that people encounter during their vacations has been motivational. The task at hand proved to be challenging, interesting and educational. In addition, through this project I have gained knowledge, experience and friendships, which I am grateful for.

I would like to thank my supervisor, John Krogstie, for his invaluable feedback and guidance during this project. In addition, I extend my gratitude to Shang Gao and Muhammad Asif for their assistance, and to all participants that volunteered to this research study. Last but not least I would like to thank my family for their continuous support and encouragement throughout my years at the university.

Trondheim, June 17, 2011

Per Christian Røine

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction and Overview

1.1 Motivation

Mobile technology is a field in rapid development. Especially over the past few years, the improvements in mobile technology have been tremendous. The increase in performance of these devices, as well as the additional hardware in form of GPS, sensors, etc., make them powerful devices which can develop comprehensive services to aid people in various scenarios. However, the small screens, wireless network and battery present challenges that needs to be addressed carefully.

As one of the biggest sectors in the world[Kabassi, 2010], the tourism industry is a very attractive sector for mobile technology. Tourists spend considerable time planning their activities, both before and during their visits. The plans made do not take personal preferences into account other than that the tourist chooses attractions based on what he/she thinks are good places to visit. In some cases, reviews from other tourists are available but the user has no knowledge whether this other tourist shares the same passions and/or has the same preferences.

Most mobiles today have features that are helpful in tourist-situations. GPS (or other means to locate the user's position), accelerometer and maps to mention a few, provides useful information that can be used to aid users. Further, these features combined with a recommender system can be of great assistance to users.

In spring 2010, [Wium, 2010] developed a personalized mobile application, named the Mobile Tourist Service Recommender, which is designed for smartphones using Android OS. The system is intended to help tourists find points of interest (POIs) such as hotels and restaurants in order to let them schedule their time more efficiently and increase the probability that they will visit places that they'll actually enjoy. The latter is powered by a recommender system that bases its recommendations on ratings by the user himself/herself, as well as other users' ratings. This project aims to further develop this system and perform thorough usability tests.

Cheverst et al. [2000] presents GUIDE which is a tourist service spanning the city of Lancaster using a client-server model. GUIDE employed a map-based interface and provided information on tourist attractions filtered by context information. Location and previously visited places are among the context information used to provide points of interest. The user evaluation exhibited a high level of acceptance, but also presented some challenges regarding its customization strategy in relation to context information.

The Mobile Tourist Service Recommender application, MTSR, is a personalized mobile tourist application which is developed for smartphones with Android OS. MTSR also uses a map-based interface and provides search-functionality for POIs, recommendations, complete travel plans, and more. Context awareness is central in MTSR's design, and user's preferences are used to provide good recommendations. In addition of searching for recommended POIs, the system will send suggestions of POIs that the user might enjoy. These suggestions are known as pro-active recommendations. MTSR also provides general information about places, as well as ratings from other users regarding the POI in question.

1.2 Project Scope

A fully functional prototype was developed last year based on research in relevant areas. The work described in this project builds further upon this foundation which was developed by [Wium, 2010]. The aim of this project is composed of the two following main tasks: 1) further development in order to improve the current version of the system, and 2) evaluate the prototype using a set of usability metrics.

1.3 Project Description and Context

The purpose of this project is to further develop the Mobile Tourist Service Recommender and examine whether the ideas behind it serve the overall interest of tourists and their user experience. This will be examined by performing thorough usability tests with a group of users. In order to assess perceived usefulness, intention to use and more, we will be employing the Mobile Services Acceptance Model.

1.4 Problem Definition

The purpose of this project is to further develop and assess the Mobile Tourist Service Recommender. We will implement more functionality and improve parts of the existing system if needed. The evaluation includes a large scale usability test where the Mobile Services Acceptance Model is applied to explore the users' attitudes toward the system. As a final step, the results will be analyzed and further work will be identified.

1.5 Report Outline

The rest of this report starts with presenting background information in Chapter 2. We will then state our research questions and method in Chapter 3, before we in Chapter 4 discuss usability and user acceptance testing. In Chapter 5 we describe the services of MTSR, and in Chapter 6 we elaborate on its architecture. Chapter 7 contains information regarding the scenarios and user acceptance survey used, as well presenting and discussing test results and the validity of these. In the final chapter, Chapter 8, we will discuss the results, present our conclusions, and state our contributions and suggestions for further work.

Chapter 2

Background

2.1 Location Based Services

[Spiekermann and Berlin, 2004] defines the term location based services (LBS) as "services that integrate a mobile device's location or position with other information so as to provide added value to a user". Location based services has been in use for several decades, as far back as the 1970s when the U.S. Department of Defense started operating the global positioning system (GPS), a satellite infrastructure serving the positioning of people and objects.

Location based services can be classified based on whether they are person-oriented or device-oriented [Schiller and Voisard, 2004]:

- **Person-oriented LBS:** Applications where the service is user-based, meaning that the position of the user is used to enhance the application. Typically, the user can control the service (e.g., friend finder application).
- **Device-oriented LBS:** Applications whose focus may be the position of a person, but it may also be the position of an object (e.g., car) or a group of people (e.g., convoy). These applications are said to be external to the user. Unlike person-oriented LBSs, the user is usually not controlling the service (e.g., car tracking for theft recovery).

Additionally, two application designs can further distinguish services[Schiller and Voisard, 2004]:

- **Push services:** Services that sends information to the user based on his/her whereabouts without having the user request it. An example of this could be if the person is shopping in an unknown city and the application is a shopping-guide that knows what his/her favorite shops are. The application sees that the user is nearby his/her favorite shop, and sends this information.
- **Pull services:** Services where the user has to interact with the application in order to receive information. This information may be location-enhanced. A typical use is to locate nearby services (i.e., services like ATMs).

2.2 Context Awareness

Context awareness has become an attractive field that is used to enhance user experience of applications. The number of articles published regarding this matter increased by approximately seven times from 2000 to 2007[Hong et al., 2009].

[Dey, 2001] defined context as follows:

> Context is any information that can be used to characterise 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.

Context is location, time and preferences among other things, and is very important as it provides information about the current status of people, places, things and devices in the environment[Korpipaa et al., 2003][Kwon, 2004]. Context awareness simply means that one is able to use context information. According to [Byun, 2004], a system is context aware if it can extract, interpret and use context information and adapt its functionality to the current context of use. Applications and services should be context aware and automatically adapt to changes in context, in order to provide the user with adequate service [Bolchini et al., 2007][Dey, 2001][Zhu et al., 2005].

2.3 Recommender Systems

In this section we will explore recommender systems. We will however only briefly discuss this matter as it is not one of the focuses of this study.

Recommender systems are simply information filtering systems that attempts to provide recommendations of items. Examples of items in this context are movies, books and restaurants. It analyzes available information, such as age and occupation of the user, to present items that the system believes are the best fits for the user.

These systems exist in many variants, but we can distinguish them into types based on their filtering technique. The two most common are content-based filtering and collaborative filtering. We will now take a closer look at these before we discuss the cold start problem and address an importance aspect of such systems.

2.3.1 Content-Based Filtering

Objects of interest are denoted by their associated features which can be the type of cuisine served and price level if the object is a restaurant. The recommended items are items that have similar content to items the user has liked in the past, or matched to predefined attributes of the user. This sort of filtering is depended on that much descriptive data is available about items to be recommended, and recommendations can only be based on the explicit information available. This scheme makes it difficult to provide the user with refreshing new items[Burke, 2002].

2.3.2 Collaborative Filtering

Collaborative filtering is often referred to as social filtering. This is because the technique it utilizes recognizes similarities between users based on their ratings and creates new recommendations based on this inter-user comparison[Burke, 2002]. It is also the most common filtering technique because of its power and simplicity[Joseph et al., 2000]. As opposed to content based filtering, this technique does not have problems providing the user with refreshing new items. A weakness of this system is the dependence on having many active users in the system to ensure a proper distribution of ratings across all items in the system. This presents a problem when new items or users are added to the system as these do not have any ratings affiliated with them. This is known as the cold start problem which we will describe in the next section.

2.3.3 The Cold-Start Problem

This problem relates to users or items that are new, and the system therefore has no ratings to base recommendations on. This is a common problem in collaborative filtering. As tourists travelling to a city most likely will be new users, makes them part of the cold-start problem. This makes it hard to provide recommendations to tourists as they only stay in the system for a short period of time and we have no information about them prior to their visit. [Lillegraven and Wolden, 2010] aims to minimize the impact of this problem by using Bayesian reasoning. This system has been implemented to MTSR, but is not utilized at his moment. We will introduce this system in later chapters.

2.3.4 Level of Required User Information

It is important to notice that different recommender systems require different amount of user information. The system may request demographical data such as age and nationality, ask the user to rater some items, or use external user history among other things. It is important that the system do not require too much information in order to make the experience better for the user. After studying several approaches, [Lillegraven and Wolden, 2010] decided to use demographical data in terms of age, gender, occupation, type of holiday, nationality and budget.

2.4 Previous Studies

In this chapter we investigate how other systems similar to MTSR have been developed. We will first look at three systems that [Wium, 2010] addressed in his thesis to create a foundation of existing solutions, before we investigate a more recent system.

2.4.1 GUIDE

GUIDE was developed by [Cheverst et al., 2000] for visitors of the city Lancaster. It is a prototype system that provides personalized tourist services. By involving both experts on tourism and tourists themselves, they discovered some key requirements. These requirements were flexibility, context sensitive information and support for dynamic information. GUIDE used both a browser interface and a map based interface to present recommendations on tourist attractions such as museums and historical buildings. The latter displayed a map with the position of the user and the recommendations.

The system used cell based WLAN on the PDA client to detect the location of the user, and to communicate with the server. GUIDE employs the widely used client-server model, and stores all information about tourist attractions on the server. To provide recommendations, they use content based filtering that is also executed on the server before it is sent to the user.

According to [Schwinger et al., 2005] and [Kabassi, 2010], personal context includes the user's current location, previously visited POIs, and personal preferences in terms of individual user models of features paired with probability of liking. Information about tourist attractions is considered as environmental context. This information includes features and links between nearby attractions and their opening hours. GUIDE distinguishes between these two types of context when filtering information.

In association with recommendations, users are prompted for their preferences. In addition to use this information, GUIDE learns from the user's interaction with the system. When tested on a wide range of users, the GUIDE system received a high level of acceptance.

2.4.2 CRUMPET

CRUMPET is a prototype system introduced by [Schmidt-Belz et al., 2003] and aims to serve as a user-friendly mobile service for tourists. It uses an interactive user interface which includes a map. Through this interface, users are able to request recommendations about restaurants, tourist attractions and tours. Updated information from external providers are also available, these are tailored to the user's current situation. The innovative feature introduced by CRUMPET, is the personalized pro-active recommendations.

The architecture is in three layers, namely client tier, middle tier and data access tier. At the bottom, the data access tier integrates several external sources of tourist information, and provides an accessible interface to the middle tier for this information. The middle tier adapts this information and applies geo-coding. The adaption process involves content-based filtering using individual models that are created and maintained by the user specifying preferences and the system performing adjustments by analyzing browsed data. The final layer, the client tier, automatically contacts the middle tier and provides the system with the user's movement by acquiring GPS-information.

User evaluation of this system shows that users are pleased with the application and say that they would consider paying for it. It also shows that the map interface proved to be a useful and intuitive abstraction. However, they ask for a better search function as they were unable to search for anything else than what the predefined recommendation requests stated.

2.4.3 TIP

The goal of the research project TIP [Hinze and Buchanan, 2006], Tourist Information Provider, is to develop a mobile infrastructure for cooperating information services. It uses the user's context as basis (i.e., location, personal preferences and travel history) to deliver information to the user.

The researchers stated that cooperating services is a key requirement for systems providing context sensitive information. Such services allow users to access and display information about sights in different forms depending on the requirements of the user and the limitations of the client device [Hevner et al., 2004]. Displaying information through a map display or as a list of recommendations are two such forms.

TIP used a third party library for visualization of sights in the map, but this library only supported basic rendering, leaving TIP with no ways to design distinctions between categories of sights. Also, the researchers focus was invariable tied to the architectural aspects of personalized mobile tourism applications. It is therefore difficult to assess the user experience and quality of this system.

2.4.4 Google Places

Google is present in many areas including Internet search, cloud computing, advertising technologies and operating systems (OSs). They developed the mobile OS Android which has become very popular, and since Android 1.6 the OS has been shipped with an application called Places[Google, 2011b]. Through this application, users may search for restaurants, coffee-places, and attractions among other things. We were only able to find general data (i.e., information about features and not detailed technical information), but its services clearly present themselves as tourist services.

Places utilize some of the available Google services such as Google Maps, making it easy to get directions, street view and more to all POIs. It uses two user interfaces for displaying results: one where the recommendations are presented through a list, and one displaying the points of

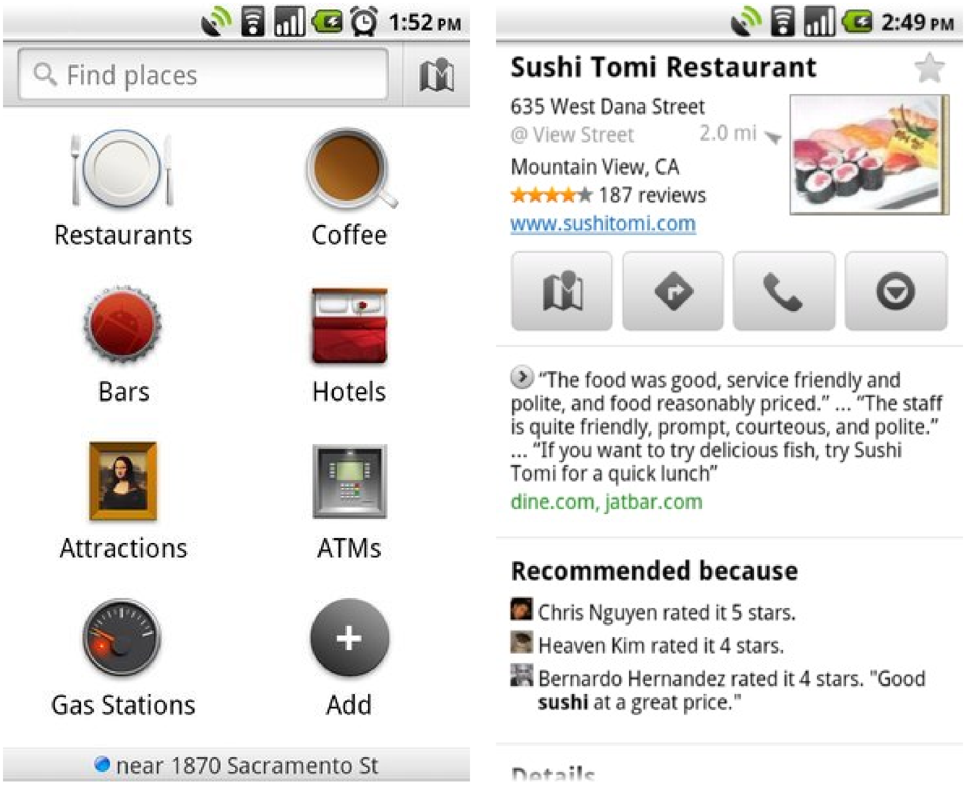


Figure 2.1: The search dialog and detailed information displayed in Places, respectively

interest in a map. To start a search, they present a very basic setup where eight categories (e.g., restaurants and attractions) are predefined and a search field (see Figure 2.1), allowing the user to quickly search for one of the predefined categories or conduct a custom search.

It provides detailed information about POIs such as details, ratings, explanations of why it is recommended to the user, and reviews, as shown in Figure 2.1. As Google is developing both the OS and the application, Android devices have a specialized widget (i.e., the "Rate Places"-widget) that allows users to perform 1-click rating. The application is available for Android, iOS and as a web application.

Part II

Design Science

Chapter 3

Research Approach

This chapter provides a description of the research approach used in this project. At first, the research questions are explored, and then a presentation of the employed research method is described.

3.1 Research Questions

The following is the research questions that we seek to find answers to in this project.

1. Can the prototype of the personalized mobile tourist application, MTSR, enhance the experience of tourists and others when searching for a place to eat, sleep or experience something?
2. How user friendly is the current design/user interface?
3. What are people's attitudes toward adapting this system?

3.2 Research Method

The design science paradigm is fundamentally a problem-solving paradigm which involves analysis, design, implementation and evaluation. This process will give us knowledge about problem domains and solutions.

[Takeda et al., 1990] suggests a general methodology that is widely used. Figure 3.1 illustrates this process which was developed after they analyzed the reasoning that occurs in the course of a normal design cycle. As depicted in this figure, we see that it all starts with *Awareness of Problem*. The next step is *Suggestions* where suggestions for the solution can be deduced from existing information and knowledge of the problem area. In the *Development*-step, an artifact gets implemented according to the results from the *Suggestions*-step. This implementation is evaluated in the next step, *Evaluation*, and is either approved in which case it will get to the final step, namely *Conclusion*, which indicates the termination of a specific design project[Vaishnavi, 2011]. However, the steps *Suggestions*, *Development* and *Evaluation* are usually performed several times during this process. The *Circumscription*-arrow indicates this path.

Research in design science focuses on the development and performance of artifacts aiming to improve the functional performance where artifacts include, but are not limited to, human-computer interfaces (HCIs) and algorithms. Regarding such research, Hevner et al. [2004]

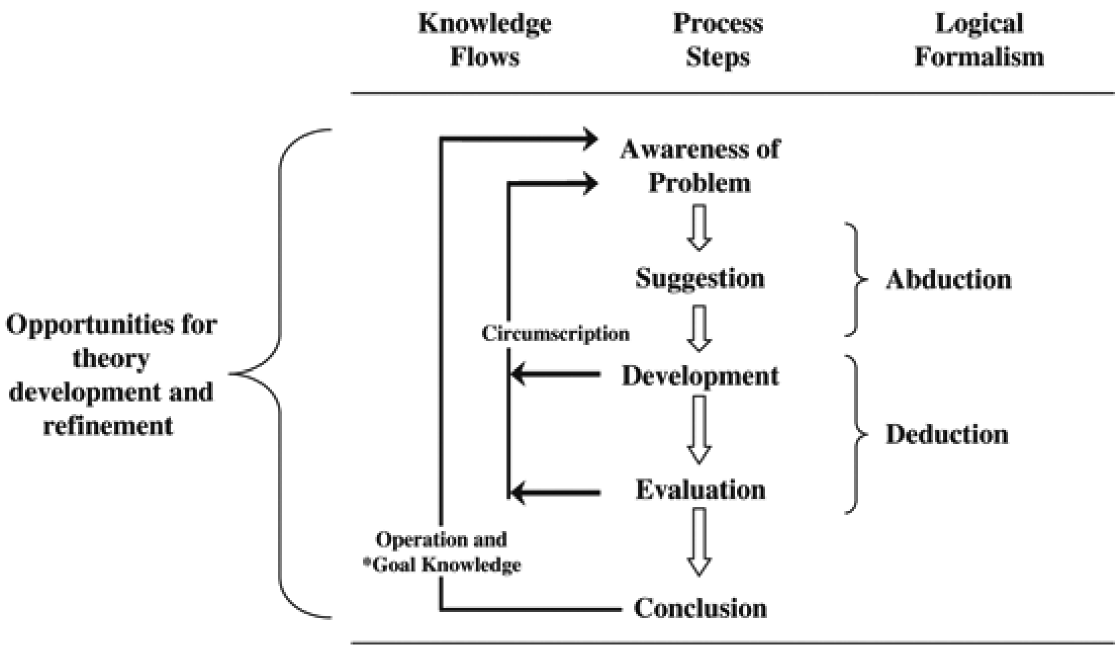


Figure 3.1: Design cycle proposed by [Takeda et al., 1990]

presents a set of guidelines. As this project is an extension of the Mobile Tourist Service Recommender, some of the guidelines will be linked to Wium [2010].

Guideline 1: Design as an Artifact

The definition of result in design science research is a purposeful artifact in form of a construct, model, method or instantiation. The application itself is in this case the artifact and is mainly described in Wium [2010], but new parts are presented in later chapters.

Guideline 2: Problem Relevance

This guideline requires that a technology-based solution is developed to important and relevant business problems. We introduced the tourism industry in Chapter 1 which is a major market and the need of information systems in this sector is clearly a business problem.

Guideline 3: Design Evaluation

Properties such as utility, quality and efficacy of the artifact need to be thoroughly demonstrated with well-executed evaluation methods. In this study we have performed usability testing, applied acceptance models and analyzed the results.

Guideline 4: Research Contributions

Clear and verifiable contributions made by the research must be provided. The contributions made in this project will be stated in Section 8.4

Guideline 5: Research Rigor

Guideline 5 states that design science relies on the use of rigorous methods in both the construction and evaluation of the design artifact. Methods used in this study include thorough usability testing, applying technology acceptance model for mobile services, and analysis of results.

Guideline 6: Design as a Search Process

During the iterative process of the research, it is important to continuously search for an effective solution to the problem. As Wium [2010] states, "the goal ... is not to solve the problem of information support in the tourist domain, but rather to gain a better understanding of how the optimal solution might look like. As we are using theory as a heuristic, the result of our research process will be presented as design suggestions that might guide future iteration" . These design suggestions can be found in Wium [2010], and newly acquired suggestions in later chapters.

Guideline 7: Communication of Research

Proper communication to both technology-oriented and management-oriented audiences is important and must be presented effectively. Through the chapters of this report, the performed work of this study is communicated.

Chapter 4

Usability and User Acceptance Testing

In order to find out how users will react to the system, we performed a usability evaluation. There are several elements that may be including in such an evaluation. The following is a list of the common elements defined by [Ivory and Hearst, 2001].

- **Capture:** Collect usability data such as time to complete tasks, errors and guideline violations
- **Analysis:** Analyze the data captured and identify usability problems
- **Critique:** Suggest solutions or improvements to mitigate problems

There exists a wide variety of models to perform usability evaluation. However, most of them do not take attributes and issues of mobile devices into account, thus making them less suitable for usability testing on such devices. Use of mobile devices has increased very fast, and people rely on them every day. These devices have also a lot of services available to them through Apple Store and Android Market among others. [Hussain and Ferneley, 2008] states that this increasing number of mobile devices significantly implies how important it is to assure that the application is useable through usability evaluations. Clearly, there is a need for a model that addresses mobile devices' attributes and issues. We will look at a model, aiming to address this need, later in this chapter.

The "think-aloud protocol" based on the work of [Ericsson and Simon, 1980][Ericsson and Simon, 1984], is a commonly used technique when performing usability evaluations. Using this technique, users will perform a set of specified tasks and are asked to elaborate on their thoughts and experiences as they solve these tasks. Useful information is given directly and it is therefore important to capture this information. This is usually done by the observer by taking notes and/or by video-/audio-recording of the test sessions. These results are usually quite reliable and close to users' experience in real life environments. However, users may be disturbed from the unnatural environment used in test-phases, and the feeling of being observed might stress some of the participants. This process is helpful in finding problems with the application, as well as ideas to improvements. In addition, it is considered to be a useful tool to detect the users' attitude towards systems.

4.1 Usability in Mobile Applications

As we have already mentioned, mobile devices faces several issues in terms of performance. However, this is not the only problematic part as usability and usability testing with such devices also presents challenges.

Considering the usability part, designers have to address several factors to obtain user friendly devices. Perhaps the most challenging one is the screen size. Mobile devices come in a large variety of sizes, making it hard to design the user interface. Designers have to figure out how they can present information in a clear way when the space is limited. Another issue related to screen sizes is that applications sometimes are available for a large variety of mobile devices, so designers have to consider how the application will look at several different screen sizes.

As for usability testing, one of the problems is the small screens that make it hard for the observer to keep track of what the user is doing. We will discuss this matter in Chapter 7.

4.2 Acceptance Models

Earlier in this chapter, the usability in general and in context of mobile devices was briefly introduced. This section presents two acceptance models where one of these is designed for mobile services.

4.2.1 Technology Acceptance Model

The Technology Acceptance Model (TAM) aims to model how users come to accept and use a technology. Several factors influence the users' decisions where the two most notably are Perceived Usefulness (PU) and Perceived Ease of Use (PEOU). Perceived Usefulness and Perceived Ease of Use is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance" and as "the degree to which a person believes that using a particular system would be free from effort", respectively[Davis, 1989]. The model is illustrated in Figure 4.1.

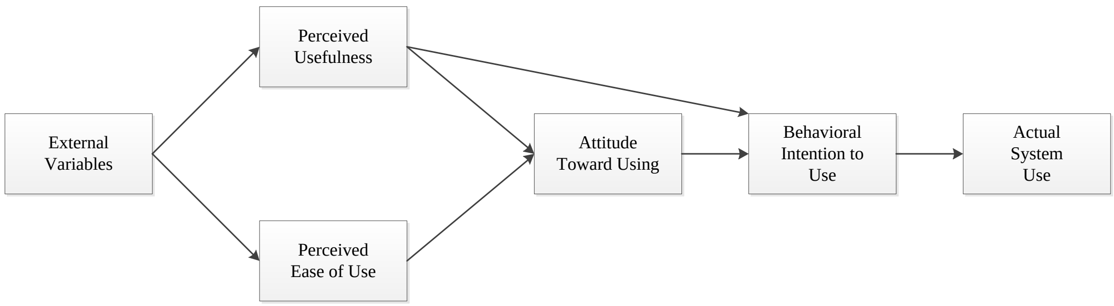


Figure 4.1: The Technology Acceptance Model by Davis [1989]

TAM is a widely used acceptance model, and when applied to various new information technologies, the model successfully predicts user acceptance[Gao et al., 2008]. However, [Bagozzi et al., 1992] states that the actual usage might not be a direct or immediate consequence of user's attitudes and intentions. This assertion builds on the fact that TAM implies that users will be free to act without limitations when they form an intention to act, while they in reality will be constrained by time, environmental or organizational limits, limited ability and unconscious habits.

4.2.2 Mobile Services Acceptance Model

Several models and theories have been created to assess user acceptance, such as TAM. Previous models did not take mobile information systems into account, and is therefore not particular suited for user acceptance testing for mobile systems. However, [Gao et al., 2008] proposes Mobile Services Acceptance Model (MSAM) which is an extended technology acceptance model based on TAM, that also studies the factors that influence user acceptance of mobile applications in particular. In addition to the considerations of TAM, it considers the influence of trust, context, and personal initiatives and characteristics in user adoption of mobile information systems.

MSAM introduces the constructs Context, Trust and Personal Initiatives and Characteristics, and argues that these influence, either directly or indirectly, the user's intention to use the mobile system. Location and identity is examples of Context and have a direct impact on Perceived Usefulness and Perceived Ease of Use as depicted in figure-reference. As an example, an application taking location into account is more useful than an application that does not take it into its consideration. Trust concerns the user's beliefs in the mobile system considering potential security and privacy threats. Trust has always been an important topic, but as applications and services have evolved, the information sent and stored may include personal information or other information that others should not know about, which makes security and trust even more important. For example, communication links are not always secure (e.g., eavesdropping) and data encryption is therefore required to secure the data being sent and received. Age, gender, knowledge and skills, educational background, culture and preference are the elements in Personal Characteristics. Personal Initiatives can be defined as the user's willingness to try out new applications.

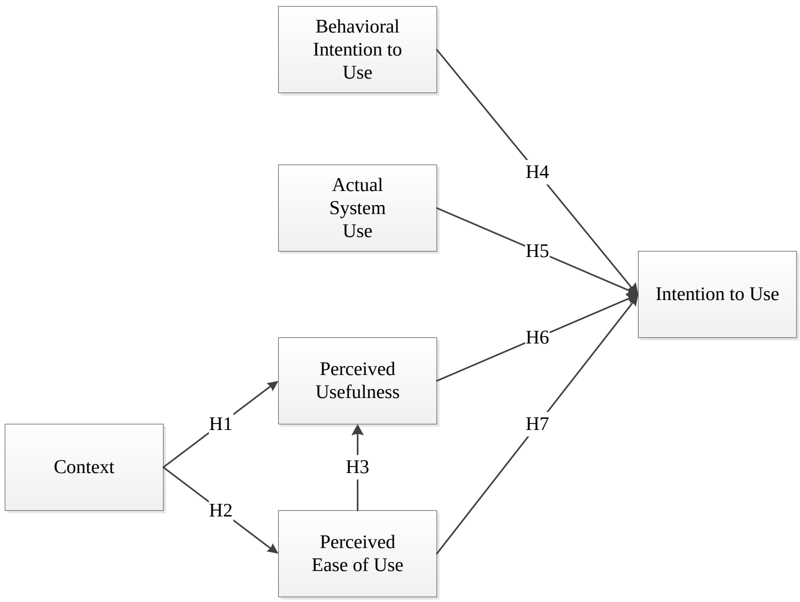


Figure 4.2: The technology acceptance model by Gao et al. [2008]

[Gao et al., 2008] states that Perceived Usefulness and Perceived Ease of Use are positively correlated with the user's intention to use. Also, Perceived Ease of Use has been confirmed to have a positive effect on Perceived Usefulness in previous research studies [Moe, 2009].

A location based service named FindMyFriends aimed to provide people with the location of each other during a festival in Trondheim. MSAM was applied to the evaluation of this system, and demonstrates the applicability of this model as well as suggesting that the model is suitable when evaluating user adoption of mobile applications [Moe, 2009].

This model has also been used in other studies. In order to evaluate Mobile Student Information System (MSIS), [Moe, 2009] used this model. Also, [Asif, 2010] used the same model to evaluate MSIS after implementing improvements. However, they are using a revised edition of MSAM so it better captures the distinctive characteristics of the MSIS system. As in their case, we also had to make smaller adjustments to some of the items in some of the constructs. These adjustments have been approved by our supervisor, John Krogstie, whom is one the authors of [Gao et al., 2008].

[Gao et al., 2010] proposes seven hypothesis that is illustrated in Figure 4.2, and relates to users' adoption of mobile services in correlation between the measurement constructs. These hypothesis are defined as follows by [Gao et al., 2010]:

- **H1:** The appropriate context has a direct positive effect on Perceived Usefulness
- **H2:** The appropriate 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 have 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

Part III

The MTSR System

Chapter 5

The MTSR Services

In this chapter, we will take a look at what features the system has to offer. We will also discuss the requirements of the systems, the features in greater detail and see how they fulfill the requirements.

The Mobile Tourist Service Recommender system is intended to help tourists find points of interest such as hotels and restaurants in order to let them schedule their time more efficiently and increase the probability that they will visit places that they'll actually enjoy. By taking location, among other things, into the consideration, the system aims to provide better information to users.

The MTSR application has many features that will help the users in their search for POIs that they will like. To provide a clear and simple interface, the MTSR client uses a standard Google Map which can be panned and zoomed. In order to separate different functionality, a tab bar with three elements is used where each tab has its own purpose.

The most common functionality is found in the "Find"-tab, where users may search for POIs, get information about them or the route. Several facts may be stored, namely title, address, photo, description, phone number, opening hours, category, type (e.g. Italian restaurant) and rating. However, what information that is available depends on what have been entered into the system. We will take a look at how information is collected later.

Figure 5.1 display this feature. In this figure, the user has searched for places to eat and is represented with a number of restaurants. More information about these places may be found by simply clicking on one of the icons.

"Travel plan" is the second tab and is used whenever the user wants to obtain a complete itinerary. The complete itinerary will find restaurants and activities recommended for the user, and will do so by taking time as a factor.

The last tab, the "Pending ratings"-tab, is used to display places the user have or may have visited and has yet to rate. This feature ensures that the user is reminded to rate places. There are two ways a place may be rated, either by the user explicitly selecting the place and rate it, or by pending ratings received. The latter will be presented in the next chapter.

In the rest of this chapter, we will dive further into these activities and see their connection to the requirements. The requirements and classifications to come, is partly of the original work by [Wium, 2010] and partly of the new iteration's work.

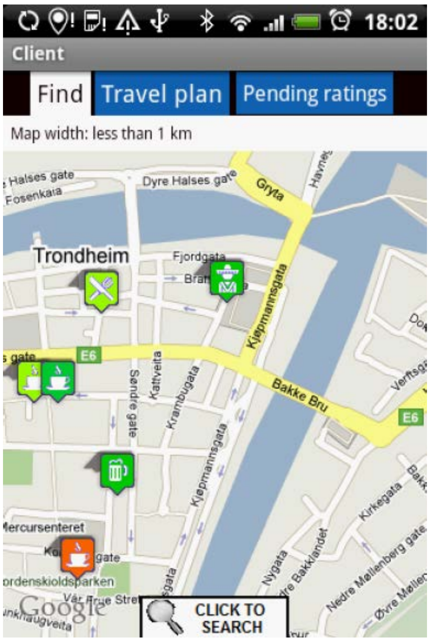


Figure 5.1: Results displayed in map after searching for restaurants

5.1 Requirements

During the first iteration of MTSR, [Wium, 2010] stated the following requirements that we have inherited.

Allow tourists to easily find information on- and physically locate points of interest

- **Requirement 1 (R1):** The system must have pre-defined requests for typical tourist services as part of the user interface.
- **Requirement 2 (R2):** The system must allow the user to search freely after information
- **Requirement 3 (R3):** The system must provide the user with the distance and route to each service from the user's current location
- **Requirement 4 (R4):** The system must allow the user to request and manage complete travel itineraries
- **Requirement 5 (R5):** The system must provide the user with automatic/proactive recommendations

Facilitates transparency and user control in the personalization process

- **Requirement 6 (R6):** The system must provide explanations of the personalization process to the user
- **Requirement 7 (R7):** The system must allow the user to override the adaption strategy taken by the system

Gather information on the user's preferences in a way that is non-obtrusive, accurate and requires little effort from the user

- **Requirement 8 (R8):** The user must be able to control the personal preferences used to create recommendations
- **Requirement 9 (R9):** The system must allow the user to express his opinion about the services presented
- **Requirement 10 (R10):** The system must automatically provide the user with an overview of places the user has visited

5.2 Context Awareness

We introduced context awareness in Chapter 2, and Wium chose to classify the context properties as follows:

- **Field of interest:** The area currently displayed to the user. This constitutes the area that will be searched for POIs
- **Location:** The geographical location of the user in terms of latitude and longitude
- **Preferences:** The set of opinions expressed by the user
- **Time:** The current time of day at the user's location
- **Visiting history:** The set of places that the user has physically visited

In addition to these, we have added the following context properties in order to support the new features of the system:

- For users (representation of the demographical data)
  - **Age:** Specified in certain age intervals (e.g. '25-44')
  - **Gender:** Either 'Male' or 'Female'
  - **Occupation:** The job or profession of the user (e.g. 'Sales')
  - **Type of holiday:** Whether the stay is a 'One-day stay', 'Regular city break' or 'Backpacker'
  - **Nationality:** Where is the user from in terms of certain regions (e.g. 'Central Europe')
  - **Budget:** The tightness of user budget, represented as 'Low', 'Medium' or 'High'
- For points of interest (representation of the demographical data)
  - **Night-life:** Type of night-life (e.g. 'Pub')
  - **Sightseeing:** Type of sightseeing (e.g. 'Art museum')
  - **Cuisine:** Kind of food (e.g. 'Italian cuisine')
  - **Budget:** How expensive it is, represented as 'Low', 'Medium' or 'High'

As of the original system, the strategies used to adapt the user interface and information sent to the user is the following:

- The maps centers to match the user's location
- Searches for POIs are filtered by preferences and field of interest
- Requests of travel plan is filtered by preferences, location and time
- Based on location and preferences, the user is sent automatic recommendations
- Based on location and visiting history, the user is sent rating requests

5.3 User Interface

We have already introduced some of the most basic features of MTSR. Now we will show how the requirements are resolved by presenting the typical tasks and activities:

- Receiving pro-active recommendations
- Evaluate information received from the system
- Searching for information
- Providing feedback
- Using the travel plan

5.3.1 Receiving Pro-Active Recommendations

This feature fulfills **Requirement 5:** The system must provide the user with automatic/pro-active recommendations.

This feature was introduced to allow the tourist retrieve recommendation without having to search, and is favorable for two reasons: 1) it can be annoying/difficult to search in outdoor environments using a small screen, and 2) without knowing that there is something interesting nearby, it can be difficult to search for it.

Such a feature has been used in earlier prototype systems, like TIP [Hinze and Buchanan, 2006] and CRUMPET [Schmidt-Belz et al., 2003]. TIP referred to this feature as "browse by walking". Their research includes models and details on interaction design reference. However, information about how to include the feature as part of the user interface is nearly non-existent.

In order to overcome this lack of information, general usability guide lines have been used to make the process as non-obtrusive as possible from the user's point of view. The following measures were taken:

- Never recommend the same place twice
- There has to be at least 10 minutes between such recommendations
- If the recommendation is not addressed by the user within three minutes, the recommendation will disappear
- A light vibration informs the user of the recommendation (as the user most likely will be carrying the mobile device out of sight)

5.3.2 Evaluate Information Received From The System

In this section, we will collaborate on how the system fulfills **Requirement 3:** The system must provide the user with the distance, route and description of each service, and **Requirement 6:** The system must provide explanations of the personalization process to the user.

After receiving information about POIs, the user will typically seek answers to questions like [Cheverst et al., 2000]:

- Is this something I would like to do?
- How much time and effort would it take me to get there?

To see how MTSR aids the user in answering these questions, we will look at them separately and explain the thinking behind the solutions/approaches.

Is This Something I Would Like To Do?
To aid the user in answering this question, three measures have been introduced by [Wium, 2010]:

- Color of icons representing POIs indicates its predicted subjective value in relation to the user
- Explanations
- List of "top rated"-places based on correlation with other users

Instead of plain markers, a variety of icons represent the different categories and types. Furthermore, these icons are displayed in colors ranging from red to green, symbolizing the predicted subjective value in relation to the user. This color scheme has been used in the design of MobyMapRec [Averjanova et al., 2008], and reports say that this approach strongly contributed to efficiency and satisfaction of the system, as it allows users to compare and evaluate POIs directly. Combining these two elements, the user has more information readily available in the map without complicating the user interface.

As a second measure, explanations that show the average rating as stars (between 0-5) and a textual description explaining the group of people that make up the average rating. A simple star-scale can be useful as overall metric of the quality of the recommendation. However, as the subjective element of preferences can be so spread in the tourism domain, [Wium, 2010] argues that the user can benefit from an additional mean for quality assessment. This mean is information about the similarity in taste between the user and the group the average rating is based on.

To get a better understanding of how this is carried out in the application, consider Figure 5.2. The user has selected the main building of NTNU and we can see the dialog displaying some information about it. The explanation discussed, is the text right below the stars.

The last measure is found in the "More info"-view that users may enter by pressing "More.." in the dialog box appearing when pressing POIs. In the bottom of this view there is a section that lists "top rated"-places that other users, who also liked the current place, enjoyed. Upon pressing one of the items in this list, the application will automatically center the map on this point of interest.

All of these three measures also aim to contribute to the overall goal of making a transparent system interface. As for the explanations-part, Wium tried to expose the underlying filtering algorithm in a way that highlights the importance of rating in relation to the personalization

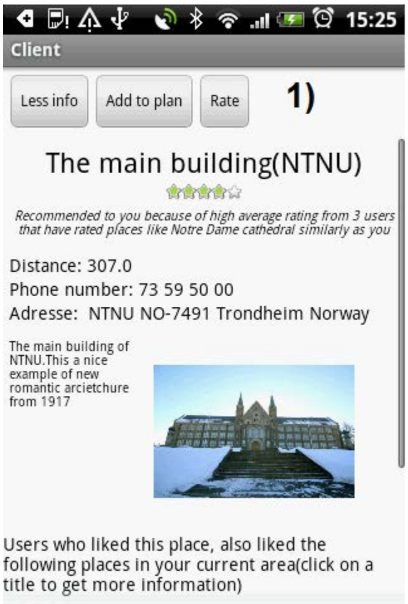


Figure 5.2: Detailed information for a POI

process. The latter measure is meant to further stress how important ratings are in relation to the system's perception of preferences.

How Much Time And Effort Would It Take Me To Get There?
The route and distance to the selected point of interest is presented to users through the map and information box, respectively. This route is automatically generated as soon as a place is selected, so goes for the distance as well.

5.3.3 Searching For Information

After researching relevant topics, [Wium, 2010] found out that tourists are most likely to be looking for places to eat, sleep or tourist attractions (e.g. museums). Easy access to these categories should be present in order to be a useful personalized mobile tourism application [Cheverst et al., 2000]. Searches are conducted in "the sliding search window" where these three categories (represented by "Eat", "Sleep" and "Experience") are readily available to users. The user may press one of these and the press search to get results. Upon pressing search, the search field will disappear and the results will be displayed. By typing a custom text into the search field, a free/custom search is performed.

The system will only search for POIs within the field of interest which is the current area displayed on the map. Users are free to pan and zoom this map/field of interest as they see fit. Filtering by personal preferences can be bypassed by unselecting the checkbox "Only stuff recommended for me". Doing so will search for all POIs that match the text in the search field.

These features fulfill **Requirement 1:** The system must have predefined requests for typical tourist services as part of the user interface, **Requirement 2:** The system must allow the user to search freely after tourist services, and **Requirement 7:** The system must allow the user to override the adaption strategy.

5.3.4 Providing Feedback

We will now see how the system implements functionality to meet **Requirement 8:** The user must be able to control the personal preferences used to create recommendations, **Requirement 9:** The system must allow the user to express his/her opinion about the services presented, and **Requirement 10:** The system must automatically provide the user with an overview of places the user has visited.

Allowing the user to control his/her personal preferences is difficult in this context as the user would have to be exposed to the full model, which is very complex, and adjust it explicitly. To avoid this complexity, MTSR let the user change and control the underlying user model by providing feedback. This makes the process easier, but the user does not possess complete control of the model. Further, it is believed that the user unlikely would use and maintain the complex solution as it is challenging presenting the needed information in an intelligible fashion on mobile screens. This deduction is representative for the system in both iterations. However, as the new recommender system is activated the user will be able to control his/her personal preferences by demographical data about themselves along with their feedback. This will result in the same flexibility, but the recommendations will eligible be better. Users are free to change these demographical data and may play with them to receive other recommendations. As this can give unforeseen results not personally made for the user, we do not consider this as a sign of flexibility.

Figure 5.3: Dialog box with information about a POI

The process of rating places were focused on being as simple as possible and thereby differencing it from other systems that did so by exposing their full content based model [Wium, 2010]. MTSR presents two ways to provide feedback:

1. **Direct rating:** By simply clicking a point of interest in the map, the user is presented with

a dialog box as shown in Figure 5.3. From this point, the user clicks the “Rate”-button and gets a row of stars presented. By using his/her finger, the user determines the number of stars to give the place.

2. **Pending rating:** The system keeps track of all the places visited by users. As the system detects a new visit, it automatically sends a request for rating the place in question to the user. This is presented to the user by a vibration and the “Pending ratings”-tab begins to blink. In this tab, a list of rating requests, or pending ratings, is displayed for the user to rate, investigate or remove it. A system for determining when a user has visited a place is presented in the next chapter.

5.3.5 Using The Travel Plan

The last requirement, **Requirement 4:** The system must allow the user to request and manage travel itineraries, is realized by the “Travel Plan” feature.

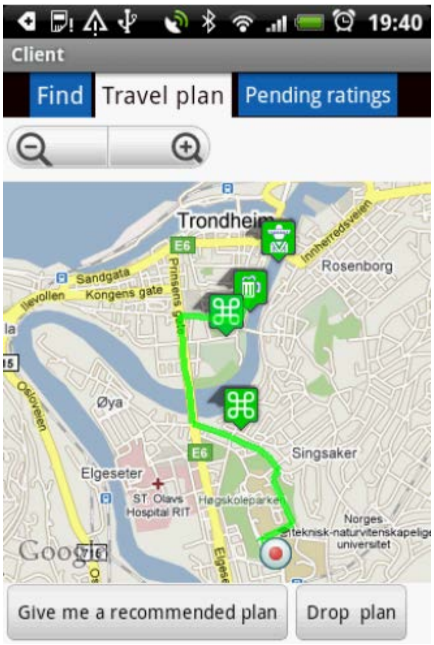


Figure 5.4: Travel plan

Accessing the “Travel plan”-tab, the user is presented with the view illustrated in Figure 5.4. Again, the simplicity of the process has been the priority. The only interaction needed by the user to obtain a travel plan is to press the button “Give me a recommended plan”, and it is easily removed by pressing “Drop plan”. Requesting a travel plan will make the system generate a set of places that includes activities as well as places to eat, all done while taking time into consideration. Remark that this means that the user will be presented with alternatives for lunch and dinner only if the time left is sufficient. Users are unable to influence this process. The reason for this is simply that it was believed that it would complicate the interface more than it would benefit the user. Places in the travel plan may be removed by clicking them and pressing

"Remove from plan", and added by accessing the "Find"-tab, selecting the preferred place and press "Add to plan".

Wium studied previous systems, and discovered that Requirement 4 is an important feature for the two following main reasons:

- It can provide the user with much travel information quickly
- It allows the user to save recommendation/information on POIs between sessions with the application

He also discovered that previous systems, such as CRUMPET and the system by [Yu and Chang, 2009], required a considerable amount of information before plans could be retrieved.

5.4 Collection of Information Regarding Points of Interest

As of the last edition of MTSR, we have introduced a GUI that will collect information entered by the user or administrators. This data will go through a process, which we will shortly describe, in order to ensure that all needed data is provided, that the information is validated and that the information is presented in a way that is suitable for the system.

We have developed a three step process where a user is needed for the first two. The first step is entering information into a form. This form has been carefully constructed and provides some nifty features. Most of the fill in spots are regular text, but we do have some drop down menus where the possible values is limited to a predetermined number of choices, and to make sure that the geographical attributes are correct we have developed a JavaScript that devolves geographical points (in terms of latitude and longitude) from addresses, as well as devolving them from marked spots in the map.

The second step is an overview over all the values that have been entered. This is the validation-process, and has been constructed to let the user verify that the entered information is correct. A map is also provided so it is more convenient to verify the map data if a spot were used to represent the position of the point of interest in the last step instead of an address.

For the last step, the system will make appropriate transformations of the data. Data from the form is gathered and inserted into a database query after refinements. Refinements include giving geographical data proper Spatial Reference System Identifier (SRID) and calculating area based on the location (represented as a polygon). The step is finalized by executing the query.

So far the data being used has only been entered by the test observers. One of the reasons of this is that the system is developed by an individual or group for one period and then it is shut down for some time before another individual or groups starts the next iteration of development. There is no centralized server which can be used for several iterations, meaning that all data from the last iteration is lost (i.e. information about users, POIs and ratings). This feature will more likely be used by users as well if the system is made available for the public. As this will make the application more convenient for them, the users will contribute more.

Chapter 6

The MTSR Architecture

In this chapter we will dive into the architecture of the MTSR system.

The system utilizes the widespread client-server model as depicted in Figure 6.1, and consists of two servers, a client and the Google Maps information service. The backbone is the MTSR-server where the main functionality is and where the service requests are handled. This server is depended on the second server, namely the Postgis server, which is the database where information about users, POIs, geographical data, and so on is stored.

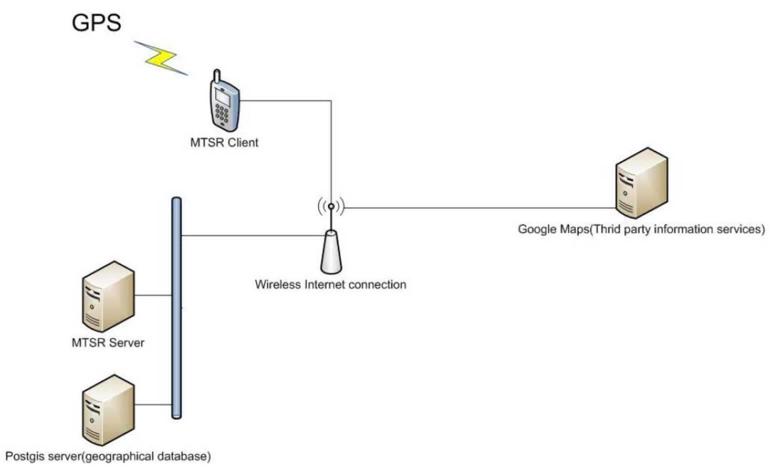


Figure 6.1: Functional architecture (courtesy of Wium [2010])

The last entity created is the MTSR client which is depended on the Google Maps information service. Users can access this client using mobile devices running Android. When using the application, users can request information using the content delivery technology ***pull***, and they might receive pro-active recommendations and pending rating requests which are delivered using ***push***.

6.1 The MTSR Client

In this section, we will investigate the MTSR client further by first explaining the choice of platform before we look at some components of the Android SDK, and finish off on how the client operates.

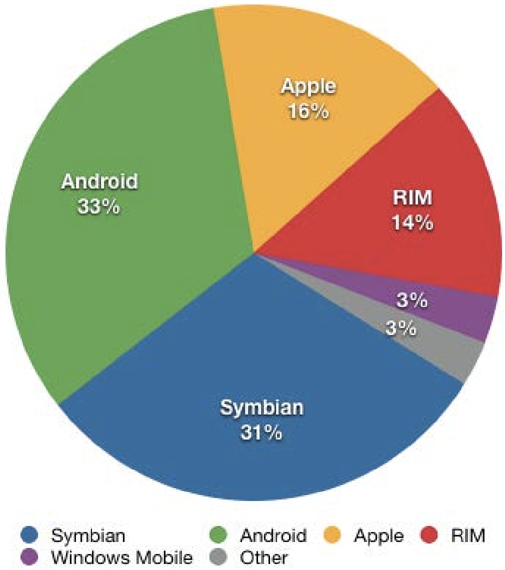


Figure 6.2: Share of worldwide smartphone sales in 2010 Q4[Canalys, 2011]

6.1.1 Choice of Platform

As mentioned, the client runs on mobile devices running the Android OS. Android provides convenient interfaces to positioning hardware such as GPS, dedicated libraries for developing advanced maps applications and framework for handling events from touch, which are all important parts of the target solution. Also, Android has had a steady growth in its market share which is 33% in the fourth quarter of 2010 (see Figure 6.2), making it a very attractive platform to make use of.

6.1.2 Relevant Compontents of the Android SDK

The Android SDK consists of several classes and services. In order to understand the architecture, we will have to investigate two of them, namely the activity class and the location manager class.

Android uses an activity to present the user with a visual interface. Such activities run as single threads in a Linux process and uses remote procedure calls to communicate with each other. They also handle user interactions. The MTSR client makes use of three such activities, specifically “Find”-, “Travel plan”- and “Pending ratings”-activity.

Information about location is provided through one of Android’s services, but in order to access this information we will have to use the location manager class. This class provides all the information we need regarding location, including a convenient interface to the underlying GPS-implementation. This is one of the great flexibilities of Android, the developer only have to inform the system of how often, fast or far the user's position have to change before receiving updated information.

6.1.3 MTSR Client Operation

Perhaps the most significant operation of the client is to provide the user with a map and additional information such as restaurants' name, location and description. By extending the MapActivity-class in Android, developers are able to provide an interactive Google Map. Along with the map, we are also able to access functions that the Google Maps API supports. By implementing a custom Android SDK Overlay-class, icons and other information is displayed. The origin of the icons is the "Google Maps Icon project" which is trying to make the interaction with Google Maps more user friendly[Google, 2011a].

In addition of displaying information, the user may request information. This is typically the action of searching for POIs, such as restaurants. In cases like this, the client calls the corresponding method at the MTSR server interface. Among methods in this interface, we have retrieving POIs (both recommended and not recommended), sending ratings and manage travel plans. The server handles the request and sends back its response which is interpreted by the activity that generated the request in the first place and performs appropriate actions.

The remaining action at the client is to make sure that the server is updated on the user's current context. To minimize the user interaction needed, the client utilizes the location manager class that can detect when the user has moved 30 meters or when the time between updates have passed five minutes. Also here, the client generates a request to call the corresponding method on the MTSR server interface.

Combining all this, Figure 6.3 depicts the high level class diagram of the MTSR client.

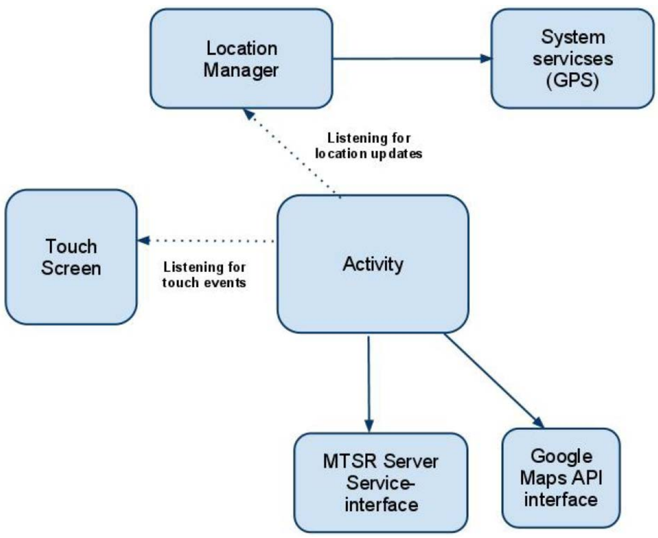


Figure 6.3: Functional architecture (courtesy of Wium [2010])

6.2 The MTSR Server

The server runs on an Apache Tomcat application server and is implemented as a web application with three layers, specifically the service layer, the business layer and the data access layer. We will take a closer look at these and then provide a short description on the third party components and frameworks used by the server.

6.2.1 The Service Layer

The service layer handles all the requests received from users. It calls the appropriate methods and sends the resulting data from the business layer. As depicted in Figure 6.4, it is implemented using Java Servlets.

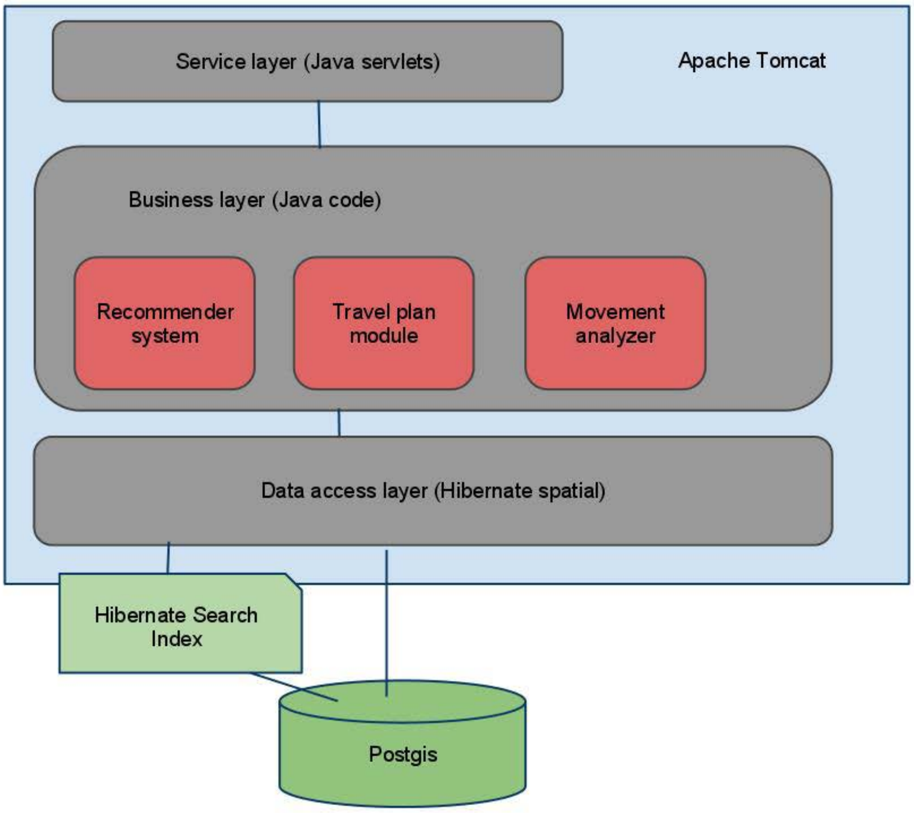


Figure 6.4: The client-server model used by MTSR (courtesy of Wium [2010])

6.2.2 The Business Layer

The business layer serves requests from the service layer, and uses the data access layer to obtain and send necessary information. Within this layer, we have three modules: 1) Movement Analyzer, 2) Recommender System, and 3) Travel Plan Module. Each of these implements an abstract interface, making it easy to exchange implementations. We will soon take a deeper look at these modules.

The MTSR currently has two recommender systems, but utilizes only one of them. One of the recommender systems has been present since the start, while a new recommender system was introduced by [Røine, 2010] using a specialized recommender system developed by [Lillegraven and Wolden, 2010]. The latter has yet to be tested, and is not used in the test phase carried out in this Master's thesis.

6.2.2.1 Movement Analyzer

In order to figure out whether the user should receive a pro-active recommendation, a pending ratings request or nothing, the system uses a movement analyzer.

Send Pending Rating Request?
As mentioned before, the system receives context updates from the user after certain criteria are met. When the system receives an update, the movement analyzer will investigate the updated information against the existing information. Each POI has a geometrical attribute (polygon) that represents the POI's area, and by using this information combined with the updated information the system can determine whether the user is within the area of a POI and which POI that is by using simple spatial mathematics. If the user is not within such an area, the system will take no action. However, detecting that the user is within such an area will trigger the system to check if this is the same POI as the POI last visited by the user. Given that it is the same, the system calculates the time spent there so far and updates the information for last known visit and in the event of this time exceeding 30 minutes, the system will send a pending rating to the user. If the area belongs to another POI, the system will try to determine whether the user stayed in the previous area for a period of over 30 minutes and send a pending rating if so. As a last step, the system will update the last known visit of the user.

Send Pro-Active Recommendation?
The context update also starts the process of determining if the user should receive pro-active recommendations. Upon reception of such data, the system creates a circular spatial area with the location of the user as center and the radius as the maximum distance from the user. All POIs within this circle is investigated by checking if they have been sent to the user before. If not, they will proceed to the next step which is collaborative filtering that may output a set of recommendations. Given that this set is not empty, it will be sent to the user.

Concerns

- The Movement Analyzer requires the user to provide information about his/hers whereabouts. As no information will be sent back to the user at most times, a lot of network bandwidth and computer resources are wasted. One could argue that it is necessary in order to achieve such a feature, and therefore not wasteful. In either case, this feature might be quite expensive where mobile data transfer rates are high (which they will be if the tourist is outside his/her country).

- The accuracy of the mobile devices may present a problem as some of the algorithms require very accurate geographical locations. In the field test conducted by [Wium, 2010], the device's (HTC Hero Smart) GPS provided sufficient accuracy. This is promising, but different devices will have different performance. Also, the GPS-module in mobile devices is not the only factor related to GPS accuracy. Satellite and receiver clock error, satellite geometry measures and tropospheric delay to mention a few, are all sources that can impact the accuracy [Klobuchar, 1996][El-Rabbany, 2006].

- As the pending ratings algorithm uses polygons (representing the area of POIs) to determine whether the user is at a specific POI, it may be impractical or impossible to use external content providers.

6.2.2.2 Recommender System

As mentioned before, the system has two recommender systems where only one of them is active. We will present both of them in this section.

The Original Recommender System
As part of the original design of the system, [Wium, 2010] created a recommender system that used collaborative filtering. This implementation was mainly created to fill the basic need of a recommender system. Certain modules are depended on that such a system is available, and they needed an operational recommender system for the field experiment. Their contributions are not connected to the accuracy or efficiency of this part of the system. However, they hope that coming iterations of the system will address this implementation against other options, and that they are compared using metrics such as precision [Burke, 2002].

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Figure 6.5: The Pearson product-moment correlation

To implement this, they used the Pearson product-moment correlation coefficient (see Figure 6.5). In this equation, the vectors X and Y are the two users being evaluated. The resulting value, r, represents the linear dependency between the users' ratings, and will have a value between -1 and 1. The following list shows the actions taken by the system when the method *createRecommendations* (the method that creates a list of recommendations based on given POIs) is called.

1. Go through the list of POIs that is provided and fetch all users that have provided ratings to one or more of these places
2. Use the Pearson correlation equation on all found users and the user searching for recommendations
3. Users with low correlation are removed
4. For the users left, calculate the average rating for each point of interest

5. Fetch the POIs with the highest average rating
6. The resulting POIs are then used to create recommendations
7. For each recommendations, insert explanation and prediction

The Added Recommender System
Lillegraven and Wolden [2010] presents a recommender system designed for tourism using a location-aware Bayesian model. The recommender system addresses the cold-start user problem by using a demographical survey involving six characteristics, namely age, gender, occupation, type of holiday, nationality and budget. The presented overview of the recommendation process is illustrated in Figure 6.6 and consists of several stages.

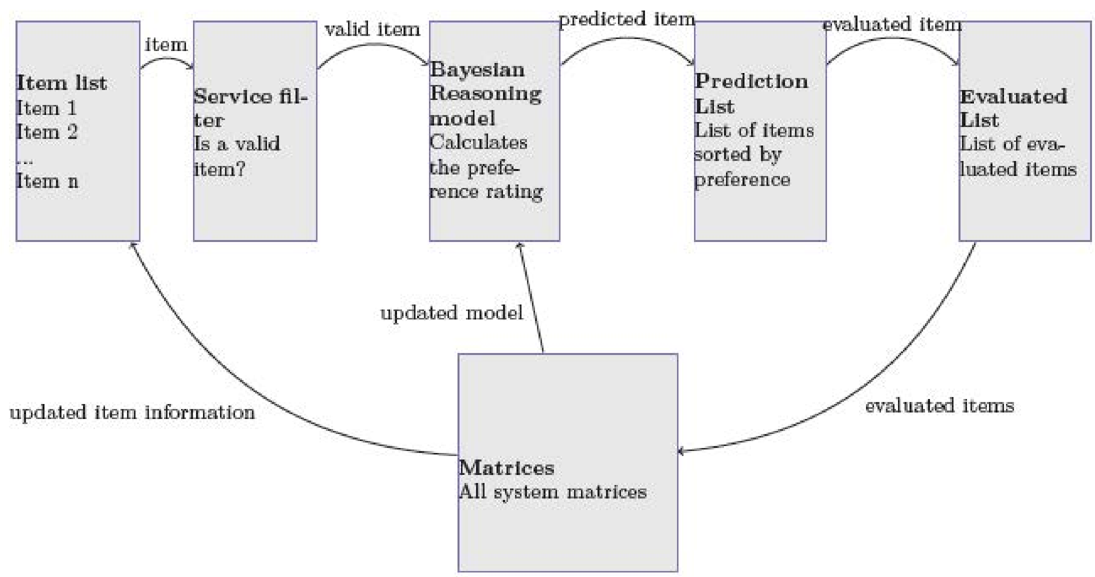


Figure 6.6: An overview of the recommendation process (courtesy of Lillegraven and Wolden [2010])

We will now take a look at how this system operates. If a new user enters the system, the person concerned is required to submit the needed demographical information. The system starts off with a list of POIs which are sent to the service filter where the number of valid items that will be used to calculate the preference rating is minimized. By using the *Bayesian reasoning model*, a sorted list of items is created. Ratings are uploaded to the server which updates the Bayesian reasoning model. When the user provides new information, such as ratings, the Bayesian reasoning model needs to be updated. By using several matrices with simple Boolean-variables, they state that they increase the speed of calculations. This, however, requires a specific user- and item-model. Detailed description of the process can be found in [Lillegraven and Wolden, 2010].

As mentioned, this approach requires a specific user- and item-model. To overcome this obstacle, users and items/POIs get an extra attribute each, specifically a matrix with Boolean-values that will represent them.

6.2.2.3 Travel Plan Module

The implementation of this module uses [Yu and Chang, 2009] as basis. It will take the user's location and time as foundation for the coming steps. A set of POIs are sent to the recommender system which returns one recommendation that gets added to the plan. The location is updated to the location of this point of interest, and the time is calculated to be the time now plus the estimated visiting time. Then the process starts again by sending a set of POIs to the recommender system, and so on. It will stop this process when the time has reached bed-time.

The process used by [Yu and Chang, 2009] required information from the user and that content-based filtering was applied. These dependencies were removed by making the module function with any underlying filtering algorithm, and by not requiring any information from the user.

[Wium, 2010] states that this implementation is very simple and that it will probably make mistakes considering what is optimal for the user. Even though, they believe that it will provide a useful enchantment as long as the users are able to correct such mistakes by adding and removing objects manually.

This simple implementation poses a couple of limitations. First of all, it makes many assumptions such as the bed-time for the user. Secondly, after the recommendations closest to the user has been found, the POIs in the plan drift in a certain directions. If highly recommended objects are positioned in the opposite direction of this drift, they may be left out.

6.2.3 The Data Access Layer

The data access layer provides an interface between the business layer and underlying database. It simplifies the process of storing and receiving of Java-objects. By using this approach, the data access layer returns a reference to an object with all of its attributes, thus eliminating the need to handle data, in terms of a row of fields, retrieved from a database. Figure 6.7 illustrates the model that has been employed.

Location attributes such as location and coveringArea are represented by the JTS classes 'Point' and 'Polygon', respectively. "Point of intrest" represents a tourist service, "User" is the mobile user, and "Visit" represents POI that the user physically visited. Each Visit is marked with the date and time, as well as the duration of the visit. User's fieldOfIntrest is the area that is currently being displayed to the user, and is used to limit the area that is searched for POIs. Users are given recommendations, represented by "Recommendation", which is connected to both POIs and Users. The system also has a connection between users and POIs by keeping track of all users that have rated the different POIs. The user model is made up of a set of preferences, represented by the "Preference" class. POIs have several attributes, including their title, location, category and area. "Rating" is the last class and simply consists of the ratings given by users.

MTSR uses data access objects (DAOs) which is an object that provides an abstract interface to the database. It provides a mapping from the system calls to the persistence layer. This has been designed so that it will be easy for further iterations to extend this architecture. Even if the basic spatial database is exchanged with an external resource, the same interface can be used. The only addition needed is new implementations of the DAOs. The reason for this design, is that [Wium, 2010] believed that MTSR may need tourist service information from external content providers.

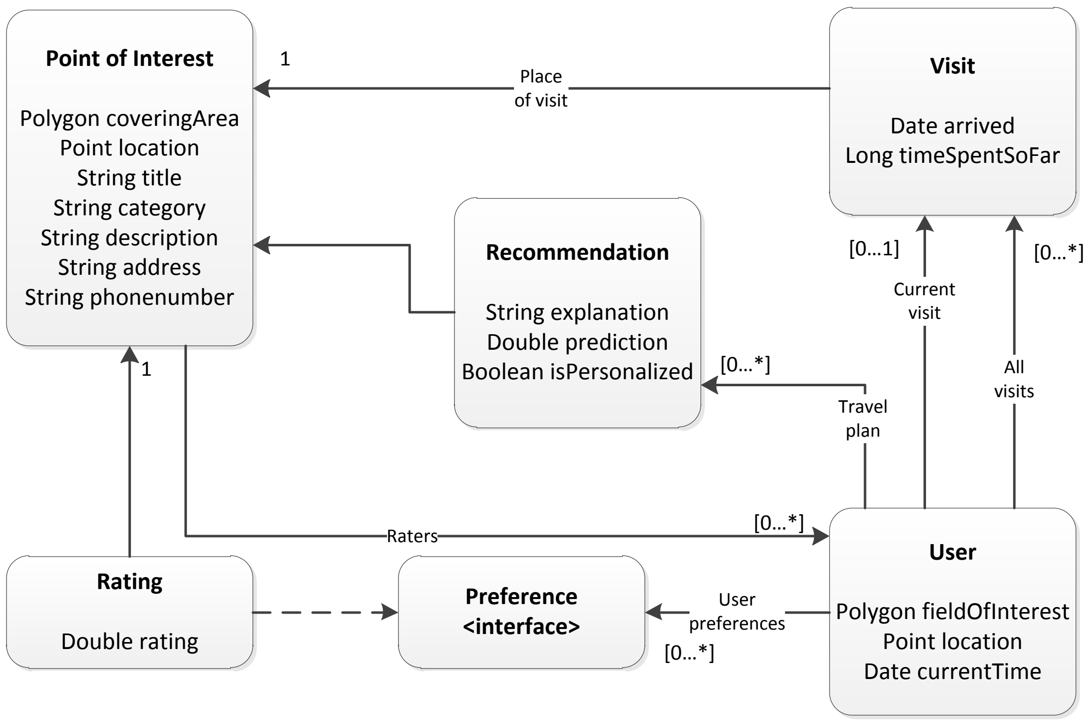


Figure 6.7: Domain model

6.2.4 Third Party Components and Frameworks

The original MTSR server utilized several third party components and frameworks. These are still present in the system and used in the same manner as before. The following components and frameworks are used:

Apache Tomcat
Apache Tomcat is an open source software implementation of Java Servlets and JavaServer Pages [Tomcat, 2011]. It is a widespread technology and is even used by some major corporations such as WalMart [Ciurana, 2011]. MTSR uses this technology as its container for the server since it supports Java Servlets.

Hibernate Search
By combining Hibernate Core with Apache Lucene search engine capabilities, Hibernate Search brings the power of full text search engines[Search, 2011].

Hibernate Spatial
Hibernate Spatial is also open source and is an extension for handling geographical data[Spatial, 2011]. This technology has simplified the MTSR implementation.

JTS Topology Suite
This technology is open source and completely written in Java, and provides an API for 2D spatial predicates and functions[Solutions, 2011]. The inclusion of this technology allows us to

check spatial intersection among other things.

PostGIS
PostgreSQL does not originally support geographical objects, but this lack of functionality is overcome by using PostGIS as well. This results in a spatial database[PostGIS, 2011] that we use in MTSR to store and manipulate geographical objects.

Java Servlets
As we have shown earlier in this chapter, Java Servlets have been used to implement the service layer. Java Servlets handles HTTP-messages[Oracle, 2011] which most devices are capable of using, making it suitable for this system.

Part IV

Evaluation and Results

Chapter 7

Evaluation

7.1 Introduction

The Mobile Tourist Service Recommender system is intended to help tourists find points of interest such as hotels and restaurants in order to let them schedule their time more efficiently and increase the probability that they will visit places that they'll actually enjoy. To inspect how well the system meets the aimed goals, we have completed a test phase using usability scenarios and a user acceptance survey. The testers were recruited through academic and personal channels resulting in a test group with various fields of study and experience with mobile services. We were also interested to see whether nationality could have an effect on the result. To inspect this matter we recruited international testers as well, but this group only included 12 of the testers, making the test group too small to make any scientific conclusions.

The recruitment resulted in 47 testers with some variety in field of study, age and experience with mobile services. Among the participants, we had 1 unoccupied and 46 students where 5 of these were bachelor students, 43 master students, and 6 doctoral students. 41 of these students studied at a department of science or engineering. 43 participants were between 20-30 years old, while two were under the age of 20 and two were over 30. Males were best represented with 36 participants, and the number of females was 11. The nationality distribution in this study is 35 Norwegians, 10 Chinese, 1 Swede and 1 Russian. We received a spread distribution in experience with mobile services. For the three alternatives, 0-1 year, 2-5 years and more than 5 years, the numbers of participants were 15, 17 and 15, respectively. This demographical data can be found in (append

At this time, the system has no functionality to handle users in an appropriate manner. Users were hardcoded into the database and client. This is something that should be addressed in coming iterations, and included in the test phase. Especially when the new recommender is going to be tested, as this requires the user to input demographical data and it is very important to investigate how the users feel about entering the required amount of data.

The database was populated with 68 points of interest including 38 restaurants, 20 hotels and 10 experiences of which five restaurants were localized in Oslo. We deliberately did not add any restaurants with good ratings in Oslo of reasons that will be clarified in a subsequent section.

Mobile devices have small screens that make performing usability testing complicated even while staying at a single location. Also, testing a system where users probably find different places makes it more or less impossible to perform on-location testing. Due to these considerations and that the system is depended on Internet access, we decided to perform testing in conference rooms where we knew the wireless connection was stable. We also chose to simulate the position of the user to a location in downtown Trondheim. All testers were informed about this matter.

The users were prompted to answer smaller questions, such as the name of a hotel, during each scenario as well as to comment on what worked, what did not work, response time, and illogical/difficult parts after each scenario. This provided us with valuable feedback

Usability testing on mobile devices introduces some additional difficulties to the “general” test phase, especially because of the small screens. We will discuss the approaches used in this study in Section 7.4.3.

As not all testers are familiar with the geography of Norway, we provided them with a map where Trondheim and Oslo was highlighted.

7.2 Usability Test and Scenarios

In order to test the system, we prepared three scenarios (see Appendix 8.5) where the user was giving tasks that they had to solve using the system. These tasks involved questions a tourist typically would encounter during their stay, which includes finding a restaurant among others, and other features the system provides. At the end of each scenario, they were prompted to comment on how the application worked in terms of response time, illogical/difficult parts, what did work and not, and any other comments that they would like to emphasize. Unfortunately, these scenarios did not include all features and we could not make more scenarios as the existing ones already consumed all the time that we had available (75 minutes). It should be noted that most participants completed the test before this limit, but some did need 75 minutes. Usability testing can make participants uncomfortable, so to make the process more pleasant for the participants we started off with easy tasks and then gradually increased the difficulty as the test went on. Starting off with easy tasks will likely boost the confidence of the participants.

Each test session endured for approximately one hour and during each test we had up to four testers at a time. We provided the users with user guides (see Appendix 8.5) which consisted of screenshots of the application and short explanations to each of these. We decided to do so to let the facilitator focus on observing what the testers did. To further increase this focus, the testers were instructed to ask each other first if they encountered problems.

7.2.1 Scenario 1

In the first scenario the testers were asked to find points of interest and information about them. In this part we wanted to investigate how easy it would be for each of the users to find places and the information associated with them. Some information, such as address, is not readily available when pressing a place on the map and we wanted to make sure the users were able to find.

For the first part they had to find a restaurant and write down the name and address of the restaurant they chose. This included finding a place and the extended information dialog in order to obtain the name and address of that particular place.

As for the second part, the testers would have to use the information about the place they found and use that to find a new place in their current area that were recommended by users who also liked the current place. This task may not be as intuitive as it was designed to be, so we included this to investigate how intuitive this task really is.

7.2.2 Scenario 2

The system may filter points of interest on whether or not they are recommended for the particular user. The user has to make one additional step to get the recommended points. The motivation for the first task is to investigate how easy this is to deal with. The task at hand is finding a hotel that is recommended to the user.

As the system is meant to provide points of interest proximal to the user, it only searches for places that are within the field of interest (the map currently being displayed). This makes the task of finding places far away from the user's current location cumbersome as the user will have to pan the map until it is over the desired area. The second task is therefore to find a restaurant in Oslo.

The second task also explores another interesting case: If the user selects to only find recommended places, as in the prior task, and all places within the current area of the map is not recommended, the system will display a dialog that it could not find any places and that the user could try to uncheck the "Only stuff recommended for me"-option and/or zoom out. We intentionally let all restaurants in Oslo have low ratings so that they were not recommended in order to see if they realized that they had to turn the "Only stuff recommended for me"-option off to get any result.

7.2.3 Scenario 3

The final scenario explores the handling of complete travel plans. This feature is meant to provide several sights, such as museums and landmarks, and a couple of places to eat so that the whole day is planned by the system. All of these points of interest are recommended for the user.

As the initial task for this scenario, the user is asked to acquire a travel plan and write down the name of one of the places. The second task is to add a place to the travel plan which cannot be done from the "Travel plan"-view. The user has to enter the "Find"-view, search for a place and then add it to the plan which may or may not be intuitive to the users.

7.3 The User Acceptance Survey

We have already introduced the Mobile Service Acceptance Model (MSAM) in Chapter 4. This model is the basis for our user acceptance survey. MSAM presents a survey, however this is not directly suitable for our application. Their survey has been used when evaluating MSIS (Mobile Student Information System), a mobile application aiming to aid students in their search of information about schedules, class information and more. As the survey could not be used directly, we had to modify the following items: PU1, PU2, PU3, PU5, TU1, CT1, CT2, CT4 and CT6. These adjustments have been approved by our supervisor, John Krogstie, whom is one the authors of [Gao et al., 2008]. The original survey from MSAM and the survey used in this study may be found in Appendix 8.5 and Appendix 8.5, respectively.

[Gao et al., 2011] did further research on MSAM and detected that a correlation existed between some of the items, making it practicable to remove some as the survey contained a great number of items. Instead of using the stripped-down survey as basis, we decided to include all items so further research could be done to validate the removal of these items.

The resultant survey consists of 33 items where each item has a 7-point Likert scale ranging from "Strongly disagree" to "Strongly agree". These ratings will be analyzed after the test phase and used to estimate the participants' intention to use the service.

7.4 Presentation of the Test Results

In this section we will present the results from the test phase in two sections. First we will discuss the qualitative data from the usability test, before we discuss the quantitative data gathered from the user acceptance surveys. For the first part, the data being analyzed comprehends comments given at the end of each scenario by each user and the observations of the facilitator.

7.4.1 Presentation of Qualitative Data

In this section we will go through the qualitative data gathered during the test sessions. We will present results from the scenarios and tasks in the order they appear in the Usability Scenarios (see Appendix 8.5).

Scenario 1

The first step for the testers was to find a restaurant. At this point, the testers either searched using the predefined terms or they tried to perform a custom search with the distribution 64% (30) and 36% (17), respectively (see Figure 7.1). Testers using the first mentioned approach quickly found POIs and its information. However, testers using a custom search met some problems. For unknown reasons, the custom search rarely gave results.

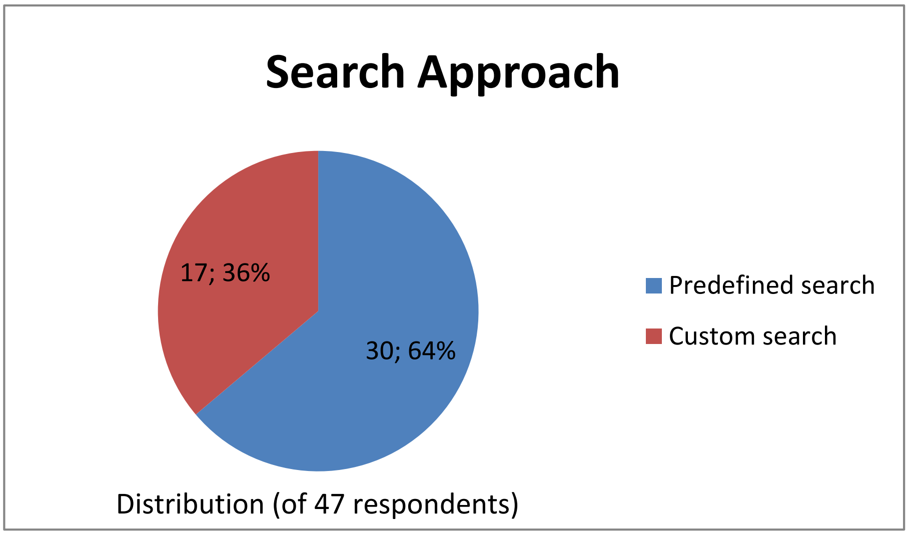


Figure 7.1: Distribution for approaches in Scenario 1

They seemed to think that their search query simply did not have any results, resulting in multiple empty queries. Everyone that used custom search had to ask the other testers to overcome this step. However, when presented the solution of using the predefined terms they seemed to appreciate that approach.

None of the testers had problems after using the predefined terms. Using this approach, the users easily completed the first task and found the additional information about the POI.

The second task was to use information for the particular place they found to find a new place in their current area recommended by users who also liked the current place. This task turned out to be easy for the testers. They swiftly found the new places and appreciated the feature.

Considering response time, only a few users expressed discontent with this matter and those who did pointed out that there should be some kind of indicator that show that the search is in progress. With no indicator, users may be led to believe that the search procedure is not in motion and try to do it again while a search already is performing. This will however only be a case when the search time takes a couple of seconds or more, which did rarely happen during the test phase. However, as the number of users and POIs increases, this becomes more and more important.

Scenario 2
In this scenario, the first task is an extension of the first and involves finding a hotel that is recommended to the tester, which they completed with no help. We knew people would easily be able to search, but due to this task we were able to determine that they also knew how to search for recommended places, which implicitly means that they know that there is a difference between just searching and searching for recommendations.

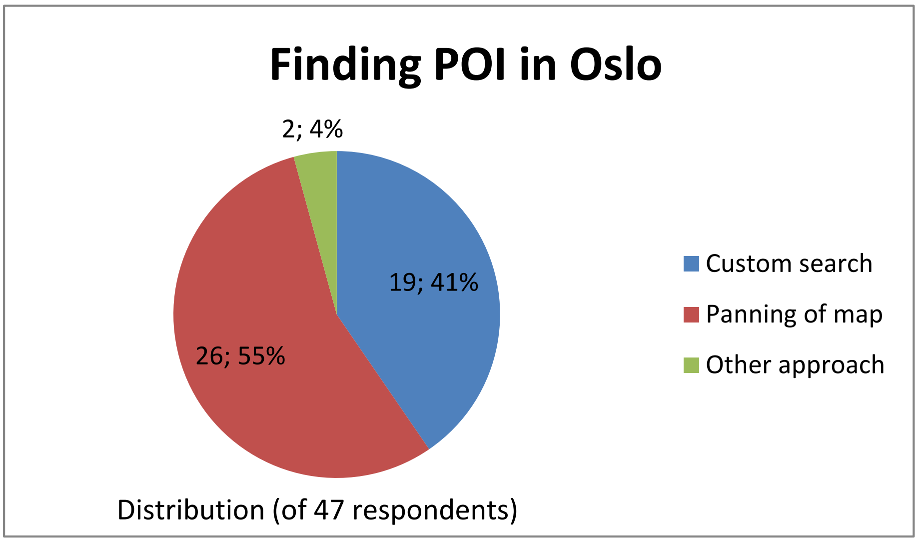


Figure 7.2: Distribution for approaches in Scenario 2

The second part of this scenario prompted the user to find a restaurant in Oslo. We expected this to be an issue an, and as Figure 7.2 states this was the case. Almost half of the testers encountered problems. 41% (19) of them tried to use search with 'Oslo' as a keyword, while 4% (2) accessed the "Travel Plan"-window in search for the solution. Even after the disappointing results from performing custom search in Scenario 1, 41% (19) of them still tried to search with a custom string. Most of these figured out that they had to pan the map after some time, but almost half of these had to get assistance from fellow testers to complete the task. 55% (26) quickly found the solution, but commented that the approach was tedious.

Scenario 3
The testers were first asked to retrieve a travel plan. All testers found this to be easy and quickly obtained a travel plan. As part of the study, we wanted to see if users figured out that they would have to go back to the "Find"-window to add POIs to the travel plan. As Figure 7.3 states, 68% (32) of the users quickly navigated to the "Find"-window and added a POI to the plan. This means that 32% (15) encountered problems and struggled to solve this task. Most of them tried to press "Give me a recommended plan" several times to add a place. However, most users found a way to add a POI after a while.

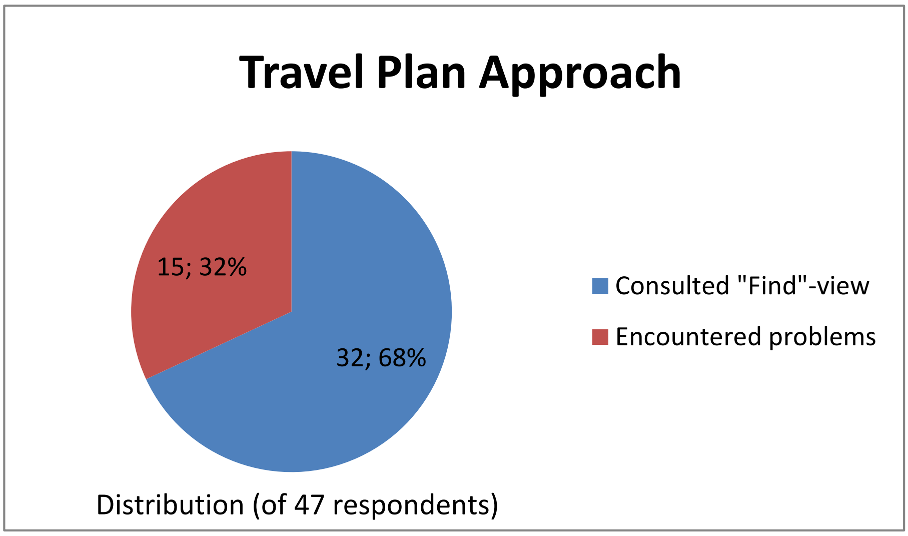


Figure 7.3: Distribution for approaches in Scenario 3

Even though two thirds of the testers were able to handle travel plans, they had several suggestions to improvements of this feature. The most common suggestions were to have a list of the POIs, opportunity to search and that a route through the POIs should be available. In this way users are presented a clear overview of POIs, may search directly from the active view, and are kept informed of what route that is recommended. Seven users were confused whether the displayed places were part of the plan or not. Using a list makes it easier to manage these places, and it will assure users that the displayed POIs are part of the plan. Further, five users expressed that they would like a feature that saves travel plans. At last, one user did not like that the added POIs were discarded when asking for another recommended plan. The function should probably act as before, but should inform the user that the added places will be removed and let the user have an option to abort this operation.

We also encountered a bug in this feature. There are two ways to add a POI to the plan, either by selecting a place and press “Add to plan” or by going into the “More”-view of that POI and then press “Add to plan”. The latter button is not functional.

In General

The response time were perceived as good or better by almost all participants. Only a few expressed some dissatisfaction on the matter, especially when requesting a travel plan. We are happy with this result as the server is running on a desktop at the university using the network provided there. Despite some concerns, the response time never exceeded three seconds which is an acceptable amount of time, but only if an indicator is present to show that the request is being handled.

Users who tried to perform custom string searches, found a bug in the entering process. Upon hitting the input field for search terms, the keyboard covered this field making it impossible to see what was typed without discarding the keyboard.

Two minor issues revolve the dialog presented when searching for POIs. The first issue is that when users tried to hit search, they accidentally hit the “Only stuff recommended for me”-checkbox instead. This is easily fixed by making the text unselective. The second issue is that even though users pressed "Don't tell again", the same dialog kept coming. This should also be easy to fix by simply keeping track of this condition in a Boolean-variable, or similar.

When users search for recommended places and no POIs match this request, the user has little or none benefit of being informed that no matches were found. It was suggested by one the participants that in such a case the system should inform the user that no recommended results were found but the results displayed is POIs that are present in the given area, but not recommended.

Testers were surprised by the sensitivity of the map. In addition, we experienced some problems with map loading. Whether this is an issue with the application, the services provided by Google Maps or the wireless connection is unclear. At some points during the test phase we did observe loss of wireless connection, but we were unable to assess whether this caused the map problems or not. This loss of connection also resulted in some minor interruptions during some sessions.

The route provided to users is not optimal at all times, and the user is not given the choice to specify whether they want directions for driving, public transportation or walking.

As not all testers finished the task at hand at the same time, they were left free to play with the application. One result of this is that some tried to use the application with the landscape-orientation, i.e. rotating the mobile device 90 degrees so that they get a widescreen display. This turned out to cause trouble for the application, even making it stop working. According to [Android, 2011], when the orientation is changed the current activity is restarted, and this might be causing the problems.

Discussion

In general, the feedback provided by the testers was good. The biggest source of dissatisfaction involved the power of the search engine as it rarely gave results to custom strings. The system was perceived to provide valuable information. The general feeling to the concept was good, but they would really appreciate a stronger application with the suggested improvements added.

During the test-phase, four testers told us about other similar services. Among them is Google Places which we introduced in Chapter 2. In the next chapter we will discuss the results found.

7.4.2 Presentation of Quantitative Data

In this section we will describe the results from the user acceptance survey. We will first discuss what representative data can be, and then we will go through each of the six areas.

When analyzing results from a survey using Likert scales, we cannot assume that the respondents mean that the difference between strongly agreeing (coded as 7) and agreeing (coded as 6) is that same as between undecided/neutral (coded as 4) and somewhat disagreeing (coded as 3). As a result of this, many researchers advise against using the *mean* as a measure. In some cases the *median* or *mode* is more appropriate as these are less sensitive to extreme observations. Such a case is illustrated when combining two observations, one given 3 and one given 4. It makes little sense to combine these and get a mean of 3.5. According to [Breakwell, 2006], as researchers in some cases rather consider Likert results to interpreted as interval data, more powerful statistical methods may be used.

When using median or mode, we have to make sure that the used method does not give an incorrect picture. Considering the case where seven responses are 7, one is 2 and one is 4. The median in this example is 4 even though most the respondents gave it a higher value. As mode represents the value that occurs most frequently, it will be 7 in this case which is more representable for this data set than the median. However when using mode there might be several

Response: (1) Strongly Disagree -- (7) Strongly Agree															
Item	(1)		(2)		(3)		(4)		(5)		(6)		(7)		Avg.
PU 1	10.6%	(5)	4.3%	(2)	10.6%	(5)	19.1%	(9)	27.7%	(13)	19.1%	(9)	8.5%	(4)	4.4
PU 2	0.0%	(0)	0.0%	(0)	0.0%	(0)	12.8%	(6)	12.8%	(6)	42.6%	(20)	31.9%	(15)	5.9
PU 3	0.0%	(0)	2.1%	(1)	10.6%	(5)	21.3%	(10)	31.9%	(15)	23.4%	(11)	10.6%	(5)	5.0
PU 4	6.4%	(3)	14.9%	(7)	17.0%	(8)	29.8%	(14)	10.6%	(5)	17.0%	(8)	4.3%	(2)	3.9
PU 5	0.0%	(0)	4.3%	(2)	2.1%	(1)	4.3%	(2)	25.5%	(12)	36.2%	(17)	27.7%	(13)	5.7
Total 1	2.2%	(1)	4.3%	(2)	4.3%	(2)	8.7%	(4)	21.7%	(10)	43.5%	(20)	15.2%	(7)	5.3
Total 2	3.4%	(9)	5.1%	(14)	8.1%	(21)	17.4%	(45)	21.7%	(61)	27.7%	(85)	16.6%	(46)	5.0

Table 7.1: Perceived Usefulness: Frequency results

values that have the same frequency, causing ambiguity in the result set.

Despite these arguments, we believe that the mean still provides us with useful information in terms of indicating where the majority of responses are. Further, we will calculate the average response to a particular construct based on the mode calculated from each participant's set of responses (represented by 'Total 1'), summarized mean (represented as 'Total 2'), and the mean for each construct (represented in the last column as 'Avg.'). The tables are labeled as in the survey (see Appendix 8.5). Both the percentage and number of responses are represented.

Perceived Usefulness

The results from the survey concerning Perceived Usefulness is presented in Table 7.1. PU2 received the highest average score of 5.9. This item represents the fundamental operation of finding restaurants, hotels and other places. None of the testers disagreed with this item, while 74.5% (35) either agreed or strongly agreed. The two items, with the lowest scores, are PU1 and PU4 with average score of 4.4 and 3.9, respectively. This is disappointing, but may be explained by the unnatural relation to the scenario. Testers may have considered their current situation (mostly students), and how these items would help them in their everyday living. Students are mostly not exploring the city looking for places to eat or visit, making a tourist system less susceptive to increase the efficiency of their daily work or better allow them to schedule their time. Despite these two lower values, the average score is 5.0. Most respondents agree that the system would be useful to them as tourists, which is the intended use of the system (PU5). 89.4% (42) somewhat agrees (coded as 5) or more on this item, with 63.8% (30) either agreeing or strongly agreeing.

Perceived Ease of Use

The average scores for Perceived Ease of Use, along with other results, is presented in Table 7.2 with values between 4 and 5 except EOU1. Most of the responses are neutral, somewhat agreeing or agreeing. Through EOU1, 78.7% (37) of the respondents show that they somewhat agrees or more with that it would be easy to learn.

Trust

One of the important factors for users when considering using a mobile service is trust. To assess this, respondents were presented seven items.

Respondents' attitude towards TU4, the importance of that the system provider is widely acknowledged, received the lowest mean value of 4.5. This is not that important to the average user, however we see that still 31.9% (15) find this important (coded as 6) or very important (coded as 7).

Response: (1) Strongly Disagree -- (7) Strongly Agree															
Item	(1)		(2)		(3)		(4)		(5)		(6)		(7)		Avg.
EOU 1	2.1%	(1)	2.1%	(1)	4.3%	(2)	12.8%	(6)	38.3%	(18)	23.4%	(11)	17.0%	(8)	5.2
EOU 2	4.3%	(2)	0.0%	(0)	14.9%	(7)	27.7%	(13)	25.5%	(12)	21.3%	(10)	6.4%	(3)	4.6
EOU 3	4.3%	(2)	10.6%	(5)	19.1%	(9)	23.4%	(11)	25.5%	(12)	14.9%	(7)	2.1%	(1)	4.1
EOU 4	6.4%	(3)	6.4%	(3)	23.4%	(11)	31.9%	(15)	17.0%	(8)	8.5%	(4)	6.4%	(3)	4.0
EOU 5	4.3%	(2)	8.5%	(4)	21.3%	(10)	14.9%	(7)	25.5%	(12)	21.3%	(10)	4.3%	(2)	4.3
Total 1	6.7%	(3)	4.4%	(2)	11.1%	(5)	17.8%	(8)	35.6%	(16)	22.2%	(10)	2.2%	(1)	4.4
Total 2	4.3%	(13)	5.5%	(15)	16.6%	(44)	22.1%	(60)	26.4%	(78)	17.9%	(52)	7.2%	(18)	4.4

Table 7.2: Perceived Ease of Use: Frequency results

Response: (1) Strongly Disagree -- (7) Strongly Agree															
Item	(1)		(2)		(3)		(4)		(5)		(6)		(7)		Avg.
TU1	0.0%	(0)	4.3%	(2)	2.1%	(1)	8.5%	(4)	40.4%	(19)	27.7%	(13)	17.0%	(8)	5.4
TU2	8.5%	(4)	8.5%	(4)	8.5%	(4)	12.8%	(6)	29.8%	(14)	21.3%	(10)	10.6%	(5)	4.5
TU3	2.1%	(1)	0.0%	(0)	8.5%	(4)	14.9%	(7)	17.0%	(8)	34.0%	(16)	23.4%	(11)	5.4
TU4	2.1%	(1)	2.1%	(1)	8.5%	(4)	14.9%	(7)	14.9%	(7)	40.4%	(19)	17.0%	(8)	5.3
TU5	0.0%	(0)	4.3%	(2)	2.1%	(1)	10.6%	(5)	23.4%	(11)	29.8%	(14)	29.8%	(14)	5.6
TU6	0.0%	(0)	0.0%	(0)	6.4%	(3)	12.8%	(6)	19.1%	(9)	31.9%	(15)	29.8%	(14)	5.7
TU7	2.1%	(1)	0.0%	(0)	6.4%	(3)	6.4%	(3)	17.0%	(8)	34.0%	(16)	34.0%	(16)	5.7
Total 1	2.1%	(1)	0.0%	(0)	6.4%	(3)	10.6%	(5)	19.1%	(9)	38.3%	(18)	23.4%	(11)	5.7
Total 2	2.1%	(3)	2.7%	(3)	6.1%	(15)	11.6%	(28)	23.1%	(43)	31.3%	(80)	23.1%	(63)	5.4

Table 7.3: Trust: Frequency results

82.9% (39) of the respondents state that the reliability of the data returned by the system is important. The most important items are TU6 and TU7 which concerns the safety of using the system.

Rather surprising, only 74.5% (35) is concerned about privacy. 14.9% (7) stands neutral on this manner, 8.5% (4) finds it somewhat not important (coded as 3) and 2.1% (1) do not find this important at all. This may be explained by that the application does not handle very sensitive information, such as financial information.

Personal Initiatives and Characteristics

The responses to the items pertaining to Personal Initiatives and Characteristics is presented in Table 7.4. One of the important items in this construct, is PIC1: "I am capable of using the system". For this item, none of the testers disagreed (to any degree) while 91.5% (43) considered themselves capable of using the system (score 5-7). Respondents did not agree on PIC2, whether the system was fun to use or not. We have responses on all values, but we see that the majority of responses signals that they had fun.

Considering PIC5, we see that 82.9% (39) would only use the system if it is available for free. One can argue that the system is useful, but not that important or useful that it is worth paying. The result might have been different if not most of the testers were students who tend to have smaller budgets than the average worker.

Combining PIC1 and PIC7 gives a good indication of usability and usefulness of the system. Even though PIC7 presents slightly lower scores than PIC1, 82.9% (39) gave a score between 5

Response: (1) Strongly Disagree -- (7) Strongly Agree								
Item	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Avg.
PIC1	0.0% (0)	0.0% (0)	0.0% (0)	8.5% (4)	23.4% (11)	34.0% (16)	34.0% (16)	5.9
PIC2	2.1% (1)	2.1% (1)	19.1% (9)	21.3% (10)	23.4% (11)	23.4% (11)	8.5% (4)	4.7
PIC3	6.4% (3)	8.5% (4)	17.0% (8)	36.2% (17)	10.6% (5)	10.6% (5)	10.6% (5)	4.1
PIC4	2.1% (1)	4.3% (2)	12.8% (6)	17.0% (8)	38.3% (18)	21.3% (10)	4.3% (2)	4.7
PIC5	0.0% (0)	4.3% (2)	0.0% (0)	12.8% (6)	17.0% (8)	17.0% (8)	48.9% (23)	5.9
PIC6	0.0% (0)	4.3% (2)	6.4% (3)	31.9% (15)	40.4% (19)	14.9% (7)	2.1% (1)	4.6
PIC7	2.1% (1)	2.1% (1)	0.0% (0)	12.8% (6)	34.0% (16)	31.9% (15)	17.0% (8)	5.4
Total 1	2.1% (1)	2.1% (1)	4.3% (2)	25.5% (12)	27.7% (13)	21.3% (10)	17.0% (8)	5.2
Total 2	1.8% (7)	3.6% (13)	7.9% (28)	20.1% (78)	26.7% (101)	21.9% (82)	17.9% (67)	5.0

Table 7.4: Personal Initiatives and Characteristics: Frequency results

Response: (1) Strongly Disagree -- (7) Strongly Agree								
Item	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Avg.
CT1	2.1% (1)	0.0% (0)	0.0% (0)	6.4% (3)	10.6% (5)	29.8% (14)	51.1% (24)	6.2
CT2	6.4% (3)	4.3% (2)	8.5% (4)	27.7% (13)	27.7% (13)	17.0% (8)	8.5% (4)	4.5
CT3	0.0% (0)	6.4% (3)	2.1% (1)	12.8% (6)	31.9% (15)	27.7% (13)	19.1% (9)	5.3
CT4	0.0% (0)	2.1% (1)	0.0% (0)	14.9% (7)	21.3% (10)	36.2% (17)	25.5% (12)	5.7
CT5	0.0% (0)	0.0% (0)	2.1% (1)	6.4% (3)	10.6% (5)	34.0% (16)	46.8% (22)	6.2
CT6	0.0% (0)	0.0% (0)	0.0% (0)	4.3% (2)	12.8% (6)	53.2% (25)	29.8% (14)	6.1
CT7	2.1% (1)	2.1% (1)	0.0% (0)	10.6% (5)	8.5% (4)	21.3% (10)	55.3% (26)	6.1
Total 1	2.1% (1)	0.0% (0)	0.0% (0)	6.4% (3)	12.8% (6)	38.3% (18)	40.4% (19)	6.2
Total 2	1.5% (6)	2.1% (7)	1.8% (6)	11.9% (42)	17.6% (64)	31.3% (121)	33.7% (130)	5.7

Table 7.5: Context: Frequency results

and 7. The respondents did not agree upon the importance of being the first to use the system or not. 36.2% (17) remained undecided, while the rest were equally split on both sides of undecided.

Context

We will now look at the results from the Context-construct which are depicted in Table 7.5. As stated when presenting results for Perceived Usefulness, PU1 might have been graded low as the testers might have assigned the value considering themselves in their current situation and not as a tourist. We are however presented with how many could use the system if they were on vacation (CT1). 91.5% (43) could use the application on vacation (scores 5-7), where 51.1% (24) stated that they strongly agreed (coded as 7) resulting in an average score of 6.2. This is very promising, and indicates that the system could be useful when they stepped in the role as tourists.

The easiness to obtain and install (CT5) proves to be important to users. 80.8% (38) either agrees or strongly agrees with this statement, where almost half of the respondents states that they strongly agree (46.8%).

The prior experience using mobile services has an average score of 5.3 (CT3), but 78.7% (37) scores the item between 5 and 7. The values indicate that this is important to most users.

We observe that respondents disagree whether it is important that most people around them are using the system. This seems to have a lower effect on their decision. However, it is more

Response: (1) Strongly Disagree -- (7) Strongly Agree														
Item	(1)		(2)		(3)		(4)		(5)		(6)		(7)		Avg.
IU1	2.1%	(1)	6.4%	(3)	2.1%	(1)	27.7%	(13)	21.3%	(10)	29.8%	(14)	10.6%	(5)	4.9
IU2	0.0%	(0)	8.5%	(4)	6.4%	(3)	12.8%	(6)	42.6%	(20)	17.0%	(8)	12.8%	(6)	4.9
Total 2	1.1%	(1)	7.4%	(7)	4.3%	(4)	20.2%	(19)	31.9%	(30)	23.4%	(22)	11.7%	(11)	4.9
Total 3	2.1%	(1)	8.5%	(4)	6.4%	(3)	25.5%	(12)	31.9%	(15)	17.0%	(8)	8.5%	(4)	4.7

Table 7.6: Intention to Use: Frequency results

important to them that the official tourist information office encourages tourists to use it. Only 2.1% (1) disagrees with this statement, while 14.9% (7) remains undecided and the rest of the users (82.9%) agree at different extents.

Intention to Use

The last construct in the user acceptance survey is Intention to Use. The results from this construct is presented in Table 7.6. As it only coincides two items, we neglected the modular scores and added a new row, denoted by 'Total 3', that takes the minimum of the two values from each respondent and computes the frequency distribution.

The first item, IU1, indicates the respondent's intention to use the system assuming that he/she has access to it. For scores of 5 to 7, we find 61.7% (29) of the respondents. Only 10.6% (5) disagrees while 27.7% (13) remains undecided. The second item, IU2, is the respondent's prediction on intention to use the system given that he/she has access to it. For this item, we have a higher concentration of respondents giving the score 5. In general, we have slightly higher responses for the second item. 72.3% (34) predicts that they would use the system (score 5-7) given that they have access to it.

'Total 3' was computed to get the most conservative estimate of user adoption. Combining the scores 5, 6 and 7, we find that 57.4% (27) intends to use the system. 17.0% (8) does not intend to use it, while 25.5% (12) remains undecided.

7.4.3 Validity of Results

The users were prompted to answer smaller questions, such as the name of a hotel, during each scenario as well as to comment on what worked, what did not work, response time, and illogical/difficult parts after each scenario. This provided us with valuable feedback.

Usability testing on mobile devices introduces some additional difficulties to the "general" test phase. As already mentioned, small screens make it harder to observe what the user is doing. Several tactics have been tried to solve this problem. [Kaikkonen et al., 2005] used an external camera to capture the mobile screen, but this is challenging in itself due to the small size of the screen and the user occludes the screen most of the time [Balagtas-Fernandez and Hussmann, 2009]. Another approach is to log events [Paternò et al., 2007]. This approach provides accurate information but requires preparation of the system (must specify log-elements throughout the system) and the extensive logs have to be analyzed.

We decided to not use cameras as we felt that such an approach was not necessary and in addition they might affect the users' opinion of the system. The test phase already includes measures to obtain the data necessary (think-out-load protocol and comment-fields) so logging was neither necessary, but as the system had some logging-like functionality we decided to use it during testing. More or less everything the user does generates a request to the server. Requests to the server is displayed on the server which the observer has readily available during the tests.

In this way, the observer can keep track of what the testers do by observing them and paying attention to the server output.

A sample size of 20 testers is sufficient to find 95% of all usability problems according to [Faulkner, 2003]. Our test group of 47 users is well within this boundary. As the percentage approaches 100%, the number of users needed to increase it by one percent is higher than when increasing one percent with 50% as a basis. [Faulkner, 2003] further states that a test group size of 40 testers will reveal 98% of all usability problems. This means that even though 47 users are over twice as many needed to find 95% of all usability problems, it will only increase by three percent. In addition, [Galtung, 1969] states that 40 is the needed sample size to do quantitative statistical analysis, which we also exceed.

The test results do probably not represent the view of the average man as 46 of the 47 testers were students of higher education, specifically 5 bachelor students, 35 master students and 6 doctoral students. Also, 41 of them are students in a science or engineering department making them unrepresentative of the general population.

As a second note, is that they only used the system for a short amount of time (i.e., about one hour) and that the location used was simulated. It would be interesting to see what users thought of the system after using it over an extended period of time as they would have gained more knowledge about the features available and the real life usefulness. Even though we had limited time, the scenarios they went through included most of the features in the system. The left out features are the pro-active recommendations and pending ratings. The reason we decided to simulate the location was that it would be impossible to follow several testers at a time, especially as they might have selected different POIs. However, we do not believe that this approach had any significant impact on the results as the testers were informed about the matter. Even though, the absolute real need and usefulness of this system is something one would have to use “on the fly” to experience.

Likert scales are subject to three well-known biases: 1) acquiescence response bias: agreeing with statements as presented, 2) central tendency bias: avoiding the use of extreme response categories, and 3) social desirability bias: trying to portray themselves in a more favorable light. The first mentioned problem can be eliminated by designing a scale with balanced keying, meaning an equal number of positive and negative statements. Social desirability bias can be that they might want to please the facilitators by agreeing with the statements, or wanting to give the impression of being capable of using the system. It is difficult to assess central tendency bias, but we did not discover tendencies similar to/any prove of this bias.

7.4.4 Summary

The evaluation successfully identified problems and issues with the system. Participants in the test provided us with valuable input and insight to how they perceived the system. In addition, they presented suggestions to improvements.

The response regarding intention of use could have been better, but as stated before, this might be due to the missing role of actually being a tourist. This statement is supported by the findings of CT1 which regards if the users could use the system as a tourist where 91.7% stated that they could (51.1% strongly agreed), and that 89.4% states that it would be useful to them as tourists (PU5). Even though scenarios stated that they were tourists, they might have disregarded this role when answering the survey. As for further testing, this should be clarified right before handing out the user acceptance test.

The low intention to use cannot be solely contributed to this assumption, and that is neither our opinion. One cannot expect that all users will use a system even if it is bug-free. But we believe that the real value of intention to use is higher than in the results. Other factors that contributed to this low score is probably the custom search engine that rarely gave results, and the little overview and functions of travel plans.

Chapter 8

Discussion and Conclusion

In this chapter we will discuss the results from the usability test, answer the research questions, conclude our study, state our contributions, and present our suggestions for further work.

8.1 Discussion of Results

In order to evaluate the system, we performed usability testing with 47 users. These participants provided us with usability data through comments and a user acceptance survey. Presentation of these results can be found in Section 7.4.

The overall response to the system was positive, and participants liked the idea. However, during test sessions we encountered some bugs and minor defects which affected the participants' opinion. Several testers stated that they liked the idea, but that the bugs found has to be removed to increase user satisfaction. As an example, when losing Internet-connection and requesting information, the application stopped working and did not notify the user of the cause, leaving the user uninformed about what happened.

Through analyzes of the test results, we see that 57.4% of the respondents intend to use the service. We believe that making the system more robust and user friendly through simple improvements, the intention to use would increase and be closer to the 100%-mark. This is especially based on the fact that 91.5% of the participants stated that they could use the system when on vacation, and that 89.4% states that it would be useful to them as tourists (PU5). However, we emphasize that we do not expect that 91.5% is going to use the service, but that the low intention to use percentage would probably be higher if they considered themselves as tourists. Another contributing factor to the low intention to use, can be contributed to the custom search engine that rarely gave results. In general, the responses were positive but we see, as mentioned earlier, that the response could be even better.

An interesting observation is the use of Android's menu button. Not all Android devices possess such a button, but several do. During the test sessions, only three participants tried to use this button. It did not provide any functionality at the moment, but it provides an opportunity to add more functionality without complicating the user interface.

Despite some problems/bugs found during the test-phase, we believe that this system has potential to be used by many. However, several other tourist applications arise from time to time, making it harder to be the application that stands out. It becomes even more difficult when all mobile devices with Android 1.6 or higher are shipped with a tourist application, namely Google Places, and Android's share of users have been on a steady rise since its introduction.

8.2 Research Questions

We will now look at the answers to the research questions defined in Section 3.1.

Research Question 1: Can the prototype of the personalized mobile tourist application, MTSR, enhance the experience of tourists and others when searching for a place to eat, sleep or experience something?
As stated in the “Discussion of Results”-section, most users intended to use the service and over 90% stated that they could use it when on vacation, and 89.4% stated that it would be useful to them as tourists (PU5). This indicates that the service provides information that the users find valuable, especially when they are tourists. Thus, we can say that the service enhances the experience of tourists. Some improvements are suggested and should be implemented as it will increase the user experience.

Research Question 2: How user friendly is the current design/user interface?
As there are more parts of the system, we will evaluate this question in terms of each of them. The main part of the system is the “Find”-window where users may search for POIs. Users found this interface to be a very easy to use, and that it provided a good overview of the results. However, participants were not that fond of the “Travel Plan”-view. Some users were not sure whether the presented POIs were part of the plan, 32% struggled when asked to add a POI to the plan, and they did not know which order they should visit the POIs. This last view clearly needs some improving.

Research Question 3: What are people’s attitudes toward adapting this system?
We have already discussed the answer to this question several times: we have seen that 57.4% intend to use the system, while 91.5% stated that they could use the system when they are on vacation and 89.4% stated that it would be useful to them as tourists. We argued that the low percentage for intention to use may be caused by participants considering themselves as students or residences instead of tourists. It is important to remark that we do not expect 91.5% to use the system when they are on vacation, but that we reason that the low intention to use percentage would probably be higher if they considered themselves as tourists. Users liked the idea, but called for improvements such as better custom search, more organized travel plan and indicators.

8.3 Conclusions

In this study, we added functionality, fixed bugs and performed a large scale usability test. Through different channels, we were able to get 47 testers whom mostly were students. The participants were divided into groups of maximum four, and asked to accomplish tasks given through three scenarios. As a final step of each test session, participants were asked to fill out a user acceptance survey that was a revised edition of the original survey proposed by MSAM. The user acceptance model applied to this study is also MSAM.

Through analyzes of our results from the usability test, we found that there is a positive general attitude towards the system. Disappointingly, the intention to use score was lower than desirable, but the results indicate that this might be due to the lacking role of actually being a tourist or pretending to be one. We are not able to determine how much this potentially affected the outcome, but believe that it had some degree of effect on it. Again, we would like to emphasize that the low intention to use is of other factors as well, such as the custom search engine.

We have also discovered the participants' feelings toward topics like how important the reliability of received data is. For this matter, 82.9% stated that the reliability of the data returned by the system is important.

The overall results from the study suggest that the service has potential, but also call for improvements. Users need to be advised with what is happening, especially when things do not go as planned, and the user interface could need some refinements. The suggested improvements do not require an extreme amount of work, and will significantly improve the service.

This study lays the foundation for future iterations after performing a usability test and analyzes of the results. We see potential in this application and are curious of what the results of a usability test, after the improvements have been implemented, will reveal.

8.4 Contributions

During this project, we have further developed MTSR through various fixes and additions. The most comprehensive addition is the three step process that allows easy insertion of new points of interest. In addition, we have contributed with data to further validate MSAM. The paper, whose author is Shang Gao and co-author is ourselves, is unfortunately not finished at this moment.

Our final contribution is the identification of possible further improvements which are presented in the next section.

8.5 Further Work

In this section, we will present some suggestions for further improvements that have been identified through the usability testing process.

- Indicators when something is processing. This includes when searching for POIs, acquiring travel plans and loading maps.
- Inform the user when something is not functioning as it should (e.g., when the client is unable to reach the server).
- When the map is panned or zoomed, the last search could automatically be executed again and new POIs should be added to the map. [FINN, 2011] has already implemented this as part of its search for residences, and can be tested at their website. This will increase the load on the server, as well as the bandwidth used and thereby probably be more expensive for the users, so this should be well considered and left as an option, that users may turn on and off, if implemented.
- As stated in Section 7.4, the map is very sensitive, in terms of zooming very fast when using pinch zoom (zooming by performing a pinch-move with two fingers on the screen), and it may be a good idea to lower this sensitivity.
- An interface for modifying existing data. The current solution can only add points of interest, and it is therefore a need for more functionality where it is possible to modify the data already in the system.
- A little, round button with "i" could be added in one of the top corners of each window. Pressing this would provide the users with relevant information about the current window and how to use it.

- The inactive recommender system using Bayesian reasoning should be activated and evaluated. This will pose some challenges as it requires data in a certain manner, making it harder to get sample data. However, it will be interesting to see how well this recommender system performs.
- Perform another iteration of implementation and evaluation. Usability testing may done in usability labs or in the field using Wireless Trondheim[Trondheim, 2011]. It would be favorable if the participants are from different cultures, so that it may be investigated whether culture differences affect the result.
- Finish and try to publish the MSAM validation paper that was mentioned earlier.

Appendix A

Demographical Data of Participants

	Amount	Percent
Gender		
Female	11	23.4%
Male	36	76.6%
Age		
Less than 20 years	2	4.3%
20-30 years	43	91.5%
Above 30 years	2	4.3%
Department		
Science or Engineering	41	87.2%
Other	6	12.8%
Education level		
Bachelor student	5	10.6%
Master student	35	74.5%
Doctoral Student	6	12.8%
Other	1	2.1%
Experience		
0-1 year	15	31.9%
2-5 years	17	36.2%
More than 5 years	15	31.9%
Nationality		
Norwegian	35	74.5%
Chinese	10	21.3%
Swedish	1	2.1%
Russian	1	2.1%

Appendix B

Scenarios

Usability Scenarios

Scenario One: Find a Restaurant and Its Information

You are a tourist in Trondheim looking for a restaurant

Task 1: Find a restaurant

- Open the search dialog and search for a restaurant
- Choose one of the closest places and write down the name and address
    - Name:
    - Address:
- Use information for this particular place to find a new place in your current area recommended by users who also liked the current place
    - Name (of one of them):

Comments: What worked? What did not work? What do think of the response time? Something illogical/difficult? Any other comments?

Scenario Two: Search for Information

You are a tourist in Trondheim and want to find a hotel for the night. Next week you are traveling to Oslo as well, and decide to find a restaurant there.

Task 1: Find places

- Find a hotel that is recommended for you by using the search feature and write down the name of one of them:
    - Name:

Task 2: Find a place far from your location

- Find a restaurant in Oslo and write down the name of it
  - Name:

Comments: What worked? What did not work? What do think of the response time? Something illogical/difficult? Any other comments?

Scenario Three: Using Travel Plan

You are a tourist in Trondheim and want to acquire a complete travel from the MTSR-system.

Task 1: Get travel plan and information

- Request a travel plan
- Choose one of the places and write down the name of it
  - Name:

Task 2: Add a place to the travel plan

- Add a place to the travel plan

Comments: What worked? What did not work? What do think of the response time? Something illogical/difficult? Any other comments?

Appendix C

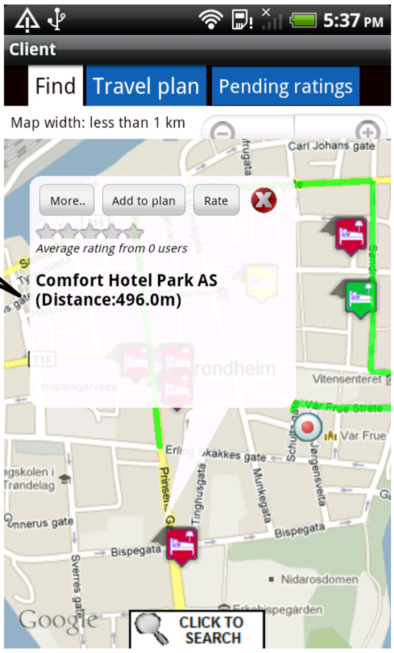
User Guide

User Guide

User Guide for Scenario One and Two

The following screenshots with brief descriptions of each provides an introduction to the application and its usage.

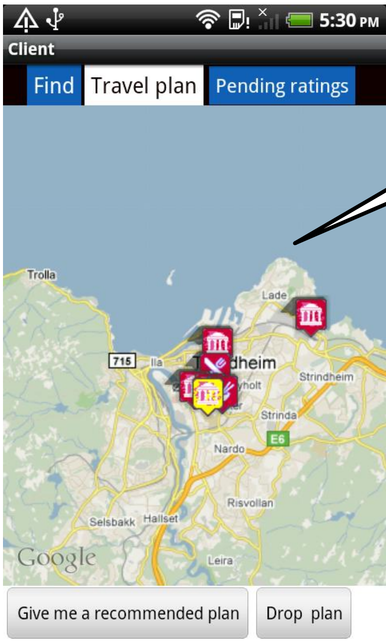
By clicking a place, the system will display this box



By pressing "More..", the system will provide more information if available

User Guide for Scenario Three

Travel plan view



View with recommended plan

Appendix D

User Acceptance Survey Prior Revision

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.	❑	❑	❑	❑	❑	❑	❑
PU2. The system would allow me to find rooms and buildings at NTNU.	❑	❑	❑	❑	❑	❑	❑
PU3. The system would make it easier to keep track of my weekly tasks.	❑	❑	❑	❑	❑	❑	❑
PU4. The system would allow me to better schedule my time.	❑	❑	❑	❑	❑	❑	❑
PU5. The system would be useful for me as a student.	❑	❑	❑	❑	❑	❑	❑

Perceived Ease of Use	**Strongly disagree**						**Strongly agree**
PEU1. Learning to operate the system would easy for me.	❑	❑	❑	❑	❑	❑	❑
PEU2. I would easily find the information I am looking for using the system.	❑	❑	❑	❑	❑	❑	❑
PEU3. I would find the user interface of the system clear and intuitive.	❑	❑	❑	❑	❑	❑	❑
PEU4. I would find the system to be flexible to interact with.	❑	❑	❑	❑	❑	❑	❑
PEU5. I would find the system to easy to use (user-friendly).	❑	❑	❑	❑	❑	❑	❑

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

Personal Initiatives and Characteristics	**Strongly disagree**						**Strongly agree**
PIC1. I am capable of using the system.	❑	❑	❑	❑	❑	❑	❑
PIC2. I have fun using the system.	❑	❑	❑	❑	❑	❑	❑
PIC3. I prefer to be the first one using the system.	❑	❑	❑	❑	❑	❑	❑
PIC4. Using the system gives me an advantage over those who don't.	❑	❑	❑	❑	❑	❑	❑
PIC5. I would only use the system if it was available for free.	❑	❑	❑	❑	❑	❑	❑
PIC6. I find it rewarding to use the system.	❑	❑	❑	❑	❑	❑	❑
PIC7. Using the system is a good idea.	❑	❑	❑	❑	❑	❑	❑

Context

I could use the system…

	Strongly disagree						Strongly agree
CT1. if I am being out of home or the office.	❑	❑	❑	❑	❑	❑	❑
CT2. if most people around me are using the system.	❑	❑	❑	❑	❑	❑	❑
CT3. if I had nice experience in using mobile services before.	❑	❑	❑	❑	❑	❑	❑
CT4. if the University encourage students to use the system.	❑	❑	❑	❑	❑	❑	❑
CT5. if the system was easy to obtain and install.	❑	❑	❑	❑	❑	❑	❑
CT6. if it is meaningful/relevant to my daily tasks.	❑	❑	❑	❑	❑	❑	❑
CT7. if I did not have access to a desktop computer or laptop.	❑	❑	❑	❑	❑	❑	❑

Intention to Use

	Strongly disagree						Strongly agree
IU1. Assuming I have access to the system, I intend to use it.	❑	❑	❑	❑	❑	❑	❑
IU2. Given that I have access to the system, I predict that I would use it.	❑	❑	❑	❑	❑	❑	❑

Appendix E

User Acceptance Survey After Revision

User Acceptance Survey

Part 1: Personal Information

1. Gender

☐ Female ☐ Male

2. Age

☐ Less than 20 years ☐ 20-30 years ☐ Above 30 years

3. Department

☐ Science or Engineering Department ☐ Other Department. Please specify:

4. Education Level

☐ Bachelor Student ☐ Master Student ☐ Doctoral Student

☐ Other. Please specify:

5. Experience of Mobile Services

☐ 0-1 year ☐ 2-5 years ☐ More than 5 years

6. Nationality

Nationality: ________________________

Part 2: Post-Experiment Questionnaire

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

Perceived Usefulness (PU)

	Strongly disagree						Strongly agree
PU 1. Using the system would increase the efficiency of my time.	☐	☐	☐	☐	☐	☐	☐
PU 2. The system would allow me to find restaurants, hotels and other places in Trondheim.	☐	☐	☐	☐	☐	☐	☐
PU 3. The system would make it easier to keep track of places to visit.	☐	☐	☐	☐	☐	☐	☐
PU 4. The system would allow me to better schedule my time.	☐	☐	☐	☐	☐	☐	☐
PU 5. The system would be useful to me as a tourist.	☐	☐	☐	☐	☐	☐	☐

Perceived Ease of use (EOU)

	Strongly disagree						Strongly agree
EOU 1. Learning to operate the system would be easy for me.	☐	☐	☐	☐	☐	☐	☐
EOU 2. I would easily find the information I am looking for using the system.	☐	☐	☐	☐	☐	☐	☐
EOU 3. I would find the user interface of the system clear and intuitive.	☐	☐	☐	☐	☐	☐	☐
EOU 4. I would find the system to be flexible to interact with.	☐	☐	☐	☐	☐	☐	☐
EOU 5. I would find the system to be easy to use (user-friendly).	☐	☐	☐	☐	☐	☐	☐

Trust (TU)

I could use the system …

	Strongly disagree						Strongly agree
TU 1. if I have a clear conception of the functionality of the system.	☐	☐	☐	☐	☐	☐	☐
TU 2. if the system provider is widely acknowledged (e.g. the university).	☐	☐	☐	☐	☐	☐	☐
TU 3. if the system protects the privacy of its user.	☐	☐	☐	☐	☐	☐	☐
TU 4. if I feel confident that I can keep the system under control.	☐	☐	☐	☐	☐	☐	☐
TU 5. if I feel confident that the data returned be the system is reliable.	☐	☐	☐	☐	☐	☐	☐
TU 6. if I believe it is risk-free to use the system.	☐	☐	☐	☐	☐	☐	☐
TU 7. if it is safe to use the system.	☐	☐	☐	☐	☐	☐	☐

Personal Initiatives and Characteristics (PIC)

	Strongly disagree						Strongly agree
PIC 1. I am capable of using the system.	☐	☐	☐	☐	☐	☐	☐
PIC 2. I have fun using the system.	☐	☐	☐	☐	☐	☐	☐
PIC 3. I prefer to be the first one using the system.	☐	☐	☐	☐	☐	☐	☐
PIC 4. Using the system gives me an advantage over those who don't.	☐	☐	☐	☐	☐	☐	☐
PIC 5. I would only use the system if it was available for free.	☐	☐	☐	☐	☐	☐	☐
PIC 6. I find it rewarding to use the system.	☐	☐	☐	☐	☐	☐	☐
PIC 7. Using the system is a good idea.	☐	☐	☐	☐	☐	☐	☐

Context (CT)

I could use the system …	Strongly disagree						Strongly agree
CT 1. if I am on vacation	☐	☐	☐	☐	☐	☐	☐
CT 2. if most people around me are using the system.	☐	☐	☐	☐	☐	☐	☐
CT 3. if I had nice experience using mobile services before.	☐	☐	☐	☐	☐	☐	☐
CT 4. if Visit Trondheim (official tourist information office) encourage tourists to use the system.	☐	☐	☐	☐	☐	☐	☐
CT 5. if the system was easy to obtain and install.	☐	☐	☐	☐	☐	☐	☐
CT 6. if it is meaningful/relevant to daily tasks as a tourist.	☐	☐	☐	☐	☐	☐	☐
CT 7. if I did not have access to a desktop computer or laptop.	☐	☐	☐	☐	☐	☐	☐

Intention to Use (IU)

	Strongly disagree						Strongly agree
UI 1. Assuming I have access to the system, I intend to use it.	☐	☐	☐	☐	☐	☐	☐
UI 2. Given that I have access to the system, I predict that I would use it.	☐	☐	☐	☐	☐	☐	☐

Please comment on the scales above:

Bibliography

Android (Accessed on June 14, 2011). Android developers. `http://developer.android.com`.

Asif, M. (2010). Student centric mobile services. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU).

Averjanova, O., Ricci, F., and Nguyen, Q. N. (2008). Map-based interaction with a conversational mobile recommender system. In *Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, UBICOMM '08, pages 212–218, Washington, DC, USA. IEEE Computer Society.

Bagozzi, R. P., Davis, F. D., and Warshaw, P. R. (1992). Development and test of a theory of technological learning and usage. *Human Relations*, 45(7):659–686.

Balagtas-Fernandez, F. and Hussmann, H. (2009). A methodology and framework to simplify usability analysis of mobile applications. *Automated Software Engineering, International Conference on*, 0:520–524.

Bolchini, C., Schreiber, F. A., and Tanca, L. (2007). A methodology for a very small data base design. *Inf. Syst.*, 32:61–82.

Breakwell, G. (2006). *Research methods in psychology*. Psychology Series. Sage Publications.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370.

Byun, H. (July 2004). Utilizing context history to provide dynamic adaptations. *Applied Artificial Intelligence*, 18:533–548(16).

Canalys (Accessed on June 16, 2011). Google's Android becomes the world's leading smart phone platform. `http://www.canalys.com/pr/2011/r2011013.html`.

Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. (2000). Developing a context-aware electronic tourist guide: some issues and experiences. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, pages 17–24, New York, NY, USA. ACM.

Ciurana, E. (Accessed on June 13, 2011). Migration to java 5 at walmart.com. `http://www.theserverside.com/news/1364419/Migration-to-Java-5-at-walmartcom`.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):pp. 319–340.

Dey, A. K. (2001). Understanding and using context. *Personal Ubiquitous Comput.*, 5:4–7.

El-Rabbany, A. (2006). *Introduction to GPS: The Global Positioning System*. Artech House.

Ericsson, K. A. and Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, 87(3):215–251.

Ericsson, K. A. and Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. Cambridge, MA, US: The MIT Press.

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, and Computers*, 35(3):379–383. cited By (since 1996) 48.

FINN (Accessed on June 15, 2011). Kart over bolig til salgs på FINN kart. `http://kart.finn.no/?mapType=norge&tab=soek_i_annonser&searchKey=search_id_realestate_homes&mapTitle=Kart+over+bolig+til+salgs+&ztr=1&areaId=20061`.

Galtung, J. (1969). *Theory and Method of Social Research*. Columbia University Press.

Gao, S., Krogstie, J., and Gransaether, P. (2008). Mobile services acceptance model. In *Convergence and Hybrid Information Technology, 2008. ICHIT '08. International Conference on*, pages 446 –453.

Gao, S., Krogstie, J., and Siau, K. (2011). Developing an instrument to measure the adoption of mobile services. *Mob. Inf. Syst.*, 7:45–67.

Gao, S., Moe, S. P., and Krogstie, J. (2010). An empirical test of the mobile services acceptance model. In *Proceedings of the 2010 Ninth International Conference on Mobile Business / 2010 Ninth Global Mobility Roundtable*, ICMB-GMR '10, pages 168–175, Washington, DC, USA. IEEE Computer Society.

Google (Accessed on June 13, 2011a). Google maps icons. `http://code.google.com/p/google-maps-icons`.

Google (Accessed on June 14, 2011b). Google places. `http://www.google.com/mobile/places/`.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quartely*, 28(1):75–105.

Hinze, A. and Buchanan, G. (2006). The challenge of creating cooperating mobile services: experiences and lessons learned. In *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ACSC '06, pages 207–215, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Hong, J.-y., Suh, E.-h., and Kim, S.-J. (2009). Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36(4):8509 – 8522.

Hussain, A. and Ferneley, E. (2008). Usability metric for mobile application: a goal question metric (gqm) approach. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '08, pages 567–570, New York, NY, USA. ACM.

Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33:470–516.

Joseph, J. H., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. pages 241–250.

Kabassi, K. (2010). Personalizing recommendations for tourists. *Telematics and Informatics*, 27(1):51–66.

Kaikkonen, A., Kekäläinen, A., Cankar, M., Kallio, T., and Kankainen, A. (2005). Usability testing of mobile applications : A comparison between laboratory and field testing. *Journal of Usability Studies*, 1(1):4–16.

Klobuchar, J. A. (1996). Global positioning system: Theory and applications. vol. 1. *Washington, DC, American Institute of Aeronautics and Astronautics, Inc. (Progress in Astronautics and Aeronautics*, 163:485–515.

Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., and Malm, E.-J. (2003). Managing context information in mobile devices. *IEEE Pervasive Computing*, 2:42–51.

Kwon, O. B. (2004). Modeling and generating context-aware agent-based applications with amended colored petri nets. *Expert Systems with Applications*, 27(4):609 – 621.

Lillegraven, T. and Wolden, A. C. (2010). Location-aware bayesian recommender systems. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU).

Moe, S. P. (2009). Design and evaluation of a user-centric information system. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU).

Oracle (Accessed on June 13, 2011). Java ee at a glance. `http://java.sun.com/javaee`.

Paternò, F., Russino, A., and Santoro, C. (2007). Remote evaluation of mobile applications. In *TAMODIA*, pages 155–169.

PostGIS (Accessed on June 13, 2011). Postgis. `http://postgis.refractions.net`.

Røine, P. C. (2010). Tourist application for android.

Schiller, J. and Voisard, A. (2004). *Location-Based Services*. Morgan Kaufmann.

Schmidt-Belz, B., Laamanen, H., Poslad, S., and Zipf, A. (2003). Location-based mobile tourist services - first user experiences.

Schwinger, W., Grün, C., Prüll, B., Retschitzegger, W., and Schauerhuber, A. (2005). Context-awareness in mobile tourism guides – a comprehensive survey. Technical report.

Search, H. (Accessed on June 13, 2011). Hibernate search. `http://www.hibernate.org/subprojects/search.html`.

Solutions, V. (Accessed on June 13, 2011). Vivid solutions. `http://www.vividsolutions.com/jts/jtshome.html`.

Spatial, H. (Accessed on June 13, 2011). Hibernate spatial. `http://www.hibernatespatial.org`.

Spiekermann, S. and Berlin, H. U. Z. (2004). General aspects of location-based services. In *In Location-Based*. Morgan Kaufmann Publishers.

Takeda, H., Veerkamp, P., Tomiyama, T., and Yoshikawa, H. (1990). Modeling design processes. *AI Mag.*, 11:37–48.

Tomcat, A. (Accessed on June 13, 2011). Apache tomcat. `http://tomcat.apache.org`.

Trondheim, T. (Accessed on June 16, 2011). Trådløse Trondheim. `http://tradlosetrondheim.no/`.

Vaishnavi, V. (Accessed on June 14, 2011). Design research in information systems. `http://desrist.org/design-research-in-information-systems`.

Wium, M. (2010). Design and evaluation of an personalized mobile tourist application. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU).

Yu, C.-C. and Chang, H.-P. (2009). Personalized location-based recommendation services for tour planning in mobile tourism applications. In *Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, EC-Web 2009, pages 38–49, Berlin, Heidelberg. Springer-Verlag.

Zhu, F., Mutka, M. W., and Ni, L. M. (2005). Service discovery in pervasive computing environments. *IEEE Pervasive Computing*, 4:81–90.