Miika Nordström

A case study in social media mashup concept validation

Aalto University
School of Science and Technology
Department of Electronics, Automation and Communication

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Espoo, 28.04.2010.

Supervisor: Adj. Prof. Timo Korhonen

Instructor: Adj. Prof. Timo Korhonen

Aalto University
School of Science
and Technology

AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Miika Nordström	
Name of the Thesis:	A case study in social media mashup concept validation	
Date: 28.04.2010	Language: English	Number of pages: 11+88+23

Faculty of Electronics, Communications and Automation
Department of Communications and Networking
Professorship: Communications Code: S-72

Supervisor:	Timo Korhonen, Adj. Prof.
Instructor:	Timo Korhonen, Adj. Prof.

In social media user-generated content and sharing are emphasized. As a subcategory for social media there are social networks that connect people. Mashups are applications that combine data from existing sources and display the data in a new context.

This study considers the case of AmiCarta, a social network service that focuses on location information. The study focuses on validating the concept and evaluating the used methods for validation as well as the obtained results.

The concept is validated with user tests. Each user test was conducted in three parts. In the first part, the user was demonstrated a mock-up prototype of the mobile user interface for the product, that was created using paper prototyping and the prototype responds to the user in a wizard-of-oz implementation. The second phase consisted of a semi-structured interview and the third phase of a questionnaire.

The methods of the study produced mainly qualitative information. As a drawback, the use of a mock-up prototype mainly generated information about the existing features of the concept. Qualitative results gave insight on how the concept could be further developed. The quantitative results of the questionnaire were useful in ranking the features.

As a result of the study, the focus of the service was moved from a map application that has different features to an application that enables communication between users and supports this communication with the users' location information.

Keywords: Location-Based Services, Social Media, Mashup, concept validation

AALTO YLIOPISTO **DIPLOMITYÖN TIIVISTELMÄ**
TEKNILLINEN KORKEAKOULU

Tekijä:	Miika Nordström
Työn nimi:	Sosiaalisen median mashup -konseptin validointi

Päivämäärä: 28.04.2010 Kieli: Englanti Sivumäärä: 11+88+23

Elektroniikan, tietoliikenteen ja automaation tiedekunta
Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoliikenne Koodi: S-72

Valvoja:	Timo Korhonen, Dosentti
Ohjaaja:	Timo Korhonen, Dosentti

Sosiaalinen media korostaa käyttäjien välistä vuorovaikutusta ja käyttäjien luoman sisällön mahdollistamista ja jakamista. Sosiaaliseen mediaan kuuluvat myös sosiaaliset verkostot, jotka yhdistävät ihmisiä. Mashupit taas ovat palveluja, jotka yhdistävät tietoa eri lähteistä ja näyttävät tämän tiedon uudessa kontekstissa.

Tässä työssä käsitellään sosiaalista verkostoa nimeltä AmiCarta. AmiCarta keskittyy paikkatiedon hyödyntämiseen sosiaalisessa verkostossa. Työn tarkoituksena on validoida palvelukonsepti käyttäjillä. Työssä analysoidaan validointiin käytettyjä metodeja sekä käyttäjätestien tuloksia.

Käyttäjätestit koostuvat kukin kolmesta osasta. Käyttäjille annetaan prototyyppi tuotteen käyttöliittymästä, jonka he käyvät läpi. Prototyyppi on luotu paperiprototyypeillä ja toimii wizard-of-oz metodilla. Protoyypin tarkastelun jälkeen käyttäjiä haastatellaan puolistrukturoidulla haastattelulla ja esitetään kyselylomake.

Käytetyt metodit tuottivat pääasiassa kvalitatiivista tietoa. Prototyypin käytöllä oli haittavaikutuksena se, että se tuotti pääasiassa tietoa olemassa olevista ominaisuuksista, eikä juurikaan auttanut uusien ominaisuuksien ideoinnissa. Kvalitatiiviset tulokset antoivat ideoita siitä, mihin suuntaan konseptia tulisi viedä. Kyselylomakkeen tuottamat kvantitaaviset tulokset taas olivat käytännöllisiä ominaisuuksien vertailussa.

Tutkimuksen perusteella konseptin perusajatus muuttui karttapalvelusta, johon on lisätty erilaisia ominaisuuksia palveluun, joka perustuu kommunikaation helpottamiseen ja tukemiseen paikkatiedolla.

Avainsanat: Paikannuspalvelut, Sosiaalinen media, Mashupit, Konseptin validointi

Acknowledgements

I would like to thank especially Matti Lankinen and Esa Mäkelä at Indagon for giving me the opportunity to do this thesis for Indagon. Validating and further developing the concept has been an interesting task in an inspirational work environment.

At Indagon especially Eero Nevalainen has given valuable advice and helped with this thesis. I am especially grateful for the useful tips in writing a scientific paper. For supervising and instructing the thesis I would like to thank Adj. Prof. Timo Korhonen, who gave lots of excellent views on the content of the thesis.

For assistance in the thesis I wish to express my gratitude to the whole AmiCarta team at Indagon as well as to Jukka Liukkonen for keeping morale high at the workplace and to Samiseppo Aarnikoivu for his advice and feedback on this thesis.

Additionally thanks to Meri. I wish to also thank my parents Anne and Kaj.

Espoo, 28.04.2010

Miika Nordström

Table of Contents

Figure Index

Table Index

Chart Index

Key Concepts

Location Based Services

Services that use user location as a key service element. These services give the users the possibility to locate other users' devices as well as themselves. The location requests can be made by the users or by other parties, such as the service operators. The two major actions in an LBS are obtaining the user's location information and utilizing it in the service. [GSM Association, 2003]

Anonymous Instant Messaging

Anonymous instant messaging is a service in which the conversational partners do not know before-hand, who the opposite partner is. The conversational partners are chosen according to the distance between the users as well as matching the profile information given by the users. If the users both decide to like the conversational partner, the system allows them to exchange contact information automatically. [Burak et. al, 2004]

Social media

Services that are based on web 2.0 technologies and allow the creation and sharing of user-generated content. [Kaplan et. al, 2010] Social media has five key characteristics in participation, connectedness, openness, conversation and community. [Mayfield, 2006]

Social networks

Boyd et. al define social networks as having three distinct characteristics. Social networks construct a profile for the users, that is either public or somewhat public. These networks also list the users' networks of connections and allow the viewing of these lists if not specifically denied. [Boyd et. al, 2007]

Mashups

A service that combines existing data sources with each other and new data sources and present the information in a new context. Mashups may also include presentation elements from different sources as well as application functionality. The aim for mashups is to present these elements in a new and useful way as an application or a service. [Yu et. al, 2008]

Abbreviations

AC-GEO	AmiCarta's location knowledge protocol
AC-GPS	AmiCarta's interface for sending the GPS location to the service using a mobile terminal
AC-WEB	AmiCarta's interface for displaying data in the UI
AIM	Anonymous Instant Messaging
API	Application Programming Interface
BBS	Bulletin Board System
CEO	Chief Executive Officer
CGW	Content Gateway
CGW3	Content Gateway version 3.0
CRUD	Create-Read-Update-Delete
GeoRSS	Geographic Really Simple Syndication
GIS	Geographical Information Systems,
GMLC	Gateway Mobile Location Centre
GW	Gateway
HTML	Hypertext Markup Language
IDL	Interface Description Language
IM	Instant Messaging
ISO	International Standardization Organization
LBS	Location Based Services
MMO	Massively Multiplayer Online
MMS	Multimedia Messaging Service
OSM	OpenStreetMap
P3	People-to-People-to-Geographical-Place
PHP	Hypertext Preprocessor
Podcast	Personal-On-Demand broadCast
POI	Point of Interest
RSS	Really Simple Syndication
SMS	Short Message Service
Snaptz-IF	The interface for the Snaptz service that receives area information about the terminals

SWOT	Strengths, Weaknesses, Opportunities, Threats
UCD	User-centered design
UI	User Interface
WEB-FE	Web Front End
WMS-C	Web Map Service – Cached
WSDL	Web Services Description Language
XML	Extensible Markup Language

1 Introduction

Social media carries with it the transformation of media to include the consumers of the media. Traditional media broadcasts content from one-to-many, the idea behind social media is changing this approach to many-to-many broadcasting. Social media has been subject to incredible growth, even many traditional news sites have adapted to social media by adding the possibility to participate in the news by commenting them.

Perhaps the best example of the growth rate of social media is the social network Facebook[1], which has grown in active users from 200 million to 400 million in under a period of one year [Zuckerberg, 2009] [Zuckerberg 2, 2010]. There seems to be little limit to the growth of social media except the total number of users using the internet, which is why it seems like an interesting area for business.

There are several types of social media. One of these types is mashups. Mashups combine data from one or many existing sources and present the data in new contexts. Mashups can also introduce new data sources, it is not necessary to use only the existing sources. Mashups save time and effort for developers since they use user generated or existing data that the developers don't have to produce to the system.

Early mashups were relatively simple, for example one of the first mashups, housingmaps.com[2], took for sale ads and visualized the locations of the houses on a map. [Clarkin et. al, 2007] Housingmaps.com is still up and running. A common feature in early mashups was that they were consumer-driven. Lately though business mashups are increasing in number, as businesses are finding mashups a useful tool.

Mashups can either support the company's existing business or be a source of direct revenue by offering a useful service [Clarkin et. al, 2007]. For example, let's consider the call center example as an example of a support mashup. In it, the call center has a mashup that provides local information about the client's neighborhood providing a more intimate customer experience [Clarkin et. al, 2007]. The mashup can also be a complete service as is the case in the study of AmiCarta discussed in this thesis.

AmiCarta is a mashup service that is a social network. The service shows the location of the users to the users' friends. The location information is based on showing the location of the user's mobile phone. In the social network the users can communicate with each other using different methods.

The AmiCarta system uses network positioning in conjunction with GPS positioning where available. The system doesn't require a mobile client for the network positioning, which is a huge advantage, since that way all users of mobile phones can be tracked. If the positioning required a client, it would rule out the majority of mobile phone users. The mashup brings together a map server that uses OpenStreetMap.org[3] maps with the mobile phone operator's location information and the user's own GPS location information. In addition to this the other features of the system are discussed in this thesis.

[1] http://www.facebook.com

[2] http://www.housingmaps.com/

[3] http://www.openstreetmap.org

In this thesis the AmiCarta concept is validated with potential users of the system using different methods. The results of the validation are then used to evaluate and shape the existing concept. The scope of the study concerns only the validation process of the concept.

In the validation process, the user is presented with a mock-up prototype of the system in question. The user is allowed to get to know the prototype and is encouraged to share thoughts about the defects in the system. After this the user is interviewed using an open interview that is completed with a questionnaire. These interviews are conducted until the set criteria for the process are achieved. The validation follows the ISO 13407 standard in an iterative approach.

Once the validation results are obtained, they are analyzed in a focus group session with the development team of the project. In the focus group sessions the features of the service are clustered into smaller clusters that form entities within the service. Out of these clusters, the best ones for implementation are chosen and a feature implementation timeline is created. The feature implementation timeline is the result of the analysis of the user tests.

The basis of the thesis comes from the research questions. The research questions are the following:

1. What kind of a method will be used to validate the AmiCarta concept?
2. How will the chosen validation methods affect the results of the study? What are the nature of the results of the chosen validation methods?
3. How can AmiCarta be linked to other social media services?
4. What kind of choices are there for the evolutionary pathway of the concept and what defines the path chosen?
5. How is the concept molded by the results of the validation?

The first question regards the outcome of the process of validation. The choice of methods to use affect the nature of the results. For this study, it is wished that as much of the results are qualitative in nature as possible. Qualitative information is desired, because the concept is at an early stage, and further development is something the team wishes to do. Also an interesting question about the chosen methods is whether the chosen methods created a hybrid of the methods or whether it was just a set of methods used together.

The question of how the methods affect the results of the study can mean more than one thing. On the other hand it implies, whether the data that the researchers are looking for is quantitative or qualitative. On the other hand it calls into question if the study itself prevents some kind of results from surfacing in the test. The ideal situation is, of course, that the study doesn't limit the possibility of a certain type of results for the study. This is a pretty hard thing to predict beforehand.

An important aspect in all social media is that it links to other social media. Several successful social media services offer the possibility to link information in to them and from them. This allows the services to get important publicity for free. Another benefit from this is the spreading of information to a very large group of people easily. For the user it also adds value to a service to allow to use it in conjunction with other services the user has used before using AmiCarta.

As a result of the research, there will be a large amount of knowledge of the concept itself. That allows the design team to make choices that will define the evolutionary pathway of the concept. The results will also answer such further questions as which features appeal to the users the most, which will provide important information as to what needs to be implemented to the service at the first stage.

An important part of the results is the question if there will come new ideas from the test subjects. These ideas can significantly shape the concept. It is hard to imagine a situation where the design team had thought of everything that the users can come up with.

The study itself is focused on studying how well the selected methods worked for the study as well as how the users react to the concept.

2 User-centered design process

A user-centric design process takes into consideration the needs of the users and incorporates them to the design process. One of the best known models for user-centered design is the ISO 13407 that is employed also in this study.

2.1 ISO 13407 process

The ISO 13407 standard is titled "*Human-centred design processes for interactive systems"*. This means that the standard discusses user-centered development for systems that interact with the user. The standard suggests that user-centered design consists of five stages, out of which four are iterative. These stages are planning the human centered process, specifying the use context, specifying the organizational and user requirements, producing design solutions and evaluating the designs against the user requirements. If the product satisfies the user requirements, the process stops. If the product does not satisfy the requirements, the process starts over with specifying the use context thus skipping step one.

The stages are best represented as the following figure 1.

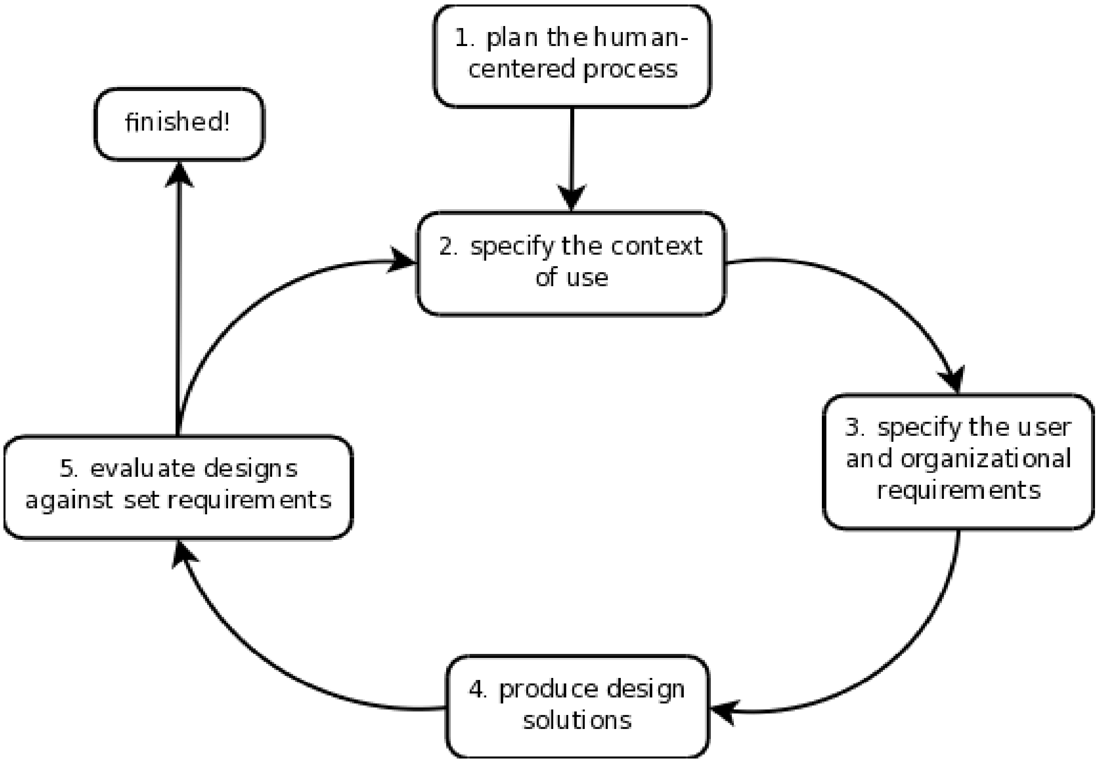

1. plan the human-centered process
finished!
2. specify the context of use
3. specify the user and organizational requirements
5. evaluate designs against set requirements
4. produce design solutions

Figure 1: The ISO 13407 design process [EMMUS, 1999]

Like every successful process, the design process starts with planning. In the planning process, the designers choose which User-Centered Design (UCD) methods they use in the development. Planning serves as a way to gather the usability team together. In planning, the success criteria are defined. This means that it is defined, if there is a minimum level of usability that must be achieved or if the process is to be limited by either the amount of time available or by the budget available. The result of the planning phase is a validation plan. This plan defines what kind of user interaction is used in which part of the project. The validation plan is not set in stone after the planning phase, but is constantly modified.

The second phase of the ISO 13407 process is specifying the use context. The overall quality of the final product depends on understanding the context of use. [EMMUS, 1999] The context usually consists of the physical and organizational environment for the users, the characteristics that define the users and the tasks that need to be completed by the users. To help with understanding the context of use, the developers can use data that has been collected earlier of the product, if the product has been launched. If the developers get to work with an unreleased product, they can use project managers and user interviews in determining the context of use.

There are at least seven things designers have to put in context in order to successfully specify the use context. These include the following.

1. The characteristics of the target user
2. The tasks that the users will complete with the system
3. The hierarchy of the tasks in a global level
4. The overall tasks that need to be completed by different user groups and characteristics of the tasks that might affect the usability of the system
5. The allocation of activities and the operational steps between the user and the system
6. The use environment
7. Some minimal and optimal system requirements in order to better test the system

The results of this stage can be described in a Context of Use Description, that is used to describe the tasks, users and environment. This description also identifies what aspects have an important effect on the system design.

When the use context has been described, it is time to specify the user and organizational requirements. These requirements are connected to the use context. Requirements are set for the following things.

1. Human computer interface and the design of the workstation
2. The tasks conducted by the users and the content of the tasks
3. Effectiveness in performing the tasks, especially in transparency of the application to the user
4. Co-operation and communication between users
5. Performance against operational and financial objectives

When assessing these requirements, at least the following ISO 9241 part 11 model variables should be considered: efficiency, effectiveness and satisfaction. This part of the process is extremely critical, and the success of the final product is most dependent on this stage of the process [EMMUS, 1999].

The fourth part of the process is creating design solutions. Design solutions should take advantage of the current state-of-the-art solutions that exist as well as the experience and knowledge of those in the design team. The process should involve at least the following aspects.

1. Using standards and guidelines to design
2. Creating a prototype of the design solution
3. Testing the prototype with users
4. Creating a new prototype with help from the users' comments
5. Iterating the process for as long as necessary.

There are several different standards and guidelines for designing solutions.

Prototypes created in this stage vary from lo-fidelity prototypes, such as paper prototypes to hi-fidelity prototypes, such as wizard of oz -prototypes and actual working prototypes of the final product. Generally the fidelity of the prototype depends on the phase it is created in, since the ISO 13407 process is iterative. Prototypes should be tested with only a few users, and then a new and improved prototype needs to be created from the users' suggestions. This way the designers don't get the same results from the user tests over and over again. If the first prototypes are of less fidelity to the completed system, they require much less work than the prototypes created closer to completing the project.

One of the hard parts in this stage of the process is checking the set requirements to actually be valid in relation to how the users use the system. Different types of prototypes also work best on different user groups. Some users might work best with a mock-up, while some might get the best results by just simply looking at storyboards. This greatly affects the type of feedback the users will give. [EMMUS, 1999]

The final stage of the process is evaluating the design solutions against the previously created system requirements. Evaluation can be done with at least two different approaches. The evaluation can be formative, which means that the designers are given feedback in what needs to be done better. Another approach is summative evaluation, which means that the system is evaluated on the basis of whether or not it covers the set system requirements.

Depending on the summative evaluation, the process either stops here, or continues again from stage two of the process. If the set requirements are fulfilled, the process stops, and the design is sufficient when compared to the requirements. The overall quality of the system depends on how good the set requirements were. If the requirements are not met, the iteration continues from stage two of the process as mentioned before.

2.2 Applicable UCD method selection

There are several alternative methods for all the stages in the ISO13407 development model. How can designers choose the best method for themselves? There are several criteria on choosing the methods. A few of these criteria include the time available for conducting the design, the available money for the design process or something as simple as the number of users available for testing.

For the AmiCarta project, the most important criteria are the amount of time and money. UsabilityNet [UsabilityNet, 2006] offers a table that deselects UCD methods from the pool of suitable ones according to three criteria. The criteria are limited time/resources, no direct access to users and limited skills/expertise. This table was used, because it is an easy and effective way to sort out some of the methods that aren't suitable for the study.

UsabilityNet's methods table includes 39 different methods. These methods are categorized according to the part of the UCD process they are used in. The categories are “planning and feasibility” with 6 methods, “requirements” with 13 methods, “design” with 8 methods, “implementation” with 2 methods, “test and measure” with 6 methods and “post release” with 4 methods. For this project, the filter “limited time/resources” was switched on, leaving us with a total of 14 methods.

With the selected criteria, the methods that are left for selection are for “planning and feasibility” getting started, stakeholder meeting and analyse context. For “requirements” the remaining methods are called context, evaluating existing systems, affinity diagramming, scenarios of use and requirements meeting. For the “design” phase of the process, the only suggested option is paper prototyping. “Implementation” is handled with two alternatives, style guides and rapid prototyping. For testing and measuring the concept, UsabilityNet suggests the use of one of two methods, the diagnostic evaluation or subjective evaluation. Finally, once the product has been released it is time for post release evaluation. For this purpose, UsabilityNet suggests subjective assessment.

For the analysis of the results the chosen methods are a focus group session and SWOT analysis. These are chosen because of the amount of time needed to complete these analysis' is very little.

2.2.1 Stakeholder meeting

The purpose of a stakeholder meeting is to decide the way the project shall be done. The meeting will decide which methods shall be used in the project, the timetable and the budget. Even though the process is iterative, in order to get results, there needs to be a time frame for the studies. Present in the meeting should be at least the stakeholder or their representative and the project manager and a representative of the users of the product.

2.2.2 Evaluating existing systems

Very rarely is a product in development completely different from everything else in the market. Usually there are similar products. These other products can be evaluated when designing a new project. This way the pitfalls of the existing products can be avoided. This method also gives important information about the existing systems, which allows the designers to better understand in which direction the product should be taken.

In short, this method consists of selecting important tasks in the system that is to be created and evaluating how they are done in earlier versions of the product or competing products. The results from this method include usability problems highlighting the areas of improvement, and an overview of what kind of problems might arise when designing the new system.

2.2.3 Affinity diagramming

Affinity diagramming is a powerful tool for organizing data. This method is based on organizing data on little pieces of paper, for example post-it notes, in a way that the little pieces of the concept create bigger pieces of the concept that create the concept. This method allows a visual representation of the concepts within the concept, and allows fast regrouping of the data.

Affinity diagramming was developed in the 1960's in Japan [ASQ, 2009]. Affinity diagramming is meant to sort out a lot of data, that seems like a chaos at first. It is especially useful after an ideation session, such as brainstorming.

The procedure works in four simple steps. The first step begins with recording all concepts and ideas on sticky notes. In the second phase of affinity diagramming, the ideas are grouped in to small groups, silently. It is very important that no one speaks during this phase to prohibit others from disapproving of ideas at this stage. In the third stage, the participants discuss the fruits of their labor, and create labels for the small groups of concepts, eventually ending up with supergroups in the fourth stage of the process, that contain a number of smaller groups.

The real benefits of affinity diagramming include the ability to create concepts that are not self-explanatory. This means that the thought process isn't ordinary, and the outcome might contain something unusual. Thinking outside the box is usually very hard, and can result in a competitive edge for the project.

2.2.4 Paper prototyping

Paper prototyping is a method that describes itself pretty well. Paper prototyping is what it says it is, creating prototypes on paper. The strength of paper prototyping is that it allows designers to very rapidly change their prototypes without having to make a fully functional system every time. Also correcting errors in the designed system is very easy, just erase the sketch, and make a new one!

Scanlon suggests that paper prototyping is not only a good method of visualizing a user interface, it also works as great team building for the team creating the prototype. [Scanlon, 1998]

A strong good quality of paper prototyping is that, it takes a small amount of time to make iterations to the interface. This also is in a sense a drawback of the method, it doesn't generate anything that can be used as such. The code required for the user interface is not generated with paper prototyping, but that is mostly a good thing, because the more iterations are made before coding, the less code needs to be rewritten.

Bolchini et. al also describe how paper prototyping can be used in a creative way they call paper in screen prototyping [Bolchini et. al, 2009]. In paper in screen prototyping the drawn paper prototypes are presented to the user on the screen of a mobile device. This helps the users figure what the actual product will look like on the screen of the intended device.

2.2.5 Wizard of Oz

The Wizard of Oz design method tries to create a believable experience to the user who is testing the system. This way the designers can give an authentic experience to the user without actually building a functioning system. This reduces the workload of creating the actual system, because there is less need for iteration at that stage of the process.

Wizard of Oz traditionally requires a user, an observer and a person acting as the system that is being tested, referred to here as the computer. The user makes the choices of action within the system while the computer executes the required modifications to the system to make the system look like the user had done the action. This way the actual mock up won't need any logic behind it, it just has to look good enough so that the user believes that the system would actually function like this.

2.2.6 Rapid Prototyping

Rapid prototyping or rapid application development is a methodology that uses minimal planning in favor of creating lots of prototypes and molding the application as the process goes on.

The logic behind rapid prototyping is to create a prototype, then test it, and create a new one rapidly, that fixes the flaws of the previous prototype and possibly improves the prototype in other areas. This allows the team to take liberties in the design process. Also the product doesn't have clear specifications at this stage of the process, which means that the specifications are subject to change at all times.

2.2.7 Focus group session

A focus group session is a meeting in which a group of users gather to discuss a set topic. The focus group session is usually loosely planned and aims in provoking emotions, ideas and perceptions on the subject at hand from different perspectives. Usually focus group sessions are held with a number of users of the final product, but in this study the focus group session included the developers of the system.

A big advantage in a focus group session is that the participants in the focus group have a chance to learn from each other, since different people have different views on the discussed topic.

In the focus group session one of the participants is the moderator for the exchange of views so that not everyone talks at once. The moderator keeps the conversation on topic and hands out turns to speak to the participants. The moderator takes no stand in the conversation and presents no personal views and focuses on moderating the discussion and giving new topics to the group.

2.2.8 SWOT analysis

The SWOT analysis describes how to evaluate something using it's strengths, weaknesses, opportunities and threats [European Commission, 2006]. The method helps improve the evaluated product's competitiveness. In the analysis the product is looked at from different perspectives to find out answers to the four categories. The strength category discusses the products good features and competitive edges when compared to others. The weaknesses category focuses on narrowing out what needs to be improved in the product. The opportunities category comes down to finding out what external conditions could help the product. The threats category discusses the external obstacles to the product.

The SWOT analysis has a few pros and cons. Perhaps the best quality in SWOT is that it is easy to implement. It is very simple and flexible and requires no technological expertise. The drawbacks for SWOT include not getting any suggestions on fixing the situation, having long lists of factors that need to be taken into consideration and lacking prioritization of the factors found in the analysis.

3 Social Media

Social media is a constantly growing field of interaction for users in the world wide web. It is something that has surfaced in the recent years and as a new area of research, it is still going through changes. This chapter will present different types of social media and the concepts that come with it.

3.1 Definition

Andreas Kaplan and Michael Haenlein define social media as "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content" [Kaplan et. al, 2010].

This definition states, that the key ingredient in social media is the user-generated content and the technological foundation that enables generating the content easily. Some of the well-known social media services include Youtube[4], MySpace[5] and Facebook.

Anthony Mayfield [Mayfield, 2006] describes social media as having five descriptive characteristics: participation, connectedness, openness, conversation and community. Social media services give the users a platform to interact with each other in various ways. Social media takes the traditional approach of creating content from one-to-many to many-to-many. This may put a doubt on the credibility of the information shared. In practice this means that potentially every user can create content and comment on another user's content. Not all the users however do this. Most users are satisfied just looking at the content created by others. Typically users are divided so that 1% of users create the content, 9% of the users comment on the content and 90% of users passively view the content [Mayfield, 2006].

3.2 Types of social media

There are several types of social media. This chapter discusses the different types and what is distinct about the discussed type. Common forms of social media include for example bulletin board systems, blogs, wikis, podcasts, social networks, multimedia sharing services, online gaming and mashups.

3.2.1 Bulletin boards

One common form of social media are various bulletin board systems. They have been around since before the term "social media" was coined up. Electronic forums existed even in the time before internet, when people connected to BBS systems. Bulletin board systems can be dedicated to a single topic or accept conversation from all kinds of areas. Usually bulletin boards are moderated by some users who have the rights to delete messages that don't abide to the rules of the board.

Bulletin boards are usually referred to as forums. Bulletin boards are websites that display messages sent by users to the board. These messages can be replied to by all the

[4] http://www.youtube.com

[5] http://www.myspace.com

users of the board. The original message and it's replies are automatically sorted in to threads that show the message and replies to it. Threads can be categorized by the topic in question. This is done manually and it helps the users find what really interests them.

Bulletin boards usually have so called moderators. These moderators are super-users of the bulletin board and have administrative privileges. They can edit, move and delete single messages and entire threads. Moderators can also ban individual users from the bulletin board if they consider the banned users don't obey the rules of the bulletin board.

3.2.2 Blogs

Blogs are said to double in quantity every five months [Mayfield, 2006]. The amount of blogs goes in the hundred millions, which is too much to overlook the power of blogging. Blogs can center around a theme, whether it is someone's personal diary or a blog that discusses new technology. There are also corporate blogs, that offer insight to different corporations and to market the corporations' products. Blogs are usually created with tools that allow the user not to have any knowledge in programming to create the blog. This allows more users to create their own blogs. These tools also allow commenting the blog as well as embedding different types of media on the blog.

Blogs are by definition weblogs, so they exist in the web. In 1999, Brad Graham came up with the term blogosphere to depict all the blogs and their interconnections [Graham, 1999]. The term blogosphere was meant as a joke initially, but it stuck, and now defines all the blogs and the connections between them.

3.2.3 Podcasts

Podcasts, or Personal-On-Demand broadCasts, are basically taped “radio shows”, that the user can download and listen to at his/her own convinience. The emergence of handheld mp3 players made this form of social media possible. Since podcasting doesn't have the users communicating directly to each other via the podcasts in real-time, it is hard to understand why they are considered social media. Perhaps it is the ease in replying to a podcast by creating one that discusses another. Podcasts can be created by anyone and there are websites that collect user-generated podcasts and distributes them. An example of a similar service is zencast.com[6]. Real radio stations also have started to save their programs into podcasts and distributing them on the web. An example of a radio station that podcasts their transmissions is the finnish radio station Radio Rock[7].

3.2.4 Wikis

Wikis are web pages that allow their content to be modified by anyone or by a closed group of users. This enables users who are not familiar with HTML to collaborate in creation of web content. One of the most famous wikis is the encyclopedia Wikipedia[8]. Wikis provide users with huge amounts of information, but the information cannot always be trusted as such, since anyone can write to a wiki.

[6] http://www.zencast.com

[7] http://www.radiorock.fi

[8] http://www.wikipedia.org

Wikis were first developed by Howard Cunningham, who described wikis as “the simplest online database that could possibly work”. [Cunningham, 2002] Wikis are commonly mistaken to be one site: “Wikipedia”, while wikipedia is just one wiki. The first wiki was created by Cunningham and was called “WikiWikiWeb”. WikiWikiWeb still exists today at c2.com[9].

One of the biggest benefits of a wiki is that it can be edited with a basic browser, and it requires no add-ons or coding skills. This basically enables every web user to edit and contribute to the wikis of their interest. Wikis also make linking to other wikipages easy, so that the user can easily connect the topic to other topics in the wiki and in the web.

Wikis usually display the user's modifications instantly, which puts a question mark on the reliability of the data in the wiki. The flip side of the coin here is that when wrong information is put into a wiki, it is likely that another user will spot the information and remove it from the wiki.

3.2.5 Social Networks

Social networks allow users to create social networks that include their friends and other connections. These services also allow the user to create a profile for themselves. The services provide users with a way to connect with lost friends or find new ones. Social networks can also be specialized in something, for example LinkedIn[10] provides users with a social network dedicated to users' careers. Some popular social networks of the time are Facebook, Bebo[11] and MySpace.

On January 7th 2009, Facebook reached 150 million active users [Zuckerberg 3, 2009]. The same report also claims that Facebook has members from all the continents in the world. Further on April 9th 2009 Facebook had already 200 million users. [Zuckerberg, 2009] Facebook has been subject to incredible growth between the year 2008 and 2009. According to a blog at compete.com the number of unique visitors per month has risen from 28,563,983 visitors [Freiert, 2008] to 68,557,534 unique visitors per month [Kazeniac, 2009] making the site the biggest social media service. In february 2010 Facebook had over 400 million users [Zuckerberg 2, 2010], which means the number of users had doubled in under a year.

Social networks are meant to help people keep in touch. They include services to communicate with other users, share pictures, do tests as well as play games that have a shared high score list to compare your results to your friends. The idea behind these networks is to allow people to easily stay in touch via the Internet. They also provide a cheap way to communicate since there are no telephone bills or travel fares that would accumulate otherwise.

The business model behind social networks includes advertising. The social networks are free to the user and collect their revenue from adverts. The bigger the network, the bigger the ad revenues, and the more probable that the service survives.

[9] http://c2.com/cgi/wiki?WikiWikiWeb

[10] http://www.linkedin.com

[11] http://www.bebo.com

3.2.6 Multimedia sharing

Multimedia sharing services are communities that evolve around a specific type of content. Examples of multimedia sharing services would be YouTube for video. Multimedia sharing services usually allow their content to be embedded to user's own websites or systems. This helps the service to gain visibility and advertises the service. Users naturally benefit from this by being able to share content easily.

Another type of multimedia sharing is sharing one's meta data on their media usage. For example, Last.fm is a popular service for sharing what music you like to listen to. Last.fm works by collecting the information of the songs played on the user's computer or ipod and creates a database containing eg. the number of plays the song has gotten.

3.2.7 Online gaming

Online gaming is a very big business. Players often pay a monthly fee for playing games online with other players. These games are called massively multiplayer online -games, or MMO games. The biggest subscription-based MMO in the western countries is World of Warcraft with over 11 million subscribers [Lee, 2008]. MMOs have turned out to be a huge business, and new MMOs are emerging on the market constantly. Some games have their own societies. One good example of this is LittleBigPlanet, a game that has it's very own user-generated facebook-clone, the sackbook.

Online gaming is a mainly commercial field of social media. Almost all online games either require a fee to acquire or display adverts to fund for the development of the game. Some online games require a monthly subscription of the game in order to keep the game running.

Online games can be categorized according to if they require an independent client program to run, or if they work directly from a web browser. Typically browser-based games are funded with adverts and client-based games cost money to the user. Online games sometimes require also a monthly fee.

Another type of MMOs are free browser-based games, such as Farmville, that has a huge amount of players, over 70 million monthly users by 2009 [O'Neill, 2009]. The players of Farmville don't have to pay for the game, but they can buy extra credits that help them with the game via micro-transactions.

3.3 Social Web

Social web is a concept that describes how people connect with each other via the World Wide Web. Shaun Connolly [Connolly, 2008] has described social web via seven key attributes. These attributes are identity, reputation, presence, relationships, groups, conversations and sharing.

Identity represents the user. What's interesting is that in the social web people can choose to provide whatever information of themselves not depending on it's truthfulness. However, users can be encouraged to use their real identities by creating an application such as Facebook, which connects real life friends in virtual reality.

Reputation stands for what the other users think of the user. In other services this is done by giving a star rating to users, while other services don't need a rating. Especially those systems that show the user's real name seldom have a rating system for the users.

Presence in this context means the user's geographical location. Presence information has been quite hard to include to social media applications, because machines connected to the web can't usually tell their location in sufficient detail. With the popularity of GPS-enabled devices this is soon to change. Especially, because an increasing number of mobile phones have built-in GPS.

In social web applications, users want to keep track of their relationships. This is useful to the user, since it allows them to receive information on only other users who they are interested in. It also helps with privacy issues, because users can block each other from the services. In some services users can also block other users, who are not their friends.

Grouping can also be important to users, especially when they have lots of contacts. Anything that helps to organize the user's contacts is counted as grouping.

Since social web allows the user to connect to their friends, allows it also for users to converse with each other. This conversation can either be synchronous or asynchronous. The conversation is synchronic when both users are online at the same time and respond to each other's messages. Asynchronic conversation occurs when only one user is online and wants to communicate with an offline user. Applications can contain the possibility to leave messages, sort of like voicemail, or have a bulletin board, where also other users can read the left messages.

Content sharing is a key concept in social web applications. Information is shared between users, for example as pictures, sound or text. This is extremely useful for the user's if they can choose whom to share the information to.

3.4 Mashups

Mashup is a term used for so called “hybrid” applications in the web. Mashups combine existing services to provide a new service that neither of the original services provided. Mashups can also add new information to be combined with existing services. Mashups can be for business use, data use or consumer use. Mashups are a great way to create new services without having to start from the scratch.

Mashups are always meant for the end-user, and often designed by users themselves [Crupi, 2008]. Mashups can be categorized with the composition and the component model [Yu, 2009]. The composition model focuses on how the components are integrated, and the component model focuses on the nature of the components and how well they are glued together.

Component model

The component model defines how the mashup can be composed in the composition model. So in lay-mans terms, the component model describes the components that form the mashup together. The three key variables to the component model are the component's type, extensibility and interface.

The type of the model can be either data, application logic or user interface. If the type of the component is data, it means that the component is purely a data source.

When the type of the component is application logic, the component's function is to provide access to the application logic. Lastly, if the component's type is user interface, it means that the component also provides the user a graphical user interface.

The extensibility property of the component is an important one, when creating mashups that can be recycled. If the component is properly designed, it can be used again. The two main factors in extensibility are whether the user can use the component to create new components and whether the user can extend the existing component to allow new functions in the component.

The interface of the component defines how the component can be interacted with. Yu et. al present four usual types of interfaces. The first one is a simple create-read-update-delete, or CRUD, that is very simple and allows only these four operations on the component. The component can also have an interface that is accessible with a known programming language. This is most common in APIs. The interface can also be a markup language, such as HTML or XML or IDL or WSDL. And last but not least, the interface can be graphical user interface elements that are for the user to use.

Composition model

The composition model considers the mashup on the mashup's orchestration style, data-passing style, output type and whether it is instance-based or continuous. Or simplified, how the components of the mashup work together.

The orchestration style can be one of the following three main approaches: flow-, event- or layout-based. The flow-based orchestration style can be thought of as a flow chart. This orchestration style thinks of the mashup as working in a specific order between the tasks and components resulting in something reminding a flow chart. The event-based orchestration thinks of the system as a bunch of subscribers and publishers. In this orchestration the system works with some of the components publishing data, and others subscribing it. This is a powerful orchestration style, when the system needs to be synchronized. The layout-based orchestration sorts the components of the mashup according to the applications common layout. It also observes the system on how the components are affected by the user's interaction with other components.

The data-passing style is defined as one of two by Yu et. al [Yu et. al, 2008] as either a dataflow approach or as a blackboard approach. The dataflow approach considers the data as flowing from component to component when required, while the blackboard approach has a more traditional approach, in which the data is written to and read from variables when required. The blackboard approach is easier to understand as it is similar to programming languages.

The output type of the mashup can be either pure data, application logic type or user interface type. When the output type is data, the output is just this, data that can be used by other components in the mashup. The application logic type of output allows the mashup to be accessed via programmable APIs. If the output type is UI type, the composition provides applications with a user interface.

Whether the mashup is instance-based or continuous can be described as whether the mashup works all in one single thread or whether the mashup is multi-tasking. If the mashup is instance-based, the mashup works as following. All new actions within the mashup are activated when an appropriate signal arrives at the process. So in a way, all action needs to be provoked first kind of like a program that is run once. In the continuous model, there are threads for all the components that work simultaneously. These threads do the necessary actions that are needed to generate the output.

Development using existing editors

There are several different tools or editors for creating mashups easily. These include tools from large companies, such as Google, IBM, Intel, Microsoft and Yahoo. The companies offer these services to get developers to use their data and applications in the mashups that the developers create. The catch in these tools is, that if one creates a mashup using them, it gets increasingly more difficult to make money off the mashup. Another bad thing about these is that the mashup is tied to the services and it's hard to change the components.

The mashup development tools and editors have different properties. The two things that are best to categorize the editors are the development environment and the runtime environment. The development environment means how the editor is used to create the mashup, while the runtime environment deals with how the mashup is executed by the user.

The development environment consists of two different aspects. The first one is the interface of the editor. The interface is mostly dependent on the target audience of the editor. Usually the easier the editor is to use, the more difficult it is to create complex functionality that the editor doesn't directly support. The interface can be one of several options. The simplest editors are pure text editors, that let the developer do the coding, and then the editor compiles it. Some editors have an HTML markup approach, in which the developer also needs to do the coding self. The interface can utilize a drag-and-drop approach. In these editors the user simply needs to drag and drop components in the editor to create the mashup. The editor can also be a combination of these interfaces to make the editor more versatile. The second aspect are the system requirements of the editor. The editors might require different plug-ins or additional components or some browser features that not all browsers support. For example, Yahoo Pipes[12] works in a standard browser, while the Microsoft Popfly[13] required the Microsoft Silverlight plug-in until the service was shut down in 2009.

The runtime environment of the editor has two functions. First of all, it provides an environment to test and run the mashups, and second of all, it determines how the mashup is delivered to the users. The runtime environment that is created by the editor can be broken down into four crucial parameters. These parameters are the location of the runtime, the style it is deployed, the runtime's system requirements and how well the runtime scales. The runtime location can be either on the server side or the client side. It determines whether the mashup is run in the user's browser or in the server, while the browser just displays the results. The deployment style tells whether the developer can deploy the mashup on a server of the developers own or whether the mashup must or can be deployed on a server that is offered by the owners of the mashup editor. The system requirements, as in the editor's system requirements, tell whether users of the

[12] http://pipes.yahoo.com/pipes/

[13] http://en.wikipedia.org/wiki/Microsoft_Popfly

final mashup need just a standard browser, or if they need to employ some plug-ins or extensions. The fourth parameter is the scalability of the system. The mashup can scale in a few dimensions. The factors that can scale, are the number of data sources, the number of the compositions and the number of users. Scalability is generally a problem only on the server side mashups, because the client side mashups are always executed in the user's browser thus providing no problems to the server, except for possibly overloading the data sources. But if this is the case, it's not the mashup that has problems with the scalability, but rather the data source.

Manual development

Manual mashup development on the other hand takes more time and effort, but lets the developers keep all the strings in their own hands. When the developer team decides to develop the mashup manually from the scratch, there are a few things that should be kept in mind.

First of all, how can the data sources be used. By the definition of a mashup, it always uses some service that it combines with others. The data source might have a programming API that allows the developers to tap into the system easily. If there is no API, the data has to be extracted by mining it. The content can be in one of several formats. Usually the content is in RSS or Atom feeds, XML or HTML markup or Flash or other graphical formats.

When the developers have the content, there needs to be a way to handle the data. This is usually done with different APIs that are provided in some components. This provides the functionality to the mashup. APIs are usually done in some programming language, for example JavaScript is a popular language for APIs.

When the developers have the content and the functionality for the mashup, they are missing the final key ingredient to a mashup, the presentation. The presentation elements usually come in various graphical formats, such as the aforementioned Flash format. The presentation can contain previously available parts, such as a map view from another service or it can be completely custom-made.

Now that the developers have the three key components, the content, the functionality and the presentation, it all needs to be put together to create the mashup. The mashup can either be made on the server side or on the client side. If the mashup is made to work on the client side, or in the web browser, it is often made with a programming language, such as JavaScript, while server side applications are done usually with scripting languages, such as PHP or Ruby. Mashups can also be made using more traditional programming languages, such as Java or C#.

3.5 Web Business models

According to Michael Rappa [Rappa, 2010] there are nine different basic categories for business models on the web. These categories are in alphabetical order: advertising, affiliate, brokerage, community, infomediary, manufacturer, merchant, subscription and utility. The business model for a web service can comprise of one of these or many of these categories. These business models can also be extended to mashups to create revenue from the mashup.

The advertising model is familiar from earlier medias than the web, and really is an extension of earlier advertising. If the website is to create money from advertising, the website needs to have a high amount of visitors, since the advertisers usually pay by the numbers of how many people have seen the ad. The way the advertising model can be cultivated is by providing more accurate advertising. It's important to the advertiser, that the right people see the right ads. The logic is based on user accepting to view ads in exchange for getting to use a service provided by the website or program. Rappa offers several different examples for ways of using ads ranging from the traditional banner ads to ultramercials. There are a few interesting ad types, that will be gone through here. Intromercials are ads that fill the user's entire screen when navigating to the website that either leave the screen after a period of time or require some feedback to continue to the intended content. Ultramercials on the other hand can occur anytime on the site, and require feedback to leave the screen. Ads can also be placed on the screen contextually. If this is the case, the site observes the user's actions and chooses the most suitable ads accordingly. This is done especially in search engines by choosing ads that suit the user's search terms. A nice way of looking at ads is also to set up a classified ads website. In this model the user's place ads on the site and pay a fee for it, and in exchange receive visibility for the stuff they wish to sell. Aggressive advertising has lead to the emergence of adblockers that ease the use of the internet by blocking adverts altogether. An example of these is AdBlock Plus[14].

Affiliating model is a way of getting visibility by placing ads on another site, and offering the affiliates incentives from generating traffic. Affiliation can be done as a sort of exchange, a number of websites can choose to affiliate each other, thus advertising for free on each other's websites. This is traditionally done by exchanging banners and placing them on all the affiliate sites. When the sites use banner exchange they are trying to create more traffic on both sites, and not focusing on getting paid from every click of their banner. The site then collects the revenue from other sources. This model can also be turned into including the exchange of money. The affiliates can pay each other with a scheme of “pay-per-click” so that the other site is paid when they generate traffic to the other site. Another option for this is sharing revenue, when the affiliate has brought in the customer. This way the affiliate receives a portion of the sales, but nothing from bringing customers that don't buy anything.

Brokerage means that the website creates a marketplace and takes a fee from the transactions that occur in it. The site can be a broker in several types of different transaction models. The marketplace can be a place for auctioning, transactions themselves or even distributing. Perhaps the most known brokers are eBay and PayPal, of which eBay allows users to auction away merchandise and PayPal provides a safe way to easily do transactions between users from different countries. It's also important to remember, that the brokers don't need to be only from the business to the customer, but brokers can be in business to business markets and customer to customer markets.

The community business model is a very familiar model to social networks. The idea in the community business model comes from Metcalfe's Law in which the value of the network is the square of the number of it's users. Simply relying on this law has destroyed many attempted internet services, since the revenues don't actually grow that fast. [Briscoe et. al, 2006] After the burst of the dotcom bubble, the community business

[14] http://adblockplus.org/en/

model is seen still as a strong model, but not nearly as good as before. The community model focuses on gaining the loalty of the users. When the users are loyal to the service, they are emotionally attached to the product, and that's where the revenue can be made. The revenue can come from selling the users merchandise that promotes the service, or by offering better service in exchange of a payment, or just by collecting donations from the users to keep the service up and running. If the service wishes to use the community in another way, the community is an excellent place to advertise efficiently. Open Source development is based on the community business model, where all the users create programs or parts of programs and share them for others to use. This relies on the user's contribution, and when the users are active, it thrives. Also Wikipedia[15] uses the community model to keep the service going. Firstly, it allows the community to edit the entries in wikipedia, thus creating the content, and also keeps the financial side of wikipedia maintained by asking for donations from users. The users of wikipedia are glad to pay for the service not getting ads placed to it. Social networking sites use the community business model to to sell targeted adverts to the advertisers. These ads can be well targeted, because social networking sites include well detailed profiles of the users, and their activities in the service.

The term infomediary comes from information intermediary. What this means is that some businesses get their revenue from selling information about consumers to the sellers or vice versa. This information is particularly useful for consumers when they are thinking about purchasing something valuable, and want the best value for their money. The sellers benefit from this information most when they are planning a new ad campaign and wish to get their message to the right customers. The infomediary model has been used in a few different ways. Firstly, there are ad networks. These ad networks place the advertisers ads on sites in the network that they believe to have the most suitable audience. The ad networks collect data on the shown ads, and thus improve their services accordingly. Another way of collecting data about users is to offer incentives to users. These incentives can be points the user can spend on a store, or a discount on a certain product. The incentives will be offered to users who give the company information about them. This information is then sold to companies, who wish to do targeted marketing. A metamediary service is the most abstract type of infomediary service. It provides information for both the consumers and the sellers about each other, while never taking part in the actual transactions between the parties.

The manufacturer model is a model, where the manufacturer of a product or a service sells the product or the service online. The manufacturer thus saves money by skipping the distributor channel out of the process of selling the goods. In this model the company tries to either sell, lease or license their products directly to the consumers. Especially popular in software business is licensing software, since it creates future business when the license expires. An approach to the manufacturing model is also creating content that utilizes product placement. In this approach, the product placement isn't considered advertising, since the product placement happens on the company's own website.

The merchant model is perhaps the most traditional approach to doing business on the web. In this approach, the seller has an online store, that the users use to buy products from the vendor. The merchants can be divided into a few groups. There are those shops

[15] http://www.wikipedia.org

that are completely virtual stores, in which the user orders the product directly from the merchant via the Internet. The merchant can also use a catalog approach to selling it's products. In this, the catalog is placed on the web page, while the orders are done via telephone or email. Another approach in merchanting is for stores that have a physical store already. This is called the “click-and-mortar”. In this, the traditional brick-and-mortar store has it's own web storefront. Finally, the store can be focused on selling purely digital products, such as downloadable movies or music.

The subscription model funds itself by having a periodic fee for using the service. This can vary from being a so called freemium model, in which the users get to use the basic service for free, but have to pay a price, to use premium features, to having to pay the fee just to get to use the service. The subscription fees are always billed, no matter how rarely the user uses the service. It is also not uncommon to use the subscription model with the advertising model. Services that use subscription fees can vary from the internet operators, who offer internet connectivity to users for a periodic fee to content services, that for example host the user's files for a fee. Other types of subscription models are for example employed by Spotify[16], which combines the advertising model with the subscription model. In Spotify, which allows user's to listen to all sorts of music without a fee, the free users have to listen to ads between the songs, while the premium users pay a price for not having to listen to the ads.

The utility model bills users on the basis of usage of the service. This is a business model that has been traditionally employed in services, in which the usage is easy to meter. For example, telephone companies and electric companies have used billing, that is based on metering. As a model for web business, it is used in a way, that the users buy themselves for example credits in an online game with real money, and use it, and buy more, when they need more.

[16] http://www.spotify.com

4 Overview of the targeted mash-up system

AmiCarta is a service that allows users to locate each other using a web-based map and to chat with each other via the service. The service is based on push-marketing. Users allow the service to send them x adverts per day via SMS in exchange for the free use of all the features. These adverts are sent to users who have indicated that they are interested in the area of business the company works in. For example, if the user is to indicate that he is interested in home electronics, he would receive adverts from a nearby store that sells home appliances. AmiCarta can be classified as a P3-system, a system that links People-to-People-to-Geographical-Place. [Jones et al., 2004]

AmiCarta is a mashup. What other parts of social media also fit into AmiCarta? It's a social web application, because it connects users with each other. As a social network, the service is more comprehensive in sharing the user's status information. Location information is the key to more accurate status information.

The key components of AmiCarta are location information, social network, communication and meeting new people.

Social media can be easily used for commercial purposes when promoting a new product. This is called social media optimization [Bhargava, 2006]. The five rules created by Bhargava are increasing linkability, enabling tagging and bookmarking, rewarding inbound links, enabling content travelling and encouraging mashupping.

So, how can this social media optimization actually be done? Tagging and bookmarking are done easily by inserting so called sharing buttons on the product's web pages. The users can then click on the sharing button to easily show the page to their friends. Bookmarking is also easily done by inserting a bookmarking link on the web page. This button creates a bookmark to the user's browser or bookmarking service.

4.1 Features

AmiCarta provides real-time information of your buddies' locations by portraying them on a map. The service supports both GPS positioning and network positioning.

AmiCarta supports so-called "geofencing" which allows users to place areas on the map and define the areas to portray status information about themselves while they are moving. For example, one can place a geofence around his/her home and name the area "home". This information will then be triggered to his/her status information, when the user is located inside the geofence. There are no limits to the size of the geofence. Overlapping geofences are also supported.

The system allows the users to send instant messages (IM) to each other while both users are online. This way users can communicate fast and eg. set up meetings.

AmiCarta saves the users location history, so that the user can afterwards check his/her movement routes for any period of time. This is referred to as history information. The tracking service can be turned on and off at any time by a simple procedure in order to give the users control over their privacy..

AmiCarta allows users to set up points-of-interests (POIs), that can be shown only to themselves or also to their friends. These POIs are surrounded by geofences that trigger when the user or one of his friends enters the area. This can be used to advantage eg. at companies that have lots of customer meetings. The user can specify if he doesn't want to receive these POIs on his map.

The system allows the user to create an extensive profile of himself. This profile can be partly public and partly only shared to his friends. The profile contains pictures of the user, the user's interests. Perhaps even other users can send messages to the user's profile similar to the wall at Facebook. The profile info is used to target the user with benefits, such as discounts, offered by the service sponsors.

All functionality of the service takes place on layers above the map-view.

4.2 User interface description

The user interface is centered on the map that displays the user and his friends. The idea behind the user interface is that the user never leaves the map view, but instead information is brought on top of the map to allow the user to quickly return to the map when required. Another source of inspiration for the UI was to try and make the UI use as little menus as possible and to show as much information as possible in one view. This way not many features have to be hidden from the users.

Although the UI tries not to hide important features, it is still important not to show all the features on-screen at all times, as this would confuse the user easily. This is why a tabbed approach was the one that the team went with. Clearly named tabs can easily make a good interface, if the names are carefully selected. Rationally thinking, it shouldn't be that easy to get a good name that applies to all the actions available in a UI view.

The user interface is created using OpenLayers[17] on top of the openstreetmap maps. OpenLayers offers a way of displaying different kinds of information on top of the map view.

4.3 Business model

AmiCarta uses a so-called “freemium” business model. In this model, the users are categorized into free users and premium users. The users using the free version of the product have only limited functionality of the product or the free users are funded with advertising. The premium version has full functionality, with the catch that the user has to pay a price for the whole product.

The idea behind freemium is to create a large demand for the product, whilst creating on the side a larger demand for the additional services. Freemium works with digital products, because the core product can be duplicated for a cost of next to nothing. In a service this is even easier, because nothing needs to be supplied to the user, other than a user account and possibly a downloadable client program.

In addition to the users, the AmiCarta business model includes external developers. The aim is to offer the developers comprehensive tools to create new content to the service. The tools will allow the developers to use the data that is collected of the users

[17] http://openlayers.org

and to create new applications that are attached to AmiCarta. This way the service gets more value because of the new content, and the developers get visibility by adding their applications to AmiCarta.

4.3.1 AmiCarta free-of-charge

The free version of AmiCarta includes basic functionality of the service. The user can have an unlimited amount of friends, the user can use instant messaging to communicate with other users as well as use anonymous instant messaging. There are no limitations to the usage of the service. The anonymous instant messaging includes the chance for the users to send an ad about themselves to potential partners, that aren't using the service at the moment. These ads will cost the user an amount of money, in the ball park of one euro per message.

The main disadvantage for regular users of AmiCarta will be the fact, their usage will be paid for by advertising. The users will receive targeted SMS or MMS ads to their mobile phones. These ads will be based on the user's location and profile, which means that they are well targeted.

4.3.2 AmiCarta premium

AmiCarta premium costs the user a certain amount of money to use. The advantage of the premium version is that the premium members don't receive ads. Another advantage the premium users get is that they can send a certain amount of AIM ads per month for free. At this time, there is no estimate on how the ratio of free users and premium users will be divided. This depends on how well the advertising scheme will work on not annoying users.

4.3.3 AmiCarta for developers

The developer version of AmiCarta will include a programmable API to create own mashups of the system. This enables the early adopters of AmiCarta to create content for AmiCarta to suit their needs and to benefit the system with new content.

The programmable API will contain the functions of AmiCarta and a description of how to use the functions. Other than this there is still no plan on the developer interface due to the concept being in validation phase. Because of this the developers do not know what to add to the developer API.

4.4 Enabling technologies

The AmiCarta system consists of three different entities. The first entity is the client's side of the system. This includes the device that is tracked, via network positioning or GPS positioning. The second entity is the openstreetmap server. Openstreetmap offers the possibility to extract the map data in the openstreetmap servers to create an own map server for the system. The third entity comes from the servers that control the logic of the service and are created by Indagon. The system is depicted in figure 2.

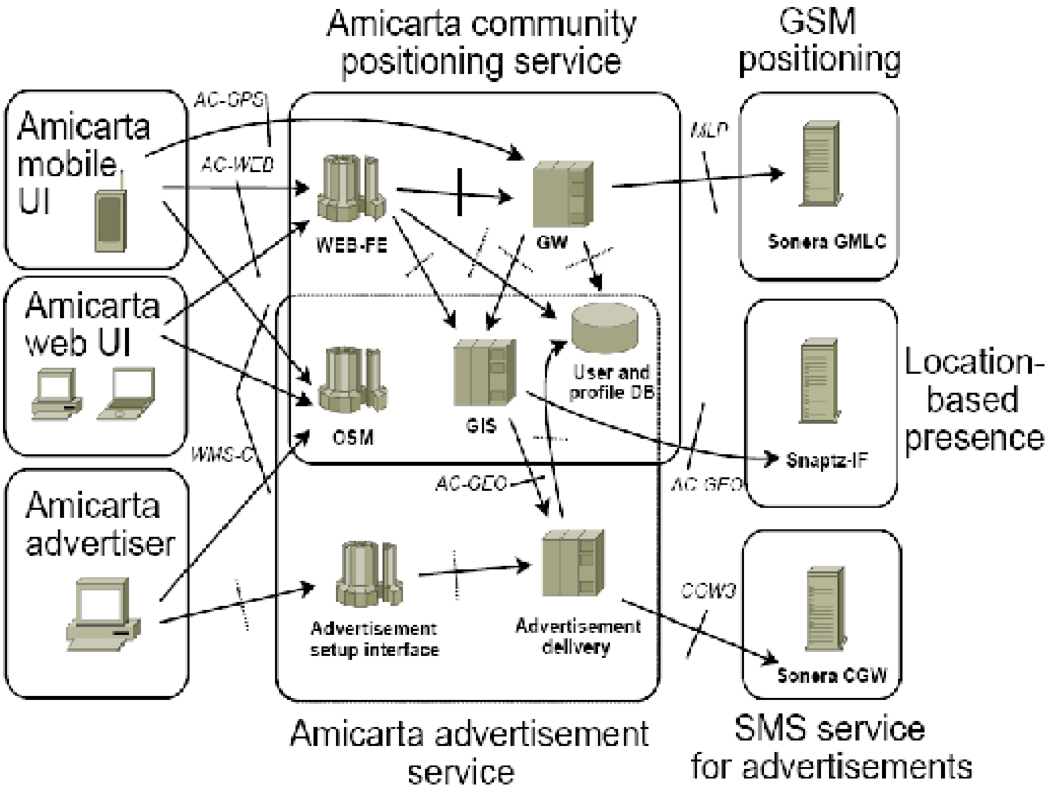


Figure 2: The AmiCarta system architecture

In figure 2 there are lots of abbreviations. In the abbreviations, AC always stands for AmiCarta. In the figure, AC-GPS stands for the interface in the system that allows the mobile client to send it's location information to the service. AC-WEB is the interface between the user interface and the servers in the system. WMS-C is a standard interface for the map tiles to be transferred to the user's screen. WEB-FE stands for the web front end, that is the logic between the databases and the user's interface. GW is Indagon's gateway server that does the most work in the system, it acts as the operator between the databases and the user interface. GIS is the database that controls the location information in the system. OSM stands for the map server, short for OpenStreetMap. AC-GEO is a protocol for area information. It provides information to the system, when the user's terminal moves from one area to another. CGW3 is the operator's SMS interface and CGW is the operatior's SMS service. Snaptz-IF is a service that receives the user's area information. GMLC is the operator's network positioning interface for tracking the user with operator-based network positioning.

The user can be tracked via mobile phones or other devices in GSM or 3G networks. The mobile device can be tracked with network positioning and with a client in the mobile device via GPS positioning. The network positioning can be implemented in two ways, the operator can directly give the system the user's position, or the positioning can be done with a client that is downloaded to the user's phone. Operator-based positioning is a much easier method of implementation, because it doesn't require a client. Developing the client takes time and money, and creates problems with compatibility, because each operating system in a mobile phone needs it's own version of the client. With operator-based positioning all of this can be avoided, and there are no compatibility issues, since the operator tracks all of it's clients automatically. Operator-based positioning also enables tracking older devices, that don't support the generated clients. It is vital for the system to have operator-based positioning to have the possibility of including as many people as possible to the system. GPS-based tracking always requires a client.

The map base for the system comes from the openstreetmap organization. Openstreetmap offers user-generated maps, that can be used under the creative commons 2.0 license [Creative Commons, 2010]. The map data is copied to an internal server to ensure that the system loads fast, because the openstreetmap central server is in Great Britain and can be under a lot of stress. Locating the AmiCarta map server near the service will speed up the system significantly.

The servers that control the logic of the service consist of a few servers. The system needs different servers for the web front end, that shows the content to the user, a gateway server that is the brains of the operation and communicates with the other servers, a GIS server that stores locations, a database for the users' profile information and an advertisement server that delivers the ads to the users.

The web front end server consists of the user interface to the user and handles the question interface for getting the content in the user interface. The web front end asks the gateway server for the users' locations, to the GIS server for saving the points of interest and to the user database to confirm the users' identities and to receive profile information about the users.

The gateway server is the brains of the operation. The gateway communicates with several different servers. In order to use the operator-based network positioning, the gateway communicates with the operator's GMLC server. The gateway also communicates with the user database to authenticate the users identities. The gateway server also communicates with the GIS server to trigger events in the system, when the user's location changes.

The GIS server is the active database for the location information. The main functions of the GIS server is receiving the user's locations and defined areas in the system and to report changes in locations to the other components in the service.

The user database contains the information stored about the user. This includes profile information such as the user's name and contact information and other basic information. The user database also includes the user's friends and the users preferences.

The advertising engine in the system sends the users targeted adverts that pay for the user's usage of the service. The engine combines the user's location with the user's interests that are located in the user's profile. This way the service sends the user ads when the user is at a suitable location near the place of business when the business is open.

5 Concept evolution

This chapter explains the evolution of the concept during the validation process. It starts from the concept prior validation and explains the improvements to the concept as well as new features suggested by the users. This chapter also explains the progress of the validation starting from the test plan.

5.1 Concept prior validation

The system had some features prior validation that were thought to be useful to the product. These features were implemented to the validation mock-up.

Registration

The registration to the system is straight-forward. The user is required to fill in personal data and other profile information. Once the user has filled all the forms, the user receives a link to his/her e-mail address asking to confirm the address. Once the address has been confirmed, the account is ready to be used.

Registration is a necessary part of AmiCarta, because it distinguishes the users from each other and allows the system to store the user's personal data, including the users' friends and location. For a service that is designed to be personalized, registration has to be done one way or another.

Login

Logging in to the service is done by visiting the AmiCarta site. The user is required to type the user's e-mail address and the user's password. After the user has done this, the service opens and is ready to be used.

Basic map functionality

Once in the service, the user is presented with a map. This map shows the user's current location as well as the location of the user's friends. Additionally the user can configure the map to display different points of interest, routes to different places etc.

The user can manipulate the map by zooming in and zooming out as well as moving the view of the map to a different target. The action is performed by using the mouse. The mouse wheel zooms in and out, while the user can drag the map by clicking it with the left mouse button and moving the mouse to the opposite direction to what the user wishes to view. The map controls are done this way because it is the de facto standard OpenStreetMap configuration.

Buddy list tools

The buddy list displays the user's friends and data about the friends. The default view of the list shows the user's friends avatar images, the friends' names and their latest status messages. The user can click on an icon to center the AmiCarta map view to display the friend's location.

The buddy list also has an icon to view the friend's profile. This icon opens the friend's profile in it's full size on the screen.

The third icon with an action in the buddy list is an icon that starts a chat between the users. The chat opens in a small window above the map view in the web user interface and to it's own tab in the mobile user interface.

Finding new friends is done via the buddy list. The buddy list has a large icon that starts the process once clicked. Users can search their friends by using different parameters. These parameters are the e-mail addresses and the first and last name of the users.

Profile tools

The user's profile contains a wide array of information about the user. The information can be pictures, text and audio information as well as geoblogged location information. Location information can also be added to the user's pictures, text and audio.

The profile tools include a message box that saves messages left to the user since last visiting. These messages can also be forwarded to the user's e-mail address. The message box can also be set up to save the chat's between the users.

Each user of AmiCarta has a profile home page that shows their recent activities. The page contains the user's latest status messages, the location where the status was updated, the user's friends' latest status messages and other relevant information.

Privacy tools

An important aspect in keeping the users satisfied with the offered service is to give the users control of their privacy. If the service were to reveal too much about the user, the users wouldn't trust it.

The user has control over everything privacy-related. The user can control which part of the user's profile are visible to other users. Different settings exist for the user's friends, users who are not the user's friends, but exist in the system.

Especially important is the privacy setting of the user's location. This is because by showing the location of a single user, the user can be monitored extensively, especially if the user uses GPS positioning, that gives the location with an error margin of merely meters.

Adverts for AmiCarta

AmiCarta is to be free-of-charge to the user, but still the makers of AmiCarta need revenue to keep the service running. This is where adverts come in. The users choose to accept advertising to their mobile phones, when using AmiCarta. These adverts are very targeted ads, that use the user's location information as well as the user's profile information to be targeted accurately to the right users.

The aim of the adverts is to give the users information on products that they really are interested in. This way the ads go in the right address, and both the users and the advertisers are satisfied.

The process of choosing which ads end up on which users' screens is hidden from the users. The system uses only the data available to it in the users' profiles and location information.

Point-of-interest tools

AmiCarta supports so called points of interest, in short POIs. These POIs can either be created by the user or the system administrators. POIs can be used to include location information into different types of information. For example, advertisers can place POI-ads on the map to locate the vendors for different products. Another example would be users creating their own POIs. This way they could for example, when having a party, place a POI on the location of the party on the map and share the POI with the user's friends.

POIs can have a so called geofence attached to them. This geofence provides functionality to the POI. The geofence can alert the user when someone enters or leaves the geofence, or is inside or outside the geofence for a determined period of time.

The tools offered to the user to create POIs are simple. The user can create a POI straight on the map by right-clicking the map and choosing from the menu the option to create a POI. As an alternative to this, the user can choose from AmiCarta's main menu the option to create a POI. If the user started from the main menu, AmiCarta will ask the user to place the POI on the map. After the POI has been placed on the map, a dialog box opens. The dialog asks the user to fill in a name and a description for the POI. At this point, the user also specifies to whom other users the POI is to be shared with. Also the user is asked if the POI should include a geofence. The creation of a geofence is handled in the chapter “geofencing tools”.

Once the user has filled in the information required by the dialog, the POI appears on the user's map view and also on the map view of those users the user has chosen to share the POI with.

Geofencing tools

Geofences are a part of POIs. Geofences are shapes formed around POIs that cause alerts to the user who has created them. Once the user has created a POI, the application offers the user the possibility of having a geofence around the POI, if the user chooses to add a geofence, the application opens drawing tools for the user. The user then draws an area on the map depicting the desired geofence. Once the user is done with the drawing, the user clicks ok, and the geofence is saved. Next the user must define what action is taken when users either leave or enter the geofence.

History tools

AmiCarta saves the user's location information on it's server, unless the user turns positioning off. This information is used to display the user's location history, should the user choose to view it. The user can also specify that the user's history data is public to the user's friends. The history data cannot be made public to all users.

History data is displayed on the map as a line, that shows where the user has been at what time. The line shows all the points of actual data that exists and extrapolates a line between the points in the order in which the user has been at the locations. The history data of multiple users can be viewed at once, the data will be represented with different colors for different users to distinguish the users from one another.

Users will be limited to only viewing the history data of the last 24 hours. The user will choose the interval from which the history data is wanted for by using a slider that chooses the start point and the end point of the history. This history data cannot be exported to a file, but the user can save the history on the server for further review. This way the user can access history that is older than 24 hours.

Widgets for linking to other social media

In order to lure in the users, AmiCarta must be able to adapt to having a strong competition in social media. Since belonging to one service doesn't restrict users from belonging to others, a good strategy is to provide the users with additional content to their existing services. This way AmiCarta will get free visibility and hopefully lure in new users.

The biggest social media service at the end of 2009 was Facebook with over 400 million active users. Since it is logical to go to the potential users, Facebook seems like a likely place to start from. What can AmiCarta offer to the users of Facebook? The one advantage that AmiCarta has, the user's location information. When other users see their friends publishing their own location, they might just be intrigued to get all their friends locations in one package at AmiCarta.

The idea for the widgets is simple. AmiCarta will provide a simple way to embed the user's own location information to the media of choice for the user. The widget will be a small image that shows the user's location on a map, and tells the viewer, that AmiCarta was behind this information. If the viewer of the information clicks on the widget, the viewer will be directed to the AmiCarta homepage to create their own account.

Anonymous instant messaging

Anonymous instant messaging, in short AIM, is a mode of messaging where the users don't know the conversational partner before-hand. The idea behind this AIM is that the system uses the users' profiles and location information to match conversational partners easily. So in a sense the system tries to find optimal conversational partners for the users. An example of an unsophisticated AIM would be the service Chatroulette[18], in which the users get to chat with random strangers. This service doesn't include any kind of profile matching.

The AIM system is easy to use. The user starts the process by choosing AIM from the AmiCarta main menu. At this point, the AIM system goes through the online users and chooses the best four matches for the user. The AIM system provides some basic information of the users, like age and sex, and then lets the user choose from these best candidates.

[18] http://www.chatroulette.com

Once the user has chosen the conversational partner, the instant messaging begins. After the users have stopped messaging, the system will ask both partners, whether they would like to be able to contact each other again. If both users agree that they would like to stay in touch, they are added to each others contact list. This way they can contact each other. If one or both of the users choose not to want to be in contact with the other, a message will be displayed to the users, that they weren't added to each others contact lists.

Local news

Local news as an idea requires location information to the news. AmiCarta system uses OpenLayers to function, and OpenLayers supports so called GeoRSS. With GeoRSS news providers can implement location coordinates to their news, allowing easy placement on a map. This way the system could support news on the map and on a separate news feed view.

The news feed view would display the news so that the nearest news would be displayed first. This way the users would get local news everywhere in the service's operating area. The news would obviously need a period of time that they are displayed on the service. Depending on the number of news inserted in to the system per hour, this time could vary. If the time interval is too small, most news are missed by the users, and if it is too long, heavy users will get bored of reading the same news all the time.

While categorizing the news by distance to user, the users also expressed a wish to be able to categorize the news by topic. Not all users wish to receive eg. sports news.

Shared calendars

The idea of calendar sharing sounds a bit too revealing. The idea is not to share the entire calendar with other users, but to be able to make entries that can be shared. This would enable easy party planning, organizing meetings and other social events. The entries could include the location of the meeting displayed on the users' maps as well as a list of those invited to the event.

As well as the location, the users need to define the date and time for the event. When sharing the message, they should include a possibility for the invited people to respond easily whether they are coming or not. This way the organizer also gets the information and can check it regularly from one place.

This feature includes the risk of crowding the user's calendar. That is why the participants must have a possibility to delete events that they do not wish to attend from their calendars. This way the users feel in control of the application.

User developer platform

In order to offer users constantly new content, AmiCarta offers a user development platform. This way anyone can develop content for AmiCarta. Judging from the amount of applications developed for Facebook, over 350 000 applications to date [Facebook, 2009], there is a real need for developers to be able to tap into social networks. The AmiCarta user development platform offers developers with an interface to AmiCarta's functions allowing them to use AmiCarta's data for their applications.

5.2 Concept validation

Validating a concept is a crucial part in developing a new product. The validation process gives the developing team valuable insight in what the potential users of the product think of the product.

5.2.1 User groups

The validation will focus on the early adopters of the AmiCarta concept. It is important to get the early adopters excited about the concept, so that the service has actual potential to grow. If the early adopters are scared away from the service, the service will never reach the masses.

Since the service is a hybrid of two current services (Facebook and Twitter) with it's own twist, the potential early adopters of AmiCarta should be quite similar to the early adopters of Facebook and Twitter.

The very early adopters of Facebook were the students at Harvard University, because the service was limited to only them. From this you could conclude that the early adopters are most likely grounded to their computer skills and to the fact that they have a community that adopts new ideas quickly.

Twitter's most active early adopters are people over 35 years old [MarketingVOX, 2008]. This is an interesting fact, since traditionally it is considered that young people are the fastest to adopt new ideas. This would suggest that age is not a big factor in the potential early adopter user group.

Due to the mobile nature of AmiCarta, it is reasonable to assume that the early adopters of the product will have to have flat-rate connections on their mobile phones, or have to have free mobile phone usage altogether. Also a relatively new mobile phone is required at least at beta stages, because of the requirement of GPS positioning while mobile network positioning is not implemented.

In addition to the users having a flat-rate connection in their mobile phone, they should have experience with computers and social media to be considered as possible early adopters for AmiCarta.

So in conclusion, in order to be considered a possible early adopter of AmiCarta, the user needs to have a) some experience in computers, b) a community around them that would be interested in location information, c) a flat rate data connection in their mobile phone and d) a smart phone.

5.2.2 Validation methods

Validation of the concept will be done using a combination of the wizard-of-oz methodology, rapid prototyping and paper prototyping. The key to validation is a wizard-of-oz paper prototype that is evolved rapidly in between testing sessions.

The wizard-of-oz prototype is made in a similar manner to paper in screen prototyping [Bolchini, 2009]. The difference to this method is that the images showed on the screen are not hand-drawn, but made with the computer to allow them to be modified easily.

The mock-up prototype will be tested with giving the user tasks, that the user needs to find how to accomplish. This way all the features of the prototype will be covered. In actuality, it plays an insignificant role on behalf of the concept, how fast the users complete the tasks. In regard to the user interface, it is very significant though.

After the user has completed the tasks, an interview will be conducted, in which the user will be asked questions about the concept. The questions include questions about the concept, new ideas for the concept and a few questions about the test itself.

At the end of the session, the user will be asked to fill a questionnaire in which the user will give a grade on a likert scale of 1 to 5 to each feature in the concept to rank the best ideas in the concept.

Once the user tests have been concluded and analyzed, there will be held a focus group session, with the AmiCarta team in which the results of the study, and the new features that were suggested, will be gone through, and the service will be broken down into different types of clusters. These different clusters will be analyzed and used to create a road map for the service.

5.2.3 Test plan

The concept will be validated using a wizard of oz prototype of the release version of the product. The great thing with wizard-of-oz prototypes is that they don't require the prototype to actually work, but just look like they work.

The prototype focuses on the mobile phone user interface of the product. In order to have the product succeed commercially, we need the users to use the product a lot. The best way to do this, is to get the mobile phone client easy to use, because everyone carries their mobile phones with them. Now if the users have the potential to use the product all the time, it is more likely that they will use it at least sometimes.

The prototype portrays the mobile user interface for Amicarta using touch-screen phones. The mobile user interface was chosen to be used because it is a bit simpler to produce, because the screen resolution is limited to only 640*360 pixels, while a web interface would have to be at least 1024*768 pixels large.

The testing will focus on a small group of early adopters. In order to get the product to appeal to the mass markets, the early adopters have to be lured in first. Who are considered early adopters in social media? The best way to look at this, is to look at who used similar popular services first. The product being developed here is a hybrid between facebook and twitter while adding it's own with positioning technology.

The validation will be done with one early adopter at a time. The method will include a wizard-of-oz prototype with use scenarios and an interview at the end. This means that the setup is based on contextual inquiry, in which there is an observer looking at the user working. The results from the validation session will then be applied immediately and the prototype will be changed accordingly for the next session. This will include some rapid prototyping in to the validation process. This allows the designers to immediately test the quality of the suggested improvements.

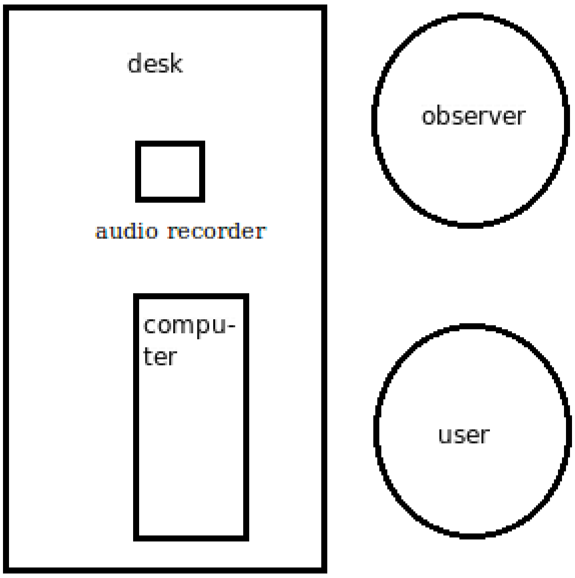


Figure 3: Test setup

The validation will take place in a room, which contains a computer, two chairs and an audio recorder. The user will be sitting in front of the computer and the observer next to the user. The test setup is pictured in figure 3.

The user will not be given any tasks to perform with the mock-up prototype, because the focus of the study is on the concept, not on the user interface. The test subject is encouraged to search through the menus to find the features. If the user doesn't visit some of the features, the interviewer will ask the user to go through the feature. This shows of what the user is most interested in, by showing which menus the user goes through first.

The test will begin with the interviewer describing the AmiCarta system in detail. The user is told the basic ideas behind the concept. After the user has been told what the system is about, the user will be told to study the mobile UI to get a grasp of how things are done using the mobile UI. At this stage, the user is encouraged to think out loud, what the user is about to do and if there are some suggestions to make the UI better. When the user has gone through the mobile UI, the interviewer asks a series of questions about the features of the product. The interview questions are presented in appendix A. Once the questions are dealt with, the user is asked to rate the features of the product on a likert scale of one to five. This concludes the user test.

The user tests will be carried on for as long as the user gives the concept adequate reviews in the questionnaire. The criteria for this is that the user gives the system the maximum grade in the questionnaire's questions about the user interface and understanding the concept. At this point the process of interviewing users can be stopped. The criteria are listed as “understood the concept,” “the product seemed easy to use,” “the prototype was polished” and “the user interface was reasonable.” The questionnaire is presented in Appendix B, and the results of the questionnaire in Appendix E. Appendix E also presents the basic information of the users, such as sex, age and marital status.

The new ideas presented by the users will be evaluated by the developer team on whether they are possible to make or not. As many of these features as possible will be implemented to the product. If new large features arise of which the design team is not sure, should they be implemented or not, a new round of interviews will be organized.

The data from each user validation session will be analyzed with affinity analysis. Affinity analysis consists mainly of grouping the observations collected in the test to get a better picture of what kind of results were gathered in the tests. All the observations will be individually categorized to not be biased by previous categories. Surely some observations will go into the same categories, but this way not all. The categorization will be done again once all the interviews have been conducted.

Affinity analysis is used to get an out-of-the-box view of the concept. Categorizing problems and thoughts that have occurred during the tests, the researchers will find new features and concepts or use scenarios for the product.

The results of the tests will help in specifying how the evolutionary pathway of the product is formed. Especially interesting is how the users feel about the features of the product. Which features are most important to the users? This will certainly affect the order of implementation for the features.

With the test results, the designers will gather in a focus group session to determine what kind of different clusters can be formed from the gathered features. These clusters will be created on different criteria. For example, what kind of a cluster would have the most “wow-factor”, what kind of a cluster would be fastest to implement leading to a short time to market and for example, what kind of a cluster would cost the least to create.

Once these clusters have been evaluated with SWOT analysis, a decision will be made on which clusters are to be implemented, and what is the order. Also, there will be a road-map created to visualize the development.

5.2.4 User tasks

The user has no designated tasks the user needs to perform. The user is however required to go through all the views in the system. For that, there is a checklist that the interviewer will go through as the interview progresses, if the user doesn't find some function when going through the system self, the interviewer will ask the user to test it. The features that are implemented in to the prototype are the following: a map view, a friend list, the user's personal messages, anonymous instant messaging, a user profile, a shared calendar, traffic information, client settings, navigation, discounts, local news and user-generated POIs.

The check-list for all the features in the mock-up is presented in Appendix C. In the appendix, the list contains the features presented in the main menu and what the user can do once inside the menu feature.

5.3 Validation results

The results obtained by validation are displayed in this chapter. The results consist of the changes that were made to the mock-up that further help the UI to be more usable as well as what the users thought of the features of the concept.

5.3.1 Changes made to the validation mock-up after user tests

After each user test, the mock-up prototype was altered to correct inconsistencies found by the testers and to add small features that the users suggested. The changes that were made after each test are collected in this chapter.

Changes after test 1

The first test user had very few suggestions to the UI of the prototype. The user was obviously thinking about his own needs, when he suggested that to the system be added maps with cycling routes. This is an easy feature to accomplish since OpenStreetMap has it's very own map layer called “CycleMap” that includes roads specifically for cycling.

Privacy seemed like an issue to the user, because he suggested that for every geofence, that tracks other users than the one who created the geofence, the system must ask permission for from the other users. This seems like a nice feature, but at the same time, it seems a bit redundant, since the other user can see your movements at all times, so why does the user have to ask for permission to track the other user again and again? This feature was not implemented to the system, but left in the background to see if other users had similar concerns.

A purely user interface specific thing was the question if the main menu can be customized. This is an excellent feature to have, to allow users to hide services that they do not use or need. This would also support the developer API, since the user can add user generated content on the main menu easily. An easy implementation would be to offer a simple customization view in which the user can drag and drop the icons on the menu to change places. The view should include a button for new icons and a trash bin to which the user would drag and drop icons that the user doesn't need. This feature was presented to latter users verbally.

Changes after test 2

The second interviewed user was much more elaborate in what needed to be fixed before she considered the system usable. First off, the user felt that the main menu needed an indicator for scrolling to understand that there were icons off-screen also. This comes from de facto standards in computer design. This was implemented to the system before the next test.

This user was also extremely annoyed by the fact that there was a button for closing the program on every screen. She felt that she saw it so many times, that it just made her want to press it over and over again. That is an extremely undesirable feature for any program. This led to the removal of close button from other screens than the main menu and the map view. The close button was replaced with a help button that displays help to the user when pressed. The user also suggested that the close button is too equal to the other buttons, which is a good observation since the close button was aligned with the other buttons in a grid. So the close button was separated a bit from the rest of the buttons to make it harder to press accidentally. Also the menu icon for messages showed the total number of messages, it was corrected to display the number of unread messages. The old and the new main menus are represented in figure 4.

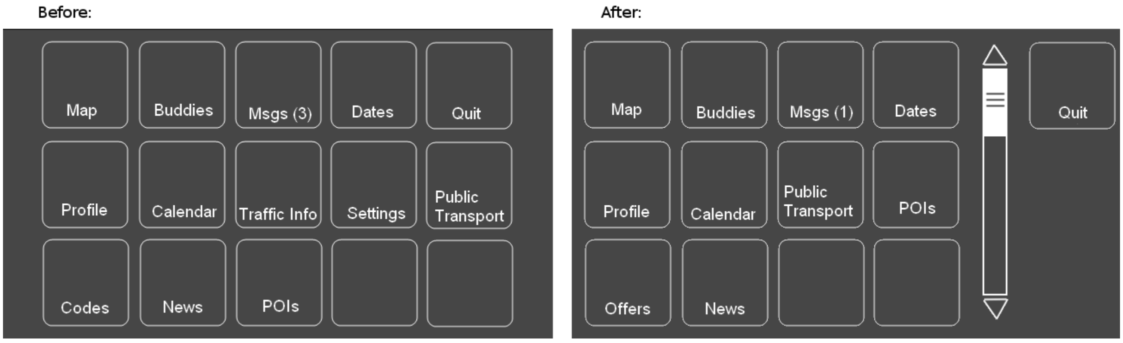


Figure 4: The main menu before and after the user's comments

The map view needed an indicator to show the user if new messages arrive either in IM, AIM or via normal messaging. This is a good point, so indicators were added to the mock-up as a result.

The user found it confusing that in the map view, she couldn't tell whether the user's friends were available for IM or were just being positioned by the service. This needed a change, so that available friends were highlighted. The coloring of friends was changed to green meaning a user that is available, red meaning that the user is just being positioned, and blue to indicate the user herself. The added indicator and coloring are displayed in figure 5.

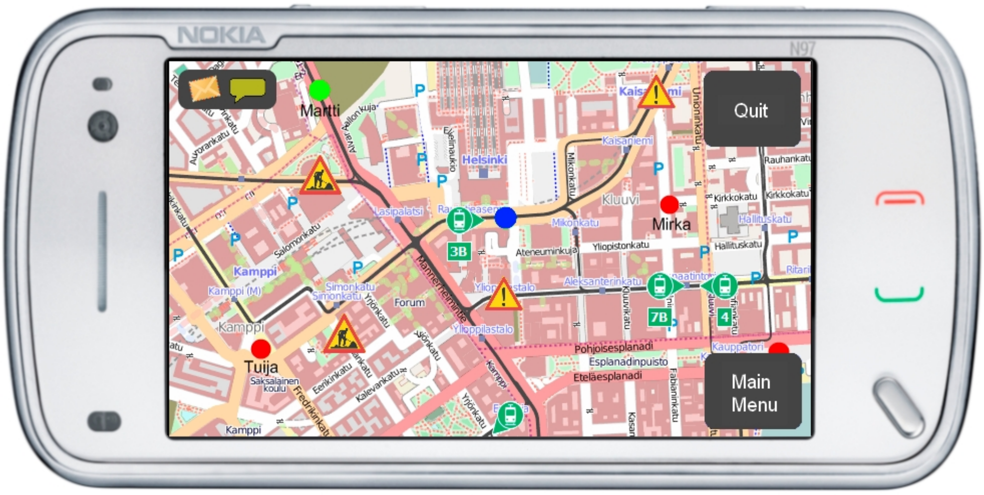


Figure 5: The improved map view, indicators for messaging added, the place of the menu button switched with the place of the close button and the coloring of users changed to indicate whether the user is online or not

For AIM the user had several propositions to make it more interesting. A few of these were additions to the user interface. For example, the user wanted the possibility for users to add a photo of them to the service that would be their chat avatar. This is a tricky question whether or not the service would still remain as anonymous as intended, but still is a feature worth thinking about. Another feature the user wished to access straight from the menu of candidates, was a short description of the candidate. Something that gives the user more motivation to engage in communication with the candidate. The user felt that the grade given by the system to show how well the user's profiles match, wasn't a good idea. So it was removed and a button to view a profile of the user was added. These changes are described in figure 6.

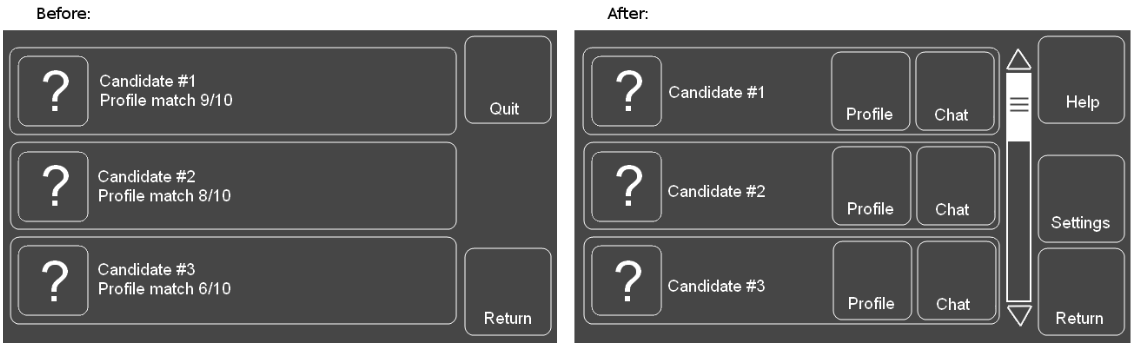


Figure 6: The old and new AIM menu

Before this test, the AIM process started at the AIM settings. The user thought that this was unnecessary, since the settings are only set up once, so the user doesn't need to see them every time. So the settings were moved to the candidate view only to be accessed if the user wishes to. This is shown in figure 5. Also, the user thought that having to save the settings was a waste of time, the system should do it automatically, so saving the settings was removed. The modified AIM settings can be seen in figure 7.

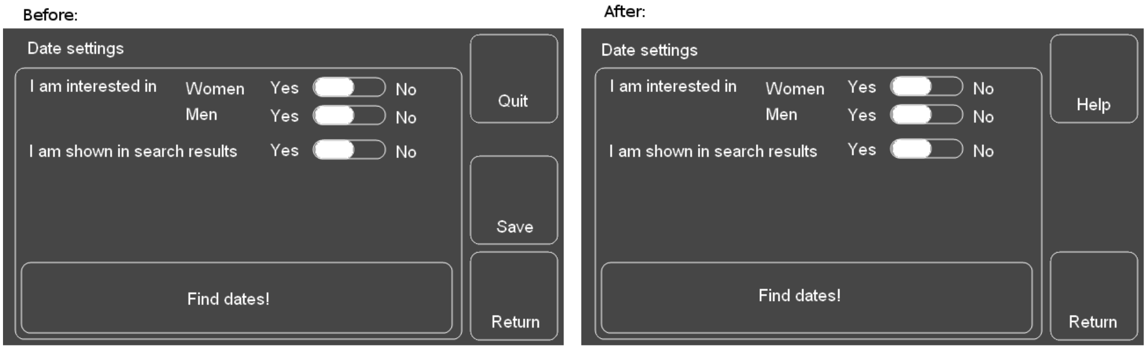


Figure 7: The AIM settings with saving removed and closing changed to help

The mock-up had it's own view for traffic announcements that listed the nearest announcements to the user at all times. The user felt that this kind of information was pointless, since the traffic announcements are displayed on the map view at all times. The user felt that the traffic announcements should only be displayed on the map, and as a list when planning a route. The traffic announcements menu was removed from the mock-up as can be seen in figure 4.

The original concept of advertising was to be done with advertising code-phrases that the user has to insert in to the system to get the discount offered by the advertisers. The interviewed user felt that this is too inconvenient, and the user should just be able to search through the adverts and not have to hunt codes. So the user suggested that there would be a menu that shows ads from the advertisers and the user can search through them whenever. The code feature was removed from the mock-up and replaced with the ad marketplace view. This is presented in figure 8.

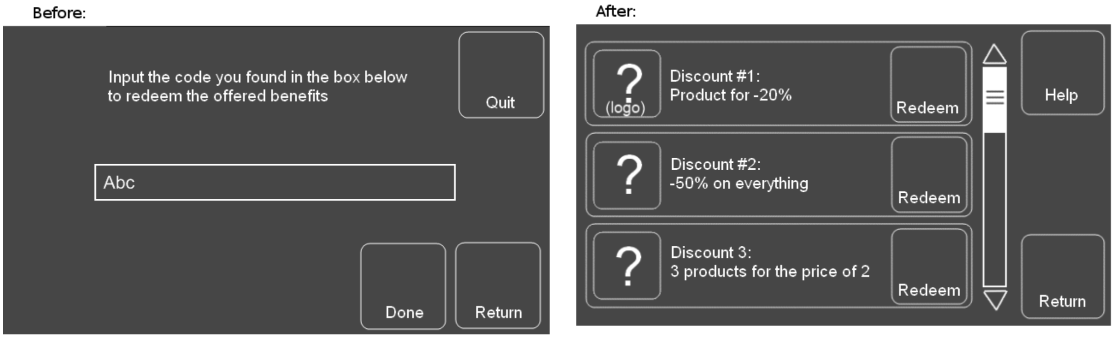
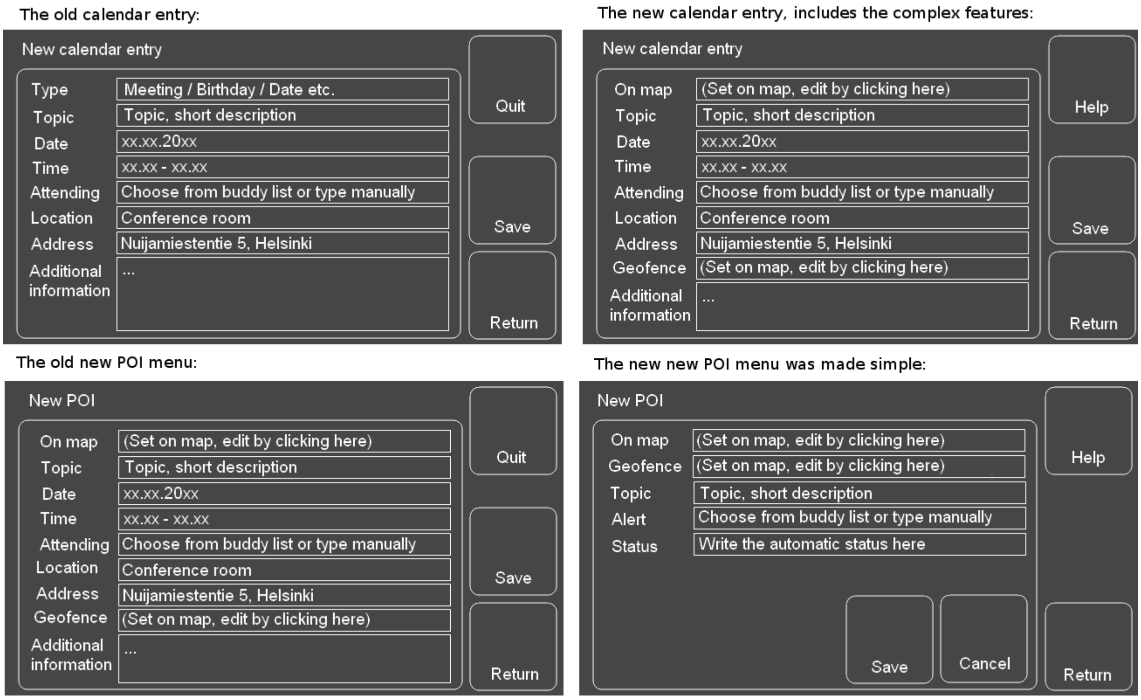


Figure 8: The removed code menu was replaced with the ads menu

When it came to calendar markings and map markers, the user felt that there were a few shortcomings on the demoed version of the user interface. The calendar markings should leave the map view automatically, when the meeting becomes obsolete. A good feature would be also to show the markings on the map only during the day that they are relevant. The calendar markings and user-generated POIs contained mainly the same fields, and were both complex. The user suggested, that the complex markings that included dates and times were left to only the calendar markings, and the POIs were made simple. Figure 9 shows the difference.



The old calendar entry:

New calendar entry

Type	Meeting / Birthday / Date etc.
Topic	Topic, short description
Date	xx.xx.20xx
Time	xx.xx - xx.xx
Attending	Choose from buddy list or type manually
Location	Conference room
Address	Nuijamiestentie 5, Helsinki
Additional information	...

The new calendar entry, includes the complex features:

New calendar entry

On map	(Set on map, edit by clicking here)
Topic	Topic, short description
Date	xx.xx.20xx
Time	xx.xx - xx.xx
Attending	Choose from buddy list or type manually
Location	Conference room
Address	Nuijamiestentie 5, Helsinki
Geofence	(Set on map, edit by clicking here)
Additional information	...

The old new POI menu:

New POI

On map	(Set on map, edit by clicking here)
Topic	Topic, short description
Date	xx.xx.20xx
Time	xx.xx - xx.xx
Attending	Choose from buddy list or type manually
Location	Conference room
Address	Nuijamiestentie 5, Helsinki
Geofence	(Set on map, edit by clicking here)
Additional information	...

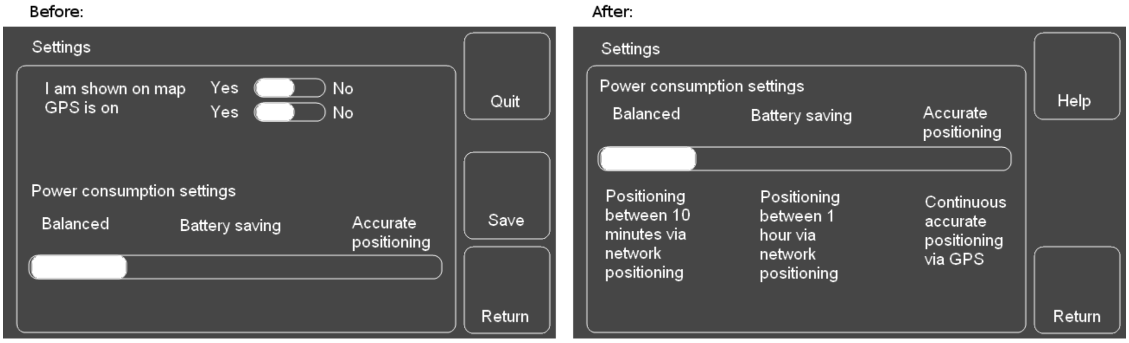
The new new POI menu was made simple:

New POI

On map	(Set on map, edit by clicking here)
Geofence	(Set on map, edit by clicking here)
Topic	Topic, short description
Alert	Choose from buddy list or type manually
Status	Write the automatic status here

Figure 9: The modified calendar and POI entries before and after modification

Several more minor details were updated to the mock-up. The settings for the mobile client were also moved from the main menu to under the user's profile. This can be seen in figure 3, where the settings is not in the main menu anymore in the new version of the main menu. This seemed like an intuitive location for the user, since the settings aren't modified that often. The settings for GPS were removed, because the user couldn't understand what it did, and why she should be able to control it. The user also felt that the interface needed a better term for the user's visibility on the map to the user's friends. The visibility setting was moved under the user's profile. The changes can be seen in figure 10.



Before:

Settings

I am shown on map Yes No
GPS is on Yes No

Power consumption settings

Balanced Battery saving Accurate positioning

After:

Settings

Power consumption settings

Balanced Battery saving Accurate positioning

Positioning between 10 minutes via network positioning

Positioning between 1 hour via network positioning

Continuous accurate positioning via GPS

Figure 10: The settings before and after test 2

The map view also needed some adjustments. The user thought that the used OpenStreetMap maps were colored horribly. The streets and buildings use the same color, which makes them a mess. The colors were quite impossible to change in a short period of time, so they were left untouched, but the idea of changing them in the actual implementation was taken. Also the user wanted the close and menu buttons to be changed to each other's positions to maintain consistency in the UI, since the menu button is located in that place in all the other views of the UI. This is visible in figure 5.

Last, but not least, the user felt that the routing needed to have an option for using the current location instead of always having to type the address of the location by hand. This saves the user lots of time, since typing without a “real” keyboard is slow. The change is shown in figure 11.

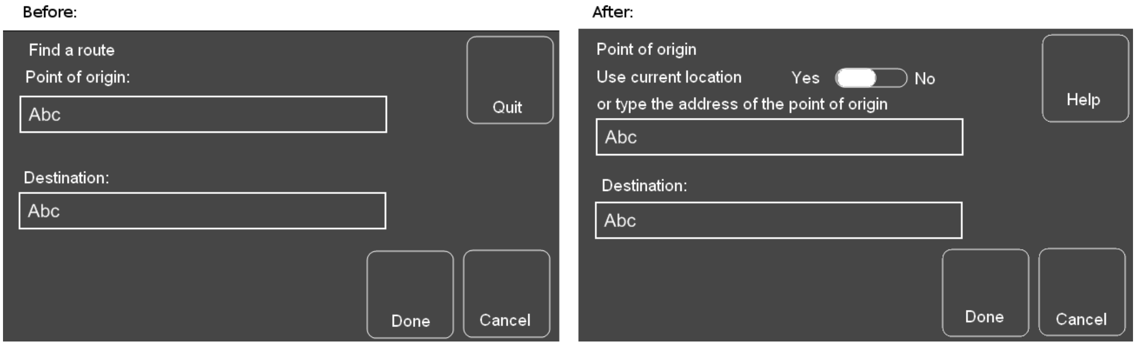
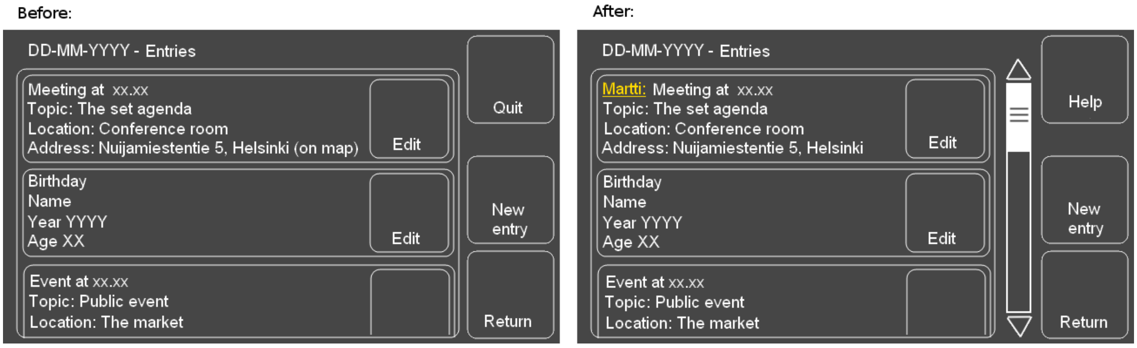


Figure 11: Using the user's current location was added to routing

Changes after test 3

The first thing the third user said about the prototype was “does the color of the user represent the user's sex?” This obviously states that the used colors are not clear enough to indicate the user's status of using the system or just being tracked. Also the confusion might have been caused by the fact that in the interface, all visible men were displayed with the color green and all visible women with the color red. This puts some doubt to the system of using colors to indicate online status, but no changes were made to the system because of this.

The user requested a few things to the calendar markings in the system. First, the markings should allow the possibility to forward the marking directly to a number of the user's friends' friends. This would make throwing parties much easier, when the users could send the marking directly to their dates. Also the user requested that the marking would show the name of the person who made the marking, and that the name would act as a link to the user's profiles. The link is shown in figure 12.



Before:

DD-MM-YYYY - Entries

Meeting at xx.xx
Topic: The set agenda
Location: Conference room
Address: Nuijamiestentie 5, Helsinki (on map)

Birthday
Name
Year YYYY
Age XX

Event at xx.xx
Topic: Public event
Location: The market

After:

DD-MM-YYYY - Entries

Martti: Meeting at xx.xx
Topic: The set agenda
Location: Conference room
Address: Nuijamiestentie 5, Helsinki

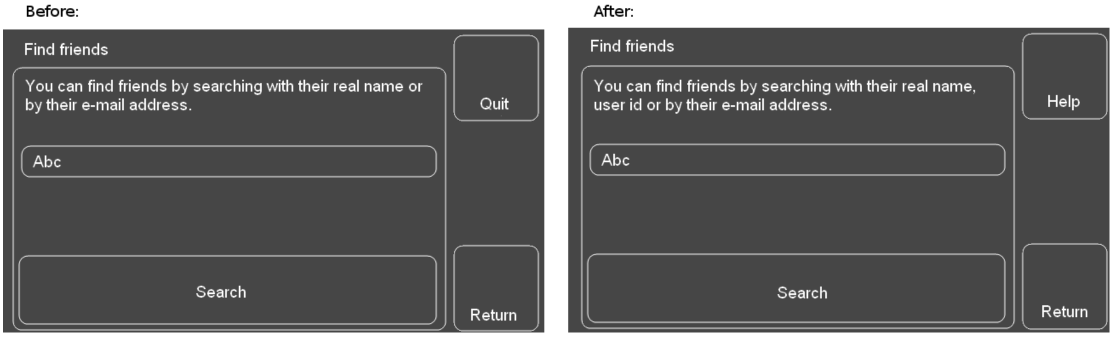
Birthday
Name
Year YYYY
Age XX

Event at xx.xx
Topic: Public event
Location: The market

Figure 12: A link to the profile of the user, who has created the shared entry

The map markings in the client needed the option to group them according to whom the markings are shown. The groupings need to correspond to the groupings in the friends list, and allow new subgroups to be created. The user also had a new idea to the geofence functionality. When the user enters a geofence, the system could make the user invisible to others, and visible again when exiting the geofence. Also when it comes to geofences, the status updates or messages to users could occur also when leaving a geofence. These seemed like a good addition, and both ideas were presented to latter users orally.

For the search criteria for friends, the user wished to be able to seek friends with their user ids. This was added to the mock-up. The change is visible in figure 13.



Before:

Find friends

You can find friends by searching with their real name or by their e-mail address.

Abc

After:

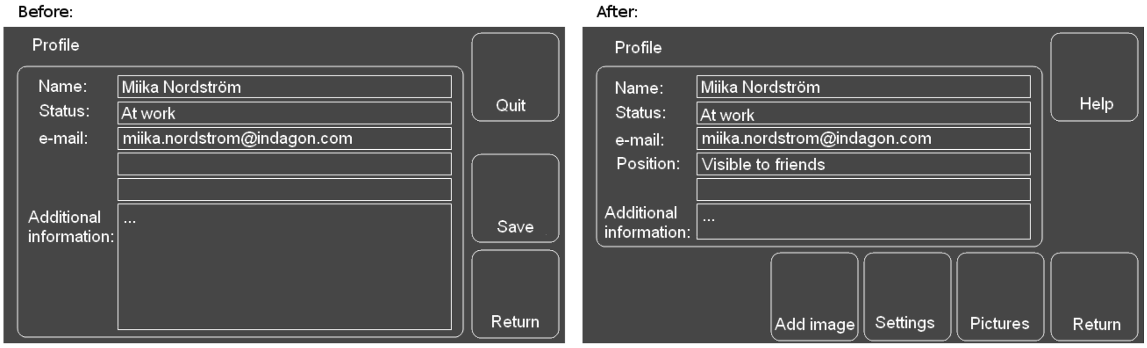
Find friends

You can find friends by searching with their real name, user id or by their e-mail address.

Abc

Figure 13: Searching friends with the user's user id was added to the mock-up

The profile contained very little information in the start of the tests, which was intentional, to get the users to tell what they thought was the information they wanted to share. This user wished a “personal field” to the profile, where the user could share some thoughts with the world. Kind of a thought-box, where you can express your self to the world. The user was also interested in finding out other users' interests. The changes to the profile thus far are shown in figure 14.


Before:

Profile

Name: Miika Nordström
Status: At work
e-mail: miika.nordstrom@indagon.com
Additional information: ...

After:

Profile

Name: Miika Nordström
Status: At work
e-mail: miika.nordstrom@indagon.com
Position: Visible to friends
Additional information: ...

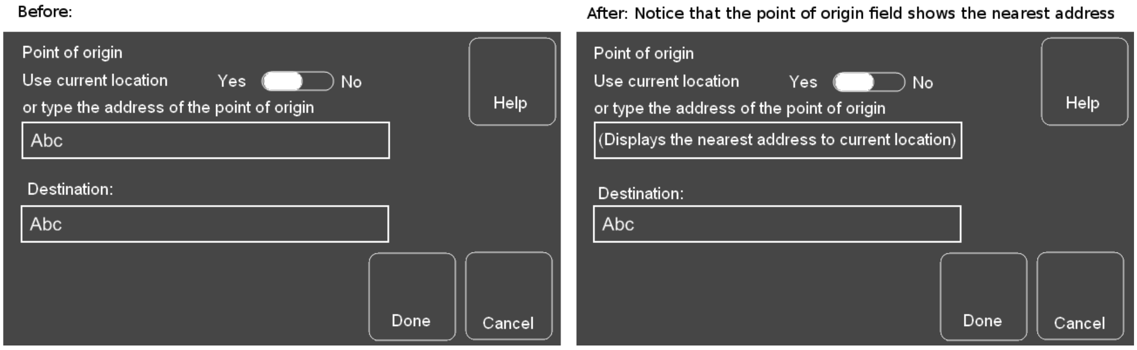
Figure 14: The old and new profile

Contrary to the previous interview, the interviewee wished that the settings for the program would be displayed on the main menu, and not under the user's profile. The settings remained under the profile to further see, what other users thought of hiding the settings under the profile.

A useful little detail was brought up in the interview concerning the photos that the user shares in the service. There should be some control to whom can see the shared photos. Once again, grouping the user's friends was brought up, so it seems like a really important aspect of the service. Later in the interview, the user mentioned that grouping the user's friends would be extremely helpful.

The user wished that the switch to turning the user's visibility on the map be made more easily accessible. The user wished there would be an icon on the map, that would allow hiding and showing the user easily. The placement of a shortcut to hiding and showing the user's location was taken under advisement, and will be placed either in the main menu or on the map view.

Navigation needed an automatic function of showing the nearest address to the user as a default on the search field. Even if the user was using network positioning, this would be a good guess on where the user is, and would allow the calculation of the route to start from a good guess of the user's location. The improved routing is shown in figure 15.


Before:

Point of origin
Use current location Yes ○ No
or type the address of the point of origin
Abc
Destination:
Abc

After: Notice that the point of origin field shows the nearest address

Point of origin
Use current location Yes ○ No
or type the address of the point of origin
(Displays the nearest address to current location)
Destination:
Abc

Figure 15: The routing has the nearest address to the point of origin added

Regarding the user-generated public points of interest, the user wished that there would be a function of searching through them with keywords in order to easily find information on the map. This was explained as a possibility in later interviews.

Group messaging was an option that was not considered before the tests, but this user brought it up, and it sounded like a reasonable form of communication. This was brought up in the grouping parts of the interview in the later tests.

Perhaps the best idea the third user had was about the anonymous instant messaging. The user said, that there should be a reputation system in effect in AIM to weed out the troublemakers in the service. Because let's face it, there will be those that just want to create trouble using the system. The user propositioned a system where the conversational partners rate each other at the end of their discussion. It would have to be presented to the users as not a questionnaire on whether they thought the conversational partner was a nice person, but rather as how was the partner as a conversationalist. This way those who are troublemakers, would get low scores, and others would most likely avoid them. A system like this would greatly improve the use experience of those who wish to use the service to really find a conversational partner.

Changes after test 4

The fourth test subject was critical to the menu icons in the mock-up. Or rather, the absence of the menu icons. The user felt that it didn't convey a finalized view of the product that the icons were only blank buttons. Unfortunately, there was no time to create proper icons during the test process for it to be reasonable to add the icons to the mock-up. Of course icons will be used in the final product to symbolize the different functions of the system. The missing icons are demonstrated in figure 16.

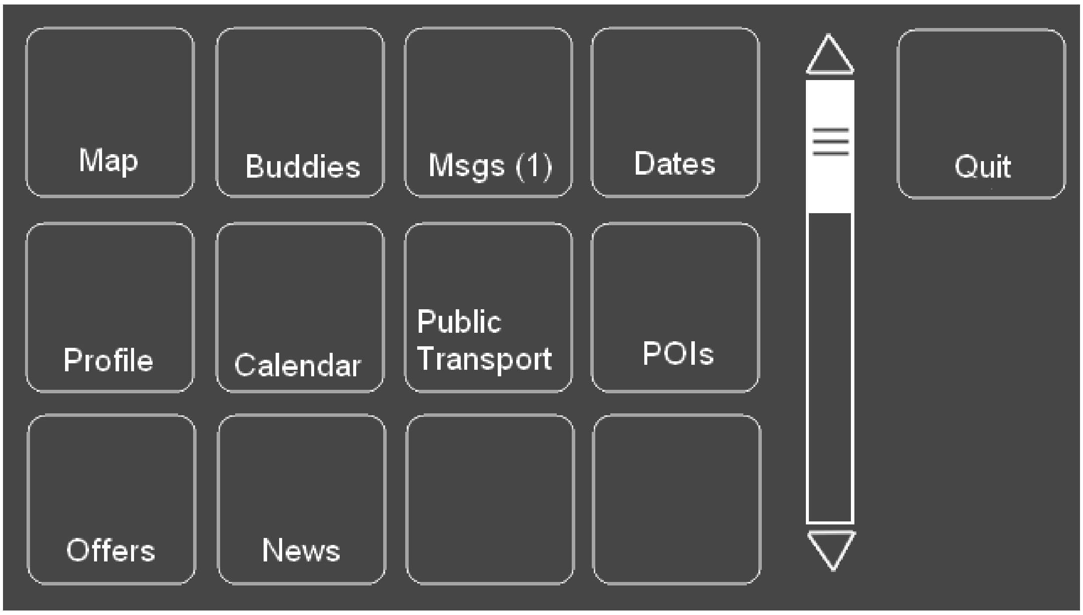
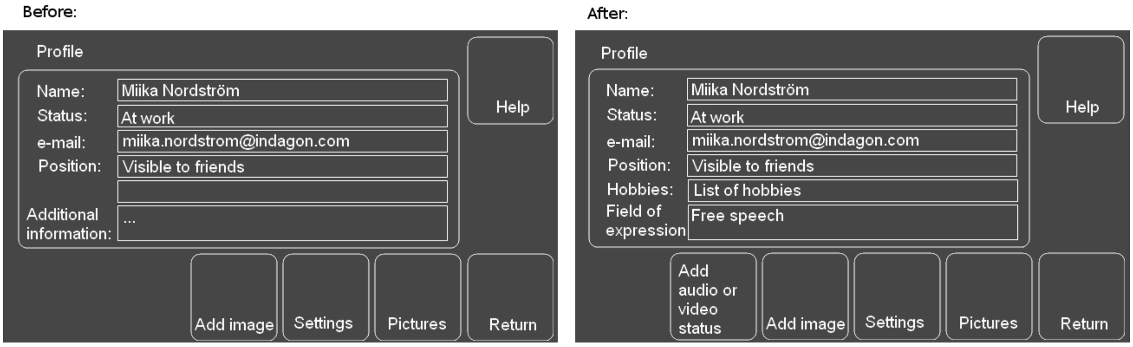


Figure 16: The mock-up UI had no icons implemented

When it came to presenting the user with how to search for new friends, the user had an excellent source of information that had yet to be utilized. The user presented the option of having public friend lists, that would allow a user to view friends' friend lists. This would make it easier for groups of friends to be aware of other people in the system. In later interviews, this was presented verbally to the users as a method of searching for friends.

The fourth user was interested in other users' work, studies and human relationships and suggested that they be added to the user profiles. This was also verbally presented in the following user tests. Another idea for the profile, were video and audio status messages. This would be something that is a competitive edge for the product, since most services don't use them. This was added to the UI for the later tests. The user also requested that there be a period of the day, that these could not be added, for example during nighttime. This was mainly to prevent users from uploading video and audio when drunk. The option to add a audio or video status is shown in figure 17.



Before:

Profile

Name: Miika Nordström
Status: At work
e-mail: miika.nordstrom@indagon.com
Position: Visible to friends
Additional information: ...

After:

Profile

Name: Miika Nordström
Status: At work
e-mail: miika.nordstrom@indagon.com
Position: Visible to friends
Hobbies: List of hobbies
Field of expression: Free speech

Figure 17: The profile with the voice and videostatus

The user was cautious when it came to having a shared calendar. The user felt that there should be an amount of control added to making entries in the calendar. The user suggested the use of the ability to deny events from the calendar. This would work in a fashion, that when the entry is created and shared, each user that it was shared would have the option of denying the entry from appearing in the user's calendar giving a sense of control to the users. This is a reasonable request, so it was presented as an option, although it was not implemented in the mock-up.

On the other extreme of this was the suggestion of having completely public events, that anyone could join and get an entry to their calendar. This way large public events would have an easy channel of sharing information to users attending their events.

The finnish language does not include a term for “geofence”. The user felt that a direct translation for the term isn't sufficient enough to describe the concept of a geofence. The term used in the mock-up was “geoaita”, in which geo doesn't have a finnish alternative, while fence is directly translated to “aita”. This will be a real challenge to translate properly, so the term wasn't modified in the mock-up.

The navigation view, was once again under criticism, this time on which vehicle should be used for transportation. The user felt that it would be too frustrating to use, if the system asked for the mode of transportation every time the user made a search. The user suggested that there would be two search buttons, one for using the default settings and one for doing a search with non-default settings. This was corrected in the next version of the mock-up. Figure 18 illustrates this.

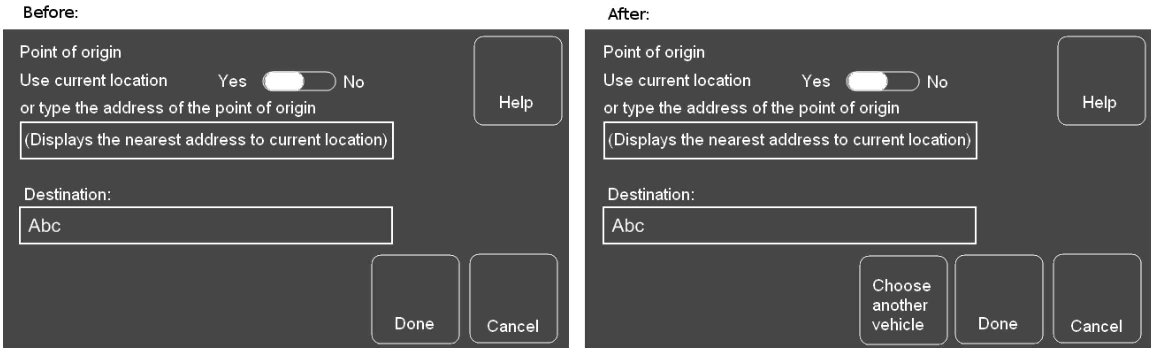

Before:
Point of origin
Use current location
Yes
No
or type the address of the point of origin
(Displays the nearest address to current location)
Destination:
Abc
Help
Done
Cancel
After:
Point of origin
Use current location
Yes
No
or type the address of the point of origin
(Displays the nearest address to current location)
Destination:
Abc
Help
Choose another vehicle
Done
Cancel

Figure 18: The routing was updated with default transportation and an option to change the transportation when needed

On matters of privacy, the user had an opinion about user-made geofences that track other users. The user oddly felt that consent should be asked for each geofence, while the same information can be acquired just by looking at the service at any time. Perhaps the user felt threatened by the fact, that the system would send a message to the user, allowing better tracking of the user to other users. This was left as a threat to the system and was not considered in the mock-up.

An excellent idea from the user was that the service should allow the users to respond to whether targeted ads came to the right address. This way, the user gets to influence which kind of ads are sent to the user, and feels in control of the service. This also benefits the advertisers, because it allows the ads to be sent much more accurately, which is what the advertiser wants and pays for. This could generate revenue for the service, and thus make the future brighter for the service. This feature was presented to the users in the later interviews verbally, and it received a very warm welcome.

Changes after test 5

The first change to the mock-up in test 5 was made to the way POIs are handled. The user pointed, that scrolling around the map might not be the ideal way to handle a POI that is to be placed far away, for example to another city. So adding POIs with an address was made possible, in which the POI is placed automatically on the map. When asked about POIs with geofences that send the user alert messages, the user said that they would be useful, but would need an option to turn on and off whenever the user feels like not receiving messages. This is a fair point, one might not want to receive messages for example, in the middle of the night, or when on a trip abroad.

An excellent point about anonymous instant messaging was the fact, that the user didn't want the dates to be from too close to the user. For example, the user felt that it would be disturbing if after an AIM session, the user would find out, that the other user would live in the same building. This is a good point, there needs to be an option to set a safety zone so that the user doesn't end up talking to people that are too close by if they do not want it.

When it came to dealing with the map, the user wished that the map markings could be configurable as in which markings are shown and when. This could be dealt by using different layers for different types of map markers. This way the user could choose the visible layers and that way customize the view to suit the user's taste.

In the map view, when the user had clicked the other user and the system had displayed the other user's status message, the user was baffled on how to get the message off the screen. This does not bode well for the system, so a clear exit from this state must be marked.

The grouping of friends was brought up again, this time in a form, that the user must be able to have different types of friends regarding location information. The user insisted that there must be an option to group friends in a group that can't see the user's location. This is understandable, the user might not want to share the location information with all of the user's friends. This functionality emphasizes the focus of the system towards more of a social network. This was taken under consideration.

An important observation about the adverts was that even though the user might have a narrow field of interests, the user can't get the same adverts every single day at the same location. The emphasis on this was the location. For example, if you got the same ad every day when commuting to work, it would really start to annoy and feel like harassment. There must be a limit taken into consideration in the ad machine, that makes sure the user doesn't get the same ads every day.

Changes after test 6

The sixth test subject was interested in developing the advertisement portion of the service. The earlier added menu for adverts needed some improvement. The user wouldn't be satisfied with just having a menu that allows browsing of ads. The user wished to have keywords implemented to the ads. These keywords would need to include the location of the advertised product, product brands and dealership brands. An option to compare discounts with separate variables, such as the product's price were considered to be a welcome feature. The changes to the ad menu are presented in figure 19.

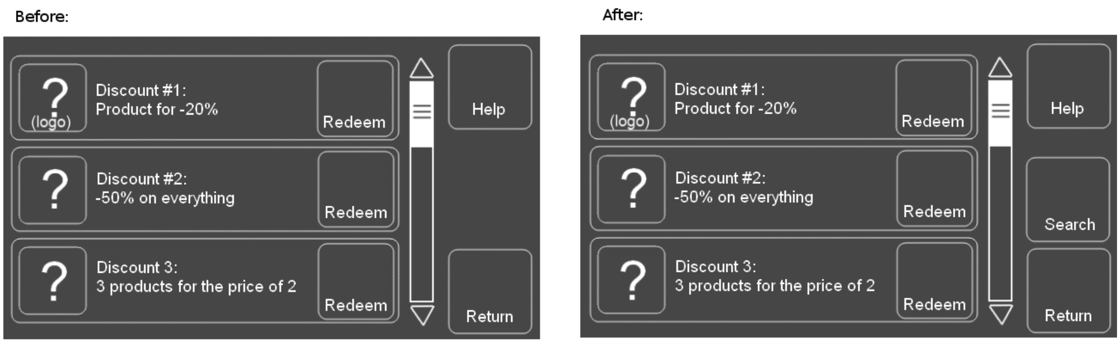
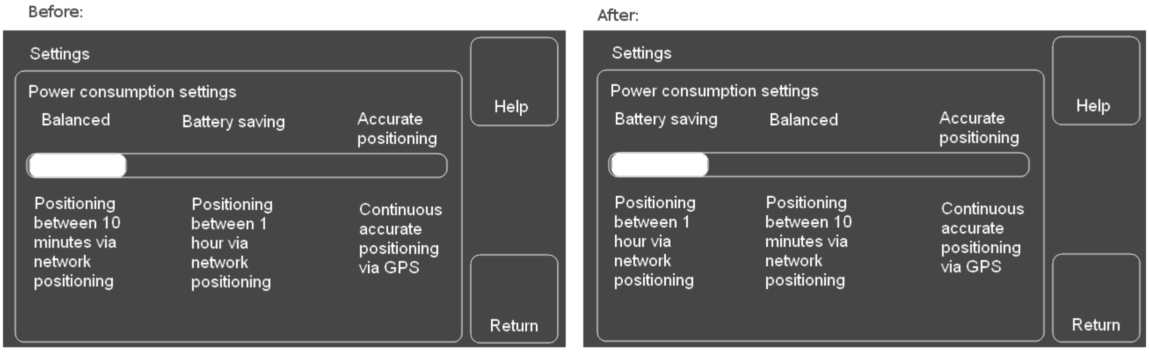


Figure 19: The ad menu was implemented with the search function

There were a few cosmetic UI changes, that the user wished that were done differently. The settings for the mobile client mock-up contained a setting for battery usage. The options were presented from left to right in the order: “balanced consumption” - “battery saving scheme” - “accurate positioning”. The user thought that it would be better to keep the options going from the most energy efficient choice to the most power consuming option. This is a good idea that helps the user get a better grasp of the battery consumption. The order of the settings was changed to: “battery saving scheme” - “balanced consumption” - “accurate positioning”. This is demonstrated in figure 20.


Before:

Settings

Power consumption settings

Balanced — Positioning between 10 minutes via network positioning

Battery saving — Positioning between 1 hour via network positioning

Accurate positioning — Continuous accurate positioning via GPS

Help

Return

After:

Settings

Power consumption settings

Battery saving — Positioning between 1 hour via network positioning

Balanced — Positioning between 10 minutes via network positioning

Accurate positioning — Continuous accurate positioning via GPS

Help

Return

Figure 20: The rearranged battery saving options

For the main menu, the user had an interesting idea. In the main menu, the close button that needs to be separated from the other buttons takes a lot of screen space, because it is separated. The user was concerned of the lost screen space, and suggested that perhaps the close button doesn't need to be shaped as a button, but instead as another form. This is an interesting idea, that was not implemented to the mock-up, but was left as an idea to think about. Figure 21 demonstrates the problem with the close button.

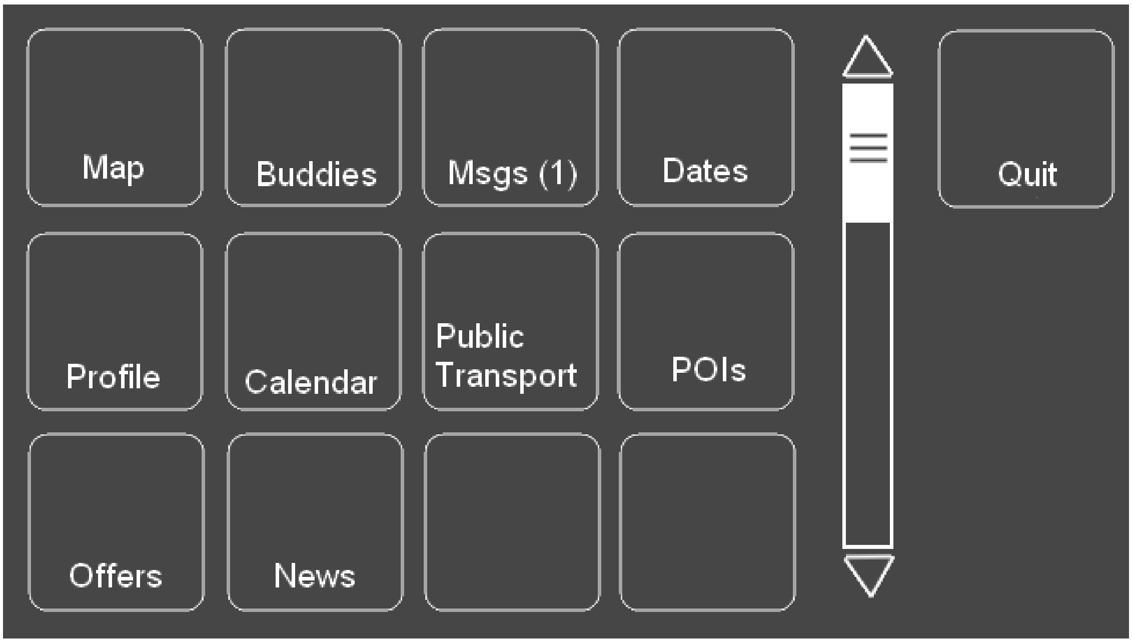


Figure 21: In the main menu, the close application button takes lots of space, the user suggested that the buttons shape could be changed to save screen space

The user thought that the user-generated POIs need to be expanded to include public POIs that are available to all the users. These POIs should include restaurants and other services. And the user should have a search tool to conveniently find businesses. This was not implemented to the service, but was described verbally in the following interview.

When it came to geoblogging, the user was excited about it. The user uses Nokia's SportsTracker[19]. It is a service that allows users to record their routes with the aid of GPS positioning. The service is aimed to users that do endurance sports, such as jogging or cycling that are usually done outside, where GPS positioning is possible. The user propositioned that the geoblogging in AmiCarta is done in a similar matter, so that it records average speeds and the time it took to travel the route. The user thought that he would use these features to record hiking or cycling trips. The user also mentioned that geoblogging would be a nice feature when abroad. The user would record routes traveled and edit them later, when the user has the time to go through the routes. When it came to placing pictures on the map and sharing text on the map the user had a vision, that video should be also synchronized on the map. So that when the video was playing, a marker would move across the screen in sync with the video. This is a feature that would be really hard to implement to work automatically. The only reasonable way to accomplish this kind of functionality would be to implement the location to the video itself, and this would require hardware changes to video cameras. No doubt, this will be a feature on future video cameras, but is something that isn't available yet.

When talking about the navigation, the user brought up a few extra features that would add value to the functionality. The first feature was for navigation with public transportation. Since public transportation is not always on time, it would be extremely useful to have information about the disturbances that affect the user's life. This would be a huge improvement, if the bus stops and train stations would show the abnormalities in traffic on the user's map view. Another feature for navigation was to routing by car. The user suggested that the system use flow information about the traffic to calculate the optimal routes. The user said that some navigators already have this feature, and that it would be useful. These features were not implemented to the mock-up, but were presented orally.

The AIM searching for conversational partners worried the user. The user was concerned that when the user fills out a list of criteria for the partner, the system will not find any users that would suit the user. So the user suggested that the partners don't need to fill out all the criteria entered, so that if no perfect matches are found, the system would still give results of the best matches. This was taken into consideration when creating the system.

Grouping came up in this interview also, but this time the user wished that grouping would be used in conjunction with geofencing. The user wished that the system would have an option to filter the groups of friends to be able to contact the user only when a certain set of conditions are met. For example, the user would like to receive messages from co-workers only when the user is at work. Also the user could want to control when the user's friends can see the user. This is an excellent feature that the users would definitely appreciate, since especially location information is something that the users

[19] http://sportstracker.nokia.com

surely want to keep private from some of their friends.

A really cool feature the user suggested was that the service adapts to the users location. This would mean that the service would adapt starting from the menus so that the user could control the way the service adapts. For example, when the user is at work, the user might not want to use AIM or instant messaging with the user's friends. This is a good idea, but also possesses a little problem regarding the usability of the system. Hiding the features when in an area, the user might get frustrated if the user needs to use the features. Also this is a real challenge to implement technologically.

Changes after test 7

The seventh and final user test provided little improvements to the UI. This is the reason it was the final test, it satisfied the condition of having less or equal to two suggestions for improvement on the UI itself. The test had more new features than this though.

The two direct UI improvements regarded the user's friends. The user wished that if it was possible to create POIs that give the friend a status, it should be possible to view the friend's original status message somewhere. Otherwise, it would be really inconvenient if the user wished to see the friend's original status message. The user also wished that there was a possibility to choose the user's screen name to be something other than the user's real name. For example, if the user has lots of friends called “John”, it might get confusing to have a map full of Johns while not knowing which John is which. The screen name should be editable for both the user and the user's friends.

When the user is traveling around, the user would wish for the system to take it into consideration. For example, if the user is driving around, the service should notice it, and advertise roadside services. This is a good feature, which is not that impossible to implement, because it requires only that the system knows how fast the user is traveling, and which highways are nearby and what kind of discounts those businesses have.

The user was very positive towards mobile marketing. The more profiled the ads are, the more the user wished to receive them. The user saw no problems in receiving multiple SMS ads each day, which was completely opposite to the other test subjects.

When discussing the local news, the user thought they are useful only when traveling and getting to know your surroundings. This is a fair point, there could even be a home region for the user, and whenever the user would arrive in a new location, the system could show local news to the user or ask the user if the user wishes to be displayed local news.

Another new feature for POIs were public events. The user wished that there could be POIs for public events, that would show to users. The user should also be able to search through the events with keywords. An additional feature to this would be if the system would send the user tips on events that the user might be interested in. This feature would be a sort of “what to do today” service that gives tips, for both advertised and not advertised events and similarly to events that have tickets that cost and free events.

In this test the set requirements to the testing were met and thus this was the final user test.

5.3.2 Opinions on the original features of AmiCarta

The users were asked to fill in a questionnaire that dealt with the existing features of AmiCarta. In the questionnaire, the users were asked to evaluate on a scale of one to five, how important a feature was to the user in the service. The results are presented in Appendix E. User opinions were also gathered in the interviews, and both the questionnaire results and the interview results on the existing features are presented here. The references to the interview results are presented as x.yy, in which x represents the number of the interview and yy the number of the observation in the interview. All the observations can be found in Appendix D. The scale of the questionnaire was from one to five, so results of over 3.0 are favorable, and below that unfavorable.

In chart 1 it is presented what average scores each feature received from the users. The top features were navigation with public transportation, the map view and tracking the user's friends. Other features that received an average of 4.0 or over included traffic announcements, getting all the news on the service, being able to use geofences and instant messaging with the user's friends.

Chart 2 presents the amount of variation in the grades for the feature. The smaller the variation in this chart, the more unanimous the users were about the feature. As presented, the most solid grades in the service were for the navigation with public transportation, messaging with friends and the shared calendar. On the other hand, the instant messaging with friends received the most mixed grades, which is particularly because of the fourth interview, as seen in Appendix E.

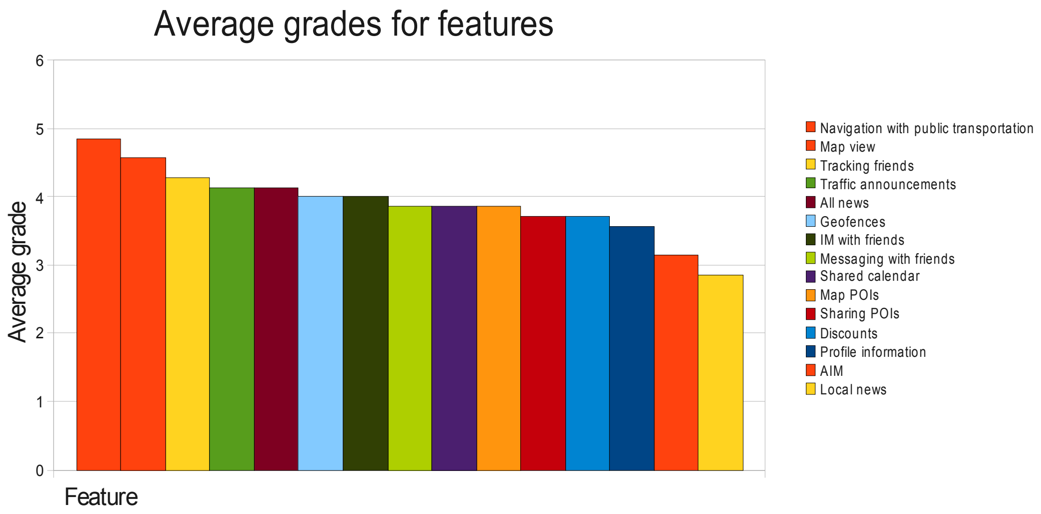


Chart 1: Average grades for the tested features on a scale of one to five

Almost all features received an average grade of over 3.0. The only feature that received a score of under 3.0 was the local news view. The users felt that this was an absurd concept, and would rarely be of any use to the users. This was demonstrated specifically in the observation 6.39 and almost all observations in the category AQ support this.

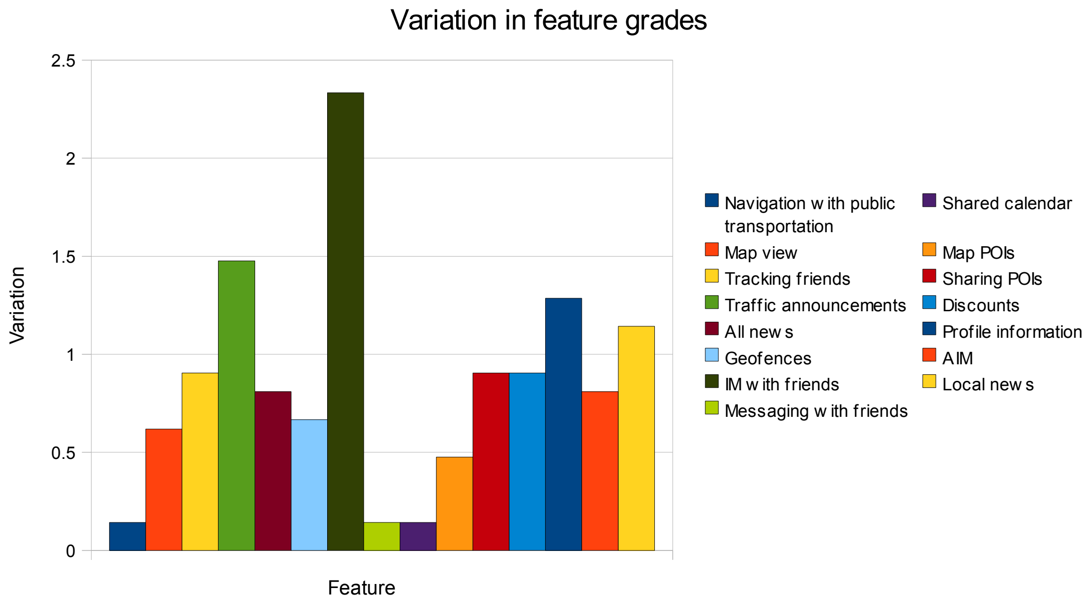


Chart 2: Variation in grades received by different features

Navigation with public transportation

As the favorite feature in AmiCarta, the navigation with public transportation routes received lots of praise. One user even felt that this feature would be enough for them to use the service, even if it had no other features. This is observed in the entry 1.22. Other users listed the feature as something that they would definitely use in the service, as can be seen in entries 2.44, 5.10, 6.1 and 7.62.

The navigation would require all methods of public transportation. The users thought that the navigation would be handiest, when in going to a place they hadn't earlier visited or if the user got lost somehow as was the case in observations 1.24, 3.43, 5.30, 6.19 and 6.20.

The users found several ways to improve the navigation. For example, the system could have automatic routing to shared calendar markings, that had a location. All the new features are listed in the category AX.

Map view

The map view was highly appreciated by the users as is visible in chart 1. Since the map is mostly a part of the user interface, the map view was largely covered in chapter 5.3.1 with the exception of a few opinions.

In the fourth interview, observation 4.49, the user thought that the best part of the service was how everything was integrated to the map view. An almost exactly similar comment was made in observation 6.40, where the user thought that the best feature in the concept was the map view and the possibilities it gives to the system. Also the type of maps used received some suggestions. One user wished that the service would use cycling maps (1.21), and another wished orthographic maps (5.55).

Tracking friends

When it came to tracking the user's friends, the most requested feature was the ability to group the friends to different groups. This came up in several interviews (3.9, 3.24, 5.22, 6.30). Also users felt that the more friends the user had, the more of use the service would be (3.25, 6.6). Generally, the users found tracking friends a good feature in the service.

One user though was skeptical to whether it would provide any real benefit of seeing the user's friends on the map (6.9). In one interview, the user was also concerned that users would get anxious from having their location presented on a map at all times (7.33). The same user also noted, that controlling the user's visibility would be something that the user would use a lot.

Traffic announcements

The importance of having traffic announcements in the system depended on the user having a car or not having a car. The first user had no car, and no use to the traffic announcements (1.10), while users with cars felt that they were an important source of information to the service (3.44, 4.30, 5.31, 6.21, 7.40).

In the second interview, the user pointed out, that having traffic announcements shown on the main menu, provided no use to the user (2.10). Since the announcements are highly dependent on their location, it would be more useful to only display them in the map. The presence of traffic announcements also inspired the users to innovate that a similar system should be in place for the public transportation (6.50). Another detail mentioned by the users was that the traffic announcements could be included in navigation (6.51) with road speed information.

All news

When talking about news, the user was faced with a question of wanting to read news by category, or by proximity. Nearly all users wanted to receive news by category (1.19, 4.37, 5.44, 6.31, 7.52). One of the users also felt that news shouldn't be included in the concept at all (2..54). A similar observation was made in the last interview, where the user didn't feel like news would fit in the concept right (7.63).

Also in the last interview some nice points were brought up. The user would like to have entertainment news sent to the user (7.7), but those kind of news are usually hard to locate on a map, since they happen across the globe. With only the map view, it would be very frustrating to scroll around the world to find news. So it is necessary to have a view for categorically organized news.

A reasonable concern was, what the system will do, if there are news that cannot be placed on the map (7.16). This is a conflict that must be resolved. Will the news appear in the categorical view, or will they be completely ignored?

A great suggested feature would be that news would appear by proximity, when the user is traveling (7.5). This would require that the system would have a home area for the user, and when the user left that home area, the system would ask the user, if the user wishes to find out, what is happening around them. This would feel like a natural use case for local news.

Geofences

An odd thing about using geofencing to track the user's friends was brought up in a few interviews. The users wanted to be asked for consent on every single geofence created by another user (1.28, 4.16, 4.43, 5.49). This seemed curious, because basically it is the same feature as seeing the user's location on the map. Tracking with geofences just lightens the user's workload on tracking. Perhaps it makes it too easy to follow another user and that is the reason people want to be able to deny the possibility to do it.

Other users weren't that bothered with someone following their movements with the geofencing, since it requires lots of work from the friend who wants to stalk (2.43, 7.59). Perhaps it is purely coincidental, that the users who were least worried about stalkers were women, and the users who wished to be consulted about tracking were men. Or perhaps the interviewed male users have used social media to find out stuff about other people and feel that they can easily violate someone's privacy.

Overall geofences were seen as an easy way to control the service. A nice way to control the service was suggested in the third interview, where the user suggested that geofences could be used to control when the user is being positioned and when not (3.20). This way the user could create geofences around locations, where the user doesn't want to be seen by the user's friends.

Some concern was brought up about overlapping geofences (3.15). What will happen, if the user creates two geofences that overlap each other? A possible solution to this would be to have the geofence dominate that the user is closer to, or to have the geofences cancel each other out, or to add to each other.. This way for example, if two geofences had a status message, the messages would combine.

A use case for geofences the users mentioned was planning parties. If there was a geofence around the party venue, the user would receive information about upcoming visitors before they would get the chance to ring the doorbell (4.44, 5.5).

If the user would use geofences for tracking the user's friends, one critical aspect needs to be considered. In the original concept, the possibility to turn geofence alerts on and off was not presented. This is something that needs to be added (5.2, 7.58).

IM with friends

Instant messaging is something the users found to be a basic functionality for any social service, and the users had very few opinions on it. Perhaps the most striking opinion was made in interview four. The user felt that IM is so stressful, that the user wouldn't like to have it at all (4.27).

Other opinions suggested that there needs to be the possibility to send a chat automatically as a message, if the conversational partner turns offline (2.35). Also the system needs to display to the user, when other users are just being positioned and when they are logged in to the system so that the user knows when an IM session can be instigated (2.33).

An important point was made in the last interview, that IM would be much more useful, if the IM client in AmiCarta could be used to IM with other IM accounts as well (7.34). This would make life easier for the user, because it would be possible to use just one IM client instead of using different clients for different IM networks.

Messaging with friends

Asynchronous messaging with friends corresponds directly to sending e-mail. This didn't go unnoticed by the users, as it was pointed out in an interview (6.12.). Some users felt that there was no need to add another messaging channel, and the asynchronous messages could be removed from the concept or the messages could directly sent to the user's email (1.12, 2.53, 7.21).

A good suggestion made by the users was that, the messaging could be used in synchrony with the geofencing. The user would group the user's friends and categorize the groups. When the user has the categories, the user can define when creating a geofence, which groups can send messages to the user, when the user is inside the geofence. All other messages would be put on hold, and sent to the user, when the user enters an appropriate geofence (6.53). For example, the user could define, that messages from the users' friends wouldn't get through, when at work. This would help users control their usage of time. If this system would be put in place, it would be important to notify the sender of a message, that the user will not receive the message before leaving the current geofence, or entering an appropriate geofence (7.22).

Another way of using geofences with messaging was that the user could leave messages that are tied to coordinates. This way, when the targeted friend would enter the geofence, the friend would receive the message (3.22). This would be a nice feature, but the users didn't come up with lots of use cases for it (7.48).

Shared calendar

The number one reason to use the shared calendar was to share events with friends. These events ranged from sporting to partying (3.42, 5.26, 6.15). Also public events were wished to the calendar (4.11).

The users considered a shared calendar as a good feature (5.25, 6.14), but also had requests to the functionality in the calendar. A nice feature would be to be able to invite plus ones directly from the calendar (3.2) and also adding the possibility to control the invited friends' number of plus ones they can invite (3.3).

Also the shared calendar raised some questions on whether it can be managed properly. The user needs the option to deny calendar entries (4.1, 4.8). Also the entries need to leave the map view, once the date has passed, so that the map view doesn't get cluttered (2.32).

Map POIs

The users wished to see on the map not only user-generated POIs, but also POIs for businesses and services (4.28). The users wanted to be able to grade services in these POI locations, so that it would be easy to recommend the services to friends (7.49). A way to control the POIs would also be that those POIs that receive low scores from the users would automatically be removed from the map (4.28). Also it was requested that the user can search through POIs with a search functionality (3.21). Also an important factor with POIs is that they can be hidden from the map when not needed (5.24).

Sharing POIs

When talking about sharing POIs, control was an important feature once again. As with shared calendar markings, the user wished to have control of what was shown on the user's map view. A suggestion was made to be able to block POIs from appearing on the user's map (4.22).

POIs were a feature that clearly was understood, because it generated very little discussion with the users.

Discounts and ads

The original concept had a system in which the user had to find discount codes in other media, that the user would enter in the system to get a discount coupon. This was considered as a useless way of advertising. If the advertiser wants to advertise properly, the ads need to be as accessible as possible. It was suggested that the code system was to be thrown in the bin, and a menu for the discounts was to be added (2.55).

All the users felt that they would rather receive ads than pay for the service (3.29). This tells us, that the customers are used to not having to pay for web-based services.

An interesting fact in the SMS ads was that users were more willing to accept the messages, if they were well profiled (4.17). This is good news to the service, since the more profiled the ad is, the more it can cost to the advertiser, because the ad is more targeted. An even better feature suggested in the tests was that the user would like to be able to rate the ads sent to the user (4.18). This would then further help the system create an accurate profile for the user.

The SMS ads would prove to be annoying to the user on three found conditions. The first one would be too many SMS ads per day (3.16). The second one would be getting ads too frequently (3.17), and the third one is that the users don't want to receive the same ads in the same place every day (5.17).

Profile Information

For the profile information, the users wanted two kinds of profiles, one for AIM that doesn't tell any information that could help identify the person and the other for the users' friends to view. Also for the profile, the users wanted to have the opportunity to show different parts of the profile to different friends in the service (6.30). This way they could fill in even intimate details of their lives, because they could control that only those closest to them could see the information.

Perhaps the wildest opinion on the profile was that it could change what is shown according to the user's location using the geofences (3.54). This way the most important information to the user would be shown corresponding to the location of the user.

The profile information was something that was left almost blank intentionally to get the users to tell what kind of attributes the user would like to fill in the user's profile. The attributes are covered as new features in chapter 5.4 and not covered here.

Anonymous dating

Anonymous dating was something that the users found to be controversial. On the other hand, they thought that the idea was a nice one (5.36, 7.44), and sounded like a safe way of getting to meet potential dates, on the other hand the users felt that they probably wouldn't use it (1.1, 6.22). Some impact in this must come from the current status in the user's relationship. Indeed four out of the seven users were involved in a relationship at the time of the interview.

Some users had little faith, that the service could find the potential conversational partners for them , if the matching was done automatically (1.4, 3.48, 5.38). To solve this problem, it was suggested that there would be many search criteria in the system (3.46, 4.31). The users preferred lists of attributes in the person to help find the dates easier (5.34, 6.27). If the system used an open description that the users would fill, the interviewed users felt that it wouldn't help at finding the right partners, because it would be harder to search through.

The new features wished for AIM are presented in chapter 5.4 .

Local news

The ability to receive news by proximity was something that most users didn't find very interesting. Nearly all users found that the news are better displayed categorically (1.19, 4.37, 5.44, 6.31, 7.52). It must be seriously considered whether local news is to be included in the concept or not. Another point that needs to be understood, is that the users haven't had a similar service in use before, so they couldn't give the best possible results about the feature, because it was something that needed to be seen in action, with lots of local news.

5.3.3 Privacy issues

Privacy is something that needs to be taken seriously. If the users don't feel like they can control the service enough, they will not continue to use the service. The most critical privacy-related issue in this study was the usage of the user's history data.

Each user was asked, if it would offend the user's privacy, if the service recorded the user's location history and allowed it to be displayed in the service. Five of the seven users interviewed felt that the history information offends their privacy or that they didn't want the feature at all in the service (2.52, 3.50, 4.40, 5.45, 6.32). Out of the two users that didn't have as much concerns relating to the history information, one had underlying conditions, that would make the usage of history information bearable. The conditions to making history not offend the user were that it could not be misused in any way and that the history data could not be shown to anyone, except for the user self (7.53, 7.54). Only the first user that was interviewed, didn't care if other users would get to see the user's history information (1.27).

The users were then asked how they would feel, if the history information was used anonymously to improve the service, and for that the users were willing to approve the use of their history information (3.39, 4.40, 5.47, 6.33, 7.56).

The users also found tracking with POIs to be problematic in regards to the user's privacy. Some users wished that the service would ask for consent on every geofence that the user's friends created. This would probably end up being too much work for the user, if the user's friends were to create lots of geofences for tracking the user.

On the other hand, when it came to AIM, some users thought that the system protected the user's privacy too much. The users wished to know more about the conversational partners to have more interest in contacting the possible partners (2.36, 2.37, 2.39).

It must be seriously considered that the history information should be removed from the system, because it doesn't invoke trust in the users. Instead it could still be used to make the positioning in the service more accurate.

5.4 Features evolving from the validation

The users had lots of new features or improvements to existing features in mind when going through the interview. The received ideas are described in this chapter.

Weather predictions displayed on map

Because the service employs a map basis that shows location information, it would be logical to have a weather prediction displayed on map. The map could also display a rain detector, that shows in real-time where it is raining. This would provide easily obtainable added value to the service, and be a perfect media for weather forecasts.

The data from a rain detector can easily crowd the map, so it must be easy to turn the weather layer on and off. It might be worthwhile to even have a completely separate map view just for the rain detector, while displaying a regional forecast on the regular map view.

Wikipedia information on map

A fine way of providing free information is to include wikipedia[20] articles on the map. This is already done in google maps, so it definitely is possible. Whether or not it is feasible, remains a question mark. The wikipedia entries also crowd the map with icons, making the map less readable. However, it adds value to the service to be able to view what is nearby when the user is on the move.

[20] http://wikipedia.org

Shared call log

Sharing a user's call log lets other users know, who the user has been talking with. The feature would be completely optional, so there would be no privacy issues, since the user would have to switch the feature on himself/herself. The users could not present any viable use cases for the feature, so it won't most likely be a part of the final product. The feature can be classified as “nice to have, but not important”.

Shared media usage information

The idea of sharing media usage information comes from services like Last.fm that collects the users latest played songs. The AmiCarta service could include a location aspect to media usage, and provide the user with statistics of what kind of music the user has listened where. This could provide some information on what kind of a mood the user has been on while visiting different places. These statistics could be anonymized to produce a record of what kind of music people listen to on which parts of a city. This could also be used to create location based advertising, to advertise the most listened bands in a neighborhood to the users.

Perhaps it would be possible to include FM radio usage while using a mobile phone. This all would require a client to the mobile phone.

Proximity alert

The proximity alert would alert the users of their friends that are nearby. This would need to be configured per friend, so that all friends could have different radiuses of distance to the user. If two users would live nearby each other, they might not want to receive messages of each other constantly.

Another attribute that was wished for this feature was the ability to turn it on or off easily. This way the user doesn't have to get information if the user wishes not to get these reports. This feature also raised some privacy issues, as some of the users felt that they should be asked for permission first before being able to give alerts about them.

Social cloud

A social cloud would be a vision of a service that could create ad hoc clouds whenever a number of similar people would be in an area. This would create a cloud on the user's map view that would display an area where there are lots of people with similar attitudes or philosophies on life. That could make the area more attractive to the user.

Inside the cloud there could be a cloud chat for the similar persons allowing for an ad hoc meeting possibility. When there are large events planned, the social cloud could include these events and provide a media for the people in the event to communicate and share thoughts or pictures and videos.

Routing with public transportation

A strongly desirable feature for AmiCarta would be the possibility to navigate via the map view. The users wished for not only car or walking navigation, but also cycling navigation and navigation that uses public transportation. The last one would have to have cooperation from existing public transportation timetable services, in Finland namely “Reittiopas.fi”.

All the users that were interviewed found that they would definitely use navigation that has been enhanced with public transportation. Navigation with routing that can be used only for cars, bicycles and walking wasn't as nearly as desirable.

Anonymous Instant Messaging reputation system

An excellent idea for AIM was the ability for users to rate the conversational partner with a rating system of either thumbs up or down or with a star rating system. This way good conversationalists would gain a better reputation and would be considered better mates. This could also keep users from being obnoxious to each other, since users with low ratings might not attract other people as easily.

The user's reputation could be displayed in the search results as well as when starting a conversation with a person that has searched for company and wanted to chat with the user. This way the user being summoned to chat might get some information about the other user and decide whether the user wants to chat with this person.

Map view indicators for new activity

A keen observation made by one of the interviewed users was that the map view of the mobile client did not contain any indicators for new activity within the system. By adding these indicators the user wouldn't have to go through the menus continuously to check if there are new IM messages or messages.

These indicators were added to the mock-up prototype as a result of this interview, and users tested since agreed that they were useful.

Service usage indicators for users

A point made about the map view said that there is no way to tell the difference between users that are logged in at the service at any time and the ones that are just being tracked, but are not actively present. The user suggested that different colors were used for those that are available for discussion and those that are just being tracked.

This idea was implemented to the service mock-up. The next user confused the colors to represent the users sex. The same user noted that it sounded more reasonable that the color represented the online status, and thought that it was a good idea. But it would require an explanation before it was correctly understood.

Anonymous Instant Messaging dog tagging users

A feature wished by the users was the ability to save some of the most interesting AIM results for future use, if the user doesn't have time to chat with them at the moment. This seemed like a nice feature, and would help keep the users from feeling “what if he/she was the spouse of my dreams and this stupid system won't let me chat with him/her now.”

The system could show these dog tagged users when searching for a new acquaintance as the first matches on the list, if they are using the system at the moment. Once the user has chatted with them, the dog tagged entries would be deleted from the results.

Anonymous Instant Messaging with anonymous profiles

The interviewed users felt strongly, that there wasn't enough of a hook to start just chatting away with a user the system has chosen for you. Instead, the users would like to have a little more information about the user the system has chosen for you to be more motivated to start a chat with the user.

Offering the user a description of the other user would motivate the user further to get to know the other user. This way the feature adds value to the system, because it makes AIM more desirable for the users.

New profile attributes

The users felt that their profiles should include lots of information. The attributes requested included pictures, career information, education information and human relationships information. Other requested attributes were a personal field for the user to express self, a list of the user's interests. Also the possibility to give a user's pictures different visibility options was requested.

The coolest feature that was suggested was that there should be a possibility to share status messages in audio or video format. This would work with the mobile client so that the user would record a status message and then post it. After that others could playback the status messages via the web and mobile interfaces.

These new profile attributes give the user more ways to share information about themselves which seems to be desirable.

Grouping options for friends

The users recognized a real need in grouping the user's friends. These were deemed necessary when using the mobile UI. The options that require choosing from a list of friends was thought to be too clumsy to choose every user individually.

Grouping the user's friends was thought to have a few benefits. First of all, you could have different groups that get to see your information and when they get to see it. For example, the users felt that there should be different groups for different kinds of friends. The user could have a group that includes the user's friends from work. Not necessarily will the user want to display location information to these friends when on holiday or not at work. Furthermore these groups could effect also the communication between the users. This would be useful when the user wishes to be uninterrupted by certain groups.

Another good purpose for having grouping options would be a situation when the user needs to create a shared calendar event or a new geofence quickly. This way the user could determine the events or geofences to groups easily.

Ad rating system

As a part of the profiled ads, the users would like to actually have more ways of making the ads even more profiled. One of the interviewed early adopters suggested that the user is given the choice of giving feedback to the ads.

The system would work as such. The user receives an ad and after viewing the ad, the user is asked for feedback. The feedback can be in the form of giving the ad a grade on how well it suited the user's needs, or just a simple plus or minus on whether the user found it useful or not.

The feedback given by the users will be used to better profile the ads in the future. This gives valuable insight to the advertisers on how well the automatic profiling works as well as profiles future ads to suit the users better.

Map view customization

A feature that was asked for was the ability to select which kind of information is displayed on the user's map. In technical terms, the system adds different information to it's own layers. This way all the types of information have their own layer. For example, the system can add the users on one layer and the trams in Helsinki on one layer.

The users wished to control which kind of information would be displayed to them. This could be easily done by giving the user the freedom to choose which layers the system displays to the user. This is a feature that is native to OpenLayers, so implementing it should be no problem.

Geocaching

The automatic geofences were a feature that could be used in geocaching. Geocaching is a hobby in which the players leave objects hidden somewhere outside. The players then take the GPS coordinates and post them on the internet. Other players will then try to find the cache with the coordinates and a small description of where the object is hidden.

The geofences in the system could be used to help the geocachers find caches made by others. The caches could be displayed publicly to the other users, so that they could view them on the map. The user would then be informed the specifics of the geocache, when the user arrives nearby the target. This message could be given as an automatic message, when the user triggers the geofence.

This could even be used to create a route that the player needs to follow in order to find the cache. The player would be given instructions on how to find the next cache when the player enters the previous geofence. The system could even be used to make finding the caches a speed race in which the speed is measured by the system, and it would start automatically when the user enters the first geofence, and turned off automatically when the user enters the last geofence.

Tracking children and the elderly

The last user in the tests suggested that the service could be sold to parents as a method of tracking their children or the elderly. This way the parents could always be sure that their child is safe, and where they are allowed to be. Also if the parents of a family have elderly folks, who for example, can't take care of themselves anymore, the service could be used to track them, to see that they are ok and are not wandering around anywhere.

The user pointed out, that for this kind of a service that ensures the safety of those related to the user, and would be something that the user would pay for.

6 Evolutionary pathway

The evolutionary pathway of the product describes different approaches for the concept. After the validation of the concept, there are several approaches to implementing the concept. To decide how to progress with the development of the concept, a focus group session was held with the development team. In the session, the results from the user tests were processed and the features that came up in the interviews, were clustered into entities that centered around a keyword. The results of the focus group session and the feature implementation timeline are presented in this chapter.

For the analysis of the features, a chart was needed on which features must exist before another feature is possible to exist. The required features are presented in figure 22.

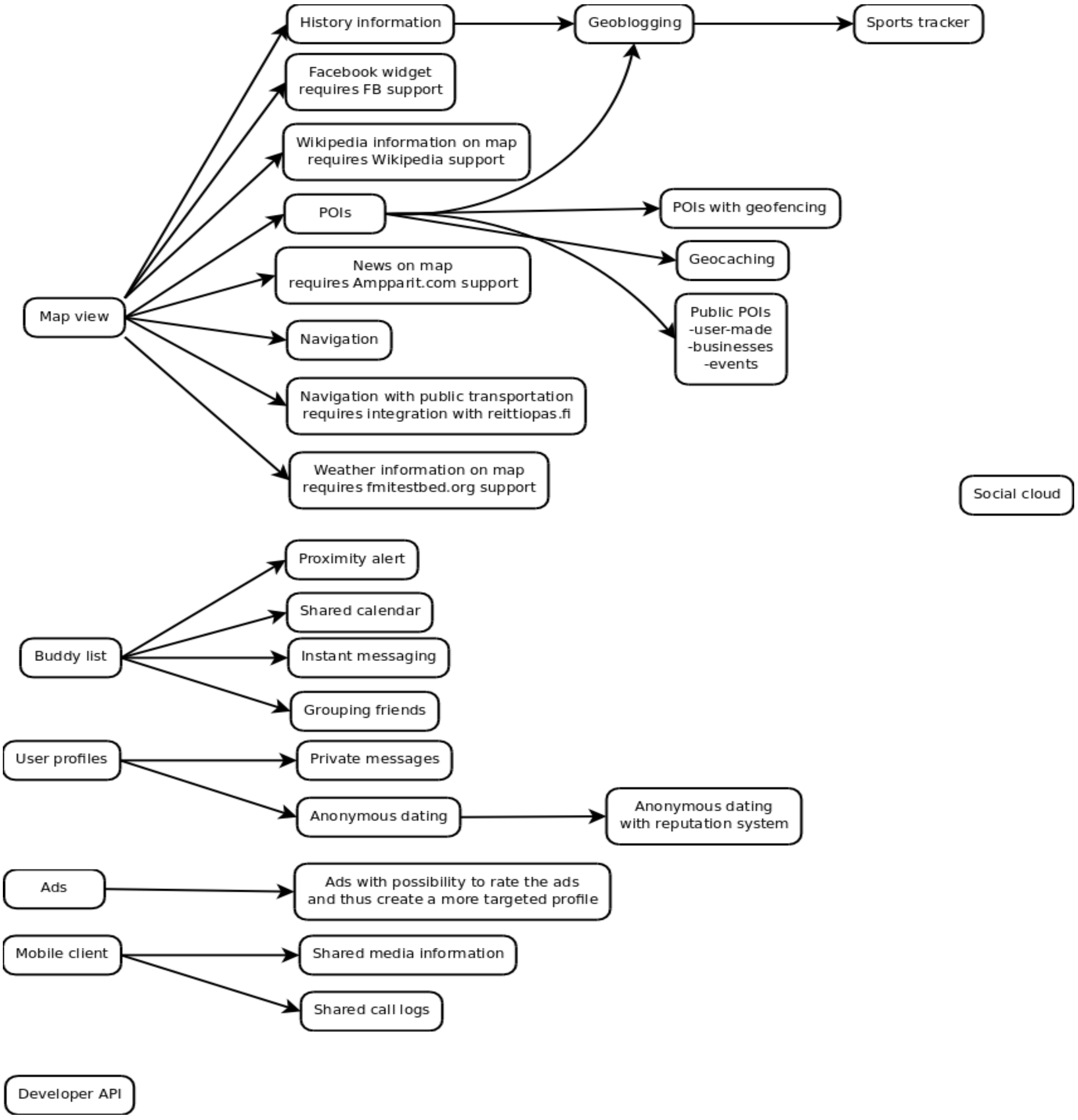

History information
Geoblogging
Sports tracker
Facebook widget requires FB support
Wikipedia information on map requires Wikipedia support
POIs
POIs with geofencing
Geocaching
Public POIs -user-made -businesses -events
News on map requires Ampparit.com support
Map view
Navigation
Navigation with public transportation requires integration with reittiopas.fi
Weather information on map requires fmitestbed.org support
Social cloud
Proximity alert
Shared calendar
Buddy list
Instant messaging
Grouping friends
User profiles
Private messages
Anonymous dating
Anonymous dating with reputation system
Ads
Ads with possibility to rate the ads and thus create a more targeted profile
Mobile client
Shared media information
Shared call logs
Developer API

Figure 22: Required features to create advanced features

6.1 Focus group results

The focus group session had four people present. In addition to the author, present were the project's head programmer, to evaluate the technological challenges in the project, the head graphic designer of the project and the CEO of Indagon to assess the business aspects of the features in the project.

In the focus group session, the team created different clusters around ten different keywords. The keywords for the clusters were the following:

- Distinct
- Fast technological implementation
- Revenue logic
- Attractive
- Communication
- Collaboration
- Technologically hard to implement
- Addictive
- Cheap to implement
- Expensive to implement

In chapter 6.1.1 the clusters of features are presented and evaluated using SWOT analysis. Based on these clusters, the timeline for implementation is created.

6.1.1 Clusters

In the focus group session, the team came up with ten different clusters. These clusters are presented here, and the clusters are evaluated using SWOT analysis, to find out what kind of strengths, weaknesses, opportunities and threats the cluster includes. The created clusters will then be used to create and evaluate the feature implementation timeline.

Distinct

The first cluster that was considered was the cluster that focused on the features that are different. The features the team picked for this cluster were the following:

- Weather data displayed on the map
- User profiles with grouping options
- Social cloud
- Tracking the children and the elderly
- Anonymous dating

These features would require the existence of the map view, the user profiles and the buddy list. The requirements for the social cloud don't exist, because it is not properly defined, what the feature does, and how it is done. The analysis for this cluster's features is presented in table 1.

Table 1: SWOT analysis for the "distinct" cluster

Strengths: • Lots of different dimensions in the service (Maps with friends and weather information, dating system, social cloud) • Tracking the children and the elderly useful for a certain group of users that take care of either their children or their elderly folks	Weaknesses: • Lots of different dimensions in the service, no clear focus for the system • No plan for the implementation of the social cloud • Implementation of the social cloud can take a lot of time • Requires many basic features to implement
Opportunities: • Weather data can be imported • Creates a feeling of safety with tracking the children and the elderly • Combine with existing dating services	Threats: • Importing the weather data does not succeed

Fast technological implementation

The cluster for fast technological implementation consisted mainly of features that exist at the moment or have been implemented before by the members of the team in other projects. The features in this cluster include:

- Map view
- POIs with geofencing
- Profile information
- Friends with grouping
- History information

These features have no impairing with the need for previous features, but are basic features. The POIs with geofencing feature requires the map view, which is included in the cluster already. The SWOT analysis for this cluster is presented in table 2.

Table 2: SWOT analysis for the "fast technological implementation" cluster

Strengths:	Weaknesses:
• Fast time to market • Map view already done • Experience on creating the cluster's features leads to less complications • Doesn't need other features to be implemented	• History information not wanted by the users • Not much content, the user can only view the user's friends on the map and use POIs to share information and work with the history information
Opportunities:	**Threats:**
• If the tracking works operator and client independent, there are no matching systems	• Operator-based tracking to other tracking services

Perhaps the biggest advantage in this cluster is that, the service could enter the market in some sort of beta-stage rather quickly, and would thus create awareness of itself. The flip-side of the coin is that, if the system has no useful content to the user, the early adopters might get bored of the service and that could result in bad word of mouth for the service.

Revenue logic

The revenue logic cluster focused on finding the features that would bring the service the cash flow and allow it to make profit. The features that were considered to help with the revenues were:

- Tracking the children and the elderly
- User profiles with ads based on location and profile
- Ads rated by users
- Anonymous dating
- Ad menu

The sources of revenue come from two different models, subscription fees and advertising. The features in the concept that are worth subscription fees are tracking the children and the elderly and the anonymous dating. The SWOT analysis for the cluster is presented in table 3.

Table 3: SWOT analysis for the "revenue logic" cluster

Strengths: • Cluster includes two different kind of ways to create revenue • Anonymous dating is interesting	Weaknesses: • The cluster itself is useless, it needs at least another cluster to be appealing, minus the anonymous dating, which could work by itself
Opportunities: • Advertisers could be really interested in the ad menu concept of a centered marketing place	Threats: • Customers might not be willing to pay for tracking the children and the elderly, needs to be researched further

Attractive

The cluster with the attractive features contained also some more abstract keywords, such as the overall user experience. The features and keywords listed here included:

- Graphical user interface
- User experience
- The map view
- Public transportation on map
- Friends visible on the map
- Easy to use technologically

These features would also require implementation of the buddy list to the system in order to determine, which users are the user's friends. The public transportation shown on map would require also integration with existing systems that plan routes for users, such as Reittiopas[21]. The SWOT analysis for the attractive cluster is presented in table 4.

Table 4: SWOT analysis for the cluster "attractive"

Strengths: • Strong user experience helps keep the gained users	Weaknesses: • Relies too much on just two features • Provides no means of communication between users
Opportunities: • Public transportation services could have interest in sharing their route information, since more visibility to them attracts more passengers to use public transportation	Threats: • No collaboration between the existing route planning services can be established

[21] http://www.reittiopas.fi

Communication

The communication cluster included all the ways of communicating between the users in the AmiCarta system. These features included the following.

- Instant messaging
- Private messaging with geofencing
- Geocaching messages
- Status messages with message history and audio and video status messages
- Anonymous dating
- Shared calendars
- Shared POIs

If this cluster was to be implemented, it would also require implementing the user's buddy list, the user profiles for anonymous dating and the map view for the shared POIs and private messaging with geofencing. A SWOT analysis of this cluster is presented in table .

Table 5: SWOT analysis for the "communication" cluster

Strengths: • Communication is an important aspect in a social media service, since it allows users to interact • Private messaging with geofencing allows users to reduce stress by not receiving work-related messages while spending their spare time and vice versa	Weaknesses: • The cluster requires nearly all features to be completed, so implementation takes lots of time
Opportunities: • The big social media services, like Facebook and Twitter don't offer audio and video status messages or private messaging with geofencing yet	Threats: • Existing dating services might be more attractive

Collaboration

The collaboration cluster is centered around the features that help the users work together. These features are listed below.

- Instant messaging
- Private messaging with geofencing
- Shared calendars
- Shared POIs

These features would require also the implementation of the map view for the geofencing and the shared POIs and the buddy list for the instant messaging. A SWOT analysis for this cluster is presented in table 6.

Table 6: SWOT analysis for the "collaboration" cluster

Strengths: • Multiple types of communication, synchronous and asynchronous	Weaknesses: • Requires the map view and the buddy list
Opportunities: • Collaboration with POIs and geofencing	Threats: • Specialized collaboration services

Technologically hard to implement

The technologically hard to implement features were evaluated by the team as those that would require the most work, and don't have a clear plan for implementation. The features are the following:

- Social cloud
- Rendezvous
- Geofencing
- Targeting ads and allowing the users to profile the ads
- Navigation

In this cluster the rendezvous navigation was something that was suggested in this stage. The basic idea in this is that when two users are moving, the system automatically calculates an optimal meeting place and provides directions to both users dynamically. The reason for this cluster is to evaluate whether or not these features should be implemented, and to assess when to implement such features. A SWOT analysis for this cluster is in table 7.

Table 7: SWOT analysis for the "technologically hard to implement" cluster

Strengths: • Social cloud a rare service • Rendezvous navigation a rare service	Weaknesses: • No plans on how to implement social cloud and rendezvous navigation
Opportunities: • Offering something the competition doesn't have creates potential for the service	Threats: • Competition figuring out how to implement these features first

Addictive

The focus group evaluated that there are three features that are more addictive than the other features in the service. These features are:

- Anonymous dating
- Having the friends displayed on the map
- Private messaging with geofencing

These features would require user profiles, the map view, instant messaging as well as private messaging. Also this cluster was evaluated with SWOT analysis, which is presented in table 8.

Table 8: SWOT analysis for the "addictive" cluster

Strengths:	Weaknesses:
• Addictive features cause the users to use the service more • More active users generate good word of mouth resulting in more users to the system • Versatile features	• Requires many features to implement the main cluster
Opportunities:	Threats:
• Addictive services create good word of mouth that helps the service gain users	• Users' interests can change quickly, usually faster than the techonology adapts to the needs

Cheap to implement

The features that are the cheapest to implement are the features that have the most work done already or require little time to implement. The cheapest features are:

- The map view
- Friends on the map
- Buddy list
- POIs

These features are very basic in the system and require no other features to be implemented before the feature is created, expect for the POIs, which require the map view to be shown. A SWOT analysis of this cluster is presented in table 9.

Table 9: SWOT analysis for the "cheap to implement" cluster

Strengths:	Weaknesses:
• Fast implementation • Provides background functionality for other clusters	• Little content for the user
Opportunities:	Threats:
• Fast entry to the market provides the service better chances of survival	• Lack of content makes the users forget the service

Expensive to implement

The features that require the most capital to develop are estimated to be the following:

- Map material other than OpenStreetMap material, such as satellite images
- Social cloud
- Navigation
- Shared calendar
- Anonymous dating
- Ad matching

A good deal of the cluster's expensiveness comes from the prerequisites of the cluster. The features in the cluster require the map view, the user profiles and the buddy list. In case of the map material, the material itself costs a lot of money, because the owner's of the material want compensation for using it. The cluster was evaluated using SWOT analysis, and the results are displayed in table 10.

Table 10: SWOT analysis for the "expensive to implement" cluster

Strengths:	Weaknesses:
• Social cloud a rare service	• Expensiveness to implement • Requires lots of features
Opportunities:	Threats:
• Equally expensive for all players to implement, • An opportunity to differentiate the service from other services	• No clear center for the service, hard to profile the service as fulfilling some need

6.1.2 Feature implementation timeline

In order to come up with a feature implementation timeline, it must be chosen, which clusters to implement and in which order.

The features were evaluated individually on how long it would take one employee to complete the feature from scratch to completely done. The evaluation was done mainly by the head coder at Indagon. The time that is required is presented as working months. This is not necessarily the best way to describe the amount of time needed, but the months should be considered as a relative amount of time. This means that when it is evaluated that feature 1 takes one month to complete and feature 2 takes two months to complete, the feature 2 requires twice the amount of time feature 1 required. The estimated times are presented in table 11.

Table 11: Required amount of time to complete a feature

Feature	**Time needed per employee in months**
Map view	0
Buddy list with grouping of friends	1
Instant messaging	1
Private messaging	1
POI with grouping of POIs and geofencing	1.5
News	1
Navigation (if solutions are purchased externally and integrated to the system)	3
Navigation with public transportation from Reittiopas.fi	1.5
Shared calendar (by implementing icalendar)	1.5
Shared media and call log	Requires a mobile client, no approximation
Profile information	0.5
Weather information from fmitestbed.org	0.5
Wikipedia information on map	0.5
Proximity alert	1
Social cloud	No specifications, no approximation
Anonymous dating	1
Ad engine with user rating	1.5
Geocaching	0.5
Facebook widget	1
History information	1.5
Developer API	No specifications, no approximation
Ad menu	3
Total	22.5

In order to create the timeline, the clusters that are going to be implemented need to be chosen. With ten clusters, there are several different combinations of the clusters. In choosing the clusters, there can be several different approaches. For this case, three different approaches are considered: the fastest to market timeline, a solid business logic approach and an attractive timeline to the user. The workloads for the timelines are estimated using two employees that implement the features.

Fastest to market timeline

The timeline for a fast entry to market is comprised of two main clusters. The clusters in question are the fast technological implementation and the cheap to implement clusters. These two clusters are chosen to be the center of the timeline, because the features in them are quite similar, and are fast to implement.

The features included in the clusters were as previously stated:

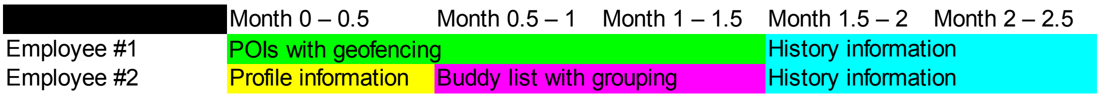
- The map view
- POIs with geofencing

- Profile information
- Buddy list with grouping
- History information

Now considering the results from the user tests, the history information is a very controversial feature in the system. If the history information is added, there needs to be a lot of customization options for the user, including the ability to deny saving the history information and deleting the existing history information at any time.

As presented in table 11, the development times for the different features would be for the map view, 0 months, for the POIs with geofencing and grouping 1.5 months, for the profile information 0,5 months, for the buddy list with grouping 1 month and for the history information 1.5 months. The total development time for this timeline would then be 4.5 months.

When assigning the feature development to occur simultaneously by the two employees, an optimum arrangement would be of the following kind presented in chart 3.



	Month 0 – 0.5	Month 0.5 – 1	Month 1 – 1.5	Month 1.5 – 2	Month 2 – 2.5
Employee #1	POIs with geofencing			History information	
Employee #2	Profile information	Buddy list with grouping		History information	

Chart 3: Development timeline for the fastest to market timeline

The product would thus be ready to launch in just two and a half months from the start of the development. In chart 3, the history information takes more time than the planned 1.5 months, it has been evaluated to take a little more time to allow the employees to coordinate their work efficiently and it is a “safe bet”. It is also worth noting that the map view is not presented in the timeline, because it is already done at the start of the development.

As stated in the SWOT analysis' for the two main clusters in tables 2 and 9, the main disadvantage of this implementation is the lack of content for the users. The user can view the user's friends locations on the map, create POIs that can include geofencing areas that can update the user's status automatically and alert the user of other users' whereabouts and also the user can view own history information.

The good qualities of the product is that it creates lots of the background functions needed to create more complex features, and the product remains easy to further develop. Also getting on the market raises awareness of the product and creates opinions on the service.

A solid business logic timeline

Creating a solid business logic centers around the presented ways of earning revenue in the revenue logic cluster. The cluster includes two different approaches to creating revenue, an ad based approach, and a subscriber based model. The features that support the ad based approach are the following.

- User profiles with ads based on location and profile

- Ads rated by users
- Anonymous dating
- Ad menu

The feature for a subscriber model is the tracking the children and the elderly feature that would require a monthly fee to work as a service.

The user profiles with ads based on location and profile with user rating to the ads takes a total of 2 months to complete and is based on the sum of the user profiles and ad engine with user rating tasks. The anonymous dating takes 1 month to complete and includes dating ads the users can send to each other for a small fee. In addition, the anonymous dating and the ads based on profiles require user profiles to be created, which takes 0.5 months of development time. The anonymous dating also requires instant messaging, which takes 1 month to develop. The ad menu was approximated to take 3 months to complete.

The ad based approaches require additional content that keeps the users using the service and receiving the adverts. So the solid business logic timeline cannot function only on the revenue logic cluster, but needs additional content. The timeline to create this ad functionality is displayed in chart 4.

	Month 0 – 0.5	Month 0.5 – 1	Month 1 – 1.5	Month 1.5 – 2	Month 2 – 2.5	Month 2.5 – 3	Month 3 – 3.5	Month 3.5 – 4
Employee #1	Instant messaging		Ad menu					
Employee #2	User profiles	Ads based on location and profiles with user rating				Anonymous dating		

Chart 4: Development timeline for the solid business logic timeline with ad based approach

For the subscription based business model, the tracking the children and the elderly consists of a few components. The system is required to have the map view, which takes no time to implement, and at it's most basic the buddy list with grouping to tell the system, which users the user has the right to track. The implementation of the buddy list is approximated to take 1 month of development time. A feature for efficient tracking of children and the elderly is the user's history information, so their whereabouts can be traced back. The implementation of history information is approximated to take 1.5 months of development time. The timeline for development is presented in chart 5.

	Month 0 – 0.5	Month 0.5 – 1	Month 1 – 1.5
Employee #1	History information		
Employee #2	Buddy list with grouping		

Chart 5: Development timeline for the solid business logic timeline with the subscriber model

The timelines for the business logic both create some sort of a service. The ad based approach creates a tool for anonymous dating and reviewing ads and searching for discounts by searching through the ads. The subscriber model creates a product that is used mainly to monitor those that need to be monitored.

The features in the concept show potential of being divided into different services. It remains open whether it would be better to have many differentiated services or to stick with the original concept of having a social media mashup that integrates features.

Attractive to the user timeline

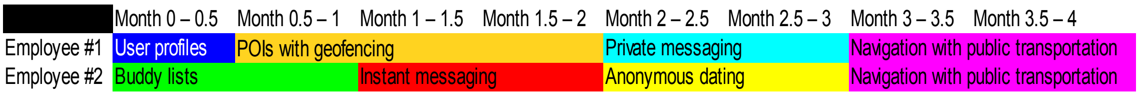
When creating an attractive product to the user, it must be considered which clusters were the clusters that would drive the most customers to the system. This is why the attractive timeline comprises of the clusters attractive and addictive.

These clusters included the following features:

- Anonymous dating
- Having the friends displayed on the map
- Private messaging with geofencing
- Graphical user interface
- User experience
- The map view
- Public transportation on the map
- Friends visible on the map
- Easy to use technologically

The features proposed here require in addition a few basic functions. Once again, the map view doesn't require more development time. The anonymous dating requires user profiles, which take 0.5 months to implement. Anonymous dating also requires instant messaging, which takes 1 month to implement. Friends displayed on the map require buddy lists, which take 1 month to develop. The private messaging with geofencing requires POIs with geofencing to be implemented, which takes 1.5 months of time.

The actual features in the clusters have the following development times. The anonymous dating needs 1 month of development, the private messaging with geofences takes 1 month and the navigation with public transportation 1.5 months. The timeline is presented in chart 6.



	Month 0 – 0.5	Month 0.5 – 1	Month 1 – 1.5	Month 1.5 – 2	Month 2 – 2.5	Month 2.5 – 3	Month 3 – 3.5	Month 3.5 – 4
Employee #1	User profiles	POIs with geofencing			Private messaging		Navigation with public transportation	
Employee #2	Buddy lists		Instant messaging		Anonymous dating		Navigation with public transportation	

Chart 6: Development timeline for the attractive to the user timeline

Such a timeline would provide an attractive and addictive service for the users. The timeline would create a service that offers multiple ways of communication. Private messaging with geofencing would help the users organize their communication needs and keep work separated from their free time. Instant messaging would allow instantaneous messaging between friends and provide a technological requirement for the anonymous dating service. Navigation with public transportation was evaluated as the best feature in the service in the user tests, and was considered something the users would use even if their friends didn't have the service, which is why it is something that is attractive and addictive.

Proposed feature implementation timeline

Since the service is desired to be a social media service, that has a free-of-charge possibility to join the service, the service needs to have a solid business logic and be attractive to the users in order to keep the users using the service. The timeline for the implementation of the service must then consist of a hybrid of these two timelines. For the business logic, it is needed to choose between the ad based approach and the subscription based approach. The original concept included a freemium business model. To implement a freemium model, there needs to be some extra content that is available only by subscribing to the service. This extra content could be tracking the children and the elderly. This way a person could integrate communication with the person's peers to a solution that provides safety to the person's family. Combining these two timelines, the features that need to be implented are listed below:

- Anonymous dating
- Having the friends displayed on the map
- Private messaging with geofencing
- Graphical user interface
- User experience
- The map view
- Public transportation on the map
- Friends visible on the map
- Easy to use technologically
- Ad menu
- User profiles with ads based on location and profile
- Ads rated by users
- History information

As a prerequisite, the map view is also needed, and is already implemented in the system. Also POIs with geofencing are required to implement to create a UI for the messaging with geofences. The timeline for creating this service is presented in chart 7.

	Month 0 – 0.5	Month 0.5 – 1	Month 1 – 1.5	Month 1.5 – 2	Month 2 – 2.5	Month 2.5 – 3
Employee #1	User profiles	Ads based on location and profiles with user rating				POIs with geofencing
Employee #2	Buddy lists with grouping		Instant messaging		Anonymous dating	

	Month 3 – 3.5	Month 3.5 – 4	Month 4 – 4.5	Month 4.5 – 5	Month 5 – 5.5	Month 5.5 – 6
Employee #1	POIs with geofencing		Private messaging		Ad menu	
Employee #2	Navigation with public transportation			History information		

	Month 6 – 6.5	Month 6.5 – 7
Employee #1	Ad menu	
Employee #2	Ad menu	

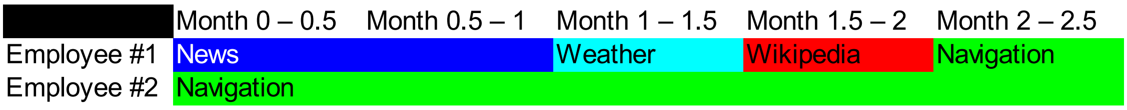
Chart 7: Proposed feature implementation timeline

When following this timeline, the free-of-charge part of the service would be ready to enter the market in just 5 months after the start of development, followed by the premium service after 6 months of development. In order to get the product faster on the market, it is proposed that it is released after the navigation with public transportation feature is completed, and the ad menu should be created once already on the market.

After the service has hit the market, there needs to be a plan on how to further develop the service to keep the users using it. The features that are “left over” from the features that came up during validation or existed prior validation include the following.

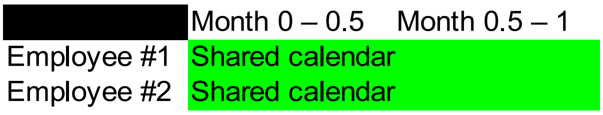
- News
- Navigation
- Shared calendar (by implementing icalendar)
- Shared media and call log
- Weather information from fmitestbed.org
- Wikipedia information on map
- Proximity alert
- Social cloud
- Geocaching
- Facebook widget
- Developer API
- Audio and video statuses

Out of these features, there are different paths to expanding the service. Some paths on development come from the nature of the features themselves. For example, it could be decided that the service needs to improve it's map services. Automatically this would mean that the features such as news on map, wikipedia information on map, weather information on map and navigation would be the features that are to be implemented. Other paths come from implementing existing feature clusters, such as collaboration, which would require the implementation of the shared calendar feature. Timelines for implementing the two presented paths are in charts 8 and 9.



	Month 0 – 0.5	Month 0.5 – 1	Month 1 – 1.5	Month 1.5 – 2	Month 2 – 2.5
Employee #1	News		Weather	Wikipedia	Navigation
Employee #2	Navigation				

Chart 8: Timeline for implementing further map functionality



	Month 0 – 0.5	Month 0.5 – 1
Employee #1	Shared calendar	
Employee #2	Shared calendar	

Chart 9: Timeline for completing the "collaboration" cluster

The actual needs of the users would need to be studied further once the product has entered the market to properly evaluate, which features in the service require improving.

7 Discussion

The validation of the AmiCarta concept proved to be useful. The focus of the system shifted from having a service that focuses on utilizing a map view with as many gadgets as possible to having a service that allows the user to communicate with other users easily as well as offering useful map tools.

7.1 Evolution of the concept in the validation process

The original concept of AmiCarta was centered around the user's location information. The features of the original concept had a sort of a shotgun approach with implementing as many features as possible to the service.

The following features from the original concept have been removed in the proposed timeline for the feature implementation of the actual product:

- News
- Shared calendar
- Developer platform

These features didn't belong in the chosen clusters for implementation. Instead, the following features were added to the feature implementation timeline:

- Private messaging with geofencing
- Navigation with public transportation
- Ad menu

These three features create a more solid core for the service in communication, map services and advertising. At the heart of the AmiCarta concept after validation is communication between the users. For this the concept offers several different ways, such as instant messaging, private messaging with geofencing properties on which messages to receive where and anonymous messaging for dating other users. The inclusion of geofencing to the private messaging was perhaps one of the best ideas in the validation, because it would help users not receive messages from work while on their spare time.

The bad results received for anonymous dating raised some questions. There could be a few reasons for this. First of all, four of the seven interviewed users were not single, and therefore didn't feel the need to use a dating service. Second, the basic functionality of the anonymous dating was something the users didn't feel sufficient enough to use. The users wanted to have a less anonymous dating service, that would give at least some information of the other user in order to get interested. So in a way, the service wasn't finished enough to make an impression on the interviewed users.

Another reason for the relatively bad results for the anonymous dating could be that anonymous dating including location information is something that isn't common for the users. Since they have no experience of it, they have a hard time relating to it. Previous research done in an anonymous dating service showed that users gradually used the dating service more and more [Burak et. al, 2004]. At the start of the service, only 40%

of the users used the anonymous dating feature, while 13 months later over 70% of the users used the feature, and it was clearly the most popular feature in the service. In comparison to IM with the user's friends in the service, at start 60% of users used it and 13 months later only 20% of the users used IM with their friends. In Burak et. al's research the feature gained popularity throughout the research which would indicate that in AmiCarta, the feature would only gain momentum through the product's life cycle.

The anonymous dating evolved significantly during the validation. In the beginning of the validation, it only included the possibility to chat with other users that were online anonymously. Requested features included anonymous profiles, dog tagging users for later contacting, rating conversational partners etc. The feature evolved in a direction of more sharing. The concept of anonymity was bended a little to disclose something of the user, but still keep the information that could identify the user hidden.

Private messaging evolved with a small but significant addition to the feature. The users wished to have a possibility to use categories of the users in conjunction with geofences to have certain messages only appear in certain places. If such a feature would allow integration to the user's different email addresses, the user would benefit from having all the messaging integrated to the system, with appropriate messages in appropriate areas. For example, the user could configure the system to display work-related messages only when at work. This would make it easier to detach from work, when not at work, which would reduce the user's stress.

Geofencing with private messaging was something that hadn't been considered before the validation. This suggests that geofencing might have other use cases that could prove to be useful.

Advertising in the system went through considerable revision. The original concept of having discount codes presented in other media and having the users enter the codes to the system seemed pointless to the users. The evolution to having a centered system with all the discounts listed and having the option to search through the ads made the system much more like the Internet.

Considering the business model of the product, there was a slight shift in what the premium users of the system receive in comparison to the original concept of having premium members just not receive adverts. After the evolution, it is considered that the premium version allows users to use the system ad-free as well as tracking their children and elderly on consent. Also the premium users will have access to the history information of the tracked persons. This gives additional value to the premium model, and makes it into a product of it's own, that can be added to the free of charge service.

7.2 How the methods served the study

The mock-up prototype proved to be very agile to develop further. The prototype consisted of only images made by computer and a slide show, that triggered transitions on different parts of the screen. Making adjustments to the mock-up proved to be easy, only the images needed to be edited and the corresponding triggers moved to their correct places. The prototype clearly helped the users understand the concept, since the average score for understanding the concept was 4.29.

The methods used in the study generated mostly qualitative results. The only quantitative results came from the rating of the features in the system using the likert scales. The results obtained from the questionnaire might not be completely reliable. The grades given to the feature might have depended too much on the completeness of the feature in the mock-up. If the feature was not completed to satisfaction, it received lower scores, such as the anonymous dating. Also, if the users had further ideas to the feature, the scores given were lower.

The actual problem might not be in the grading of the features, rather than the problem being in the effect the presentation had on the grading. The population for a quantitative research was far too small anyway to produce scientifically valid results in the grades. The grades for the features are more directional, or “in the ball park” so to speak.

The open interview in the study gave perhaps the most important information, ideas on what kind of new features would be good for the service. In the interview the users were also presented with the new features that other users suggested.

The use of a prototype had it's pros and cons. On the pros side, the prototype gave the users lots of ideas on what to improve in the existing system. As a negative effect, once the users had seen the prototype it was harder for them to come up with new features to the service. With the prototype, the users also focused on improving the existing user interface, which is by itself fine, but not at the core of the study.

Affinity analysis helped with giving an overall picture of the concept and the required work to different features. The main help to the project in affinity analysis was the categorization of the results of the study, new features didn't emerge in the process.

Using the focus group session to cluster the features of the concept gave clarity to the feature implementation timeline. Seeing the concept as a group of wholes gave new insight into what should be considered the core of the concept. And it all came down to communication between the users, and not in maximizing the use of the location information. It's the other users and the content they create that keeps the service interesting, not the technology behind it.

8 Conclusions

This chapter concludes the study with the answers to the research questions as well as highlighting some of the results of the study. This chapter also presents what actions need to be taken, and whether or not further studies on the subject are required.

8.1 Study results

Chapter 5.3 contains the results of the study detailed at length. In this chapter, the main results are discussed.

The main results of the study are the findings on how to implement the service. As a conclusion on the most important aspects of the service, two key factors are presented: the location information and communication between users. These factors dictated the evolution for the service.

For the business logic, the study showed that the users are more interested in mobile marketing and receiving ads, if the adverts are highly targeted. This is an extremely good quality for the service, since the more profiled the ad is, the more effective the ad is and since the user is more interested in the ads, the user is willing to accept more advertising.

A key ingredient in making the service successful is enabling customization in all aspects of the service. The user wants to be in control of the service and it's features. The desire to customize can be seen for example in grouping users and wanting to remodel the main menu of the client to suit the user's needs.

8.2 Answers to the research questions

The research questions were all answered at least to some extent. The answers are narrowed down here to include the key answers to the questions.

1. What kind of a method will be used to validate the AmiCarta concept?

The AmiCarta concept was validated using a mock-up prototype and an open interview and a questionnaire. The mock-up prototype consisted of a hybrid between paper in screen prototyping and wizard-of-oz prototyping. The paper in screen part came from iterating the prototype between the tests and showing the prototype using a computer. The wizard-of-oz part of the mock-up was the functionality of the mock-up, the mock-up reacted to the user's actions in the prototype, and changed the views in the prototype to correspond with the actions. When the users were getting to know the prototype, they were encouraged to voice their opinions when using the prototype. This gave a contextual inquiry type of part of the interview.

The open interview was conducted to get the users to tell their views on the existing features and to get new ideas. The questionnaire asked the users to fill out grades for the existing features to get an idea on how the users liked the features.

2. How will the chosen validation methods affect the results of the study? What are the nature of the results of the chosen validation methods?

The chosen methods define the type of the results. The results can either be qualitative or quantitative. Qualitative results are results that aren't easily quantified and usually answer questions openly. Qualitative results on the other hand are results that answer a specific question with a short and precise answer. Quantitative answers are usually figures or values in a multiple choice question.

In the validation, the use of a prototype might have affected the results in making it harder for users to focus on coming up with new features, because they were occupied with the user interface and the little flaws in it. The use of a prototype gave the users a feeling, that the prototype represented the final product one to one. This was part of the reason the answers didn't revolve around finding new and exciting features to the product.

Out of the used methods, the review of the prototype and the open interview gave qualitative results. This means the results aren't something that can be quantified to for example a grade, but give insight to how the system works and what should be improved from the user's point of view. The questionnaire gave results that were quantitative.

The fact that the majority of the received results are qualitative is a good thing for a prototype at this stage. Since there was no consensus on what the service should be before validation, qualitative results gave good descriptions on what needed to be improved. If the study had focused on quantitative results, the concept would have needed to be readier, otherwise, the concept could have failed on bad implementation of a prototype or on bad choice of features for the concept. Also a quantitative study would have required very different methods and a larger sample in order to be justified.

3. How can AmiCarta be linked to other social media services?

As a social media service, AmiCarta is a social network as well as a forum for sharing points of interest and other content. Ways of sharing the information created in the AmiCarta system included creating an application for Facebook that would show the user's location on a miniature map. Also users thought that using the shared calendar should be linked to other services that include calendar information.

The interviewed users weren't that keen on having the service expand to another social media service, and the majority found no use in that. In order to get better results, the topic should be studied further.

4. What kind of choices are there for the evolutionary pathway of the concept and what defines the path chosen?

For the choice of an evolutionary pathway, the results of the user tests were studied and analyzed. Chapter 6.1.1 discusses the possible clusters of features for implementing the service. The factors that chose the evolutionary pathway are the appeal of the feature to the users, the possibility and required amount of time for implementation and the effect on the business model of the service.

The chosen clusters had very clear priorities. The concept needs to attract the users to use the service, which is the most important criteria. This is because without users a social network isn't interesting to advertisers. Also it needs to have a solid business side to it in order to keep the service attractive to the company that keeps it running. The chosen pathway satisfies both criteria, so it has real potential in being a good service.

5. How is the concept molded by the results of the validation?

At the start of validation, the concept had no clear focus, it was just a bunch of features collected together. At the end of validation, it is clear, that the focus of the concept is now communication. This was made possible with just a few adjustments to the concept, and taking into account the numerous suggestions for improvement offered by the users. The users felt that news didn't fit the concept perfectly, so the feature was removed from the initial concept and needs further studies on how to integrate it better to the concept. Also the developer platform was taken under review and needs to be thought through. The shared calendar was also removed from the concept to be implemented later due to technological constraints.

The included features to the concept were private messaging with geofencing properties that enable the users to control their messaging. This improves the communication experience in the system. The reason for adding navigation with public transportation is that the users found it to be something that would be attractive enough to lure the users into using the service even if their friends didn't use it. The third added feature, the ad menu is an extension on the original idea of having discount codes, and provides the service with a more solid business logic, and would enable the service to go on by giving advertisers more reasons to advertise in the system.

8.3 Further actions

The study proved that the users were interested in sharing their location information to a select number of friends given the right amount of tools to protect their privacy. The concept can be implemented to some extent, but also further studies are required.

The news feature of the system was something the users couldn't see working. It would require a working demo and more studies on users to determine what to show on the map view, and how.

Also further studies should be conducted on linking the service to other social media. The topics for this research should include what kind of information the service provides users are willing to share in other media also, how the sharing should be done and if there are some privacy issues to sharing the information in other media.

Some of the users were concerned with losing their privacy, if they were to be tracked in a service like this. Further studies on how the users' privacy is protected should be conducted.

On the business side of the concept, it would need to be studied what kind of a price the advertisers are willing to pay for a location-based and well profiled ad. This greatly affects the possibilities of implementation of the concept. The amount of revenue created with the adverts should compensate the costs of the system with the revenue received from the premium users of the system. There definitely is room for research on the economy side of the project, for example on what is the expected ratio of regular users vs. the amount of premium users, and what are the break-even numbers of users and advertisers.

9 References

[Bhargava, 2006] Bhargava, Rohit, 5 Rules of Social Media Optimization (SMO), 2006, available: http://rohitbhargava.typepad.com/weblog/2006/08/5_rules_of_soci.html, retrieved 11.11.2008

[Bolchini et. al, 2009] Bolchini, Davide, Pulido, Diego, Faiola, Anthony, "Paper in screen" prototyping: an agile technique to anticipate the mobile experience, 2009, available: http://portal.acm.org/citation.cfm?id=1551986.1551992, retrieved 1.4.2010

[Briscoe et. al, 2006] Briscoe, B., Odlyzko, A., Tilly, B., Metcalfe's Law is Wrong, IEEE Spectrum, 2006, available: http://spectrum.ieee.org/computing/networks/metcalfes-law-is-wrong, retrieved 2.3.2010

[Boyd et. al, 2007] Boyd, Danah M., Ellison, Nicole B., Social network sites: definition, history and scholarship, 2007, available: http://consommacteurs.blogs.com/files/socialnetworksites_boyd-ellision_2007.pdf, retrieved 9.4.2010

[Burak et al, 2004] Burak, Asaf, Sharon, Taly, Usage patterns of FriendZone – Mobile Location-Based Community Services, 2004, Mobile and Ubiquitous Multimedia; Vol. 83, Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia

[Clarkin et. al, 2007]Clarkin, Larry, Holmes, Josh, Enterprise Mashups, 2007, available: http://msdn.microsoft.com/en-us/architecture/bb906060.aspx, retrieved 31.3.2010

[Connolly, 2008] http://connollyshaun.blogspot.com/2008/05/7-key-attributes-of-social-web.html

[Creative commons, 2010] Creative commons, Creative commons – attribution – share-alike 2.0 generic, 2010, available: http://creativecommons.org/licenses/by-sa/2.0/, retrieved 17.3.2010

[Crupi, 2008] Crupi, John, Warner, Chris, Enterprise Mashups Part 1: Bringing SOA to the people, SOA Magazine Issue XVIII, May 2008, available: http://www.soamag.com/I18/0508-1.asp

[Cunningham, 2002] Cunningham, Howard, What is Wiki?, 2002, available: http://www.wiki.org/wiki.cgi?WhatIsWiki, retrieved on 20.1.2009

[EMMUS, 1999] European Multimedia Usability Services, Introduction to ISO 13407, 1999, Figure 1, available: http://www.ucc.ie/hfrg/emmus/methods/iso.html, retrieved on 4.2.2009

[European Commission, 2006] European Commission, SWOT (Strengths Weaknesses Opportunities and Threats) Analysis, 2006, available: http://forlearn.jrc.ec.europa.eu/guide/2_design/meth_swot-analysis.htm, retrieved 31.3.2010

[Facebook, 2009] Facebook, Tilastot | Facebook, 2009, available: http://www.facebook.com/press/info.php?statistics, retrieved on 29.9.2009

[Graham, 1999] Graham, Brad, Friday, September 10, 1999, 1999, available: http://www.bradlands.com/weblog/comments/september_10_1999/, retrieved 23.1.2009

[GSM Association, 2003] GSM Association, Location Based Services, 2003, available: http://www.gsmworld.com/documents/se23.pdf, retrieved 9.4.2010

[Jones et al., 2004] Jones Q., Grandhi S., Terveen L., and Whittaker S.,(2004). People-To-People-to-Geographical-Places: The P3 Framework for Location-Based Community Systems. Journal of Computer Supported Cooperative Work. Kluwer Academic Publishers.

[Kaplan et. al, 2010] Kaplan Andreas M., Haenlein Michael, (2010), Users of the world, unite! The challenges and opportunities of social media, Business Horizons, Vol. 53, Issue 1, p. 59-68.

[Lee, 2008] Lee, James, World of Warcraft hits 11m subscribers, 2008, available: http://www.gamesindustry.biz/articles/world-of-warcraft-hits-11m-subscribers, retrieved 14.11.2008

[MarketingVOX, 2008] MarketingVox.com, Young early adopters don't drive Twitter's traffic, available: http://www.marketingvox.com/young-early-adopters-dont-drive-twitters-traffic-043778/?utm_campaign=newsletter&utm_source=mv&utm_medium=textlink, retrieved 22.4.2009

[Mayfield, 2006] Mayfield, Anthony, What is social media?, iCrossing, 2006, available: http://www.icrossing.co.uk/fileadmin/uploads/eBooks/What_is_Social_Media_iCrossing_ebook.pdf

[Nielsen, 2001] Nielsen, Jakob, Usability Metrics (Jakob Nielsen's Alertbox), 2001, available: http://www.useit.com/alertbox/20010121.html, retrieved 5.5.2009

[Nielsen, 2006] Nielsen, Jakob, Participation Inequality: Encouraging More Users to Contribute, 2006, available: http://www.useit.com/alertbox/participation_inequality.html, retrieved 20.4.2009

[O'Neill, 2009] O'Neill, Nick, Farmville blows past 70 million monthly users, 2009, Allfacebook.com, available: http://www.allfacebook.com/2009/12/farmville-blows-past-70-million-monthly-users/, retrieved 12.4.2010

[Rappa, 2010] Rappa, Michael, Business models on the web, 2010, digitalenterprise.org, available: http://digitalenterprise.org/models/models.html, retrieved 2.3.2010

[Scanlon, 1998] Scanlon, Tara, Paper Prototypes: Still our favourite, User Interface Engineering, 1998, available: http://www.uie.com/articles/paper_prototyping/, retrieved 8.12.2009

[UsabilityNet, 2006], UsabilityNet, Methods: ISO 13407, 2006, available: http://www.usabilitynet.org/tools/13407stds.htm, retrieved: 3.2.2009

[Yu et al., 2008], Yu, J., Benatallah B., Casati, F., Daniel, F., Understanding Mashup Development, IEEE Internet Computing, vol. 12, no. 5, pp. 44-52, Sep./Oct. 2008, available: http://www.computer.org/portal/web/csdl/doi/10.1109/MIC.2008.114

[Zuckerberg, 2010] Zuckerberg, Mark, Six years of making connections, available: http://blog.facebook.com/blog.php?post=287542162130, retrieved 31.3.2010

[Zuckerberg 2, 2009] Zuckerberg, Mark, A great start to 2009, 2009, available: http://blog.facebook.com/blog.php?post=46881667130, retrieved 22.1.2009

[Zuckerberg 3, 2009] Zuckerberg, Mark, 200 million strong, 2009, available: http://blog.facebook.com/blog.php?post=72353897130, retrieved 9.4.2009

10 Appendices

The appendices for the thesis are presented in this chapter.

Appendix A.Questions for the validation interview

This appendix presents the list of questions the interviewer asked in the user tests.

(To be asked after the user has tried the mock-up thoroughly)

Intended questions are probes on previous questions, if the user doesn't seem to understand the asked question

Do you have any questions on the concept?

Was there something that you didn't understand?

What kind of thoughts did the concept evoke?

> Did you get ideas on improvement or new features?

To what purpose could you use the service depicted in the concept?

What did you think of the concept?

> Was it fun or pointless or useless?

Could you see yourself using a service like this?

> What would be required in order for you to use it?
>
> Would it be important if your family or friends used the service?

Is the service worth having to receive adverts?

> Would you rather pay to use the service?
>
> If you would rather pay, what kind of a sum would be appropriate?

What did you think about seeing your friends on a map view?

> Was it fun or pointless or useless?

What did you think about instant messaging with your friends?

> Was it fun or pointless or useless?

What did you think about creating and using points of interest?

> Was it fun or pointless or useless?

Are you interested in sharing your calendar with your friends?

> What kind of markings would you share with your friends?

Would it be useful for you to plan routes while on the move?

> Would you need information on public transportation? If so, what forms of transportation?
>
> In what kind of a situation would you need route planning?

How would you use traffic information to your advantage?

> Do you see traffic information as an integral part of the service?

What are your views on anonymous dating?

Would you need more specific criteria in searching for a date?

What about date advertising? The possibility to promote yourself to potential dates?

Are you intersted in anonymous instant messaging?

Would you trust the service to find the right kind of dates for you?

What kind of profile information would it require that you'd trust the service to find the right dates?

Would you use private messaging in the service?

Would you like to leave messages to friends that they would receive once they entered an area that you had specified before-hand?

What kind of profile information would you want in the service?

Location information, pictures, geoblogging?

How would you like to filter news in the system?

By location, by category?

How do you feel about the service recording your location history?

Does it offend your privacy?

Are you interested in seeing your friends' history information?

Do you think that it is useful to view location history?

Would it be useful for you to be able to create geofences that would change your status when you entered the area?

What would you use the feature to?

Would you like to use geofences in tracking your friends?

Would you be interested in using geofences for events you are planning?

Would you like a message when your friends are at some range from you?

What would be a suitable range?

Would you like to share your location information in some other form of social media?

For example, a blog or your facebook profile?

What did you consider best about the concept?

What did you consider the worst in the concept?

Is there something you'd like to revise in the concept if you had the chance?

Appendix B.Questionnaire for the validation interview

This appendix presents the questionnaire the users filled out in the test.

Questionnaire about the features of AmiCarta

Background information

Sex male / female

Feature	**Dislike**	**1**	**2**	**3**	**4**	**5**	**Like**

The concept

The map view	Dislike	1	2	3	4	5	Like
Anonymous dating	Dislike	1	2	3	4	5	Like
Profile information	Dislike	1	2	3	4	5	Like
Shared calendar	Dislike	1	2	3	4	5	Like
Traffic information	Dislike	1	2	3	4	5	Like
Public transportation	Dislike	1	2	3	4	5	Like
Discount codes	Dislike	1	2	3	4	5	Like
Points of interest	Dislike	1	2	3	4	5	Like
Geofencing	Dislike	1	2	3	4	5	Like
Sharing POIs	Dislike	1	2	3	4	5	Like

Buddies

Friends on map	Dislike	1	2	3	4	5	Like
IM with friends	Dislike	1	2	3	4	5	Like
Private messaging	Dislike	1	2	3	4	5	Like

News

Local	Dislike	1	2	3	4	5	Like
Categorized	Dislike	1	2	3	4	5	Like

Statements about the test

I understood the concept	Disagree	1	2	3	4	5	Agree
The product seemed easy to use	Disagree	1	2	3	4	5	Agree
The prototype seemed polished	Disagree	1	2	3	4	5	Agree
The UI was reasonable	Disagree	1	2	3	4	5	Agree

Appendix C.List of functions in the mock-up

This appendix shows the main functions in the mock-up prototype as well as what the user can do in the mock-up prototype.

1. Map view
   1. Find out another user's status without leaving the map view
   2. Find out a traffic announcement's message
   3. Find information about the public transportation
2. Friend list
   1. Check out a friend's profile
   2. Start a conversation
   3. Add a buddy
3. Messages
   1. Read a message
   2. Write a message
4. Anonymous Instant Messaging
   1. Check out the settings
   2. Find a chat buddy
5. Check out your user profile
   1. Tell the interviewer what kind of information you would like to display in your profile
6. Calendar
   1. Check out a date
   2. Make a new mark
7. Traffic information
   1. View the menu
8. Settings
   1. Find the settings
   2. Would you need other settings?
9. Routing
   1. Would you need other settings?
10. Discount codes
    1. What kind of codes would you find most useful?
11. Local news
    1. View a piece of news
12. POI
    1. View a POI
    2. Create a POI

Appendix D.Observations from the user tests

In this appendix the observations made during the tests are collected in one list and shown in categories that were categorized with affinity analysis.

Interview number	Observation number	Group	Observation
1	**0**	**A**	**Dates**
1	1	A	The user wouldn't use AIM dating service
1	2	A	The user still might consider using AIM
1	3	A	Criteria for searching dates: ikä, region
1	4	A	The user doesn't believe that the service will find the right matches for the user
1	5	A	The user doesn't want dating ads
1	**0**	**B**	**Map markers**
1	6	B	Markers useful when meeting friends
1	7	B	Sharing markers is only interesting if the user's whole circle of friends uses Amicarta
1	8	B	Would use automatic statuses for sports hobby
1	9	B	Would share parties and sports events
1	10	B	Wouldn't use traffic announcements
1	**0**	**C**	**Friends**
1	11	C	An appropriate distance for proximity alerts: 100m
1	12	C	Wouldn't use asynchronous messaging
1	13	C	Leaving messages on the map is too complicated, to meet friends, landmarks do the trick
1	14	C	Implement nearest landmarks to the service?
1	**0**	**D**	**Conditions to usage**
1	15	D	Too many features in one service
1	16	D	If the user's friends would use the service, the user would use the service
1	17	D	Receiving SMS ads is an ok price for the service
1	18	D	“code” is a bad term for hunting discounts
1	19	D	News must be displayed categorically
1	**0**	**E**	**User Interface**
1	20	E	The user interface needs more colors
1	**0**	**F**	**Routing**
1	21	F	Implement cycling maps to the service
1	22	F	Navigation with public transportation routes would be enough for the user to use the service

1	23 F	Plans routes once or twice per year
1	24 F	Would be most helpful if the user got lost when cycling
1	**0 G**	**History information**
1	25 G	Geoblogging is a nice feature
1	26 G	Doesn't want to see other user's history information
1	27 G	History information doesn't offend the user's privacy
1	28 G	Would like consent from other users to using geofences
2	**0 E**	**User Interface**
2	1 E	On the map, the streets need to be the same color, different colors confuse too much and get mixed up with the buildings and other map icons
2	2 E	The map view needs an indicator for activity in the system while viewing the map
2	3 E	The user's name in the status bubble on the map is good
2	4 E	The user's map icon could be the user's profile image
2	5 E	The term for user's visibility on the map to the user's friends needs a new term. Current: “paikka näkyvissä”
2	**0 H**	**Menu system**
2	6 H	The icon for messages needs to view “Messages (x)”, where x is the number of unread messages, not the number of all messages
2	7 H	If the menu gets icons, they need to be standard icons that represent the function well
2	8 H	In the map view, the buttons for “close” and “main menu” need to switch places for consistency throughout the system
2	9 H	For the buttons, simple text is better than icons
2	10 H	Remove traffic announcements from the main menu
2	11 H	Map markers is a bad term in finnish, “karttamerkinnät”
2	12 H	Remove AIM dates from the main menu
2	13 H	Get a scrollbar to the main menu
2	**0 I**	**Map**
2	14 I	The map view has to remember where the user left off when the user uses other functions and then returns to the map view

2	15 I	How is the map centered?
2	16 I	Get ad messages on the map
2	17 I	The mobile UI needs mostly the map, and everything other can be done in the browser UI
2	**0 J**	**“Close” button**
2	18 J	The close button is too handy in every screen, the user wants to touch it
2	19 J	The close button needs to be smaller than the other buttons
2	20 J	Group the close button away from the other buttons
2	21 J	Close = exit, Poistu = Sulje
2	22 J	The close button needs to be separated from the other buttons
2	**0 K**	**Settings**
2	23 K	Why does the user need to save the dating settings, why can't they save automatically?
2	24 K	The user should be able to share and edit pictures in the mobile UI
2	25 K	Remove the settings from the main menu, and put under the profile
2	26 K	Move the dating settings under the profile
2	27 K	GPS setting is unnecessary
2	**0 L**	**Map and calendar markings**
2	28 L	Display the name of the user who created the marking
2	29 L	There are many steps in making a simple marking
2	30 L	Add routing instructions to the markings
2	31 L	Make the calendar markings the ones that are complex, and the map markings simple
2	32 L	Time stamps to map markers, remove old ones from the map
2	**0 M**	**Communication**
2	33 M	Show users that are available for IM in a different color that those that are just being positioned
2	34 M	Make the proximity alert have an option to turn it on and off
2	35 M	For IM, send the chat as a message, if the other user leaves the system
2	**0 N**	**Dating privacy**
2	36 N	When searching for dates, show an anonymous profile of the user
2	37 N	AIM: Show the user's profile image

2	38 N	Lose the profile compatibility scale
2	39 N	Give the user more information on the possible date, for example, name or alias
2	40 N	Give an ability to dog tag interesting partners for later contact
2	41 N	The dog tag could prompt the user for permission to get in touch later
2	42 N	Dating: show the conversational partner's gender
2	**0 D**	**Conditions to usage**
2	43 D	Geofence tracking does not violate the user's privacy
2	44 D	Could use the service without friends that use it, would use navigation and calendar
2	45 D	Skeptical about fiddling with the user's mobile phone
2	**0 O**	**Dating controllability**
2	46 O	The ability to emphasize some search criteria
2	47 O	Dating ads are ok, 1€ would be a good sum, 10€ would be too much
2	48 O	The user should make a small description of self for the dating service
2	49 O	Dating profile needs long lists of what the user likes and dislikes to create better profiles
2	50 O	The ability to control the distance between the AIM search results and the user
2	51 O	Skip the dating settings every time the dating searching starts
2	**0 P**	**Bad features**
2	52 P	No history
2	53 P	No need for asynchronous messaging
2	54 P	No news feed
2	55 P	Replace discount code system with just offering the discounts
3	**0 Q**	**Networking**
3	5 Q	New criteria for finding friends: user id
3	26 Q	Sharing with a facebook widget would bring more users
3	27 Q	Sharing with a facebook widget would not add value to the user
3	28 Q	Sync the user's facebook and amicarta information
3	25 Q	Amdahl's law, more users brings more value to the network

3	**0 R**	**Sharing information**
3	42 R	User would share events with the shared calendar
3	35 R	The ability to choose which users or groups can see which of the user's pictures
3	33 R	Profile needs a field of self-expression
3	41 R	Not sure if users would use a shared calendar
3	34 R	Profile needs the user's interests
3	**0 S**	**Shared calendar**
3	2 S	Sharing a marking to an avec
3	3 S	Ability to control to how many the user can forward the marking
3	10 S	A link to the user's profile to the calendar marking that the user has made
3	**0 T**	**Privacy**
3	39 T	History information is ok, if it is used anonymously
3	50 T	The user is somewhat offended by saved history data
3	51 T	The user is interested in seeing other users' history data
3	52 T	The user feels that there is additional value in the history data
3	38 T	The user would be more comfortable with history data, if there was a period of time that could be set that and after it, the history data was deleted
3	**0 U**	**Searching for dates**
3	46 U	The system needs more criteria for searching for partners, at least age
3	31 U	A reputation system in which the users would rate each others on manners
3	32 U	A reputation should be given to users with for example stars
3	**0 V**	**Requirements for dating with AIM**
3	47 V	The user could try AIM dating
3	45 V	A dating service would be cool, if there are enough users for it
3	48 V	The user doesn't trust that AIM finds the right people
3	49 V	AIM needs user's interests
3	**0 F**	**Routing**
3	11 F	The user has no feelings to where the system should ask for the chosen method of transportation

3	12 F	Routing needs to have the current address as a default on point of origin for the route
3	44 F	Traffic announcements are important to the user
3	43 F	Uncommon situations are an important use case for routing
3	**0 M**	**Communication**
3	30 M	The system should include group messaging
3	22 M	Messages that appear to the recipient when in given coordinates, a “surprise” for the user's friends
3	9 M	The possibility to group friends
3	**0 W**	**Ad customization**
3	18 W	If the user is not interested in an arriving ad, don't show ads of the same category to the user again in a short period of time
3	29 W	Rather ads than having to pay for the service
3	**0 X**	**Ad frequency**
3	16 X	Two ads per day by SMS is ok
3	17 X	Over five or more ads per day or ads that arrive rapidly in a short period of time would annoy the user
3	**0 Y**	**Ad comparison**
3	19 Y	Compare prices between stores
3	**0 Z**	**Geofences**
3	53 Z	Geofences are a good idea
3	55 Z	Geofence alerts need grouping of users
3	20 Z	Turn off user's positioning by entering or exiting a geofence
3	4 Z	Status changes by leaving geofences
3	14 Z	No possibility for drawing a geofence is necessary
3	54 Z	Geofence could change the user's profile
3	13 Z	Grouping of map markings
3	15 Z	What about overlapping geofences?
3	**0 G**	**History information**
3	37 G	Geoblogging could include sports tracking
3	40 G	History information for the user's own use would be ok, the user could save traveled routes
3	36 G	Geoblogging as a feature is ok
3	**0 AA**	**Proximity**
3	23 AA	Proximity alert configuration for all users individually
3	24 AA	Grouping of friends for proximity alerts

3	6 AA	Defining the proximity threshold
3	**0 E**	**User Interface**
3	1 E	The user thought that color of the user on the map meant the user's gender
3	7 E	Settings to the main menu
3	8 E	A shortcut for turning positioning on/off to the map view or the main menu
3	56 E	The user wishes to customize what is shown on the map and the menus
3	21 E	Add possibility to search for POIs
4	**0 A**	**Dates**
4	31 A	More search options
4	33 A	Has not tried anonymous dating before
4	**0 D**	**Conditions to usage**
4	25 D	Use the “freemium” business model, like spotify
4	14 D	Worried how long the mobile phone's battery will last
4	24 D	No ads during business hours
4	17 D	The more profiled the ads are, the better
4	19 D	No beeping or flashing ads to the map view
4	**0 AB**	**Customization**
4	28 AB	Interesting POIs: Public locations, that the user's can review, and the POIs that receive low grades are hidden off the map
4	1 AB	Filter for the shared calendar to hide markings the user doesn't wish to view
4	22 AB	Filter for map markings also
4	18 AB	Give the user the chance to review ads that were sent to the user, and profile future ads accordingly
4	47 AB	Proximity alert: needs customization for friends, distance according to the user's speed?
4	**0 P**	**Bad features**
4	50 P	The worst part of the concept is the battery life and the advertising
4	39 P	History information is not relevant to the concept
4	32 P	Dating ads: wouldn't pay to send one, but would be interested in receiving
4	27 P	Instant messaging is too stressful, the user wouldn't use it
4	42 P	History information

4	15 P	Map markings and geofence statuses are features that are too easy to use for stalking
4	**0 R**	**Sharing information**
4	5 R	Show the user's friend's friend lists for finding friends
4	6 R	Profile needs: work, human relations, studies
4	48 R	Would use facebook widget for sharing location
4	41 R	Not interested in other user's history information
4	**0 E**	**User Interface**
4	13 E	Default choice for method of transportation to the navigation options
4	12 E	Geofence is a bad term in finnish, “geoaita”
4	2 E	The menus need icons
4	3 E	Get the Apple feeling to the product with enough polishing
4	4 E	Get rid of the user's own name in the friends list
4	**0 AC**	**Ok features**
4	29 AC	Routing is good for all transportation needs that aren't the usual routes for the user
4	36 AC	Geoblogging is ok, the user wouldn't use it often
4	**0 AD**	**Good features**
4	37 AD	News by category
4	26 AD	Friends' locations on the map
4	49 AD	The best thing is the way everything is integrated on the map
4	23 AD	Would use the service for stalking friends and communicating with them with the community features
4	35 AD	Leaving messages on the map is a good idea
4	30 AD	Traffic announcements are great information for the user
4	**0 AA**	**Proximity**
4	45 AA	Proximity alerts are ok
4	46 AA	500 meters would be a good distance for an alert
4	**0 S**	**Shared calendar**
4	11 S	Public events to the calendar
4	10 S	The option to invite plus ones to the calendar
4	16 S	Ask for permission to make map markings with geofences
4	9 S	Make the calendar visible to all
4	8 S	Make an option to deny markings to keep the

		calendar tidy
4	43 S	For geofences, it is important to have the option to deny tracking
4	44 S	Geofences are nice when planning events
4	**0 G**	**History information**
4	38 G	Using history information anonymously for statistics is ok
4	40 G	History information stored anonymously is ok
4	**0 AE**	**Audio & Video**
4	20 AE	Audio and video statusmessages
4	21 AE	Audio and video statusmessages should not be possible to send when intoxicated
4	**0 AF**	**Profile information**
4	7 AF	Take notes on how Facebook creates a user's profile
4	34 AF	Add education to dating profile
5	**0 AG**	**Opportunities for the user**
5	19 AG	It is nice to see the locations of the user's friends
5	15 AG	If the user's friends would use the service, the user would use it also
5	25 AG	Sharing the calendar is a good feature
5	26 AG	Would share parties and events
5	28 AG	Shared markings need to alert even when not logged in the system
5	47 AG	Anonymous history is fine by the user
5	46 AG	The user is interested in seeing other users' history data
5	40 AG	Would use messaging
5	41 AG	Could use messages that are activated by geofences
5	**0 P**	**Bad features**
5	52 P	Wouldn't use facebook sharing
5	43 P	Wouldn't use geoblogging or geotagging photos
5	42 P	Wouldn't want to fill in a profile
5	**0 AH**	**Threats to the user**
5	20 AH	Doesn't want own location on the map
5	21 AH	Locationing is intimate, not all the friends need to be tracked or need to track the user
5	54 AH	Privacy problematic when using the service
5	**0 AI**	**Threats to the service**
5	11 AI	Showing the user's location worries the user

5	9 AI	The service feels like it's spying the user
5	18 AI	Wouldn't pay for using the service
5	45 AI	Doesn't like the service recording the user's location history, invades the user's privacy
5	14 AI	Wouldn't use, the service shares too much about the user
5	**0 U**	**Searching for dates**
5	36 U	The user is interested in AIM
5	33 U	The dating service is fine, as long as the conversational partners are far enough
5	35 U	Wouldn't send dating ads, could receive them
5	37 U	Filter out those who aren't serious about the service
5	34 U	Lists for qualities the users are looking for
5	38 U	Not sure whether the service could find the right matches or not
5	39 U	Profile information hardly useful in finding conversational partners
5	**0 AD**	**Good features**
5	53 AD	The service as a whole is the best thing about the concept
5	12 AD	the concept is versatile
5	23 AD	IM is ok
5	13 AD	The shared calendar is a nice feature
5	32 AD	If the user learned to use traffic information, it would be extremely handy
5	31 AD	Traffic announcements are handy for motorists
5	27 AD	The shared calendar has to be better to use than the calendar in the mobile phone
5	**0 AJ**	**Public transportation**
5	10 AJ	Would use navigation with public transportation information
5	29 AJ	Would like to see in the map the location of commuter trains, so the user could optimize his arrival on the train station
5	30 AJ	Would use for other than commuting
5	**0 AK**	**Ads**
5	16 AK	Ads need to be discrete, and not bother the users
5	17 AK	Ads can't always pop up in the same locations
5	**0 AL**	**Suggestions for improvement**
5	44 AL	News preferably by category
5	55 AL	Use orthographic maps supplied by

		Maanmittauslaitos
5	22 AL	Grouping of friends for tracking
5	**0 AM**	**Tracking others**
5	5 AM	Place a geofence around the user's home → the user knows when friends are coming to visit
5	50 AM	Grouping of friends for proximity alerts
5	49 AM	Permission needs to be asked for tracking with geofences
5	48 AM	Changing statuses with geofences is fine
5	51 AM	The proximity alert alerting distance according to the user's place
5	**0 AN**	**POI control**
5	24 AN	POIs are nice as long as they can be hidden when not needed
5	1 AN	Place a POI with the address of the location
5	**0 AO**	**Map control**
5	6 AO	The user thought that the return button in the states of the system would always lead back to the map view, and not stop in the main menu
5	2 AO	The user needs to be able to turn off geofence-alerts
5	7 AO	Customization options for map layers
5	8 AO	Status messages on the map, how are they hidden?
5	**0 AP**	**Date control**
5	3 AP	A minimum distance to the proximity of the AIM conversational partners
5	4 AP	Give the user control on how near the date can be from the user
6	**0 U**	**Searching for dates**
6	25 U	Not that interested in AIM
6	26 U	AIM might find the right persons to chat with
6	27 U	No free descriptions to AIM, fixed sets of qualities
6	52 U	The candidates for AIM don't need to fill all the search criteria, if perfect matches are not found
6	**0 A**	**Dates**
6	24 A	Wouldn't send dating ads, would like the option to turn them on/off
6	22 A	Not much use for a dating service
6	23 A	Lots of search criteria needed for the service
6	**0 M**	**Communication**
6	11 M	IM better than SMS

6	12 M	email replaces asynchronous messaging perfectly
6	28 M	Might find some use for sending messages that trigger with geofences
6	**0 AQ**	**Freshness**
6	54 AQ	The features and the menus should change according to the user's location
6	34 AQ	Might be interested in other user's history data
6	39 AQ	Facebook widget excellent idea
6	53 AQ	Messages should be tied to location, for example, when at work, only work-related messages would be delivered to the user, while other messages would wait for the user to change position
6	31 AQ	News by category, by location is a new concept, the user can't imagine how it would work
6	**0 AR**	**Best in concept**
6	3 AR	Would use the service for locationing friends and chatting with friends
6	4 AR	Thought the idea was practical
6	40 AR	The best thing is the map and the possibilities it gives
6	10 AR	IM would be useful in a mobile phone
6	**0 AM**	**Tracking others**
6	36 AM	Automatic statuses via geofences are good for controlling the service, if the service is on 24/7.
6	37 AM	Tracking others with geofences is not that interesting
6	**0 D**	**Conditions to usage**
6	5 D	Would require the mobile UI
6	6 D	A large network of friends would encourage to use the service more
6	8 D	Could pay under 5€ per month for usage
6	41 D	The worst part of the concept is the dating service, because the user is taken
6	**0 AF**	**Profile information**
6	29 AF	Necessary information: real name, address, contact information
6	30 AF	Different groups of friends could see different things in the profile, the users would be categorized when they are accepted as friends
6	**0 R**	**Sharing information**
6	15 R	Would share all meetings and hobbies

6	14 R	Sharing the calendar is a good feature, if the sharing can be limited efficiently
6	17 R	The ability to export the calendar to other calendar formats
6	**0 AS**	**Annoying features**
6	7 AS	SMS ads might annoy, the user would prefer browsable ads
6	9 AS	Hard to come up with benefits from seeing the user's friends on a map
6	38 AS	No proximity alert, it would cause the cell phone to ring constantly
6	**0 AT**	**Usage of history information**
6	35 AT	History information is useful when geoblogging
6	49 AT	Insert videos to geoblogs and sync the video with the map marker
6	48 AT	Would geoblog pictures and text
6	46 AT	Sports tracker features to geoblogging
6	47 AT	Would use sports tracking for sports
6	**0 G**	**History information**
6	32 G	History information offends privacy if there is no choice for denying the system of recording the history information
6	33 G	Anonymous history is ok
6	**0 AJ**	**Public transportation**
6	18 AJ	Navigation with public transportation information is a good idea
6	19 AJ	Navigation with public transportation information is needed when the user is going somewhere the user doesn't regularly go
6	**0 AU**	**Announcements**
6	21 AU	Traffic announcements are important for motorists
6	50 AU	Add traffic exceptions in public transportation to the service
6	**0 Av**	**POI**
6	2 Av	Would use for finding services
6	13 Av	POI markings are a nice feature
6	**0 AW**	**Finding**
6	42 AW	Keywords to Ads
6	44 AW	Possibility to search through ads that are in the same place physically, for example in the same mall

6	**0 E**	**User Interface**
6	43 E	In the settings, change the order of the battery life settings from “Balanced – Savings – Accurate positioning” to “Savings – Balanced – Accurate positioning”
6	45 E	Give a totally different form for the close button
6	**0 AX**	**Navigation**
6	1 AX	Would use the service for navigation
6	16 AX	Include routing to calendar markings
6	20 AX	Navigation important outside of the user's home town
6	51 AX	Add road speeds to the calculation of the fastest route
7	**0 E**	**User Interface**
7	15 E	Add the possibility to center to a friend on the map to the mobile UI
7	17 E	The default view of the system should be the map view
7	1 E	Using geofences to change friend's statuses is fine, but there should be the possibility to show the friend's original status message
7	26 E	If there are lots of icons in the menus, the service must be customizable
7	20 E	The user's nickname to the screen name on the map, give the freedom to choose one and the freedom for friends to name their friends
7	**0 A**	**Dates**
7	23 A	It's a good idea to be able to grade the conversational partners
7	42 A	Needs some criteria for searching mates
7	43 A	Wouldn't send dating ads, could receive them, if they weren't perverted
7	45 A	With sufficient search criteria, the user would trust the service to find the right partners for the user
7	46 A	Information to the dating profile: living situation, work, future prospects
7	**0 D**	**Conditions to usage**
7	13 D	At least one friend needed in the service
7	12 D	Would want a new phone to efficiently use the service
7	63 D	The worst part: news seem like they don't fit in

7	51 D	Geoblogging sounds weird
7	25 D	GPS only needed when near a meeting point, automatic gps?
7	61 D	Not likely that the user would use the facebook widget
7	60 D	Proximity alert needs customization for all friends
7	**0 AD**	**Good features**
7	37 AD	Sharing parties is nice
7	40 AD	Traffic announcements are brilliant information
7	44 AD	AIM is an interesting feature
7	62 AD	Best features in the concept are navigation with public transportation, shopping the ads, points of interests and sharing POIs
7	16 AD	Would use the service
7	41 AD	Dating seems like a fun idea
7	10 AD	A fun and practical idea
7	8 AD	A handy tool for partying
7	32 AD	Could pay 5-10€ per month for the service
7	6 AD	Tips on activities or public events from the system would be nice
7	38 AD	If the service could be used to monitor kids or the elderly, it would be ok for the service to cost an amount of money per month
7	**0 G**	**History information**
7	53 G	History information is ok, if it can't be misused, for example, it can't be stolen
7	54 G	History information doesn't offend privacy, if it is visible only to the user
7	55 G	The user is curious, and would be interested in seeing friends' history information
7	56 G	Anonymous history is fine by the user
7	**0 R**	**Sharing information**
7	33 R	Friends locations on the map: some might get scared or anxious, if they are shown on map, would control visibility a lot
7	29 R	Sharing calendar information to other calendars, for example by exporting
7	19 R	What about hidden phone numbers?
7	**0 M**	**Communication**
7	21 M	The possibility to forward messages to the user's email
7	18 M	Message history?

7	22 M	Messaging with geofencing would reduce stress, but require that the sender be given notice, that the message will not be received until the user changes location
7	47 M	Messaging is a nice add-on to the service
7	34 M	Chatting would require integration to other chatting programs, but it is ok if the user's friends use it
7	48 M	The user can't tell whether messages tied to a location would be useful or not
7	**0 AK**	**Ads**
7	14 AK	SMS ads are better than browsing ads in the system
7	3 AK	SMS ads are fine, as long as they are well profiled
7	9 AK	Hunting bargains is a lot of fun
7	**0 AF**	**Profile information**
7	24 AF	Audio and video statuses are a good idea
7	50 AF	Basic information of the user to the profile
7	**0 AQ**	**Freshness**
7	2 AQ	When traveling, roadside service ads would be nice, for example, when the user is traveling at a speed near a main road, the service would recognize that the user is traveling by car
7	64 AQ	Weather forecast and especially rain radar on the map
7	**0 AY**	**Markings**
7	27 AY	The possibility to leave messages in the calendar even after their expiration date?
7	28 AY	Alerts to calendar markings
7	30 AY	Different types of markings
7	**0 AZ**	**News**
7	5 AZ	By location is fine when traveling and entering new places
7	16 AZ	What if the news can't be placed on map
7	7 AZ	Entertainment news
7	52 AZ	News by category and in some situations by location
7	**0 Av**	**POI**
7	36 Av	It is important to have a visibility option
7	35 Av	Would mark basic locations on the map: workplace, gym etc.
7	49 Av	Recommending services on the map and messages
7	**0 AM**	**Tracking others**
7	59 AM	Geofencing: The user isn't worried about stalkers

7	58 AM	on/off to geofencing alerts
7	57 AM	Geofencing: Would definitely add one around the workplace
7	**0 AX**	**Navigation**
7	31 AX	Use current location in routing, pull the users location when this action is invoked, so that the location information isn't too old!
7	39 AX	The service needs all public transportation methods
7	**0 AW**	**Finding**
7	4 AW	Searching is an important feature in the system: shops, malls, product groups for ads and services for pois etc

Appendix E.Answers from the questionnaire

This appendix displays the results obtained from the questionnaires as well as some background information about the test subjects.

Feature	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	average	median	variance	min	max
User's sex	m	f	m	m	m	m	f					
The user has a car	n	n	y	y	y	y	y					
The user's age	26	26	29	26	27	26	27					
Relationship status	single	taken	single	single	taken	taken	taken					
Map view	3	5	5	5	4	5	5	4.57	5	0.62	3	5
AIM	3	3	5	3	3	2	3	3.14	3	0.81	2	5
Profile information	2	3	5	3	3	4	5	3.57	3	1.29	2	5
Shared calendar	3	4	4	4	4	4	4	3.86	4	0.14	3	4
Traffic announcements	3	2	5	4	5	5	5	4.14	5	1.48	2	5
Navigation with public transportation	4	5	5	5	5	5	5	4.86	5	0.14	4	5
Discounts	2	3	4	4	4	4	5	3.71	4	0.9	2	5
Map POIs	3	4	5	3	4	4	4	3.86	4	0.48	3	5
Geofences	3	4	5	3	5	4	4	4	4	0.67	3	5
Sharing POIs	3	3	5	3	5	3	4	3.71	3	0.9	3	5
Tracking friends	3	5	5	5	4	3	5	4.29	5	0.9	3	5
IM with friends	3	5	5	1	4	5	5	4	5	2.33	1	5
Messaging with friends	3	4	4	4	4	4	4	3.86	4	0.14	3	4
Local news	3	1	4	2	3	4	3	2.86	3	1.14	1	4
All news	4	5	4	3	3	5	5	4.14	4	0.81	3	5
Understood the concept	4	2	5	5	4	5	5	4.29	5	1.24	2	5
Product seemed easy to use	5	1	5	4	4	4	5	4	4	2	1	5
The prototype was polished	4	1	4	4	3	3	5	3.43	4	1.62	1	5
The UI was reasonable	5	1	4	4	4	4	5	3.86	4	1.81	1	5