Discovery
A helping hand for new experiences
IN 3405 BACHELORPROJECT

Berend Klein Haneveld Ulaş Ülgen
1217089 1249460

November 23, 2009

Project coordinator: P.R. van Nieuwenhuizen
Project supervisor: W.P. Brinkman
Company correspondent: Lauwerens Metz
Technical mentoring Desmond van der Meer

Abstract

With the development of 3G technology information is getting more and more accessible. While being very practical this availability of information can have adverse effects on people. The research shows that some people on their leisure time enjoy and value uncertainty. A literature and user research revealed that people with certain personality types were generally more welcoming to uncertainty. Therefore a location based application was implemented for the iPhone that allows users to create Points of Interest (Poi) on a map that they can share with other users. The users are allowed to see other Pois on the map of their iPhone but information about unvisited Pois is very limited. The central idea of Discovery is making places more interesting not by revealing more information, but by hiding information from the users. Discovery focuses on the experience that a place provides rather than the information of that place. Discovery is designed by confirming to the Apple guidelines for interface design for an intuitive use for the iPhone users. The application was also designed by recognizing that different people are motivated in different ways and thus aims to stimulate different personalities offering a wide range of emotions from a sense of achievement to exploration.
Foreword

Founded in 2009, Innovattic is a company located in Leiden, The Netherlands that is run by specialists in innovative mobile technology. We focus on the development of mobile applications, but our expertise reaches beyond this area to also include web based technologies. Innovattic started on an attic and its main goal is to keep innovating so that our human ambitions of creativity can be fulfilled.

For this project, Innovattic has combined several frameworks to develop a platform on which mobile client-server applications can be built. Our platform was introduced to Berend Klein Haneveld and Ulaş Ülgen. The setup of the assignment provided a lot of freedom to both students: a lot of creativity and fun, but also quite a risky business to prevent the projects scope from being too large.

We thank Berend and Ulaş for their professional attitude during this project. During our conversations our communication was fast and at a high level. They participated in a professional project as full project members and as a result, the road to the final product has been paved.

Desmond van der Meer & Lauwerens Metz
## Contents

1 Introduction .................................................................................................................. 1

2 Assignment Description ................................................................................................. 3
   2.1 Initial assignment and its constraints ................................................................. 3
   2.2 Creative process behind the assignment ............................................................ 3
   2.3 Final assignment and vision .................................................................................. 3

3 Research .......................................................................................................................... 5
   3.1 Literature research .............................................................................................. 5
      3.1.1 Research ...................................................................................................... 6
      3.1.2 Conclusion of literature research .................................................................. 11
   3.2 User research ....................................................................................................... 12
      3.2.1 Program and methods .................................................................................. 13
      3.2.2 User research results and conclusion ............................................................ 14

4 Design ............................................................................................................................ 17
   4.1 Functionality ......................................................................................................... 17
      4.1.1 General functionality of Discovery ................................................................ 17
      4.1.2 Moscow principles of Discovery .................................................................. 18
   4.2 Design choices ....................................................................................................... 20
      4.2.1 Information hiding and revealing .................................................................. 20
      4.2.2 Positive emotions ......................................................................................... 20
      4.2.3 Game elements ............................................................................................ 21
      4.2.4 Apple guidelines .......................................................................................... 21
      4.2.5 Quality vs Quantity ..................................................................................... 21
      4.2.6 Rating system .............................................................................................. 21
      4.2.7 Negative rating ............................................................................................ 22
      4.2.8 Negative comments ..................................................................................... 22
      4.2.9 Titles ............................................................................................................ 22
      4.2.10 Classification .............................................................................................. 22

5 Implementation ............................................................................................................... 23
   5.1 Used frameworks and design patterns ................................................................. 23
      5.1.1 Cocoa framework .......................................................................................... 23
      5.1.2 Google Maps API ......................................................................................... 24
      5.1.3 Model-View-Controller ............................................................................... 24
      5.1.4 Client-server model ...................................................................................... 24
   5.2 Organization of Discoveries components .............................................................. 24
      5.2.1 Database entity relationship diagram ............................................................. 25
Chapter 1

Introduction

It is an age old problem to have the time but not the place to go. It is a common feeling that can be felt by both tourists and locals. People who have a general idea about what they want to do or how they want to feel, but do not know where to go to experience that. In modern cities people can have the feeling that there is some place close that could be quite interesting. If only they had a chance to peek into the collective experience pool of everyone who has ever been in that city. A more recent problem can be having too much to choose from and having to explain our needs to an unachievable level of precision. As people have finer and finer ways to describe areas they end up with descriptions that are ambiguous to others. It is very hard to describe one place at a fine level because the words that someone uses to describe a place might have other meanings to someone else. It is not only because of peoples shortcomings of explaining and understanding experiences, it is also because of personal differences. Even at times when people reach a destination that is exactly what they wanted, people can be unsatisfied. Since the place they wanted to go to is so well outlined, actually going there could be a dull experience like seeing the same movie twice. It might be nice, but not at all surprising.

Of course when people are looking for a hospital, they need to find to the best one (by their own definition of ‘best’) and there are many tools on the Internet that can help with that (take Google Maps for example, that has a function to find dry cleaners in your local area). However when one is not seeking functionality but seeking to spend some leisure time, it seems like the choice is between disappointment and reaching the expected. There could be an opening in the market for a product that would hide just enough information to trigger the curiosity of users. Like a mysterious friend that grabs the user by the arm and suggests them a place that might be of interest to them.

Each day smart phones are getting more common and 3G technology enables data transfer rates to run bigger applications. Therefore an application was designed to run on a mobile device using GPS technology to show users local points of interest (Poi). It was named Discovery since its intention was to motivate people to discover their surroundings. Discovery was designed to be an application which hid the information instead of revealing it. The points reveal their identity slowly teasing the users to walk towards them. Poi’s will be in the center of the application. They will represent areas created by users that might be interesting to other users. Once a point is reached users can, and should be encouraged to, provide feedback to make good points stand out from the poor ones. Pois can also have hidden content which is revealed when the point is reached. Discovery, however, was not designed to be a city guide. It was not made to take the user to a city tour. It was about the experience at a point of interest and was less about informing the user about that place. Once again Discovery respects differences and leaves room for personal interpretation.

Discovery was commissioned by Innovattic to be implemented as a bachelor project for
two TU Delft students (Berend Klein Haneveld and Ulas Ulgen) as a bachelor project in the first quarter of the academic year 2009/2010. On Innovattics side, Lauwerens Metz was the correspondence point and oversaw the planning of the project, while Desmond van der Meer supplied technical support and implemented the server side of Discovery.

The project was implemented by combining two software development methods that were in TU Delfts Computer Sciences curriculum for bachelor students. First one of those two methods was the waterfall model (from the course IN 1405: Introduction to software development) which divided the project into different stages in successively followed each other with a deliverable in stage interfaces. From the start it was decided to implement a soft version of the waterfall model, allowing consecutive stages to overlap. For example the design phase overlapped slightly with implementation and research. The second method of development was inspired from the implementation of the Game Project (IN 2905). Agile programming principles like rapid prototyping were used. From the beginning of the implementation phase, prototypes were made to communicate Discovery’s vision. The first part of the implementation interval was reserved for producing a very basic prototype. In the second part of the implementation, new prototypes arose by adding features to the preceding prototype. Also other user based development concepts like pair programming were prominent in the implementation phase of the project.

The project team have engaged both Innovattic and TU Delft as an influential parties to the project. With TU Delft coordinator W.P. Brinkman weekly meetings were arranged to ask for advice and report about the progress made. Innovattic set up a wiki page that the project team made daily use of. Project planning, finished documents, new ideas, TODO’s were all kept here and were accessible by both the students and Innovattic at all times. Also special care was taken to write daily executive summaries to summarize what was done that day. This wiki page allowed both the coders and company to keep a helicopter view of the project and to be able to always provide a total picture of where the project was, where it should have been, what was done and what needed to be done.

A report at the end of the project was one of the deliverables set by TU Delft. Also to allow Innovattic to further develop Discovery, clear documentation was essential. Instead of postponing all of the reporting work to the last phase of the project and then try to trace back every single step, the decision was made to document every step on the way once the step was taken. These documents were presented both to Innovattic and TU Delft supervisor for feedback.

The report will start with an assignment description explaining how the initial assignment evolved into a final assignment. Then the structure of this report will follow the division of the waterfall model. The research chapter will be followed by the design, implementation and conclusion chapters.

\[^{1}\)See appendix D.1 for the project planning with its different stages.

2
Chapter 2

Assignment Description

This chapter will elaborate on the assignment description and how it came to be. It ends with the final assignment description.

2.1 Initial assignment and its constraints

In the initial phase of the project a meeting took place between the project team and Innovattic to talk about the possibilities of working together. Innovattic wanted to develop a game based on users GPS coordinates that should run on an iPhone. They were prepared to provide an iPhone for testing, a MacBook Pro to run XCode for development and financial compensation. The project team was to implement the application as their bachelor project. Therefore the project needed to be finished by the start of the second quarter of the academic year 2009-2010.

2.2 Creative process behind the assignment

The project team was given one week to develop the concept and came up with a concrete application idea. Innovattic’s initial vague project definition was intended to allow the development of creative ideas. The project team decided to focus on quantity of ideas and welcome many unusual ones. It was planned to combine and improve good ideas and drawing inspiration from the bad ideas. Therefore a directed brainstorming session was held [Osborn, 1963]. In order to reduce the effect of criticism and bias for expressed ideas. Both members of the project team took 15 minutes to come up with their own ideas. Afterwards both members expressed their ideas. All of the ideas are discussed and new ones are derived. Once a promising idea was found (by any of the project team members) it was further developed for five minutes and than the process was repeated with other ideas. In the end a few promising ideas were left that were laid out to Innovattic for further discussion and selection.

After a discussion with Innovattic, Discovery was chosen for further development. The project team went on to identify what functionality Discovery needed to provide and how to motivate the potential users to use Discovery.

2.3 Final assignment and vision

Discovery would be an application running on a mobile device (like an iPhone) using GPS technology to show users local points of interest. Instead of information revealing it focuses on hiding information from the users. The application would enable users to view Poi in their surroundings. Users would not be informed of the contents of the points, merely of their
corresponding rating and classification. The classification of the points can be functional (like cultural, entertaining etc.) or emotional (active, established, relaxed etc.). Discovery would recognize that different people define things differently. That is why the classification paradigm was to be very abstract. Once a user had reached the point of interest, he would be granted the contents of that point and the opportunity to give a rating and leave a comment. Discovery would be application purely for leisure purposes and would not be designed for efficient or functional use. Discovery would not become a city guide at should not take the user on a tour through the city. It would be about the experience at a point of interest and less about informing the user about that place. Discovery would respect differences in users expressions and would leave room for personal interpretation. The points and their contents would be generated completely by the users themselves. No central content management would be exercised. It was assumed that the poor points would be filtered out by their low rating.

The project was initiated with a research phase. The first part of the research involved understanding how to motivate people to explore and rate local points of interest. The second part consisted of exploring how users would like to experience such an application in terms of displayed information, color and interaction. Hereafter a location based application was designed, based on the preceding research.
Chapter 3

Research

After the assignment description was accepted by TU Delft and Innovattic a research phase was started to conduct a more specified product description and fulfill the goals of Discovery. The research phase had the first path on existing literature. Literature research was mainly focused on how to motivate people who would be likely to use a product like Discovery. The second phase of the research was on users. It involved the input of a group of iPhone users and was oriented mainly on three goals: 1) testing if the results of the literature search were applicable to Discovery, 2) testing design decisions made by the project team and 3) answering some of the design questions that the programming team came up with during the Design phase. Although the literature research was done alone, much of the user research was conducted parallel to Design phase in order to spend research time on what users would want to see in Discovery and how those findings could be translated directly into Discovery.

3.1 Literature research

After the initial assignment description, the project team started a literature research phase. The main goal of the project was directly translated as the main question of the research. It was: “How can people be motivated to explore local points of interest?”. To answer the main question, first the assumptions and constraints of the research were defined and then the research took two paths, one on tourism and the other one in motivating people in general. The chapter is ended with the conclusion and its consequences for the design phase.

Assumptions and constraints

As it was declared in the assignment description Discovery would be an application that was intended to be used strictly for leisure purposes. This meant that the users should be free of time consuming activities. It was also assumed that Discovery was not for everyone. Conjecture was that some people will not enjoy the level of uncertainty that Discovery was based on. That is why during the research on personality types the types that were unlikely to enjoy Discovery were not taken into consideration.

Tourism was researched because the description of the potential users of Discovery have a lot in common with tourists. Tourists have time to spend and want to visit interesting places.
3.1.1 Research

Tourism

As described above, Discovery is not suitable to be used in a daily rush. It is essential that the users are free with their time and location. This open state carries many similarities to tourists as far as their time and location are concerned. Since there exists a lot of literature on motivating tourists, literature was searched to find out what tourists like to do with their free time in order to build those ideas into Discovery to motivate users to explore points of interest.

Tourists are people that visit unfamiliar places in order to experience and learn something new [Park, Nam, and Shi, 2006]. They want to see all kinds of attractions like buildings, shops, museums, etc. Actually anything to do with leisure is appreciated. To have the best experience possible, tourists typically research the place they go to by studying tourist guides, information folders and maps. This orientation will give them an idea about what can be experienced and helps them get familiar with the place. The information that they gather is then often put into a planning that will lead them along the places of their interest. The plan that is made before the real visit is not carved in stone and most tourists expect that they may have to make sudden changes during the execution of their plan. Public transport and its inconsistencies can hinder the tourists in going to the places that they want to visit and if the weather turns bad a lot of tourists happily trade their walk in the park for a visit of the nearest museum. And when the weather is fine a walk in the park can take a lot longer than expected. Apparently the experience counts more than the idea of having kept to the scheme and having seen everything on their list. If they happen to come along something that is not in the planning but can give them a nice experience they are willing to change their plans.

A lot of tourists find it important to document their trips [Brown and Chalmers, 2003] and they do this by taking pictures and video's of the things they see and experience. These documents can be shown to friends and family to share those experiences, but also provide a way for themselves to relive those memories [Lueg, Goth, and Bidwell, 2006]. This sharing of experiences over the Internet is becoming very important. Social networking sites as Facebook, Hyves and MySpace show that a number of people are already using the web to document their trips. Technology is really helping the tourist to make the most of their leisure time.

Another aspect of tourism is the social interaction between different groups of tourists and the interaction of tourists with local people. From these interactions tourists can get a lot of insightful information that is not documented in typical tourist guides. From fellow tourists they can learn what places are worth seeing and what experience they are likely to have at those places. Local people can provide directions and their view on places of interest. They also give insight in the local culture and habits. Though some of that information is likely to be documented in tourist guides these live encounters are more valuable than written words.

Motivating people

Although there are many similarities between the expected user profile of Discovery and tourists, the tourist label does not fully cover all motivational issues of the potential users. Discovery is an application that recognizes personal differences and aims to motivate different personalities differently. Yet, as explained in the assumptions section 3.1, it may not be suitable for some personality types [Fogg, 2002] even if they have all the time of the world in their hands. For example: some people are very much information or precision focused and other people do not deal well with uncertainty.

In this section an attempt is made to understand how to motivate users on certain tasks. Deductions are made to understand who Discovery has to motivate, and how to actually do it.
For an application to be used in a large scale it must help users to accomplish their goals. This is their main motivation to use the application. However attempting to solve this problem by defining users goals and how the application will help users achieve them is an underestimation of human diversity. Users have different values and personality traits. These differences affect users motivation to use an application, what their goals are and how they strive for them. For values Schwartz Value Theory [Schwartz, 1994] is assumed. For personality Big five (or Five Factor Model) of [McCrae and Costa Jr, 1997] will be used.

**Values, Personality and goals**  In a general sense, values are conceptions of the desirable [Kluckhohn et al., 1951]. Although more specific definitions are definitely possible (and valid) in the domain of values, proliferation of descriptions tends to hinder the research [Hitlin and Piliavin, 2004]. The term values, were used strictly to describe values as principles that can be defined as: *Learned beliefs that serve as guiding principles about how individuals ought to behave.* [Parks and Guay, 2009]

![Schwartz circumplex showing value bound motivational similarities](image)

For the taxonomy of values Schwartz Value Theory [Schwartz, 1994] is used. This theory divides the personal values in ten categories and places them on a circumplex with four poles (see figure 3.1). The values that stand close together are related, and the values that are in the opposite sides of the circumplex are in conflict with each other.

Personality is the enduring dispositions that cause characteristic patterns of interaction with one’s environment [Goldberg, 1993]. Until early 1980s most of the research concluded that personalities did exist but did not matter [Barrick, Mount, and Judge, 2001], however these studies were focused on work-environment and motivation was only significant as its effects in efficiency. In later years this perception is changed by the rebirth of personality research and Big Five model.

The Big Five model (also known as Five Factor Model, FFM) is currently the most accepted model in scientific community to explain and classify different personality traits [Mount and Barrick, 1995]. Although there are many variations of the simple FFM, this paper refers to the NEO Personality Inventory Revised (NEO-PI-P) model of Costa [Costa Jr and McCrae, 2008], which appears to produce the most consistent cross cultural results [Parks and Guay, 2009].
Figure 3.2: Personality types in the Big Five Model

Figure 3.2 shows the personality types as defined in NEO-PI-R Big Five Model.

**Goals and Motivation**  Personal goals and how people relate to them can be viewed in two main categories. One is what a goals content is, and the other one is how the person chooses to pursue that goal. Pursuing involves both the method of and the persistence of pursuing. Motivation is an energizing force that induces action [Pinder, 1997]. It relates to peoples decisions on when, how and why to exercise effort on certain goals (see figure 3.3). Research shows that peoples values motivates the decisions on what their goals are and that personalities motivates how much persistence people are going to display once a challenge arises. Although there are other goal types that can be identified the most common non conflicting goal classification that stood out of the research were the following:

- **Learning goals:** Goal is to learn [Elliot and Harackiewicz, 1996]
- **Performance goals:**
  - Goal is to achieve high performance
  - Prevent showing low performance [Smith, Duda, Allen, and Hall, 2002]
- **Work avoidance:** Goal is to avoid work altogether
Discovery is an application that should motivate people to discover their surroundings. For this reason, points of interests (Poi’s) are created, visited and rated. Because of inherited properties like uncertainty and leisure orientation, Discovery assumes a personality type and values from users. For this application to reach success it must motivate users in ways that would be aligned to their personality type, values and goals.

Discovery is an application that depends on people’s natural curiosity. Users should have a positive disposition and be flexible in their expectations. That is why the application is not suitable for people who are too neurotic or not open to the unexpected. People who are extroverted are expected to be outgoing and social. Local Pois that they do not know would be of interest to them. People of open nature should like the applications surprising character and people who are very goal oriented should enjoy it when they have a sense of accomplishment from the application. From this point on the focus will be on extrovert, conscientious and open personality types.

Values are harder to predict in bigger scale because they are mainly influenced by the environment. Yet the values that are relevant to Discovery can be directly derived from personality types, using a meta-analysis of Parks and Guay [2009] that shows correlations between certain personality types and values. The system should appeal to people with Learning Goals since the application is directly based on hiding and revealing information. This should translate directly into learning their surroundings. Among the people who are oriented in Performance-Approach goals, the application can offer some kind of rewarding system to recognize their achievements. People in the Performance-Avoidance group are not discussed here since the application does not allow users to “fail” in any sense.

In contrast to the work environment the focus here will be set on Work-Avoidance group. Since the activity of using the application is a leisure activity the users would like to commit as
little as possible to the application while (and if) they are using it. Table 3.1 shows the results of research on how different types of goals can be motivated for different personality types. This is done by adding all the correlation numbers from different goals. When the total result was higher than .30 like aesthetics, feelings, warmth, positive emotions this was taken as an indicator that overall design of the Discovery reflected those values. Activity had a huge negative impact on the people with work avoidance goals and since Discovery’s focus was the work avoidance group this value is also marked as a point that should be looked into. Deliberation had a negative impact in all goal groups, therefore Discovery was designed in a way to eliminate deliberation.

As it can be seen from the previous table, the application needs to have certain properties in order to motivate user to begin and keep on using it. In order to appeal to all relevant goal orientation types Discovery should be beautiful (Aesthetics), should involve emotion (Not too functional). It should be warm and contain rather positive emotions. Discovery should favour achievement rather than competition. So reaching rewards is fine as far it is not compared to other users (this would turn off work avoidance type). Discovery should further be designed to contain minimum amount of activity and deliberation (see table 3.1).

<table>
<thead>
<tr>
<th>Open</th>
<th>Learning Goals</th>
<th>Performance Goals</th>
<th>Work Avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fantasy</td>
<td>.16</td>
<td>-.02</td>
<td>-.01</td>
</tr>
<tr>
<td>Aesthetics</td>
<td>.25</td>
<td>.14</td>
<td>.04</td>
</tr>
<tr>
<td>Feelings</td>
<td>.31</td>
<td>.21</td>
<td>.04</td>
</tr>
<tr>
<td>Actions</td>
<td>.31</td>
<td>.02</td>
<td>-.08</td>
</tr>
<tr>
<td>Ideas</td>
<td>.29</td>
<td>0</td>
<td>-.14</td>
</tr>
<tr>
<td>Values</td>
<td>.17</td>
<td>-.49</td>
<td>-.07</td>
</tr>
<tr>
<td>Extroverts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warmth</td>
<td>.23</td>
<td>.10</td>
<td>.07</td>
</tr>
<tr>
<td>Gregariousness</td>
<td>.10</td>
<td>.11</td>
<td>.05</td>
</tr>
<tr>
<td>Assertiveness</td>
<td>.05</td>
<td>.01</td>
<td>-.05</td>
</tr>
<tr>
<td>Activity</td>
<td>.23</td>
<td>.02</td>
<td>-.28</td>
</tr>
<tr>
<td>Excitement</td>
<td>.08</td>
<td>.12</td>
<td>-.13</td>
</tr>
<tr>
<td>Positive emotions</td>
<td>.25</td>
<td>.05</td>
<td>-.01</td>
</tr>
<tr>
<td>Conscientious</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Competence</td>
<td>.20</td>
<td>.05</td>
<td>-.20</td>
</tr>
<tr>
<td>Order</td>
<td>-.03</td>
<td>.16</td>
<td>-.13</td>
</tr>
<tr>
<td>Dutifulness</td>
<td>.10</td>
<td>-.01</td>
<td>-.29</td>
</tr>
<tr>
<td>Achievement</td>
<td>.30</td>
<td>.13</td>
<td>-.31</td>
</tr>
<tr>
<td>Self discipline</td>
<td>.03</td>
<td>-.01</td>
<td>-.32</td>
</tr>
<tr>
<td>Deliberation</td>
<td>.16</td>
<td>-.07</td>
<td>-.02</td>
</tr>
</tbody>
</table>

Table 3.1: Personality types and motivational methods for different goal orientations

1 Only the personalities and goals are shown that are relevant to Discovery.
3.1.2 Conclusion of literature research

**Surprises** A tourist (one that does not follow a guided tour) can have moments where there is nothing special planned while they still have time to spend. And the plans that tourists make with the help of information from tourist guides is often not definitive and leaves room for a lot of experimentation and adventure. That creates a lot of moments where Discovery can help people spend their undefined time in the best way possible. When there is no need any more to see a specific place, then there is only the need for a place that gives you a certain feeling or experience. With Discovery tourists can look up those places and go on adventure.

**Social interaction**

Discovery could be seen as an extra social tool to be used by tourists. This can be a new way to share your thoughts and feelings about points of interest and benefit from other people who have done this. It also can be integrated with already existing social networks with the help of an applet that can display the users exploration status and pictures made on their journeys. If the user base will be honest and open minded Discovery can become a tool of great value.

**Motivation**

In line with the assumptions that were previously made, a model was constructed to identify how users with different personality types, values and goal orientations can use Discovery to reach their own goals. Discovery has to motivate the users on

- Making Pois
- Going to Pois
- Rating Pois

Table 3.2 shows a table of the actions that can be taken in Discovery and how they are encouraged for different personality types using different values.

<table>
<thead>
<tr>
<th>Action</th>
<th>Open</th>
<th>Extroverts</th>
<th>Conscientious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making Poi</td>
<td><strong>Stimulation</strong>: The content, description and name of a Poi is free, users are allowed to experiment with their imagination and enjoy variety.</td>
<td></td>
<td><strong>Achievement</strong>: User can earn titles for rating and going to number of Pois.</td>
</tr>
<tr>
<td>Rating Poi</td>
<td><strong>Universalism</strong>: Users are given the chance to rate points both negative and positive. Letting them to do the just thing.</td>
<td><strong>Self direction</strong>: Possibility of extending a point with own comments lets user to express his/her feelings. And tagging lets the user to share this with other users.</td>
<td></td>
</tr>
<tr>
<td>Going to Poi</td>
<td><strong>Stimulation</strong>: By making visited points visually different than unvisited points, the users are stimulated to visit different places and the openness if the system guaranties variety.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Actions, values and personalities in Discovery
Table 3.3 shows how the above defined actions would actually be pursued. In short how people from different personalities can be motivated to complete some selected tasks (to satisfaction, with integrity) once they start them.

<table>
<thead>
<tr>
<th></th>
<th>Open</th>
<th>Extroverts</th>
<th>Conscientious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making Poi</td>
<td>Work Avoidance: It takes under 2 seconds to complete these activities. The user is also not presented with any obstacle on the way.</td>
<td>User can earn titles for rating and going to number of Pois as well as making good points.</td>
<td>The ratings are only earned if task is completed.</td>
</tr>
<tr>
<td>Rating Poi</td>
<td>So it is safe to assume, once the user decides to take action (to make or to rate a poi) the action will be completed before the user can decide not to do it after all.</td>
<td>This feeling can be elevated when the application is connected to social networks (facebook, hyves etc...)</td>
<td></td>
</tr>
<tr>
<td>Going to Poi</td>
<td>A theme (probably) bubbles: Theme will be implemented in the interface to make it aesthetically pleasing to open personality types. The theme would also appeal to the extroverts by stimulating positive and warm emotions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Solutions to actions, values and personalities in Discovery

### 3.2 User research

During the design phase the project members came across some choices that potential users should answer (and not the programmers). The programmers also wanted to test out their own assumptions on users. To do this a focus group [Carroll, 2000] was used which were shown to help the design process new technologies [Bruseberg and McDonagh-Philp, 2002]. Assumptions were that an emotional classification will be a big asset to Discovery and the users would use tags to associate adjectives with places. It was unknown which adjectives could be used to associate with places. After discussion with the project supervisor the following methodology was devised, the focus group members were presented with a number of pictures and were asked to associate different places with different adjectives. These adjectives were later analyzed to create a list of words that could be used as tags in Discovery. Although there were existing emotional classification paradigms, none was found that the programmers could use for Discovery to quickly understand and classify random places. The focus group members were asked to describe their favorite places, by not using sentences but just keywords. This setup was intended to simplify the task of emotionally classifying places to emotionally classifying words.

Scenarios are useful in the design process since they capture the consequence and trade-offs of designs. The narrative nature of the scenarios enables users to imagine the users situations and context of the new technology. That is why the focus group was presented with a scenario followed by a group discussion to test the following design concepts that the project team intended to build. A sub goal was to find out any usable idea that users might provide. The questions were:

- Can not knowing exactly what a place contains make that place interesting?
- How much time are users willing to commit to the creation of Pois?
- Is there a need for classification?
Would game elements make Discovery more interesting?

### 3.2.1 Program and methods

#### Focus group

For the focus group a young group of people were invited. Since Discovery would be implemented as an iPhone application, people who are iPhone users were selected. The group consisted of 3 female and 10 male attendants. First the observers introduced themselves to the focus group. After this introduction, the schedule and the objectives of the evening were explained. The introduction ended with a general description of the application. The execution of the whole user research took 90 minutes with a 10 minute break in between. The first part of the research was managed with an introduction in the beginning followed by individual tasks. The second half was an open discussion where the focus group members were free to discuss their ideas about Discovery.

#### Tags and classification

First the observers presented the focus group with 34 pictures (shown in appendix section B) and asked them to describe the pictures by using words from a list of adjectives. The pictures were chosen to represent a variety of Pois that users could visit using Discovery. They were printed in color, and some of them were known tourist places to allow the focus group to go beyond the visual and provide more emotional content [pau]. The list of the words consisted of positive adjectives. The positive wording was chosen to encourage the group to focus on the positive sides of the places. The group was further instructed to write down any other word that they missed in the list. In the second part the users were given a piece of paper and instructed to describe their favorite place. Focus group members were told not to write down whole sentences, but to select keywords that described their favorite place.

The first session had no time bound, it was rounded up after 25 minutes when the majority of the focus group seemed to be done. This approach was used to understand which words were commonly used to describe (or tag) Pois and to find any adjectives that observers might have missed. The second session was shorter (10 minutes) in order to prevent focus group members from second guessing their intuition. This rapid pace is consistent with what the observers expect from the creation of a Poi with a hand-held device. Second session is intended to lead way to an emotional classification for Discovery. The results from both sessions were handed in anonymously and were not further discussed with the group. The list of the presented adjectives can be found in the appendix at table C.3.

#### Scenario

In order to convey the concept of the application better, the users were presented with a story (scenario) of a young couple using Discovery on vacation in Paris (the full scenario can be found in the appendix section A). The observers gave little detail about the design choices they made in order not to bias the focus group to the choices that the observers already found suitable. Another reason for the limit was that to explain some of the aspects of Discovery it would require some fundamental understanding of the application. The observers felt that giving more details would accomplish little, yet cause lots of confusion.

After the scenario was read, the focus group was asked what they thought of the application, what they would expect from such an application, if they would use it, what would they change to make Discovery better. The ideas were thrown into the conversation and the focus group discussed the ideas that are expressed. Extra care was taken to steer the conversation back
on topic when the focus moved away or when more expressive people started dominating the
group. The session took 5 minutes for the scenario and 40 minutes for the discussion. The
discussion was recorded on video for later observation.

3.2.2 User research results and conclusion

It must be remembered that the following results are initial indicators towards a conclusion
since they are based on a single focus group. Yet it is taken as a source of inspiration and a list
of indications to help the programmers.

Tags

The results of session one and two showed that not all the words were equally used. Some words,
it seemed, were more applicable to describe places of interest. The users also provided some
words that were not on the list. Although some of the provided words were too specific to that
particular place (thus not directly usable as tags) there were some words (for example: Buzzing)
that were used by multiple users to define multiple places. By taking this in consideration a
new list was created for tags.

Generally words that were used less than 10 times were removed from the initial list because
they were not frequently used. The words “Comfortable” and “Funny” were still kept in the
list because their low frequency is explained by the lack of pictures in the experiment setting
that could be associated by those adjectives. From the adjectives that were used more than 10
times, “Beautiful” and “Great” was removed because they told too little about the place that
they were associated with and the adjective “Relaxed” was removed because it would be used
as a part of the emotional classification. After shrinking the list, the provided words from the
first and the second session were taken into consideration. These were added to the tag list
after removing the ones that were too specific or negative. Removed words with their reasons
and user provided words can be found in tables C.1 and C.2 consecutively in the appendix. The
final list can be seen in table 3.4.

<table>
<thead>
<tr>
<th>Adventurous</th>
<th>Exiting</th>
<th>Proud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affectionate</td>
<td>Friendly</td>
<td>Quiet</td>
</tr>
<tr>
<td>Alive</td>
<td>Funky</td>
<td>Respectable</td>
</tr>
<tr>
<td>Amusing</td>
<td>Funny</td>
<td>Romantic</td>
</tr>
<tr>
<td>Antique</td>
<td>Happy</td>
<td>Sad</td>
</tr>
<tr>
<td>Artistic</td>
<td>Impressive</td>
<td>Safe</td>
</tr>
<tr>
<td>Big</td>
<td>Interesting</td>
<td>Satisfying</td>
</tr>
<tr>
<td>Brave</td>
<td>Joyful</td>
<td>Sexy</td>
</tr>
<tr>
<td>Buzzing</td>
<td>Lovable</td>
<td>Small</td>
</tr>
<tr>
<td>Calm</td>
<td>Organized</td>
<td>Smells good</td>
</tr>
<tr>
<td>Challenging</td>
<td>Overwhelming</td>
<td>Strong</td>
</tr>
<tr>
<td>Chaotic</td>
<td>Passionate</td>
<td>Surprising</td>
</tr>
<tr>
<td>Surprising</td>
<td>Peaceful</td>
<td>Traditional</td>
</tr>
<tr>
<td>Cool</td>
<td>Playful</td>
<td>Warm</td>
</tr>
<tr>
<td>Curious</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4: Tags to be used in Discovery
Emotional classification

It was not directly asked to the users how the emotional classification ought to be. It is a hard question to explain. That is why an emotional classification was derived by generalizing the words that the focus group was provided with by describing their favorite places. It should be noted here that emotional classification that was aimed to be found here did not exist in literature. So an attempt was made to create an emotional model to classify the places on user experience. When the results were reviewed it was seen that all the places could be placed on a two dimensional space (see figure 3.4). The activeness of the Poi on one axis, and the conventionality of the place on the other. On the activeness scale the places that were ranked high were typically fast paced, energetic places. Places ranked low were calm, quiet and generally relaxed. The conventionality axis had on freshness hand places that were nice because they were new, innovative and/or different. On the established side were places that were nice because they were old, traditional and/or familiar. It was decided by the observers to use this as the emotional classification for Discovery. Although there were many more dimensions of emotionality, this element of two dimensionality was found in all provided answers. The classification is simple and there are already websites\(^2\) using a similar classification for describing the moods of music. Four colors can be chosen for each pole in the model and by giving every pole a unique color, it would be possible to represent the experience (or emotion) of every Poi by interpolating its color between the poles.

![Figure 3.4: Emotional classification of Discovery](image)

\(^2\)For example: http://www.musicover.org
Concept testing

This testing was mainly done indirectly by analyzing the discussion that took place after the scenario. By not revealing design choices in the explanation of the scenario the observers aimed not to bias the focus group. Instead only the concept and user experience was explained. In the discussion that took place after the scenario, the users seem to agreed among themselves upon some design choices that were parallel to concepts that the observers were intending to implement. This is assumed to be a positive indication that the concepts that were to be implemented were likely to be acceptable by users. A list of the approved concepts is as follows;

Q Can not knowing exactly what a place contains, make a place interesting?
A Idea of not knowing where one might end up, is an interesting idea and some people would like to experiment with this idea, if there were available software.

Q How much time are users willing to commit to creation of Pois?
A The users want to commit as little as possible to the creation of Pois.

Q Is there are classification needed?
A Despite Discovery’s main strength of making places interesting by hiding information, some kind of classification is needed to prevent users from ending up in places that would turn them off.

Q Would game elements make Discovery more interesting or irritatingly childish?
A Game elements (like leveling and experience) would give a fun feel to the application. Achievements besides the satisfaction of exploring a new Poi, would be appreciated by the users.

Q Would users rather leave comments or just tags by the Pois that they have visited?
A The users wants to have the option of leaving comments to say whatever they want over Pois, yet they would work with a list of selected words since it is faster and easier on a hand held device.
Chapter 4

Design

By this stage the goals and subgoals for the application were defined by literature/user research and some design choices were already made by Innovattic. The project team concentrated in two domains to reach the identified goals. On one hand the functional specification of the application was developed (in constant contact with Innovattic) and on the other hand design decisions were made in order to motivate users in the ways that were indicated by the research phase.

4.1 Functionality

4.1.1 General functionality of Discovery

In order to motivate people to visit the interesting locations in their surroundings Discovery needs to facilitate the following actions.

- Definition of a Poi: Poi is point of interest. Can be anything or anywhere. The content is left entirely to the users. A Poi is defined by a location and a classification. A title, comments and tags can be added

- Obtain local Pois: Obtain users location using GPS on iPhone. Connect to the Innovattic server and request Pois around the users location.

- Display the Pois on iPhone screen: Pois that are obtained from the server are displayed as semi transparent circles (bubbles) in the screen with an icon to reveal the functional classification of the Poi and a size parameter which is depends on the rating of the Poi. If a Poi is visited by the user, the user can enter a “Detail View” screen where all of a Poi’s content is revealed.

- Rating the Pois: Rating of the Pois effects their size. The Pois with a high rating will be displayed bigger and will draw more attention on the map screen. The users can rate a Poi if and only if they have visited that Poi. The user can either like or dislike a Poi, making it either bigger or smaller. The growth is not linear to prevent Pois from growing indefinitely. Shrinkage of the Pois can result a Poi to disappear from the map entirely.

- Create a Poi: The creation process of a Poi consists of three taps at its minimum. The user taps the add bubble button from the Map View and than choices a functional classification. Tapping on the done button from the Poi Creation View adds the Poi to the system. If desired the creator reserves the right to give a title to the Poi, give it an emotional classification which will be visible to other users. Other than that a created Poi is added
right away to the visited list of the creator hence the creator has all the rights over the
Poi like other users who have visited it.

4.1.2 Moscow principles of Discovery

In order to fulfill these functionalities some pre-research MoSCoW principles are defined as
follows. But first a definition of the MoSCoW terms are given as they are used for this project.

Must haves: Concentrated effort for the basic functionality. Foundation for the prototype.
The simplest way to represent every functionality, the way the Discovery will work and how the
different modules in the system will communicate with each other.

Should haves: Prototype as it will be handed in to Innovattic at the end of the project.
Some extensions to “must haves”. A dry working product enough to represent what the end
product can be. Not made to deal with great amount of data.

Could haves: End product as it can be offered to users. All functionality is finished and
polished in a way that has the necessary fun elements to encourage users to use it. Made to be
scalable and reusable.

Want to have (won’t have): All the ideas that are interesting, appealing, functional yet
can not be implemented because of time, cost and quality constraints.

Must Have

- Ability to send users GPS coordinates to the Innovattic server and receive the coordinates
  of local Pois.
- User interface that displays user location and local Pois, using a map obtained from an
  external server.
- An interface that supports all the above functionality.
- Reusable system components.

Should Have

- Ability to allow users to create their own Pois.
- Ability to allow users to rate the Pois that they have reached.
- Ability to filter out Pois of interest according to the users preferences.
- An intuitive (conforming to Apple User Interface Guidelines) user interface that motivates
  users to explore, add and rate Pois of interest.
Could Have

- Capability to offer users Pois that they might like (using similarities with what they previously liked or what similar users liked).
- Ability to allow user write comments.
- Plug-ins for social networks (Facebook, etc).
- Ability for the user to favorite some points.

Want to Have

- Ability to add other multimedia elements. Possibly using other 3rd party servers (YouTube etc).
- Ability to import other peoples Pois and export your own.
- Ability to keep track of Pois in terms of popularity and added comments (self created or favorite Pois).

After the research phase some points in these MoSCoW principles were changed. From the Must Haves, “Reusable system components” have been removed because of the constraints of Objective C making it hard to port to other mobile devices. The re-usability was still an issue only not as important. Elements from the Should Haves were divided and distributed between Must Haves and Should Haves. To the must haves the basics of the functionalities were placed and the Should haves contained the extended versions of the same functionality. Writing comments and favoring Pois were added to the scope of Must/Should haves because it appeared in the user research that these elements were essential to the user.

Also new elements were added to the post research MoSCoW. Experience was added to the application as a game element (see section 4.2.3). A user account bounded to the user details is made a Must have to enable saving user history, favorites, preferences and experience. Displaying users history is become a priority confirming to the advice of Innovattic. Also an offline mode is built in with a local database to deal with internal data. In light of those decisions the pre-research principles were specified, divided and redistributed. As a result the post research MoSCoW principles were as follows: Here only Must haves and Should haves are displayed since on the prototype only these parts will be implemented.

Must Have

- Data server: connect to a data server, download the Pois, and place them on the map. Create a user account, log in and out.
- Map server: connect to the Google Maps server, download and display the maps.
- Visited Pois: keep track of the visited Pois by user, recognize when the user arrives at a point and display visited Pois differently than other Pois.
- User information: retrieve the users location, place him on the map and keep track of the users movements.
- Establish Poi details: retrieve visited and update Pois details. Retrieve the title and update the rating by voting +1.
- Poi creation: create a new Poi in the map, provide an optional title for it. Upload the Poi and its title to the server.
4.2 Design choices

In this section all of the design choices that had to be made will be discussed.

4.2.1 Information hiding and revealing

With the increasing number of mobile devices (like laptops and smart phones) information has become more accessible than ever. It is very convenient to reach the contents of the Internet when one is searching for a function. However when one aims spending leisure time seemingly limitless accessibility of the information can reduce the spontaneity and surprise. People in their leisure time do not see surprises as inconveniences but instead they enjoy this unexpectedness. That is why the overall design of Discovery is made on hiding and revealing information. The users are presented only with a location and a general classification (relaxed, cultural place) of the places that they might visit. The functional classification is displayed by using different icons and the emotional classification is displayed by using different colors. The rating of the places are displayed with the size of the Pois. So the users have a third attribute (other than proximity and classification) to help them decide to visit a Poi. Once a Poi is reached the user is given the ability to view the rest of Pois content. This includes a title, the associated tags and comments left by other users.

4.2.2 Positive emotions

It is showed that use of a positive language in applications encouraged users made users happy and more inclined to use the software. That is why every articulate part of Discovery is designed to reflect positiveness. When it comes to user input the users are again encouraged (but not forced to) to use positive language. When users want to associate tags to a visited Poi they are presented with a list of positive adjectives (see table 3.4 in the appendix). When the users decides to enter comments to a visited Poi, they are first given the chance to complete the sentence “About this place I like ...”. If users wishes, they can edit, or remove this text. This
is likely to result in users leaving more positive comments and draw other users attention to the positive things about the Poi when they read those comments. Also people who are used to completing the pre-set sentence “About the place I liked ...” will be more likely to look for positive things to write about when they are leaving comments.

4.2.3 Game elements

From the user and literature research it was apparent that users were likely to be stimulated by game elements in a leisure application. This can translate to more frequent use of application and more satisfaction from using the application. So a game element from RPG (role playing game) genre was chosen to be embedded into Discovery. Every user has to make an account to use the application and every account starts its life with zero experience and a title “Novice Explorer”. As the user visits more Pois, the experience increases. Once it reaches a pre-set threshold they get a new (better) title. The users can see their title and progress from their status view. This is likely to give people a sense of achievement. This system is also very flexible for future extensions and can be very easily displayed in online social networks to display ones accomplishments.

4.2.4 Apple guidelines

Apple has very strict guidelines over the interfaces regarding how applications that are running on Apple products can display information. That is why in the orientation phase special time was reserved to read through and experiment with Apple guidelines and conventions. Throughout the implementation almost every interface element was used from the Apple’s standard Cocoa libraries. This choice spared some time on implementing custom interface elements. It also should give Mac users a familiar feeling and easy learning curve when they first turn on the application.

4.2.5 Quality vs Quantity

The application is highly dependent on the number of Pois a person can find. No matter how good the idea or the implementation of the idea is, unless some Pois are to be found in the walking distance, the system is “pointless”.

The creation and content of the Pois are managed mainly by users therefore the users should be motivated to create Pois. Here a design decision was needed between quality and quantity. Making more fields that needed to be filled by the creation would push users to fill in more content and by doing so create more detailed (quality) Pois. Yet is assumed that making the creation of Pois complex or time consuming would have an adverse effect on the number of Points that users will be creating. However a low commitment creation process might result in creation of too many points, sacrificing quality over quantity.

After in group deliberation and advise from the TU coordinator it was decided to make the creation of the Pois very simple and quick. This might result in less quality of the Pois content. However because of the low commitment that is needed, it is expected that the users will create more Pois, which would mean more content for the whole application. The application’s rating system is expected to filter out the poor Pois in time.

4.2.6 Rating system

Although there are many ways to rate a Poi, many problems arise on relativity, simplicity and with the time a just voting system demands from the user. Therefore a choice was made for a
4.2.7 Negative rating

It is a common property of applications to allow users to rate the content that other users provide. This separates the good quality content from the poor and ordinary. Because the content of Discovery will also be provided by users, the users are allowed to weed out the poor points. In many applications the users are only allowed to give positive ratings to content. This positive way of thinking, that your content can only get positive reaction, encourages users to provide content without the fear of disapproval. This approach however makes it difficult to separate poor content from ordinary content. Therefore a design decision was made to allow users to negatively rate a Poi. This bipolar rating model would separate good and poor from the ordinary.

4.2.8 Negative comments

The users are encouraged by the application to provide Pois with positive comment. When user opens the comment window to type a comment, he is greeted with a half complete sentence as: “In this place I like: ...”. All that the user needs to do is to complete this sentence. On the other hand if the experience of the user is negative about that Poi, he/she is allowed to delete the “In this place I liked” part, and type in whatever comment the user is willing to express or simply rate the Poi negative.

4.2.9 Titles

The Pois default title is location based. It is just, where the Poi is located. However the users allowed to name the Pois that they have created themselves. This might give users a more personal feeling than a cold street name. Users are also allowed to name the Pois that they have visited. These names are not displayed to other users because they are intended for personal use. By doing so the users are allowed to choose a name with which they themselves can recall the Poi.

4.2.10 Classification

Classifying the Pois means that they have to be grouped together. This would allow users to find Pois that they would like. However there are many problems with classification. Users can intentionally or unintentionally misinterpret the classification of a Poi or a Poi might belong to a number of classifications. Even worse, the classification could take away (to a degree) what Discovery is all about; not knowing where you would end up. Yet, not classifying the points at all is not a care free answer either. For example many people are likely to be very dissatisfied when they end up in a loud noisy shopping mall when they were looking for a place of relaxation.

So a compromise was made between classifying and not classifying. A classification will be made, but it will be kept in a very abstract level, or it will be very functional. In either case, the classification will generally be true, yet will not tell very much about the Poi. As an example, one can think of a Bob listening shuffle from a large play list of music. Bob doesn’t want to know who the artist will be, or from which album (if he did, he would be listening that song) but he does know what kind of song he likes to hear. The genre of the song if you like.
Chapter 5

Implementation

In this stage of the project a concrete idea had been formed by the team about the different elements that were needed to be built in Discovery. Global design and frameworks that were to be used were identified and specified program requirements were obtained from the post research MoSCoW principles (see appendix E.1). This stage slightly overlapped with the design phase since implementation of certain elements required a redesign of Discoveries components. For example in the beginning the programmers intended to store local Poi data in arrays. But later it appeared that arrays were a poor choice to handle the amount of data which Discovery was expected to cope with. So a database using SQLite was designed and implemented for efficiency and endurance.

In this chapter two design patterns that are used for the implementation of Discovery will be briefly explained as well as their relevance to Discovery. This will be followed with the global outlines of program components (like architectural design and entity relationship diagram for the database). At last the navigation between different views will be explained using a state machine diagram and screenshots from the application.

5.1 Used frameworks and design patterns

In this section frameworks and design patterns that were used in Discovery will be mentioned. The explanation will involve a brief explanation of the frameworks and design patterns that were used, why they were used in Discovery, and how they were implemented.

5.1.1 Cocoa framework

Cocoa [app] is not a single framework but Apple’s name for the collection of frameworks, API’s, and accompanying runtimes that make up the development layer of Mac OS X. By using Cocoa the developers were driven to write software with a similar structure of the operating system. Because of this conformance with the operating system Cocoa was chosen because it is the best way to make the application as native as possible. Cocoa also offers bindings that eliminate most of the glue code associated with Model-View-Controller implementation (see section 5.1.3).

Since Discovery is an iPhone application the use of Cocoa strengthened the Mac look and feel. The use of Cocoa also enabled the project team to make use of Apple’s Interface Builder which greatly reduced the costs of the interface implementation.
5.1.2 Google Maps API

Google Maps [goo] is the most frequently used free map provider in the market right now. Google Maps API is a collection of definitions which allows the integration of Google Maps into any webpage using JavaScript. It also provides utilities for manipulating the maps and adding content to it using layers. The development of Google Maps V3 took the robust working of Google Maps on mobile devices into consideration.

iPhone devices already come with a Google Maps application that can show the user’s location on a map. For developers there is also an API for the use of Google Maps. This API provides multitouch support, can show the user’s location and annotations on the map. Since it was intended that Discovery would confirm as much as possible to the iPhone conventions, the use of Google Maps API was a reasonable choice. Since Google Maps API is free yet permitted limited commercial use it was also an attractive choice for Innovattic.

5.1.3 Model-View-Controller

The Model-View-Controller (MVC) is an architectural design pattern that stimulates the separation of data, logic and user interface. The interface elements show the data from the models and the controllers handles input from the view to make calls on the model. The separation of these three elements make it possible to have tight control over the representation and manipulation of data. Cocoa encourages the use of MVC by storing all interface elements in view containers and making corresponding controllers to handle the input from the view containers.

Discovery uses MVC by using the Interface Builder to build most interface elements of the application. For every interface element there is a controller that handles all input. The controller has links to the data and can perform actions on the data. When the data is changed, it will notify the corresponding views to update itself to represent the new data. This functionality is used by conforming to the Key-Value Coding guidelines [key].

5.1.4 Client-server model

A client-server model is a model where all data from the application can be found on the server and the client can get the data it needs from the server. This is needed for Discovery because it is an interactive system where most information will be shared between clients.

This model extends the use of the MVC model by using the server for the manipulation of data while the client only represents the data that it gets from the server. This is particularly useful because the iPhone is limited in its processing power and memory use, so by using the server for the manipulation of data the clients only task is the representation of the data.

5.2 Organization of Discoveries components

In this section two main components of Discovery will be explained in more detail. First the implementation of the database will be explained together with how different tables relates to each other. Further on the architectural structure of Discovery will be explained. From the architectural design it will also be apparent how Discovery communicates with its local database, the Innovattic server and with Google Maps server.
5.2.1 Database entity relationship diagram

Figure 5.1 shows how the database on the iPhone is designed. There are entities for Pois, Users, Comments and Visits. The Pois, Users and Comments are sent by the server to the iPhone, so it is understandable that there is a table for each of these entities. But the Pois that are sent from the server contain information like the visit date of the current user. In the database those 'personal' values are stored in a separate table so that a lot of information of the Pois can be shared by multiple users on the iPhone. It is not very likely that a lot of accounts are to be used on one iPhone, but if there are then they can make use of each other’s data. This can be used to save as much bandwidth as possible.

For a Poi the title can take 4 shapes: the coordinate, the street name + city name, the name given by the creator or a custom name. The coordinate and the name given by the creator (if it exists) are send by the server, but the street name and custom name are not provided. The street name is queried with the Google API and stored for network and processor performance and the custom name is saved as part of a visit because a user can only change a title when he/she has visited a Poi.

By keeping as much information as possible separated from each other it has become a very flexible database model and can be easily extended for future purposes.

5.2.2 Architectural design

Because one of the ideas for Discovery was that it should be usable offline, it became important to let the application make use of some kind of cache system where Pois could be stored offline and still be manipulated. The reason Discovery had to be able to work offline is that an Internet connection in foreign countries is quite expensive and is not part of the contract with the telephone company.

This cache takes the shape of a Sqlite database on the phone where every Poi the server sends will be saved and thus can be viewed offline. This should make it possible as well that the
user can select a region like London and save all the Pois that are there. Then he/she can visit London and make visits and comments/notes that are all saved in the local database. When the user comes back home he/she can connect again and upload all the activities and everything will be synchronized again.

To display the Pois on the map they have to be put in the annotations of the map view element. This array serves as a kind of cache as well. Every part of the application that needs information about a Poi gets an annotation from this array and if changes are made to this Poi it will notify every element that is using it that they need to update their views. This makes sure that everywhere in the application the information will be consistent and up to date. This notification system uses the model view controller handles that are build in the Cocoa Framework called Key-Value Observing.

To make the communication with the server transparent to the user, Discovery makes use of server operations that are handled in a different thread on the iPhone. This means that if Discovery asks for an update of Pois it makes a server operation that is put in a queue in another thread. This keeps the interface responsive to the users touch because the server operation and the response of the server do not execute in the interface thread. This brings some difficulties though because some of those interface elements can request something from the server, but when the server responds the interface element that send the request could be released already. So the addition of another thread in combination of the client-server model requires precise handling of objects and memory in order to work properly.

5.3 User interface and navigation

In this section the user interface will be explained. A view is basically a full screen look to a state of the application. Because the screen of an iPhone is limited in size there are no windows to control. This implicates that for almost every little action in the application there should be a view. Because for small tasks a whole view is used navigation with fingers gets really easy because the interface elements can be big enough to allow easy interaction.

The state of a view can thus usually be coupled with a certain function of the application. Discovery’s interface works on the principle of navigating between views. First a number of screenshots will be shown representing each view that can be found in Discovery. At the end of this section the navigation between different views will be explained.

5.3.1 Views of Discovery

Each view is a possible screen in Discovery. In this section different views from Discovery will be explained with their corresponding interface elements. The navigation between different views will be explained in the following chapter. Some trivial screens (like Login View) are not shown since they are assumed to be self explanatory. In many screens a tap bar controller is shown. This is a constant element in the interface that allows users to switch between Map, Favorites, History and Status Views.

Map View

In figure 5.2 a screenshot of a Map View is captured from Discovery. This is the screen where users location is displayed with a blue ball. The Pois around the user are displayed as bubbles displaying their rating by their size and classification by their icon. The functional classification of a Poi is displayed by a blue icon in the bubble. A green bubble shows the user that he/she created the bubble and an orange check mark shows that the bubble has been visited by the user.
The Pois are downloaded automatically from Innovattic server and placed on the map. The users can scroll through the map and can zoom in and out. At any time they can come back to their location using the Center on User button (it uses the same icon as the Google Maps application, so it should be recognizable by users). When a user taps a Poi a title is shown. This title can be the coordinate or street name + city name if the Poi is not yet visited. If the Poi is visited the title can be the name the user gives to it. Not yet implemented is the feature where as the creator of a Poi changes the name that name is shown if the user hasn’t given the Poi a custom name. The subtitle is always the number of visitors and the number of comments. The rating is shown by the size of the icon.

Figure 5.2: Screenshot of the Map View
Detail View

From the Map View the user can see the details of a Poi by touching the blue arrow that is in the callout of the Pois. Figure 5.3 shows what it looks like. This is what the user sees if a Poi is created or visited. If the Poi is created by the user than the rating control is not show as the action of creating a Poi in itself is a rating for that place.

A feature for a future version of Discovery should be a button that takes you to the Map View to the current Poi.

History View

The History View in figure 5.4 shows a list of visited and created Pois. The list shows the titles for the Pois with the amount of visitors and comments and the address of the Poi. When Discovery development continues the time/date of visit or creation should be displayed to give the Pois more context.

By tapping a Poi the user is taken to the Detail View of the corresponding Poi.
Figure 5.4: Screenshot of the History View

Figure 5.5: Screenshot of the Status View
**Status View**

The Status View in figure 5.5 shows the status of a user by displaying a title, a level and progress bar for the next level. This view can be extended in the future to contain the number of Pois a user has visited and the number of comments he/she has made. Also settings for the integration with social networks can go here.

**5.3.2 State machine**

In figure 5.6 different views are represented with a rectangle with the views name on it. Arrows represent actions that can be taken from each view. Every action starts and ends in a view. The tail of the arrow is connected to the view that the action was taken and the head of the arrow is connected to the view that the action ends in. For some actions the start and the end of an arrow can be the same view. This means action does not result in a change of views. For example if users mistype their passwords from the Login View, the application warns the user that the password does not match to the username, but the application still stays in the Login View for a retry or the creation of a new user. The figure shows how different views in Discovery are connected to others and by what actions they can change.

There are three places where rectangles are used with dotted lines. These represent grouping where different views are grouped together to avoid clutter. Favorite view is a subset of History view, so from both views the same set of actions can be taken. Comment and Tag views are also similar in terms of navigating into and navigating from. The grouping of Map, Status, History and Favorite view is a little different since different actions can be taken from these views. Here a grouping is made to show that users can switch between these views any time, using tap bar controls. Also once they turn off the application previously logged in users will return to any one of the these views when they turn on the application.

The begin and end point of the application is the double circle on the top left side of the figure. The application starts either on the Login View or in any of the tab views if the user was already logged in. The application can be turned of anytime bringing it to this end state. These arrows however are not explicitly drawn to avoid clutter. The ellipse with the label Visited Poi is drawn to underline the fact that Poi Detail View is reached from Map View if and only if that Poi is visited by the logged in user.
Figure 5.6: State machine of the views in Discovey
Chapter 6

Testing

In this chapter the tests will be discussed that have been performed on Discovery. There have been 2 test rounds, one functional test and a usability test.

6.1 Functional test

Because of time constraints there was no opportunity to perform a complete unit test on the code of Discovery. Another way to test the implementation that takes considerably less time is black box testing. The test consists of actions that have to be tested with pre-conditions and expected results. The tests are then performed on the application by a person and the actual results can be compared to the expected results to see how well the test went. This test is displayed in table 6.1 with the results.

From the results it shows that the Must haves from the MoSCoW principles have been implemented and that there is some of the functionality described in the Should haves that have not made the implementation, such as the ability to add tags to Pois and to favorite Pois. Everything else that did make the implementation seemed to work fine, though some issues were noticed in the usability test (see section 6.2).
<table>
<thead>
<tr>
<th>What to test</th>
<th>How to test</th>
<th>Pre-conditions</th>
<th>Expected Results</th>
<th>Actual Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a user account in the data server and be able to log in and out.</td>
<td>Create two user accounts and log in to one account. Afterwards log out and log into to the other account two times.</td>
<td>Both accounts should have different experiences, one favorite (different for each), only two visited Pois (only one in common).</td>
<td>Users status, visited Pois and favorites must change accordingly as another user logs in.</td>
<td>Accounts were successfully created and login worked fine. Status/experience were updated to the values send from the server.</td>
</tr>
<tr>
<td>Download Pois from the server and display them on the map.</td>
<td>Turn the application on and scroll a bit in the map. Afterwards log in to another account and view the same parts as with the last account.</td>
<td>There should be Pois in the parts of the map that are displayed. There should also be two user accounts in the server. There should also be at least one visited Poi.</td>
<td>In the beginning the application should download Pois from the server and display them on the map. As the user scrolls through the map, new Pois should appear. Visited Poi should be displayed differently.</td>
<td>Pois are displayed when viewed with both accounts. If a Poi is visited by the current user, then it is displayed differently.</td>
</tr>
<tr>
<td>Recognize when the user arrives to a unvisited Poi.</td>
<td>The user stands between two Pois. Walks to the each of them in turn.</td>
<td>One of the Pois should be visited by the user and the other one not. The visited Poi should already be in the visited list.</td>
<td>As the user comes close to a Poi within 5 meters and the Poi is an unvisited one, it should turn to visited and be added to the users visited list. Otherwise, the system should not react.</td>
<td>The Poi was visited with success. Though buildings can be problematic for the iPhone’s GPS capabilities.</td>
</tr>
<tr>
<td>Recognize when the user arrives to a unvisited Poi.</td>
<td>The user stands between two Pois. Walks to the each of them in turn.</td>
<td>One of the Pois should be visited by the user and the other one not. The visited Poi should already be in the visited list.</td>
<td>As the user comes close to a Poi within 5 meters and the Poi is an unvisited one, it should turn to visited and be added to the users visited list. Otherwise, the system should not react.</td>
<td>The Poi was visited with success. Though buildings can be problematic for the iPhone’s GPS capabilities.</td>
</tr>
<tr>
<td>Retrieve and Update Poi details.</td>
<td>Attempt to come to the Poi details screens of three Pois. Attempt to change the title, comments and rating of 2 Pois. Then log in as a different user to observe the effects.</td>
<td>One of the Pois should be created by the user(1), the other should be visited(2) and the last one not visited(3).</td>
<td>The user should not be able to see all of the detail screen of Poi 3. For Poi 1 and 2 all the information should be updated. When logged in as a different user the updated values should be changed on Poi 1, but not for Poi 2. The rating should be visible by the change in size of a Poi.</td>
<td>The rating and comments were updated correctly, but the title updates were never displayed for other users. The rating was updated, but the visible changes were too small to see (the scale was too big), though debugging showed that it did work.</td>
</tr>
<tr>
<td>What to test</td>
<td>How to test</td>
<td>Pre-conditions</td>
<td>Expected Results</td>
<td>Actual Results</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create a new Poi</td>
<td>First try to create a Poi avoiding functional classification. Create a new Poi, and than log in as a different user to see to verify the newly created Poi.</td>
<td>There are enough free space around the place that the user wants to make a new Poi.</td>
<td>It should not be possible to create a Poi without the functional classification. No effect if not enough space is available, if enough space is available a new Poi is created and added to the visited list of the user. When logged in as a different user, the Poi should be visible on the map.</td>
<td>Discovery shows an alert when trying to add a Poi without having selected a functional classification. The restriction on the space between Poi combined with the fluctuation of the GPS receiver makes it possible to add a lot of Poi almost at the same location without even moving. Poi is shown to other users.</td>
</tr>
<tr>
<td>Gain experience by visiting Pois.</td>
<td>Go to multiple Pois and observe the changes in users experience from the user status view.</td>
<td>The user should go to both visited and unvisited Pois.</td>
<td>Visiting already visited Poi should not have an effect on the experience. By visiting unvisited Pois the experience of the user should increase.</td>
<td>Not implemented, this is something for server side.</td>
</tr>
<tr>
<td>Keep track of visited and favorite Pois</td>
<td>Log in and out between two different accounts.</td>
<td>Two users should have two common and two different visited Pois. Also they should have two different and two same favorite Pois.</td>
<td>As we log in and out the visited and favorite Pois should change accordingly. The Pois should be displayed in a favorite and history list, from which each Pois details must be reachable.</td>
<td>The Pois are displayed different for each user for each state the Poi is in. Favorites isn’t implemented.</td>
</tr>
</tbody>
</table>

### 6.2 Usability test

The usability test was done using a method called Hallway Usability Testing hal where the idea is to get a small number of random people to test the application on performance, accuracy, recall and emotional response.

Because of time constraints and the fact that not every random person in the hallway has an iPhone (or experience in using one) the test has been done with people that the project team knew to have an iPhone.

#### 6.2.1 Test plan

The 4 test persons involved were told a minimal amount of information about what Discovery can do. No explanation of the user interface was provided so the test persons had to find out for themselves how the application worked.

Each test person was given a personal assignment with goals he (all testers were male) should try to achieve. This was to get the testers an idea where to begin and to encourage them to navigate through the whole application. The goals were actions like ”Create 3 bubbles” and ”Visit 2 bubbles not created by you and leave a comment”.

The test was held in the TU Delft and people walked around for almost an hour making, visiting and rating Pois. One of the project team walked around outside to answer questions if...
needed and to see if the application worked well. The other project team member monitored
the data traffic to and from the server.

6.2.2 Results

The results are put into three categories to give a clear overview of the different aspects of the
test.

Performance and stability

About the performance of the application were no complaints although the dataset that the
application had to work with wasn’t that big.

Some test persons told of stability problems that were encountered mainly when trying to
give a custom name to a Poi. Though most of the time after a few restarts of the application
the users could succeed in giving custom names. There were also a few problems with entering
comments, but not as much and as frequent as the custom name problems.

The performance on the server side showed that there was a lot of processor time used for
the request of iPhones to get new Pois. This method is called on the iPhone every time the user
scrolls a little bit in the Map View, so it gets called a lot of times. This can easily be optimized
by the use of timers and conditions for the returning of Pois so that for example only once an
hour an update is requested from the server. In a real life situation this would probably suffice
because the need for the latest up to date information is not necessary for the functionality of
the application.

Accuracy

A lot of accuracy problems were detected with the GPS receiver of the iPhone. When the
test persons tried to add Pois sometimes the iPhone would think that they were at a complete
different spot. This was especially the case when the application is just opened and when the
device is updating it’s position.

This problem can be fixed by making an element in the Add Poi View that displays a fine
tuner so that the user can adjust the actual position of the Poi that is to be created. This
should be limited to a certain region in order to prevent users from creating Pois at complete
different places from where they actually are.

Emotional response

The overall response was very positive. A lot of test persons found that it was nice to see other
people’s comments on Pois. The idea of sharing interesting places was appreciated and the test
persons said that they were interested in future developments.

One of the test persons told that he really liked the fact that he could not get the information
of unvisited Pois. It made him curious about Pois and said that it added to the fun factor of
the application. He said that: “The iPhone lacks an application like this, it’s a new concept and
it stands out from the rest. The interactive elements make it great to play with.”.
Chapter 7

Conclusion

Discovery is an application that enables people to discover their surroundings in their leisure time. It runs on an iPhone and uses GPS technology to obtain (and update) user location. Interesting locations in the walking distance are displayed on a map, downloaded from Google Maps. An iPhone always comes with an Internet connection so the data about the locations are to be stored in a server that will be maintained by Innovattic. It is free software that focuses on hiding certain information in order to make places more interesting.

The concept Poi (point of interest) plays a central role for the application. Pois contain on their minimum a functional classification and a location (for example an entertaining place around the corner). There are optional fields for a title, comments, emotional classification and rating. The Pois and the feedback is the whole content of the software and is fully provided by the users. A rating system is in place to separate the good Pois from the poor ones. Pois with higher ratings will be displayed bigger on screen. On the other hand poor Pois will become smaller and smaller as they get negative ratings and eventually disappear. Users can also report Pois as inappropriate. The Pois that are unvisited do not display any information other than their classification and location. Once users reach a Poi they are rewarded with the full content and are allowed to provide feedback on the Poi in the form of rating, tags and comments. There is also room for personal data in Discovery. Although a Pois title can only be set by the creator of a Poi, anyone can change the title for their own use.

The users run the application with an account that saves their user information. Users have an experience (giving them a title on their status screen like “explorer”) and a list of Pois that they have visited and favorited. This information is relatively small so it is saved both in the iPhone for the speedy start up and at Innovattics server to allow a user to reach his own data from another iPhone. The users can provide feedback or change Poi details later on once they have visited them. Because the critical information is saved locally Discovery can still be used (although not up to date) when the user can not make an Internet connection for any reason.

For every Poi there is a functional and an emotional classification. Discovery recognizes the personal differences of users. That is why the classification paradigm for Pois used in Discovery is of a very high level and functionally discreet. On the functional side a simple classification is chosen that is used by many tourist services. On the emotional side, the Pois are mapped on two scales. One is about the active-relax dimension of the place and the other one is about the fresh-established dimension. For every option of the functional classification there is an unique icon that can be placed on the map. For the display of the emotional classification color is used. The color of the Poi depends on how much it scores on both emotional dimensions.

Overall Discovery confirms to Apple Guidelines for the interface design. Also in order to motivate its users it is warm and contains (where possible) positive emotions. Discovery favors achievement via experience driven leveling rather than competition. Discovery’s vital content
providing actions contain minimum amount commitment and deliberation. Each action of creating a new Poi or rating/tagging a visited Poi can be accomplished in under 3 seconds. People who are goal driven can be motivated by gaining experience and accomplishing to get to the next title. Discovery is uncensored providing freedom and offers a wide range of unknown places to the users and so provides stimulation of exploration.
Chapter 8

Recommendation for Innovattic

During the different stages of the project team came up with lots of ideas that would be nice assets to Discovery. Users also added to this process by providing their own original ideas during both user research and user testing. After discussions between the project team members and Innovattic some of these ideas were eliminated while others were marked as promising. The most promising ideas were also researched as a part of the contract between Innovattic and the project team. The following ideas were the ones that were selected and probably would need to be implemented for a market ready product.

8.1 Favorites

Talking to users (see section 3.2 for the user research) it was apparent that users wanted to separate some special points from rest of the points that they have visited. The concept of favorites is also a very common concept in other applications (like YouTube, Deviantart etc.) where the content is provided by the users. Also the iPhone allows its users to favorite almost anything in their interfaces from applications to contacts. So a subset of the history page can be made with a simple interface element where users can flag their favorite places as favorites. Necessary hooks are already built into the system for further development.

8.2 Tags

Tags (in the context of Discovery) are positive adjectives that are associated to a Poi by a user. The user research also revealed that users did not always want to type whole comments to a place from a hand held device. Leaving a tag from a preselected list instead of writing down the whole comment does not have the expressive freedom of a comment but it is convenient. Also having a tag selected from a positive list of adjectives means that users will be lead to seeing more positive things of the places they visit. Necessary hooks for implementing such idea are already in the system architecture and the database design.

8.3 Bubbles

Many applications running on smart phones that are oriented for leisure activities have a theme for their interface elements. This can add character to the application to make it more appealing. It must be noted that more research is needed to conclude that implementation of a theme would make an application appealing. Having said that, the most memorable theme idea from the prototyping phase of Discovery was Bubbles. The idea is to implement each Poi as a bubble (in
the Map View interface) that emits random tags that are attached to that Poi. These emissions can be implemented as smaller bubbles (in compression to the Pois) which would pop after a while. More popular Pois can emit tags more frequently in effect making them more visible to users.

8.4 Plug in for social networks

In the Internet there are many social networks (Hyves, Facebook, etc.) available. The lightweight incremental nature of the Discovery application fits quite nicely into scheme of publicizing private life. So in future releases a possibility can be made available to synchronize the application with ones social network and display the users experience, level, visited places and favorites.

8.5 Suggest Pois to the users

There are two ways identified by the project team to suggest Pois to a user. For one, an analysis of the users favorited Pois can reveal Pois with similar classification and tags. Another method can be, instead of trying to find similar Pois, one can find similar users and simply suggest favorited Pois to each other. There are other methods, yet they all (including the ones above) need some kind of AI or datamining. That is why this functionality is left out for later releases. Once a user has a list of Pois that is favorited, the list can potentially be shared with other users. Yet this will also mean more implementation time.

8.6 Photos

The users could be allowed to add photos to the Pois that they have visited. This property of the application will allow users to look back at the Pois that they have been later on, and relive those moments. These photo’s however should not made public to other users since photo content is notoriously subject to misuse and inherently hard to control.
Bibliography


Hallway testing. URL http://en.wikipedia.org/wiki/Usability_testing#Hallway_testing.


A to z of positive word. URL http://www.mindmapinspiration.com.


Appendix A

Scenario

Marloes is op vakantie met haar vriend in Parijs. Ze hebben alle toeristische plekken al gezien en ze hebben nog een middag over. Om die middag goed te besteden willen ze graag wat dingen zien die niet in de toeristische boekjes staan. Marloes pakt haar iPhone en start Discovery. Ze zien een aantal punten op het scherm verschijnen bij hun in de buurt in de vorm van belletjes. Ze zien aan de grote van de belletjes hoe populair een punt is en aan de doorzichtigheid hoe een punt is beoordeeld. Ze zien echter niet wat voor een plek het is. Ze zien dat er op 5 minuten loopafstand een leuk punt is en ze besluiten om erheen te gaan. Ze zien een heel mooi parkje waar de rust van af straalt. Ze hebben een leuke tijd in het park en besluiten om dit punt een goede beoordeling te geven.

Onderweg vinden ze een vrolijk gekleurde Caribische bar waar het gezellig druk is. Marloes ziet dat deze plek leuk is om te delen met anderen, dus ze voegt deze locatie toe.

Thuisgekomen van de vakantie zit het stel met vrienden op de bank om vakantieverhalen te vertellen. Marloes pakt haar iPhone en vertelt over die spontane middag in Parijs aan de hand van de bezochte punten. Ze laat gelijk de foto’s zien die ze gemaakt hebben bij deze locaties.

1. Wat zou je willen toevoegen aan het systeem?
2. Wat zou je willen veranderen?
Appendix B

Pictures of random places

Figure B.1: Pictures used for the user research
## Appendix C

### User research words

<table>
<thead>
<tr>
<th>Words</th>
<th>Number of times used</th>
<th>Status</th>
<th>Reason</th>
<th>Words</th>
<th>Number of times used</th>
<th>Status</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affectionate</td>
<td>11</td>
<td>Kept</td>
<td>High frequency</td>
<td>Loveable / Loving</td>
<td>14</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Alive</td>
<td>18</td>
<td>Kept</td>
<td>High frequency</td>
<td>Passionate</td>
<td>12</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Amusing</td>
<td>19</td>
<td>Kept</td>
<td>High frequency</td>
<td>Peaceful</td>
<td>26</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Accepting</td>
<td>9</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Playful</td>
<td>12</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Beautiful</td>
<td>19</td>
<td>Removed</td>
<td>Too ambiguous</td>
<td>Pleased / Pleading</td>
<td>5</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Calm</td>
<td>19</td>
<td>Kept</td>
<td>High frequency</td>
<td>Proud</td>
<td>11</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Caring</td>
<td>8</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Quiet</td>
<td>2</td>
<td>Kept</td>
<td>Bad presentation of the word in the list</td>
</tr>
<tr>
<td>Comfortable</td>
<td>9</td>
<td>Kept</td>
<td>Not enough comfortable pictures in the list</td>
<td>Relaxed</td>
<td>21</td>
<td>Removed</td>
<td>“relaxed” will be used for emotional classification</td>
</tr>
<tr>
<td>Cheerful</td>
<td>9</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Relieving</td>
<td>3</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Competent</td>
<td>1</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Respected / Respectable</td>
<td>10</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Concerned</td>
<td>2</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Safe / Secure</td>
<td>18</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Courageous / Brave</td>
<td>10</td>
<td>Kept</td>
<td>High frequency</td>
<td>Satisfying</td>
<td>11</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Confident</td>
<td>2</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Silly</td>
<td>1</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Content</td>
<td>2</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Special</td>
<td>1</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Curious</td>
<td>18</td>
<td>Kept</td>
<td>High frequency</td>
<td>Strong</td>
<td>12</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Delightful</td>
<td>5</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Supportive / Supporting</td>
<td>5</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Dearable</td>
<td>6</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Sympathetic</td>
<td>10</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Eager</td>
<td>1</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Tender</td>
<td>8</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Exciting</td>
<td>16</td>
<td>Kept</td>
<td>High frequency</td>
<td>Great</td>
<td>10</td>
<td>Removed</td>
<td>Too ambiguous</td>
</tr>
<tr>
<td>Forgiving</td>
<td>7</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Grateful</td>
<td>6</td>
<td>Removed</td>
<td>Low frequency</td>
</tr>
<tr>
<td>Friendly</td>
<td>12</td>
<td>Kept</td>
<td>High frequency</td>
<td>Happy / Glad</td>
<td>20</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
<tr>
<td>Fulfilling</td>
<td>5</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Humorous / Funny</td>
<td>5</td>
<td>Kept</td>
<td>Not enough funny pictures in the list</td>
</tr>
<tr>
<td>Generous</td>
<td>7</td>
<td>Removed</td>
<td>Low frequency</td>
<td>Joyful</td>
<td>16</td>
<td>Kept</td>
<td>High frequency</td>
</tr>
</tbody>
</table>

Table C.1: Frequency of the used words in the user research
Table C.2: Words provided by the users

<table>
<thead>
<tr>
<th>Words</th>
<th>Status</th>
<th>Reason</th>
<th>Words</th>
<th>Status</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affectionate</td>
<td>Accepted</td>
<td>Delightful</td>
<td>Alive</td>
<td>Accepted</td>
<td>Pleased</td>
</tr>
<tr>
<td>Amusing</td>
<td>Accepted</td>
<td>Playful</td>
<td>Antique</td>
<td>Accepted</td>
<td>Interesting</td>
</tr>
<tr>
<td>Accepting</td>
<td>Accepted</td>
<td>Positive quality</td>
<td>Angry</td>
<td>Rejected</td>
<td>Negative</td>
</tr>
<tr>
<td>Exciting</td>
<td>Accepted</td>
<td>Interesting</td>
<td>Buzzing</td>
<td>Accepted</td>
<td>Used by multiple people in the group</td>
</tr>
<tr>
<td>Small</td>
<td>Accepted</td>
<td>Interesting</td>
<td>Overwhelming</td>
<td>Accepted</td>
<td>Used by multiple people in the group</td>
</tr>
<tr>
<td>Accepting</td>
<td>Accepted</td>
<td>Positive quality</td>
<td>Order / Organized</td>
<td>Accepted</td>
<td>Interesting</td>
</tr>
<tr>
<td>Sensual</td>
<td>Rejected</td>
<td>Too specific</td>
<td>Threatening</td>
<td>Rejected</td>
<td>Positive quality</td>
</tr>
<tr>
<td>Funky</td>
<td>Accepted</td>
<td>Interesting</td>
<td>Impersonal</td>
<td>Rejected</td>
<td>Negative</td>
</tr>
<tr>
<td>Interesting</td>
<td>Accepted</td>
<td>Positive quality</td>
<td>Static</td>
<td>Rejected</td>
<td>Boring</td>
</tr>
<tr>
<td>Adventurous</td>
<td>Accepted</td>
<td>Positive quality</td>
<td>Sexy</td>
<td>Accepted</td>
<td>Positive quality</td>
</tr>
<tr>
<td>Wonderful</td>
<td>Rejected</td>
<td>Too ambiguous</td>
<td>Romantic</td>
<td>Accepted</td>
<td>Positive quality</td>
</tr>
<tr>
<td>Surprising</td>
<td>Accepted</td>
<td>Positive quality</td>
<td>Youthful</td>
<td>Rejected</td>
<td>Too specific</td>
</tr>
<tr>
<td>Undressed</td>
<td>Rejected</td>
<td>Not commonly applicable to places</td>
<td>Sad</td>
<td>Accepted</td>
<td>Can be interesting</td>
</tr>
<tr>
<td>Boring</td>
<td>Rejected</td>
<td>Negative</td>
<td>Angry</td>
<td>Rejected</td>
<td>Negative</td>
</tr>
<tr>
<td>Cold</td>
<td>Rejected</td>
<td>Negative</td>
<td>Cool</td>
<td>Accepted</td>
<td>Positive quality</td>
</tr>
<tr>
<td>Hungry</td>
<td>Rejected</td>
<td>Not commonly applicable to places</td>
<td>Smells good</td>
<td>Accepted</td>
<td>Interesting</td>
</tr>
</tbody>
</table>

Table C.3: Words presented to the users

<table>
<thead>
<tr>
<th>Words</th>
<th>Status</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affectionate</td>
<td>Delightful</td>
<td>Playful</td>
</tr>
<tr>
<td>Alive</td>
<td>Desirable</td>
<td>Pleased</td>
</tr>
<tr>
<td>Amusing</td>
<td>Eager</td>
<td>Proud</td>
</tr>
<tr>
<td>Accepting</td>
<td>Exciting</td>
<td>Quiet</td>
</tr>
<tr>
<td>Beautiful</td>
<td>Forgiving</td>
<td>Relaxed</td>
</tr>
<tr>
<td>Brave</td>
<td>Friendly</td>
<td>Relieving</td>
</tr>
<tr>
<td>Calm</td>
<td>Fulfiling</td>
<td>Respected</td>
</tr>
<tr>
<td>Capable</td>
<td>Generous</td>
<td>Safe</td>
</tr>
<tr>
<td>Caring</td>
<td>Glad</td>
<td>Satisfying</td>
</tr>
<tr>
<td>Cheerful</td>
<td>Grateful</td>
<td>Secure</td>
</tr>
<tr>
<td>Cherishing</td>
<td>Great</td>
<td>Self-reliant</td>
</tr>
<tr>
<td>Comfortable</td>
<td>Happy</td>
<td>Sexy</td>
</tr>
<tr>
<td>Competent</td>
<td>Humorous</td>
<td>Silly</td>
</tr>
<tr>
<td>Concerned</td>
<td>Joyful</td>
<td>Special</td>
</tr>
<tr>
<td>Confident</td>
<td>Lovable</td>
<td>Strong</td>
</tr>
<tr>
<td>Content</td>
<td>Loving</td>
<td>Supportive</td>
</tr>
<tr>
<td>Courageous</td>
<td>Passionate</td>
<td>Sympathetic</td>
</tr>
<tr>
<td>Curious</td>
<td>Peaceful</td>
<td>Tender</td>
</tr>
</tbody>
</table>
Appendix D

Project planning

Figure D.1: Project planning with its different stages
# MoSCoW Principles

### Must Have
- Concentrated effort for the basic functionality. Foundation for the prototype. The simplest way to implement every functionality in the way the Discovery will work and how the different modules of the system will communicate with each other.

### Should Have
- Prototype as it will be handed in to InnovAttic at the end of the project. Some extensions to “must haves”. A dry working product enough to represent what the end product can be. Not made to deal with great amount of data.

### Could Have
- End product as it can be offered to users. All functionality is finalized and polished in a way that the necessary functionalities to encourage users to use it. Made to be scalable and reusable.

### Want to Have (won’t have)
- All the ideas that are interesting, appealing, functional yet cannot be implemented due to time cost & quality constraints.

### Appendix E

#### Figure E.1: MoSCoW Principles of Discovery

<table>
<thead>
<tr>
<th>Must Have</th>
<th>Should Have</th>
<th>Could Have</th>
<th>Want to Have (won’t have)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Concentrated effort for the basic functionality. Foundation for the prototype. The simplest way to implement every functionality in the way the Discovery will work and how the different modules of the system will communicate with each other.</td>
<td>- Prototype as it will be handed in to InnovAttic at the end of the project. Some extensions to “must haves”. A dry working product enough to represent what the end product can be. Not made to deal with great amount of data.</td>
<td>- End product as it can be offered to users. All functionality is finalized and polished in a way that the necessary functionalities to encourage users to use it. Made to be scalable and reusable.</td>
<td>- All the ideas that are interesting, appealing, functional yet cannot be implemented due to time cost &amp; quality constraints.</td>
</tr>
</tbody>
</table>

---

- Connect to a data server
- Create an user account in the data server
- Log in with your own user account
- Connect to the map server
- Display the map
- Download map from the map server
- Display Pois as data on the map
- Retrieve user location, and place it on the map
- Move the user on the screen as he walks in the world
- Recognize when user has arrived to a landmark or PoI
- Keep track of visited Pois
- Display previously visited Pois different than others
- Retrieve details (only title) from the visited Pois
- Update details of a visited PoI in the most simple-way (rating +1)
- Create a new PoI on the map only with GPS location & optional title
- Ability to go out.

- When a PoI is created without a creator provided title, the street name is downloaded from the maps server and used as a title to be displayed in the history.
- Download Pois only if they are from a predefined distance to the user
- Ability for users to attach tags to visited Pois.
- Ability for users to upload comments to visited Pois
- Ability for users to change titles of the PoIs (this is only visible to that user)
- By creation of a PoI compulsory functional & emotional classification. By creation of a PoI compulsory add opening time and change title.
- PoI details page is expanded with PoI’s comments, tags, icon, opening/closing times.
- Implement a history page of the visited Pois, where the PoI details can be reached.
- Display Pois rating by its transparency.
- Users gain experience as they visit Pois.
- Users gain experience as they visit Pois
- Ability to search for Pois in emotional or functional classification.
- Allow Pois to disappear when their rating drops a predefined level.
- Display a PoIs tags as bubbles.
- More popular PoIs (the ones that are more visited) emit more bubbles.
- Ability to download Pois from another GPS location and then use this on the offline mode.
- Add photos to the PoIs (personal use only)
- Log in with as a different user.
- As the users gains experience, gains new titles and rights.
- The user gain experience as the PoIs he made gets visited or positively rated
- Start up the application from the window that it is turn off from.
- Remind users to rate “comment a PoIs.
- Ability to favorite PoIs.
- Visit favorited PoIs as a subset of the history
- Remind users to rate “comment a PoIs.
- Suggest PoIs to the users that they might like.
- Plugins for social networks.
- Ability to import / export routes, favorites and history.
- Presence for users with challenges.
- Temporary PoIs that appear is limited for example but are more visible as they exist, since they have little time to grow.
- Remind users to rate “comment a PoIs.
- And all the other good ideas from the wiki page...