Location based functionality in public transport applications

Jacob Arvidsson and Jonathan Vidmar

CERTEC | DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2016
MASTER THESIS
Location based functionality in public transport applications

Implementation and evaluation from a user perspective

Jacob Arvidsson          Jonathan Vidmar
dat11jar@student.lu.se   dat11jvi@student.lu.se

June 15, 2016

Master’s thesis work carried out at Softhouse Consulting Öresund AB.

Supervisors: Kirsten Rassmus-Gröhn,
kirsten.rassmus-grohn@certec.lth.se
Lars Isberg, lars.isberg@softhouse.se
Examiner: Joakim Eriksson, joakim.eriksson@design.lth.se
Abstract

In recent years, a new technology for location based functionality has emerged. The technology is called Bluetooth Low Energy, or Bluetooth Smart, and allows a device to emit low energy Bluetooth signals to its surroundings. When picked up by modern smart phones, these signals can be used to provide the user with a physical location. Unlike other positioning technologies, such as GPS, this technology can be used at closer distances, especially indoors, and with the added advantage of low battery consumption.

An area which could benefit from positioning technology is public transportation. As travelers are moving between different locations, accurate low energy positioning could provide contextual information based on their current location, both indoors and in moving vehicles.

In this thesis, a proposed solution is presented as an Android application, showcasing location based functionality. This functionality, enabled by Bluetooth Smart, improves the traveling experience by simplifying and removing many of the steps of interaction needed in current applications. From the usability tests performed, we can conclude that context aware functionality provides a more convenient and time-saving experience for the user. However, functionality requiring more accurate distance estimation still needs to be developed further before implementations in real life applications can be done.

Keywords: Location based functionality, Public transport, Bluetooth Low Energy (BLE), Bluetooth Smart, iBeacon, Eddystone, Usability, Interaction Design
Acknowledgements

We want to extend a big thank you to Softhouse Consulting Öresund, especially our supervisor Lars Isberg and our contact person Ola Person. They and the rest of Softhouse have been most helpful during this whole process, providing us with a place to work as well as constant feedback.

We would also like to thank our supervisor from LTH, Kirsten Rassmus-Gröhn, who has given us guidance during this thesis and aided us with knowledge about the interaction design area and usability test process.

And lastly, we would like to thank everyone who participated in our usability tests. Without your help this thesis work would not have been possible.
## Contents

<table>
<thead>
<tr>
<th>1 Introduction</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Background</td>
<td>9</td>
</tr>
<tr>
<td>1.2 Purpose and goals</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Softhouse Consulting</td>
<td>10</td>
</tr>
<tr>
<td>1.4 Contribution</td>
<td>10</td>
</tr>
<tr>
<td>1.5 Distribution of work</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2 Technical background</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Bluetooth and Bluetooth Smart</td>
<td>13</td>
</tr>
<tr>
<td>2.2 Beacon protocols</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1 iBeacon</td>
<td>14</td>
</tr>
<tr>
<td>2.2.2 Eddystone</td>
<td>14</td>
</tr>
<tr>
<td>2.3 Indoor positioning</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1 Received Signal Strength Indication and The Log-Distance Path Loss Model</td>
<td>15</td>
</tr>
<tr>
<td>2.3.2 Kalman filtering</td>
<td>16</td>
</tr>
<tr>
<td>2.3.3 Self-correcting beacons</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 Approach</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Pre-study</td>
<td>19</td>
</tr>
<tr>
<td>3.1.1 Literature study</td>
<td>20</td>
</tr>
<tr>
<td>3.1.2 Brainstorming</td>
<td>20</td>
</tr>
<tr>
<td>3.1.3 Interviews</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Prototyping method</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Evaluation methods</td>
<td>21</td>
</tr>
<tr>
<td>3.3.1 Demo-sessions</td>
<td>22</td>
</tr>
<tr>
<td>3.3.2 Usability testing</td>
<td>22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4 Results</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Technology</td>
<td>25</td>
</tr>
</tbody>
</table>
## 4.1.1 Distance estimation

### 4.1.2 Distance estimation evaluation

## 4.2 Functionality

### 4.2.1 Context awareness

### 4.2.2 Smart notifications

### 4.2.3 Ticket purchase

### 4.2.4 Ticket validation

## 4.3 Usability testing

### 4.3.1 First round of testing

### 4.3.2 Improvements

### 4.3.3 Final round of testing

### 4.3.4 Survey

### 4.3.5 Product reaction word list

## 5 Discussion

### 5.1 Methodology

#### 5.1.1 Brainstorming

#### 5.1.2 Scientific research

#### 5.1.3 Interviews

#### 5.1.4 Prototyping

### 5.2 Technology

#### 5.2.1 Estimation of distance

#### 5.2.2 Maintenance and battery consumption

#### 5.2.3 Communication between validation point and user device

### 5.3 Usability testing

#### 5.3.1 First round

#### 5.3.2 Final round

### 5.4 Future work

## 6 Conclusion

## Bibliography

## Appendix A Interviews

### A.1 Questions

### A.2 Answers

## Appendix B Usability test plan

### B.1 Test plan 1.0

#### B.1.1 Overall objective for the study

#### B.1.2 Research questions

#### B.1.3 Measures

#### B.1.4 Location and setup

#### B.1.5 Methodology

#### B.1.6 Participants

#### B.1.7 Moderator role

#### B.1.8 Task list
Chapter 1
Introduction

In this chapter a brief introduction will be given to the background, the main issue and the goals of the thesis. Our contribution to the field will also be outlined.

1.1 Background

With the introduction of Internet of Things, many companies now seek different ways to interact with users and give a more personal and context based user experience. This is achieved by determining what is important to users and when, and providing them with this information. One way to do this has been with the use of Bluetooth Smart / Bluetooth Low Energy (BLE) devices. In the rest of the report these will be referred to as beacons. By using beacons, context awareness or location based functionality can be introduced in mobile phone applications in order to provide smarter interaction.

One area which could be improved by location based functionality is public transportation. Today the process of interacting with public transportation is difficult and time-consuming, with applications requiring the user to search and ask for the information explicitly. By introducing location based functionality this process could be simplified giving travelers a more enjoyable trip. This type of context awareness would improve the interaction both at stations and in transport vehicles.

Another part of the interaction process in these environments is the validation of a purchased ticket. We define validation as the process where the user in some way shows the public transport company that the ticket they have purchased is valid. Currently popular methods for ticket validation in public transport applications has been implemented with the use of either matrix barcodes e.g. Quick Response (QR) codes or Near Field Communication (NFC), or by simply buying and showing the ticket through an application. The future for public transport application could be to automate the validation task resulting in less user interaction or no user interaction at all.

An example of this would be when a user is located a certain distance from a beacon,
the validation action would be performed. This would give the possibility for a user to buy a ticket and validate it, with their phone still in their pocket. Similar interaction could also be interesting in other areas, as this serves to automate routine tasks and improve the experience for the user.

1.2 Purpose and goals

The purpose of this thesis has been to investigate different location based functionality in a public transportation environment. In traditional applications, beacons are usually used to provide the user with a location, to enable the application to show information based on the specific location. This is done with a minor impact on user’s battery life, since it does not need to use GPS, Wi-Fi or other high battery draining technologies. In this thesis we have investigated how this can be done in a public transportation application. Furthermore we have looked into how beacons can be utilized in order to provide new ways of interaction and additional functionality that extends that of simply displaying information. The goals for the thesis work are listed below:

- Research the different ways beacon technology can be used in public transportation environments. This will include using already existing techniques to display context information for the user, but also looking into new functionality to enable automatic ticket validation.

- Implement an Android prototype to give a concept of how a public transportation application, centered around the user’s location, would work.

- Study users thoughts and opinions of using location based functionality, by means of beacon technology as interaction method. This will be carried out by performing minor interviews and larger usability tests.

1.3 Softhouse Consulting

The thesis work was done in cooperation with the IT consultant company Softhouse Consulting Öresund AB. They have extensive experience in implementing applications with payment methods for public transport systems. The thesis was suggested by them since they have seen a demand for smarter solutions concerning public transportation, with the use of beacon technology. During the thesis they have contributed with both continuous feedback and their extensive knowledge in the area.

1.4 Contribution

This thesis work will help to contribute in developing the next generation of public transportation applications. Since modern smart phones were invented, the public transportation applications has mostly looked the same, with a travel planner and in later years an option to buy tickets inside the application. During this thesis, smarter ways of interacting with public transportation has been explored and a prototype has been developed to test
these features. This will help the public transportation industry with smarter functionality but also show how users react to this functionality and that there is a demand for smarter technical solutions.

1.5 Distribution of work

Throughout the thesis we, the authors, have worked in parallel, dividing the tasks between us as they surface. Some parts of the thesis work required both of us to work together in cooperation, e.g. the brainstorming session and some especially hard issues that arose while developing the prototype. In general however, we worked separately albeit on the same area of focus.
Chapter 2
Technical background

This chapter will cover the main technical parts of the thesis, starting with describing the Bluetooth technology, Bluetooth Smart and beacons. The different beacon protocols are outlined and the theory used behind indoor positioning is explained.

2.1 Bluetooth and Bluetooth Smart

Bluetooth has during the last decades become popular, due to typical uses like mobile handsets, headsets, speakers, human interface devices (mice and keyboards) and even within cars. Bluetooth is both a protocol and a technology to transfer data wireless over short distances. Both parts are governed by The Bluetooth Special Interest Group (Bluetooth SIG), which is a collaboration of over 27 000 companies as 2015 [1].

In 2006 Nokia introduced the new Bluetooth Smart, or Bluetooth Low Energy (BLE), which is a new type of Bluetooth communication aimed to lower the energy consumption of existing and future Bluetooth devices [2]. In 2010 the Bluetooth Smart protocol was merged with the Bluetooth Core Specification [1]. The Bluetooth Smart protocol has been adopted by many mobile manufacturers because it is a technology suitable for connecting devices to the Internet of Things, due to its low energy consumption.

2.2 Beacon protocols

Beacon protocols are simply protocols that enable beacons, a type of peripherals, to communicate with devices such as mobile handsets or other devices with support for Bluetooth Smart. In this report we have studied two different protocols, those being Apple’s iBeacon protocol and Eddystone by Google. iBeacon, announced by Apple at WWDC in 2013, is the more popular of the two seeing as it has been around much longer, with numerous
applications in the industry. Eddystone however provides extended functionality to that of iBeacon in addition to being fully open source.

2.2.1 iBeacon

Built upon Bluetooth Smart, iBeacon is a proximity protocol created by Apple which allows for communication with BLE devices, supported by iOS 7 and higher [3]. With the use of iBeacon, a beacon can create a region around itself which is done by so called advertisement. The beacon sends out advertisements at short intervals, which can be picked up by nearby devices. These advertisements or packets contain, among other protocol specific information, an UUID, a major value, and a minor value. These values are what identifies each beacon. The major and minor values, simply being values in the range of 0 to 65 535, gives the possibility to divide the beacons into sub-regions. The UUID is large number highly probable of being unique. For instance, a chain of department stores could have one UUID, while each specific store would have a major value, and each department of that store would have a specific minor value [3].

The iBeacon protocol measures accuracy and proximity using the received signal strength indicator, RSSI. When the signal strength is high, it is highly probable the beacon is near, but if the strength is weak it is uncertain how far away the beacon is. This RSSI signal can not always be relied on since signal attenuation affects its value and creates a very noisy result. This is discussed more in section 2.3.

2.2.2 Eddystone

In 2015 Google released its own protocol for working with Bluetooth Smart, called Eddystone, with support for both Android and iOS. Unlike iBeacon, Eddystone is fully open source and in addition to that, supports three different data frames to be used individually or in combination [4].

**Eddystone-UID**

This is the corresponding functionality to that of iBeacon, which lets mobile applications locate beacons advertising a simple ID.

**Eddystone-URL**

Instead of an ID like the frame above, this frame advertises a URL of limited length to all the devices near the beacon. This enables regular websites to provide similar contextual information which was previously only available for native applications. Eddystone-URL works with Chrome widget on iOS and as of the spring of 2016 also on Android, allowing users to interact with URL’s transmitted by beacons through Chrome.

**Eddystone-TLM**

The final frame of the three is a telemetry frame, created to give the devices information about the beacon e.g. battery power and coordinates. This frame was created to provide
means for easier management and maintenance of large beacon fleets.

2.3 Indoor positioning

Indoor position is a lot harder than outdoor positioning, mainly due to the accuracy expected inside small areas such as rooms. Techniques used for outdoor positioning, e.g. GPS, will not work well because of attenuation caused by construction materials (roofs, walls etc.). There are also too many surfaces that will reflect the signals which causes multi-path propagation resulting in large errors \[2\]. In this report we implement and evaluate a prototype that utilizes beacons and RSSI as an indoor positioning technique.

Beacon technology has already been used for indoor positioning in many commercially successful applications. For example, Elle magazine has set up beacons in selected retail stores to give their readers suggestions when they are near those stores, through a third party application \[5\]. Another example is Macy’s, that provide location based coupons in their stores, by means of beacon technology \[6\]. These applications display information when a user is near a specific beacon. This has been tested and works well, even in crowded environments. The difficulties lies in determining exact distances using the technology. Saying a user is near a beacon is easy, determining precisely how near is significantly harder.

Some techniques used for this are Time of Arrival, Angle of Arrival, Trilateration and Received Signal Strength Indication (RSSI). Time of Arrival measures the time between a transmitter and receiver and uses the relationship between the carrier signal frequency and the speed of light to calculate the distance. To be able to do this the device must have a very accurate clock since an error in 1 \(\mu\)s results in an error of around 300 meters in distance. Clocks in normal devices like mobile phones are insufficient for this \[2\].

Angle of Arrival is a localization technique mostly used in Triangulation. The technique uses the angle between a reference direction and the propagation signal. This means multiple beacons are needed in order to estimate the distance with this method. In addition to several beacons being needed, the method is also affected by signal noise and therefore might give inaccurate results at close distances \[7\].

2.3.1 Received Signal Strength Indication and The Log-Distance Path Loss Model

The most popular method of estimating distances with beacons is the Received Signal Strength Indication (RSSI), since it is easy to estimate with the use of only a single beacon a smart phone. Previous research done for Wi-Fi and similar can also be applied here. Calculating the distance using only RSSI can be inaccurate, since a transmitted signal propagated through space will experience path losses. A well known model to predict path loss for a signal is the Log-distance path loss model. The model describes the total path loss as following:

\[
PL = P_{T_s} - P_{R_s} = PL_0 + 10 \cdot \gamma \cdot \log_{10} \left( \frac{d}{d_0} \right) + X_g
\]
2. Technical background

\( P_T \) and \( P_R \) is the transmitted and received power, \( PL_0 \) is the received signal at the distance \( d_0 \) after the path loss, \( d \) is the distance we are interested in calculating, \( d_0 \) is the reference distance, \( \gamma \) is a path loss exponent and \( X_g \) is a variable describing the attenuation. Depending on environment, different path loss exponents are suitable, as described in Wireless Communications Principles And Practice by Theodore S. Rappaport [9].

Into this equation we can add the variable txPower. The txPower value reflects the signal strength at \( d_0 \) which is commonly set to 1 meter. The value is set by the manufacturer in the beacon and transmitted along with the other information.

\[
PL_0 = P_T - \text{txPower}, \quad d_0 = 1 \quad \Rightarrow \\
PL = P_T - P_R = P_T - \text{txPower} + 10 \cdot \gamma \cdot \log_{10} \frac{d}{d_0} + X_g \Rightarrow \\
P_R = \text{txPower} - 10 \cdot \gamma \cdot \log_{10} \frac{d}{d_0} - X_g
\]

For the initial implementation of the prototype, the Android beacon library and their distance estimation algorithm was used. Instead of the Log-Distance Path Loss model, they did a power regression against a known table of distances and their corresponding RSSI values, which resulted in the following formula [2].

\[
d = (0.89976) \cdot \frac{P_R}{\text{txPower}}
\]

Since txPower is constant and included in the advertisement by the transmitting beacon, it is easy to see that the calculated distance is directly affected by the received power. If there are any fluctuations in the received power of the signal, this will show in the calculated distance. Fluctuations can be caused by a lot of things such as reflecting surfaces and blocking objects. Water is effective in causing signal attenuation. The human body is approximately 65% water, meaning signal loss is great just by putting your own body between the device and the transmitter. Only relying on unfiltered RSSI when measuring distances is therefore not very stable nor reliable.

2.3.2 Kalman filtering

A famous technique for estimating the state of a system corrupted by noise is Kalman filter. The technique uses measurements that has been gathered over time, which often contain errors. The Kalman filter can then estimate a more accurate result by looking at these measurements together, than by only looking at one of the measurements alone.

In our project Kalman filtering is applied to smooth out noisy measurements. This is done recursively by estimating likelihood between a model and measurements made based on previous values [10]. Due to the nature of RSSI and its noisy readings, we decided that this a was a necessary step in order to get a more smooth distance prediction. Refer to the implementation section for model and value specifications in section 4.1.1 on page [25].
2.3.3 Self-correcting beacons

In the article *Measuring a distance between Things with improved accuracy* the authors propose the use of additional beacons to improve the accuracy of the distance estimation [11]. To estimate the distance between the original beacon and the user device, the original beacon and the additional beacon are placed at fixed points with a fixed distance between them. Since the distance is known to always be constant between the beacons, the difference in RSSI is then considered as fluctuations and can then be measured. This knowledge can then be applied to the signals received in the user’s device to reduce the impact of noise, by continually calibrating txPower based on the noise in the environment, rather than having it constant as set by the manufacturer.

In their study, Hosik et. al, improved accuracy of the distance estimation by a significant amount. Their results yielded 41.6 percent decrease in average error rate. It is however worth noting that the decrease was diminished dramatically when the distances grew large. In the case of this report, this is of small importance since we were only interested in more accurate results when measuring small distances.
Chapter 3

Approach

In this chapter the approach of the thesis work and the different methods used during the thesis work are described. The methods are divided into the parts pre-study, prototyping and evaluation. They are described in the order they are performed in, but some parts were used throughout the whole project. As an example, the pre-study was performed in the beginning of the project but further research also occurred in lesser extent throughout the project. Evaluation was mainly performed in the end but also during the prototyping. This can be seen in the timeline in figure 3.1.

Figure 3.1: The figure shows a timeline of all the stages of our thesis work, illustrating how they at some points overlapped each other.

3.1 Pre-study

In this section we explain our ways of research and how we established our ideas. We started by gathering a scientific base and then performed a small set user interviews. This was done to see what work was already out there and to get an understanding on how users felt about the subject.
3. **Approach**

3.1.1 **Literature study**

We researched scientific papers, both published papers and other thesis works, to gather knowledge which served as a foundation on which our own implementation was built on.

Two main areas of scientific foundations were studied. The first being the field of Interaction Design. Important sources for this has been Donald Norman’s Design of Everyday Things, which gives a overview of the whole Interaction Design subject [12]. The interaction design process we have worked with is partly taken from Interaction Design - Beyond Human-Computer Design and Thoughtful Interaction Design [13], [14]. Throughout the course of this thesis project, we have performed usability testing, based on the theories presented in the Handbook of Usability Testing by Jeff Rubin and Dana Chisnell [15].

The second area was the technology which has been the foundation of our implementation. The main area here was the distance estimation. Here, we combined different approaches to distance estimation, both using Bluetooth and other technologies such as Wi-Fi, in order to form our own implementation which served our functionality in an optimal way. The resulting approaches we utilized can be found in the technical background in chapter 2 with implementation details in the results in section 4.1.

3.1.2 **Brainstorming**

The first step we took to formulate our ideas was to have a brainstorming session. We followed the steps described by Jonas Löwgren and Erik Stolterman in *Thoughtful Interaction Design*. Their process consists of collecting a group of people, discussing each others’ ideas without critique, and gathering the results [14]. The group consisted of the authors of this thesis.

To come up with ideas, a timeline was drawn on a whiteboard. The timeline showed users’ current interaction with a public transportation system. The ideas were presented in the group and written down on post-it notes which were added to the timeline. As described in *Thoughtful Interaction Design* nobody was allowed to criticize the others’ ideas. The goal was to come up with as many ideas as possible, to later sort out the good ones. Each idea was written down on a post-it note but some of the larger ideas were divided onto multiple post-it notes.

In order to clarify and structure the ideas, they were written more thoroughly on paper. Jonas and Erik proposes to divide the ideas through an affinity diagram. Since they were already divided by means of the timeline, it was decided an affinity diagram was unnecessary. The ideas which worked best for the schedule and project description were chosen and written in a document which was sent to our supervisors. Their feedback was taken into consideration and resulted in the final pick of the ideas which were to be implemented in our prototype. The ideas were drawn on a mind map and with the help of this, the prototype phase was divided into iterations, with each iteration covering one or two ideas. The mind map of ideas can be seen in figure 4.3 in section 4.2.

3.1.3 **Interviews**

The interview questions were developed with the help of the article *Interviewing Users* by Jakob Nielsen [16]. In the article he describes that user interviews may fail due to two
3.2 Prototyping method

Based on the resulting technological background of the pre-study, a prototype was then developed. In this section the method of implementing the prototype in iterations is described.

Prototyping in iterations

The process of working in iterations is taken from the agile software development methodology. Here, work is divided into small increments where the duration of an iteration is between one to four weeks. Each iteration contains planning, requirement analysis, design, programming and testing. This gives the project the ability to adapt to changing requirements from the stakeholders as well as to change course early should a vital discovery affecting the project be made [17].

Since our thesis work both included designing a prototype, by implementing it as an Android application, as well as usability testing of said prototype, we choose to work in an agile fashion. In Agile Development Iterations and UI Design, Jennifer Ferreira et. al. explains how design can be integrated into agile software development process. "While UI design may work from various guidelines and metaphors, the evolution of the design amid iteration and evaluation is an accepted aspect of User-Centered Design."[18]

With this in mind we divided our work, based on our brainstorming, into six iterations, which focused on implementing different types of functionality in our prototype. These iterations followed the steps of agile software development, however omitting the last part of code testing. Usability testing was done, but not at every iteration due to time limitations.

3.3 Evaluation methods

The evaluation part of this work was split into two parts. First, we focused on the technology to see what was possible to implement. Secondly, we focused on how users responded to the developed prototype and the interaction with the technology. The technological aspects were evaluated by us iteratively during the implementation stage, by looking at how well our implemented prototype performed at carrying out our suggested functionality. In order to assess the users’ reactions we hosted demo-sessions and performed usability testing, which is described in this section.


3.3.1 Demo-sessions

The demo-sessions were held for a small focus group, which consisted of supervisors, people from Softhouse who showed interest in our work and people from a local public transport company. In our demo-sessions we showcased the implemented functionality from the previous iterations and allowed for discussion and feedback, highlighting the lessons we learned so far.

At the final stage of the project we also held a larger presentation at Softhouse where we described our thesis work, explained what we had implemented and described our results. At this point we received feedback and questions which we discuss later in the report.

3.3.2 Usability testing

Usability testing is a part of the user-centered interaction design process and is an effective way of getting insights into how users react to a product. In our thesis we performed usability testing in two rounds. One in the middle of the project and one in the final stages. For both rounds, specific testing plans were created in which the tests were described in detail, see appendix B on page 73.

When doing our first round of testing we tested on a small group of people, seeing as our prototype application was still in its early stages of development. The aim was to test on eight people, but in the end six people was assessed to be enough. The group consisted fellow students who were also conducting their thesis work in related subjects.

The purpose of testing at this point was to get feedback on the implemented functionality in order to have time to improve it. To account for the state of the prototype, we divided the testing round into five separate tasks where the participants performed each task separately. After the first round was performed, the test results were used as foundation for improvements made to incorporate what the users felt. Both the results and the improvements can be found in section 4.3.1 on page 34.

The final round of usability testing was performed at the end of the project as an evaluation of the prototype. It was tested on 15 people, with focus on diversity between the participants. The participants were recruited from the Malmö office of Softhouse and fellow students on LTH. Specifics on the demography is discussed in section 5.3 on page 51. The testing was performed as a user story, to give the participants a feeling for how they would use application in real life.

At the end of the final usability tests the participants were given a survey with 13 statements, as seen in table 4.1. The statements were written both in a positive form and a negative to not influence them in a certain direction, as proposed by David Travis in Measuring satisfaction: Beyond the usability questionnaire [19]. The survey was conducted in order for us to gather some statistical data, by letting the participants express their opinion about the prototype in more structured way. For each statement, the participant was asked to write a number ranging from 1 to 5 where 1 meant that they completely disagreed and 5 that they completely agreed.

In addition to the survey, each participant was given a product reaction word list. Using a product reaction word list is an effective way of eliciting feedback from the user, both negative and positive. In normal usability questionnaires users tend to be more positive than negative and by using this method, it gives the user "permission" to be negative [19].
In our case, the list consisted of 80 words, both negative and positive. The users first had to choose all the words they thought were fitting for the prototype they just tested. When they were done they had to go through the words they had chosen and choose five of these which were the most significant ones. The complete list of the words can be found in appendix C at page 79.

After the final round was performed, we analyzed the results in a similar fashion to that of the first round with the results shown in section 4.3.3 on page 38. In addition to this, we also propose future improvements to be made in section 5.4 on page 59.
3. Approach
Chapter 4
Results

In this chapter the resulting prototype application is described. The first section covers the technology implemented in the prototype followed by the description of the different functionality. The final section covers the results of the usability tests.

4.1 Technology

This section describes the implemented technology used in the prototype application. The prototype was implemented as an Android application with the use of the Android SDK. In order to communicate with beacons, we used the Android Beacon Library which is a third party open SDK. It lets your Android device detect beacons and parse their transmitted advertisements and supports both Eddystone and iBeacon.

4.1.1 Distance estimation

Throughout the thesis work we tried different methods of distance estimation. In our final version of the prototype we used the Log-distance path loss model with the path loss exponent set to 2, as described previously in the technical background. In order to remove outliers from measured RSSI values, we used the median value from a window of the 20 most recent values. This value was derived from extensive testing, finding a balance between smoothness and latency, and is highly dependent on the advertisement frequency.

RSSI update frequency

We set up the beacons to advertise every 20 ms. The application was set up using the Android Beacon Library with a 120 ms listening period and a 0 ms resting period. This means the device will have a window of 120 ms to pick up an advertisement before calculating the distance to the advertising beacon and starting over again. If no advertisement was
picked up, the calculation would be omitted. This frequency was chosen to ensure maximum responsiveness as required for the functionality involving validation by proximity. A decrease in ratio between beacon advertisement and listening period resulted in loss of responsiveness, as the application would sometimes miss picking up an advertisement. The implementation however had severe impact on the battery life of both the beacons and the mobile phone, as detailed in the discussion in section 5.2.2 on page 48.

**Kalman Filtering**

We translated an implementation written in JavaScript into Java which was initially written for the same purpose [20]. We used a modified model to account for user movement, which followed a linear transition as such:

\[ x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \]

in which we assumed the beacon to be static, setting \( A = 1 \). \( u_t \) is a control command which in our case resulted in a walk detector as described below, and setting \( B = \frac{x_1 - x_2}{2} \) to account for movement from the user. The latter part is a primitive prediction as it only relies on relative change from the previous two calculations. The standard deviation for measurement noise of the filter is updated continuously to account for current noise, ranging between 45-70 in our experiments, while the process noise was set to around 10. Due to the fact that a user’s movement is close to random walk, one cannot predict when a user will start or stop moving. Because of this, our process noise was derived from various experiments, trying to find a balance between latency and smoothness.

**Walk detection**

Due to the fact that our Kalman filter needed a control command, we decided to add a walk detection algorithm in order to determine if the user was currently moving or not. Based on the findings in *Walk detection and step counting on unconstrained smart-phones*, we implemented a simple walk detection algorithm using normalized auto-correlation, as specified by Rai, Chintalapudi, Padmanabhan and Sen [21][22]. The detection works by checking the standard deviation of the accelerometer measurements, if it exceeds a certain threshold during a time frame of 800 ms. Should this be the case, the algorithm checks the auto-correlation for a larger time frame of 2000 ms. If it exceeds a second threshold, we assume the user is currently walking, until the standard deviation of the accelerometer measurements fall below a minimum threshold.

**Self-correcting beacon**

By connecting a user’s device to a second device via Bluetooth, we were able to reproduce the implementation of self-correcting beacons, as described in section 2.3.3 in order to mitigate RSSI fluctuations induced by the surrounding environment. By letting the second device listen to the advertisements of a beacon at fixed distance, we could send the values to the user’s device over Bluetooth from the secondary device. These values then served as an update to the reference value txPower in the prototype application, which meant that the reference value was continually calibrated and mitigated noise of the surrounding environment.
4.1.2 Distance estimation evaluation

In order to view the performance of the different implementations of distance estimation, we conducted some tests as can be seen in figure 4.1 which shows distance estimation while a user is moving and figure 4.2 which shows distance estimation on a fixed distance over time. The test which showed a user moving consisted of a user walking along a marked path up to the beacon and then turning to walk back again, whereas the static test consisted of a user simply holding the phone at a fixed distance from the beacon. When performing the moving test, the user would mark the actual distance when passing a marked point, at intervals of 1 and 2 meters for regular and fast walk respectively, which would in turn log the distance estimations by the different implementations. We can see that both implementations using the Kalman filtering performs substantially better than the other two, although the comparison against the raw data was presumed to be superior from the beginning. We can also see how our implementation adapts to change at a much quicker pace than that of the implementation offered by Android Beacon Library, which is something that was needed for the functionality regarding ticket validation to work.

Figure 4.1: This figure shows the difference in accuracy between four different distance estimation implementations, for both regular and fast walking speed.

4.2 Functionality

This section describes the functionality implemented in the prototype application. We begin by explaining each functionality separate and then how it was put together into one application.

The resulting final ideas of the brainstorming which was performed in the pre-study, as can be seen in figure 4.3, served as a foundation for the iterations in which the prototype was developed. During the iterative process some functionality was further developed and combined with other similar functionality, while some functionality was omitted entirely. The reason for this being the feedback received from usability testing and demo-sessions.
4. Results

**Figure 4.2:** This figure shows the difference in accuracy between four different distance estimation implementations, when looking at a static distance over time.

**Figure 4.3:** The figure shows the mind map of ideas from the brain storming.
4.2 Functionality

The implementation uses the Eddystone protocol since it was decided the use of Eddystone-URL was a useful way to promote an application. This by providing an easier way to find the application for users who do not have it installed.

On top of Android, several open source SDKs were investigated for the location based functionality. Both Google’s Nearby and Proximity API were tested, but in the end the open source Android Beacon Library was chosen since it was a better fit regarding the needs of the application.

4.2.1 Context awareness

Context awareness functionality refers to displaying information based on the location of the user. In our application this is implemented by providing the user with different information based on if the user is on a station or near a bus stop, or if the user is on a transport vehicle e.g. train or bus. The information is also dependent on which station or vehicle the user is on.

With the Android Beacon Library we perform background monitoring to send a silent notification to the user when they are nearby one or more beacons placed at a station. After a user has detected one or more beacons, we begin ranging in order determine the distances to all nearby beacons.

At the station the context awareness functionality is used to display the following information:

- List of nearby places, e.g. tracks, customer service and coffee shops.
- Timetable information, i.e. the departure/arrivals board you usually find at stations.

The list of nearby places was implemented as a list with different items which was displayed together with the distance to the object. The distance was constantly updated and was implemented as stated in section 4.1.1. The distances were displayed as texts being grouped together as "Less than 5 meters", "5 - 15 meters", "15 - 40 meters" and "More than 40 meters". The timetable of departing trains was also implemented as a list, which displayed the type of train, the direction and the time of departure.

On a vehicle the context awareness is used to display the following information:

- Previous and upcoming stops/stations.
- Status about the vehicle and the specific wagon the user is located on.
- Status of train e.g. delays.
- Time until the user arrives at their destination.

The previous and upcoming stops are implemented as a horizontal scroll view, where the next stop is animated by blinking, the previous stops are grayed out and the upcoming stops are in color. The time until arrival is estimated based on the ticket the user has bought, since it contains the information about destination. A print screen of how context awareness was implemented in our prototype can be seen in figure 4.4 and 4.5.
4. Results

Welcome to Lund’s Central Station
Expresso house
Less than 5 meters
Kundservice
Less than 5 meters
Track 1
Less than 5 meters

NEARBY TIMETABLE

12.01 - Pågatåg
Destination Ystad
Leaves from track 2
12.22 - Øresundståg
Destination Köpenhamn
Leaves from track 1
12.31 - Øresundståg
Destination Helsingborg
Leaves from track 2
13.02 - SJ
Destination Stockholm
Leaves from track 1

Figure 4.4: The figure shows the screen displayed when the user is located at a station. The left image show the nearby facilities and tracks, displaying the distance to each. The right image shows the departure board.

4.2.2 Smart notifications

A smart notification system was developed by utilizing the context information provided by the beacons. A user is notified with a persistent non-obtrusive notification when he or she enters a station. This means the notification will show up in the notification center but will not vibrate or make a sound. It will also persist there until the user leaves the station, and will not disappear when the user clicks it. The purpose of the notification is to make it easier for the user to interact with the application while performing other tasks on the phone.

The notifications were created to be smart, e.g. if the user has bought a ticket the notification displays an option to go straight to the ticket in the application. This is useful when the ticket needs to be displayed fast, e.g. when you need to validate your ticket on the train. When the ticket expires or if the user hasn’t bought a ticket yet, the same button takes the user to the purchase page. In addition to this, a status text which could be a welcome message or information about the ticket is also provided together with a final button that says "Nearby" and brings the user to the context aware screen of the application. The smart notifications were further developed to provide more feedback with the automatic validation, which is described later in section 4.2.4. The notifications can be seen in figure

30
**Figure 4.5:** The figure shows context awareness on a train. Relevant information is displayed together with a list of stops. The right image shows a user with a ticket and the left one a user without.

**Figure 4.6:** The figure shows two possible notifications displayed when a user arrives at a station. The right image shows a user with a ticket and the left one a user without.

### 4.2.3 Ticket purchase

The ticket purchase functionality was implemented with the help of beacons to simplify the usual payment process. In today’s public transportation applications, the providers charge differently depending on how far the person travels. To provide this information to
the application the user usually has to provide both current station and their destination. Since beacons can provide information on where the traveler is located, they would only have to choose their destination to purchase a ticket. This was implemented into the prototype application, together with the smart notification previously described, to simplify the purchase process.

The ticket can then be viewed in a separate screen, where information about the ticket is displayed. This information tells the user about the next departure where the ticket is valid. Beacons are also used to display the distance to the track where the train or bus departs from. The purchase screen and ticket screen can be seen in figure 4.7.

**Figure 4.7:** The image to the left displays the purchase screen, where a user is buying a ticket to Helsingborg C, whereas the image to the right shows the bought ticket.

### 4.2.4 Ticket validation

In this section the functionality around ticket validation and the associated gate application is described. Two ways of performing the validation were developed which we refer to as manual validation and automatic validation.

**Gate application** In order to perform the usability testing involving validation of tickets, we implemented a second Android application that acts as a mock gate, intended to run on a tablet or similar.
We choose to implement the connection using normal Bluetooth. This works by placing a beacon to mark a validation point, with the tablet device next to it. In order to enable the self-correcting beacon technology, we placed the beacon exactly 1 meter away from the tablet device. The user’s device discovers the beacon signal and calculates the distance to the beacon. When the distance is within a certain interval the user’s device is paired to the tablet device and relevant data is sent between them to validate the user’s ticket. The tablet application then displays a message telling the user if the validation was successful (and in practice opens the gate or similar). To estimate the distance previously described techniques were used, see [4.1.1 on page 25](#). This connection functionality could have been implemented in different ways. This is discussed in the discussion in section [5.2.3 on page 49](#).

The tablet device connects to the user’s device over a standard Bluetooth connection, receiving commands on what to show to the user from the prototype application. In a real world scenario, the gate would receive a request to open with ticket information attached, which would in turn be checked against a back-end solution and elicit a appropriate response towards the user. However, for our purposes, we felt that this was not in the scope to implement and therefore chose to simply send commands from the prototype application on what the gate application should show, as pictured in figure 4.8.

![Awaiting passenger… Waiting for connection… Please enter! Got message: 100 Sorry, you are unauthorized! Got message: 400](#)

**Figure 4.8:** This figure shows all stages of the gate application. In the image to the left the application is waiting for a user to approach. In the images to the right a message is being displayed telling the user the result of the validation.

**Automatic validation** Automatic validation refers to the functionality where validation of tickets is performed automatically when the user is within a distance from a validation point. When the user’s device detected a beacon representing a validation point, and was within a certain distance, the gate application and the user’s application connected...
to each other. After exchanging relevant information and determining if a user is authorized to enter, the gate application shows the response when the user is within 2 meters distance.

**Manual validation**  In addition to the automatic validation a manual method was also implemented, using the same techniques. When the user approaches the validation point, the gate application and the user’s application connect to each other, and a notification is sent to the user, which can be seen in figure [4.9] Clicking this notification begins the validation process, determining if a user may pass or not, and displays the response on the gate application. Deciding between automatic or manual validation was left as a setting in our prototype enabling us to test both functionality in our usability tests.

![Approaching gate at Track 2](image)

*Figure 4.9:* The notification displayed to open a gate when manual validation mode is active.

### 4.3 Usability testing

This section contains the results from both rounds of the usability testing. The improvements made after the first round as well as the results from the survey and product reaction word list, which were a part of the final round of testing, are also covered.

#### 4.3.1 First round of testing

This section contains the results from the first round of usability testing. The section is divided by each task the participants were asked to perform. The participants were studied when performing the task and then questioned about their thoughts of the functionality.

**Task 1: Eddystone-URL**

In the first task the participant was given an iPhone with iOS 9. The phone had Chrome installed with the widget Chrome Today activated. The task consisted of the participant navigating to the "Today"-page in notification center and looking for the Chrome widget. In the notification center, the Chrome widget displayed a notification triggered by the Eddystone-URL being transmitted from nearby beacons. The participant was then told to inspect the notification and then to share their thoughts about the functionality.

In this task almost all participants thought the functionality made it easier to download the application from the App Store, since they didn’t have to search for the application
4.3 Usability testing

themselves. The task also showed that the first time the participant tested the functionality they found it hard to find, even though they were given clear instructions. In conclusion, our participants found Eddystone-URL not to be a functionality they would use, since they didn’t know it existed previously.

Task 2: Nearby

In the second task the participants were given an Android phone with the prototype application installed. The participants were told they had entered a station and were to navigate to the nearby screen in the application.

The functionality in this task was received relatively positive. They thought the list of nearby objects was good when used to get an overview of what was available at a certain station, but they didn’t want to use it as a navigation tool. It was thought to be especially useful when they were at a station for the first time, as it can be hard to know exactly what the station has to offer. They also thought the distances displayed for each object were far to detailed and that millimeter accuracy was unnecessary. Some participants thought showing the direction would be a good complement to the distance, but most said they would probably just look around by themselves.

When asked if they preferred to know how close they were in terms of words like "very close", "far away" etc. they all answered they’d like to keep it in meters because they could relate to it, rather than the alternative. One person also said that it could prove difficult to find the objects of interest to her, in a list like this one, should there be a large number of objects.

Task 3: Notifications

In this task the participants were told to explore the notification center and test the different options provided (Nearby, Buy ticket, Show ticket).

The functionality was received differently between the participants. The ones who were used to the notification center thought it was useful, giving them easy access to buying a ticket and showing a bought ticket, while doing other things on their phone. However, many participants weren’t used to reading their notifications in the notification center and would rather prefer to get the notification on their lock screen instead, saying it wouldn’t bother them and it could be very useful. One participant thought differently and said it would bother him.

Some of the participants also felt it went a bit too fast when buying the ticket and they would like some kind of feedback before the transaction occurred. They wanted to be sure the ticket they were buying was the intended one.

Task 4: Automatic validation

For this task the participants were told they already had a subscription payment plan registered in the application so they didn’t need to worry about buying a ticket. They were then shown the gate application, as described in 4.2.4 on page 32. The task was then to use the phone and walk up to the tablet with the gate application, which would activate automatically, telling the participant to walk on through. The application did most of the
4. Results

work in the background in this task, and the participant did almost no interaction at all. The participant was told to explain their thoughts and reactions since this was hard to study due to the lack of interaction in this task.

The idea of automatic validation was received positively by all, but they also felt very unsure about the accuracy of the technology. Many participants also felt if it failed to work (which was the case for many of the participants on the first try) it would become something they would easily be bothered by and they would just stop using it. They also expressed concern about the lack of feedback. Even though the gate application showed feedback messages, they still wanted to have feedback on their phone, e.g. a vibration when the gates opened to indicate and reassure the user it was their phone which had opened the gate. Several of the participants also said they thought it might be easy to sneak by without paying.

The participants concluded they were generally uncertain to how well it would work and also felt it would take some time to get used to. One said "It feels weird not holding anything in my hand", as opposed to today's solution when validation is usually performed by ticket or RFID-card.

Task 5: Manual validation

This task was similar to the previous task, with the main difference being that the participant would have to click on a button in order to open the gate. At this time in the project the manual validation worked by navigating to a separate screen in the application and clicking on a button in that view. The task started by the participant starting the prototype application and navigating to the validation screen. When the participant was within a certain distance from the validation point and the gate application, a button would become visible and they would have to click it to validate their ticket and open the gate.

This functionality was well received by the participants as they were used to the interaction. Many felt the concept of this interaction was similar to the already existing solutions with RFID-cards used in today’s ticket systems. The exception being that they did not need to have a separate card to pay but could use their phones instead. Even though the participants felt more comfortable with this solution, and the implementation was significantly more stable, most of them agreed that if the previous method would have worked perfectly it would have been preferable.

They also noted that the technique in this task was more responsive and did what they intended at the moment they wanted it. However, they did not like that they had to open the application to do this, and afterwards they had a hard time locating the functionality.

4.3.2 Improvements

After the first round of testing had been performed, we discussed the results and made some changes in the prototype to improve our functionality according to the results of the testing. The first major change we did was to move the functionality from the ticket validation screen to a notification. The separate screen for validating a ticket and opening a gate was both hard to find and unnecessary. Moving the functionality to a notification made it more accessible for users and it also opened up the ability to use the functionality from a wearable device, e.g. a smart watch. In addition, we also added more user feedback
4.3 Usability testing

Here, as this was requested by most of the participants. When the user approaches the gate the user is notified with a sound and an updated notification. Clicking this notification opens the gate and the user never has to navigate to the application.

The nearby functionality was also changed slightly since most of the participants thought it was too detailed in regards to the distance reported by the items in proximity. Instead of showing the distances in centimeter precision, we divided them into these four groups: "Less than 5 meters", "5 - 15 meters", "15 - 40" and "More than 40 meters". Another minor improvement was adding a confirmation dialog after clicking the buy ticket button, since many thought the process was too fast. Seeing as the participants didn’t have to fill in where they were going from, it made them uncertain of where the ticket was valid. The added confirmation dialog can be seen in figure 4.10.

![Figure 4.10](image.png)

**Figure 4.10:** The figure shows the confirmation dialog which was added to provide more feedback to the user about the ticket purchase.

In the first testing round, the tasks performed by the participants were separate and focused on testing the different functionality implemented. This functionality was also shown on different screens in the prototype application, without connection between any two functionalities. This was also changed after the first round of testing, so that all functionality felt cohesive and connected. In the final tests, the tasks were part of one big story and represented how a public transportation application could be used from start to end by a user traveling from one station to another. As a result of this, the overall design of the application was also changed before the final tests. Example of the design changes can be seen in figure 4.11.
4. Results

Figure 4.11: The figure shows the design changes made between the first and last rounds of testing. This was made so that the participants would feel like they were using a real public transport application instead of just testing separate functionality.

4.3.3 Final round of testing

This section contains the result of the final round of testing which was performed as a user story where the participants were given the prototype and performed different actions matching a real world scenario. The tests were performed as single story, but the results are divided by functionality for easier readability.

At the station

The participants were given the prototype and were told they currently were at Lund C station. All participants quickly realized that the first screen, which showed the nearby functionality, depicted a list of things in close proximity to them. The majority expressed this would be more valuable to them when they were visiting a new station or when visiting a large station. During their regular travels it would be less valuable, but could prove to be useful tool in order to find new things at your regular station. Some of the participants also wanted to interact with the different items displayed, and suggested clicking the items would display them on a map together with their location. They also suggested the relative direction to the item should be visible. Another frequent suggestion was that clicking tracks would allow you to get more information about the departures and arrivals on that
4.3 Usability testing

track, or that clicking the coffee shop would display offers or similar.

One participant found it weird that the distance would sometimes change even though the participant was standing still, stating a disbelief in the distance estimation provided by the application. Most participants felt the rough distance was adequate, and further detail was unnecessary. However, one participant suggested the distance could be shown in minutes, in order to show if one could get there in time before their transport left the station.

The other part of the nearby screen, being a list of upcoming departures from the station, was also received favorably by the participants. They said that during their regular travels, they would benefit from having this screen with them in order to get up to date with the upcoming departures and possible delays and/or track changes. Most of them would usually either search for the same trip in the existing application or go look at the station departure and arrivals board to get this information. Generally, the participants favored the departure list over the nearby functionality, saying they would more often be in a situation where they would want to know up to date information regarding their trip, and would like this to be the primary screen.

Notification

When the participants entered the station area, a silent notification appeared in their notification center with a welcome text together with the two buttons "Nearby" and "Buy/Show ticket", as described in section 4.2.2 at page 30.

Our main feedback from this part of the test was that having shortcuts to certain functionality within the application was largely appreciated from all of our participants, especially the "Buy ticket", as this really reduced the effort required in order to purchase a ticket. In addition to this, many of the participants asked for the notification together with the buttons to be available on the lock screen for even easier access.

Another feedback we received from the participants was lack of indication of the notification updating its content, one example being when the participant bought a ticket. Nevertheless, the participants felt positive over the fact that the notification did update itself depending on the different context. They suggested even more information to be available regarding the ticket such as track number and departure time. Similarly, they also wanted different information to be shown when on the train.

Purchase and overview of ticket

For this part of the test, we told our participants they had previously entered payment information into the application, thus being able to simply pick a credit card to pay with. We asked our participants to proceed and buy a ticket through the notification, and received favorable feedback on this type of interaction, being a fast and easy way to navigate without having to search through the application. One participant did note however that it felt unnatural to use the notification when currently in the application. The participants also said it was convenient to have the departure location already filled in, only having to fill in the destination, making the process fast and easy. They did want the possibility however to alter the departure location manually, should they want to. Some of our participants were missing some options such as ticket type and delayed ticket activation.
Regarding the overview of the ticket, the initial response from the majority of our participants was that they did not know it was the ticket they were looking at, rather a summary of the trip they had just purchased. It was nevertheless received favourably, but with the wish of deeper functionality. Being able to see the earliest upcoming departure and from where was helpful, but some of the participants wanted to further interact with this information, going deeper and exploring more in regards to the departure, while others wanted even more departures visible. Some said they liked the functionality, but for regular commuters this information might be unnecessary and unwanted. In contrast, some said they regarded this information to be more valuable to them than that of the ticket validity time. One participant also felt the departure and destination information was a bit cluttered and hard to see within the image, and that our station abbreviations didn’t clarify either.

Validation of ticket

In order to perform the test as a single user story, we divided the participants in half where one group, group A, used the manual method of validation and the other one, group B, used the automatic method. In both groups the participant walked towards a validation point, consisting of a beacon and a tablet device running the gate application. The participant performed their assigned method of validation and the gate application told them the result of the validation. When successful the participant could pass and move on the next part of the test.

Group A The manual method of validation refers to the user opening a gate by clicking a notification, received when within a certain distance of the validation point. The method is described in detail in section 4.2.4 on page 32. It was tested by the participants in group A and was in general received favourably. Many said it would be useful since you would always have your phone with you and therefore not need a separate travel card for every public transportation company. A few participants said they would feel unusual in the beginning using their phone instead of a travel card, but would most likely get used to it.

One participant had concerns about what would happen if their phone ran out of battery, how they then would get passed the gate. Another participant also had a hard time understanding how to open the gate. Instead of clicking on the notification the participant tried to click on the tablet. The majority of the participants also never realized they had received a notification asking them to open the gate. One participant also said he didn’t realize the notification belonged to the application and ignored it completely.

In the end all participants were positive of moving the validation functionality from a travel card to their own phone. Some participants noted it would be bothersome if the technology didn’t work 100 percent of the time and would then rather use a travel card. Other participants wanted to take it even further by doing it completely automatically. They thought it was annoying take out their phone from the pocket, but also said it would be okay since they only needed to do it once per trip and it would probably become part of the routine.

Group B The automatic method of validation refers to the gate opening by itself when a user is within a certain distance. The method is described in detail in section 4.2.4 on page 32. It was tested by group B and received mixed reviews. The general opinion was
that the feature would be very useful and would facilitate the validation process, but the technology felt too untested and didn’t work 100 percent of the time.

For some participants the process was smooth and went without any problems. For some the gate did not open in time and the participant had to wait a few seconds before they could pass the gate, whereas for some the gate opened long in advance. When the gates opened at the exact time the participant reached the gate, the response was always positive. When the participants had to wait before entering, they often felt the technology was slow, even though some felt it was nice to have time to read the display properly. It also made them more sure the gate was opening up for them and not anyone else.

When the gates opened up too early the participants questioned the technology and started wondering if this would make it easier for people to pass the gates without paying. When asked again if they felt if it would bother them if people went by without paying, they said they probably would not care since it would not be their problem, but it would be a problem for the transportation company.

On board the train

When testing the train functionality, the participants were told they were entering a train which was departing in few seconds. A beacon had been placed behind the gate application, in a separate room or at separate location outdoors. When the application registered this beacon as the closest beacon the context aware screen automatically switched from the station page to the train page. In our scenario the train was headed towards Helsingborg C and a print screen of the setup can be viewed in figure 4.5 at page 31.

The functionality as a whole was received very positively by all our participants, even though, in a few cases, the application crashed. In the first test we performed, the participant bought a ticket to Copenhagen, but since this screen was hard-coded it still showed the train to Helsingborg. In future tests, we explicitly told the participants to buy a ticket to Helsingborg.

One participant raised an issue with the list containing the previous and upcoming stops, saying that if it was too long it could be hard to navigate through in order to find the station they were looking for. Another suggestion was to have more interaction with the list, e.g. clicking on a station would display information about connecting trains.

The section which displayed important messages received both positive and negative reviews from the participants. They generally felt it was a nice feature since it is usually time-consuming to find this kind of information, but also felt they did not want to be bothered by unnecessary messages. One participant also suggested the really important messages could be pushed as a notification to the user.

When asked about the transition from the station view to the train view, most participants thought it seemed logical. Showing information specifically for the train or station felt relevant since it would be what they wanted to see at that time. When later asked about the flow of the application, they were still unsure how the navigation worked. They were unsure how to go between the states and many felt they might want to be able to go back to the station page when on board a train. A few definitely felt they should be informed when the transition occurs, because if they are looking at something at the station view and it is suddenly replaced by the train view they would become confused. In our prototype the whole view was replaced, but one suggestion was to keep part of the view the same, so
4. Results

the user would know that the views all belonged to one page displaying the current context based on the location. Many also felt they should manually be able to go between the views, even though one participant commented that seeing the station page when not on a station would be a somewhat unnecessary. As an example the participant said: "You do not want to know that you are 3000 km away from a coffee shop at Lund C Station".

The majority of the participants also appreciated how easy it was to display their ticket, in case a controller would ask them to show their ticket. They also liked that the time remaining until the arrival at their destination was shown on the page. One said it would make it easier to tell if you had time to do other things on the train, such as starting your laptop. Many said they didn’t realize the functionality was connected to the destination they had provided when buying the ticket, but rather the final destination of the train. In addition, a few of the participants also requested to know more about things on the train, e.g. if you could buy food in any of the wagons. One last feature a few participants said was missing was to have something fun to do during the train ride, e.g. a game.

4.3.4 Survey

After each usability test in the final round, the participant had to fill out a survey consisting of 11-12 statements, depending on which group they belonged to. The results were normalized to show a positive mean, meaning the higher number the better the participant liked the feature. The positive mean can be seen in figure 4.12 and the statements can be seen in table 4.1.

![Figure 4.12](image)

**Figure 4.12:** The figure shows a bar chart of the results of the survey, where 5 was the maximum score. Looking at the chart, it is easy to see the participants were most positive about question 13 and least positive to the functionality detailed in question 2 and 9.
### 4.3 Usability testing

#### Table 4.1: The statements used in the survey, which was performed after each usability test.

<table>
<thead>
<tr>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I think it was nice having the app update its content based on my location (at station/transportation vehicle)</td>
</tr>
<tr>
<td>2. I didn’t like the nearby feature (seeing what was nearby me at the station)</td>
</tr>
<tr>
<td>3. It was meaningful for me to have access to the arrivals board directly in the app</td>
</tr>
<tr>
<td>4. It was difficult to buy a ticket</td>
</tr>
<tr>
<td>5. I liked getting a notification that I was at a station, to give me quick access to purchase, tickets and other contents.</td>
</tr>
<tr>
<td>6. (Group A) I liked the manual ticket validation, by opening the gate by clicking the notification.</td>
</tr>
<tr>
<td>7. (Group B) I liked the automatic ticket validation, when the gate opened automatically</td>
</tr>
<tr>
<td>8. (Group B) The automatic validation was very difficult to use.</td>
</tr>
<tr>
<td>9. The ticket validation felt secure and reliable.</td>
</tr>
<tr>
<td>10. I didn’t like the train page, and didn’t find it useful.</td>
</tr>
<tr>
<td>11. It was meaningful to me to see where I was and the upcoming stations.</td>
</tr>
<tr>
<td>12. The timer showing how much time was left until we arrived my station felt unnecessary.</td>
</tr>
<tr>
<td>13. I liked that I had easy access to my ticket on the train page.</td>
</tr>
</tbody>
</table>

#### 4.3.5 Product reaction word list

After the usability tests were performed, all the marked words from the product reaction word list were summarized into one word cloud, specifying the strongest reactions to the prototype. The word cloud can be seen in figure 4.13. The words sizes are based on how many participants chose the word, and if they chose it as one of their top five words.
Figure 4.13: The figure shows a the word cloud based on the reaction the participants had to the prototype application.
Chapter 5

Discussion

This chapter contains the discussion covering both the methodology used throughout the thesis work as well as the technology used in the making of the prototype. The results from both rounds of usability tests are also discussed with a final section covering the proposed future work.

5.1 Methodology

In this section we discuss the methodologies used during the thesis work together with some of the decisions we had to make, why we made them and how they impacted the results.

5.1.1 Brainstorming

This was the first stage of the thesis work and it was here that the project plan was formulated, going from a description to actual ideas of what we wanted to investigate. In this stage, we decided to go through the ideas by adding them to a timeline in the order of when a user would interact with them. The decision to line up all the ideas was good since it gave us a rough overview on how to implement our ideas in a real application. One of the problems with the way our brainstorming was performed was the number of participants, seeing as we were only two people in the project. We did not want to include other people at this stage of the project either, mainly because of time limitations and possible lack of in-depth knowledge. Instead we performed the brainstorming between ourselves, and then extended this by interviewing other people to get more feedback on the ideas, before we eventually took them to the next step.
5. Discussion

5.1.2 Scientific research

The research part of the thesis was one of the most important ones, since it gave us both inspiration on what we wanted to develop but also a foundation of what was possible and how to do it. Looking at our own ideas, we studied several news articles and blog posts online, of which a few are cited in the report, to see what was already out there. It also gave us a notion on how previous projects involving beacons had utilized the technology and how well the projects had been received. It also showed that even if beacon technology hasn’t had a breakthrough in Sweden, it has been successful in other countries, especially the United States.

When researching the technology, especially when working with the distance estimation, we used scientific resources e.g. conference papers and scientific journal articles to a larger extent. We did part of this research in the beginning to see if what we wanted to create was possible as well as to get a grasp on how to do it. We also performed some of this research during the implementation stage seeing as it required a lot of understanding in order to implement correctly. We found our initial implementation of the distance estimation to be lacking in accuracy to such an extent that we had to go back to research stage in this iteration of the implementation. Our additional research payed off and resulted in the use of a Kalman filter as well as walk detection and a self-correcting beacon. The walk detection was however decided to be left out of the final prototypes since it didn’t offer much improvement to our functionality.

5.1.3 Interviews

In the beginning of the project, we performed interviews to see if the ideas we had were reasonable and if there was any interest in them from users. The interviews were very short, since we didn’t want to spend too much time on user feedback during that point in time of our project.

Even though we didn’t interview a lot of people there was still diversity between the people being interviewed, but regardless of this, almost all the people we interviewed had the same responses. It was clear that people’s thoughts on public transportation applications were that they could be a lot smarter than they are today. When our proposed solution was described, people were generally in favor of it even though many aspects of it seemed unclear. The main concerns were that of the security of the application and if an automatic ticket validation could work in crowded places with many people passing at once.

In the end, we can see that the questions that arose in the interviews in the beginning are still the questions requiring further exploration. Since our thesis has been focused on the user experience, we never experimented with the techniques in crowded environments. Based on how we implemented our prototype it is easy to see where there might be problems and we will discuss these and possible solutions in later sections of the discussion.

5.1.4 Prototyping

By prototyping in iterations, we were able to change our prototype and the final functionality throughout the entire thesis. This proved valuable after receiving feedback from the initial usability test, which resulted in us re-purposing the prototype to provide a more
realistic user experience. In addition to this, we were also able to omit certain iterations which would have included functionality which was decided to be of less importance to the thesis, focusing on functionality that was considered vital as a result of our usability tests.

5.2 Technology

In this section we discuss the initial technology we decide to explore as well as the technology we ended up using. We start by looking at the techniques used for distance estimation and their results. Then the hardware is discussed by looking at how maintenance is performed and if the battery life was adequate for our application. Lastly we discuss how the communication between our prototype application and our validation application was performed and other possible solutions which could solve the disadvantages of the solution we chose.

5.2.1 Estimation of distance

The distance estimation is one of the key technologies in our application and is used for much of our functionality. Since we started implementing the prototype by using the Android Beacon Library, we already had access to the distance estimation implemented in their framework. However, it quickly became evident that this estimation was somewhat inaccurate and far too unresponsive for our purpose. Especially when looking at the automatic payment functionality it didn’t work as intended seeing as it took a few seconds before the distance value was updated, and even then the value was unfortunately not to our satisfaction. Seeing as the validation was to be triggered when the user was within a certain distance, the unresponsiveness resulted in the user having to wait for the validation to occur. In some cases it didn’t trigger the validation at all. Because of this, we decided to research our own way of doing distance estimation, finally ending up with implementing it by using the Log distance path loss model. In addition we also used self-correcting beacons and a Kalman filter to improve the results.

Testing was done and the result is shown in section ?? on page ???. More testing is however needed, as we only performed the tests to view the differences between the different implementations, rather than using it as a tool to improve the results. Trying different settings on the process noise for the Kalman filter while tweaking the size of the window frame containing the most recent values might yield even more accuracy, without a change in implementation. This was however decided to be out of scope for this thesis, as we had to begin our usability testing, and the results we got were in our opinion adequate enough.

Kalman filtering The Kalman filtering would be more accurate if it incorporated a model based on actual velocity and direction relative to the beacons, rather than just a mean value of the previous change that we ended up using. This was something we never explored, as this would require a set up with detailed beacon locations, preferably done via a back-end solution, together with extraction of relative direction from the user’s phone using the gyroscope and compass sensors.
Our model is also based on constant movement from the user, which isn’t the case in a real life scenario. In order to account for this, we implemented a walk detection that would serve to alter the model based on if the user was moving or not, i.e. remove the relative change parameter. However, we found the walk detection not to be accurate enough as well as a bit too unresponsive for our liking, so in the end we settled with a unaltered model and no walk detection at all.

As previously mentioned, we tried different values on the process noise for the Kalman filter, which serves to aid the filter in deciding how much it can trust the model. Due to the fact that our model isn’t entirely true in regards to reality, different levels of process noise are appropriate for for different scenarios. The best and most reasonable future approach would be to implement a better model, but an easier alternative to this would be to have an adaptive process noise depending on the scenario e.g. if a user was moving or not or rather the speed of the user.

Self-correcting beacons  As can be seen by the results in figure 4.1 and figure 4.2, the implementation of the Kalman filter together with self-correction yielded either no or less improvement to that of only Kalman filtering. We knew from the beginning that the improvements made possible the technique would diminish as the distances grew large, but we were surprised to see that it actually made our results worse. Further investigation should be done before this method is dismissed, but similar to that of improving the Kalman filter, this was decided to be outside the scope of this thesis.

5.2.2  Maintenance and battery consumption

One of the main advantages of using Bluetooth Smart is said to be the low battery consumption and maintenance of the devices. Even though this can be true in many cases, in ours it was not. Our beacons lasted for about eight weeks before the batteries ran out, while the standard usage states that a beacon device should be able to last for more than a year with a regular lithium cell (coin-shaped) battery.

Battery consumption is entirely dependent (unsurprisingly) on how the beacon is configured, and two parameters are of special significance. The first one is the transmitted power setting, which specifies at what power the beacon will transmit its advertisements. The second parameter is the frequency at which the beacon transmits its advertisements. In our application we transmitted at the highest possible power, a recommended setting by beacon manufacturers, in order to ensure accuracy. In addition to this, we set up the beacons to transmit at the highest possible frequency, in order to make the distance estimation as responsive as it could be. The combination of these settings resulted in our batteries draining faster than usual, both regarding beacons and the mobile phone. If this would be used in production this would require a lot of maintenance to replace the batteries.

The beacons we used, came with an option to configure them via an application or a web panel. In this administration panel the battery levels could be monitored, meaning we could see when the beacons were running low on battery and replace them accordingly. It is not an option however to have to change the batteries this frequently, nor is it acceptable for the user to have their mobile phone discharged whenever they entered an area with beacons in proximity. When we developed the prototype, we chose to have all beacons transmitting at the same high frequency, as this needed to match a setting in the prototype,
which simplified matters for us seeing as it was only a prototype. But it is only the ticket validation functionality which actually needed the high frequency we set, whereas the other functionality worked just fine with the recommended frequency settings which would allow longer lifetime for both beacons and mobile phones. It is possible to only have the beacons associated with ticket validation to transmit at that level of frequency while having the other beacons transmit at a lower while still acceptable rate. At the same time, the mobile phone application could switch its listening setting when in proximity to a ticket validation beacon, and then switch back when out of proximity in order to save battery life. There would still be a problem with the battery drainage of the ticket validation beacons, but this could be solved with an external power supply as there are beacons which can be bought with support for external power supply such as via USB or similar. This puts a limitation on the mobility of the beacons, but since only a few beacons would need to have this, it is still an acceptable compromise.

Another maintenance issue was the problem of adding or updating beacons to a venue. In our case, the beacons were added to the application in a configuration file. To update this file the whole application would need to be updated. One way to solve this would be to create a back-end solution where each beacon was linked with data in a database. This means the application would need to connect to the database before being able to present location based functionality to the user, but it does make it much easier to manage large fleets of beacons. It should be noted that Google recently released an API to deal with exactly this called The Proximity Beacon API.

5.2.3 Communication between validation point and user device

When implementing the automatic validation functionality we found that the communication between the user’s device and the validation point could be implemented in different ways.

We chose to go with Bluetooth pairing, previously described in section 4.2.4 at page 32. Another way to do it would be to use a server solution as communication. This solution differs from the implemented one, as it allows the user to connect to a server which in turn connects to the validation device or gate, rather than connecting directly to the validation device or gate. This is shown in figure 5.1. The different methods are discussed below.

Looking at security and usability the choice seems to be easy: using a web server has most advantages. Even though a separate web server would have to be implemented, this is a standard solution and there are already many good solutions out in the open source community which could have been used. Security concerning web servers are also well developed and users seem to trust them, since they are widely used in for example bank transactions. From a usability point they are also very good since they don’t require anything extra from the user. This would also most likely evade problematic integrations between mobile phones and possible validation devices. So why did we choose Bluetooth pairing instead?

The problem with using a web server is simple, the user needs to be connected to the internet. In our application, the amount of data is very small but it still needs to be performed fast and without any errors. This is a lot to demand from every user’s cellular
connection especially in the environments concerning public transportation. If the device is on a vehicle the velocity of the vehicle might affect the connection. If the station is underground, like many subway stations, this will also have an effect on the connection. The cellular networks might not be equally developed in all countries and a solution like this might not even be possible in some places.

Bluetooth pairing obviously eliminates the need for an internet connection on the user side. The validation of tickets can then be done by the gate application, where the internet connection can be controlled. However, Bluetooth imposes new problems such as higher battery consumption. Normal Bluetooth pairing is very battery consuming, meaning the Bluetooth communication should only be open for a limited time. Bluetooth pairing also need to be accepted by the user. This means the user would be prompted to accept Bluetooth discovery every time the user wants to to pay, which is not optimal from a usability point of view since the idea was to eliminate user interaction. In the interviews we performed in the beginning users also had doubts about the security of sending data over Bluetooth, even though this is not the case.

In the end we chose to stick with the Bluetooth pairing since it would work for our usability testing where we could just have the devices paired in advance making it possible to study the user’s reactions and opinions on the ticket validation. In a real life scenario, the local conditions and limitations would apply to the technique chosen.

We later also discovered Bluetooth Smart could have been used for gate communications in addition to proximity and this would have solved the problems that surfaced with our original implementation. From our point of view, it is a recommended variation of the implementation we did, as it would be done in similar way to that of normal Bluetooth, offering the same security on the connection[23][24]. The reason this is a recommended solution to that of the current one, is because this method does not need the explicit approval of the user in addition to not having all the gates listed in the paired devices catalog. For these obvious reasons, the current solution is not viable as it would remove the simplified interaction which was our goal. This was however discovered too late and we chose to not implement this since it wouldn’t affect our usability testing and would take too much time from the rest of the work, but would pose a highly viable option for further development.

**Figure 5.1:** The figure describes the two choices we considered to implement for the communication between the user’s device and the payment station.
5.3 Usability testing

The usability tests and their results are discussed in this section. It will also include a discussion of how the usability tests contributed to our work and what we could have done to improve them.

5.3.1 First round

This section discusses the results from the first round of usability tests and is divided by the tasks the participants performed.

The first round of testing was done in the middle of the project and was done to get feedback of the prototype while we still had time to change it. These usability tests were performed on six people. At this point, this was enough to get a good understanding of what people thought of the prototype, while not spending too much time on it. The participants were fellow students and there was no focus on finding diversity among the participants. The test was divided into tasks and each participant went through each task separately. This gave us good feedback on each of our functionality but never gave us any feedback on what the participants thought of the application in whole.

Task 1: Eddystone-URL

The results showed that Eddystone-URL simplified the process of downloading the application, but it also became evident that the participants weren’t used to this kind of interaction. None of the participants had heard of the Eddystone-URL functionality before and even though they thought it was useful, they were unsure if people would use it. This was not a problem with how we had implemented it, but rather with the implementation of Eddystone-URL and how it is presented in iOS. These nearby beacons are shown under the Chrome widget in the "Today"-page in the notification center, instead of the notification page. This is illustrated in figure 5.2. However, it is clear if the usage of Eddystone-URL becomes more popular and adopted by more developers, users will both like and use it.

Task 2: Nearby

As seen in the results, the nearby list would be hard to look through if there were many items. A solution to this could be to implement a filter functionality making it possible for a user to filter out what they specifically are interested in.

The results showed that the participants wouldn’t want to use the nearby list as a navigation tool. A different approach would be to implement the list in such a way so the user can click on an item and receive a map displaying the user’s position as well as the clicked item. Also, for people with vision disabilities, a compass using tactile feedback could be a good way of helping them to the relevant places nearby. When asked about this the participants were divided, as it wouldn’t be an optimal way of interaction, mainly because the user then would have to look at their phone while walking around, which isn’t very practical.
Figure 5.2: The figure shows the Chrome widget, displaying the URL sent out by the beacons.

Task 3: Notifications

The different results to the notification functionality might be because users have different experiences with the notification center due to the fact the implementation differs in Android and iOS.

The results also showed that most users would not be bothered by lock screen notifications, but one participant stated they would be. A solution to this, would be to have a setting to customize how the notification should be displayed. If the notification would be displayed on the lock screen, it could give users even easier access to buying and showing tickets at stations and on transportation vehicles, which could be very useful. This is something that as of Android Lollipop 5.0 comes as standard for notifications.

Task 4: Automatic validation

In this task we think the mixed responses by the participants can be attributed to the unevenness of distance estimation algorithm. We had problems with the technology in all of our tests resulting in the solution being less impressive than it could have been, which was our concern from the beginning. In some cases the technology and distance estimation worked very smoothly, validating the ticket at the precise moment the participant approached the tablet. However, in some cases the gate application validated the ticket
way ahead of the participant approaching the tablet and other cases the participant had to wait a few seconds before it performed the validation.

The results also showed that users would be unfamiliar with this type of interaction. People tend to like what they are used to whereas this type of interaction with automatic validation is uncommon.

**Task 5: Manual validation**

The biggest problem with the manual validation functionality is that you still have to interact with a device, similar to the systems used today, but with the difference being a phone is used instead of a ticket or a RFID-card. The participants had a hard time locating the validation screen, mainly due to the fact that the prototype application never intended to have a functional navigation.

### 5.3.2 Final round

The final usability tests were done in the end of the project and were performed on 15 people. The tests were performed as a user story and, as previously described, performed by dividing the participants into two groups. This gave us feedback on the both separate functionality but also on how a public transportation application based on location functionality worked as a whole.

We wanted to have a diverse set of people to get feedback that represented all types of people using public transportation. In the appendix [B.2.6](#) the test plan for the final testing round can be found, and the aim for the demography of the test participants can be seen. The goal was to have equal numbers of participants in both gender, travel frequency, method of payment and reason for traveling. We also wanted to have people from all age groups.

<table>
<thead>
<tr>
<th>Gender</th>
<th>Women</th>
<th>Men</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Women</td>
<td>40%</td>
<td></td>
</tr>
<tr>
<td>Men</td>
<td>60%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age</th>
<th>&lt; 25</th>
<th>25 - 39</th>
<th>&gt; 39</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 25</td>
<td>33%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 - 39</td>
<td>47%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; 39</td>
<td>20%</td>
<td></td>
<td>33%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Leisure</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leisure</td>
<td>47%</td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>53%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Travel card</th>
<th>Mobile app</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13%</td>
<td>87%</td>
</tr>
</tbody>
</table>

As can be seen in figure [5.3](#) and figure [5.4](#), we were not fully successful with this in any of our groups. We performed tests on 15 people so being completely equal was never possible in some of the groups. In the "reasons for travel" category we were closest to being equal and in the "method of payment" category we were the furthest away. It was
5. Discussion

Figure 5.4: The charts shows the type of payment and the reason for travel for the participants.

Hard finding people who usually use any kind of mobile application to pay, seeing as most our participants still used a travel card. In the area we live, you get a discount from using a travel card but not from the mobile application, which is why most people still don’t use the mobile application for payment.

Concerning gender, we were also pretty close, as with the frequency of travel. The age groups also looks good, being divided into "Under 25", "Between 26 and 40", and "Over 40", where the last group was hardest to represent. However, one participant from this group noted that people from the younger part of this group might have a different opinion from that of the older part. This was something we were aware of but we had trouble finding people over 40 who wanted to participate in the tests.

For the purpose of readability we divided the discussion into sections of functionality even though the tests was performed as a single user story.

At the station

Due to the prototype being just that, a prototype, we had some feedback that came from mistakes on our part. On the context aware screen which contained the timetable, the departure times in the timetable were static. This meant that depending on what the time was when we conducted the test, the participant could be looking on times in the past or times many hours in the future. This together with list being named "Timetable" rather than "Departures/Arrivals" lead to some confusion whether it was a static timetable or a list with up to date departures. Another mistake on our part, also being a naming mistake, was that the button in the notification said "Nearby" while the nearby functionality was only part of the screen. The button should have been named differently as it always took the participant to the screen with context aware functionality depending on their position, which in the case of the train page turned out to be very confusing as the nearby functionality was nowhere to be found.

Notification

The recurring feedback regarding having the smart notifications displayed on the lock screen rather than in the notification center was a good point. This would make the in-
interaction more natural seeing as iOS users are less used to interact with the notification center in comparison to Android users, relying on notifications on the lock screen in general. It should be noted that the Samsung S3 on which our prototype application was tested on, was running Android Jelly Bean 4.3, whereas Android Lollipop 5.0 introduced notifications on the lock screen as a default behavior. When we tried our prototype on another phone running Android 5.0, it did in fact act just as expected, displaying the notification on the lock screen, meaning that our current implementation would work as suggested by our participants. The notifications on Android Lollipop are also designed differently, where they are portrayed as a card containing both the text and the buttons. This would address the feedback from some of the participants concerning uncertainty if the buttons were connected to the notification or something else.

As previously mentioned, some participants became confused with the "Nearby"-button taking them to the context aware screen, which sometimes would show the timetable or when on board the train would simply show the train page. The participants also wondered when the notification would be visible, as they didn’t want it to be shown always. This was a fault on our end as the testing environment began at a station, which meant the notification already was there when the testing began. However, when we explained how the notification worked, only appearing when at a station or on board a transport vehicle, the participants agreed that this was a desirable solution.

A solution to feedback regarding the lack of indication that notification had updated itself could be by alerting the user with a sound or a visible toast when viewing the application. The participants also suggested updating the notifications when they were on a train, showing more information useful here. This should have been implemented in our prototype, but time limitations prohibited us from implementing this final part before the usability testing began.

**Purchase and overview of ticket**

From the results we can clearly see that this was a feature which was greatly appreciated by the majority of our participants, saying that shortcuts to certain functionality can be a real time saver. We received some feedback regarding the lack of options during the ticket purchase process such as being able to pick a ticket type e.g. student ticket, and being able to set when the ticket was to become active. This wasn’t seen as feedback to the interaction from our part, rather being shortcomings of an unfinished prototype.

We also want to highlight the fact that in a solution designed to be fast and easy to use, entering credit card information might not be the optimal way to go. However, with the rise of integrated payment methods e.g. Apple Pay, Samsung Pay, as well as Swedish Swish, the need for manually entering this information might be unnecessary in the near future.

Regarding the ticket overview, the results showed that it was difficult to understand that it was in fact a ticket and not a summary of what they had bought. This might have had something to do with the fact that we missed to write "Ticket" explicitly on the screen. But after opening the ticket screen from the notification, they understood it was intended to be the actual ticket.
5. Discussion

Validation of ticket

The validation of tickets was one of the biggest parts of the thesis since it used a lot of new technology and new methods of interaction. It was therefore very important to perform usability tests on the validation methods. We were also interested in knowing what type of validation method worked best and therefore decided to do testing of both the manual method of validating a ticket and the automatic method.

The results from both groups differed, but one thing that was clear from both groups was that more feedback was needed. When the gate application had validated the ticket and displayed the success message, almost all participants said they expected some kind of feedback in their phone as well. Some participants did note however, that this might actually not be necessary in reality since the gate opening might be all the feedback they needed.

Group A  In addition to the favourable reception of the functionality, the results showed some concern about what to do if the phone’s battery ran out, when the user has bought a ticket. This would be a problem which would be hard to deal with, like with every other system that moves over to being completely mobile. There are many possible solutions to this. You could have a station nearby where the user could log in and print their tickets. You could also have service personnel validating the travelers or provide power stations at the gates. Of course the best would always be to keep your phone charged.

In our implementation of the prototype, the participant opened the gate by clicking a notification. This way of interacting with an application is uncommon. Especially for people using iOS, notifications are only shortcuts to the application and usually do not provide any functionality of their own. This might also be why one participant didn’t understand how to validate the ticket. The participants who expressed this as a concern also said if they had done it once they would feel more comfortable with doing it again and they could see themselves using it in the future.

The results also showed the participants didn’t realize they had received a new notification to open the gate. This was partly due to how the test was performed but also because of how the notification was displayed. When it was time for this part of the test, the test leader had to change the settings in the application before handing the phone back to the participant. The notification only alerted the participant by sound and this sound often came before the participant had been handed the phone, so they never thought it was part of the test. In newer versions of Android a notification can also be displayed at the top of the screen. This would have been better for our usability tests because the participants would then have noticed the notification.

The results also showed that some participants thought it was bothersome to bring out their phone from their pocket. This could be solved with the proposed automatic solution, but could also be solved by using the manual method together with a smart watch or similar, removing the need to take out your phone.

Group B  For all our tests regarding the automatic validation functionality, though the technology worked differently every time, all our participants were allowed to pass the gates. Many still said if the technology would not have worked at all this would have been a big problem. Since you do not use anything to interact with the gate, you would not have
any way of influencing it and you will just be stuck on the wrong side.

With the automatic validation, the benefit is that the user does not have to interact with anything themselves, but at the same time it also appeared to be a downside. With normal travel cards and tickets you are in control of the process yourself, but with automatic validation you lose this. For some participants this was a problem, but they also said when you get used to the technology and it becomes part of the routine it would not be a problem anymore. Others did not have a problem with this.

In the end the validation process is something that is done by the transportation company to make sure everyone has paid. It does not provide anything for the users and if it can be performed without any effort from their side it would be much better. Automatic validation could be done together with a monthly subscription of the transportation service, eliminating the process of buying tickets every time. This would be an optimal solution since it would provide an even smoother experience of traveling while still making sure everyone pays for their trip.

On board the train

The most appreciated part of the train page, was the timetable showing all the coming stations. This is a functionality that everyone wants and are used to. In today’s trains there are often screens in every wagon showing this, but only during certain times. In some cases this is because advertisements are shown on the screens while other times the screens cannot fit all the text. To have this ready for the users when they open up the application gives them the ability to find out where they are, when they want to know it.

The results showed that a few participants were unsure if the station list showed the next or current stop with the blinking marker. In reality this would not have been a problem since someone on board a train would definitely know if they are currently at a station or moving between them. Our test environment did not reflect this and the participants felt it needed to be clarified. One suggestion was instead of having markers at every station, there would only be a line with an image of a train moving between the stations.

One user also found it hard to navigate in the station list because it was very long. This was probably due to it being represented as a horizontal scroll view and not a vertical, in order to have more space for other content on the screen.

The results showed that some participants thought the important messages were unnecessary. In the usability tests the chosen messages were maybe not very thought through. The messages were: "This wagon is reserved for passengers with pets" and "The toilet in wagon three is out of order". These messages only served a purpose for a minor amount of travelers, none of which were represented by our participants. If they were told the messages could be something different such as explaining why the train had stopped, e.g. "Unauthorized people on the train track, waiting for permission to continue", they said it would have been really useful.

One participant suggested using push notifications together with the important messages. For example, if you walk onto a train that has broken down and will not depart, you would probably want this information as quickly as possible. On the other hand you would also have to be careful not to push notifications or even displaying messages that are not important or informative. This could quickly become annoying and people would probably stop using the feature.
5. Discussion

In the prototype, the context aware screen showed either the station view or train view depending on where the user was located. The results showed that participants sometimes want to be able to decide which mode they are in. The user might want to be able to navigate to the train their friend is on to see when they arrive at the airport or to see if there are any trains departing before they get to the station. The functionality to search for buses or trains and plan your trip is still necessary. Displaying information based on your location should be seen as additional functionality that would help the traveler on their trip and not as a replacement.

As the usability tests showed, the ability to present the ticket directly from the train page was very appreciated. This functionality could also be extended to a wearable device, making it even easier to show the ticket without even needing to retrieve your phone from your bag or pocket.

The reason for why the participants were confused about how the application knew how much time was left before the train arrived might be due to the final destination and the participants’ destination were the same in the test. The participants therefore assumed the time displayed was how long until arrival at the final destination, instead of their own destination.

The usability tests also showed that there was a demand to know if there was any wagon on the train that served food. Since our imagined scenario always was commuter trains which usually do not provide food to be bought on board, we never even thought about this. If the scenario would be on trains that travel longer this would of course be a good functionality to have. The application could then not only tell the user where the food could be purchased but also display special offers.

The suggestion to include something more fun in the application was considered. This could be a fun way to pass the time while interacting with other users on the train. However with the aim to please everybody, a functionality like this would be hard to implement. Should someone want to play a game, they would probably download another application for this purpose.

Survey

The results from the survey was very even between all participants, showing they were positive to all the features, with a slight variation. The nearby functionality and the perceived security around the ticket validation was least favorable, while the train page functionality was received the best. The reasons for this has already been discussed in previous sections in this chapter.

Product reaction word list

In the word cloud, seen in figure 4.13 on page 44, it’s easy to see that most reactions are very positive with words like convenient, informative, time-saving, easy to use and helpful being the biggest. Among the smallest words we find ambiguous, insecure, bland, slow, frustrating, amateur and cluttered. These are words that only a few participants chose, but still represents the negative qualities we discovered in the usability tests.

Afterwards, when looking at the results, we realized we should have asked follow up questions on these words as to why the participants chose them, in order to get a deeper
understanding as well as to further pinpoint the reasons of the negative responses.

## 5.4 Future work

As the results has shown, there is a lot to be done with the technology proposed in the thesis. One thing which became evident when performing the usability tests is that providing information based on location was appreciated by almost everyone. It is a high rewarding and easy way of simplifying how travelers will interact with a public transportation application.

As such, the context aware screen has a lot to bring to the table. The nearby functionality, which was available when on a station, was received positively but we still feel it needs further improvement. It would be better if the application could provide more precise navigation in the form of a map, and also clicking on the items in the nearby list would provide more information about the item. This information could include next departures when clicking on a track or offers for coffee discount when clicking on a coffee shop. This would provide more functionality to the nearby list instead of just an overview of the station and would promote continued use of the nearby functionality. The timetable showing upcoming departures received a lot of praise as well, with the participants in the usability tests saying that this was a highly sought-after feature, while being fairly simple to implement. The same goes for the train page which provided essential information expected by the participants, but at one click away.

The future of the validation process is harder to predict as it requires much more precise distance estimation, in contrast to the previous functionality. In reality, if a company is looking to implement this, testing on a much larger scale needs to be performed, both for the manual and automatic validation. This is needed in order to establish how the technology performs in real public transport environments and crowded situations. But our recommendation is to pursue this, as our usability test suggested, users looked very favourably on the interaction. When looking at manual validation, other technologies offer similar interaction with a smart phone. One idea would be to combine the beacon technology with other technologies to prompt the user to validate a ticket when on the station, with the use of context awareness through beacons, but performing the validation using a different technology which has been more tested.

When looking at the future of automatic validation, it is clear that no other technology today offers the kind of interaction possible here. Users responded well to the idea of not having to interact themselves as it both simplifies the whole process while saving everyone time. The real problem with the automatic validation is that the technology isn’t ready today. The proposed implementation in this thesis works when doing usability tests but would not hold in a real environment. Much further work needs to be done to improve the distance estimation in order to make it more precise and more responsive, and only then should it be tested in real situations. Not surprisingly we are not the only ones realizing this as other companies are working full-time improving the technology in order to enable these kinds of solutions in the future.

Another thing to investigate further which has been outside the scope of this thesis, is the integrity of a user. By using Bluetooth Smart together with beacons, the application becomes aware of the users position. This information could be used by the public
transportation company to see where the most crowded areas are and at what times, but it could also be used for targeted advertisements as well as surveillance. Users are usually skeptical about giving this information to third parties, not to mention the fear of being monitored. Further investigation would be needed before the launch of this kind of functionality, specifying both advantages and disadvantages to ensure that the users always feel comfortable using the functionality.
In this thesis, we propose location based functionality for a more convenient and time-saving public transportation application. We implemented an Android prototype using both existing as well as new and improved methods which, by means of Bluetooth Smart and beacons, allowed the prototype to provide the user with context aware information. We also propose a method for ticket validation using the user’s smart phone, both manual and automatic interaction, in order to simplify the process for the user.

To evaluate the prototype, usability tests were performed to measure the reactions from the users. The tests showed a significant interest for an improved public transportation application and a positive reaction to Bluetooth Smart technology. The context awareness functionality at the station and on board the transport vehicle was received favourably, in addition to being easy to implement, concluding that this functionality is worth actualizing.

Ticket validation by means of manual interaction was also received favourably by our users, stating that it was easy to use and that they would prefer such a method over regular tickets. However, automatic interaction did not yield the same positive reaction, as the distance estimation was too uncertain, requiring further development before becoming a viable solution.
Bibliography


Appendices
Appendix A
Interviews

A.1 Questions

1. What type of phone do you own?

2. How do you usually pay for trips using public transportation? What do you think of the different payment methods available today?

3. We are looking into payment solutions using location based technology, which means that when the phone knows it is near a payment station the transaction will be performed automatically. This would mean that a user can leave your phone in your pocket or bag and just board the vehicle. Does this sound interesting? Do you see any problems with it?

A.2 Answers

Woman (20-30 years old)

I use an Android mobile. When I use public transport I have a card which is preloaded with money, to pay for my trip. I have used the mobile application to pay for a trip once and I would use it more if it wasn’t more expensive than the normal bus pass.

I think the applications could be smarter, but they are limited by the fact that there needs to be ways to control that a person has payed for the trip.

This sounds smart and would certainly work well in places where control of the tickets is more automated, for example, where gates are used. I do not think it is possible to use this with buses because it would be too difficult to control.
**Man (20-30 years old)**

I use an iPhone. When I go by public transport, I have a season ticket. I have never used the mobile application, since it is more expensive and does not have the season ticket.

The apps could certainly be smarter.

Yes, it sounds like a good solution. However, it seems difficult to control.

**Woman (30-40 years old)**

I use an iPhone 6. When I travel by train and bus, I pay with a travel card with travel funds. I’m happy with it.

It sounds like an interesting idea. I do not think I see so many problems except that I would be worried that the money would be deducted unnecessarily.

**Man (40-50 years old)**

I use an iPhone 6. Usually pay with the app then I think it works well.

I think your solution seems interesting but it does not feel so reliable.

**Woman (20-30 years old)**

I have an Android phone from Sony. I usually use a travel card with travel funds but right now I have a season ticket because I go more often now. I’ve used the app, it works well but it can be a hassle if the phone has been restored and I’ve just gone on a bus and have to fix the settings to pay the ticket. I also think the app has a few too many steps to actually get a ticket. It takes many button clicks before you’ve bought a ticket, then you must also activate the ticket.

Yes, it sounds interesting, especially for people who travel frequently. I feel it would be too much set up to download such an app with that payment solution, if you only use it once. I would also be concerned for safety reasons of using the app, for example, if you somehow were able to get out my payment information. I also wonder if it would deduct money by mistake if I happen to go near a payment position. It would feel safer if I could confirm the purchase before money was deducted, but then had a bit of the good functionality would be lost.

**Woman (20-30 years old)**

I have an iPhone 6s. I’m using a season ticket for the first time right now. I buy tickets with the phone sometime when I forgot the bus card at home or when I don’t have any money left in my traveling funds. I do not think the card system is particularly good. I don’t like that I have to buy traveling funds instead of just buying one ticket. Sometimes you just miss a few bucks but then have to reload your bus pass with a lot more money. When you
A.2 Answers

buy a ticket with the vending machines you still have to keep the receipt and show it on
the train even though the information is on the card. It feels like there are two systems in
one. The good thing with the cards is that when I go on the bus, I just have to hold it up
against the machine and then it’s done.

It sounds good. It would probably take a while for me to learn how to use it and get
used to it. I would also be concerned about security. How far away can I be, for example
before it withdrawals money from the app? What happens if you do not have the money
on the card and so on? But it would be good in rush hours because it would speed up the
process.
Appendix B
Usability test plan

In this appendix the test plans for the two rounds of usability testing is attached.

B.1 Test plan 1.0

This test plan describes the first round of testing.

B.1.1 Overall objective for the study

We will gather information about the users’ reactions to the prototype. The tests are performed at this time because the development of the prototype has come so far that it can be tested, but there is still enough time left to make changes in response to the user tests. The goals of the study is to see how users responds to technology that does not require user interaction. Will the users react positively to automatizing the validation process completely?

B.1.2 Research questions

This study will try to answer the following questions:

- What do users feel about using beacons to provide contextual information?
- What questions do the users have when their are provided with information based on their location?
- How easily and successful is the automatic validation? Are users hesitant when approaching this or does it feel natural to them?
- How do users feel about the time taken with automatic validation?
B. Usability test plan

• How forgiving are users about errors and delays in opening the gates (response times etc).

B.1.3 Measures

At the end of the session, we will have quantitative data

• Errors in opening and closing gates - we will know how often the gates open and not open when users approach.

We will also have qualitative data:

• A verbal protocol - the running commentary from the users when they perform the tasks.

• Interviews - answers to questions we ask during the tests.

B.1.4 Location and setup

The study will take place at the Technical Faculty at Lunds University. The sessions will be audio recorded by an iPhone 5. The participants will use an iPhone 6 for the first task and an Samsung Galaxy S3 for the rest of the tasks. Other materials needed is the Android tablet where the gate application will be run on as well as beacons.

B.1.5 Methodology

In this between-subject study, every participant will go through all the tasks in the task list. The session will take approximately 30 minutes. We’ll use 5 minutes of each session to explain the session to the participant. During the session the participants will have to answer to give comments on what they are doing and answer questions.

Pre-test arrangements

• Sign recording permissions.

• Answer to background questions

Introduction to the session (5 min)

• Explain moderator’s role

• Present the protocol for the rest of the session

• Discuss the importance of thinking out loud

Tasks (20 min)

• Participants will perform the 5 tasks

Post-test (5 minutes)

• Ask for feedback
### B.1.6 Participants

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Desired number of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of participants</td>
<td>8</td>
</tr>
<tr>
<td><strong>Travel frequency</strong></td>
<td></td>
</tr>
<tr>
<td>Irregular (A few times per month)</td>
<td>2</td>
</tr>
<tr>
<td>Moderate (A few times per week)</td>
<td>3</td>
</tr>
<tr>
<td>Regular (Every day)</td>
<td>3</td>
</tr>
<tr>
<td><strong>Types of travel</strong></td>
<td></td>
</tr>
<tr>
<td>Mostly business</td>
<td>4</td>
</tr>
<tr>
<td>Mostly leisure</td>
<td>4</td>
</tr>
<tr>
<td><strong>Payment experience</strong></td>
<td></td>
</tr>
<tr>
<td>Travelcard</td>
<td>4</td>
</tr>
<tr>
<td>Mobile app</td>
<td>4</td>
</tr>
<tr>
<td><strong>Age</strong></td>
<td></td>
</tr>
<tr>
<td>21-40</td>
<td>6</td>
</tr>
<tr>
<td>41-60</td>
<td>2</td>
</tr>
<tr>
<td><strong>Gender</strong></td>
<td></td>
</tr>
<tr>
<td>male</td>
<td>4</td>
</tr>
<tr>
<td>female</td>
<td>4</td>
</tr>
</tbody>
</table>

### B.1.7 Moderator role

We will be in the same space as the participant while conducting the session. We will introduce the session, as a few questions about the background, and then introduce the tasks. Because this study is somewhat exploratory, we will ask questions to clarify the participants’ behavior and expectations. We will also take some notes and do audio recordings of the participants’ behavior and comments.

### B.1.8 Task list

**Task 1 - Eddystone-URL**

The participant is given an iPhone with iOS 9 and Chrome with the widget Chrome Today activated. The task will be for the user to navigate to the notification center. In the notification center, the chrome widget will display a notification. Inspect what the participants do with the notification and interview them about their thoughts of this functionality.

**Task 2 - Nearby**

The participant is given an Android phone with the prototype application installed. The participant is told it has entered a station and will have to navigate to the nearby screen in the application. Inspect what the participants do on this screen and interview them about their thoughts of this functionality.
Task 3 - Notifications
The participant uses the same phone as in previous task. In this task the participant is asked to explore the notification center. The user will test the different options (Nearby, Buy ticket, Show ticket) and will then be asked questions about this functionality.

Task 4 - Automatic payment (validation)
The participant uses the same phone as in previous task and is told that they have a subscription payment plan that is already registered in the application. They are shown the application that represents the gate on the separate tablet device and is told how it works. The task is then to use the phone to open the gates. Since this task doesn’t use interaction it is important that the participant explains what their reactions are.

Task 5 - Alternative to task 4
This task is similar to the previous task, with the difference that the user will have to click on a button to open the gate. The button appears when the participant is within a certain distance from the gate application.

B.1.9 Deliverables
Using the recordings and our notes we will analyzed the data end answer the key questions. The results will be presented in our master thesis report.

B.2 Test plan 2.0
This is the test plan for the final rounds of testing. Some sections are the same as in previous test plan and will not be written out again.

B.2.1 Overall objective for the study
The usability tests described in this document is the final tests for our thesis work and they will serve as an evaluation of the implemented prototype. In the tests, we will gather information about the users’ reactions to the prototype. The goals of the study is to see how users responds to location based technology in a public transportation application.

B.2.2 Research questions
The same as previous test plan, see section B.1.2 at page 73.

B.2.3 Measures
In addition to the measures from previous test plan, see section B.1.2 at page 73 a survey and word reaction list will be gathered as quantitative data.
B.2.4 Location and setup

The study will take place at the Technical Faculty at Lunds University and at Softhouse Consulting’s offices in Malmö. The sessions will be audio recorded by an iPhone 6. The participants will use a Samsung Galaxy S3 to perform the tasks. Other materials needed consist of the Android tablet where the gate simulation application will be run on as well as the beacons.

B.2.5 Methodology

The same methodology is used as in previous test plan, see section B.1.5 at page 74.

B.2.6 Participants

This was the set-up of participants we aimed for, see discussion on usability testing in section 5.3 on page 51 to see how the actual set-up was and how it affected our results.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Desired number of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of participants</td>
<td>18</td>
</tr>
<tr>
<td>Travel frequency</td>
<td></td>
</tr>
<tr>
<td>Irregular (A few times per month)</td>
<td>6</td>
</tr>
<tr>
<td>Moderate (A few times per week)</td>
<td>6</td>
</tr>
<tr>
<td>Regular (Every day)</td>
<td>6</td>
</tr>
<tr>
<td>Types of travel</td>
<td></td>
</tr>
<tr>
<td>Mostly business</td>
<td>9</td>
</tr>
<tr>
<td>Mostly leisure</td>
<td>9</td>
</tr>
<tr>
<td>Payment experience</td>
<td></td>
</tr>
<tr>
<td>Travelcard</td>
<td>9</td>
</tr>
<tr>
<td>Mobile app</td>
<td>9</td>
</tr>
<tr>
<td>Age</td>
<td></td>
</tr>
<tr>
<td>Under 30</td>
<td>6</td>
</tr>
<tr>
<td>31 - 45</td>
<td>6</td>
</tr>
<tr>
<td>Over 46</td>
<td>6</td>
</tr>
<tr>
<td>Gender</td>
<td></td>
</tr>
<tr>
<td>male</td>
<td>9</td>
</tr>
<tr>
<td>female</td>
<td>9</td>
</tr>
</tbody>
</table>

B.2.7 Moderator role

The same as previous test plan, see section B.1.7 at page 75.

B.2.8 Task list

The participant is given an Android phone with the prototype application installed. The participant is told they have entered a station and will have to navigate to the nearby screen in the application. Inspect what the participants do on this screen and interview them about
their thoughts of this functionality. They should also inspect the notification center in the application. They will explore the home page of the application and look at nearby objects and timetables. They will also be asked to buy a ticket, either through the notification center or through the application’s menu.

The second part of the test consist of validating the ticket to open a gate, before the user can enter a train. Half of the users will do one of the tracks and the other half will do the other.

**Track A**  Group A will explore the functionality of manually open the gate by using the notification in the notification center. They will be asked about the experience and the feedback the notification gives (sound, text etc). Is this way of validation reliable?

**Track B**  Group B will not manually open the gate, but have the application being set on automatic validation. They will just have to walk by the validation point. They will be asked about the lack of feedback and how reliable it felt.

The last part will be for both groups. When the user has validated their ticket, they will move on to enter a train. The home-page of the application will update to being on board a train and they will have a chance to explore this page further. They will be asked about having a home screen that changes based on the context.

### B.2.9 Deliverables

Using the recordings and our notes we will analyzed the data end answer the key questions. The results will be presented in our master thesis report. The results from the survey will be summarized and also presented in the report. The product reaction word list will be presented as a word cloud.
Appendix C

Product reaction word list

This is the product reaction word list. See resulting word cloud in section 4.13 on page 44.

Step 1: Read over the following list of words. Considering the prototype you have just used, tick those words that best describe your experience with it. You can choose as many words as you wish.

- Amateur
- Ambiguous
- Annoying
- Authorative
- Awkward
- Bland
- Boring
- Busy
- Clean
- Cluttered
- Common
- Complex
- Confusing
- Contradictory
- Convenient
- Counter-intuitive
- Cutting edge
- Difficult
- Dull
- Dynamic
- Easy
- Easy to use
- Efficient
- Effortless
- Engaging
- Exciting
- Eye-catching
- Flexible
- Friendly
- Frustrating
- Fun
- Hard to use
- Helpful
- Illogical
- Impressive
- Improving
- Inconsistent
- Ineffective
- Informative
- Innovative
- Insecure
- Intelligent
- Intimidating
- Intrusive
- Messy
- Misleading
- Modern
- New
- Nice
- Not nice
- Old-fashioned
- Overwhelming
- Personal
- Playful
- Powerful
- Professional
- Responsive
- Satisfying
- Secure
- Sensible
- Serious
- Simple
- Slick
- Slow
- Sophisticated
- Straightforward
- Stressful
- Surprising
- Tedi
- Time-consuming
- Time-saving
- Ugly
- Unconventional
- Unfriendly
- Unpredictable
- Unresponsive
- Unusable
- Usable
- Useful
- Wow!

Step 2: Now look at the words you have ticked. Circle five of these words that you think are most descriptive of the prototype.