Institutionen för datavetenskap
Department of Computer and Information Science

Final thesis

Developing Android application prototype for internal social networks for companies

by

Kristina Ferm

LIU-IDA/LITH-EX-G--14/002--SE

2014-02-26
Final Thesis

Developing Android application prototype for internal social networks for companies

by

Kristina Ferm

LIU-IDA/LITH-EX-G--14/002--SE

2014-02-26

Supervisor: Arne Jönsson
Examiner: Arne Jönsson
ABSTRACT

Social media is becoming a bigger part of society and has now become a regular part of the world of professional companies. Incentive is a social intranet for companies, developed by Mindroute in Sweden. Incentive is available online but for the moment not as an application for Android.

This thesis focuses on developing a prototype Android application for Incentive, evaluating the usability of the result. Two designs were developed and the study showed that while both were good and user friendly a combination of the two should be considered.
ACKNOWLEDGEMENTS

I would like to thank Mindroute for letting me do my thesis work there and making me feel welcome.
I would also like to thank my supervisor Arne for letting me bug him with the million questions I thought of and for giving me great feedback during my thesis work.
Thanks to Baafen for server space, letting me store the json files that I use in my prototype application.
Oh, and Rickard – thanks for the lattes! I miss them already.
# TABLE OF CONTENTS

1. INTRODUCTION ..............................................................................................................1
   1.1 Background ..............................................................................................................1
   1.2 Purpose ...................................................................................................................1
   1.3 Limitations ...............................................................................................................1
   1.4 Objective ................................................................................................................1
   1.5 Method ....................................................................................................................1

2. BACKGROUND ..............................................................................................................2
   2.1 Incentive ..................................................................................................................2
   2.2 Stream design ........................................................................................................3
   2.3 Android related .......................................................................................................3
      2.3.1 Android ...........................................................................................................3
      2.3.2 Application .......................................................................................................3
      2.3.3 Parcelable and Serializable ...........................................................................3
         Serializable ............................................................................................................3
         Parcelable ...........................................................................................................4
         Which solution should one use? ..........................................................................4
   2.4 Evaluating design ....................................................................................................5
      2.4.1 Methods of evaluation ....................................................................................5
         User-based ............................................................................................................5
         Expert-based .........................................................................................................5
         Model-based .........................................................................................................5
      2.4.2 Prototyping .......................................................................................................6
         Low-fidelity prototype (Lo-Fi) ..............................................................................6
         High-fidelity prototype (Hi-Fi) ..............................................................................6

3. METHOD ..........................................................................................................................7
   3.1 Tools ........................................................................................................................7
      3.1.1 Eclipse IDE .....................................................................................................7
      3.1.2 Java ................................................................................................................7
      3.1.3 Android SDK ...................................................................................................7
      3.1.4 ADT Plugin .....................................................................................................7
   3.2 Design decisions ......................................................................................................8
   3.3 Search bar versus search button ............................................................................8
      3.3.1 Search bar ......................................................................................................8
      3.3.2 Search button ...............................................................................................8
      3.3.3 Evaluating the designs ..................................................................................8
         Why only test after wards? ................................................................................8
   3.4 Evaluation ................................................................................................................8
      The test .....................................................................................................................9

4. RESULTS ........................................................................................................................10
   4.1 Functionality ...........................................................................................................10
   4.2 Evaluation results ....................................................................................................12
      4.2.1 User feedback ...............................................................................................12
         Loading posts .......................................................................................................12
         Comments icon ....................................................................................................13
         Searching ............................................................................................................13
         Overall design ..................................................................................................14
      4.2.2 System Usability Scale results .......................................................................15

5. DISCUSSION AND CONCLUSIONS .........................................................................17
   5.1 Discussion ..............................................................................................................17
   5.2 Conclusions ..........................................................................................................18
5.2.1 Future work.................................................................................................................18
6. REFERENCES..................................................................................................................19
7. ILLUSTRATION REFERENCES......................................................................................20
1. INTRODUCTION

1.1 Background
I have for my final thesis been working at Mindroute for about 10 weeks. It is a relatively small company based in Malmö, Sweden. Mindroute provide a social intranet software, called Incentive, for companies. During my time there I had the task of developing a prototype for an Android application for Incentive, comparing two designs to see which was preferred by potential users.

1.2 Purpose
The goal of this thesis is to develop an Android application prototype with basic functionality for Incentive. There are no previous applications for this particular purpose and it would be beneficiary to have such a program for cellphones running Android. Rather than opening up a web browser on the mobile, having an application devoted to Incentive would simplify accessing it. My hopes are to successfully create a base for this, which then can be used as a foundation for further development of a fully working application for Incentive.

1.3 Limitations
There have not been any specific hard goals concerning the applications functionality. It was clear from the beginning though that I during this time frame would not have the time to develop a fully functioning application. A significant limitation is the fact that Mindroute were developing their latest version of Incentive as I was doing my thesis work. This has meant that I have not been able to implement any kind of authentication functionality for my application. Therefore I have not been able to work towards Mindroute’s own Incentive API and load for example the stream messages live, but working with statical uploaded text- and image files.

1.4 Objective
The main goal of my work was to create an overall, though not fully functioning, design for the application. As cellphones have limited battery time it is, apart from having a user friendly interface, important for an application to be optimized and efficient.

1.5 Method
My approach in the beginning of my thesis work was to follow a guide on how to develop a very basic Android application. The use of this guideline made the development of the application prototype for Incentive easier.

A lot of information on application development can be found on blogs. It may be viewed as an unreliable source, but when it comes to a subject such as this it is a good place for both inspiration and explanation on how to solve problems. Many technical blogs are written by software developers, making them knowledgeable and qualified in the area.

In the final stages of my thesis work I evaluated my application prototype by having potential users test the application, gathering feedback on its design and usability.
2. BACKGROUND

2.1 Incentive

Incentive is a social intranet for companies. A company has their own instance of Incentive where the employees have personal user profiles. Incentive allows users to easily share and take part of information about for example ongoing projects within the company. It has got the following features (Incentive, 2013):

- **Stream**
  Users can write messages which are displayed on a general posting feed called the activity stream. Anyone who is familiar with Twitter will find it very similar to a Twitter feed. Other users can be mentioned in a message post (@username) and will then be notified of the message. Like on many social networks the hash tag (#) functionality is available so a message can be tagged with relevant words, making it more prone to come up as a search result if it has anything to do with what was searched. For instance, a message concerning a project about apple pie could be tagged with “#pie”.
  Other users can read and interact with a message by liking it and leaving comments.

- **Wikis**
  Users can create wikis. A wiki is a page with text and articles, where a group of users can share information and discuss for example an ongoing project. They can in a collaborative manner both write and edit the contents of the wiki. (Wiki: What Is Wiki, 2002)

- **Blogs**
  Users can create blogs where they can publish posts which others can read and comment. Posts can be tagged with relevant words, making it possible for the post to show up if someone searches for something the post contains. Users can subscribe to blogs they want to keep up to date with, and as with messages in the activity stream users can be mentioned and notified in the blog posts.

- **Find people**
  It is possible to search for people within the company based on skill and/or expertise. These are set by the users in the user profile settings.

- **Search**
  Everything within an instance of Incentive is searchable, for example shared network files, message and blog posts and users. Suggestions based on what other users have searched for are provided.

- **Dashboards**
  Users have their own dashboards with widgets. A widget is a small application that displays some kind of information from another website or system. A user can personalize their own dashboards with for example widgets for blogs and external resources such as YouTube (YouTube, 2013).
2.2 Stream design

The application prototype being developed in this project is very similar to other traditional social media applications. The Stream of Incentive works essentially as Twitter's message feed and Facebook's time line.

In 2009 Janelle Estates, Amy Schade and Jakob Nielsen conducted a user research on what users want and need in terms of design from social networking applications. One outcome of this study, which has been of use to me, was that most users want a stream of messages to have the simplicity Twitter and Facebook has. The most recent message is displayed on top and then eventually leaves the main page. (Nielsen, J., 2009)

Seeing as how both Twitter and Facebook are widely used and appreciated applications, and the research showed that it is in fact what users want and not only because it was what was popular at the time.

2.3 Android related

2.3.1 Android

Android is an operating system for cell phones and tablets, developed by Google Inc. and the Open Handset Alliance (Open Handset Alliance, 2011). It is an open platform operating system, which means that anyone is allowed to develop Android applications. (Android 2013, Android Developers 2013)

2.3.2 Application

An application is a program run on electrical devices such as cell phones, computers, tablets etc. In this case I have been working on developing an application prototype for cell phones. Popular cell phone applications are for example Facebook, Twitter and Instagram. They are all socially interactive applications.

2.3.3 Parcelable and Serializable

An Android application is structured so that each screen in the actual application is represented by what is called an activity. When an application is launched the first screen you see is the main activity, that is however not necessarily the actual name of the java file. Other activities may be launched from the main activity. Sometimes you want to pass information between these activities. Take for example an application where you can post a message on a message board. On one screen you write the message in a text box, press "send" and then the posted message is shown on the message board in a new screen. You want to pass information, the written message, from the writing activity to the displaying activity. There are two ways to achieve this: Passing either a serializable or a parcelable object between the classes. Both are used for marshaling and unmarshaling objects in Java. That is, an object is transformed to a form that's good for moving it between for example parts of a computer program or a network.

Serializable

In Java there is a built in mechanism which is called object serialization. It works as such that an object is converted into a stream of bytes and then converted back into an object. Implementing serialization is very simple. Essentially the only thing that has to be done is to add "implements Serializable" in the Java classes declaration, there is no need to implement any methods. When a serializable object is passed between activities the marshaling operation in Java Virtual Machine uses a java Reflection API which does its best at identifying the objects data members. This results
in a lot of temporary objects.

Parcelable

When an object is passed by being parcelable it is split up into primitives, to then be rebuilt. To achieve this you have to implement certain methods and write code customized to the specific objects which are to be passed. This solution results in a lot less temporary objects since the code is written specifically to transform each data member of an object.

Which solution should one use?

While using serialization is less work than implementing the needed methods of parcelable, the latter is more efficient, and as I mentioned in my introduction efficiency is an important factor in Android programming. Studies comparing the difference in performance between the two have come to the same conclusion: parcelable is preferred as it is more efficient. A software designer named Philippe Breault runs a blog where posted the results of a conducted test (Breault, P. 2013). He tested passing an object multiple times on the three following devices running Android:

- LG Nexus 4 (Android 4.2.2)
- Samsung Nexus 10 (Android 4.2.2)
- HTC Desire Z (Android 2.3.3)

His test showed that parcelable is more than ten times faster than serializable. He also says that you rarely notice that serializable is a lot slower but judging by the test results I decided to take the time to implement the required methods for parcelable instead of just using serialization.

Three other tests which has had the same outcome can be found at the following websites:

- Bon-app-etit, 2013
- 3Pillar Global, 2013
- ¿Rocapal or Lapacor?, 2013
2.4 Evaluating design

There are different ways of evaluating designs. It can be done both during and after the development process. In short you can say there are three different methods to evaluating designs: User-, expert- and model-based (Dillon, A., 2001).

2.4.1 Methods of evaluation

User-based

In order to find out if a system works as intended people who fit the criterion of a potential system user get to try it out. This can be done in a more or less strict way. A group of test users may receive a set of specific tasks to complete using the system in question, but they may also be allowed to use the system more freely. The test users are observed and it is not unusual for them to recorded so it can be viewed and observed again in more detail. The users can receive papers to fill out both before and after the test.

The testing may be done in a laboratory environment but can also be as in hallway testing, where people “in the hallway” are chosen at random for trying out the system.

User based methods are regarded as giving the most real and reliable user feedback on a systems usability. However, putting together groups of users to test systems can be costly and resourceful to conduct.

Expert-based

In expert-based methods one or more specialists on human-computer interaction (HCI) are brought in to test and evaluate the system. Their expertise allows them to give a reliable guess and estimate on how users would react to the system. Two of the most common expert-based methods are heuristic evaluation and cognitive walkthrough.

Heuristic evaluation lets several specialists go through the system step by step and judge whether it fulfills the heuristics or not, the heuristics being like a list of rules. Examples of common rules for user interface designs are visibility of system status, user control and freedom and aesthetic and minimalistic design (Nielsen, J. 1995a). If a specific part of the system does not fulfill one of these heuristics the specialist takes note of this as a user problem. While The specialists are not allowed to speak to each other about their findings until after the system evaluation has been completed. (Nielsen, J. 1995b, 1995c)

Cognitive walkthrough uses a more specific and detailed procedure when evaluating how a potential user would find the system. The specialist conducting the evaluation goes through the system step by step, as in a heuristic evaluation, but for each step he or she tries to identify the goals of a potential user. For each step in the evaluation the specialist must answer eight questions, the first two being “First/Next atomic action user should take” and “How will user access description of action?” (Sauro, J. 2011b)

Using an expert-based method will give a faster and cheaper result than using a user-based method. It does not provide a result that is as user representative as if actual users are involved in the testing (Gulliksen & Göransson, 2002).

Model-based

Out of the three methods model-based is the one that is most uncommon. When using a model-based method a HCI expert creates a model which predicts and assumes how a user would behave when using a system. It can be for example how long time it takes for a user to complete a specific task or how hard it is to execute.
2.4.2 Prototyping

Prototyping is a method which can be used throughout an entire software design development process. There are different types of prototyping but they all have in common that unfinished versions of the system, prototypes, are being tested for receiving feedback from users. This way improvements can be made to the system when it's being developed rather than getting that user feedback after it has been finished. The prototypes can range from low- to high-fidelity. Two examples of versions of prototyping are where the prototype is thrown away after being used, and where the prototype is modified and updated after the feedback has been received and reviewed.

**Low-fidelity prototype (Lo-Fi)**

A Lo-Fi prototype is a very simple prototype, such as a paper prototype. They are used in the early stages of software development to test for example the user interface of a design. It can be as simple as a sketch on a paper, where a user is asked to try the design by “interacting” and clicking the prototype. It gives fast and cheap feedback. (Paper Prototyping: The Fast and Easy Way to Define and Refine User Interfaces, Carolyn Snyder)

**High-fidelity prototype (Hi-Fi)**

A Hi-Fi prototype is much more close to the finished product than a Lo-Fi prototype is. If the system being developed is an Android application a Hi-Fi prototype can be tested on an Android cellphone.
3. METHOD

3.1 Tools
This part is to give some basic knowledge of the tools which I have used when developing my application.

3.1.1 Eclipse IDE
Eclipse IDE for Java Developers is a computer software for building Java applications. It provides good functions such as giving a good overview of all files and possibilities for installing helpful plug-ins. (Eclipse, 2014)

3.1.2 Java
The programming language I have used for developing my Android prototype application is Java.

3.1.3 Android SDK
The Android Software Development Kit is downloaded and used together with Eclipse. It consists of several packages containing tools for debugging and testing for developing an Android application, the Android library for accessing Android like methods, to name a few select features.

3.1.4 ADT Plugin
Android Development Tools is a plug-in for Eclipse IDE which greatly simplifies Android application development. Using the ADT plug-in gives you access to, for example, Android specific features and Java- and XML highlighted text.
3.2 Design decisions

When deciding on how to design my application prototype I looked on mainly two things: current leading social networking applications, such as Twitter and Facebook, and the design of the online version of Incentive. One important feature is the search functionality. Search should be easy to access but at the same time not take up unnecessary screen space, leaving room for other main features. Cellphones vary in screen resolution and especially older models have lower resolution. Therefore I have decided to focus on design, mainly the search part.

3.3 Search bar versus search button

When it came to designing the search interface I ended up with two ideas that I felt both had pros and cons. I decided to implement both designs and later test which one users felt was the best.

3.3.1 Search bar

One solution was to have a constant search bar at the top of the Stream. On the one hand that should make the search functionality very easy to find and access. On the other hand a user may feel that it takes up too much space of the screen.

3.3.2 Search button

The other solution was to have a magnifying glass button in the action bar. Clicking it redirects the user to a new page where a word or phrase then can be entered and searched for. This version leaves more room for the main page and displaying the stream, however it may not be quite as obvious and fast to find as a search bar.

3.3.3 Evaluating the designs

Two designs were developed, one with a search bar and one with a search button. Those two were evaluated and compared in a study with potential users of the application. I took a user-based model approach as it gives the most real user feedback, and I do not have the resources to hire any human-computer interaction experts.

Why only test afterwards?

In the background chapter I mentioned different ways of evaluating and testing designs. Using prototyping is probably a fairly sure way of getting a finished product which is user friendly and that the majority of users would like using. The reason I have not used this method is because I feel that as I have been developing a foundation or prototype application for a software that already exists online, I did not have free rein to create the design any way I wanted. The design of Incentive is determined by the programmers at Mindroute. Using prototyping would be more suited for a new system that is under development and may be altered in more ways based on the results of the prototyping.

3.4 Evaluation

I first conducted a pilot study with two users to see if the test I had created would work. The pilot study went well so I decided not to make any changes to the test and include the result of it in my report. After the pilot study I let five users take my usability test. Jakob Nielsen has written two articles (2000, 2012) about how many users one should include in a usability study. With a few exceptions, the required number of users in a study is five. Having more will not give much more
information. Based on this I decided that my pilot study plus my main study was enough for reliable results to include in my report, resulting in having seven people evaluating my application prototype.

The seven users included in my study were of ages 22-32.

**The test**

My study was structured as follows.

I shortly explained that I created a prototype application for the social intranet Incentive, and that it is not a fully functioning application but has only very basic features. The study was mainly to evaluate which of my two designs people prefer and to get feedback on the general design.

The test person then got to use the application on a Sony Xperia Ray cellphone running Android 4.0.4 with a 3.3 inch screen, and received four tasks to execute. These tasks were done twice, once for each of my designs. I used the thinking aloud method and asked the participants to say what they were thinking while performing the tasks (Lewis, C. & Rieman, J., 1994) while I took note of what they said. I also noted if they managed to complete each individual task.

| Task 1: Load the application and scroll through the stream. |
| Task 2: Load more posts. |
| Task 3: Read the comments to a post which has comments, then return to the stream. |
| Task 4: Search for a word or a phrase. |

After this the test person received two papers to fill out. One for checking which design they preferred and one with a system usability scale (SUS) form. System usability scale is a way to measure a systems usability. It is a questionnaire with ten statements which are graded on a scale of 1 to 5, from strongly disagree to strongly agree. The statements are the following:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very awkward to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

To calculate the SUS score the odd numbered and the even numbered statements are treated differently. For odd numbered statements you take the user response and subtract one, for even numbered statements the user response is subtracted from five. You then add the result from all ten statements and multiply it by 2.5. A SUS score over 68 is considered average. (Sauro, J. 2011a)

System usability scales are probably most efficient for system that are closer to being fully functioning, but still gave me some useful feedback. It may still indicate the applications future usability level if it were to be further developed. Lastly I asked them if they had any other comments or feedback concerning the application and its design.
4. RESULTS

4.1 Functionality

I have made two slightly different versions of my Incentive application prototype. Other than the difference in the design for searching they have got the same functionality. Illustration 2 and 3 shows the stream with design A with a search bar, and design B with a search button.

The application prototype has got the following features:

- When the application is started it displays the messages on the stream. As I only was able to read from static saved text files it is the same messages being loaded each time the application is loaded.
- When scrolling to the bottom of the stream there is a “Load more” button. Clicking this will load more messages.
- Clicking a user name redirects to a new screen displaying information about a user. As with messages it is the same file being read whichever user name is clicked, and the name user information is displayed.
- Underneath each message text are three icons displaying how many likes the message
has, how many times it has been starred and how many comments it has got.

- Clicking the comments icon will, if the message has comments, open up a new screen displaying comments. It is the same comments being loaded each time.

- For Design A you are able to enter text into the text field which is at the top of the screen. Clicking search then opens a new screen stating what you have searched for, but as my prototype does not have any actual searching functionality you are not able to actually search for any content.

- For Design B clicking the magnifying glass in the upper right corner opens a new screen with a text field for entering search text.

- When a screen which is not the main stream screen is being displayed, clicking the Incentive in the upper left corner or clicking "STREAM" returns to the main screen.

Illustration 4 and 5 shows how the search design looks in design B.

Illustration 4: When having clicked the magnifying glass this new screen opens up, allowing the user to type something in the search box and then hitting "Search".

Illustration 5: Someone has attempted to search for "Ben & Jerry's Peanut Butter Cup", but without the search functionality implemented this is what the search query results in.
4.2 Evaluation results

My study showed that 72% preferred design B, with a search button instead of a search bar, see illustration 6. However after the test three people said that they thought the optimal solution would be to have a search button that, when clicked, opens a pop up search bar for entering text.

Illustration 6: Diagram showing what design the test people preferred.

4.2.1 User feedback

These are the comments received by the test users. In a few cases more than one person commented on the same things. Column number two represents how many people gave that particular feedback.

<table>
<thead>
<tr>
<th>COMMENT</th>
<th>NUMBER OF PEOPLE WHO SAID IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neat “Loading older posts” pop up dialogue.</td>
<td>1</td>
</tr>
<tr>
<td>Standard with a “Load more” button, have seen it in many other apps.</td>
<td>1</td>
</tr>
<tr>
<td>Logical.</td>
<td>1</td>
</tr>
<tr>
<td>Having the new posts load automatically when scrolling to the bottom of the stream would have been better than having to click the “Load more” button.</td>
<td>2</td>
</tr>
</tbody>
</table>
## Comments icon

<table>
<thead>
<tr>
<th>COMMENT</th>
<th>NUMBER OF PEOPLE WHO SAID IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>The icon was hard to hit.</td>
<td>2</td>
</tr>
<tr>
<td>The icon was perfectly sized</td>
<td>1</td>
</tr>
<tr>
<td>It is logical to have a speech balloon as an icon. It is clear that it indicates that there are comments and how many comments there are.</td>
<td>1</td>
</tr>
<tr>
<td>It was not completely obvious that the icon was a speech balloon which was meant to be hit in order to read the comments.</td>
<td>1</td>
</tr>
</tbody>
</table>

## Searching

<table>
<thead>
<tr>
<th>COMMENT</th>
<th>NUMBER OF PEOPLE WHO SAID IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>When having entered text in the search box it would have been preferred if pressing enter submits the search query instead of creating a new line.</td>
<td>2</td>
</tr>
<tr>
<td>Both the search box and the search button were logically placed, easy to find and easy to use.</td>
<td>1</td>
</tr>
<tr>
<td>Having a search button with presents a pop up search box when hit would have been preferred to the static search box on top or the search button.</td>
<td>3</td>
</tr>
<tr>
<td>Regarding design A: It was annoying that the search result opens up an entirely new screen and that when going back from this screen you end up at the top of the stream, instead on where you were before.</td>
<td>1</td>
</tr>
<tr>
<td>Regarding design B: When the magnifying glass (search button) is clicked the keyboard should be brought up instantly rather than having to first click inside the search box. The search results should also be shown beneath the box instead of opening up a new screen.</td>
<td>1</td>
</tr>
</tbody>
</table>
### Overall design

<table>
<thead>
<tr>
<th>COMMENT</th>
<th>NUMBER OF PEOPLE WHO SAID IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>The time stamps are way too detailed.</td>
<td>1</td>
</tr>
<tr>
<td>“I don’t know if I want to know micro seconds..”</td>
<td></td>
</tr>
<tr>
<td>Having the format “date [space] time” would have been sufficient.</td>
<td></td>
</tr>
<tr>
<td>The icon was perfectly sized</td>
<td>1</td>
</tr>
<tr>
<td>It was extremely annoying that you have to press the <em>icon</em> when navigating back from a screen in the application and that it is not enough to press the name</td>
<td>1</td>
</tr>
<tr>
<td>Design B was preferred because it allowed more space for content. The gray bar at the top of the screen was annoying because it takes up valuable screen space</td>
<td>1</td>
</tr>
<tr>
<td>The avatar to the left of each message on the stream made all text feel small and tall, it feels unnecessary on such small screens. There could be text under the avatar as well.</td>
<td>1</td>
</tr>
</tbody>
</table>
### 4.2.2 System Usability Scale results

The following table displays how the test people answered the System Usability Scale questionnaire.

*Table 1: How the seven test people answered the System Usability Scale questionnaire.*

<table>
<thead>
<tr>
<th>Nr</th>
<th>Statement</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I think that I would like to use this system frequently.</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>I found the system unnecessarily complex.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>I thought the system was easy to use.</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>I think that I would need the support of a technical person to be able to use this system.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>I found the various functions in this system well integrated.</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>I thought there was too much inconsistency in this system.</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>I would imagine that most people would learn to use this system very quickly.</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>I found the system very awkward to use.</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>I felt very confident using the system.</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>I needed to learn a lot of things before I could get going with this system.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The average SUS score for my application prototype is 87.86.

Illustration 7 and table 2 illustrates and demonstrates the individual SUS score for each test person.

**System Usability Scale**

**Results**

<table>
<thead>
<tr>
<th>Test people</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>70</td>
</tr>
<tr>
<td>P2</td>
<td>75</td>
</tr>
<tr>
<td>P3</td>
<td>80</td>
</tr>
<tr>
<td>P4</td>
<td>85</td>
</tr>
<tr>
<td>P5</td>
<td>90</td>
</tr>
<tr>
<td>P6</td>
<td>90</td>
</tr>
<tr>
<td>P7</td>
<td>95</td>
</tr>
</tbody>
</table>

**Illustration 7: Chart over the SUS form score.**

**Table 2: Table over the SUS form score**

<table>
<thead>
<tr>
<th>SUS Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.5</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>92.5</td>
</tr>
<tr>
<td>95</td>
</tr>
</tbody>
</table>
5. DISCUSSION AND CONCLUSIONS

5.1 Discussion

Based on the usability study it appears that the application prototype is something a possible user finds logical and usable. Obviously with the very limited functionality it does have it is hard to make any real assessments of how an actual fully working application would behave – but it gives an indication. The test showed that both design A and design B were logical and for the most part easy to understand, telling me that what I have found about social media application design and my own instincts can be trusted.

The biggest limitation during this work has been the fact that Mindroute is currently developing their latest version of Incentive, causing me to not be able to implement an authentication service in the prototype. If that had been a possibility the focus on this thesis would probably have been on implementing such a feature, for gaining access to reading the stream and other files live. Doing so would have been fun, instead of reading from the same statical files as the case is now. I also believe that if I were to have had a lot more time than these ten weeks I would have been able to produce something with a bit more functionality, even without the authentication service implemented.

Regarding the feedback on the design I feel that I tend to agree with a lot of them, and if the prototype in the future were to be further developed they are good notes on improvements. The two mains things which I agree on would improve the application concerns the search design and the implementation of loading older posts.

As three test persons pointed out having a search button in the form of the current magnifying glass should open a pop up search box. This would free up screen space for when the user has no intention of using the search function, and in my opinion it looks neater with a magnifying class instead of a static search box on top. To further improve this the comment about not opening an entire new screen for the results should be considered.

Having the older posts load automatically instead of having to click the “Load more” button was suggested by two test persons, which would be an improvement. For one it might make the application flow better, and many popular social network applications work like that so users may expect it and find having to click a button annoyingly slow.

User comments on smaller details such as the feedback on the time stamps are also valuable. Even if it is a minor feature it contributes to the application as a whole. As the screen size of cellphones are so small, simplicity and a clean design is important. If the application prototype is developed further the time stamps would definitely be changed to a more simple and less detailed format.

One thing that could have been made clearer before starting the usability tests with each person is the fact on how simple the application is in its functionality - perhaps specifically mentioning each thing which would not work as a real application would. However, I felt that it was better not to say too much about the application before hand to get a more spontaneous and real feedback from the test users. The fact that it is the same messages being loaded each time, the same comments being loaded to all messages that has one or more comments and that the search functionality is not actually implemented seemed to have confused some of the participants a little.
5.2 Conclusions

The goal for my final thesis has been to develop an Android application prototype for the social intranet Incentive and then evaluate the design and usability. Judging by the results and the feedback from the usability study the prototype created meets its goals. The System Usability Scale forms shows that the design is not overly complicated or hard to understand, but can probably be used as a foundation for an actual application.

The main part of the test session where the test people got to try the application on a cellphone went smoothly and all of the participants managed to complete the tasks. Asking them to think aloud while executing the tasks was a good decision. This is how the most useful feedback on things which can be improved was received, and I was surprised by how good people were at talking and pointing things out while using the prototype.

5.2.1 Future work

If the application prototype created is to be developed further there are existing features which can be improved and additional functionality that can be added. As discussed in the previous chapter the following features can be improved.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search design</td>
<td>Having a combination of design A and B, with a magnifying glass button which opens up a pop up search box when pressed.</td>
</tr>
<tr>
<td>Search design</td>
<td>Having the search results showing up beneath the search box, without opening up an entirely new screen.</td>
</tr>
<tr>
<td>Loading older posts</td>
<td>Automatically loading older posts when scrolling to the bottom of the stream.</td>
</tr>
<tr>
<td>Time stamps</td>
<td>Less detailed time stamps, such as “14-01-22, 10:10”</td>
</tr>
</tbody>
</table>

As the prototype is just that, a prototype, future work could and should mean adding more functionality. When Mindroute are finished with the latest version of Incentive authentication should be one of the first things to be implemented and added to the prototype. This would allow for displaying the actual stream with real comments, avatars and information. Searching and accessing other features in Incentive, such as blogs and wiki pages, could be a next step.

In short there are a lot of things which can be added to the prototype in order to make it a real working application.
6. REFERENCES

- Sauro, J. 2011a. Measuring Usability With The System Usability Scale (SUS). 2 February
7. ILLUSTRATION REFERENCES

På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© [Kristina Ferm]