Application development for smartwatches

AN INVESTIGATION IN SUITABLE SMARTWATCH APPLICATIONS

DAN ISACSON
Application development for smartwatches
An investigation in suitable smartwatch applications

Dan Isacson
disacson@kth.se
June 2015

Swedish title:
Applikationsutveckling för smarta klockor
En undersökning inom lämpliga applikationer för smarta klockor

Master's Thesis
KTH Royal Institute of Technology
Masters Programme, Computer Science
School of Computer Science and Communication (CSC)

Supervisor at The Mobile Life: Kenneth Andersson
Supervisor at KTH: Arvind Kumar
Examiner: Olle Bälter
Abstract

The smartwatch has been predicted to be the next big thing in the ecosystem of wearable and mobile devices. The success of the smartphone has a lot to do with the support from third party developers and their applications. This support will most likely be of utmost importance if the smartwatch is going to be successful. But what are suitable applications for this type of device? What are people’s expectations and opinions on smartwatches? This thesis work delved into these matters with focus on development for the Apple Watch. This was done through an online survey that reached more than 1400 people, through a field study for one person that used the smartwatch as an air-travel tool and finally through a usability test conducted on five people that tried out several applications, most of which were developed for this project. Four different applications were implemented: a timetable application, a museum audio-guide, a game and an airline application.

The results indicate that people generally expected the smartwatch to act as an extension for their smartphones and notifications seemed to be an area in which the smartwatch excelled. The results also show that there are many applicable areas the smartwatch can be used for, but that interaction with these should be kept quick and simple. This is extra important given the limited screen-size and special considerations should be taken on what content to display.
Referat


Resultaten visar att folk verkar förvänta sig att en smart klocka ska fungera som en förlängning av deras smarttelefon och det verkar som att notifikationer är ett väldigt bra användningsområde för formatet. Resultaten visar också att det verkar finnas många olika användningsområden för smarta klockor men att interaktioner med dessa bör vara korta och lätthanterliga. Detta är extra viktigt givet den begränsade skärmstorleken och man bör noga överväga hur och vad man presenterar för användaren.
# Contents

1 Introduction  
1.1 Background ....................................................... 1  
1.2 Thesis objective .................................................. 1  
  1.2.1 Aim ............................................................ 1  
  1.2.2 Delimitations ............................................... 2  
1.3 Approach ......................................................... 2  
1.4 Thesis structure ................................................ 2  

2 Background  
2.1 History of smartwatches ......................................... 3  
2.2 Previous research ................................................ 4  
  2.2.1 Smartwatch modalities .................................... 4  
  2.2.2 Smartwatch fields of use .................................. 6  
2.3 Current platforms ............................................... 6  
  2.3.1 Pebble ........................................................ 7  
  2.3.2 Android Wear ............................................... 8  
  2.3.3 Apple Watch ................................................ 9  
2.4 Development for the Apple Watch ......................... 12  
  2.4.1 Xcode ....................................................... 12  
  2.4.2 Obj-C and Swift ........................................... 12  
  2.4.3 WatchKit ................................................... 13  

3 Method  
3.1 User survey ..................................................... 15  
  3.1.1 Objective ................................................... 15  
  3.1.2 Approach .................................................... 15  
  3.1.3 Questions ................................................... 15  
3.2 Development ..................................................... 16  
  3.2.1 Objective ................................................... 16  
  3.2.2 Applications ............................................... 16  
3.3 Usability study .................................................. 16  
  3.3.1 Objective ................................................... 16  
  3.3.2 DECIDE framework .................................... 16
C Resource links 69
   C.1 Google analysis of the full survey ................. 69
   C.2 SL Watch ........................................... 69
   C.3 Museum application .............................. 69
   C.4 Memory game ....................................... 69
   C.5 Flight application ............................... 69

Bibliography 71
Chapter 1

Introduction

1.1 Background

The last few years have been predicted to be the year of the smartwatch and wearables[1], but the trend has yet to be set and people are yet to include wearables in their daily lives[2]. This is most likely to change with Apple’s entrance into the field with their Apple Watch[3]. The interest for this device is tremendous and many companies are now interested in the possibilities of smartwatches[4]. The Mobile Life is one such company and is where this thesis work has been carried out. The Mobile Life is a company that focuses on mobile solutions and aims to be in the forefront of this evolving market.

1.2 Thesis objective

1.2.1 Aim

There were two main goals for this thesis project. One of the goals was to delve into application development for smartwatches, with focus primarily on the Apple Watch. The second aim was to find out what types of applications are suitable for the smartwatch format. To answer this, the following two questions were addressed in this work.

- What properties describe a usable smartwatch application?

- What is expected of a smartwatch by the users?

Indirectly this brought up questions about how to interact with smartwatches given their limited screen size and interactability. Because of this, various modalities were discussed and tested. Most of the conclusions were aimed to be made for smartwatches in general, but there was a slight bias towards the Apple Watch and its technology since it was the main platform used for evaluation.
1.2.2 Delimitations

This work did not intend to figure out new modalities but rather work with existing technology and focus on the software aspect of smartwatches. It did however, intend to cover this important subject and relevant research that has been carried out in this field was presented.

1.3 Approach

The work was carried out in three different phases, excluding a literature study on previous work. The first phase was an online based survey, aimed to get an overview of people and their smartwatch experiences and expectations. The second phase was the implementation phase, where a few applications were developed for the Apple Watch. Lastly a user study was carried out using these applications to gain more understanding of what a usable smartwatch application might look like. The first and third phase covered the goal of finding out what suitable applications for the smartwatch might be, while the second phase covered the goal of getting into Apple Watch development. Apart from these phases a real world field study on using the Apple Watch as a flight traveling tool was carried out and personal reflections were presented.

1.4 Thesis structure

This thesis is structured in seven different chapters. What you just read was the introduction chapter containing the objective of this work. Following this is the background chapter. This chapter provides a summary of the topic and goes through the history of wearables and previous research done in the field. It also presents the smartwatch platforms currently out on the market. The chapter after that is the method chapter which describes the methods carried out in this work. Following that is the implementation chapter containing a thorough description of the various Apple Watch applications that were implemented. Chapter 5 and 6 provide the results and discussions of the various outcomes for the different methods. Ending this thesis is the conclusion chapter that summarizes all the work done and suggests further studies.
Chapter 2

Background

This chapter provides a quick summary of the topic of smartwatches. It goes into previous research that has been done in the field and provides an overview of the current smartwatch platforms. Ending this chapter is a section about development for the Apple Watch, which is the platform this thesis mainly will revolve around.

2.1 History of smartwatches

The idea of a smart watch, or a computer attached to your arm, is not very new. Some argue that the first wearable technology dates back to the 17th century in China, where a fully functional abacus ring has been found[5]. This calculation tool in the form of a ring was of course not a computer, but it indicates that the concept of useful wearables is quite old. In the mid 20th century there were many examples of wearable devices in science fiction, such as Knight Rider[6]. In the mid 1970s the first digital watches started to appear and with them a new type of watch, the calculator watch, was born. The first calculator watch was introduced by Pulsar and this type of watch was made quite popular in the 1980s with companies such as Casio in the lead[7]. Up until late 1990s watches were being stuffed with technology, such as Seiko’s TV watch[8] and the WatchPad, a Linux smart watch made by IBM and Citizen[9]. But since then the hype for smartwatches has died out, much due to technical limitations and how expensive the hardware was. Simpler cheaper watches became more popular, but even these were about to get less popular with the boom of mobile phones in the 2000s[11]. The wristwatch had gone from replacing the pocket watch to being replaced by a pocket phone as the main source for telling time. In a survey carried out by the market research group YouGov it was shown that almost 60% of 16-34 year olds use a phone as their primary timepiece[10]. Now after 40 years of technological improvements the smartwatch seems to make a comeback and is one of the most hyped electronic devices of our time[33]. Starting with the crowd-funded project Pebble in 2012, smartwatches were brought back in the spotlight and has become a commercial product waking interest in all the big technology companies[34]. The smartwatch has still not reached the same popularity
level as smartphones, but now that the two main actors, Google and Apple, both are in on the field the future looks exciting[3].

Figure 2.1: From left to right: the abacus ring, a Casio calculator watch and the Apple Watch

2.2 Previous research

There are two main areas previous research has been focused on when it comes to smartwatches. The first one handles different types of uses for it. The other field focuses more on modalities; ways of interacting with the smartwatch. These two areas are each given their own section and are presented below.

2.2.1 Smartwatch modalities

How does one interact with a smartwatch? Typically a watch is worn on the wrist and consists of a band and a watch face that shows information such as time and date. A digital watch usually has buttons as a mean to interact with the watch. The first calculator watches also used buttons as their input[7]. But there has been research on alternative ways to communicate with the watch. Xiao, Laput and Harrison presented a complementary input method using the watch face as a multi-degree-of-freedom, mechanical interface. With this watch face one is introduced to several new input commands. These methods, as shown in figure 2.2, can be used for tasks such as zooming in and out using the twist command and pan around maps using the pan command[12]. Another rather radical concept for a smartwatch is the project Facet. In this concept there is no main watch-face, but rather the whole band is made up of several screens, as seen in the figure 2.3. Using touch gestures one is able to control what each screen displays and extend a view over several screens[13].

Instead of just focusing on the smartwatch and its modalities there have been research done on how a smartwatch can augment other interactions. One project
2.2. PREVIOUS RESEARCH

Figure 2.2: Robert Xiao, Gierad Laput and Chris Harrison’s prototype’s modalities as shown in their report [12].

that did this was the Duet project [13]. This project made use of the smartwatch accelerometer and touchscreen to add new ways to interact with the smartphone. For instance, by using this sensor data from the watch an application on the smartphone could detect different types of touches, such as taps by the finger, the side of it, or the knuckle. In a text editor this could enable patterns such as marking texts using the knuckle, underline text with a fingertip touch and scroll the text using the side of the finger. Another creative way of using a smartwatch to augment user interaction was presented by Simon Mayer and Gábor Sörös. In their work they presented an interface, using Google Glass and a smartwatch, that could detect different home appliances and alert the watch to act as a controller for that device. For instance the user could see a stereo via the glasses, and after some processing done on a smartphone connected to the Internet, a user interface with music controls would be presented on the smartwatch [15].

Figure 2.3: The facet smartwatch using several screens covering the whole wrist.

Other modalities present in current smartwatches include haptic feedback, ac-
celerometers and various health sensors such as pulsemeters. A few use cases for these modalities are presented in the next section.

Although there have been many experiments on alternative modalities the current smartwatches on the market rely mostly on either buttons and/or touch-screens together with voice input. Also they typically have sensors for movement and health data and have some sort of haptic output.

2.2.2 Smartwatch fields of use

So what is a smartwatch capable of doing? Some research has been done in this field and some of it is presented in this section.

One area that the smartwatch might be of good use for is in the health industry. A research conducted by Wile, Ranawaya and Kiss showed that a smartwatch could be used as a reliable tool for distinguishing the postural re-emergent tremor of Parkinson disease from essential tremor. It made use of the accelerometer to measure different tremors. Another research that used the accelerometer was carried out by Lockman, Fisher and Olson. Their research showed that a smartwatch could be used as a tool to detect tonic-clonic seizures, which might be of great use for caregivers of epilepsy patients, since an unwitnessed seizures could cause injury and even death. Another health related application the smartwatch has been recorded to be of good use for is for diabetes patients. With a smartwatch application users were able to update and keep track of various diabetes related data in a much quicker and at-hand way than compared to a smartphone. The time saving aspect of a smartwatch has been discussed in a previous research. According to a news report a regular person picks up the phone more than 150 times per day, many times just to check notifications, which can be reduced by the use of a smartwatch. A smartwatch can also help users notice incoming messages and calls more frequently. According to a research conducted by Nokia up to 30% of men and 40% of women frequently miss their notification.

Lately the smartwatch has been promoted to be able to carry out many different tasks. Some of these areas include: navigation, weather, time, calendar, communication, games, fitness and notifications. But just because it can do something does not necessarily mean it is suited to do it. It is possible to play first-person shooting games like Doom on a watch, but does it really fit the format? As mentioned earlier this is one of the things this thesis aims to answer.

This section provided an overview of what has been done in the field. Next up is a presentation on the current market and what the current smartwatches are capable of.

2.3 Current platforms

There are many types of smart wearables out on the market today. When it comes to smartwatches one can arguably say that there are three main platforms that are currently being focused on. Each of these are covered in the following sections.
2.3. CURRENT PLATFORMS

Apart from these three there are several platforms that focus more on the fitness aspect of wearables. This type of smartwatch or smart-band is typically offering functions such as step counters, pulse meters and everyday fitness tracking functions. These usually synchronize the data with an application running on a smartphone. What these devices usually lack compared to the other main platforms associated to smartwatches is the integration of its system with a smartphone and being able to present notifications and let third party applications be run. The fitness bands can usually act more or less independent from a smartphone, which can not really be said about the Pebble, Android Wear and Apple Watch systems. These three systems are more or less depending on being synchronized to a smartphone and separating these devices from their parent devices strips down a lot of their functionality. A smartwatch that reverses this pattern of having the smartphone carrying out the heavy computation and displaying it on the watch is the Neptune Duo. Instead of using a smartphone’s computation power, all the power is put into the watch. To this watch a smartphone can be synchronized and more or less act as a display. This pattern is interesting but is not something that the general trends goes towards at the time of writing\[24\].

2.3.1 Pebble

The Pebble smartwatch is said to be the device that started the commercialization of smartwatches\[34\]. It was released in 2013 after receiving most of their fundings from the crowd-founding platform Kickstarter. When connected to an Android or iOS device via Bluetooth it will vibrate and display text messages, emails, incoming calls, and various other notifications when received on the paired device. It uses four buttons as input and the next version, Pebble Time, will make use of voice input as well. The screen used is not a touchscreen but it uses a Memory LCD screen making the watch consume much less power than other comparable smartwatches.

Pebble also has its own free SDK that lets developers make their own applications and watch faces for the watch. It also offers frameworks to communicate with Android and iOS applications directly. Development for the Pebble is mainly done in the programming language C but there is also support for Javascript. The Pebble OS in its current version allows the user to have up to 8 different applications synchronized to the watch. These are arranged from the Pebble application run on the parent device as seen in figure 2.4.

The Pebble OS is about to get an update with the upcoming watch Pebble Time. This update implements a timeline design, which is supposed to be used to display user-specific data, events, notifications and reminders in both the past and the future. This is supposedly going to make it easier to find the relevant information\[25\].
2.3.2 Android Wear

Samsung and Sony were one of the first actors after Pebble to jump on the smartwatch train, introducing devices such as Galaxy Gear and Sony SmartWatch. At the time of their release no broad platform specific for wearables such as smartwatches had been made. These devices ran on tweaked custom made operating systems and were quite limited in what devices they could be paired to[26]. On March 18, 2014 Google announced Android Wear, their initiative to bring the Android OS to smartwatches and other wearables[27]. Unlike Pebble, Google do not make the hardware and only provide the operating system to other makers. At the launch, companies such as Motorola, Samsung, LG, HTC and Asus announced their support for the platform[28]. Given this situation it is hard to list the specifications for Android Wear devices as they can vary. Typically though, a device running Android Wear has a touchscreen and microphone as their input. Also some
kind of physical button is usually present. Most watches also have accelerometers and for output a speaker and some kind of haptic device.

One interesting aspect of the Android Wear platform is that it supports both circular and rectangular watch faces. Another feature that stands out is the integration of Google’s voice command service, enabling users to dictate commands straight to the watch to send texts, make calls, add reminders and such. Just like other typical smartwatches Android Wear devices are paired to a smartphone using Bluetooth, in this case an Android device, making it respond to notifications and other applications. There is also support for developers to develop applications and watch faces using the Wear API[29].

The operating system is navigated through something Google calls cards. These cards show useful information such as weather and notifications. The cards are navigated by swiping vertically on the screen, showing the different cards in an order that the operating system believes to be most relevant at the time. For instance the first card shown at the end of the work day might be a card showing how long it takes to get home. To interact with these cards one swipes the card to the left which presents additional views or actions. Some cards can also be pressed to show more information. Swiping to the right dismisses the card. Figure 2.5 illustrates this flow.

2.3.3 Apple Watch

Apples entry into the field of smartwatches was announced in September 2014 and was released on April 24, 2015. Just like the other smartwatches introduced earlier it is paired to a smartphone and acts as an extension. Just like Android Wear it supports voice commands with Siri. Like the other platforms, Apple Watch allows third party applications, but currently no support for watch faces is implemented. These applications are synchronized using an application run on the parent device. Two new modalities were introduced with the Apple Watch; the force touch and the Digital crown[35]. The force touch is a way to detect the pressure a user puts on the screen. Pressing hard on the screen typically brings the user to a menu of some sort in the Apple Watch OS. The digital crown is an input method that is similar to that of traditional crowns used on regular watches. It can be scrolled and pushed as a button, enabling the user to scroll content without obstructing the screen.

The operating system differs a bit from Android Wear and Pebble. Applications are accessed through an application screen that shows circular icons in a grid. Notifications are presented as they come in or are accessed through swiping down from the watch face. Swiping up from the watch face brings the user to something called the glance views. This looks a bit similar to the Android Wear cards, but are swiped left and right to be changed. Clicking on these glance views brings the user to the corresponding application. Unlike Android Wear these glances are ordered and set by the user and will not be adjusted by the system. The basic flow of the Apple Watch OS is shown in figure 2.6
CHAPTER 2. BACKGROUND

Figure 2.5: A simplification of the basic flow of the Android Wear OS.
Figure 2.6: A simplification of the basic flow of the Apple Watch OS.
2.4 Development for the Apple Watch

This section provides an overview for application development for the Apple Watch.

2.4.1 Xcode

Xcode is the primary integrated development environment used for developing software for Apple’s platforms. It has a built in interface builder letting programmers design interfaces and easily integrate it with their code. Most software built follow some sort of Model-View-Controller design pattern. This pattern separates an application into three different types of objects: model objects, view objects and controller objects. The model object holds the application’s data and defines the logic that manipulates the data. The view object is responsible for displaying and representing the data from the model. The controller object ties the model with the view and keeps the application consistent. See the diagram in figure 2.7. Xcode provides an IDE that makes it easy to implement the MVC pattern, using its interface builder and tools to connect the views to controller objects.

![Figure 2.7: A diagram showing the version of the MVC-pattern for typical Xcode projects.](image)

2.4.2 Obj-C and Swift

The main programming language used for development in Xcode has for a long time been Objective-C. It is a general-purpose, object-oriented programming language that acts as a thin layer on top of C. Objective-C can be seen as a strict superset of C as it should be possible to compile any C program with an Objective-C compiler, and to freely include C code within an Objective-C class. The syntax for non-objective oriented operations are identical to that of C but when it comes to object-oriented features the syntax is different. In June 2014 Apple announced a new language that is meant to replace Objective-C as the main language for iOS and OSX development. This language is called Swift and aims to have simpler syntax, streamline application development and be more resilient to erroneous code. An example of how the syntax looks compared to Objective-C:
2.4. DEVELOPMENT FOR THE APPLE WATCH

Obj-C:
NSString *str = @"hello,";
str = [[str stringByAppendingString:@" world"]]

Swift:
var str = "hello,"
str += " world"

Much due to type inference and various operators such as ‘+’ being allowed to represent various functions such as concatenation, Swift syntax looks more clean than Objective-C. An important feature that is introduced in Swift is the optional types. This means that an object can be either a pointer towards a value or be null, implying that an object that is not declared as optional is not allowed to be null and thus cannot be the cause of a null-pointer exception error [31].

Swift and Objective-C are compatible with each other and files written in either language can be used in the same project. If this was not the case the decision to go over to Swift from Objective-C would be a very hard one to make. Given that Swift also is supposedly going to offer better performance than Objective-C [35] the implementations in this project was done in Swift.

2.4.3 WatchKit

To develop applications for the Apple Watch a software development kit called WatchKit is used. This SDK was first published in November 2014 and offers an interface builder for the format. A WatchKit application differs from a regular iPhone application in the sense that the actual computation done is not on the actual device it is run on. Rather a WatchKit application can be seen as an extended user interface for a program run on the iPhone, as shown in figure 2.8. Apple has announced that native applications will be supported later on which probably will not necessarily require the logic to be run on the iPhone [32]. Heavier computation is probably preferred on the better CPU on the phone though, so the concept of doing computation on another device and let the watch act as a extension will probably still be relevant.

There are three different types of interfaces that a developer can design. Firstly there is the main interface for the WatchKit application itself. This view can contain images, labels, buttons, sliders and other common UI elements. When putting out the these elements on the interface they are arranged by Xcode and stacked vertically on top of each other. The elements can be changed and positioned using several attributes in the interface builder, but in essence they all need to follow the layout rules provided. The other types of views are built similarly but have some restrictions in what types of elements they can contain. The glance view and notification, the other two types of views available, do not allow interactive elements such as buttons or sliders to be present. The size of the glance view is also fixed and is not scrollable like the notification view is.
The Xcode IDE provides a simulator for the Apple Watch screen and can be used for debugging. The simulator does not provide the Apple Watch OS, like it does for the iPhone. This means that the different views each have to be separately tested.
Chapter 3

Method

This chapter is divided in four sections, one for each phase carried out in this project. In these sections the objective of each phase is stated and methods used are described and motivated.

3.1 User survey

3.1.1 Objective

There were two main objectives for the user survey. Firstly it was to find out what current smartwatch users think of the format. What are its limitations and what is it being used for? Secondly it was to find out what people expect from a smartwatch. What is their mental model of such a device? This included non-users as well. This data was very useful for answering the research questions stated in chapter 1.

3.1.2 Approach

After careful consideration a form containing questions of interest was made and reviewed by the supervisors at KTH and The Mobile Life. The form was implemented using Google Forms. The main target group was current smartwatch users and people interested in the technology. In order to find matching participants several forums on the popular social networking service reddit were approached. Other relevant online forums were also reached out to, but the majority of the survey takers were from reddit. The survey was published on March 9th, the same day that Apple held a keynote presenting the Apple Watch[35], which meant that the topic of the survey was of great popularity. The survey was kept open for about a week and got over 1400 submissions, most of which came in during the first day.

3.1.3 Questions

The questions in the form can be found in appendix A. It is a paper version of the online based form published on Google Forms.
3.2 Development

3.2.1 Objective

As stated in the objective of this thesis in chapter 1, one of the purposes of this work was to develop new applications for smartwatches or more specifically the Apple Watch. Therefore some kind of application development was essential for this project. The developed applications should also be used as a method to bring answers to what applications are suitable for the smartwatch format. To do this a few different applications were developed, covering different areas of use. Also, different technologies were used for the different applications. The various applications that were developed are covered in the next section.

3.2.2 Applications

The smartwatch is predicted to be a device that provides its user with context based information. Two of the applications developed were based on this premise but with a few different technologies. These applications are covered in the implementation chapter under the sections SL Watch and Museum Audio Guide. A relatively unexplored area is the one of gaming on smartwatches. An attempt to see how a game could look like on a smartwatch was done and is described in the implementation chapter under the section Memory Game. These different applications made use of several different technologies for the Apple Watch. The reason for this was to be able to try them out in the usability study and hopefully come to some good conclusions about them.

3.3 Usability study

3.3.1 Objective

The objective of the usability study was to get a qualitative insight in how users reason about smartwatches and to get feedback about the developed applications. The different methods used for this usability study are described further down this chapter.

3.3.2 DECIDE framework

A well-planned evaluation is driven by clear goals and appropriate questions. There is a framework called DECIDE that helps one carry out a good evaluation. The DECIDE evaluation framework provides a checklist consisting of the following points:

- Determine the goals
- Explore the questions
3.3. USABILITY STUDY

Choose the evaluation paradigm and techniques

Identify the practical issues

Decide how to deal with the ethical issues

Evaluate, interpret, and present the data

Determine the goals

As mentioned earlier the goal for the evaluation was to get qualitative insight in how users reason about smartwatches and to get feedback about the developed applications. Are users finding the applications useful and will they be able to carry out the different tasks brought upon them?

Explore the questions

There were a few questions at hand that were of interest. Aside from determining what the users thought of the applications presented to them, it was of interest to see what the users felt about the different modalities for the Apple Watch and how these worked out for the applications. Were they carrying out the scenarios given to them in a timely and error free manner?

Choose the evaluation paradigm and techniques

Given the limited time for the user study a quick method was needed. A popular method used in many usability studies is the System Usability Scale[39], described more in section 3.3.3. In addition to this there were a few additional techniques used in order to gain some more specific information about the user’s opinions and thoughts during the test. One of these methods was the think aloud method, which is described in section 3.3.4 and the other was a questionnaire that contained questions of interest.

Identify the practical issues

One important issue was that the device used for this test would not be available until the very end of the project. This meant that the time-frame for the actual evaluation was limited and thus had to be planned and carried out within a tight schedule. Also given the popularity of new Apple devices there was a risk of not getting a device on time.

Another thing that could have been be an issue was to find participants. The main method used to find testers was by reaching out through social media.

Decide how to deal with the ethical issues

Every participant was introduced to the study and got an explanation of its purpose. They were informed that all data gathered would be anonymously presented in the
CHAPTER 3. METHOD

report. The participants were also asked to sign a consent form that can be found in appendix B.

Evaluate, interpret, and present the data

This study resulted in both quantitative data and qualitative data. The quantitative data, such as the SUS forms, was presented in the results chapter. The qualitative data was mostly used in the discussion chapter.

3.3.3 System Usability Scale

The System Usability Scale (from here on referred as SUS) is a quick and reliable tool to measure usability of a system. It was created by John Brooke in 1986 and has since become an industry standard for evaluating a wide variety of products and services. The SUS consists of a questionnaire of 10 items with five response options; from Strongly agree to Strongly disagree. The ten questions are:

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 cumbersome 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.

According to previous research, the advantages of the SUS is that it is easy to scale and administer to participants, it can be used on a small sample size with reliable results and it can reliably differentiate between systems that are usable or not. A sample size of as few as two people can generate reliable data, even though it is said that five is a good minimum number of participants for this type of evaluation.

One point that might be a bit unclear is on how to interpret the data. The basic method is to convert the participant’s score for each question to a value between 0 and 4 and multiply this by 2.5 to get a score between 0 and 100. This should not be seen as a percentage score though, and instead only be considered in terms of their percentile ranking. A score above 68 is seen as above average and anything below it under the average.
3.3.4 Think aloud

The think aloud method is a method that lets the participant think aloud while they are performing a set of specified tasks. It is often used in usability tests and was first carried out by Clayton Lewis. During the usability study for this thesis this method was used. The participant was asked to express what they were thinking, looking and doing as they carried out the tasks. This enabled the observer to gain an insight in the mental model of the user, which might not have been apparent from silent observations. The think aloud process can be done in two ways, either concurrently during the task or done in retrospective after the task has been carried out. During the test carried out in this work the user was asked to think aloud concurrently if he/she felt it did not hinder the actual flow of doing the task, in which the user instead was asked after the test was carried out.

The test was audio recorded and written notes were taken.

3.3.5 Test plan

According to the U.S. Department of Health and Human Services the following elements needs to be included in the test plan.

Scope The scope of this test consisted of several applications for the Apple Watch listed below. An in depth description of the first three applications can be found in chapter.

SL Watch Version 1.4 of SL Watch was tested. The functions required to fulfill the task were tested, which means that all functions were not necessarily used.

Museum The prototype version as of May 5th was tested. All functions were tested.

Memory game The prototype version as of May 5th was tested. All functions were tested.

Messages The default messaging application on the Apple Watch running Watch OS 1.0 was tested. The functions to see incoming messages and to reply using both quick replies and voice input were tested. Further documentation of this application can be found on Apple’s homepage.

Purpose The purpose of this test was to see if users found the aforementioned applications for the Apple Watch to be useful given the specific scenarios. Was the user able to carry out the tasks for SL Watch and Messaging in a timely and error free manner? How well did a smartwatch work as a notifier and remote controller for triggering location based audio guides? What did users think about the messaging service? How well did the different reply functions work? What did users think about reading content on the watch? How did the crown compare to touch input for scrolling that content?
In summary the aim was to get qualitative insights in how users reasoned about smartwatches, its modalities and the tested applications.

**Schedule and Location** The evaluation was conducted on the 9th and 10th of May in Enskededalen. Additional tests were carried out in The Mobile Life’s office on later dates.

**Sessions** The session started with a small introduction to the thesis work and the user signed a formal agreement indicating that the results would be published as anonymous data in the final report. The tester was then be introduced to the Apple Watch and given a small tutorial on how to use the device. Next the tester was given the different scenarios and carried out these. During this period the mediator communicated with the tester and asked him/her to think aloud. After the tasks were finished the participant answered a small questionnaire and filled out the Standard Usability Scale form. The whole session took approximately 45 minutes.

**Equipment** The user was equipped with a 38 mm Apple Watch and an iPhone 6 connected to a pair of earphones (used for the Museum application). Apart from these devices a second mobile phone was required (to send and receive text messages). Devices used as beacons were also required (in this case an iPhone and an iPad were used). An audio recording device as well as a timer were also used to track and record the tests.

**Participants** Participants were recruited from social networking services. People interested in this new technology was the main target group.

**Scenarios** There were several scenarios for each of the different applications.

**SL Watch** The first task for this application was to find when the next train departed from the station closest to the testing site. The second task was to mark this station as a favorite station. The third task was to find another nearby station and find the location of it.

**Museum** The tester was asked to walk towards one of the paintings/beacons and trigger a notification on the watch and use this to start playing the audio guide. The user was then asked to try and read the guide text, using both the crown and the touchscreen. The user was also asked to try and play/pause the audio guide using the Apple Watch as a remote.

**Memory game** The tester was asked to use the voice command to open up the application and play the game.

**Messages** The tester was asked to keep up with a messaging conversation. This made the user try out different ways of communication, such as quick replies, emotes and voice dictation. The first two incoming messages were: ‘Hi’ and ‘Do you like pie’. Depending on the participant’s reply the last incoming message was either: ‘Which kind?’ or ‘Why not?’.
3.4. FIELD STUDY

Metrics The System Usability Scale and a questionnaire were used as the subjective metrics for this test.

Quantitative metrics The different scenarios were timed. Whether the user succeeded or failed the task was logged.

Roles The writer of this thesis was the sole facilitator for this test and kept notes with an audio recording device and measured the time spent on the different tasks with a timer.

3.4 Field study

One of the areas that the Apple Watch has been promoted to be of good use for is in the field of traveling and more specifically for airlines. The Mobile Life is very active in this area and has several customers in this business, which makes this an interesting topic to investigate. There are two main aspects to this type of application that are being on the radar at the moment. Firstly it is the possibility to get an overview and information about the current flight. Secondly it is the ability to have the boarding pass on the watch and proceed through the security and boarding phase using this. To research this type of applications the first idea was to make a prototype application and carry out a user study. A prototype application was made and is described in the next chapter, but given the fact that the author of this thesis had the opportunity to try this functionality in a real world scenario another approach was chosen. Instead of carrying out a user study as with the previous applications, a real world field study was carried out by the author. The evaluation was based on the two areas pointed out earlier and was carried out on a British Airways flight from Narita, Tokyo to Heathrow, London and from there to Arlanda, Stockholm. The results and reflections of this study is presented in the discussion chapter.
Chapter 4

Smartwatch application development

This chapter goes through specific details of the applications that were implemented. The applications are not presented in the order they were implemented but rather in the order of complexity. This enables a good flow of new concepts specific for the Apple Watch to be presented. A quick overview of the structure of an application for the Apple Watch can be read in the background chapter. This chapter aims to go into more detail in this matter.

4.1 Memory Game

The first implemented application was a game based on the concept of the game Simon says. The game layout consists of four distinct buttons and the idea is to press these buttons in a sequence provided by the game engine. At the start of the game one button flashes. This is the first button in the sequence. The player then presses this button. If pressed correctly the game once again flashes the first button, followed by another button. The sequence now consists of two buttons and the player is asked to press the buttons in the same sequence. This game pattern goes on until the player fails to remember and press the correct sequence of buttons.

4.1.1 Interface view

As mentioned in the chapter applications developed in Xcode follow the MVC design pattern. There are three different types of views an Apple Watch application can have, as described in section. For this application only the interface view was implemented. The other types of views are presented and described more in the sections for the following applications. This game’s interface is quite simple and consists of four game buttons plus a start button and a label showing the current score and high score, as seen in figure UI elements using the WatchKit interface builder cannot be put arbitrarily out on the view. They are bound by hierarchical layout rules and each take up its own space, meaning it is not possible to put an image on top of a label for example. UI elements are placed next to each other either
horizontally or vertically, but not both at the same time. Reading this one might wonder how it is possible to implement a view such as the one for this application. The solution is to use groups that contain UI elements. Each group can be set to align elements horizontally or vertically, and the key point here is that groups can contain groups. To get the layout shown in figure 4.1 groups are used as shown in figure 4.2. The view itself places elements vertically and contains four elements, two of these being groups. The "Label Group" places its two label elements horizontally. The "Game Buttons Group" contains two groups that are vertically placed, while containing groups use a horizontal layout enabling the placement of the buttons next to each other. This way of building interfaces might not be what most iOS developers are used to, but it is an essential design pattern to grasp if one wants to build more advanced layouts for the Apple Watch.

### 4.1.2 Interface controller

Connected to the view previously described is an interface controller object. In a typical MVC design pattern there would normally be a separate model object interacting with the controller, but given the very simple nature of this game the model is integrated in the controller. Programmatic access to the view is enabled through this interface controller/model. Typically there are three different methods that are of interest to override for a controller class. These are:

```swift
awakeWithContext(context: AnyObject?)
```
4.1. MEMORY GAME

This method is called when the view is initialized. It is used for setting up the interface and can be called with a context object. The context object can be of any type and enables the programmer to easily pass on needed data between views.

**willActivate()**

This method is called every time the view becomes visible. The difference between this method and the previous one, except for the context parameter, is that this method is called every time the view becomes active, for instance after waking up the application from background mode, which won’t invoke the awakeWithContext function.

**didDeactivate()**

This method is called every time the application gets deactivated. It is useful for scenarios when one might want to save the current state of the application.

Apart from these methods the interface controller contains outlets to the UI elements in the view. An outlet corresponding to the start button looks like this in Swift code:

```swift
@IBOutlet var start: WKInterfaceButton!
```

Using these outlets it is possible to control the view and trigger code from events such as button presses.

4.1.3 Game Logic

As mentioned earlier the model and in this case the game logic is written in the same file as the interface controller. At startup the application disables all the play buttons and enables the start button. A click on this button starts the game. The game itself contains two phases, one when the game presents the sequence and one when the player inputs this sequence. To keep track of the state of the game a boolean variable called isPlaying is set. When the player clicks the start button the first element in the sequence is generated and shown. The sequence element is a randomly generated enumerate value representing the four buttons, which then is appended to the sequence list. The sequence is then shown on the view using a timer object that goes through every element in the sequence list and calls a function that flashes the corresponding button. After the sequence has finished showing all elements, the isPlaying boolean variable is set to true and enables all the buttons for the player to press. An integer variable called clickCounter keeps track of the pressed sequence. For every button-press this value is checked against the index of the sequence list. If the players succeeds to input the whole sequence correctly the game updates the score, enters the first phase again, generates an additional element to the sequence and the game goes on. When the player eventually fails the game resets and checks the current score against the high score, and saves it if a new record is set.
4.2 Museum Audioguide

The Museum Audioguide application is meant as a prototype for how a smartwatch application can act as a remote and trigger for audio guidance in a setting such as a museum. Apart from having a regular interface, from which the user can control the audio guide, the application notifies the user when standing near a point of interest.

4.2.1 Application overview

As it is not possible to playback audio from the Apple Watch, as of version 6.3 of Xcode, it has to be done on the connected iPhone instead. The watch application calls the parent application running on the iPhone to trigger the audio functionality. It is not possible to detect beacons on the watch either so this is done on the parent application as well. The figure 4.3 shows an overview of how the application is structured. The parent application running on the iPhone handles the audio playback and the beacon logic. The WatchKit applications handle the visual presentation and acts as remote for controlling the audio guide.

![Flow of the application](image-url)
4.2. MUSEUM AUDIOGUIDE

4.2.2 iBeacon

An iBeacon is a low-powered, low-cost transmitter using Bluetooth Low Energy technology. Using iBeacon technology one can notify and perform actions when a smartphone or other device is in near proximity to the device. In this implementation an external iPhone and iPad are acting out as beacons broadcasting on a specified UUID (Universally Unique Identifier). Beacons broadcasting on the same UUID are called a region. The parent application monitors this region, determining whether the device is in the region or not. Monitoring regions can be done in the background but that does not provide very accurate proximity data. In order to measure the distance to the beacons another function needs to be called that cannot be indefinitely be run in the background. By default the time the ranging function, the function measuring distance to beacons, can be run in the background is limited to less than ten seconds. This can be increased to three minutes by requesting additional background runtime. Given these constraints one must find a solution to measure distance to beacons in a way that fits for the specific application in question. Since the Museum Application is supposed to run in the background and send a notification when nearby a beacon there are two approaches to go with:

- Use a beacon with a small enough region so that when the application monitors that beacon it knows it is close.

- Trigger the three minute ranging function and hope the time is enough to carry out the task.

Of course these two approaches can be combined, meaning that the monitoring function triggers the extended background task ranging the beacons. Unfortunately the broadcasted region range cannot be adjusted on the iOS device which reduces the usefulness of the first approach a little bit. But given the prototype nature of this application this would still be a valid option to try out and test on real users.

When the parent application running on the iPhone finds that the beacons are in the proximity range a local notification is created. The application distinguishes the two different beacons by reading their major and minor numbers, which are the two numbers used to differentiate beacons in the same region. This way, different notifications can be created depending on which beacon it was that triggered from. Notifications are shown on the Apple Watch if the iPhone’s screen is turned off, meaning that the application created during this process will be pushed to the watch.

4.2.3 Notification view

The second type of view a developer can design for the Apple Watch is the notification view. Designing a notification view is fairly similar to the interface view but there are a few differences. One limitation is that interactive elements such as
buttons or sliders are not allowed on it. Buttons that are shown on the notification view are defined by the notification received, in other words these are not configured in the interface builder. Another difference for this view is the notification banner, called the sash, and the application logotype as seen in the lower left corner of figure 4.4.

In the controller class for the notification view there are two functions that are of main interest. These are:

```swift
didReceiveLocalNotification(localNotification: UILocalNotification, withCompletion completionHandler: ((WKUserNotificationInterfaceType) -> Void))
```

This function delivers a local notification object to the interface controller for processing.

```swift
didReceiveRemoteNotification(remoteNotification: [NSObject : AnyObject], withCompletion completionHandler: ((WKUserNotificationInterfaceType) -> Void))
```

This function delivers a remote notification payload to the interface controller for processing.

Using these functions one can populate the notification interface. As mentioned earlier no buttons or interactive elements are allowed on the view, but instead actionable buttons are defined in the received notification. No code for handling the button press is implemented in this interface controller. Instead the attributes of the action that should be executed is defined when creating the notification. The controller that handles notification-actions is the main interface view controller, which is described in the next section.

### 4.2.4 Interface view

The interface for the main application consists of two different views. The first screen shows a table of buttons representing different paintings as seen in figure 4.4. A press on one of these buttons takes the user to the corresponding view for more details. This view contains a button at the top for remote playback of the audio guide and a small description text down below, as seen in figure 4.4.

One very important aspect the interface controller for the initial view has is that it needs to respond to eventual notification actions. This is handled in the functions:

```swift
handleActionWithIdentifier(identifier: String?, forLocalNotification localNotification: UILocalNotification)
```

```swift
handleActionWithIdentifier(identifier: String?, forRemoteNotification remoteNotification: [NSObject : AnyObject])
```
These functions deliver a notification payload and a user-selected action to the interface controller. The difference between the functions is that one is responsible for local notifications while the other handles remote ones.

If a notification action sends the user to this view one of these functions will be called, depending on whether it is a remote or local notification. With the identifier parameter it is possible to determine which action was triggered and handle it appropriately. The payload is also supplied which in many cases is desired. For this application the payload contains data about which painting the notification was about.

### 4.2.5 iPhone application

When the WatchKit application run on the Apple Watch triggers the event to play the audio-guide it makes a call to the parent application run on the iPhone. This is done by a function in the WatchKit extension that opens the parent application. The parent application then handles this call, and in this case makes use of an audio library and plays the audio guide from the phone.
4.3 SL Watch

The idea behind the SL Watch application was to provide a quick and always-at-hand way to find out the next departure from a nearby station. To present context-aware information like this has been predicted to be one of the areas a smartwatch will be of good use for [36]. Before going into more detail about this application a small overview is presented.

4.3.1 Overview

As any other application for the Apple Watch it contains a main application. The flow of this application can be seen in figure 4.5. The initial interface presents a start screen containing two buttons, one for nearby stations and one for favorite ones. A press on one of these buttons takes the user to another view presenting a table of either nearby or favorite stations. Pressing a station button presents the

![Figure 4.5: The storyboard of the SL Watch application.](image)

As any other application for the Apple Watch it contains a main application. The flow of this application can be seen in figure 4.5. The initial interface presents a start screen containing two buttons, one for nearby stations and one for favorite ones. A press on one of these buttons takes the user to another view presenting a table of either nearby or favorite stations. Pressing a station button presents the
4.3. SL WATCH

user with a view for the departures from that station. The last option a user has is to make use of the force touch modality to show three different menu items. The selectable menu items are:

**Filter** This item lets the user filter departures depending on the type of transport.

**Map** Selecting station location presents the user with a map providing the station’s location. Clicking the map opens up the native maps application on the Apple Watch. From this application the user is able to use navigation services to get to the point.

**Glance** This option set the current station to the glance view. This is described more in detail in section 4.3.6.

The following sections give a bit more insight in the various technologies used in this application.

4.3.2 MMWormhole

According to the specifications provided by Apple there are a few things that are more appropriate to run on the parent application. Some things are not even possible to do in the WatchKit extension application. The following quote is found on their developer’s page:

> The best solution for performing any long-running tasks is to let your iOS app perform the task instead. For example, instead of starting location services in your WatchKit extension, start it in your iOS app.

In SL Watch the geographic location is retrieved from the parent application using the CoreLocation library. One problem with this pattern is that the communication between the parent and child application can be cumbersome to program. Luckily there is a very useful library called MMWormhole that eases this problem and enables the system to easily pass data or commands back and forth between the two locations. In SL Watch this is used for passing a message from the parent application to the Apple Watch application containing location data after the location handler has managed to get the current location. This data is then used for the various APIs described in the next section. A quick overview of this flow is illustrated in figure 4.6.

4.3.3 API

To get the data needed for SL Watch application two different API:s (Application Programming Interface) were used. These are provided by Trafiklab which is a community for open traffic data and was built in collaboration with Samtrafiken, Storstockholms Lokaltrafik and Victoria ICT[52]. For this project the needed functionality was to find nearby stations using geographic locations and to find departures from them. To provide the nearby stations the API ‘ResRobot - Sök resa’ was
used. Sending the longitude and latitude data plus a radius to the API’s function StationsInZone results in a JSON (JavaScript Object Notation) reply containing station data for the stations in the area provided by the parameters. To get the departures for a station another API called "ResRobot - Stolptitabeller" was used. This API provides a JSON response with the next departures within 30 minutes given the station’s location identifier, which was extracted from the previous API call. To make requests to these APIs a library called Alamofire was used. Alamofire is a HTTP networking library written in Swift and is compared to the immensely popular networking library AFNetworking written in Objective-C [53]. Using this library a HTTP request can be sent and handled with a custom response handler. In this case, if the requests were successful, the JSON data provided would be converted into a model object (presented in the next section).

![Figure 4.6: The communication flow when requesting stations in location.](image)

### 4.3.4 Model and row controllers

The data received from the API calls are stored in the custom made model classes Station and Departure. They are used to pass on data between different views. This is very handy for instance when populating data for the departures table in the departures view. Each row in the table has a controller class that is fed a Departure object to setup itself with. The class Station also follows the NSCoding protocol. Implementing this protocol enables easy archiving which is of good use since stations should be able to be saved as favorites. Stations also have their own row controller which is responsible for displaying the station name on the row and handling the favorite button. Clicking that button archives or unarchives the corresponding station saving/deleting it from the favorites. Apart from having the class Station comply with the NSCoding protocol the following two rows of code need to be executed before archiving/unarchiving can be done.
4.3. SL WATCH

NSKeyedUnarchiver.setClass(Station.self, forClassName: "Station")
NSKeyedArchiver.setClassName("Station", forClass: Station.self)

4.3.5 Force touch menu

One new way of interaction introduced in the Apple Watch is the force touch. The screen is force sensitive and reacts on how hard the user presses it. This technology opens up new possibilities in the way we interact with the watch. Typically in an Apple Watch application this method opens up a menu. The SL Watch application makes use of this on the departures interface. Using force touch on this view opens up three different options, as seen in figure 4.5. Below is a description of what happens when clicking these options.

**Filter** Choosing this option opens up a screen showing a list of switches, as seen in figure 4.5. The switches represent the different possible type of transportation that depart from the selected station. Disabling one of these options will hide all departures of that type in the departure list.

**Station location** Clicking this option takes the user to a screen showing a map image of the location of the station and the user. Clicking this image takes the user to the native maps application on the Apple Watch where it is possible to take a closer look at the location data and set up a navigation helper to find the place.

**Set as glance** Selecting this button sets the glance view to represent the given station. The glance view will be described more in the next section.

4.3.6 Glance view

In the previous applications two of the three different types of views are described, the normal interface view and notification view. For this application the third one, the glance view, is used. A glance view is accessed by swiping up from the clock screen on the Apple Watch. The idea behind this type of interface is to provide quick and timely information the user is interested in, without opening up an application. Just like the notification view it is not possible to have interactive elements on it. The space for UI elements is also fixed, meaning it is not possible to have content shown in a scrollable way, like in the other views. There are two areas UI elements can be placed. As seen in figure 4.7 there is a top and a bottom group to put content in. For the SL Watch application the top group displays the user information of which station the departures are for and the bottom part shows the next two coming departures, more than two departures do not fit the screen. Clicking a glance view brings the user to the main application. In the initial view controller for the main application it is possible to handle something called UserActivities. A user activity object is used for the Continuity feature for Apple products. Continuity is a feature that lets the user start one activity on one Apple device and continue the same
activity on another. For example one could start writing a note on an iPhone and continue editing it on a Macbook. Handling user activities on the Apple Watch application is done in the function shown below.

```swift
handleUserActivity(userInfo: [NSObject : AnyObject]?)
```

This makes it possible to make the SL Watch act different when opened from the glance view compared to a normal boot up. By defining a user activity, containing station data from the glance view controller, the main application’s interface controller can be programmed to boot up the departures view for that station. Because of this, SL Watch is able to boot up into the departures view for the station shown on the glance view when clicked, as seen in figure 4.5.

4.4 Flight Application

Although the flight application was not used for the user study it was of interest to see how a flight application for the Apple Watch might look like. The main interface of this application shows a list of all the current flights present to user. Each element shows short and observable information about that flight. Clicking one of these opens up a modal view presenting more information about the flight. Swiping left and right navigates through different screens of information. The first screen shows the flight information and other relevant data, such as the status of the flight. The next screen shows luggage information if there is any. The following two screens show a quick overview of the weather of the location and a currency conversion tool. An overview of the application flow is seen in figure 4.8.

![Glance Interface Controller](image)

Figure 4.7: The two different groups a glance view can hold UI elements in.
4.4. FLIGHT APPLICATION

Figure 4.8: All the views for the flight application.
Chapter 5

Results

In this chapter the various results from the different methods are summed up and presented. The results are not discussed and reflected upon in this chapter. That is saved for the chapter following this.

5.1 User expectations from smartwatches

In this section the parts of the survey that were of most relevance for this thesis work are presented. The full result can be found at the link provided in appendix C. A total of 1427 people participated in the survey between the 9th and 15th March 2015. Out of these participants 783 were current smartwatch users, out of which about 66% were Pebble users, while 644 were non-users. The vast majority, 94.8% of the participants were male. Full time workers and students were the most predominant occupations in the survey, and most of the participants were between 18-24 years old (42.6%) or between 25-34 years old (32.9%). For more detailed background information please refer to the link provided in appendix C. No major differences in the results could be found between the different gender, work or age groups. However, there were some differences between Pebble users and users of other smartwatch brands. These differences are presented in this chapter and are discussed in the next chapter.

The beginning of the survey contained questions that were specific for users and non-users. These questions are presented in the beginning of this section, while the questions asked to both users and non-users are presented in the end. Each of these questions are represented by a figure and an attached description text. These figures are later referred to in the discussion chapter.
CHAPTER 5. RESULTS

Figure 5.1: A word cloud representation\cite{55} of the answers for the survey question: "What do you mainly use your smartwatch for?". Asked to current smartwatch users.

Figure 5.2: A word cloud representation\cite{55} of the answers for the survey question: "If you owned a smartwatch, what do you think you would mainly use it for?". Asked to non-users.
5.1. USER EXPECTATIONS FROM SMARTWATCHES

Figure 5.3: A bar chart for the question: 'What types of application do you feel are suitable for the smartwatch format?'. Both users and non-users were asked this question.

Figure 5.4: The pie chart shows what current smartwatch users feel about the quality of their current applications. The bar chart to the right shows what platform(s) they base compare to.
CHAPTER 5. RESULTS

Figure 5.5: A bar chart showing what types of modalities current smartwatch users find useful.

Figure 5.6:

Figure 5.7:
5.1. USER EXPECTATIONS FROM SMARTWATCHES

Figure 5.8: The smartwatch is a good format for navigation services.

Figure 5.9: The smartwatch is a great tool for tracking personal health (fitness records, medication scheduling etc.).

Figure 5.10: The smartwatch can be greatly utilised as a remote for media players, smart home applications etc.
CHAPTER 5. RESULTS

Figure 5.11:

The smartwatch should act as an extending device for the smartphone.

Users
- Strongly Agree: 60%
- Agree: 31%
- Undecided: 7%
- Disagree: 2%
- Strongly Disagree: 2%

Non-users
- Strongly Agree: 42%
- Agree: 37%
- Undecided: 15%
- Disagree: 4%
- Strongly Disagree: 2%

Total
- Strongly Agree: 52%
- Agree: 34%
- Undecided: 11%
- Disagree: 3%
- Strongly Disagree: 1%

Figure 5.12:

The smartwatch can be used as a good method for payment in stores.

Users
- Strongly Agree: 27%
- Agree: 31%
- Undecided: 32%
- Disagree: 7%
- Strongly Disagree: 3%

Non-users
- Strongly Agree: 44%
- Agree: 29%
- Undecided: 20%
- Disagree: 4%
- Strongly Disagree: 2%

Total
- Strongly Agree: 35%
- Agree: 30%
- Undecided: 27%
- Disagree: 6%
- Strongly Disagree: 2%

Figure 5.13:

The smartwatch is great for communication and/or social networking.

Users
- Strongly Agree: 9%
- Agree: 31%
- Undecided: 30%
- Disagree: 23%
- Strongly Disagree: 6%

Non-users
- Strongly Agree: 9%
- Agree: 23%
- Undecided: 38%
- Disagree: 23%
- Strongly Disagree: 7%

Total
- Strongly Agree: 9%
- Agree: 34%
- Undecided: 34%
- Disagree: 23%
- Strongly Disagree: 7%
5.1. USER EXPECTATIONS FROM SMARTWATCHES

Figure 5.14: The smartwatch is good for location-based services (for example, displaying information about a nearby painting while in a museum, or finding a close by pizzeria).

Figure 5.15: The smartwatch should act as an independent device and not rely on communication with a smartphone or other device.
5.2 Usability study

The SUS scores for the tested applications are shown in figure 5.16. Although the purpose of the usability study was to gain qualitative data, quantitative data such as time taken and mistakes for each task was also measured. Five people participated in the usability study, four men and one woman. Three of these were students and two were software developers. All participants were familiar with smartphones and similar technologies, although only one of them had any previous experience with a smartwatch.

For the museum application the measured data was the time it took for one user to trigger the audio guide once the notification was received. On average this took 7 seconds while most of them did it under 5 seconds. No errors were noticed in the flow of this application.

For the SL Watch application three different moments were measured in time. The first task which involved the user to find the next departure of the nearest station took on average 21 seconds, while the fastest user did it in less than 5 seconds. The median time spent to complete the task was 16 seconds while the slowest user did it in 46 seconds. Marking a station as a favorite was done by most users in less than 3 seconds. No errors were noted except for one user that did not realize the position of the favorite button. After one mistake this user understood this error and proceeded to finish the task. Finding the location of a station took the users on average 27 seconds. Two users had troubles using the force touch gesture, but still managed to complete the task.

The measured task for the Memory Game was on how long it took for the users to launch the application with voice command. The average time spent on this was 13 seconds. The fastest user did it in 4 seconds and the median user did it in 9 seconds. The slowest user completed the task in 35 seconds.

The messaging application measured the time taken from the moment a text message was received until a reply was sent. In total three replies were sent by the user. The different reply options played a big role in how long the response time was. Using the quick reply function, containing predefined and adjusted replies depending on the received message, resulted in average response times of around 7 seconds. This was the most common reply option used for the first two messages received. The third received message contained a question that forced the user to reply with voice input. With this option the users answered the message in 34 seconds on average. No errors were noted during this test.
5.2. USABILITY STUDY

Figure 5.16: Box-and-whisker diagram displaying the SUS score for the different applications. The dotted line marks the median and the diamond shape marks the average.

The last questionnaire in the user study contained ten questions. Just like with the SUS forms, the participant filled out a Likert scale ranging from strongly disagree to strongly agree. The results are shown in figure [5.17], where each number represent their corresponding question number as shown in the list below.

1. I found it easy to read content on the watch.
2. I found the crown as a better way for scrolling content than swiping motions on the touchscreen.
3. I thought that the watch was good as a remote controller.
4. I found actionable notifications to be very useful for the watch.
5. I think voice input is a good way to interact with the watch.
6. I thought the touchscreen was a good way to interact with the watch.
7. I would imagine that it would be comfortable to interact with the watch for long sessions.
8. I found the glance view to be cumbersome to use.
9. I felt that it was easy to open applications through the application screen.
10. I feel that context aware applications, such as the SL Watch using GPS or the museum application using location tracking, is suitable for the watch.
Figure 5.17: Box-and-whisker diagram displaying how well the evaluators agreed on the 10 statements presented to them. The dotted line marks the median and the diamond shape marks the average.
Chapter 6

Discussion

The aim of this chapter is to put the results in context and discuss these in accordance to the aims of the thesis. As mentioned in the introduction chapter one of the main goals was to find out what a suitable smartwatch application might look like. To answer this two questions were stated: What indicates a usable smartwatch application and what is expected of a smartwatch? Keep these questions in mind while proceeding to read this chapter.

One thing that is not shown in the results chapter is the qualitative data gathered from the various methods. Some of this data is presented and discussed in this chapter instead.

6.1 Survey insights

To begin with, what are people expecting from a smartwatch? More than 96% of smartwatch owners in this study found their devices useful as seen in figure 5.6. Non-users were a bit more skeptical with about 30% undecided and 5% disagreeing or strongly disagreeing to this. In figure 5.1 and 5.2 a word cloud representation of what users use/would use their smartwatch for is presented. What can clearly be distinguished from this is that both figures highlight notifications as the main use for smartwatches. This claim is further strengthened by the fact that notifications topped the bar chart shown in figure 5.3. The two word clouds seem to highlight the same kind of words and the bar chart did not indicate any extreme differences between users and non-users opinions.

Other frequently occurring words in figure 5.1 and 5.2 were "checking", "control" and "tracking". These were used in combination of other frequent words such as "weather", "music", "fitness", "calendar", "messages", "e-mail", "time" and "phone". These words also appeared in figure 5.3 and it can be seen that areas that are quickly glanceable, such as time, weather and fitness received many votes while applications that need some sort of interactivity such as e-mail and messages received less percentages of the votes. Figure 5.13 also indicated that this was an area with many split opinions. One user made the following comment:
"Smartwatches are great for one way communication and social networking. Not sending responses."

This suggests that the watch is a great tool for glancing information, but not necessarily to be used for complex interactions such as responding to messages. The first part of this claim is further supported when looking at the results shown in figure 5.7. Around 91% of current user strongly agreed and about 9% agreed that the smartwatch is good for this purpose, meaning that almost no users were undecided or disagreeing with this statement.

The high praise for notifications may suggest why haptic feedback and visual output top the modalities users found the most useful, as seen in figure 5.5. Surprisingly audio output was not ranked very high and may suggest that users prefer to be alerted by haptic methods rather than alarm sounds. Another thing this diagram shows is that Pebble users differed in the preferred methods of input compared to users of other brands. Pebble users did not rank touchscreen and audio input as high as other users, while the the opposite can be said about external buttons. One explanation for this may be that Pebble users currently do not have access to touchscreen nor voice input, which might make them less inclined to mark that option in the survey, and vice versa for other users.

Some other specific use-cases that were brought up in the survey were in the field of health and fitness, navigation and in using the watch for payment and as a controller. These were fairly high up on the chart shown in figure 5.3. When it comes to health and fitness around 80% of the participant agreed that it is a good area for the smartwatch. Not many disagree, and the majority of the ones not agreeing were undecided. The numbers were fairly similar when it comes to using the watch as a controller, but when it comes to using the smartwatch as a navigation system opinions were split. Around one third were undecided about this field. Using the smartwatch for payment was also a field that split users. Right now this feature is quite limited and not usable in every country. Android Wear and Apple Watch provides NFC solutions for this, but the availability is very limited, which might be a reason for the split opinions.

As mentioned earlier notifications were one of the most endorsed features on the smartwatch. Notifications by nature are something that is interacted with in a very short manner. Short interactions seemed to be a common denominator for a good watch application. One user commented:

"The best smartwatch applications have simple, one-or-two button or click interactions: so any of the above (talking about different application types) work well as long as use is simple and straightforward."

Another comment around this matter:

"All applications are applicable, but developers must take into account the limitations of the watch and not just treat it like a smaller phone. Flappy Bird is watch-appropriate, Bejeweled is not. Controlling Netflix
is watch-appropriate, using the screen to watch shows is not. Viewing a daily agenda is watch-appropriate, month-view of a calendar is not. Using it as a control for taking pictures is watch-appropriate, viewing a gallery of those pictures is not."

In Apple’s design guidelines it is suggested that interactions with the Apple Watch should be measured in seconds and not minutes like on other devices[62]. This was also a comment commonly made during the usability test. In general smartwatch users did not seem very pleased with the quality of the current applications out, as seen in figure 5.4.

The survey results seem to indicate that most people expected and wanted their smartwatches to work in tandem with a smartphone. As seen in figure 5.11 over 90% of users and about 80% of non-users felt that the device should act as an extension for the smartphone. No significant difference could be found between users of different smartwatch platforms. The results for Pebbles users compared to other users differed at most by one percentage unit. In figure 5.15 the question was asked if the smartwatch should act as an independent device and not rely on a smartphone. Only 17% of the users and 28% of the non-users agreed on this, meaning that the majority were either undecided or disagreed to this statement. A survey participant went as far as to say that it was its only purpose:

"My smartwatch has one job: to make it unnecessary ever to take my phone out of my pocket. A quick textual description of 'why is my phone buzzing' catches 80 percent of the times I would reach for my phone. Being able to reply to texts, pay for things, and view my calendar would probably bump that up to 90 percent."

A notion that has been promoted, but which does not seem to match the results from this survey, is that the watch should replace the phone[56]. Two comments about this were expressed in the survey:

"I feel that smartwatches should be an extension of your phone, not a replacement. But at the same time they should be somewhat independant too."

"A smartwatch should be just that, a watch. It should not replace a phone; just as an extension."

6.2 Insights from the user study

The user study provided qualitative insight in what people thought of the platform. Before going into this matter more it is worth noting that all applications received fairly good scores, as seen in figure 5.16. A score over 68 is seen as above average, and most applications float around an adjective rating between good and excellent[42]. Several participants commented that these types of applications (the
ones that were tested) were fit for the smartwatch. They felt that one thing that the applications had in common was a simple and short flow of interactions. However there were a few common concerns stated by the users. One concern expressed was about all the hidden interfaces present on the platform, meaning that there are controls and features available that require knowledge of touch gestures, audio commands, or hardware to operate. This issue has also been brought up by interaction designers [57].

For instance the force touch gesture confused many users and they were not sure when it was applicable. This was mostly addressed during the SL Watch test, which might suggest why its rating is a bit lower than the rest. Several users pointed out that they would imagine that this would be of no issue when they have gotten used to these gestures; just like the way we have grown accustomed to interacting with touchscreens on smartphones.

Another gesture that got users a bit lost was how to use the digital crown. Several users felt a bit uncertain on how to navigate the OS using this, how to get between the clock face and applications etc. Most users seemed to prefer the crown to scroll content, as seen in figure 5.17 on question 2. One user pointed out though that he/she preferred to use the touchscreen to navigate, as this was a modality the user was used to. As seen in figure 5.17 on question 6 the touchscreen was generally seen as a good modality for the smartwatch. But even more so was the voice input seen as good way to interact with the watch, see question 5 in figure 5.17.

During the test users were asked to do two things using voice input: dictation for message replies and to open the Memory application. One user pointed out that using the voice instead of your hands and fingers enabled hands-free interaction, which was really nice, compared to using up both arms. A participant felt that the voice recognition was a bit slow though, and would prefer to receive visual feedback instantly instead of after a small delay. Another user noted that he/she probably would not use voice commands when in town or on a train.

To use the Apple Watch as a remote was met with good praise as seen in question 3 in figure 5.17. Many testers were interested in how this functionality could be expanded, and potentially be used for controlling smart homes.

On the 4th question in figure 5.17 it can be seen that actionable notifications was strongly agreed upon to be a useful feature for the smartwatch. This goes in accordance with what was discussed in the previous section. One user pointed out that this was a very good way to interact with the watch because it was quick and did not require much interaction. Another users commented that he/she did not want to fiddle too much with the screen, and that notifications or applications that were quick to use were ideal for this reason. Actionable notifications were used in the Museum application, when a user got near a painting it triggered an actionable notification that could start an audio guide. To use a smartwatch application for such a task, a context aware application, was also agreed upon to be of good use for the platform, as seen in question 10 in figure 5.17. Figure 5.14 shows a bit more suspicion towards location based applications though, with only about 60% agreeing to it being a good service for the watch, while about 26% were undecided.
and the rest disagreeing.

A question that received split opinions was the one about whether they would imagine it to be comfortable to use the watch for long sessions. One user was very certain that he/she would enjoy using it for long times. One user was of the exact the opposite opinion. The general idea though was that they would not want to interact it for too long, although some imagine that it would not be uncomfortable to use it under long times.

6.3 Field study

This sections contains a personal reflection of the experiences of using an Apple Watch as a tool for international air travel. As mentioned in chapter 3.4 smartwatches are promoted to be of good use when traveling by airplane. Applications can offer the functionality of having flight information at a quick glance and carry the boarding pass on the wrist. For this test the British Airways application was used.

Before flying

Before the trip online check-in was done using the iPhone application. Checking in provided a boarding pass that could be accessed using Apple’s platform Passbook. This boarding pass automatically synchronized to the Apple Watch and could be accessed on the wrist, as seen in figure 6.1. So, after the online check-in a boarding pass had been pushed electronically to both the iPhone and the Apple Watch.

On the Apple Watch application a quick overview of the flight could be seen, as seen in figure 6.1. At the time (before the flight) I was not in need for more information than departure time and from which airport and terminal my flight would depart from.

Figure 6.1: From left to right: bar code for the boarding pass, boarding pass and British Airways flight application.
CHAPTER 6. DISCUSSION

At the airport
After the baggage was dropped off at the drop-off counter I proceeded to the security check. This was the first instance where I was asked to show the boarding pass and passport. When I showed the boarding pass on my wrist to the staff I was given a doubtful look and was told that they were not sure if the airline I was traveling with would support this technology. I proceeded to pick up my smartphone and showed the boarding pass on it instead, and the staff let me proceed. Boarding passes on a smartphone is a well established technology and most flight companies supports it these days. Next up was the immigration counter where I was asked to provide my passport and boarding pass over the counter. Usually most people hand over these two documents in one go, but because my boarding pass was attached to my wrist this was not possible. So, instead of one action of handing over the passport and boarding pass over the counter. Usually most people hand over these two documents in one go, but because my boarding pass was attached to my wrist this was not possible. So, instead of one action of handing over the passport and boarding pass I was forced into two actions. Firstly I handed over the passport and then secondly turned around my arm awkwardly and displayed my boarding pass on my wrist to the staff.

Boarding
Having passed the security gate I found myself looking for what gate I should proceed to. This seemed like a golden opportunity for the watch to shine. Unfortunately this was not the case, at least not with the version of the application I was using. The application still only showed the departure time and from which terminal the flight departed from, as shown in the right image in figure 6.1. In my mind I was thinking that maybe the application would be updated if I had access to the Internet. As I was on international ground I did not have Internet access through my mobile. This forced me to look for a Wi-Fi spot. Getting on the Wi-Fi did not update the application though, and I was left disappointed and proceeded to check the electronic flight timetables to find the desired information.

At the boarding gate I was faced with the same dilemma I faced back at the immigration counter. Instead of handing over the boarding pass and passport to the staff in one go I was forced to do it in two steps, making the process take double the time. But that was not the only problem. The scanners used were too thin, and could not fit a wrist, see figure 6.2. If the scanners would have allowed for this type of boarding passes I could imagine the flow going faster than conventional methods. But now I ended up using my phone instead to enable the scanner to read my boarding pass. On a positive note though; the Japanese staff were fascinated by the new technology and even the security guards came to have a look.

Last thoughts
On the transit flight I proceeded to use my smartphone as a boarding pass instead. The process seemed much more streamlined for such a device. Maybe one day the process will be more streamlined for devices such as smartwatches, but from my experience, and from several others[58], it does not currently seem to be the case.
6.4 APPLICATION DEVELOPMENT

Figure 6.2: A typical scanner used for scanning bar codes on boarding passes. The one used during the field study did not fit a wrist and watch.

The ability to provide information is still a good idea and I believe has a lot of potential. The application I used unfortunately did not unveil the full potential for such a device, and only showed information for departure time. One problem that should be thought of though, is the very possible scenario that the user may not have access to the Internet. Application developers need to find out good ways to present data with these potential scenarios.

6.4 Application development

Before getting an actual device to develop for all testing was done in a simulator, with a computer-mouse as a clicking device for the screen. One big difference when going from a simulator to an actual device was that the screen was much smaller than one might have imagined. Running the applications on the simulator gave a notion of the screen to be quite big and by using a computer-mouse to simulate touch events instead of the finger created a false sense of what types of UI elements were easy to interact with. Another thing not taken into account on a simulator
CHAPTER 6. DISCUSSION

was the that the finger would take up space on the screen. One example of this was very apparent in the first version of the Memory game. When clicking on one of the buttons they quickly fade out and fade in to indicate that it was clicked. In the first version of the game the buttons were too small to see this visual queue since the finger would obstruct the whole field of vision for the button. Obviously the button would have to be increased in size to prevent this, which resulted in a graphical change.

When trying out the SL Watch application for the first time on a real device a problem, that was never a problem on the simulator, was revealed. The favorite button was too small to press with accuracy. The size of the button in the first version took up less than one fifth of the screen’s width which worked fine when clicked on with a mouse on a simulator but was in no way wide enough for real world device. Apple recommends no more than three UI elements in a row, so the clickable area for the favorite icon was increased to cover up about one third of the width, meaning about 6 mm of physical width. This seems to be a good minimum size for good consistent behavior, and also reflects the size suggested in Ashbrook, Lyons and Starner’s study [59]. The user tests also indicated that this size seems to be a good fit, as no one really made an erroneous button presses as explained in 6.1.

Limitations

As mentioned earlier in section 2.4.3 native applications are not yet supported on the platform. One big issue this leads to is the fact that user interaction do not happen as smooth as one might be used to on for example a smartphone. Every time a button is pressed on the watch the event is sent over from the watch to the phone via Bluetooth, processed on the phone and sent back to the watch. This creates a small, but in many cases, noticeable delay. For instance a button press during the Memory game would not always be notified before a second press, meaning the second press would be missed (if pressed too quickly). Of course this delay can be noticed in other scenarios as well. Taking the Memory game for instance again. When showing the memory sequence (flashing of buttons) on the watch the phone triggers a timer and sends each element of the sequence on a set time interval. But sometimes the signal gets jammed and one can notice a small inconsistency in the timing of the sequence. This is probably one of the reasons that Apple Watch do not currently have any support for programmable animations. Animations are forced to be done by a predefined image sequence to work. This might be changed when native applications are released though. Another limitation in the current WatchKit is that developers have no access to the sensors on the watch, such as the pulse-meter and accelerometer. Not even the digital crown is accessible and is predefined to just scroll content on the screen for 3rd party applications at the moment. This will probably also be changed with updates, but for now developers are limited to using buttons and dictation as the main input.
Chapter 7

Conclusion

7.1 Summary

Through a field study, online survey study, user testing and through application development this work has shown that the smartwatch can be a useful device. The platform is expected to work in tandem with a smartphone and not as an independent device that will replace the smartphone. Though in its early days there seem to be a few common key points for what a useful application on a smartwatch should abide by. They should provide quick and glanceable information, taking into consideration the small screen. Applications should not involve complex interactions, they should be kept short and simple. Notifications seem to be an area which fits the smartwatch very well and will probably be an area in which many more developers will put focus on. There also seem to be a lot of potential for the smartwatch, such as payment and boarding airplanes, but the technology around it is not fully implemented and is limited in its availability. Maybe this area will be further developed if the smartwatch market grows, which it is predicted to do[60].

7.1.1 Further studies

As mentioned earlier the smartwatch is great for quick glances and quick interactions that do not involve complex program-flows. Possibly artificial intelligence could further improve this field. In the Message application quick replies are suggested based on the input gotten from the received message. This is, as the usability test suggests, a very neat function. AI can provide short cuts for interactions, meaning complex things could be made simple[61]. Android Wear uses a similar approach in their OS with their card based system. Further studies in using AI for improving smartwatch experiences could possibly be of great interest.
Appendix A

Online Survey
Smartwatch application survey

This is a survey carried out as a part of an internship project about smartwatch applications. Your answers are anonymous and your contribution is greatly appreciated.

*Required

1. **What is your gender?** *

   *Mark only one oval.*
   
   - [ ] Female
   - [ ] Male
   - [ ] Other

2. **What is your age?** *

   *Mark only one oval.*
   
   - [ ] Under 12 years old
   - [ ] 12-17 years old
   - [ ] 18-24 years old
   - [ ] 25-34 years old
   - [ ] 35-44 years old
   - [ ] 45-54 years old
   - [ ] 55-64 years old
   - [ ] 65-74 years old
   - [ ] 75 years or older

3. **What is your occupation?** *

   *Tick all that apply.*
   
   - [ ] Working full time (more than 30 hours a week)
   - [ ] Working part-time (8-30 hours a week)
   - [ ] Carer (of home, family, etc.) (full time)
   - [ ] Student (full-time)
   - [ ] Temporarily unemployed (but actively seeking work)
   - [ ] Retired
   - [ ] Other permanently unemployed (e.g. chronically sick, independent means)

4. **Are you currently using a smartwatch?** *

   *Mark only one oval.*
   
   - [ ] Yes  *Skip to question 5.*
   - [ ] No  *Skip to question 19.*
Current users

5. Which smartwatch are you currently using? *

________________________________________________________________________

6. Have you owned other smartwatches before the one you’re using now? *
   Mark only one oval.
   Yes
   No

7. If yes, please list the watches you’ve previously owned and state the reason for the change.

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

8. What do you mainly use your smartwatch for? *
   If possible, please mention the names of the applications you use.

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

9. In general, what do you feel about the quality of the applications available to your smartwatch? *
   Mark only one oval.
   Extremely Poor
   Below Average
   Average
   Above Average
   Excellent
10. What other system(s) are you comparing to in answering the previous question? *
   Tick all that apply.
   - Smartphone
   - Tablet
   - Laptop
   - Desktop computer
   - Other smartwatch system
   - Smartglasses
   - Other: ______________________________

11. Which modalities (means of communication between the device and yourself) do you find useful on a smartwatch? *
   Tick all that apply.
   - Audio output
   - Audio input
   - Visual output (screen)
   - Visual input (camera)
   - External buttons
   - Touchscreen
   - Haptic feedback (for example vibration)
   - Accelerometer
   - Pulse meter and other health related sensors
   - GPS
   - Other: ______________________________

12. Do you feel that your smartwatch lacks any essential modalities? *
    If yes, please state the modalities and elaborate on why you lack them.
    __________________________________________
    __________________________________________
    __________________________________________
    __________________________________________
    __________________________________________

13. What is your preferred method of interaction with your smartwatch? *
    Tick all that apply.
    - Touchscreen
    - Voice
    - Buttons
    - Motion gestures
    - Other: ______________________________
14. **What types of application do you feel are suitable for the smartwatch format?** *

*Tick all that apply.*

- [ ] Calendar
- [ ] E-mail
- [ ] Notifications from smartphone applications
- [ ] Instant messaging
- [ ] Social networking
- [ ] Photos
- [ ] Fitness (step counter, pulse meter etc.)
- [ ] Health (medication reminder, diabetes tracker etc.)
- [ ] Timetable for public transport
- [ ] Location based services (finding nearby restaurants)
- [ ] Navigation services
- [ ] Games
- [ ] Payment/Transaction
- [ ] Remote control (for media or other smart applications)
- [ ] Sports (results, live scores etc.)
- [ ] News
- [ ] Video
- [ ] Music
- [ ] Reminders
- [ ] Phone finder
- [ ] Weather
- [ ] Watch
- [ ] Other: __________________________________________

15. **If there are any applications you feel are left out or have any other comments about suitable smartwatch applications, please comment in the box below.**

................................................................................................................................................
................................................................................................................................................
................................................................................................................................................
................................................................................................................................................
................................................................................................................................................
................................................................................................................................................
16. Is there any type of application that you feel is missing on your smartwatch? *
   If yes, please describe what you're missing.
   __________________________________________________________
   __________________________________________________________
   __________________________________________________________
   __________________________________________________________
   __________________________________________________________

17. Are you planning on getting a new smartwatch in the future? *
   Mark only one oval.
   ☐ Yes
   ☐ No
   ☐ Maybe

18. If so, which one?
   __________________________________________________________

   Skip to question 24.

Non users

19. Will you be getting a smartwatch in the near future? *
   Mark only one oval.
   ☐ Definitely
   ☐ Very Probably
   ☐ Probably
   ☐ Possibly
   ☐ Probably Not
   ☐ Very Probably Not

20. If so, which smartwatch would you be getting?
    __________________________________________________________

21. If you owned a smartwatch, what do you think you'd mainly use it for? *
    __________________________________________________________
    __________________________________________________________
    __________________________________________________________
    __________________________________________________________
    __________________________________________________________
22. **What types of application do you feel would be suitable for the smartwatch format?**

*Tick all that apply.*

- Calendar
- E-mail
- Notifications from smartphone applications
- Instant messaging
- Social networking
- Photos
- Fitness (step counter, pulse meter etc.)
- Health (medication reminder, diabetes tracker etc.)
- Timetable for public transport
- Location based services (finding nearby restaurants)
- Navigation services
- Games
- Payment/Transaction
- Remote control (for media or other smart applications)
- Sports (results, live scores etc.)
- News
- Video
- Music
- Reminders
- Phone finder
- Weather
- Watch
- Other: ____________________________________________

23. **If there are any applications you feel are left out or have any other comments about suitable smartwatch applications, please comment in the box below.**

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Quick questions
Thank you so much for your input. Only one more page remains in this survey.
Please indicate how well you agree on the statements below.
24. **I find smartwatches useful. * Mark only one oval.**

- [ ] Strongly Agree
- [ ] Agree
- [ ] Undecided
- [ ] Disagree
- [ ] Strongly Disagree

25. **If you don't find them useful, why? Tick all that apply.**

- [ ] Poor applications
- [ ] Bad battery time
- [ ] Lack of hardware and/or sensors
- [ ] Small screen
- [ ] Other: ____________________________________________________________

26. **The smartwatch is great for quick glances (check notifications, time, calendar, weather etc.). * Mark only one oval.**

- [ ] Strongly Agree
- [ ] Agree
- [ ] Undecided
- [ ] Disagree
- [ ] Strongly Disagree

27. **The smartwatch is a good format for navigation services. * Mark only one oval.**

- [ ] Strongly Agree
- [ ] Agree
- [ ] Undecided
- [ ] Disagree
- [ ] Strongly Disagree

28. **The smartwatch is a great tool for tracking personal health (fitness records, medication scheduling etc.). * Mark only one oval.**

- [ ] Strongly Agree
- [ ] Agree
- [ ] Undecided
- [ ] Disagree
- [ ] Strongly Disagree
29. The smartwatch can be greatly utilised as a remote for media players, smart home applications etc. *
Mark only one oval.
- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree

30. The smartwatch should act as an extending device for the smartphone. *
Mark only one oval.
- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree

31. The smartwatch can be used as a good method for payment in stores. *
Mark only one oval.
- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree

32. The smartwatch is a great for communication and/or social networking. *
Mark only one oval.
- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree

33. The smartwatch is good for location based services (for example displaying information about a nearby painting while in a museum, or finding a close by pizzeria).
Mark only one oval.
- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree
34. **The smartwatch should act as an independent device and not rely on communication with a smartphone or other device.** *

*Mark only one oval.*

- [ ] Strongly Agree
- [ ] Agree
- [ ] Undecided
- [ ] Disagree
- [ ] Strongly Disagree

35. **Last comments.**

If you have any feedback or opinions about smartwatches and applications that you would like to share, please comment in the box below.

..........................................................  
..........................................................  
..........................................................  
..........................................................  
..........................................................  

---

Powered by

Google Forms
Appendix B

Consent form
Consent & Recording Release Form

I agree to participate in the study conducted and recorded by Dan Isacson for his master’s thesis study.

I understand and consent to the use and release of the recording by Dan Isacson. I understand that the information and recording is for research purposes only and that data gathered will not be used for any other purpose.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date:________

Please print your name:__________________________________________________________

Please sign your name:__________________________________________________________

Thank you!

We appreciate your participation.
Appendix C

Resource links

C.1  Google analysis of the full survey
http://tinyurl.com/survey-smartwatch

C.2  SL Watch
https://github.com/aborren/SLwatch

C.3  Museum application
https://github.com/aborren/museumPrototype

C.4  Memory game
https://github.com/aborren/MemoryGame

C.5  Flight application
https://github.com/aborren/FlightWatchCompanion
Bibliography


[5] Zolfagharifard, Ellie. 'Is this the first wearable computer? 300-year-old Chinese abacus ring was used during the Qing Dynasty to help traders' Daily Mail, Science and Tech 19 March, 2014


[10] "Almost 60% of 16-34 year olds use a phone as their primary timepiece" YouGov Survey. 5 May, 2011


[20] Spencer, Ben. 'Mobile users can’t leave their phone alone for six minutes and check it up' Daily Mail News 11 Feb, 2013

[21] Cui, Yanqing, Jan Chipchase and Fumiko Ichikawa. "A cross culture study on phone carrying and physical personalization" UI-HCII’07 Proceedings of the 2nd international conference on Usability and internationalization 2007: 483-492


BIBLIOGRAPHY

[27] D’Orazio, Dante. 'Google reveals Android Wear, an operating system for smartwatches' The Verge 18 Mar, 2014

[28] Bohn, Dieter. 'Motorola, LG announce upcoming Android Wear smartwatches' The Verge 18 Mar, 2014


[38] Human Computer Interaction. DECIDE: A Framework to Guide Evaluation, Virtual University of Pakistan


[50] Young, David G. "Beacon Monitoring in the Background and Foreground" Radius Developers Blog 13 Nov, 2013


BIBLIOGRAPHY


[59] Ashbrook, Daniel, Kent Lyons and Thad Starner An Investigation into Round Touchscreen Wristwatch Interaction Georgia Institute of Technology and Intel Research 2008


[61] Vanhemert, Kyle. 'On the Apple Watch, the Best DJ Is an AI' Wired 29 Apr, 2015
