Aiming and Guiding
Navigation with a Non-visual GPS Application

Miguel Molina

Department of Design Sciences
Faculty of Engineering
Lund University

Supervisor: Kirsten Rassmus-Gröhn

September 9, 2010
I would like to take the opportunity to thank the test persons involved in the project, for their comments and evaluations of the prototype. Their findings are invaluable for prospective work and further investigation in this matter. I am specially grateful to my supervisor Kirsten Rassmus-Gröhn for all help and commitment in the project.

Miguel Molina, September 2010
Abstract

This master’s dissertation describes a study that relates to persons with visual impairments, and wayfinding using a smartphone with touchscreen. The study consists of the development of a prototype and an evaluation of its usability while interacting with haptic and auditive feedback. The prototype was developed for the Android platform. It provides a way of scanning for points of interest while pointing at them with a mobile phone and then choosing to be guided to them. The evaluation was executed by six users, five participants with visual impairments and one sighted, in an open area. The evaluation showed that users found the application usable. The scanning approach was perceived more valuable than the guiding because of gps inaccuracy and the lack of routing built into the application.

Keywords

visual impairment, mobility, navigation, GPS, walk experiences, user interaction, Android, haptic, audio, text-to-speech
# Table of Contents

1 Introduction.................................................................................................................. 1  
  1.1 Background ............................................................................................................. 1  
  1.2 Structure .................................................................................................................. 2  
  1.3 Aim .......................................................................................................................... 2  

2 Theoretical Background.................................................................................................. 3  
  2.1 Available Technology in Handheld Devices ............................................................. 3  
  2.2 Android Operative System ....................................................................................... 6  
  2.3 Data Representation ............................................................................................... 8  
  2.4 Maps ......................................................................................................................... 12  
  2.5 Visual Impairment and Navigation .......................................................................... 14  
  2.6 User-centered Software Design .............................................................................. 15  

3 Methodology................................................................................................................... 19  
  3.1 Technical Aspects On Method ................................................................................. 19  
  3.2 Iterative Development Cycle .................................................................................... 20  
  3.3 User Group and Usability Evaluations .................................................................... 20  

4 The Prototype.................................................................................................................. 23  
  4.1 The Mobile Phone .................................................................................................... 23  
  4.2 The PointNav Application ....................................................................................... 23  
  4.3 The Haptic User Interface: HUI ................................................................................ 24  
  4.4 Interaction with HUI ............................................................................................... 25  
  4.5 Scanning or Guiding ............................................................................................... 26  
  4.6 Bearing: Scanning for POIs .................................................................................... 27  
  4.7 Guiding ..................................................................................................................... 28  
  4.8 Avatar Mode .......................................................................................................... 30  
  4.9 Visual Interface ....................................................................................................... 30  
  4.10 Settings .................................................................................................................. 32  
  4.11 Validating a GPX File ............................................................................................ 32  
  4.12 Obtaining an API Key ............................................................................................ 33  
  4.13 Signing The Application ....................................................................................... 33  
  4.14 Calibrating the Sensors ........................................................................................ 34
List of Figures

4.1 HUI, a previous version ........................................... 25
4.2 HUI, the final version ............................................. 25
4.3 Bearing and Guiding States ....................................... 26
4.4 Bearing Mode ...................................................... 27
4.5 Guiding Mode ...................................................... 28
4.6 Guiding State Machine ............................................ 29
4.7 An Avatar illustration ............................................. 30
4.8 The POIs’ List Tab ................................................ 31
4.9 Map Representation Tab ......................................... 31
5.1 Workflow .......................................................... 36
5.2 Lunds stadspark .................................................... 38
Chapter 1

Introduction

This section introduces the background, structure and aim of this master’s dissertation.

1.1 Background

In the beginning of the 90’s, mobile phones were used by a small privileged group of people since the price for the devices and traffic rates were very high at that time. As microprocessors and battery development accelerated, mobile phones became smaller, the stand-by time was extended and the prices became lower. Now a bigger market has been reached and the mobile phones are available to virtually any user.

A considerable advantage with today’s technologies is the high grade of customization and adaptation to the user’s need. Software development tries to increase a mobile phone’s usability by adding more sophisticated menus that take advantage of advanced graphics in order to present an intuitive user interface. Another appreciated feature built in handheld devices is the access to maps and location information. Mobile phones are often equipped with Global Positioning System (GPS) receivers that are used to calculate a user’s position and with that value it is possible to use a navigation application to get the route to a desired destination. In an unfamiliar environment, this information makes the difference between getting to a position with minimal support from the surroundings (within a small time frame), and whether the user needs to get more information from other sources, such as signposts, known places or other pedestrians.

Unfortunately, the mentioned development does not add usability to users with visual impairments. One can take advantage of the capabilities of enhancing the usability by adding audible feedback, but this approach is also limited since the hearing sense is used to compensate the visual impairments and therefore it should not take the whole attention from the individual. A suggestion on how to provide information in a proper way is not simple. One approach is by combining technologies in proper proportions together with the users’ suggestions and
then, by implementing several consecutive prototypes and testing these, acquire a reliable solution. This solution might be acceptable for at least a specific group of people.

This master’s dissertation is performed at Certec, Department of Design Sciences, Faculty of Engineering Lund University. The core activity of Certec is research devoted to rehabilitation technology, and the master dissertation work is carried out in the scope of the HaptiMap (HaptiMap.org, 2010) project. The project is about development of a new programming toolkit which integrates haptic, audio and visual interfaces for maps and locations services. The toolkit will provide a higher grade of adaptation by simplifying map representation used by developers when representing information by other channels than visual. One could reach groups who have visual impairments, but also those who need to put visual attention away from for example a handheld device (HaptiMap, 2010).

1.2 Structure

The structure of this dissertation starts with a short introduction (this chapter) followed by a theory section in chapter 2 with state of the art sections on mobile technology, navigation skills, usability and electronic map representation. The methodology in chapter 3 describes the test procedure and how they were carried out. The prototype developed during this project is explained in detail in chapter 4 and the user evaluation of the final prototype is described in chapter 5. Finally, the chapters Discussion, 6, Conclusions, 7 and Future Work 8 wrap up this dissertation. The Appendix contains a user scenario that was written in the beginning of the project to investigate ideas for interaction and usage.

1.3 Aim

The aim for this master’s dissertation was to iteratively develop and evaluate a GPS navigation application on the Android platform for use by persons with visual impairments. The application uses a multimodal feedback approach to let users scan for points of interests (POI’s) and to be guided to them. The evaluation of the application focused on the following aspects:

- To consider the response of users with visual impairments to the Haptic User Interface approach when using a touchscreen.
- To investigate if a simple multimodal guiding approach without routing helps users finding POIs in open areas.
- Find out if this kind of application could facilitate navigation tasks for both sighted users and users with visual impairments.
- To test some filtering algorithms when scanning for POIs.
- To explore the iterative development symbiosis between developer and end users.
Chapter 2

Theoretical Background

This chapter starts with a technical description of hardware and software, and continues with an introduction to navigation, orientation and information processes for individuals. The following section describes available technologies used to represent data, their limitations and possibilities when representing it on a handheld device and the last section is focused on the interaction between users and technology, particularly introducing the user-centred design approach method.

2.1 Available Technology in Handheld Devices

In the beginning, the primary function of a mobile phone was communication, by standard voice and Short Message Service (SMS). The technical solutions implemented in mobile phones of the present day are far more advanced in software and hardware, and contain sensors for Global Positioning System (GPS), compass and accelerometers, to mention a few.

2.1.1 GPS and A-GPS

GPS receivers provide positioning and are generally used for navigation purposes. Besides the user’s current position they also offer direction and speed data calculated from the user’s change of position. The accuracy offered by the GPS system since 1 May 2000 is up to 25 and 30 meters horizontally respective vertically, depending on several factors, e.g. atmospheric error due to refraction and absorption, the number of available satellites reached by the device, satellite clock error and multipath effects caused by buildings (Zhou et al., 2009; Calcutt et al., 2001). Fortunately, Assisted GPS (A-GPS) improves the GPS system performance in mobile phones by receiving data from a mobile phone network. The improvement is achieved since the satellite signals to these assistance servers is normally good and the GPS receiver does not need to wait a longer period for a clear signal from the satellites. In this way, accuracy is improved and power consumption on mobile phones is reduced because most of the computation is done by the assistance servers. GPS receivers’ usability is extended when combined with access to location information, for example maps, from an information source such as the
Internet. The connection can normally be established from a mobile device using, e.g., General Packet Radio Service (GPRS), Bluetooth, High-Speed Downlink Packet Access (HSDPA) or Wi-Fi (IEEE 802.11x). The main difference between the standards is availability and speed.

The statement about the need of communication with a server is true for mobile devices that rely on on-line information. Depending on the manufacturer’s implementation of the handheld device, several GPS receivers are already equipped with maps that are stored locally and also include route calculation functions. Therefore they do not need to communicate with any server in order to present information to the user. The consequence is that the user, normally, has to update the maps manually when changes are available.

GPS navigation systems are mainly used for transportation with vehicles and the available maps provided on GPS receivers do not include an extensive wealth of detail for pedestrians, i.e., park trails, national parks or other tourist destinations. Considering the limitation of information about these areas, it becomes difficult to deploy a service for purposes outside the already known urban paths. One of the reasons to the lack of information within these areas is because maps on commercial handheld devices with GPS receivers primarily aim for vehicles and transportation. However, accuracy improvements achieved by using A-GPS instead of the regular GPS increases the reliability of maps for pedestrians in urban environments.

GPS systems are not available for indoor navigation since the signal coverage is very poor or it is unavailable due to absorption in building structure elements (Benavente-Peces et al., 2009). For indoor purposes, other systems based on other technologies are being developed, e.g., Radio Frequency Identification (RFID) (Chumkamon et al., 2008), Bluetooth, Wi-Fi, Worldwide Interoperability for Microwave Access (WiMAX) or Ultrasound.

2.1.2 Sensors

Handheld devices are equipped with different types of sensor devices, although all sensors about to be mentioned might not be included in all smartphones. Among the most popular functionalities built into smartphones one can find accelerometers, magnetometers/compasses and gyroscopes.

Accelerometers

Accelerometers are normally piezoresistive sensors with the ability to measure linear acceleration, vibration and shock. Piezoresistivity occurs when a mechanical stress exerts on a semiconductor and changes its resistivity. The change in resistivity occurs when the seismic mass that is attached to a moving electrode changes its position towards a fixed electrode. This change is represented by an electrical output that is proportional to the acceleration. A common problem with piezoresistive accelerometers is their temperature dependency which can lead
to slight variations in the output and to erroneous values (this issue applies to
piezoresistive sensors in general). This is especially problematic with handheld
devices because it is difficult to compensate for all environments it is exposed to.
Another well known problem with accelerometers in handheld devices is their
sensitivity to small variations in motion. A user’s normal handshaking results in
an undesired acceleration leading to jitter. The jitter can be filtered, but the filtering
process adds a delay as a result, leading to less accurate values. The acceleration
values are obtained through electronic integrators, for which velocity can be
achieved by the first integration and position by the second integration (Johnson,
1997; National Instruments, 2010). There are other transducers that can be used
as accelerometers and a detailed description for each one is outside the scope of
this dissertation.

Accelerometer can only measure roll and pitch but they cannot detect the yaw
(rotations around its own axis). Accelerometers can be used together with mag-
netometers in order to offer an orientation sensor.

Orientation Sensor
Magnetometers are used to measure the strength of a magnetic field by detecting
the response of the forces they exert on other materials. Magnetometers in smart-
phones are commonly used to detect the Earth’s magnetic field or magnetosphere.
These sensors combined with accelerometers, as mentioned before, can be used
to provide orientation, such as a digital compass in smartphones (GLOBALSPEC,
2010).

As accelerometers measure gravitational force on the handheld device, they deter-
mine the inclination of the sensor with respect to a reference vector in a reference
coordinate system. Accelerometers use the gravitational force as reference vec-
tor and with the Earth’s coordinate system, the positive z-axis points away from
the centre of the Earth. This inclination determination is used in levelling mag-
netometers for use in compasses. These values can also be used to determine
rotation on handheld devices.

Gyroscopes
Gyroscopes are constructed to perceive angular position or orientation changes
by measuring the alteration of a given direction on a two or three axis coordinate
system. These sensors offer more accurate values in terms of movement regarding
orientation changes and can become a replacement to magnetic compasses when
applicable. The gyroscopes’ stability compared with accelerometers’ derives from
the effect known as precession, i.e. any forces applied to a gyroscope’s axle will
only change its direction on its own centre. Gyroscopes offer a better response
performance in rotational motion. That is because only a single integration is
needed to obtain the displacement, as explained by Pycke (2010), compared with
accelerometers that need two integrations. The gyroscope can also be used in
conjunction with an accelerometer in order to offer six axis motion sensing, that is: up/down, left/right, forward/backwards, roll, pitch and yaw rotations.

2.1.3 Feedback

Feedback in this context refers to the output generated by a system, the handheld device, notifying the user about changes that have occurred. The interpretation depends on the conceptual model the user has about the system. The project exploits the handheld device’s output for both auditive and haptic feedback.

Haptics

Haptic technology refers to tactile feedback that combines vibrations, forces and motions. The word haptic derives from the Greek word *haptikos* which means to touch. In a handheld device, the vibration feedback is provided by an unbalanced motor. An unbalanced motor is a device that holds a weight that is irregularly attached on the shaft, i.e. the weight is not placed on the midpoint of the shaft. When power is supplied to the motor, the motor spins the unbalanced weight rapidly and centrifugal forces are generated. These forces displace the motor inside the handheld device and vibration occurs.

Text-To-Speech

Text-to-Speech (TTS), also known as speech synthesis, is an implementation in software or hardware that enables a computer system to create human speech. A TTS enabled application sends a text string to the TTS engine, which combines and concatenates pre-recorded pieces of speech from a database and translates them into speech using a waveform generator. The quality of the speech depends on the system’s synthesizer which is the one that converts the original string into sound. The synthesizer’s performance is measured in how well the speech sounds and if it is easy to comprehend. TTS can be used to read text displayed on a regular computer and also to read messages on handheld devices such as SMS. This tool is invaluable for users with visual impairments and it is also in GPS devices providing directions and addresses to drivers, for example.

2.2 Android Operative System

Android is an operating system developed by Google that offers a platform built on an open source framework for mobile devices. It uses the open source Linux kernel 2.6 which handles core services, including hardware drivers, security, network, power, memory and process management. The Android Runtime runs above the Linux kernel and together with the access to various libraries it provides the basis for the application framework. The Android API libraries included in the Software Development Kit (SDK) provides the tools needed to create native applications. Included in the APIs one can find e.g. hardware access, location-based services, map-based activities. The Android platform offers the opportunity
to create applications with no licensing, distribution or development fees. These particular properties makes it attractive on the mobile device market because developers have more flexibility and ability in making changes that are not protected by copyrights. A problem that has been seen in other operative systems is that these systems prioritize native applications over the ones written by third parties. This limitation is not an issue in Android since both native and third party applications are developed with the same APIs and execute on the same Android Runtime, consisting of Core Libraries and Dalvik Virtual Machine.

The programming language is Java, which is an object oriented application framework including its own core libraries, and the applications are executed by Android’s virtual machine: Dalvik. It means that the source code is portable and can run on other architectures, e.g. in ARM or x86. The growth of Android as a popular operative system among smartphones is more noticeable today in more analytic reports. According to one of Canalys’ studies (Canalys, 2010), Android is not the biggest OS among smartphones, but it is the one that is continuously taking more market share from other brands, such as Symbian and Windows Mobile (Gartner, Inc., 2010). Android shows in the second quarter of 2010 a market share of 17.2 % compared to 1.8 % in the same period 2009, putting itself into third place of most popular OS worldwide (preceded by Symbian respective RIM and followed by iOS on forth place). According to Google’s CEO, Eric Schmidt (MacWorld, 2010), 200 000 new Android based smartphones are activated every day as late as in August 2010. It is clear that the importance of the Android platform in the smartphone market will increase and that the possibility to develop applications that are not restricted by the operating system will become more important.

Text-to-Speech in Android

The TTS-engine in Android was introduced in version 1.6. It is limited to the following supported languages: French, German, Italian, Spanish and English (American and British accents). It is important to choose the right language because words can be pronounced differently depending on the predefined language. The TTS engine provided by Android is easy to use through the TTS API. The simplicity of the engine does however limit the grade of customization, for example, it does not allow the user to change voice gender as in other more advanced TTS engines. As for the writing moment only one voice exists.

Signing Android Applications with RSA Signatures

Digital signatures are normally used when distributing software and when a vendor wants to prove the authenticity of his/her identity. The RSA algorithm can used for both signing and encryption (Gollmann, 2006). The algorithm is based on the mathematical problem of factoring and the signing scheme consists of three algorithms: key generation, signature and verification.

The RSA signature is about picking two prime numbers $p, q$ such that:
n = p \cdot q \text{ and } \phi(n) = (p - 1)(q - 1)

The public key exponent \(e\) is chosen such that:
\[
gdc(e, \phi(n)) = 1 \text{ and } e \cdot d \equiv 1 \mod \phi(n)
\]

And the private signature key \(d\) is found such that:
\[
e \cdot d \equiv 1 \mod \phi(n)
\]

The public keys consist of the modulus \(n\) and the exponent \(e\). The private key is the exponent \(d\) together with \(\phi(n), p, q\).

The document \(m\) to be signed stays an integer \(h(m)\) after applying a suitable hash function \(h(x)\). The signing procedure between two users, Alice and Bob is accomplished when Alice forms the signature \(s = h(m)^d \mod (n)\). When Bob checks the signature, he needs Alice's verification key pair \((n, e)\) and checks \(s^e \equiv h(m) \mod n\).

If the signature is correct, the result is true because:
\[
s^e = h(m)^{ed} = h(m) \mod n
\]

2.3 Data Representation

The geographic data stored in database servers can be formatted in various ways depending on the representation models they are using and the purpose of the data collection in the first place. A common way of describing this data is necessary and various formats have been developed, e.g. GPX, GML and KLM. The data representation and formatting depends on the application that needs to access the data and the aim of the information fetching. The mentioned formats GPX, GML and KLM provide the geographic data needed for the project. They have been used as standards within the limits of this project and will be broadly described below. The existence and description of other standards is outside the scope of this dissertation. Common for these standards is that they are based on the XML (Extensive Markup Language) document language. XML is basically a specification for a collection of nested data elements indicated by tags for documents (the meaning of the substrings within the document) and it offers an easy approach in simplicity when used over the Internet. This is specially useful since geographic information is stored in databases and the data exchange between, in this case, a hand held device and the database service is commonly done over an Internet connection as a web service.

WGS84

WGS84 is the reference coordinate system representing the Earth as an ellipsoid. It is used by the Global Positioning System (GPS) and it provides a high level of accuracy in global maps. Location representation uses geographical coordinates in order to place a point on the earth. These coordinates are latitude (\(\varphi\)) and longitude (\(\lambda\)), representing a location north \((0^\circ \text{ to } +90^\circ)\) to south \((0^\circ \text{ to } -90^\circ)\).
of the equator respective east to west, beginning from zero degrees at the Prime Meridian, passing through Greenwich, to +180° eastward and −180° westward.

Calculating latitude and longitude

The latitude and longitude values are based on the hypothesis that a degree is subdivided into 60 minutes and each minute is divided in 60 seconds. The representation of a location can be typed as a sexagesimal notation 55°42′21″, 13°11′37″. With higher precision expressing seconds as a decimal fraction of a minute 55°42.3465′, 13°11.61942′ or as a decimal fraction consolidating both minutes and seconds 55.705775, 13.193657. The south and the west are represented with a negative sign.

The conversion between the formats is just a conversion of units described as follows. The degrees stays always the same but the minutes respective seconds can be expressed in two ways. When transforming from the sexagesimal format i.e. "Degrees Minute Second" to "Degrees Minutes.decimal fraction", the decimal fraction is calculated with seconds/60 and by adding the value to the minutes: 55°42′21″ becomes 55°42.3465′. The conversion from "Degrees Minutes.decimal fraction" to "Degrees.Decimal degrees" is done by calculating (Minutes.decimal fraction/60) and adding that value after the degrees: 55°42.3465′ becomes 55.705775.

Defining a Point in Space

From a start point in a geographic coordinate system (lat₁, lon₁), we can calculate a destination point (lat₂, lon₂), that is the latitude and longitude, if we know the distance to the destination and the initial bearing with the following equations:

$$\begin{align*}
\text{lat}_2 &= \text{asin}(\sin(\text{lat}_1) \cdot \cos(d/R) + \cos(\text{lat}_1) \cdot \sin(d/R) \cdot \cos(\theta)) \\
\text{lon}_2 &= \text{lon}_1 + \text{atan2}(\sin(\theta) \cdot \sin(d/R) \cdot \cos(\text{lat}_1), \cos(d/R) - \sin(\text{lat}_1) \cdot \sin(\text{lat}_2))
\end{align*}$$

Where,

- lat₁, lon₁: The latitude and longitude of the start position
- lat₂, lon₂: The latitude and longitude of the destination point
- d: The distance to the destination point
- R: The Earth’s radius. Equatorial radius is about 6378 km and the polar radius is 6357 km. A general value for the Earth’s mean radius is 6371 km (currently used here)
- \(\theta\): Bearing

Source: Ed Williams (2010)

The atan2 function takes two arguments, atan2(y, x) and it is used to compute the arc tangent with the difference that it handles x = 0 and returns values in all four quadrants, −\(n\), +\(n\). For example, for the regular arctan, arctan(1, 1) = \(\pi/4\) and
\[ \arctan(-1, -1) = \frac{\pi}{4}, \text{ which gives the same angle value but what we mean is } \\\]
\[ \text{atan2}(1, 1) = \frac{\pi}{4} (\text{or } 45^\circ) \text{ and } \text{atan2}(-1, -1) = -3\frac{\pi}{4} (\text{or } -135^\circ). \]

The above equations (2.1 and 2.2) assume that the earth is spherical and they ignore ellipsoidal effects. The error in this case is between 0.55% to 0.3% depending on latitude and direction (higher error across the equator). If higher accuracy is desired, the Vincenty formulae (Vincenty, 1975) of geodesics on the ellipsoid can be used. The direct formula provides results accurate to within 0.000015” or 0.5 mm. For the purposes of this project, the equations above offered enough accuracy for the prototype.

Calculating Distance Between Two Points

As mentioned in previous calculations, the Vincenty formulae can also be used to calculate the distance between two points. The calculation is performed with the inverse formula which offers the same grade of accuracy as with the direct formula. The calculations are iterative and the formula is developed for efficient programming. The inverse formula is however already implemented in Android’s location class and was not necessary to implement.

2.3.1 GML

The Geography Markup Language (GML) is the XML data schema defined by the Open Geospatial Consortium (OGC) which is used to formulate and represent geographical content. The purpose of GML is for modelling, transport and storage of geographic information (Cox et al., 2004), that is to provide a common language/vocabulary for expressing geographical information with attributes. GML does not contain any information about how to display the data, such as line thickness or colour codes. The standard describes features, i.e. entities that describe the real world such as rivers, buildings, roads, etc. These features are described by properties of different value types, e.g. integers, boolean and string. The schema offers a high level of flexibility and complexity due to the amount of data it could represent. The complexity is inevitable since the features can represent 2D or 3D objects. A more detailed description of this standard is outside the scope of this dissertation and more information can be found at http://www.opengeospatial.org/standards/gml.

2.3.2 KML

Keyhole Markup Language (KML) is an XML data schema that is used to visualize geographic information, on the contrary from GML, which only represents the content. KML is tailored for use with Google Earth user interface and it has been adopted as an OGC implementation standard. The geographic visualization information includes also annotation on maps and images and that makes the KML valuable as a complement to the existing standard in GML. The format is broadly used for Google Maps as a standard format.
KML sample: a placemark

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>New York City</name>
    <description>New York City</description>
    <Point>
      <coordinates>-74.006393,40.714172,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

Description of a few tags:

**Placemark** Commonly used to mark a position on the Earth’s surface with a yellow pushpin

**name** The label for the Placemark

**description** Information that appears in the "balloon" attached to the Placemark

**Point** When defined in a Placemark it defines the position of the Placemark’s name and icon.

**coordinates** A tuple consisting of longitude, latitude and altitude (if available or desirable). Values expressed in degrees.

Source: Google Code (2010)

### 2.3.3 GPX

GPX (GPS eXchange Format) is an open light-weight XML data format which is broadly used to represent waypoints, routes and tracks. The format is suitable for GPS data exchange between applications and it is broadly used on the Internet. Since the file format has been standardized, it provides the advantage of being easy to transform into other file formats in order to offer support to other handheld devices or software. The current version of the schema is GPX 1.1 and was released on August 9, 2004. Waypoints are of particular interest since they represent physical places on the earth and are closely related to landmarks. Examples of landmarks are things such as churches, railway stations, mountains and other buildings. The convention in GPX is that all the coordinates are relative to the WGS84 datum (reference ellipsoid) and all measurements are in metric units.

GPX sample: wpt

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.0"
    creator="GPSBabel - http://www.gpsbabel.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```
Maps are representations of data in a readable way to users where elements are presented in a visual format. Information about e.g. roads and paths around the globe are represented by maps with various levels of detail. There exists many kinds of maps, e.g. road maps with transportation routes and cities, geological maps showing the types of rocks in a given area and topographic maps showing rivers and elevations. The kind of map users choose depends on their intention and in this project, electronic maps for urban environments are of particular interest. The main reason is the flexibility and portability available through digital technology and because of the liberty in fetching and manipulating data suitable for explicit demands. The majority of vendors who develop devices with GPS receivers are mainly focused on road maps suitable for vehicles, and
urban environment maps preferable for pedestrians. Mostly, the level of detail in these maps is enough to, for example, display a route map to the user or to give directions with a minor error that can be corrected by the user. Vendors can also profit with these devices by selling map updates or map upgrades, depending on the user’s need. This kind of service is needed since cities and roads tend to change dynamically and while routing there is a risk to end up in a road that no longer exists.

### 2.4.1 Google Maps

A source for location information that has acquired more importance lately is Google’s application and technology: Google Maps. It offers a large amount of data and it is possible to take advantage of its potential into other applications and/or websites through their API, which was released in June 2005 (Official Google Blog, 2010; Wikipedia, 2010). Google Maps offers high-resolution satellite images, route planner, street maps, etc. and it is free to use. The current API version, as for August 2010, is Version 3 and it can be used for both desktop and mobile devices. Version 3 has improved performance on mobile browsers and the need of API keys (see section 4.12) when embedding in browsers is no longer required. Google Maps facilitates creating applications for mobile platforms such as Android and iPhone and it provides different ways in displaying the information on the smartphone depending on hardware and software capabilities. One of the common requirements regarding Google Maps is that the device depends upon continuous information fetching and needs a proper connection to the Internet in order to refresh the maps. Regarding location services one can mention the possibility to pinpoint a mobile device by using GPS, WiFi-based services or by triangulation. Thanks to this feature, even if the smartphone is not equipped with a GPS receiver, there will still be a probability to localize the user’s current position and for example, routing directions can be found.

### 2.4.2 OpenStreetMap

OpenStreetMap (OSM) is an open content project that aims for free use of geospatial data, i.e. the share, creation and use of geospatial data. The project relies on donations from people for its funding but also from partners who help with resources, such as server hosting and access to maps (Yahoo! Aerial Imagery). OSM encourage people to get involved and contribute with geospatial data in order to extend the maps. The information is normally collected with GPS receivers by individuals who upload the data to OSM’s database on the projects website, www.openstreetmap.org or by using other ways of mapping techniques such as photo and audio mapping. When a GPS receiver is used, the tracks must be in GPX format for uploading. The GPX files should contain only traces and not waypoints, as showed in example 2.3.3, because of lack of function implementation. The uploaded data can then be modified with the on-line editors available on the website in order to make the data available to public. The philosophy behind the project reminds of Wikipedia, which obviously seems to be one of the project’s inspiration, as buttons like Edit and History are available. Photo
mapping is about collecting images with a digital camera and then correlate them with a user’s GPS log. Audio mapping is more about recording voice clips about the areas the user is passing by and then correlate them with GPS data. Useful when describing, e.g. street names, street types, POIs or access restrictions which cannot be seen on GPS logs. Besides the benefit in the concept of open and free content licensing, one of the major advantages with OSM is the different levels of detail in maps that are available. Since the members of the project are not only people who drive vehicles, but also use bikes and walk, the nuance on the maps is slightly different from others, such as on Google Maps (see section 2.4.1). This approach is a direct consequence of the type of data the members contribute with. The urban environment is then combined with other paths that are not usually available on commercial maps. Other parts of the cities, such as small paths in parks and open places, are suddenly available and plotted on maps.

Credibility is a topic that can arise regarding the trustworthiness of the information on OSM and the way it can be explained is by accepting the idea of contribution and expectation from users in the same way as it has been accomplished with the Wikipedia project.

2.5 People with Visual Impairments and Navigation

According to the World Health Organization (WHO) the number of people with visual impairments is about 314 million people worldwide. 82% of them are age related and the number increases as population grows (World Health Organization, 2010). Medical intervention is not currently available for age-related macular degeneration, which is the main cause of visual impairments (Margrain, 2000) for the increasing older population.

Navigation is a complex task that involves cognitive processes which includes acquiring spatial information and movement from one place to another. The action involves planning and understanding of the surrounding environment in order to find destinations. Normally, people fetch this information when they are planning before they arrive to the desired destination. The process involves not only an internal representation of the investigated environment but also the ability to recognize destinations while orientating.

A sighted person gathers information about an unknown environment through the visual channel, which gives him/her the possibility to create a cognitive mapping of spaces in advance or at the current position (Lahav et al., 2008). The main concern for a person with visual impairments is about how to familiarize themselves in the present environment and how to reach specific targets depending on their positions. Orientation in an unknown environment is very difficult and often an unpleasant and unsafe experience for a person with visual impairments (D’Atri et al., 2007; Lahav et al., 2008).
Positioning and adaptation to a new environment requires that a person with visual impairment collects abundant information about the surroundings through the tactile and hearing senses. Because of the complexity of this task, the person in many cases needs to be guided. Research (Passini and Proulx, 1988) shows that people with visual impairments can reach a level of freedom by acquiring spatial mapping and orientation skills at a perceptual and conceptual level (Lahav et al., 2008). The perceptual level refers to the compensation for the lack of vision and this can be reached by using tactile, smell and hearing senses. The conceptual level refers to the spatial problems and how they are solved using efficient tools providing adequate mapping. The development of these tools branches into passive and active aids to the user. The passive aid represents the information that can be acquired about a location before the user arrives to the destination, such as tactile maps, models and narrations. Active aids refer to devices that provide information about the current location to the user where they are located, such as GPS equipped devices. Passive aids like tactile maps are limited in dimension and may offer poor resolution of the spatial information. These maps demand an extra user effort since the user has to remember many details in advance. Active aid devices have limitations in accuracy and missing details about places that have not been explored before.

Limitations in Maps for Persons with Visual Impairments

The representation of geospatial data on smartphones is expressed as maps in a visual format. As mentioned in section 2.4, maps offer various levels of detail that are available as advanced features in smartphones. These features are displayed through different layers, for example in Google maps it is possible to display Wikipedia and Satellite View layers. The bottom line of the mentioned features is that they are provided as visual aid and can not be expressed through other modalities, such as tactile and/or auditory. The complexity expressed in these maps makes them difficult to become available in a non visual environment, leaving out groups with total or partial absence of visual ability.

2.6 User-centered Software Design

Problem domain

The request for innovation of new services and tools for interaction between computers and users increases worldwide. Developers are often put in a stressed situation since economic profit and a constant competition for reaching new user groups is desirable. In the process of acquiring new technologies that will increase productivity, developers prioritize learning these new features, letting the small time frame that was devoted to user testing become even smaller. This behaviour leads to poor interactivity with the users resulting in a less friendly user interface.

The main objective when developing a new computer based aid, such as an application or a device, is to offer a way of making things more comfortable for the user.
These tools are intended to increase efficiency in every day’s tasks by offering reliable and optional solutions that will help the user to make better decisions or just by helping him/her to focus on other tasks simultaneously. Unfortunately, this is not always the case. On the contrary, these tools decrease efficiency through frustration and disengagement leading to stressed situations to the user. Normally, this is because the development of these tools is focused on the concept of offering more complexity in the featured design and by ignoring the users’ prerequisites in this matter. This case is very common since developers struggle to offer the latest available technology and it often means that they have to step over a time consuming knowledge threshold. Since time is limited in product development, efficiency in usability is not the first priority and a in some cases it could lead to a poor user interface.

### Human Computer Interaction Research

Interaction between computers and humans is one of the main concerns for Human-computer interaction (HCI) studies. This discipline looks deeper into the total satisfaction of individuals while using a user interface implemented in software and/or hardware. According to Gulliksen (Gulliksen and Göransson, 2002), 50% of the software development is about user interface but only 1% of the tests that are performed to ensure the quality of the resulting product are combined with user interaction. The reason for leaving out user tests depends mostly on the amount of prolonged time it would take to finish the projects. Furthermore, the testing phase that interacts with users is normally done at the end of the projects which also implies that when problems are encountered, the more difficult it is to rectify them. The cost of changing details in a product grows with the life cycle of, for example, software development. One suggestion to solve this problem is to understand the importance of product development together with customers’ feedback. This could be achieved by using a user-centered design. Furthermore, it is important to understand the difference between what the user originally think that they want and what they might need, which does not have to mean the same thing. A user’s need can not be measured in absolute terms and an agreement about things that can be covered within a product or project must be stipulated. A reference group, whom the designers can discuss design solutions with and ask for clarifications, should consist of different categories that represent the target groups the product is aimed for.

### User-centered design

User-centered design is about including users in the design process of a product. It involves the use of three principles (Rassmus-Gröhn, 2008):

**Early focus on users and tasks**  Involvement of users, both domain and end users. Domain users are those who are involved in a project but do not represent the end users.

**Empirical measurements**  Observations about the end user’s reactions to prototypes.
Iterative design  Involves the process of incremental product development. It includes a cycle of prototyping, evaluation and analysis until a product meets the requirements aspired. It also means that the users are involved in the evaluation phase giving feedback about the utilization.

Usability

According to Norman (Norman, 2002), there are three attributes that lead to a successful development of a user interface: visibility, mapping and feedback. After visibility, a natural mapping creates the bridge between the user expectations and the results while the feedback should inform the user about the action performed and its results. But, what is usability? The International Organization for Standardization (ISO), who develops and publishes international standards, presents in the “Guidance on usability” (International Organization for Standardization, 1998) some definitions, as some are quoted below:

3.1 usability  Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

3.2 effectiveness  Accuracy and completeness with which users achieve specified goals.

3.3 efficiency  Resources expended in relation to the accuracy and completeness with which users achieve goals.

3.4 satisfaction  Freedom from discomfort, and positive attitudes towards the use of the product.

3.5 context of use  Users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used.

As indicated above, to deliver a good design, in matter of properties of usability, it is important that developers consider the importance of listening to the users, who will use the products, and let them review them while developing. Their feedback about functionality will increase the product’s value when it reaches the market. The developers should take consideration to the recommendations defined in the ISO standard about effectiveness, efficiency and satisfaction in order to deliver an acceptable product to the end user.
The aim of this chapter is to describe and introduce the methods used when developing the prototype in this project. Technical aspects about the programming environment are described as well as methods used when involving test participants from the target user group.

3.1 Technical Aspects On Method

One of the purposes with the project was to explore the Android OS platform. The development environment recommended by Google is Eclipse (version 3.4 or 3.5) and since this is the environment the author is familiar with, it became the author’s development platform. The development language used in the project was Java, JDK 6. The additional software needed to complete the working environment was the Android SDK and the Android Development Tool (ADT) Plugin for Eclipse. The ADT plugin extends functionality to Eclipse such as debugging, building and signing applications.

In order to take advantage of the functionality available on the developer phone’s supported firmware (version 1.6), the Google API 1.6 (API Level 4) was selected. The reason for this choice was because the standard Android library does not include support to Google Maps and the library is only included in the Google APIs Add-On for the Android SDK.

As a complement to the existing environment, the open source, Subversion Version Control System Server (SVN) was installed. The current version available is 1.6. The SVN can keep track of changes of the source code which makes it invaluable when debugging. In an iterative development environment, the programmer needs a way to review previous changes, compare them with the new ones and eventually find errors introduced with the latest version of the existing code. The SVN is also a collaboration tool that allows various people to modify the source code from different computers in a TCP/IP network. This feature was very useful because the author could share the project resources with other users in an iterative way (Collins-Sussman et al., 2009). The SVN client is well integrated in the Eclipse environment as additional software.
The Desktop Operating Systems used in the project were Windows XP with Service Pack 3 and/or Windows 7 Enterprise. There was no special reason for these choices other than they were available on the developer’s Desktop PCs on different workplaces.

3.2 Iterative Development Cycle

In order to improve software quality and to achieve a higher rate of learning, an iterative development cycle was used. Basically, within an iterative approach, the developer implements small pieces of software that are evaluated and corrected in small iterations. The functionality implemented is reviewed in every iteration and according to the feedback, debugging and correcting of the code takes place. New functionality is then integrated and the process starts all over again. One of the main advantages of this approach is that someone else, who is not the developer (the end user perhaps), has the possibility to evaluate the usability of the product. Feedback while the product is in the process of being developed facilitates making changes to the source code. At the beginning of the project, a simple beta application was implemented with limited functionality. This beta version was used to start a discussion about the expectations of the prototype and to explore the possibilities in functionality that could be implemented within the project’s limited time frame.

Besides the review process, the developer performed testing and documentation. A large part of the documentation work has been done by using the SVN server. That is because the prototype redesign is stored as an iteration and the iteration is saved as a revision in the project’s repository history. A closer analysis of the code complexity, as for cyclomatic complexity or other measurement has not been done due to time limitation and therefore a full extended software testing has not been performed.

3.3 User Group and Usability Evaluations

The final result and usability evaluation of the prototype depended on the constellation of the end user group. Since the feedback received from these users was the main input when designing the software, it was also important that these users were representative. Considering the aim of the prototype, users with and without visual impairments in different age groups were desired. At the initial phase of the prototype development, only one user was involved in the process. Because of the user’s background and expertise, the feedback received was sufficient to get valuable information about the prototype on the ongoing process. The first iterations were about the graphical interface which was used by the developer to select POIs and to put them on the map. These first iterations (ca five) took just a couple of minutes to review.
The followed iterations were about whether the values from the orientation sensor were correctly interpreted and to test if the output was good enough, i.e. the auditory and haptic feedback. These tests were more time consuming since the developer and the user had to test the prototype outside in open air to get sufficient GPS coverage; the POIs were placed on different places around Lund: around Certec and Lundagård (at Lund’s centre). The testing time for these iterations were of about 15 to 30 minutes depending on how far the POIs the testing persons followed were located.

A larger test was performed with a bigger group that consisted of people that was involved in the HaptiMap project. At this stage, the prototype was able to perform only the bearing mode and the only user interface available was the graphical. Members of the group could use it to find POIs by following a path from the Central Train Station up to Kulturen and the developer was able to show the different settings for the auditory and haptic feedback.

The following stage of development was to implement the User Interface (UI) and to add guiding functionality. When the code implementation stage was completed, the prototype was again tested by the developer and the expert user at Lundagård. The testing involved not only the bearing but also the guiding mode. Several POIs with different distance ranges were placed on the map and the interaction with the UI could be performed. Since the iterations took place at Lund’s centre and because the tests involved more features to test, the testing were more time consuming. The evaluation of the prototype and feedback to the developer took approximately one to two hours to go through with.

The evaluations of the iterations with the prototype so far involved only sighted people and because the prototype is supposed to be used by people with visual impairments, it was necessary to iterate with a blind participant. The first participant was a younger teenager who has been blind since birth and needs a white cane for navigation. Three iterations in a semi unknown environment were performed with this user: two outside the Certec building and one iteration at Lund’s centre, Lundagård.

The final iteration before a larger testing group was put together, took place at Lund’s central park with two members of the Haptimap project and the developer, whereof one was this project’s supervisor. A selection of POIs for the final test were marked and the route was tested. Final adjustments were implemented to the prototype and the process in contacting participants began. The developer/author received a list of participants from previous tests performed at Certec and they were contact by mail and phone. The participants that were contacted received a small description of the test purpose and where it was planned to take place. The testing group consisted of one sighted user and five persons with visual impairments, representing both younger and elderly users from 14 to 80 years old. At the start point of the testing session, a walkthrough of the prototype’s functionalities was performed and the test tasks were explained. The users were video filmed during the session for later analysis of the material.
This chapter describes the prototype developed in the project, the PointNav application. A description of the mobile device and a more detailed section about the application developed will follow. The prototype description includes the user interface, validation of gpx files and the process when signing the application for release.

4.1 The Mobile Phone

The mobile phone used for this project was an Android Dev Phone 1, a smartphone developed by HTC that was specially designed for developers. It is basically a version of the HTC Dream device with a feature that differentiates it from other models. It provides the ability to manipulate the system software and configure it for different aims, i.e. the bootloader is unlocked and the developer is allowed to make changes to the Android platform and install custom system images based on the modified OS into the smartphone (Android Developers, 2010).

The Android Dev Phone 1 is equipped with touchscreen display (480x320), QWERTY keyboard, Tackball, Wi-Fi 802.11 b/g, handsfree, A-GPS and four sensors: 3-axis accelerometer, 3-axis magnetic field sensor, orientation sensor and a temperature sensor. The firmware installed on the device is Android 1.6, known as "Donut" which features the integration of the TTS-engine, which enables the speech synthesis conversion from string values. The available languages in the package are: English (American and British accents), French, Italian, German and Spanish.

4.2 The PointNav Application

Visibility, mapping and feedback are the three key attributes that lead to a good user interface according to Norman (Norman, 2002) as mentioned in the section 2.6 on page 15. How is visibility obtained when the users have visual impairments? The question is not trivial and a suggestion to how to implement a useful interface is not easy to give. The PointNav application tries to combine different technologies, such as haptic and auditive technology when interacting with the
user. While haptic technology can deliver a tactile feedback that combines vibrations, forces and motions, the auditive feedback can narrate a detailed situation or environment. When one reaches a certain level of representation of a visual interface through the mentioned technologies, then there is a chance to integrate the mapping information to representations and add some interaction feedback to the user interface. The PointNav application is developed with visual impaired users as the target group and it tries to implement the mentioned approaches by combining software development and by taking advantage of the hardware available on the smartphone.

The application allows the user to upload gpx-formatted data files into the device’s Secure Digital Card and then the waypoints, represented in the data files, are symbolized as points of interest (POI) on the map. The reason for using a map at all in a non-visual application map is to take advantage of displaying location and orientation for seeing users, such as the developers, and is not intended for the end user. The interaction with the users takes place through the Haptic User Interface (HUI) represented on the smartphone’s touchscreen. The actual interaction with the user consists of vibration patterns and synthetic speech. The location information is acquired from the GPS system and the orientation consciousness comes from the smartphone’s sensors. The PointNav gives the user the opportunity to scan the environment for POIs on different distance ranges from the user’s position. The user can then save a POI for further use and choose to let the device guide him/her to that POI. The application does not provide any routing but only indications in which direction the POI is located. It is displaying the path to the destination as a straight line and not taking the surroundings into consideration. The instructions provided from the application are: keep right, keep left, keep straight and turn around (figure 4.5 on page 28).

4.3 The Haptic User Interface: HUI

The interaction with a modern mobile phone is mediated by an interface which is normally graphical and therefore called a Graphical User Interface (GUI). Since the aim of this project is to provide a user interface to people with visual impairments, a GUI cannot be used stand-alone. Therefore, the use of other channels or modalities is necessary and in this specific case, the tactile and auditive channels are used. The Haptic User Interface (HUI) consists of a 3x3 matrix with non quadratic buttons. The buttons have the same height, but the width is varied. They cover the entire touchscreen of 320x480 pixels and the grids are separated with 1 pixel in between (figure 4.2 on page 25). The placement of the buttons in the middle tries to create a natural mapping, giving the user the feeling that close, middle and far distances are represented by the position of these buttons as seen from without. The guide me and bearing buttons placement offers an easy access to the mentioned states when the user is holding the phone in the palm of the hand. The altered size of the buttons’ width gives a good affordance since the device’s edges facilitate finding the buttons on the sides, and the ones on the middle, because larger size, become easier to find.
4.3.1 Previous HUI design

The first HUI was designed such as the whole mobile phone’s touchscreen was used. The main idea was to place 3 columns of same width, divided in an arbitrary number of buttons where each button were separated with 1 pixel in between.

The side columns consisted of 2 equally sized buttons and the column in the middle of 3 buttons. The buttons in the middle represented a way to increase or decrease the scanning radius. In this very first version, adding a POI was performed at the same time as when the user selected guide me and the user could then remove the POI when it was not longer desirable. The increase and decrease button behavior was to alter the scanning radius by 10 meters each time they were selected and if the user kept the finger on the button for more than 2 seconds, a change of 20 meters every second was achieved.

4.4 Interaction with HUI

The interaction between the user and the HUI is established in two levels. The first level, which is the one the user gets in touch with at the beginning of the interaction, indicates the actual tactile feedback which is accomplished by a small vibration occurrence when the user presses his/her finger against the touchscreen. The vibration reoccurs indicating that the user’s finger is moving away from one button to another and only then. The vibration feedback lasts for 40ms. If the user does not know which button he/she is interacting with, he/she can let the finger stay on that button and an auditory feedback is given after 1.5 seconds, telling the user about the current button’s function. The information is repeated every 1.5 seconds if the user leaves the finger on the same button. The user decides about the action to perform by simply releasing the finger from the button on the touchscreen. Depending on the current state of interaction, the user might toggle between them and multiple selections can apply to the same state without chang-
ing the mobile phones behaviour, i.e. if the user by mistake touches the mobile phone’s touchscreen while guiding, he/she might select the guiding button again and no data or interaction is altered by the selection.

Below is a small description of the buttons’ behaviour:

**Guida** The user releases the finger on this button when she/he wants to get directions to the saved POI. A POI must be selected before so that one may get guiding instructions.

**Fjärran** Sets the distance filter to: 200 to 500 meters.

**Pejla** When releasing the finger on this button, the scanning interaction begins.

**Mera Info** Used when the user would like to hear more information about the POI saved in memory.

**Mellan** Sets the distance filter to: 50 to 200 meters.

**Mute** Toggles between mute or not mute. Can be used in all modes.

**Välj POI** When in bearing mode (Pejla), the user can save the place he/she is currently pointing at.

**Nära** Sets the distance filter to: 0 to 50 meters.

**Starta/Stoppa** Starts or ends the interaction with the users. It does not end the application but it is used more for pause purposes.

### 4.5 Scanning or Guiding

The PointNav application consists of two initial states: Scanning/Bearing or Guiding. At the very start of the application and when the HUI is active, the Scanning/Bearing is chosen by default when the user presses the Starta/Stoppa button. The user can exit the application at any state (see figure 4.3).

![Figure 4.3: Bearing and Guiding States](image-url)
4.6 Bearing: Scanning for POIs

When selecting the scanning mode (default mode when starting the application from the HUI), the user lets the handheld device rest on the palm of the hand while he/she points at different directions with it. The default setting is to scan for POIs that are located between 0 and 50 meters of distance from the user’s position. The handheld device indicates when a POI has been found by giving a 100ms vibration signal and by adding auditive feedback to the action. The auditive feedback consists of the name of the POI and the distance to the POI calculated from the user’s position. The user can filter POIs by selecting other intervals on the HUI. The available intervals besides nära are: mellan and fjärran.

![Figure 4.4: Figure representing the Bearing mode with distance ranges and angle limits](image)

The POI found by the device is the one that is closest to 0° degrees and does not exceed 30° degrees such as $-15 < \text{POI's position} < 15$ (see figure 4.4 on page 27).

The user can then save the POI to the device’s memory by selecting Välj POI. If there are many POIs within the same range, the PointNav will select the one closest to 0° degrees and the application saves the POI in the temporary memory until a new POI has been found. The interaction is not interrupted when the user saves a POI and will continue to scan as long as there are POIs that satisfy earlier conditions in distance and bearing. The scan frequency is 500 ms if the previous POI is different from the last one stored in temporary memory, otherwise the frequency is 1500 ms.
4.7 Guiding

In Guiding Mode, the PointNav application guides the user to the previously selected POI. That is, the POI that the user explicitly saved to memory while in bearing mode. The handheld device guides the user by indicating the direction to the selected POI. There is no routing implemented in the application and the effort is instead based on a direction proposal. It means that the handheld device only calculates the shortest path to the selected POI and it gives instructions about in which direction the POI is located. The interaction with the device is illustrated in Figure 4.5.

![Guiding mode with angle representations](image)

**Figure 4.5:** Guiding mode with angle representations

The angle intervals used in the prototype derive from the recommendations presented in the article (Magnusson et al., 2010). The interaction with the user in this mode is through haptic and auditory feedback. The auditory feedback is generated by the TTS engine and the spoken text is a combination of the POIs name, the distance calculated to the POI, followed by a text string as specified in figure 4.6 on page 29. It can be turned off by pressing the Mute button. The haptic feedback can be changed to different predefined profiles depending on the user’s preferences. The user can also turn off the vibrations completely if desired. The application has a settings menu where these functions are available. Details on these settings are explained in 4.10. The default vibration feedback setting is inspired by the PocketNavigator (Pielot et al., 2010) with some modifications.
The patterns that come along with the respective directions are described in table 4.1. The guiding interaction ends when the user approaches the POI with a distance of 15 meters. At the arriving occasion, besides a new vibration signal, the speech generated encourages the user to stop with the message “Arriving at <The POI’s name>. No more guiding” every 10th second. The regular frequency of the interaction when guiding is 5000ms.

The state machine for the Guiding mode is represented in figure 4.6 on page 29. The Guiding mode can be reached only if the user saved a POI in the previous bearing mode. In other case, the PointNav will stay in Bearing mode (see figure 4.3).

<table>
<thead>
<tr>
<th>Direction</th>
<th>Pattern</th>
<th>Text String Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right</td>
<td>500v + 100g + 200v</td>
<td>&quot;keep right&quot;</td>
</tr>
<tr>
<td>Left</td>
<td>200v + 100g + 500v</td>
<td>&quot;keep left&quot;</td>
</tr>
<tr>
<td>Straight</td>
<td>(200v + 200g) x 3</td>
<td>&quot;keep straight&quot;</td>
</tr>
<tr>
<td>Turn</td>
<td>(150v + 50g) x 5</td>
<td>&quot;Please, turn around&quot;</td>
</tr>
<tr>
<td>Arrived</td>
<td>(50v + 50g) x 5</td>
<td>&quot;Arriving at &lt;The POI’s name&gt;. No more guiding&quot;</td>
</tr>
</tbody>
</table>

**Table 4.1:** The Vibration Patterns where v = vibration, g = gap
All values in milliseconds.
4.8 Avatar Mode

There are occasions when the regular filtering with distance ranges is not enough to find a specific POI. That is the case when many POIs are situated very close to each other on a far away distance. That is because the relative angle from the user’s position in relation to these POIs becomes too small in order to get a stable value to the $0^\circ$ bearing. The idea is to virtually change the user’s current position closer to the desirable target. When the user gets closer to one POI, the angles to the other POIs will eventually increase and a better start point is provided. Entering the Avatar Mode is performed manually by the developer or advanced users that have previous experience with the function. It is available in a submenu on the HUI and the avatar position depends on where the user is pointing at. For example, in figure 4.7, the POI that the user would like to choose is the red dot, but since the yellow ones are in the way, the avatar will virtually move the user to the new position illustrated with a green hexagonal, minus 25 meters from the calculated distance to one of the yellow POIs. The interaction continues in bearing mode and the user can save the desirable POI. If the user chooses to get guiding directions by entering the guiding mode, the avatar mode will exit automatically.

![Avatar Mode Diagram](image)

**Figure 4.7:** In Avatar Mode. The red dot is the desired POI

4.9 Visual Interface

The main interaction with the users is performed by the HUI, but settings and other utilities that customizes the prototype were accomplished through a graphical interface. The first version of the GUI consisted of tree tabs that contained two different lists and one tab with the map. The reason was to separate POIs from each other depending on the waypoint type. This approach was not practical, since waypoints cannot be divided into only two different types and the need of
more tabs in order to distinguish them was not practical. Instead, a design with two tabs was preferred and it resulted in one tab for a list (see Figure 4.8) of POIs, with the waypoint type information underneath, and one tab for the actual map and the POIs set out (see Figure 4.9). The idea to separate the POIs depending on their classification ended up with just different representations of them on the map. That was achieved by setting out different pins for the POIs, as one can see on Figure 4.9. The blue pins represent recommended POI’s, the white pin with a yellow dot was used to represent a reference POI and the red pins do not fall in any classification and were used for arbitrary types.

The POI list is empty at the start of the application. The user enters a submenu and selects to upload a gpx file into the POI list. When selecting a gpx file, a list of files that are stored on the handheld device’s SDCard will appear and the user can select one of them. When the POIs are uploaded, the application calculates the distance and the time from the current user’s location to each POI. The calculation of time is improper since it assumes that the user can walk at 70 meters per minute and that no obstacles are found on the way. One can also add that the distance appears to be very accurate which is not the case, since it depends very much on the accuracy of the GPS system at the given time the list is updated with fresh data. From the list of POIs, the user can select one POI and the POI will be placed on the map, or the user can select all POIs in an extra submenu and all POIs are then placed on the map.

Figure 4.8: The POIs’ List Tab  
Figure 4.9: Map Representation Tab
4.10 Settings

The settings submenu can be reached from both the POI list tab and the map tab. The settings are provided to let the user customize the application in order to get a more personalized profile. The functions that can be changed are:

**Vibrator**  Lets the user turn on/off the haptic feedback

**Guida mig profiler**  A list of vibration profiles to choose. Currently two profiles give proper feedback: the PocketNavigator inspired profile and Morse Code profile, representing L, R, S for left, right respectively straight patterns.

**Pejla profil**  The user can choose how he/she wants to interact with the device while scanning. The choices are: vibration on/off, a pling sound on/off and feedback about the distance to the POI. The POIs name is always spoken in all settings.

**Visa POI info**  When selected, the user can select a POI from the list and instead of placing the POI on the map, an extra window with information about the POI is displayed. The information is more about phone number, address, etc.

**Språkinställningar**  Changes the speech language. English and German are available.

4.11 Validating a GPX File

In order to create or to fetch a GPX file, one should be sure that the format that the file represents is correct. Apache provides a validation tool, SAXCount.exe which can be used for this purpose. The bin files are located at http://xerces.apache.org/xerces-c and the command

```
SaxCount.exe -v=always -n -s -f gpx_file.gpx
```

on a Windows OS gives the necessary information about the current file for correctness or indication that the GPX file is incorrect. For this project intentions, a smaller file containing just a few fields for waypoints was created. The required information for representing a waypoint are: Latitude (<lat>) and longitude (<lon>) of the waypoint. Additional information such as elevation, creation date or time and the name of the waypoint can also be provided, just to mention a few of them. The ulund.gpx file contains, besides the required information, also: name and type. The reason is that since the data is represented in speech, it would be good to have the name of the waypoint and for future work even be able to categorize them in types, which represent if the waypoint is a restaurant, a bus stop, etc.
4.12 Obtaining an API Key

In order to get the map in the application working, the api debug key must be registered. This is done on [http://code.google.com/android/add-ons/google-apis/mapkey.html](http://code.google.com/android/add-ons/google-apis/mapkey.html)

4.13 Signing The Application

Applications that are developed with eclipse and that are used in an emulator or on a developer mobile phone (which is the one used in this project) need to be digitally signed. The applications can be signed in two different ways. In the first case, when developing, the signing works automatically with a debug key that is chosen by default in the eclipse environment when one has access to the SDK tools. If the intention is to install the application on regular user mobile phone, the application must be signed with a private key rather than the debug key according to the rules stipulated by Google.

The first part of the signing process is to create the RSA keypair that will be used to sign the application. The Java’s utility keytool is used resulting in a 1024-bits RSA-keypair which is selfsigned and valid for at least 10000 days. The validation value of 10000 days is only a Google’s recommendation and it implies that if one would like to support upgrades for a certain applications then the validity period should be for at least 25 years or more. The approach applies also if the developer would like to use the same certificate for different applications, if the application is about to be released on the Android Market (Android Market). The generation of a private key is performed with the following command:

```
$ keytool -genkey -v -keystore pointnav.keystore -alias pointnav -keyalg RSA -validity 20000
```

The key pair used to sign the application does not need to be signed by a certificate authority (CA), such as VeriSign, which is the common case when managing public key infrastructures (PKI). The developer may use selfsigned certificates since the aim of this procedure is to establish a trust relationship between the author of the application and the application as such.

One of the consequences of this approach is that since the MapView class used in the application for displaying a map on the device is directly attached to the developers certificate, suitable settings need to be modified before a release. The MapView class uses data from Google Maps and it provides a wrapper around Google Maps API (Google Maps, 2010). In order to make this work, one needs to obtain the MD5 fingerprint from the certificate that is used by the developer and submit it to Google in order to obtain the Maps API Key. This key is then inserted in the developed application and the MapView will then get the access to Google Maps.
The second part of the signing process is to actually sign the application. This is done by using Jarsigner that is available in the standard JDK. Jarsigner signs a java archive (JAR) and verifies the signatures and integrity of the signed JAR files. After signing the application with jarsigner, the zipalign tool can be used in order to optimize the final APK package. The zipalign tool is provided by the Android SDK package. If the developer is using Eclipse/ADT (Android Development Tools) the signing process gets easier since no specific intervention by the developer is needed when using it in debug mode. For public releases, the developer needs to export the application as an unsigned apk. After exporting the application as unsigned one can sign it with the proper certificate by using the jarsigner tool. The regular command is:

```
$ jarsigner -verify -verbose -certificate my_application.apk
```

Once the application is signed, the zipalign tool is needed to ensure that performance optimization is reached.

```
$ zipalign -v 4 your_project_name-unaligned.apk your_project_name.apk
```

In Eclipse/ADT the process has been simplified by using an export wizard, which performs the interaction with the keytool, jarsigner and the zipalign tool when the compilation is finished. The export wizard is available when selecting the project to be exported and choosing File > Export or by accessing the Android-Manifest.xml. After completed the Export Wizard, the application is ready for distribution.

### 4.14 Calibrating the Sensors

The orientation sensors in the handheld device might need to be calibrated from time to time. The orientation sensor reminds of the regular magnetic sensors in accuracy and sensitivity. When the true north is not tuned in, the user can move the handheld device such as it creates an imaginary number eight in the air. At the time of development, no functionality or method was found that could be implemented in the PointNav application in order to provide a calibration method.

### 4.15 Orientation Issues

The PointNav application uses the 3-axis accelerometer, 3-axis magnetic field and orientation sensors that in combination represent a compass which provides orientation to the user. There were two approaches to reach a compass representation. The first representation consisted of the orientation sensor only. The second one consisted of the combination of the 3-axis accelerometer and the 3-axis magnetic field sensors, providing a higher accuracy as result. When a comparison of values were performed, no significant difference was found other than that the higher accuracy was represented in float variables rather than integers, which is the standard output from the orientation sensor.
This aim of this chapter is to provide details about the test procedure during the final evaluation of the PointNav application. Descriptions about the tasks performed by the users are explained. Some details about the participants and their previous experiences with this kind of iterations are presented. The PointNav application was also evaluated during the development phase, which is explained in section 3.3 on page 20.

5.1 Final Prototype Evaluation

The final evaluation sessions took place in Lund’s Stadspark, a calm park in the heart of Lund’s city where pedestrians can walk in open areas. Besides pedestrians, people jogging or riding on bicycles can be found on the park’s paths. The environment was considered convenient for the testing phase since the target group consisted of people with visual impairments and the wayfinding would be safer when no traffic was present.

The test group consisted of six persons that performed the test on different occasions. Of the test persons, three were male and three were females, five had visual impairments and one was fully sighted.

Test Leader’s Workflow

The PointNav application does not contain any information about any POIs when started. A valid gpx file from the smartphone’s sdcard is manually loaded by the test leader and the file is parsed into memory. When a gps signal is available, calculations about distances between the user’s current location and POIs are started. The procedure is explained in figure 5.1.

Test Group Description

User 1 Female, 80 years old with some periphery vision left. Previous experience with mobile phones of the simpler kind used for voice communication only. She does not use any mobile phone any more and has no experience with touchscreens or location services.
User 2  Female, 16 years old and completely blind. She uses a white cane to navigate. Familiar with mobile phones and synthetic speech. Previously test user of the PointNav while in development. She tested the application three times before this session. She tested a touchscreen on an iPhone once before she came in contact with the PointNav application.

User 3  Male, 14 years old. Sighted and advanced user of mobile phones. Familiar with touchscreens and location services.

User 4  Male, 52 years old and completely blind. He uses dog and/or white cane to navigate. Advanced user of mobile phones and familiar with synthetic speech. The user has experience with gps location services.

User 5  Male, 44 years old and completely blind. He uses a white cane to navigate. Advanced user of mobile phones, familiar with touchscreens and gps location services.
**User 6** Female, 45 years old and completely blind. She uses a white cane and she is familiar with synthetic speech and mobile phones. She has some experience with gps location services.

**The Tasks**

In order to evaluate the application, the users had to fulfil three tasks:

1. Find one POI (**Beechstock**) at middle range (**Mellan**) from the start position and to get there by using the guiding directions.

2. Find one POI (**Neverhood**) at far away distance (**Fjärran**) from **Beechstock** and to get there by using the guiding directions. The use of the Avatar mode was encouraged if the POI was too hard to find.

3. Stand at one location (near Fountain) and find how many POIs he/she could find within the close range (**Nära**)

At the beginning of the prototype evaluation, a consent was read to the users where they were informed about their participation in the HaptiMap project. They were also informed that the session would be video recorded. When the formalities were completed, the users were given an oral description about the functionality of the PointNav application and a walkthrough of the HUI was performed. The users took their time to get familiar with the HUI and the imaginary buttons on the touchscreen. They were able to try out the functionality by finding the start position from where they were standing in the park. The starting point from where the users were looking for the start position was near the park's cafeteria.

The users could choose between using earphones or the smartphone’s loudspeakers. They were also informed about the customization of the scanning/bearing settings, i.e. toggle the haptic feedback and/or to turn off the speech. These settings required assistance from the test leader since they were available on a submenu that was not included in the walkthrough at the start of the session. Once the users were standing at the start position, they were asked to complete the first task: to find the first POI, **Beechstock** and to get guiding directions. When arriving to **Beechstock**, the users were asked to find **Neverhood** and to get guiding directions. The path chosen by the users when heading to **Neverhood** was of importance for the last task. Besides the problem in finding the POI, there was an embankment between **Beechstock** and **Neverhood** that forced the users to make a decision of taking a left or right turn at the embankment. If the users chose the path to the left, then they had the chance to perform the last task before the reached **Neverhood**. In the other case, the participants reached **Neverhood** and then they were guided by the test leaders to the fountain, from where they could execute the last part of the test. The POIs’ positions are illustrated in figure 5.2 with yellow dots. The last part of the test began a few meters from the Fountain (close to the white star in figure 5.2) and the POIs to be found are represented in green, including the Fountain. The users did not get any assistance from the test leaders in choosing range intervals or hints in pointing directions. If the users did
not remember all the functions of the HUI then the test-leaders could give hints about where they were placed on the touchscreen. This help was provided since the evaluation of the design/layout of the HUI was not the primary purpose of the test. At the end of the test session users could give feedback about the application and their interaction experience.

5.2 Test Results

The tests were carried out within about one hour from the instruction walkthrough to the final task at the Fountain. All users were able to perform all tasks with different results at the end of the iteration. The interaction with the HUI did not create any problems and the participants got used to the functionality. The major problem with the interface was to remember what the buttons were about. Some users did remember that when letting the finger rest on a button, an
auditive feedback gave them answers about the specific function. Although some users struggled with the HUI at the beginning of the session, the users did perform really well when the first task was completed and no more questions were asked about the button placements. There was only one user that had another point of view regarding the HUI. The suggestion was about reducing the number of buttons and instead implement more functionality into the few remaining ones.

The scanning mode was appreciated by the whole group and especially by the users with visual impairments. There were nevertheless some situations that created problems while scanning. That was when two or more POIs were close to each other in such a way that they were lying almost on the same bearing. The situation occurred due to the compass jitter and the assumption is that the vibration feedback when a POI was found interfered with the compass and lead to instability. This problem was more obvious when the users were about to perform task number two and the application began to oscillate between two POIs. While in scanning mode, the synthetic speech is interrupted every time a new POI is found and for this reason, the speech was not able to finish the POI’s name, making it difficult to the users to select and save that POI to memory.

The test leaders expected the the Avatar mode approach to be needed when scanning for the second target but the users were able to find the POI without this extra feature. The users took advantage of the last “saved POI in temporary memory” and sometimes they moved away the handheld device from the current bearing. Two users found out that by tilting the handheld device, the coordinates on the orientation sensor are not remapped and the application did not find any more POIs. One of the users that took advantage of this feature was the female that had been in contact with the prototype at earlier stages. Task number two was more time consuming due to this problem but all uses succeeded with some effort.

The guiding approach was somehow misunderstood at the beginning of the test. The users misjudged the direction proposals given by the application. They took a sharp right or sharp left when the application was suggesting to “keep right” or “keep left”. The test leaders explained, when necessary, the intentions with the direction suggestions and then it became more obvious how to react to the feedback. One of the users with visual impairments did not like the guiding approach because he felt that the application was not good enough when guiding. The instability of the guiding procedure is somehow associated with the gps signals received on the handheld device, but also on occasion due the compass jitter. In order to reduce compass jitter, the users were told to calibrate the handheld device while walking from time to time and to avoid sharp horizontal movements when following directions. The calibration while walking was carried through a slight shaking with the hand and not as explained in section 4.14 on page 34.

The auditive feedback was sometimes difficult to perceive and four of the participants used earphones on the test. The participants felt that the auditive feedback offered a very poor performance and that it was difficult to understand what the synthetic voice was trying to say. This problem occurred at the walkthrough but
also when the participants were looking for POIs while performing the test tasks.

The actual vibration patterns offered by the handheld device were not perceived by the participants. They did not differentiate between the patterns until the end of the iteration, when the test leaders asked about the experiences with the haptic feedback and explained about the differences. The same person that did not like the guiding approach chose to turn off the vibration feature and relied on the auditive feedback only.

Once the participants arrived to the test POIs they stopped immediately when the PointNav changed the interaction behavior. In the occasions when the participants did not understand what the synthetic speech was saying, or they just missed what it said, they just stopped and waited for further instructions.
The development of the PointNav application offered an interesting approach for testing various designs and the iterative concept was highly appreciated. Many functions developed during these iterations were not included in the final testing phase since they did not offer the desired output. The development of the final HUI went through several versions before a final proposal was found. Even if the HUI was intuitive for some of the users, there is still room for improvement such as another solution for the distance filtering.

In a very early stage of the application implementation, the developer found some performance problems when updating the POIs’ distance information. At earlier occasions of testing, there was a list of approximately 300 POIs with no filtering approach and the calculations were very cpu demanding. The implementation of the distance filters clearly improved performance and the application ran with no lagging as a result at the final test. The limited calculation capacity is obviously a common problem when handling with mobile phones since the available cpu power is limited to save battery consumption among other things.

The results described in section 5.2 indicate that it is possible to develop an application for Android for persons with visual impairments by using a touchscreen. It also indicates that scanning for POIs is an appreciated feature by users with visual impairments since it provides an easy way to find a destination. The guiding approach did not rate as good as the scanning, but the users were still able to carry out all the test tasks with the guiding support. Since the guiding does not offer any routing, it was more difficult to get used to the direction suggestions in the beginning. The issue occurred since the users did turn abruptly right or left when indicated instead of changing direction in a smoother way. Users who used any kind of gps based location service before mentioned that one of the problems they find is the "turn right" or "turn left" recommendations which in several cases are erroneous ones because of the gps inaccuracies. As a result, the "keep right" and "keep left" let the user be more careful about where to go and the user has to make more active decisions. This issue was clear when performing task two and the users had to be more aware about the situation when an obstacle was found. In this particular situation, the user had to decide in turning left or right even if the application was encouraging for straight ahead. The straight
ahead suggestion was difficult for two participants at the beginning of the test because they just wanted to go straight ahead even if no path existed or trees were in the way. The other users, besides the sighted user, kept themselves to paths as much as possible and they were more careful when obstacles were found.

One of the drawbacks about the synthetic speech generated by the TTS engine was that the users had problems in understanding what it was saying. People did not understand even if their English skills were very good. The suggestion at the beginning of the application implementation was not to use the synthetic speech because the Swedish language was not available. Using the Swedish language was a precondition because the prototype was to be tested by elderly also. The assumption was that this group did not have very good skills in English compared with younger groups. This goal became difficult to fulfill since it was desirable that the device could inform about distances. This condition was difficult to meet if prerecorded voices were used. As a result, a constraint in POI names was introduced because Swedish names became difficult to understand, for example "A-huset" became "A- hyphen- he-use-it" and that did not make any sense, neither to the end user nor to the developer.

The second task in the test was more complex to carry out due to compass jitter and perhaps also GPS inaccuracies. There was a possibility to use the "Avatar mode" on this occasion but because of the complexity in using the function, the test leaders opted to avoid its use with the participants. It is clear that a new approach regarding filtering in these situations should be used and an algorithm that could propagate the POIs in an appropriate way should be implemented. The inaccuracy of the orientation sensor was difficult to find a technical solution for and the use of another more accurate orientation sensor, such as the gyroscope, would be preferable. The utilization of a more accurate orientation sensor does not however, compensate the inaccuracy with the GPS signals. The test was performed in an open area and even then small variations due to gps inaccuracy were perceived. As illustrated in figure 5.2, the starting point is very close to a traffic road and due to inaccuracy, the start position POI was often misplaced in the middle of the street. The margin of error in this case was approximately 10 to 15 meters.

According to the feedback from the users after the last completed task, the overall interaction worked well. The users did not feel that the updating frequency in speech or vibration was too high, besides one user that did turn off the vibration while guiding. The feedback gave some kind of safety feeling, according to two of the users when describing their experience with the application. Although the vibrations patterns were not well interpreted in the beginning of the test, the users did agree that it was possible to distinguish the patterns, but it was necessary to use the prototype for a longer period in order to get well used to them. One of the test users, the younger female, did test the application on a previous occasion with no auditive feedback and said that the vibrations were easy to follow once she was concentrated. The second vibration profile, the Morse code, was not used and perhaps the response to this pattern would be the same; the user needs to
be aware about the patterns and they need to get familiar with them in order to perceive the differences.

One of the functions that was not used more than once with one user, but gave positive reactions was the Mera info button. The idea with this function was to offer the user more information about the POI they saved to memory. The idea came up when discussing about cultural places around Lund and by giving the users the possibility to hear a small description of the POI they are pointing at.

The PointNav application suffers of diverse bugs besides the user interaction. One of these bugs is when the user receives a phone call and the application continues to run in the background with all feedback on. The elimination of bugs is always difficult and requires time.
Conclusions

According to the aim of this master’s dissertation described in the Introduction, the evaluation of the application focused on 5 items.

To consider the response of users with visual impairments to the Haptic User Interface approach when using a touchscreen
The HUI developed for the PointNav application did offer a way of interacting with the users, combining tactile and auditive feedback. It became intuitive to the participants at the evaluation occasion, to both elderly participants and those who had never been in contact with touchscreens before. Some of the participants did react to the number of buttons and probably the mapping should be changed. There is still room for improvement and further studies should be carried out.

To investigate if a simple multimodal guiding approach without routing helps users finding POIs in open areas
The guiding approach offered by the PointNav application is based on a recommendation basis. That is, the application does not steer the users in absolute directions, such as "turn to the right in 10 meters" or "turn to the left in 20 meters". Instead, the application recommends the users to keep a certain direction while walking. These recommendations are perceived through haptic and auditive feedback which captured the users’ attention. The haptic feedback was perceived but the different patterns generated for different directions were not completely intuitive. The users were told about them when the evaluation session was over by the test leaders. The offered auditive feedback was not of very good quality. The participants found it difficult to understand speech synthesis but they got used to it after some time during the evaluation. All participants were able to carry out all the test tasks with the guiding support in the park, including obstacles and larger open areas.

Find out if this kind of application could facilitate navigation tasks for both sighted users and users with visual impairments
It came to light that this kind of navigation applications was more appreciated by persons with visual impairments than sighted users. The overall reaction from the participants with visual impairments was that, the scanning approach offered an overview of available POIs and it help them with orientation. The idea of
using this kind of applications besides evaluation tests was conceivable.

**To test some filtering algorithms when scanning for POIs**
The prototype offers a filtering approach in distance and scanning/guiding angles. The recommendations provided in (Magnusson et al., 2010) for guiding and scanning worked well and the distance filtering implemented gave an initial indication that it was satisfactory in this evaluation. There is still room for additional investigations on this topic.

**To explore the iterative development symbiosis between developer and end users**
The development cycle of the prototype was performed iteratively and the developer’s experience with the process is very positive. The functionalities were added during the iterative process and the output was reviewed by an experienced user. When the prototype had enough functionality, an end user with visual impairment contributed with feedback, which the developer used to improve the next revision of the application. This type of iterative development should be the clear choice when customizing products with an end user in focus.
Chapter 8

Future Work

It is obvious that there exists a need for this kind of applications for persons with visual impairments and it is clear that further research is needed in this area. Proper algorithms with smart filtrating functionality need to be implemented and further studies on the interaction with the users is necessary. The developed prototype is not good enough for this group when guiding but it does supply a scanning approach that can be exploited further. The combination of scanning features and a proper service that provides information about a limited number of POIs by a smart filtering algorithm could be combined, resulting in a feature like: Where is the closest post office? or Where is the closest pizzeria? Another possibility is to adapt the scanning option in such a way that the device provides information about POIs close to the user while the user is walking towards a destination. The TTS engine is desirable to improve. The current version offers only a limited number of languages and an improvement in voice quality would be appreciated.


User Scenario

Ingrid is a tourist in Lund and she would like to visit Kulturen, which is the biggest museum in Skåne (Kulturen, 2010). Besides regular indoor exhibitions, Kulturen is also an open-air museum which offers more than 30 buildings that fill an area of about two blocks in the city’s centre. The buildings with respective interiors show how people lived in Skåne through the ages. The indoor exhibitions are both of permanent and of temporary kinds. Besides exhibitions, Kulturen offers a nice garden café during the summer period.

Since Ingrid is in Lund during summer time, she found out that this particular day, besides the rest of the past week, is a very sunny and warm day and she decides to visit the outdoor exhibition at Kulturen. Ingrid is around 30 years old and has been visual impaired since she was 5 years old. The visual impairment is of such a magnitude that she is unable to percept more than just a blurry image of the surroundings. This means that she does not see good enough to perform everyday tasks without special tools. She likes to be outdoor but has a small fear for open places, since she feels insecure with orientation when she puts herself in a context when the current open place is not known since before. Ingrid does not always like to be guided by someone else and she feels a strong need of doing things by herself. Because of this encouragement, Ingrid has devoted a larger part of her time in testing tools that give her the freedom she likes to experience. This particular dedication has been a key in the understanding of modern devices such as mobile phones and other handheld devices. One feature that she likes a lot is the navigation tool that was integrated in her latest Android based mobile phone.

Ingrid gets to Kulturen and the outdoor exhibition. She would like to know where the different buildings are and she picks up the mobile device in order to get an idea about where things are placed. The current mode in the supplied application is the “bearing mode” which means that the user is able to probe around by pointing to different directions while the device is providing information about the points of interest. The information given to the user is both the name and an estimated distance to the point of interest. The distance is just estimated because
of accuracy issues and can not be given in absolute terms. In order to achieve a better perspective of what she is pointing at, she then decides to decrease the bearing radius to a more considerable value. In the beginning a default value (Nära) of 50 meters is set but by decreasing or increasing this value, many or fewer points of interest will be capture by the device. The tweaking procedure is achieved by stroking the touch screen on the mobile phone and by getting a small vibration given when the buttons are found. Besides the vibrations acquired when a button is found, a small sound feedback is added and Ingrid feels now more secure about the function of the current button. Ingrid decides now which point of interest she wants to go to by heading the device at Allmogehallen and then by releasing the finger from the touch screen on the Välj POI button, she saves Allmogehallen to the mobile phone’s memory. Ingrid is satisfied with the selection and chooses to get guiding instructions. Once again, she lets her finger rest on the touchscreen and releases her finger when she finds Guida. The mobile phone begins to guide her according to three parameters: keep straight, keep to the left respective keep to the right. The parameters are in spoken format and combines small vibrating patterns. In the middle of the path, Ingrid wonders about points of interest near the position where she is standing at. Since she has an application that can be used to that, she now looks for the button which takes her to the scanning/bearing mode again. Now she can point to different locations again and receive information about those point of interest. When she finds out that she still wants to get to the target chosen before, she just selects the Guida button again in order to continue to the original destination. After a couple of meters again, Ingrid is still curious about the surroundings and by changing to "bearing mode" she finds out that there is another point of interest that she would like to visit instead: Hylla smedja. Pointing at the new point of interest, Ingrid can now make her choice by releasing her finger on the Välj POI button followed by Guida and now she is heading to the new direction. When Ingrid arrives to Hylla smedja, she exits the application by pressing the "back" button, which is a physical button placed on the front of the mobile phone. This button is familiar to her since it is frequently used in other applications too.