Utilizing web standards for cross platform mobile development

Author: Daniel Hjärtström
Supervisor: Bahtijar Vogel
Semester: Spring 2012
Level: C
Course code: 2ME10E
Abstract

This thesis has taken part as an experimental development within the Learning Ecology through Science with Global Outcomes project. It introduces the area of cross platform mobile application development and provides a possible solution for tackling the current fragmentation of mobile devices and platforms for data collection. During the process, a mobile data collection prototype was designed, implemented and deployed on Android, iOS and Windows Mango by using standards and web standards such as HTML5, CSS3, XForms and JavaScript. The deployed prototype was then tested with users in order to collect the necessary data to help answer the questions that were formulated. The results indicate an ease of use of the prototype in relation to cross platform development and also shows potential benefits such as less code and time. Cross platform development provides a way to counteract the current fragmentation between mobile platforms.

Keywords

Cross platform, Mobile application prototyping, HTML5, XForms, Open standards, Web technologies, PhoneGap, iOS, Android, Windows Mango, Mobile data collection
# Table of contents

1. Introduction .......................................................................................................................... 1  
   1.1 Purpose .............................................................................................................................. 2  
   1.2 Problem definition ........................................................................................................... 2  
   1.3 Limitations ...................................................................................................................... 2  
   1.4 Expected results ............................................................................................................ 3  
   1.5 Time plan ....................................................................................................................... 3  
   1.6 Disposition ..................................................................................................................... 4  
2. Foundations ......................................................................................................................... 5  
   2.1 Mobile computing .......................................................................................................... 5  
      2.1.1 Interoperability ......................................................................................................... 6  
   2.2 Cross platform mobile application development and tools ........................................... 7  
      2.2.1 HTML5 .................................................................................................................... 7  
      2.2.2 PhoneGap ............................................................................................................... 8  
      2.2.3 CSS3 ...................................................................................................................... 9  
   2.3 Mobile development: web VS native .............................................................................. 9  
      2.3.1 Web applications ...................................................................................................... 10  
      2.3.2 Native applications ................................................................................................. 10  
   2.4 Usability ....................................................................................................................... 10  
   2.5 State-of-the-art projects ................................................................................................. 12  
      2.5.1 The ODK platform ................................................................................................. 12  
      2.5.2 Learning Ecology through Science with Global Outcomes (LETS GO) ............... 12  
   2.6 Identified requirements ................................................................................................. 12  
3. Methods used for prototyping .............................................................................................. 15  
4. Design and prototyping ......................................................................................................... 18  
   4.1 Initial implementation of the prototype .......................................................................... 18  
   4.2 Second prototype ......................................................................................................... 19  
   4.3 Third prototype ............................................................................................................ 20  
   4.4 Encountered problems and differences during development ....................................... 22  
   4.5 System overview ........................................................................................................... 24  
   4.6 Application flowchart and use case diagrams .............................................................. 25  
   4.7 Supported features of used platforms ......................................................................... 29
5. Methods for user testing ................................................................. 30
  5.1 Tasks for user testing ............................................................ 31
  5.2 Questionnaire ................................................................. 32
  5.3 Participants and settings .................................................. 33
6. Results of user testing ............................................................... 34
  6.1 Average time and score ......................................................... 35
  6.2 Cross analysis of user testing results ..................................... 40
  6.3 Questionnaire answers and observations during user testing .......... 41
  6.4 Summary of observations and users commentary ....................... 43
7. Conclusions ........................................................................... 45
  7.1 Discussion ........................................................................ 45
  7.2 Further development ............................................................ 48
Acknowledgements ........................................................................ 49
References .................................................................................. 50
Appendix A - Questionnaire for user testing .................................. 54
Appendix B - Tasks for user testing ............................................. 58
Appendix C – File formats and file system paths ............................ 60
Appendix D – Time and score correlation between users previously used platforms .... 61
Appendix E – Results of user testing ............................................ 66
Table of figures

Figure 1.1. Time spent on different areas of the thesis………………………………………..3
Figure 3.1. Prototyping model……………………………………………………………………..16
Figure 4.1. Rendered XForm in initial prototype………………………………………………..18
Figure 4.2. Question screen of second prototype on Android……………………………………..19
Figure 4.3. Multiple choice question (radio buttons) on iOS, Windows Mango and Android……………………………………………………………………………………….20
Figure 4.4. Last screen of a form on iOS, Windows Mango and Android…………………..21
Figure 4.5. Barcode capture screens on iOS and Android…………………………………………….21
Figure 4.6 Components overview of the developed system……………………………………….24
Figure 4.7 Mobile application prototype flowchart……………………………………………….25
Figure 4.8 Use case 1…………………………………………………………………..…….26
Figure 4.9 Use case 2……………………………………………………………………...27
Figure 4.10 Use case 3……………………………………………………………………….28
Figure 4.11 Supported elements and functions………………………………………………….29
Figure 5.1 Users performing testing of the prototype………………………………………….33
Figure 6.1 Previous used platforms by test users……………………………………………….34
Figure 6.2 Average time for performing tasks on all platforms……………………………..37
Figure 6.3 Overview of average task completion time for each platform…………………..38
Figure 6.4 Average score of tasks for all platforms………………………………………….39
Figure 6.5 Overview of average task score for all tasks of each platform…………………..39

List of tables

Table 2.1. Identified requirements for prototype development…………………………….13
Table 4.1 Supported functionalities during development process………………………..22
Table 5.1 Tasks created for user testing………………………………………………………..31
Table 6.1. Platform order during user testing……………………………………………………34
Table 6.2 Differences of average time spent on tasks for all platforms…………………..40
Table 6.3 Differences of average score on tasks for all platforms………………………….40
1. Introduction

Mobile devices offer a lot of possibilities doing the things that previously required a computer or other technical devices. If your device does not have the application you need, just download it trough your wireless internet and start using it instantly (Allen, Graupera & Lundrigan 2010, pp. 1-3). Unfortunately, there are some limitations, especially related to the development of mobile applications. Since development languages vary between different platforms, one needs to write in the specified language to access all native features and to get as smooth a development process as possible (Allen, Graupera & Lundrigan 2010, pp. 5).

Since mobile devices utilize different platforms it creates problems with fragmentation and difficulties in sharing of data between them. This creates a need for cross-platform open standard applications and solutions that are supported and can be used no matter what platform a mobile device is using. One approach to tackle this issue is to use any of the current frameworks that are available to developers and then run an application in an embedded web browser (Allen, Graupera & Lundrigan 2010, pp. 6).

In addition to current frameworks, the newly introduced HTML5 offers a lot of new possibilities that will aid users in mobile application development by proving many new API’s along with other useful features of which some will be introduced later in the foundations part in section 2 (Lubbers, Albers & Salim 2010). Overlooking the issues of fragmentation, modern mobile devices provide sensors that allow people to use just one device to perform the tasks that previously required multiple digital devices such as camera, video recorder, microphone or a compass.

In this thesis, a cross platform mobile application prototype for data collection will be implemented by utilizing web and open standards such as HTML5 and XForms. XForms is a way of creating web forms that are cleaner and more structured than HTML forms (Grehan 2005). They will allow for sharing of data between different platforms and devices using XML as data format (Casteleyn et al., 2009, pp. 15).

The mobile application prototype developed will be deployed across three major platforms (iOS, Android and Windows Mango) and will render XForms into HTML5 forms. It will support inputs such as text, audio, video, pictures, visual codes and GPS which is the data that can be collected by users using the application. Thus for the assessment purposes a user study will be conducted. This thesis is an experimental development that takes part within the Learning Ecology with Technologies through Science for Global Outcome (LETS GO) project that runs in the School of Computer, Science, Physics and Mathematics at Linnaeus University.
1.1 Purpose

The purpose of this thesis is to research the possibilities of utilizing open standards in order to tackle the fragmentation of mobile applications for mobile data collection by designing and implementing a mobile data collection prototype.

1.2 Problem definition

Developing native applications is limited by the need to use different programming languages and therefore requires more knowledge, time and money. Stark [1] (2010, pp.2) explains that by not developing native applications, one does not have access to built-in functions such as camera. They also mention that it can be difficult to create effects for the user interface. However, According to Stark [2] (2010, pp. 115) the HTML5 platform PhoneGap provide the possibility of using a platforms native features when developing web applications and will therefore be used during the design and implementation of the prototype.

Different programming languages and diversity of platforms makes mobile development a challenging task, especially when dealing with the fragmentation between these devices and applications. Thus, this thesis explores a solution to this fragmentation by using standards such as JavaScript, HTML5, CSS3 and XForms in order to create a cross-platform mobile data collection tool.

Taking into account the above mentioned challenges, the main question for this thesis is formulated as followed:

- How can open-standard technologies be utilized in order to tackle the fragmentation of mobile applications for data collection?

This question focuses mainly on how open standards can be used and will provide answers to the issues of fragmentation. But in order to get a deeper understanding on the subject two sub questions were also formulated:

- What components are required by the application in order to tackle fragmentation between diverse mobile platforms?

- What are the potential benefits of using web technologies and standards for cross platform implementation and deployment?

1.3 Limitations

The platforms that will be focused on in thesis are iOS, Android and Windows Mango. These platforms were chosen because statistics found show that these are the ones most currently used. The application will first be developed for Android, then Windows Mobile and at last for iOS. The extent of the development for this thesis will be limited to the creation of the prototype client-side using open-standard web technologies and will not have any type of rendering or visual representation server-side. However, some server-side coding will be
required in order for users to upload collected data from the application during testing. For the user testing of the application, ten people will be recruited to perform a list of seven tasks for each of the three platforms.

1.4 Expected results

Expected results of this thesis are to create a mobile application prototype that will render XForms for data collection on the selected platforms by using HTML5, JavaScript and CSS3. This thesis tackles in what way it is possible to create mobile applications that overcome the current interoperability issues between different mobile devices. With that in mind, this thesis will show:

- Differences between the used platforms
- Problems that occurred during the design, development and testing
- Applications performance/look in the different platforms
- If standards are sufficient enough to create this type of application
- Potential benefits of web and open standards for deploying cross platform mobile applications

1.5 Time plan

Figure 1.1 shows the time spent on the different parts when writing this thesis. Initial work was focused on studying the different technologies that would be used along with finding the appropriate literature. These parts ran almost throughout the whole course along with the development of the three prototypes. The third prototype was finished the day before user testing. After conducting six user test sessions over two days, results were analyzed and recorded. The conclusion was written the week after user testing and the final version of the thesis was submitted about two weeks later.

![Gantt chart showing time spent on different areas of the thesis]

**Figure 1.1** Time spent on different areas of the thesis
1.6 Disposition

The first chapter has introduced the scope of this thesis covering areas such as questions, problems and limitations. The second chapter will provide a theoretical background covering areas such as mobile computing, cross platform development and some state of the art projects. The third chapter will show the methods used for prototyping and the fourth chapter the development process for the prototype. The fifth chapter will cover the methods used for user testing and chapter six the results of the user testing. The seventh and final chapter will tie everything together by presenting a conclusion where the formulated questions will be answered.
2. Foundations

This chapter will provide a theoretical background to the subject of this thesis covering mobile computing, diverse mobile platforms, cross platform mobile application development and also provide the reader with an insight of developing native applications versus developing web applications.

2.1 Mobile computing

Mobile computing is the transition from home computers to the use of mobile devices, and as mobile devices are getting smarter, they are being used by people to perform the tasks that were previously carried out by home computers (Allen, Graupera & Lundrigan 2010, pp. 1). In that context, Fling (2009, pp. 30) explain that mobile internet connectivity surpasses that of home computers and state that: “[…] over 1.6 billion […] have access to the Web through a mobile device”.

Mobile computing does not only apply to the new types of smart phones. While older phones offer the same base functionalities as the newer, Fling (2009, pp. 8) lists the following characteristics that apply to smart phones:

- They use a common operating system
- They use a larger screen
- They provide a QWERTY keyboard for user input
- They have the possibility of using high-speed internet connections

In the context of mobile computing, Billi et al. (2010) emphasize that, while mobile phones have limitations, they are providing users with many appealing features and state that: “[…] new opportunities and limitations […] call for new ways of evaluating the appropriateness of mobile applications toward the users in terms of both accessibility and usability.”

With the portability and usability of a mobile device, the availability of mobile applications and the accessibility to the internet, mobile computing is here to stay (Allen, Graupera & Lundrigan 2010, pp. 1). In addition to these positive sides of mobile devices there are also a variety of sensors that are available to users. Sensors will aid in the area of mobile data collection and some were utilized for the developed prototype.

Mobile sensors such as the camera can be explained as: “a device that reads data from the user’s physical environment” (Barton et al., 2003). This could also be applied to sensors such as the microphone, GPS or accelerometer. The following list will show some of the sensors that according to Kota et al. (2009) and Kansal, Goraczko & Zhao (2007) are available to mobile devices:

- Camera
- Accelerometer
- Compass
- GPS
Kansal, Goraczko & Zhao (2007) point out that while all sensors do not require human interaction some, such as the camera, do. The camera sensor is not just for taking pictures, it also provides users with the possibilities of for example (if the correct application is installed) as explained by Fling (2009, pp. 52) to scan a barcode or according to (Layar, 2012) use it together with the GPS sensor to support such technologies as the augmented reality platform Layar. Kansal, Goraczko & Zhao (2007) also explain that sensors such as the GPS require external aid in form of a satellite with good visibility if it is to provide accurate results.

Because of new technologies, users now have the ability to utilize these sensors by using web standards such as HTML5. This is a step in the right direction when it comes to solving current issues of interoperability.

2.1.1 Interoperability

Interoperability can be explained as having compatibleness between different mobile platforms/devices and exchange data between these (Signore, 2003). Exchanging data between different devices and platforms are as such according to Vogel et al. (2010) best addressed by utilizing open standards. In this context, Charland & Leroux (2011) explain that web technologies provide us with the possibility to develop interoperable applications and services that are accessible to several different platforms and therefore a variety of sensors without having to create one application for each available platform. This is the way that the prototype in Design and Prototyping under section 4 was developed.

One emerging solution concerning interoperability is HTML5. Lubbers, Albers & Salim (2010, pp. 3) present the following principles that follow with HTML5:

- Compatibility
- Utility
- Interoperability
- Universal access

In order to achieve this, according to Hoy (2011), HTML5 is aimed at eliminating the current requirement of plug-ins which is especially important for mobile devices where finding a plug-in can be hard, or even impossible. Henceforth, Hoy emphasizes that it will take time in order for HTML5 to reach its full potential as a standard. Speaking of interoperability, Signore (2003) emphasize that: “XML, independent from platforms and languages, plays a fundamental role towards interoperability.”

2.1.1.1 XForms

XForms is an XML-based way of creating web forms that are cleaner and more structured than HTML forms (Grehan 2005). Though XForms provide this, plug-ins is needed when rendering the forms client-side (Grehan 2005).
However, since XForms are based on the XML language it is according to W3Schools [1] (2012) readable by JavaScript trough the XMLHttpRequest object provided by the W3C standard XML DOM. This makes it possible to eliminate the requirements for plug-ins by developing a ‘program’ to read the XForm and convert them into a language that is readable by a normal browser, thereby making it available to a large amount of diverse devices. It is this type of ‘program’ that has been created for the prototype in Design and prototyping under section 4.

2.2 Cross platform mobile application development and tools

Cross platform mobile application development means creating applications that will function across diverse devices and platforms.

According to Allen, Graupera & Lundrigan (2010, pp. 2), home computers are outnumbered by the amount of cell-phone subscriptions in the world. Taking into account the vast number of different devices and platforms, cross-platform development is greatly required in order to accommodate the needs of as many users as possible (Fling 2009, pp. 65 and Allen, Graupera & Lundrigan 2010, pp. 6). Billi et al. (2010) explains that: “One of the main challenges faced by current Information and Communication Technology […] is making all kinds of information and services accessible and usable by all possible categories of users through mobile devices […]”.

Allen, Graupera & Lundrigan (2010, pp.10) explain that it is possible to take use of different frameworks that will allow a user to use HTML, CSS and JavaScript in order to create applications that will run in a web environment. According to Charland & Leroux (2011), an application written in JavaScript will show a decrease in performance compared to one written in a mobile device’s native language, but, that writing an application using JavaScript will benefit the developer in cross platform development by requiring less written code.

Richter et al. (2006) emphasizes that it is hard but important to maintain a similar user interface when developing for cross platform deployment.

Development concerning native and web-based applications will be covered later in this chapter. However, for developers there are many different tools and frameworks that can be used in order to aid in the development of cross platform applications. The next following pages will provide a short introduction to some of these tools.

2.2.1 HTML5

HTML5 is an emerging standard for HTML and is the result of cooperation between the World Wide Web Consortium and the Web Hypertext Application Technology Working Group (W3schools [2], 2012). As time goes by, browsers will incorporate more and more support for HTML5, but at this point no browser has full support, still, HTML5 provide users with new tools for development such as improved error handling, more markups, APIs and a lot of new elements (Pilgrim 2010, pp.4 and W3Schools [2], 2012). A new element that will be useful is the new video tag which according to Pilgrim (2010, pp.18-19) eliminates the
need for additional plug-ins in order to play a video or audio file, if of course, the browser has support for that element. In addition to the video element, Pilgrim (2010, pp. 25-26) explain that HTML5 also provide users with several new input types such as number, URL, email, date and a few others which will automatically validate user input (W3Schools [2], 2012) and on mobile devices, bring up a keyboard layout matching the type (For example, if the type number is used, a mobile device will display the number keyboard for input) (Pilgrim 2010, pp. 151-154).

According to Hoy (2011) HTML5 provides a new way of creating online applications using new features such as drag and drop, local storage, client-side SQL databases and offline caching. Offline caching means that the files and contents of a webpage are downloaded to a device making the page load faster in the future and also makes it accessible even if the user is not connected to the internet (Stark[2] 2010, pp. 91). Hoy (2011) explains that such functionalities are important for users with mobile devices. HTML5 also provide developers with the new touch API which will allow an application to utilize touch events (Lubbers, Albers & Salim, 2010, pp. 265-266).

The new geolocation API, when used, will find out what a user’s current location is (Hoy, 2011). Another API is the canvas API which according to Lubbers, Albers & Salim (2010, pp. 26) provide a developer or user the ability to apply graphics, lines, text or even draw on a canvas element. Both the geolocation and canvas APIs are supported in all major browsers (W3Schools [2], 2012). When it comes to using HTML5 for mobile development, Clark (2010) emphasizes that that standards and web technologies offer a substitute for native development but that it will take time before replacing it.

In development for mobile devices, one application platform that is based on HTML5 is PhoneGap. HTML5 for mobile application development has some limitations. One of the main concerns is the access to hardware components such as camera. However there are a couple of frameworks that can be used in order to tackle these limitations (Christ 2011).

2.2.2 PhoneGap

PhoneGap is an HTML5 platform which according to Stark [2] (2010, pp. 115) enables a user to develop applications using a mobile devices native functions. Allen, Graupera & Lundrigan (2010, pp. 4) explain that native application development require developers to utilize different languages in order to facilitate the requirements of application development for diverse platforms.

PhoneGap eliminates these limitations by allowing a developer to create a native application using HTML, CSS and JavaScript and then call native APIs such as the capture API (camera, video and audio) using plain JavaScript (PhoneGap, 2012 and Allen, Graupera & LaMarche, 2010, pp.131). In addition to providing these functionalities, PhoneGap also provide a service called PhoneGap Build where developers can upload their files and receive functioning applications for the supported platforms (PhoneGap, 2012).
There are also multiple plug-ins and tools for PhoneGap. Plug-ins that has been created by the developer community is accessible at GitHub while development tools can be found on the PhoneGap homepage. PhoneGap currently support development for Android, Windows Phone 7, iOS, HP WebOS, Bada and Symbian, however, there are differences in API support between these platforms (PhoneGap, 2012).

PhoneGap further states that:

    PhoneGap is an open source implementation of open standards. That means developers and companies can use PhoneGap for mobile applications that are free, commercial, open source, or any combination of these. (PhoneGap, 2012)

Although PhoneGap was used in the prototype development for this thesis, another platform that delivers similar functionalities is Titanium Mobile SDK from Appcelerator.

2.2.3 CSS3

CSS3 is the latest standard of CSS and provide a lot of new functionalities which developers can use to style their applications. However, W3Schools [3] (2012) explains that CSS2 always will be supported. They also explain that the functionalities of CSS3 have been divided into different modules and provide the following list of the most important ones.

- Selectors
- Box Model
- Backgrounds and Borders
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

From the list, one can see that CSS3 provides a lot of new features, some of which has been used in order to create the prototype for this thesis. No browser currently supports all the new features of CCS3, although Google Chrome and Safari WebKit powered browsers are very close (W3Schools [3], 2010).

2.3 Mobile development: web VS native

When developing mobile applications, one can choose to develop a native application, a web application or a hybrid application and where Christ (2011) state that: “Applications developed with a hybrid solution are native applications that use web technologies in place of a programming language […].” It is this type of application that has been developed for this thesis. He further explains that developing a hybrid application provides developers with a lot more control over application design than one would when developing native or web applications.
2.3.1 Web applications

Stark [2] (2010, pp. 1) explains a web application as an application that is accessible on the web through a URL, is not installed on a device and is not written in a devices platforms native language. Charland & Leroux (2011) emphasize that application development using web-based technologies despite some of their negative differences from native application development should not be discarded. They also emphasize that JavaScript will show a decrease in speed performance against native languages (due to the fact that native code is compiled) but can put less pressure on a devices CPU and therefore lower battery cost.

When developing web applications, native functions of a mobile device are accessible through the use of JavaScript in a mobile platform’s WebView (Charland & Leroux 2011). This is the technique that the platform PhoneGap which was introduced earlier uses. Fling (2009, pp.145) explains that the mobile web uses the same protocols and standards making it a useful platform for mobile devices.

Web applications offer a lot of positive sides for developers, but as mentioned earlier, one can also choose to develop native applications.

2.3.2 Native applications

According to Stark [2] (2010, pp. 1), a native application is installed on a device and is written in the devices platforms native language. Charland & Leroux (2011) along with Fling (2009, pp. 146-150) emphasizes that, while native development is not the wrong way to go, using this approach comes with complications such as having to use different programming languages for each platform (writing the same application in different languages will also require more time, cost and maintenance), coping with differences in development environment and available tools and also limit the number of users that will be able to use the application. They provide a list (shorted) of platforms and their native languages:

- Apple iOS - C, Objective C.
- Google Android – Java (Harmony flavored, Dalvik VM).
- Windows 7 Phone - .NET.

However, there are some positive sides to native development. As mentioned in the previous section. Charland & Leroux (2011) state that native code is compiled, which makes it faster than non-native and also mention that using native code will give a developer more freedom when trying to access native APIs (Platforms such as PhoneGap does not currently support all features on all possible mobile platforms (PhoneGap, 2012).).

2.4 Usability

Usability is important in development because it will lay the foundation on how the users will experience, perceive and handle a product (Fernandez, Insfran & Abrahão, 2009). According to Conte et al. (2007), usability is explained by ISO 9241 as: “the 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”.

Fernandez, Insfran & Abrahão (2009) present a proposal in SQuaRE (Software Product Quality Requirements and Evaluation) to divide usability into different following components:

- **Learnability**
  - Predictability, help facilities, informative feedback
- **Understandability**
  - Legibility, readability, familiarity, brevity, user guidance
- **Operability**
  - Execution facilities, data validity, controllability, capability of adaption, consistency
- **Attractiveness**
  - Font style, background color, position of elements
- **Compliance**

These components are a proposal for dividing usability into smaller parts. By doing this, one can look at different layers of components of usability and in by doing so get a better understanding of the different parts of usability and with that reach higher quality end results in development.

Billi et al. (2010) emphasizes that usability is important in cross-platform application development but state that: “[...] usability cannot be considered as encompassing all the possible problems encountered by every user.” However, according to Harms & Schweibenz (2000), usability should be considered key in development and will according to Vogel et al. (2010) benefit both users and developers.

Evaluation of usability can be performed in two separate ways. These are according to Biel, Grill & Gruhn (2010) heuristic and usability evaluations. They explain that in heuristic evaluation: “[...] the interface is evaluated against guidelines, style-guides, design rules, and the experience of the evaluator.” and that usability evaluation which has been used for the user testing in this thesis: “[...] focuses on the usage and usage related problems.” Casteleyn (2009, pp. 272) emphasize that simply following current guidelines does not make a product usable and commend performing tests with users to ensure accurate results of usability. This was done for the mobile application prototype that was developed for this thesis.

An example of this can be seen in a usability test carried out by Billi et al. (2010). They found that some users had problems using radio button elements because of their small size and that text that that was to close together make it more difficult for users to comprehend.

Mobile applications are being used by a number of different projects in the world. The following section will introduce some of these projects.
2.5 State-of-the-art projects

2.5.1 The ODK platform

Anokwa et al. (2009) explain that the Open Data Kit (ODK) are tools aiding users in the field of mobile data collection by providing functionalities that enable users to produce their own forms, use those forms in order to collect data and then retrieve that data from a server. They also explain that the ODK collect renders XForms which a technology for creating forms. There are several ODK projects which are listed on the OpenDataKit (2012) homepage:

- ODK build is an HTML5 based application that provides users with quick way of creating forms by using drag-and-drop functionalities.
- ODK collect is built for Android and support such input features as text, media (pictures, video and audio), barcode and geolocation.
- ODK Aggregate allow users to store, view and export collected data.’
- ODK Form Uploader allows users to upload collected data
- ODK Validate helps users validate their created forms.
- ODK XLS2XForm provides users with the option to create XForms using Microsoft Office Excel.

The ODK platform is used by the LETS GO project for mobile data collection in studies concerning inquiry base learning.

2.5.2 Learning Ecology through Science with Global Outcomes (LETS GO)

LETS GO is one of the current research projects of the Center for Learning and Knowledge technologies (Vogel, 2012) and is cooperation between Stanford and Linnaeus University. The project focuses on creating: “science learning collaboratories” and are developing mobile oriented technologies that will aid in the collection, analysis and sharing of environmental science data (Vogel, 2012).

Two of the current tools that have been developed by and for the project are a visualization tool for displaying and organizing collected data (Vogel et al. 2011) and a mobile application for data collection using Windows Mobile 6.0 (Vogel et al. 2010). Later in this project the ODK platform was integrated with an idea of integration of open standards in relation to data collection, aggregation and visualization. However, one thing to be noticed in this project is the mobile data collection tool, which is limited only to android native platform.

By using mobile devices as a data collection tool, high school students who are connected to the project can go outside and collect environmental data using different available mobile sensors for later analysis (Vogel, 2012).

2.6 Identified requirements

The requirements in table 2.1 were identified from the Foundations part in section 2 and the LETS GO project. They were used as a guideline for the design, implementation and testing
of the prototype. The column furthest to the right shows the source of a requirement along with the prototype version where the requirement was addressed.

<table>
<thead>
<tr>
<th>Requirement code</th>
<th>Requirements</th>
<th>Description</th>
<th>Source of requirements and prototype where addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Cross platform deployment</td>
<td>The prototype will be built for deployment on three different mobile platforms</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: all prototypes</td>
</tr>
<tr>
<td>R2</td>
<td>Open standards</td>
<td>The prototype will load and render XForms and present them as HTML forms</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: prototype 1</td>
</tr>
<tr>
<td>R3</td>
<td>Web standards</td>
<td>The prototype will be built using standards such as HTML5, JavaScript, CSS3 and XML</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: all prototypes</td>
</tr>
<tr>
<td>R4</td>
<td>Code reusing</td>
<td>The code of the prototype between the platforms will be reused and have as minor changes as possible</td>
<td>Source: Foundations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: all prototypes</td>
</tr>
<tr>
<td>R5</td>
<td>Save data</td>
<td>The prototype will save collected text data in an XML file and sensor data in a folder that is connected to a form instance</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: prototype 2</td>
</tr>
<tr>
<td>R6</td>
<td>Capturing media</td>
<td>The prototype will allow a user to capture data such as:</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: prototype 1,2,3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Images</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Video</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Audio</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Location data</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Visual codes</td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td>Multiple forms</td>
<td>A user will be able to choose between different XForms</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: prototype 2</td>
</tr>
<tr>
<td>R8</td>
<td>Upload</td>
<td>The prototype will allow a user to upload collected data to a receiving PHP file on a server</td>
<td>Source: LETS GO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prototype: prototype 3</td>
</tr>
</tbody>
</table>
| R9 | Notifications | The prototype will inform the user about occurred problems and requirements when filling out a form | **Source:** Foundations  
**Prototype:** prototype 3 |
|---|---|---|---|
| **R10** | Touch navigation | The prototype will allow a user to navigate the form using the mobile devices touch screens | **Source:** Foundations  
**Prototype:** prototype 3 |
| **R11** | Offline | The prototype will not require an internet connection in order to load an XForm (local) | **Source:** LETS GO  
**Prototype:** prototype 3 |
| **R12** | An easy to use application | User testing of the prototype will be done with ten users | **Source:** methods for user testing |

**Table 2.1** Identified requirements for prototype development

These requirements represent what the final prototype should provide and are the foundations on which the final prototype which will be explained in section 4 was built upon.
3. Methods used for prototyping

Prototyping can be found in many different forms. According to Houde & Hill (1997) something as simple as a brick can be called a prototype. Fling (2009, pp. 103) emphasizes that prototyping can help a project by saving in on resources such as time and money. But to cover prototyping as a whole it could be explained as the creation of basic versions of a non existing product in order to test or explore possibilities and issues, both technical and design oriented, with the aim of optimizing a product and reaching the end goal of a development process. This is my own interpretation influenced by Budde et al. (1992) and Houde & Hill (1997).

Budde et al. (1992) emphasizes that when creating prototypes, one is empowered with the ability to test them against users with the goal of discovering issues and then use that information to make improvements before creating a product. Furthermore, he states that:

> The application system is built on the basis of an accepted prototype. The prototype serves specification purposes only and is not used as a building block in the application system itself: it is a "throwaway". (Budde et al. 1992)

There are several different approaches of which some are mentioned by Houde & Hill (1997) that can be used when creating prototypes.

- **Role prototypes**
  - Focuses on assessing functionality needs of the users
- **Look and feel prototypes**
  - Focuses on assessing user interface with no regard for technical functionalities
- **Implementation prototypes**
  - Focuses strictly on assessing methods and requirements for implementation

The final approach is **integration prototype** which has been used in the development of the prototype for this thesis. It is a combination of the three previous explained approaches and covers all aspects of a development process. Because of that, this approach will require more time and effort than the others.

When commencing a development process it is important to have some sort of structure or guidelines that can be followed in order for it to go as smooth as possible. Prototyping traditionally consist a few different stages mentioned by Casteleyn et al. (2009, pp. 58):

- Requirements gathering
- Design
- Implementation
- Testing and evaluation
- Deployment
- Maintenance
- Evolution
These stages will help a development process to maintain a structure and has been used as guidelines along with a model proposed by CMS (2008) in the creation of the model shown in figure 3.1. This is an iterative model that was used in the development process of the mobile data collection prototype. The iterative approach was chosen because requirements can be adapted or changed as the development progress is ongoing. Each step will be explained below.

**Figure 3.1 Prototyping model**

- **Initial investigation**
  At this stage in the development process, a wide assessment of what the application should be able to do was made. It also included the decisions of what programs and languages to use for development, along with researching the selected development languages and other technologies. Information gathered from this stage was later used when gathering requirements.

- **Requirements gathering**
  At this stage, gathering of the requirements was made. Requirements were gathered by assessing where the application should work and what the application and users should be able to do with it (in relation to the LETS GO project and the Foundations part in section 2). As can be seen in Figure 3.1, requirements can be extended as the development process progresses.

- **Application design**
  This is the stage where decisions were made about how the application should be developed along with determining how it should look. Like the requirements gathering, application design can change as development progresses.
Coding

This is where the code of the application was written by taking into account the previous two stages. Testing was done simultaneously as coding progressed.

Testing

This stage is very important in order to make progress in the development process and to solve/discover different problems that occurred during development. Due to a lack of time there was no testing with users during the development process. Therefore, the only testing that was done with users was after the third prototype was complete.

- Implementation
  At the implementation stage, the finished prototype was implemented on all three mobile platforms in order to be put through the usability testing.

- Usability testing
  Usability testing was done in order to get users inputs and comments about the application along with discovering possible usability problems and other issues.
4. Design and prototyping

This section will show the design and development process from the first to the final prototype. It will also give an overview of problems that occurred during development and also figures to illustrate different parts of the application.

Of the three prototypes that were developed, the focus of the first prototype was the rendering of the forms in order to find out if it was possible to present an XForm as an HTML form. The second prototype focused mainly on allowing users to interact with the prototype and collect data. This included user inputted text, sensor capturing and saving of data. When developing the third prototype, the main part focused on was the uploading of data, but also separating code into different files, adding the functionality to read visual codes and using local XForms. This was done along with a lot of improvements and changes concerning styling and functions such the change from button navigation to touch navigation. The prototypes were built using standards in accordance with requirement R2 and R3.

4.1 Initial implementation of the prototype

When commencing prototyping, requirement R2 was the first to be addressed in order to create a base for the other requirements. As such, the first prototype rendered the different groups of an XForm along with the corresponding inputs and supported functionality for retrieving location information using the HTML5 geolocation API in accordance with requirement R6.4. At this point there was no CSS styling.

![Figure 4.1](image_url) Rendered XForm in initial prototype

Each field that is seen in figure 4.1 represents a group in the XForm that was created for initial testing. At this point, development using PhoneGap had not begun, and therefore, this prototype did not offer the possibilities of utilizing the native functions of a mobile device.
During this part of the development process, there was no critical problem that had to be addressed. However, rendering the different parts of the XForm required a lot of testing before functioning properly. This prototype was only able to render the XForm that is seen in figure 4.1 and later required a lot of changes in the code in order to work with other XForms. After requirement R2 had been addressed, knowing that it was possible to render and present XForms, a second prototype was created with the target of dealing with the capturing of sensor data along with a few other requirements.

### 4.2 Second prototype

At this point in development, requirement R7 was addressed in order for users to be able to choose different forms. Along with this, a lot of changes concerning the rendering of the XForms were also fixed. As for the styling, each group in an XForm was now showed separately and had buttons for navigating between them along with an animation when navigating between the groups. While this animation worked for Android and iOS, it did not for Windows Mango. As shown in figure 4.2, each group now also has a legend at the top.

![Figure 4.2 Question screen of second prototype on Android](image)

After fixing the navigation, the validation functionality to check the required fields in order to see if an input field had been filled out or if a button had been clicked was created. This prototype was the starting point of development using the mobile platform Android and supported (in addition to the geolocation) requirement R5 of creating required folders, capturing and moving sensor captured data to the corresponding folder of an instance and saving an XML file containing keyboard inputted data. It also supported requirements R6.1-3 of using the camera and audio sensors. As the prototype was now able to interact with the Android devices file system and allowed users to interact and capture/input data. A third prototype was commenced with the aim of uploading the collected data.
4.3 Third prototype

To this point in development, all code had resided inside index.html and was therefore split up in to different components which can be seen below. The third prototype was implemented on all three platforms shown in figure 4.3 and therefore required some changes in code in order to work.

**Files:** Index.html, master.css, createDirectories.js, dateTime.js, saveXML.js, phonegap-1.4.1.js, navigation.js, upload.js, geo.js, captureAndMove.js, only Android and iOS (swipe.js, barcodescanner.js)

For this prototype, changes in the styling include the CSS3 font-face rule which provides the possibility to use of external fonts for text and also changes in styling for input elements animation type when navigating. At this point, requirement R10 for users to be able to navigate using touch was addressed, the buttons shown in the middle of figure 4.3 was replaced with touch navigation for Android and iOS. Requirement R10 was not able to be met for Windows Mango because of issues concerning the support of the HTML5 touch API.

![Figure 4.3 Multiple choice question (radio buttons) on iOS, Windows Mango and Android](image)

In order accommodate requirement R8 of users to be able to send their collected data, a function for uploading saved files to a server was implemented along with a PHP file to receive the uploaded files. At the same time, buttons that will allow a user to navigate back to the start screen from several places in the prototype was added. A user can also go back when starting a form by pressing the back button or sliding backwards on the first group that appears.

When capturing media, each button now allowed capture of one media file which will be overwritten if the user chooses to re-capture media. The choosing of forms implemented in the second prototype was because of requirement R11 changed to using local XML files
instead of loading them from a server for Android and iOS. However, requirement R11 could not be resolved for Windows Mango because of security blocks and other issues that could not be resolved. Therefore, Windows Mango still loads forms from a server.

Figure 4.4 Last screen of a form on iOS, Windows Mango and Android

In case a user is not connected to the internet in accordance with requirement R11 and wants to send data at a later point, a simple user interface for uploading previously saved data is accessible from a button on the start screen. Another button on the starts screen allows a user to view and delete currently saved data. This function was added to avoid a massive list of saved forms when wanting to upload a previously saved form. In addition to these implementations, a few improvements concerning validation, rendering, error handling and requirement R9 of user notification were also addressed.

Figure 4.5 Barcode capture screens on iOS and Android
PhoneGap in its self does not support the functionality of a barcode reader. However, there are PhoneGap plug-ins that made it possible for the application to read visual codes on Android and iOS shown in figure 4.5 meaning that requirement R6.5 was not met for Windows Mango.

As the requirements now had been met and the prototype was able to present the XForms, allow users to interact with the prototype, collect data and also upload that data to a server, the prototype was now ready to be tested with the users.

Table 3.1 Summarizes and provides an overview of the different stages of the development process and shows the changes in functionalities between the different prototypes.

<table>
<thead>
<tr>
<th>Version</th>
<th>Render</th>
<th>Capture and save</th>
<th>GPS</th>
<th>Barcode</th>
<th>Upload</th>
<th>Touch</th>
<th>Notice</th>
<th>Offline</th>
<th>Choose form</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>iOS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Windows Mango</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.1** Supported functionalities during development process

### 4.4 Encountered problems and differences during development

HTML5 multiple file upload is a new function which allow users to select multiple files and upload them. When attempting to implement this for the prototype it was discovered that it works in the standard browser in the Android device (although, for mobile devices it is only possible to choose one file at a time, which takes way the whole point of multiple file upload), but not in the WebView. Had the function worked it would have required more work of users and work against usability instead of towards it. Instead, PhoneGap's file transfer function where all the files in a folder can be uploaded was implemented.

The final touch navigation functionality required a user to slide their finger over the text and arrows seen in figure 3.6. This was the solution that was created after initial attempts for being able to navigate by sliding anywhere on the screen were unsuccessful.

Using PhoneGap to upload files was easy. However, when attempting to upload there was no indication that the files were uploading. This was later found to be an issue of limited bandwidth and large file sizes causing a long upload time which was interpreted as the function not working. While PhoneGap offer a lot of functions to aid in development, it can be hard to resolve thrown errors. As of now, there is no current documentation of error codes.
there is however a large developer community where developers are more than happy to help in resolving errors such as ‘17’.

At first, XMLHttpRequest was used in order to load external XForms. At a late point in development, the XForms were instead added to the application in order for users to be able to use the forms without internet connection. However, for Windows Phone there is security blocks which prevent making local requests or access the files in the applications folder, therefore, Windows Mango still loads external forms.

Working with cross platform development resulted in some differences in the prototyping and with the code for the platforms used. Since the platforms use different file systems there was a lot of work with finding the correct referencing paths which can be seen in Appendix C. The platform that was easiest to get started with was Android where coding was done using Eclipse. Android is followed by Windows Mango where there is quite a lot of work with registration and connecting different accounts. Code was written using Microsoft Visual Studio 2010 Express for Windows Phone which was easy to work with. However, it required files to be added as content in order for them to work with the application which could be confusing for someone that have not used this program before. Another problem when testing the application is that the camera does not work when device is connected via USB and the Zune software (required for developing) is open, however, capturing video and audio works. The platform that was most complicated to develop for was iOS where code was written in Xcode. This is because there was a lot that needed to be done concerning certificates and different settings.

Highlighting an input field with type number on Android opens up the number input interface preventing users from entering text data. On iOS, the number input interface is opened, but users can still change back to the letter input interface enabling them to enter text data. On Windows Mango, the number input interface does not open up at all. In order to prevent users from entering text data in numeric fields on Windows Mango and iOS, extra validation functionalities were required.

Differences in file systems between the platforms resulted in some differences in functions required. In Android, video and images are saved in the same directory which made it possible to use only one function when capturing these lowering the code used. This was not possible for Windows Mango since video and images are saved in separate directories which instead resulted in an extra function. This was also done for iOS because images and audio are saved in the same directory and in order to separate sound from viewable data. Capturing video in iOS resulted in the files being saved in new folders each time. Therefore, in order to be able to reference the path, an extra function that removes the folder after each capture was added. Appendix C shows the file formats used in the different platforms.
4.5 System overview

The system consists of the different steps that can be seen in figure 4.6. The first thing that happens is the loading of the chosen XForm which is then rendered to an HTML form. User can then collect their data using normal text input and mobile sensors. The collected data will be geotagged since the system allows retrieving of coordinates. After all data has been collected, inputted text data is saved as XML which can then be uploaded to a server along with media captured from sensors.

**Figure 4.6** Components overview of the developed system

While figure 4.6 shows how the system works, the next section will show figures of the application flow and what choices can be made during a data collection.
4.6 Application flowchart and use case diagrams

The flowchart in figure 4.7 shows how the different part of the application interact with each other and what choices that can be made by a user when using the prototype. From the start screen, a user has three different options: choose form, send form and delete forms. When a user chooses send form they will be presented with a list of previously saved instances which they can upload. If a user goes to choose form they will be presented by a list of available forms which they can use. When choosing a form, the application will render it and show an animation followed by the first question. A question of the form can be required or not required and is presented following an animation. If a question is required, a user has to complete it in order to continue to the next question. If a question is not required, the user can choose to either complete the question or directly go to the next question. When a form is complete a user can save the collected data, upload it or go straight to the start screen.

![Flowchart Diagram]

Figure 4.7 Mobile application prototype flowchart
The use case in figure 4.8 provides a simple overview of how a user interacts with the prototype when commencing a data collection instance. A user chooses the form which is then rendered and presented by the prototype. A user then inputs text data and uses the functionalities provided by the prototype to collect sensor data. When the form is complete, a user uploads the data to a server where it is stored.

Figure 4.8 Use case 1
As can be seen in figure 4.9, users can choose a form that is then loaded by the prototype that also provides a user with the capabilities of collecting sensor data in addition to text data. Sensor capabilities provided by the prototype are microphone, camera and GPS. When using the camera and microphone, media is captured and then saved while GPS (location data) is inputted automatically in a text field. When users has collected all the necessary data, they can save it and upload it to a server where a PHP file receives and sorts the collected data into separate folders for video, audio, images and XML files.

**Figure 4.9** Use case 2
The use case below in figure 4.10 shows the possible choices for a user when performing a data collection instance. As the figure shows, the arrows with the text "<<uses>>" mean that a user can choose and use the different functionalities such as choosing forms and collecting, saving and uploading data. The arrows with the text "<<extends>>" means that a child element is an extension of its parent element. For example, the collecting of data is extended by collecting either text data or sensor data. Extensions of collecting sensor data are the retrieving of GPS location, capturing audio and using the camera. The camera itself also has a few extensions showing the different uses of the camera. As shown in the figure, the camera can be used to capture images, video and visual codes.

![Use case diagram](image)

**Figure 4.10 Use case 3**
4.7 Supported features of used platforms

When developing applications or websites that are to work with different platforms or browsers it is important to consider the possibility that everything might not work in the same way. Hence, during development of the prototype, there were some differences in support of new elements and functions in HTML5 and CSS3 between the used platforms.

Figure 4.11 will show these differences in used elements and functions together with current PhoneGap and PhoneGap plug-in support. Please note that the figure shows the current support for the used platforms that were discovered during development.

As the table shows, the platforms that currently has best support for HTML5 and CSS3 are Android and iOS while Windows Mango only support some of the used elements and functions. Windows Mango uses internet explorer which is not powered by WebKit like iOS and Android. When testing HTML5 and CSS3 browser support, the WebKit browsers passed over 50% of the new features in CSS3 and over 70% in HTML5 while Internet explorer only passed 30 % of CSS3 and 27% of HTML5 (CSS3test and html5test).

<table>
<thead>
<tr>
<th>Supported elements and functions</th>
<th>Android</th>
<th>iOS</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fieldset</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Number</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geolocation</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Button</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Touch event</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CSS3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Border-radius</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Box-shadow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Font-face</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Animation</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Keyframes rule</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Border-radius</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Gradient</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Appearance</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PhoneGap</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accelerometer</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Camera</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Compass</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Contacts</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>File</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Geolocation</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Media</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Network</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Notification (alert)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Notification (sound)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Notification (vibration)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Storage</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PhoneGap plug-ins</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barcode reader</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.11 Supported elements and functions
5. Methods for user testing

There are a few different types of usability evaluations that can be used for testing a product. The one that will be used in this thesis is usability testing which allows one to observe and collect data from intended users. Data collected during testing can then be used when attempting to assess the usability of the tested product. This data normally include:” […] user execution type, number of errors, and user satisfaction” (Casteleyn et al. 2009, pp. 273). Conte (2007) emphasize that testing with users will not only disclose problems in usability, but also provide user input for further improvements.

When trying to decide in what environment a test is to take place, Kaikkonen et al. (2005) mentions some differences between testing a product in its real context (for example, testing an outdoor running shoe outside) or testing out of context (for example, testing the shoe in a laboratory on a treadmill). They emphasize that both ways have their own positive and negative aspects and state that testing out of context provides: “[…] a peaceful space, where a test user can concentrate on the given tasks” but lacks some variables that could affect data results. They mention that testing in context provide these variables, but can be harder to execute depending on the methods used for data collection.

During a user test for a mobile application for assessing the differences in these approaches, Kaikkonen et al. (2005) discovered that users being tested out of context were more frustrated but also that there were no major changes in performing tasks between the two settings, although testing in context required more time. Their conclusion was that testing out of context takes less time and provides valid results that are enough to evaluate usability of a product.

Casteleyn et al. (2009, pp. 273-374) mentions the importance of being well prepared when executing user tests and explains that one need to:

- Be clear what the goal with the user test is
- Be clear on who will be testing the product
- Create “real” testing scenarios for users
- Determine how usability should be measured
- Be well prepared on the day of testing

This information provides a small theoretical background to methods of user testing and will be used as a guideline for user testing of the data collection prototype. Below I will explain the methods that were used for testing.

This test utilized both quantitative and qualitative approaches where data was gathered data by observing the test users and collecting data from a questionnaire that the users were asked to fill out. By taking into consideration what Kaikkonen et al. (2005) explains, ten users were recruited in order to test the application. This was added as requirement R12. In accordance with Casteleyn et al. (2009, pp. 273), the data collected was task execution time, errors and
satisfaction (task difficulty) along with other user input that were written in the questionnaire or observed during testing.

The testing was performed on three platforms: Android, Windows Phone and iOS. Consideration was taken that some of the tasks performed by test persons could be harder to perform if the user has not previously used a platform or device. For example, the task of capturing pictures is not the same between the platforms. Therefore, an introduction to some of the native features of the platforms was created and shown for the test persons before starting the test.

According to Kaikkonen et al. (2005), there is no significant difference between testing in or out of context (environmental setting). With this in mind, along with easier collection of user data, testing out of context was chosen for the user tests of the mobile application prototype. This means that users did not go outside to collect data but sat down in a laboratory environment when they performed the tests. It also means that the users did not collect environmental data in accordance with the LETS GO project, but more tested the functionality and feel of the prototype. Furthermore, users were informed about how the test was to be conducted before it started.

5.1 Tasks for user testing

The following table shows the tasks that were created for user testing of the application. These tasks were performed using the prototype implementation especially created for testing by all users on all three mobile platforms that are focused on in this thesis.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Please go to ”Choose Form” and load the form called testForm.</td>
</tr>
<tr>
<td>2</td>
<td>Capture location data (GPS) by pressing ”Get Coordinates” button.</td>
</tr>
<tr>
<td>3</td>
<td>On the following screens, please capture Media (Picture, Video and Audio).</td>
</tr>
<tr>
<td>4</td>
<td>Please navigate back and forth using ‘← navigate →’ function and re-capture location data (GPS). Now, re-capture the picture and continue where you left the last activity.</td>
</tr>
<tr>
<td>5</td>
<td>On the two following screens, please enter text and numeric values. <strong>But before entering your values, please try to navigate forward.</strong></td>
</tr>
<tr>
<td>6</td>
<td>On the screen with the radio buttons, please choose an option. After, on the next screen (checkbox buttons), please choose at least two options from the presented list.</td>
</tr>
</tbody>
</table>
Table 5.1 Tasks created for user testing

The test persons were instructed to fill out a questionnaire containing questions that were related to the tasks they had performed along with some general questions about their experience of the prototype and what mobile platforms they had previously used.

5.2 Questionnaire

The questionnaire (appendix A) used a five choice Likert scale for measuring degree of difficulty for each task in accordance with Vogel et al. (2011). It also contained the following questions:

- What was your overall experience of the prototype (such as ease of use, design and functionalities, understandability)? (Platform specific)

- What is your overall experience of the prototype over all platforms? (Differences concerning style, performance/functionalities, ease of use)

- Please give your thoughts on how the prototype could be improved.

- What mobile platforms have you previously used?

- Other thoughts.

The questionnaire did not require test persons to disclose any personal information about themselves except gender and age. Furthermore, any input in the questionnaire about problems they experienced that are related to the mobile device or its native features and not the application itself will not be included in the results.
5.3 Participants and settings

Testing was conducted over two days with a total of ten users whereof seven male and three female. The age span of all users ranged from 22 to 33 giving an average age of 24.9 with a standard deviation of 3.4. The testing took place in a laboratory environment at Linnaeus University where a total of six sessions was conducted with one or two users at a time and took about one hour each to complete. Figure 5.1 shows some of the users performing the testing of the prototype.

![Figure 5.1 Users performing testing of the prototype](image)
6. Results of user testing

Results of the user testing will be shown below using test users input on the Likert scale (description of values can be seen in Appendix A), task completion time, answers for questions in the questionnaire and observations made during the testing. From a question in the questionnaire we are able to see the user’s previous used platforms which are displayed in figure 6.1. The bar furthest to the right (other) are mobile devices that are not focused on in this thesis.

![Previous mobile platform usage](image)

**Figure 6.1** Previous used platforms by test users

The order of which the platforms were used in the tests was randomized. However, from the uploaded forms we are able to see in what order they were used. This order is shown below in table 6.1.

<table>
<thead>
<tr>
<th>Users</th>
<th>First platform</th>
<th>Second platform</th>
<th>Third platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>Android</td>
<td>Windows</td>
<td>iOS</td>
</tr>
<tr>
<td>U2</td>
<td>iOS</td>
<td>Android</td>
<td>Windows</td>
</tr>
<tr>
<td>U3</td>
<td>iOS</td>
<td>Android</td>
<td>Windows</td>
</tr>
<tr>
<td>U4</td>
<td>Android</td>
<td>Windows</td>
<td>iOS</td>
</tr>
<tr>
<td>U5</td>
<td>Windows</td>
<td>iOS</td>
<td>Android</td>
</tr>
<tr>
<td>U6</td>
<td>Android</td>
<td>iOS</td>
<td>Windows</td>
</tr>
<tr>
<td>U7</td>
<td>Windows</td>
<td>Android</td>
<td>iOS</td>
</tr>
<tr>
<td>U8</td>
<td>iOS</td>
<td>Android</td>
<td>Windows</td>
</tr>
<tr>
<td>U9</td>
<td>iOS</td>
<td>Windows</td>
<td>Android</td>
</tr>
<tr>
<td>U10</td>
<td>Android</td>
<td>iOS</td>
<td>Windows</td>
</tr>
</tbody>
</table>

| Android total | 4 | 4 | 2 |
| iOS total     | 4 | 3 | 3 |
| Windows total | 2 | 3 | 5 |

**Table 6.1.** Platform order during user testing
6.1 Average time and score

During testing, users were asked to fill out the time it took them to complete a task. Figures 1, 2 and 3 in appendix E will show the average time spent on each task for all the users combined on the different platforms while figures 4, 5 and 6 shows the average score for each task given by the users in the Likert scale of the questionnaire. Since users filled out the time in hours and minutes (using the mobile platforms native clock), the average time will only provide some idea of how long a task took.

Figure 1 in Appendix E shows the average time in seconds that user spent on each task when performing testing on Android. As one can see from the figure, task 1 did not take any time at all for users to complete. The average time spent on task 2 was 24 seconds with a standard deviation of 30.9. Task 3 required users to capture media on three different screens and was expected to take more time than other tasks, and did so on all platforms which can be seen by looking at figure 2 and 3 in Appendix E. The average time for task 3 was therefore 84 seconds with a standard deviation of 41.9.

Task 4 also was also expected to result in a higher average value since users had to navigate to the beginning of the form to re-capture coordinates and picture and then navigate back. The average time for task 4 was 42 seconds with a standard deviation of 28.9. Task 5 had an average time of 30 seconds and a standard deviation of 31.6. The average time for task 6 was 24 seconds with the same standard deviation as task 3 of 41.9 which could be a result of users having problems choosing their values or knowing that they could scroll down (this was a miss in the design and will be further explained later). Task 7 had an average time of 30 seconds and a standard deviation of 31.6. By combining these values, we can tell that the average time for each task on Android is 33 seconds and that the average total time for completing the form was 3 minutes and 54 seconds.

As one can tell from figure 4 in Appendix E, users rated the different functionalities of the prototype to be above neutral. While users agreed that most tasks were easy to perform, task 4 and 6 shows a lower average score with a standard deviation of above 1.0 and therefore lower usability than the others. This concludes with the user’s commentary and with the observations made during testing. Task 4 required the users to navigate back several screens in order to re-capture the coordinates and the picture and then navigate back to the screen where they started the task. The lower score could be a result of some users finding the response on Android to be slow when attempting to navigate and clicking buttons which will be explained further in the next section. However, in order to provide accurate results for this issue, measuring each mobile devices response time would be required.

Task 6 also shows a lower score and required users to select options from presented lists of radio buttons and checkboxes. The lower score could be a combination of users finding it hard to choose options from the lists and navigating forward. Since the lists did not fit on the screen they were stretched out causing the navigation interface to be placed out of screen. Therefore, in order to navigate forward, users had to scroll down in order to see the navigation interface.
Figure 2 in Appendix E shows the average time for completion of tasks on Windows Mango. Task 1 took little time for users to complete. The average time for task 1 and task 2 was 6 seconds with a standard deviation of 18.9. The reason that task 2 had an average time of 18 seconds less than Android could be explained by the fact that the time may have changed just as the users wrote it down. Task 3 shows a bit higher time than Android with an average of 72 seconds and a higher standard deviation of 47.3 which could be a result of the native media capturing functions of the Windows phone.

Task 4 had a lower average time than Android with 36 seconds and a standard deviation of 30.9. Task 5 took longer time to complete on Windows phone than on Android with an average time of 48 seconds and showing a very high standard deviation of 55.1. Although not having mentioned native features earlier, some users during testing mentioned that it was harder to write on the native keyboard of Windows Mango which could be a factor to the higher time.

Task 6 had an average time of 18 seconds with a standard deviation of 28.9. The average time for task 7 was 24 seconds with a standard deviation of 30.9 showing that the task was a bit easier to perform on Windows phone than on the other platforms. However, this could be a result of 8 users performing the test with either iOS or Android as their first platform which would make the task easier when they performed the test on Windows Mango. The average time for each task on Windows Mango was 30 seconds giving an average total completion time of 3 minutes 30 seconds.

Looking at figure 5 in Appendix E, one can see some similarities with the Android platform. However, there was more deviation between the users when giving their scores. As the figure shows, Users found task 1 and 2 to be easy to perform while they did not agree that task 3 and 4 were. Task 3 and 4 shows an average score below 4.0 with a standard deviation of above 1.0 which could be a result of users not having used Windows Phone before and that the native features of capturing media trough the prototype was hard to understand. Users were asked not to base their score on the native features of the mobile devices but may still have been influenced.

The average score of task 4 shows a bit higher average and standard deviation than on the Android platform and is most likely a result of the same problems that occurred there. With an average score of 4.2 and a standard deviation of 0.9 for task 5, most users found the task easy to perform while there was a variation raising the standard deviation. Task 6 also have a high standard deviation of over 1.0 with an average values of 4.0 showing that most user found the task to be easy while some did not. This can be explained by some users finding it hard to understand that they were able to scroll down the multiple choice menus in order to display the navigation interface and navigate forward as on Android.

Shown in figure 3 in Appendix E, the time for task 1 on iOS is a bit higher than the other platforms with an average time of 18 seconds and a standard deviation of 28.9. Task 2 took no time for users to complete. Task 3 had the same time and standard deviation as Windows Mango with an average time of 72 seconds and a standard deviation of 47.3. Task 4 and 5
shows the same average time of 48 seconds with a standard deviation of 37.9 for task 4 and 55.1 for task 5.

Task 6 shows a higher time than the other platforms with an average time of 42 seconds and a standard deviation of 28.9. The time and standard deviation for task 7 is the same as on Android with an average time of 30 seconds and a standard deviation of 31.6. The average completion time for each task on iOS was 36.6 seconds giving an average total completion time of 4 minutes 18 seconds for the whole form.

Figure 6 in appendix E shows the average score for each task on iOS. iOS had the highest average score of 4.2 for all tasks combined. Many users favored the iOS over the other mobile devices that were used for testing both in performance and ease of use of the prototypes functionalities. One can see some similarities of the scoring between iOS and Android which was expected from the beginning.

Just like Android, the tasks that show the lowest average score were task 4 and 6 which comes to show that these are the ones that lower the usability of the prototype and would require most improvements for future development. Task 4 has an average value of 3.5 with a standard deviation of over 1.0 showing that users scored the task quite different. Task 6 had an average score of 4.0 with a somewhat lower standard deviation of 0.8 again showing that users are quite divided when scoring task difficulty.

![Average time for all platforms](image.png)

**Figure 6.2** Average time for performing tasks on all platforms
Figure 6.3 Overview of average task completion time for each platform

Figure 6.2 shows the average time for each task over all platforms while figure 6.3 shows an overview of each platforms average task completion time. As figure 6.2 shows, task 1 had an average time of 8 seconds with a standard deviation of 20.7 and that task 2 had an average time of 10 seconds with a standard deviation of 22.7. The average time for task 3 was 76 seconds with a standard deviation of 44.3. Even though this task was more extensive it shows that the functions for capturing media could use some extra improvements in future development. Task 4 and 5 has the same average time of 42 seconds with a standard deviation of 32.1 for task 4 and 47.6 for task 5. The average time for task 6 and 7 was 28 seconds with a standard deviation of 34.2 for task 6 and 30.4 for task 7.

By calculating the average time to complete each task we get a result of 33 seconds. Average completion time for the form over all platforms is therefore 3 minutes 51 seconds. The results that have been calculated are consistent with my initial assumptions about the time. Task 1 and 2 does not take very long since users only has to press a few buttons while task 3 has a much higher time since it involved three different screens where users had to capture three different types of media.

Task 4 and 5 required users to both navigate, capture a picture and entering values by hand all which takes more time than for example task 6 and 7 where users chose values from presented lists before saving, uploading and deleting their data. From looking at the standard deviations we can tell that task 3 and 5 was quite different among users. However, these tasks required users to use the navigation interface and native functions such as media capture and text input which was explained to them but still could be hard to use if a user had not previously used a specific platform.
Figures 6.4 and 6.5 display the average score for each task on all platforms and an overview of each platform's total average score for all tasks. While all tasks rate above neutral, with most users finding the tasks easy to perform, tasks 3, 4, and 6 have an overall lower score and a standard deviation of above or just under 1.0, indicating that some users found the tasks easier than others. Looking at the comments and observations, these tasks were identified as having most usability problems and tested functionalities such as capturing media, navigation, and selecting options from multiple choice menus. As previously stated, when scoring task 3, some users may have been influenced by how the native media capturing functions worked, lowering the average score.

Task 4 required users to navigate several screens, putting the navigation interface to a big test. Users sometimes found the navigation to be slow and confusing (adapting on the y-axis to be placed under the question) and also that there was a lack of feedback when re-capturing GPS location. When performing task 6, some users felt that the radio buttons and checkboxes were
too small and also had trouble understanding that they were able to scroll down to reveal more options and the navigation interface.

6.2 Cross analysis of user testing results

This section will show a cross analysis of the users scores and time spent on the different tasks.

<table>
<thead>
<tr>
<th>Task</th>
<th>Android</th>
<th>Windows Mango</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 sec</td>
<td>6 sec</td>
<td>18 sec</td>
</tr>
<tr>
<td>2</td>
<td>24 sec</td>
<td>6 sec</td>
<td>0 sec</td>
</tr>
<tr>
<td>3</td>
<td>84 sec</td>
<td>72 sec</td>
<td>72 sec</td>
</tr>
<tr>
<td>4</td>
<td>42 sec</td>
<td>36 sec</td>
<td>48 sec</td>
</tr>
<tr>
<td>5</td>
<td>30 sec</td>
<td>48 sec</td>
<td>48 sec</td>
</tr>
<tr>
<td>6</td>
<td>24 sec</td>
<td>18 sec</td>
<td>42 sec</td>
</tr>
<tr>
<td>7</td>
<td>30 sec</td>
<td>24 sec</td>
<td>30 sec</td>
</tr>
<tr>
<td>Average total</td>
<td>3 min 54 sec</td>
<td>3 min 30 sec</td>
<td>4 min 18 sec</td>
</tr>
</tbody>
</table>

Table 6.2 Differences of average time spent on tasks for all platforms

As can be seen in table 6.2, the tasks with the highest average time was task 3, 4 and 5 which all show a similar average time over all platforms except for task 5 on Android which took 18 seconds less than on the others. Android and Windows Mango has a 24 second difference in total average time. But we can see the biggest difference on iOS which has an average time of 4 minutes 18 seconds, 48 seconds more than Windows Mango and 24 seconds more than Android. But even though iOS has a higher average time it still got the highest average score of all the platforms.

<table>
<thead>
<tr>
<th>Task</th>
<th>Android</th>
<th>Windows Mango</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7</td>
<td>4.7</td>
<td>4.7</td>
</tr>
<tr>
<td>2</td>
<td>4.3</td>
<td>4.7</td>
<td>4.7</td>
</tr>
<tr>
<td>3</td>
<td>4.5</td>
<td>3.4</td>
<td>4.5</td>
</tr>
<tr>
<td>4</td>
<td>3.2</td>
<td>3.7</td>
<td>3.5</td>
</tr>
<tr>
<td>5</td>
<td>4.3</td>
<td>4.2</td>
<td>4.3</td>
</tr>
<tr>
<td>6</td>
<td>3.8</td>
<td>4.0</td>
<td>4.0</td>
</tr>
<tr>
<td>7</td>
<td>4.5</td>
<td>4.4</td>
<td>4.3</td>
</tr>
<tr>
<td>Average total</td>
<td>4.18</td>
<td>4.15</td>
<td>4.28</td>
</tr>
</tbody>
</table>

Table 6.3 Differences of average score on tasks for all platforms

Table 6.3 shows the average score that users gave each task for all platforms. As we can see on task 1 and 2, users were quite consistent in their answers scoring an average of 4.7 out of 5 except for on the Android which got 4.3. This could be related to the fact that some users found the Android to have a slow response. Task 3 shows a score of 4.5 for Android and iOS but a lower score of 3.4 for Windows Mango. Users may have been influenced by Windows Mango’s native media capture interface which was poorer than Android and iOS. Task 4, 5, 6 and 7 are also quite consistent between the platforms but show a lower average for Android for task 4 and 6. This could again be because of users feeling that the Android had a slow response when reading user input. As the table shows, the average total for all platforms do
not differ by much which is a good sign that the prototype performed in almost the same way across all platforms.

By correlating the scoring between the different platforms with previous platforms used we can tell from figures 1-4 in Appendix D that there is not much difference in the scoring. When correlating the time in the same way we can tell from the figures that previous Android users got an overall better average time on iOS and previous iOS users got a slightly better overall time on Android. Previous Android and iOS users overall average time for Windows Mango differed by only about 1 second.

Looking at the correlation between the order that the platforms were used during the tests and the scoring shown in figures 5-10 in Appendix D we can see that task 3 for Windows Mango got a score of 3 when used as first platform and a 4 when being used as the third platform. We can also see that the overall score became lower for Windows Mango when being used as the second platform only to become higher again when being used as a third platform. When using Android as the third platform, the scoring was raised from 4 to 5 for task 2 from using Android as a second platform. We can also see that using Android as a third platform gave a lower score for task 3.

Looking at the average time for Android and iOS if figures 5-10 in Appendix D, we can see that the overall average time was lowered when being used as second platforms and then raised when being used as third platforms. For Windows Mango, the low average score when being used as the first platform could be a result of only two users using it since the average time is raised considerably when being used as the second platform. When Windows Mango was used as a third platform we can see a considerable lowering of the average time.

Although some have been mentioned above, because of the low number of participants that participated in the testing along with the randomized usage of the mobile platforms it is hard to correlate scores and times with previous mobile platform usage and testing order.

In the next section I will present the observations that were made during testing along with user comments and their suggestions for improvements.

6.3 Questionnaire answers and observations during user testing

When performing the tests, overall experience for users was positive concerning prototypes ease of use and understandability. However, there were a couple of things that users commented about in the questionnaire.

On Android, five of the test users complained about the navigation interface being slow which could be related to the responsiveness of the Android touch screen and not the prototype itself. According to the users, this was no problem on iOS which uses the same code for navigation furthering empowering my previous statement. For Windows Mango, there were some specific problems that users did not experience on the other platforms. Since animations when navigating did not work on Windows Mango, some users found it difficult to see if they had actually moved forward. Especially when navigating between the screens for capturing
media (screens all have the same look, just different legends and button text). Some users also commented about the design for Windows Phone being much poorer than that of the Android and iOS.

The overall experience of the prototype on all platforms shows that users favored the iOS platform over the others which could be related to the native aspects of the iPod that was used for testing being better than the other devices. Users also found the Android to be slow when reading inputs (e.g. pressing buttons) causing users to press buttons several times since they did not know if they had actually pressed it. Again, this could be a problem related to the responsiveness of the Android’s touch screen since the same problem occurred with navigation and this was not a problem on iOS or Windows Mango. In other hand, users were positive to the use of big buttons which is easy to see and to put their fingers on. However, one user found it confusing after saving the form since the save button was disabled and therefore changed shape. One user also found a crucial flaw in the design concerning the uploading function. A suggestion was made that one should not be able to press the upload button before saving since this will only upload the captured media and make the user lose the form data. This happened to one user during the testing.

From the users that commented about the difference in navigation between the platforms, three users thought that touch navigation was better than the buttons used in Windows Mango while three users favored the buttons. Concerning the navigation interface, most users assumed that they could navigate by sliding anywhere on the screen sometimes causing confusion. This is a crucial usability problem since navigation is greatly required for an application such as this. They also thought that the platform looked similar on all the platforms but that it could require some improvements concerning the styling since it looked dull.

In addition to these experiences, some users found the radio buttons and checkboxes in task 6 to be too small on Android and iOS. Some users did also not understand that there were more options if they scrolled down or that they even could scroll down. Related to the multiple choice menus, there was also a problem concerning navigation. Since the choices in the menus were too many to fit on the screen, this meant that the navigation interface would be hidden unless the user scrolled down. This caused some confusion when attempting to figure out how to move forward in the form from the multiple choice menus and was a crucial miss in the design.

Test users had no problems understanding the implemented feedback functions but suggested that the prototype should provide more feedback when performing actions, navigating forth and back and seeing whether a question is complete or not. Users especially thought that some sort of ‘loading’ screen should be applied when attempting to retrieve coordinates. As an indicator of whether an action or question had been completed, one user suggested a change in color or text. Another user suggested that a green tick could be added.

Most problems with the prototype seemed to be related to the navigation interface. When commenting about what improvements could be made, one user suggested that the navigation interface should be the same on all platforms. During testing it was also observed that one
user tried to click the arrows on iOS instead of dragging across. Some users also navigated the wrong way on the first screen returning them to the start screen. This could easily be solved by adding an alert asking users if they want to quit or not.

Two users gave the suggestion that all controls on a screen (question and navigation) should be visible at all times to avoid confusion. As an upgrade for the current navigation interface, users also suggested to have some sort of shortcut in order to navigate between several screens in one click (e.g. overview navigation of all available screens) along with having some sort of indication of what question in the form they were on (e.g. question 1/8).

Apart from these improvements, different users gave some more suggestions such as using the normal radio buttons and checkboxes style instead of the current one. This was used during the development process but changed due to the fact that they were so small that it made it hard to choose options. The radio buttons are still round and checkboxes are still square, but one user thought that it would be better to have ticks for the checkboxes and dots in the radio buttons for Android and iOS. Windows Mango was not affected by the CSS in the same way and was fine according to the suggesting user.

Another user suggested using checkboxes for loading forms and deleting data. My assumption is that the user means that you should be able to select the form you want to fill out and then press a button called ‘load form’, and also that you should be able to cross a checkbox of the form that you want to remove and then press a button called ‘remove data’ in order to make it more clear what is happening. Another user felt that the deleting function along with the uploading button at the end of the form should have clearer instructions. As an update of the current function for retrieving location data, one user noted that it could be a good idea to save the coordinates automatically when taking a picture which would require less work of the users.

When deleting data, two users only clicked the delete data button and not the button for the saved form. The function of deleting data seems not to be fully clear. However, it could be as simple as changing the name of the button to something more informative making it clearer to users that “pressing this button will show you the data that can be deleted”.

User were informed about the native features of the different mobile platforms but still experienced problems such as accidently pressing the native back button causing the application to close and also understanding some of the media capture buttons and icons. However, since this thesis does not cover native features, nothing more will be mentioned about problems of this nature.

**6.4 Summary of observations and users commentary**

I will here summarize the results of the user testing by presenting lists of the problems and improvements that should be considered in future development. The lists presented are a combination of user commentary in the questionnaire and observations that were made during the testing. The list items will be categorized by improvements to existing functions and new implementations that could be considered.
Improvements

- Navigate on by dragging anywhere on screen
- Same navigation interface on all platforms
- More indication and feedback that one has navigated in the form (e.g. question 1/8)
- Everything should fit in one screen (question and navigation without scrolling)
- More feedback on actions and question completion (e.g. green tick when question complete)
- Bigger radio buttons and checkboxes with original styling
- Checkboxes or input buttons for selecting form to be loaded and data to be deleted
- Better instructions when uploading and deleting data
- Not being able to send data before form is saved
- Improvements of exiting form when on first screen (Are you sure you want to quit the form?)

New features

- Saving coordinates automatically with picture
- Automatic saving and loading partially saved form
- Shortcut navigation between question screens

Thanks to the users that tested the prototype, these usability errors were identified and can be of help to future development. Even though I could not solve everything during the development process and during testing discovered some usability error that I had not thought of, I strongly believe that there would be no problem all to implement any of the points in the lists and that they would further enhance the prototypes usability.

As the results of the user testing now has been shown and explained, the next section will provide the conclusions of the thesis and answer the questions that were formulated in the beginning.
7. Conclusions

This section will provide a discussion and show the results of the questions that were formulated for this thesis.

7.1 Discussion

This thesis took part as an experimental development within the Learning Ecology with Technologies through Science for Global Outcome and was written with the aim of determining in what way open standards could be used in order to tackle the current fragmentation between mobile devices for data collection. To fulfill this aim there was also a need to determine what components are required by such an application and what the possible benefits it would bring.

In order to answer these questions, following the identified requirements in table 2.1, the prototype explained under Design and prototyping in section 4 was developed and later tested with users. As the development was ongoing, a literature study was also conducted focusing on cross platform development and technologies that could be used for development as seen under Cross platform mobile application development and tools in section 2.2, mobile computing introducing interoperability and mobile sensors as under Mobile computing in section 2.1 and usability under section 2.5. In addition, the literature study also covered the platforms that have been focused on along with information about web and native applications and some state of the art projects.

When developing the prototype, all platforms were able to use the same code with some differences concerning change in file system paths and some functions in order to accommodate platform specific problems which are all explained under Design and prototyping in section 4. Section 4 also provides figures showing the applications styling on all platforms. As one can tell from the figures, Android and iOS had somewhat the same looks while Windows Phone that used the same CSS was very different showing that more efforts should have been put on the styling of the Windows Phone.

The prototype could have been developed differently by using a framework such as jQuery Mobile. jQuery Mobile provides different themes that could have been applied to make the prototype look more like a ‘normal’ mobile application and maybe with that increase the usability. However, due to the time limitation, this was not used.

When the prototype fulfilled all the requirements, a user study was conducted in order to assess the usability of the prototype. The results of the user testing are provided in section 6 and show an overall positive user experience with an average score of 4.2 out of 5 for all platforms. In order to get better data from the tests, 12 users could have been recruited which would have meant that each platform would have been used as first, second and third device an equal amount of times. This would also have created an opportunity to better structure the order in which the platforms were tested instead of randomizing it. If there had been more time, user testing could have been conducted outside. Participants could then have collected
real environmental data instead of testing the prototype in a laboratory environment making the results more valid in connection to the LETS Go project.

The results show similar problems of the prototype across all the platforms. However, had there been more time, some of the problems could most likely have been discovered and therefore addressed during the development process by conducting small user tests. The problems identified suggest that the prototype would require the improvements listed under Summary of observations and user commentary in section 6.4 in order to become more user-friendly. From looking at figure 6.2 we can see that users spent most time on task 3 which involved capturing media on three different screens and had no specific instructions of how long to record. This could be one explanation to the higher average time and could have been prevented by setting a time limit. Looking at the average score for task 3 in figure 6.4 we can also see a high standard deviation of above 1 showing that this task was quite different between users.

From looking at figure 6.4, aside from task 3, we can also see that task 4 and 6 had an overall lower average score which is consistent with user’s answers in the questionnaire that have been compiled under section 6.3. Like task 3, these tasks also have high standard deviations showing that the difficulty of the task was different between users. The high standard deviations could be a result of user’s previous experience with mobile applications and/or their previously used platforms. The average score of each platform could also have been affected by the difference in the mobile devices used for testing. Many users found the iPod (iOS) with a total average score of 4.2 to be more responsive and easier to use than the Android or Windows Phone which both had a total average score of 4.1.

From the answers in the questionnaires and from looking at figure 6.1, we can see that even though no users had previously used the mobile platform Windows Mango the results for Windows Mango show the least spent time over all platforms and the same score as Android. However, since most users started the test by using either iOS or Android, the tasks may have become easier to perform when using Windows Mango. The answers provided from the questionnaires also show that user overall found the prototype easy to use and understand. But from reading the Questionnaire answers and observations during user testing in section 6.3 we can see that most problems concerned navigation and that out of six users, three found buttons to be easier than touch and vice versa.

The prototype developed for this thesis was able to overcome fragmentation issues for data collection between the mobile platforms used. However, not just thinking about mobile data collection, but mobile fragmentation overall, it is hard to say whether open-standard technologies are sufficient enough to replace the current native development languages and completely eliminate mobile fragmentation. Especially when creating larger and more advanced mobile applications.
As the scope of this thesis has now been summarized it is time to answer the question: *How can open-standard technologies be utilized in order to tackle the fragmentation of mobile devices for data collection?*

As this thesis has shown, open-standard and web technologies can be used in order to create understandable and user-friendly cross platform mobile applications. These applications will run in different mobile platforms WebView, rendering the application as it would a webpage and not being a subject to the native application requirements of the different platforms. As seen in previous sections of this thesis, all platforms may not have full support for some of the technologies. But apart from this, one can still utilize standards and web technologies such as HTML5 following principles of interoperability, universal access and compatibility and the HTML5 platform PhoneGap providing equal ways of accessing mobile sensors on different mobile devices and platforms. Furthermore, also take use of the interoperable XML technology XForms where data can easily be transferred and shared across multiple platforms and devices. In this manner, developers are provided with an easy and flexible way of deploying cross platform mobile applications that can tackle the fragmentation of mobile devices for data collection.

*What components are required by the application in order to tackle fragmentation between diverse mobile platforms?*

The components required for such an application that has been created can be seen by looking at the Design and prototyping in section 4. There we can see that the application requires the use of standards and web technologies such as the interoperable XML technology XForms providing an easy way of sharing of forms across the diverse platforms along with HTML5 in order to tackle platform fragmentation. In addition to these technologies, for supporting the use of the mobile platforms native functionalities such as capturing media, navigating the file system and managing files. A mobile platform such as PhoneGap providing the necessary API's along with user created PhoneGap plug-ins is also required. This is one of the most important components when it comes to mobile data collection and the ability to collect mobile sensor data.

*What are the potential benefits of using web technologies and standards for cross platform implementation and deployment?*

Potential benefits can be seen under Design and prototyping in section 4 and User testing in section 6. From these sections we can see that using web technologies and standards in the development of cross platform mobile applications will provide almost the same design, functionality and usability over all the concerned platforms. By looking at the results provided under the User testing in section 6 we can also see that the average scores for the tasks on all platforms rate above neutral (3) indicating a satisfactory user experience between the different platforms. However, Windows Mango shows a higher standard deviation on most tasks indicating that there was more variance in the scoring between users than on the other platforms. To summarize, by combining the Foundations in section 2, the Design and prototyping in section 4 and User testing in section 6 along with my own experiences. The potential benefits of using web technologies and standards are that it will require less code
and therefore time and that it will not require developers to have knowledge about all the different programming languages. But most importantly, it provides a way to counteract the current fragmentation between mobile platforms by functioning over an array of different mobile devices providing a similar design and functionality with only small changes in the code.

7.2 Further development

This section will provide thoughts and user suggestions for further development of the mobile data collection prototype. The lists below were created from user suggestions and observation during user testing and should be considered in future development. Added to these lists are some of the requirements that were not fulfilled during the development process along with other issues that could not be resolved.

Improvements

- Navigate on by dragging anywhere on screen
- Same navigation interface on all platforms
- More indication and feedback that one has navigated in the form (e.g. question 1/8)
- Everything should fit in one screen (question and navigation without scrolling)
- More feedback on actions and question completion (e.g. green tick when question complete)
- Bigger radio buttons and checkboxes with original styling
- Checkboxes or input buttons for selecting form to be loaded and data to be deleted
- Better instructions when uploading and deleting data
- Not being able to send data before form is saved
- Improvements of exiting form when on first screen (Are you sure you want to quit the form?)
- Downloading forms from server

New features

- Saving coordinates automatically with picture
- Automatic saving and loading partially saved form
- Shortcut navigation between question screens
- Reading visual codes on Windows Mango
- Using local XML files for Windows Mango
- Touch navigation for Windows Mango

Apart from the presented lists, the next step for furthering improving the prototype could be to connect it with the server side of the LETS GO project. This is in order to display and view the collected data since the server side programming for this thesis was limited to the uploading of files and sorting them into different folders on the server.
Another improvement that could be made is to provide a friendlier user interface. Future developers should consider implementing frameworks such as jQuery Mobile in order to increase the current usability, interface and design. jQuery Mobile was not used during the development process due to the limited time.

Acknowledgements

I would here like to give a special thanks to my supervisor Bahtijar Vogel for all the hard work he has put down in helping me to complete my bachelor thesis and for constantly pushing me forward. I would also like to give thanks to Henrik Andersen for taking the time to help me with issues related to iOS and the iPod that was used for testing. Those issues would have been hard to solve without him. In addition, I would also like to give a big thanks to all the users that tested my prototype providing me with the necessary data that was required in order to complete my thesis. At last, a big thanks to all the employees and students at Linnaeus University that have helped me whenever I have had questions.
References


http://doi.acm.org/10.1145/775152.775164


http://www.springerlink.com/content/h26t2680k7150110/


http://doi.acm.org/10.1145/1941487.1941504


CSS3Test.[online]( 2012-04-12) Available at: http://css3test.com/ [2012-04-12]


HTML5Test.[online](2012-04-12) Available at: http://html5test.com/ [2012-04-12]


DOI=10.1145/1236360.1236433 http://doi.acm.org/10.1145/1236360.1236433

Layar.[online]( 2012-03-08). Available at: http://support.layar.com/entries/161321-how-does-augmented-reality-work-on-the-layar-browser [2010-04-23]


OpenDataKit.[online]( 2012-03-06) Available at: http://www.opendatakit.org [2012-03-06]


PhoneGap.[online](2012-03-05) Available at: http://phonegap.com/ [2012-03-05]

Richter, K., Nichols, J., Gajos, K., & Seffah, A. 2006. The many faces of consistency in cross-platform design. In CHI ’06 extended abstracts on Human factors in computing systems (CHI EA ’06), pp. 1639-1642. DOI: 10.1145/1125451.1125751
http://doi.acm.org/10.1145/1125451.1125751


http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6036801&isnumber=6036584

http://ehis.ebscohost.com/eds/detail?vid=3&hid=23&sid=5b09246b-b987-4662-8a24-9cf29d0c5db%40sessionmgr11&bdata=Jmxhbmc9c3Ymc2l0ZT11ZHMtbG1ZSZZ2YW9wZT1zaXRI#db=edsbl&AN=RN279622581

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5476539&isnumber=5476518

W3Schools [1].[online]( 2012-03-08). Available at:
http://www.w3schools.com/dom/dom_intro.asp [2012-03-08]
W3Schools [2].[online](2012-03-05) Available at:
http://www.w3schools.com/html5/html5_intro.asp [2012-03-05]

W3schools [3].[online](2012-03-05) Available at: http://www.w3schools.com/css3/
[2012-03-05]
Appendix A - Questionnaire for user testing

Dear participant,

The statements below are related to the tasks that you have performed by using the prototype. Please read each statement carefully and choose one answer per statement. The data collected will be used to assess the usability aspects of the prototype in relation to my Bachelor Thesis.

Again, thank you for your participation.

Daniel Hjärström

The values of your answers are based on the scale presented in the table below:

<table>
<thead>
<tr>
<th>Strongly disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**TASK 1**

1. Choosing the form and loading it was easy.

   **Android**
   - Strongly Disagree
   - Disagree
   - Neutral
   - Agree
   - Strongly Agree

   **iOS**
   - Strongly Disagree
   - Disagree
   - Neutral
   - Agree
   - Strongly Agree

   **Windows Mango**
   - Strongly Disagree
   - Disagree
   - Neutral
   - Agree
   - Strongly Agree
**TASK 2**

- Capturing location data (GPS) was easy.

| Android |  |  |  |  |  |  |
|---------|--------|--------|--------|--------|--------|
|         | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

| iOS |  |  |  |  |  |
|-----|--------|--------|--------|--------|
|     | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

| Windows Mango |  |  |  |  |  |
|---------------|--------|--------|--------|--------|
|               | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

**TASK 3**

- Capturing media (picture, video and audio) was easy.

| Android |  |  |  |  |  |  |
|---------|--------|--------|--------|--------|--------|
|         | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

| iOS |  |  |  |  |  |
|-----|--------|--------|--------|--------|
|     | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

| Windows Mango |  |  |  |  |  |
|---------------|--------|--------|--------|--------|
|               | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

**TASK 4**

- Navigating back and forth in order to re-capture new location data and picture was easy.

| Android |  |  |  |  |  |  |
|---------|--------|--------|--------|--------|--------|
|         | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |

| iOS |  |  |  |  |  |
|-----|--------|--------|--------|--------|
|     | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
### TASK 5

- Entering text and numeric values by hand and understanding notifications was easy.

### Android

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

### iOS

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

### Windows Mango

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

### TASK 6

- Choosing options from the presented lists by using radio and checkbox buttons was easy.

### Android

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

### iOS

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

### Windows Mango

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>
TASK 7

- Using save, upload and delete functions of the prototype was easy.

<table>
<thead>
<tr>
<th>Android</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Strongly Disagree</td>
<td>Disagree</td>
<td>Neutral</td>
<td>Agree</td>
<td>Strongly Agree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iOS</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Strongly Disagree</td>
<td>Disagree</td>
<td>Neutral</td>
<td>Agree</td>
<td>Strongly Agree</td>
</tr>
</tbody>
</table>

| Windows Mango |          |          |          |          |          |
|               |----------|----------|----------|----------|----------|
|               | Strongly Disagree | Disagree | Neutral  | Agree    | Strongly Agree |

What was your overall experience of the prototype (such as ease of use, design and functionalities, understandability)?

Android

___________________________________________________________________________

iOS

___________________________________________________________________________

Windows Mango

___________________________________________________________________________

What is your overall experience of the prototype over all platforms? (Differences concerning style, performance/functionalities, ease of use)

Please give your thoughts on how the prototype could be improved.

What mobile platforms have you previously used?

Other thoughts.
Appendix B - Tasks for user testing

Dear participant.

This is a user test for a thesis in media technology concerning the mobile data collection tool prototype that has been developed and implemented on three different mobile platforms. Please read the tasks carefully before performing them and enter your start time before commencing a task. When you have finished a task, please enter your end time.

Thank you for your participation!

Daniel Hjärtström

<table>
<thead>
<tr>
<th>Task 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please go to ‘Choose Form’ and load the form called testForm.</td>
</tr>
<tr>
<td>Start time</td>
</tr>
<tr>
<td>-----------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture location data (GPS) by pressing ‘Get Coordinates’ button.</td>
</tr>
<tr>
<td>Start time</td>
</tr>
<tr>
<td>-----------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the following screens, please capture Media (Picture, Video and Audio).</td>
</tr>
<tr>
<td>Start time</td>
</tr>
<tr>
<td>-----------</td>
</tr>
</tbody>
</table>
### Task 4
Please navigate back and forth using ‘← navigate →’ function and re-capture location data (GPS). Now, re-capture the picture and continue where you left the last activity.

<table>
<thead>
<tr>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task 5
On the two following screens, please enter text and numeric values. **But before entering your values, please try to navigate forward.**

<table>
<thead>
<tr>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task 6
On the screen with the radio buttons, please choose an option. After, on the next screen (checkbox buttons), please choose at least two options from the presented list.

<table>
<thead>
<tr>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task 7
Please **save** and **upload** your data. Now, **delete** the data and repeat the same tasks using other mobile platforms.

<table>
<thead>
<tr>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Appendix C – File formats and file system paths

<table>
<thead>
<tr>
<th></th>
<th>Audio</th>
<th>Video</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Android</strong></td>
<td>amr</td>
<td>Mp4</td>
<td>jpeg</td>
</tr>
<tr>
<td><strong>iOS</strong></td>
<td>Wav</td>
<td>Mp4</td>
<td>jpeg</td>
</tr>
<tr>
<td><strong>Windows Mango</strong></td>
<td>Wav</td>
<td>MOV</td>
<td>jpeg</td>
</tr>
</tbody>
</table>

**Table 1.** Mobile devices file formats

<table>
<thead>
<tr>
<th></th>
<th>Android</th>
<th>iOS</th>
<th>Windows Mango</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Save root</strong></td>
<td>/sdcard/Thesis</td>
<td>file:///var/mobile/Applications/ (36 character id)/Thesis</td>
<td>/Thesis</td>
</tr>
<tr>
<td><strong>Audio</strong></td>
<td>/sdcard/Sounds</td>
<td>file:///var/mobile/Applications/ (36 character id)/tmp/</td>
<td>/AudioCache/</td>
</tr>
<tr>
<td><strong>Pictures</strong></td>
<td>/sdcard/Camera/DCIM</td>
<td>file:///var/mobile/Applications/ (36 character id)/tmp/</td>
<td>/CapturedImagesCache/</td>
</tr>
<tr>
<td><strong>Video</strong></td>
<td>/sdcard/Camera/DCIM</td>
<td>file:///var/mobile/Applications/ (application id)/tmp/capture</td>
<td>/VideoCache/</td>
</tr>
</tbody>
</table>

**Table 2.** File system paths
Appendix D – Time and score correlation between users previously used platforms

**Figure 1.** Scores for users who previously used Android

**Figure 2.** Scores for users who previously used iOS
Figure 3. Average time for users who previously used Android

Figure 4. Average time for users who previously used Android
Figure 5. Average scores for first used devices

Figure 6. Average time for first used devices
Figure 7. Average scores for second used devices

Figure 8. Average time for second used devices
**Figure 9.** Average scores for third used devices

**Figure 10.** Average time for third used devices
Appendix E – Results of user testing

**Figure 1.** Average time when testing the prototype on Android

![Android: average time](image)

**Figure 2.** Average time when testing the prototype on Windows Mango

![Windows Mango: average time](image)
**Figure 3.** Average time when testing the prototype on iOS

**Figure 4.** Average score when testing the prototype on Android
Figure 5. Average score when testing the prototype on Windows Mango

Figure 6 Average score when testing the prototype on iOS