Leeds City Centre Made Mobile
Adnan Mahmood
BSc Computer Science
2012/2013

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) ________________________________

Summary

This report details the development of a mobile application based on Leeds to help enhance the experience of visiting the City Centre. Existing systems are inappropriate to use on mobile devices. Furthermore, mobile searches for local information is popular among users of smartphone however, the experience of using mobile internet on small screened devices has negative impact on the user experience.

This report initially discusses the project problem and the background reading s, followed by the design, implementation and the overall testing of the system.

Finally, it determines the success of the system based on the results found in the user evaluation along with the possible further extensions to the system.

Acknowledgement

I would like to thank my project supervisor, Marc de Kamps, for his invaluable advice and support throughout the course of this project. I would also like to thank my assessor, Nick Efford, for his valuable feedback provided in the mid-project report and during the progress meeting.

I am indebted to Kashif Hussain and Adila Salis for their unselfish support, whom over the past three years, have redefined me as a person. Thank you for the laughs which we all have had, making everyday a lot brighter.

I also like thank my siblings for their endless support; putting up with me during my hardest times throughout my childhood. I would like to take this opportunity to wish them all every success in life.

Most of all, I dedicate this project to my parents, the most amazing people in the world, for their incredible love, support and encouragement throughout my life. I am thankful for the motivational speeches, teaching me the values of hard work and education. Thank you mum dad for making me the ‘never give up’ person I am today.

Contents

1. Introduction

1.1. Introduction

This chapter addresses the project problem and sets out the project aim and objectives. The minimum requirements and the possible extensions to the project are set in Section 1.4 and 1.5. Moreover, the deliverables that the project would produce is discussed in Section 1.7. Section 1.8 and 1.9 expresses the evaluation criteria and the project's relevance to the degree. Finally, a summary given in Section 1.10 concludes the chapter.

1.2. Problem Statement

Mobile phones are one of the most common means for accessing information on the go, with nearly 10 billion search queries submitted in November 2007 [38]. A study by Teevan et al [9] revealed that individuals often tend to search for local information such as directions and point of interests. Further studies by Teevan et al [9] and Church et al [38] concluded that the experience of local mobile search varies considerably from the desktop computer experience [9][38]; an issue mainly instigated by the limitations of mobile devices. Queries on mobile phones are usually shorter and less advanced, indicating that individuals put minimal effort with respect to query entry. Thus, it is vital to reduce individuals' burden, in regard to query entry and their understanding of the results obtained from such searches, in order to enrich the local mobile search experience. Moreover, in conclusion to their study, Church et al [38] emphasised the need for new context – sensitive mobile interfaces that consider the influence of temporal aspects, location features and social contexts.

Leeds City Centre is a very busy and complex place to visit, especially for new visitors. Hence people are bound to lose their way and having studied in Leeds for three years this hasn't gone unnoticed by me. From first- hand experience, I have witnessed a great deal of people enquiring about whereabouts of places of their interests. Finding a place of interest in an unfamiliar area can be a lengthy and confusing experience. Direction enquiries from others are inaccurate and unsuitable as directions tend to be based on the knowledge of the individual providing the directions. Using a paper based map is frustrating and due to social context, it is highly unlikely visitors are comfortable using a paper based map in the centre of a very busy location. It is arguable that mobile map applications can help however; visitors need to provide the location of their place of interest which involves a mobile search to identify the address of the place of their interest. The mobile searches can be slow and many visitors may not have the expertise to use mobile internet, especially the elderly visitors. Furthermore, the limitations of mobile devices have negative influences on the experience of mobile search [38]. With the increased ownership of smartphones and mobile applications, know as ‘apps’, a solution to the problem could be implemented in the form of an easy to use mobile application, providing features such as maps and directions.

1.3. Project Aim

The aim of this project is to provide visitors of Leeds City Centre an easy-to-use and easy-to-access city centre guide. This aim will be accomplished by implementing a mobile application which will provide access to various services. A map of the city centre, shopping and dining amenities, car parking information and the latest weather are among some of the services that will accessible by the users.

1.4. Project Objectives

In order to achieve the aim set in Section 1.3, the subsequent objectives have been established:

- Research and collect relevant information that are essential to the process of addressing the problem
- Select an appropriate mobile application platform
- Understand and familiarise with the programming construct and design patterns of the chosen platform
- Select an appropriate software development methodology
- Generate potential designs for the mobile application
- Implement a mobile application that offers services accessible to the users
- Test and evaluate the application with the main focus on the usability, accuracy and performance
- Evaluate the application against the project aim and minimum requirements in order to establish whether the problem has been addressed

1.5. Minimum requirements

The minimum requirements for this project are:

- A working mobile application that displays a map of Leeds City Centre
- Provides an indication of the user’s current location
- Displays a single predefined shop on the map
- Provides visual and text based directions from the user’s current location to the predefined shop

1.6. Possible extensions

The possible extensions for this project are:

- Features that provide a list of shopping, eating and seeing amenities. These features may cooperate with the map to provide directions to a specific place that is of the user's interest
- The ability to check for the latest train and bus times
- Provide information services such as the latest news and the weather

1.7. Deliverables

Along with the report, this project would produce the following deliverable:

- A mobile application for Leeds City Centre

1.8. Evaluation criteria

The final solution would be evaluated against the following criteria:

- User interface - Is the system usable?
- Technology - Does the application work? Does it use resources appropriately?
- Context - Is the application suitable for the context of use?
- Usefulness - Do the users find it useful?

1.9. Relevance to degree

This project necessitates knowledge and techniques gained in modules studied as part of BSc Computer Science. Since the project highly consists of application analysis, design and implementation, knowledge is to be drawn from technical modules such as 'Software Systems Engineering' and 'Distributed Systems', in order to investigate how a solution to the problem can be implemented. Moreover, since the solution is likely to be used by individuals and involves interface design, knowledge from 'Interactive Database Systems' module is also used. Finally, the project itself involves planning and therefore knowledge from Project Management is used.

1.10. Summary

This chapter has introduced the problem which visitors of Leeds City Centre face. The basic idea of this project has been discussed. The following chapter will detail the project management aspects associated with the project.

2. Project Management

2.1. Introduction

This chapter discusses the different methodologies that were taken into consideration and details the decision made to which methodology is deemed to be the most appropriate for this project. Finally, the chapter will show the proposed project schedule that was used.

2.2. Methodology

In regard to software development, methodology is thought of as a collection of methods [2]. The process of project analysis is concerned with selecting the most appropriate methodology for the project. The Forrester Research Group stated that many IT projects are late and over budget with the failure being pinpointed on the decision of selecting an inappropriate software development methodology [23]. This clearly illustrates the importance of selecting the most appropriate methodology in order to ensure the success of the project.

2.2.1. Waterfall

The first methodology explored is the waterfall approach. The waterfall approach is known as one of the classical model of software system development [5]. It consists of a sequence of tasks performed in a top – downward flow. However, stages can be backtracked if problems arise during the latter stages. Backtracking can be problematic to manage in the waterfall approach. The waterfall approach is regarded highly among managements due to the fact that each phase can be thought as a milestone in the project. This approach is ideal when requirements are clear and the development technologies are acquainted and well understood. The approach is not suitable in situations where there are uncertainties surrounding how a solution is to be implemented.

2.2.2. Iterative

The iterative methodology [2] [5] implements a system in a sequence of steps. Initially, the approach does not start with a full plan of requirements, instead it starts by implementing parts of the system. The partial system is then revised to identify further requirements. Each iteration consists of the design, development, testing and evaluation stages and produces a working system of partial functionality. Consecutive execution of iterations adds more functionality onto the work of the previous iteration and the system evolves until the final system is complete. Since the starting requirements are minimal and each iteration tends to meet one requirement, changes in the requirements during the latter stages are manageable with this methodology. Another advantage of this methodology is that the final system tend to consists of fewer errors since the testing and evaluation of the system is performed at each iteration. Problems spotted earlier are easier and cheaper to solve with regard to time [2]. Figure 2.1 illustrates the process of the iterative methodology.

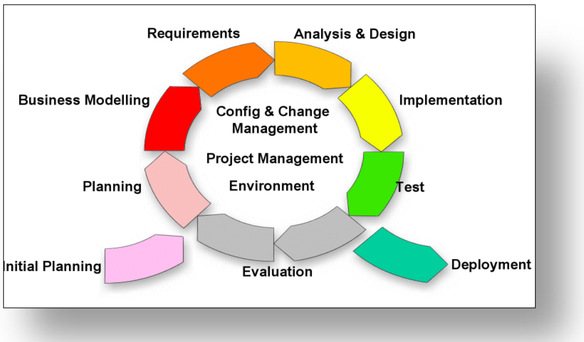


Figure 2.1. A figure to demonstrate the lifecycle of an iterative software development approach [36]

2.2.3. Incremental Delivery

The incremental (delivery) methodology [2] involves delivering the system in a series of builds. The full specification of the system requirements is broken down into modules which are built and delivered in a sequence. The release of each module must benefit the user and add functionalities to the previous release. The main advantage of this methodology is that a partial working system is generated early in the development [2]. Changes in the system requirements can be managed efficiently and the debugging of smaller release lead to fewer errors in the complete system. With time a limitation associated with the project, this helps to save time when solving errors. Nevertheless, this methodology requires extensive logical planning and design in order to get the most out of it. Similar to the other methodologies considered, the requirements of the system needs to be clearly defined and understood. Figure 2.2 illustrates the incremental delivery methodology.

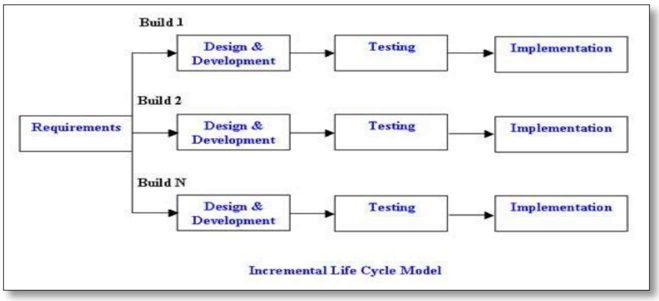


Figure 2.2. A diagram that demonstrates the stages of the incremental delivery approach

2.2.4. Justification

Since in the waterfall methodology the testing and debugging is performed during the latter stages, this methodology isn't suitable due to the time constraints associated with the project and the risks of identifying major faults in the system late in the development stage. This can be too costly and is a potential threat to the success of the project.

The iterative methodology overcomes the limitations identified in the waterfall approach. The testing is performed during each iteration hence any errors discovered are easier to solve. Since the errors are solved earlier in the development stage, the risk of discovering major errors in the complete solution is reduced significantly. However, each iteration involves three stages: the design, development and evaluation stages. This process, as it can be imagined, is lengthy and inappropriate, especially designing and evaluating during each iteration. Therefore the time constraints associated with the project does not allow this methodology to be the most appropriate. Furthermore, the key to developing a good mobile application, as stated by Matt Brain [11], is to ensure that the application is built for the users. Also, the evaluation performed at the end of each iterations is not appropriate as users may get confused during the evaluation as they may think it is the full solution.

Therefore, the decision was made to adopt the incremental delivery methodology as it allows early testing hence no major risks are associated late in the development. The system requirements could be broken down into a set of modules and the extensive planning required is thought to contribute toward the success of the project as it would clearly define what needs to be done.

2.3. Schedule

The schedule of the project was produced to identify the key tasks that are required to be undertaken in order to complete the project. The schedule was planned out using a Gantt chart, shown in Figure 2.3, to display the main tasks and the various deadlines associated with the development of the application as well as with the project as a whole. A table view of the schedule displaying the tasks, predicated start dates, predicated finish dates and the corresponding week, was also produced, as shown in Figure 2.4.

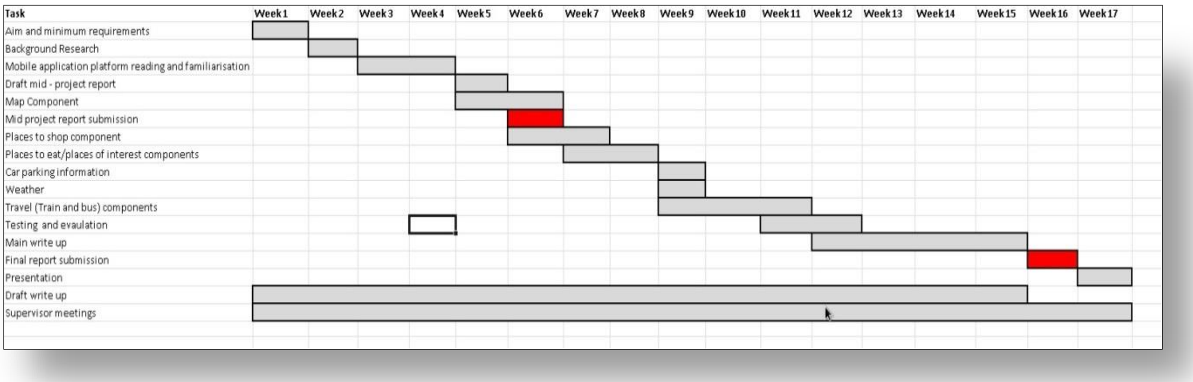
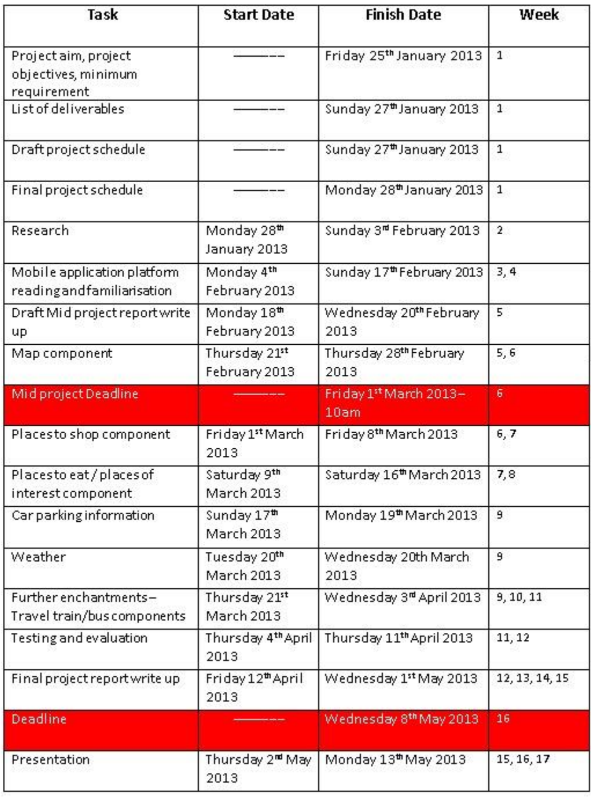


Figure 2.3. The Gantt Chart demonstrating the proposed schedule of the project.



Task	Start Date	Finish Date	Week
Project aim, project objectives, minimum requirement	———	Friday 25th January 2013	1
List of deliverables	———	Sunday 27th January 2013	1
Draft project schedule	———	Sunday 27th January 2013	1
Final project schedule	———	Monday 28th January 2013	1
Research	Monday 28th January 2013	Sunday 3rd February 2013	2
Mobile application platform reading and familiarisation	Monday 4th February 2013	Sunday 17th February 2013	3, 4
Draft Mid project report write up	Monday 18th February 2013	Wednesday 20th February 2013	5
Map component	Thursday 21st February 2013	Thursday 28th February 2013	5, 6
Mid project Deadline	———	Friday 1st March 2013– 10am	6
Places to shop component	Friday 1st March 2013	Friday 8th March 2013	6, 7
Places to eat / places of interest component	Saturday 9th March 2013	Saturday 16th March 2013	7, 8
Car parking information	Sunday 17th March 2013	Monday 19th March 2013	9
Weather	Tuesday 20th March 2013	Wednesday 20th March 2013	9
Further enchantments– Travel train/bus components	Thursday 21st March 2013	Wednesday 3rd April 2013	9, 10, 11
Testing and evaluation	Thursday 4th April 2013	Thursday 11th April 2013	11, 12
Final project report write up	Friday 12th April 2013	Wednesday 1st May 2013	12, 13, 14, 15
Deadline	———	Wednesday 8th May 2013	16
Presentation	Thursday 2nd May 2013	Monday 13th May 2013	15, 16, 17

Figure 2.4. The project schedule table view

2.4. Summary

This chapter has discussed the various methodologies considered. This has led to the selection of I think is the most appropriate methodology for this project. Moreover, the proposed schedule for the project has been defined.

3. Research

3.1. Introduction

This chapter discusses the background research that was carried out prior to the design and implementation of the solution. The purpose of the research was to gain knowledge and understanding of the relevant technologies, important material and to understand how a solution to the problem could be implemented.

3.2. Existing systems

There are systems available that offer visitor's a guide of the city centre. I investigated the various functionalities that each one offered. The analysis gave an idea of what is most likely to be expected in a city guide. Furthermore, the layout and the design of the interface were investigated.

3.2.1. Leeds City Guide

The Leeds City Guide is a fairly simple website offering a guide to Leeds as a city [6]. The website is divided into various sections, each which provide some sort of guidance on a place of interest, e.g. pubs, hotels and restaurants.

Within each section, users are presented with information about a particular amenity. The information provided includes the address and telephone number. Each amenity has its own webpage, including a description of the particular place. Moreover, a map showing the location of the particular place is also placed on each individual page. The website supports the ability to search, provided by the search box functionality.

Whilst analysing the website, it became apparent that a lot of user's input and effort is required to find some features. For example, finding the map that shows the location of a particular place requires navigating through a number of webpages.

3.2.2. Leeds, Live it, Love it

The 'Leeds, Live it, Love it' visitors guide is the official online guide for Leeds [7]. The website offers various features such as places to go, 'What's on' and accommodation details.

Each section is divided into categories. For example, the 'places to go' feature is divided into many categories such as clubs, eating, shopping and libraries. Each particular place has its own webpage providing information such as address, postcode, telephone number and a link to the particular amenity's website.

In terms of the appearance, the website is appealing. Each feature is easy to access and the website supports efficiency in terms of how much an individual needs to do before they arrive to the information they want. A map is also provided, however this feature is limited: just showing the location of the amenity. The ability to search for directions or to provide the current location of user is not available.

3.2.3. Conclusion

In conclusion, both systems contribute to providing information about Leeds and both are targeted at different audiences as they provide different types of information. The analysis has also backed up the claims, by Church et al [38], on the need for a new interface to overcome the local mobile search issues and has demonstrated the need for a new mobile solution. The reason for this is because, during the analysis, both systems were assessed on Safari web browser on an iOS based smartphone and the systems were very frustrating to use on the mobile device as they were not efficient in terms of load up speed and were not scaled to fit the screen hence required a lot of zooming in and out. Altogether, this led to a poor user experience. Moreover, this analysis was performed while staying stationary, in reality users are expected to be on the move in a busy environment. Clearly, zooming in and out on smartphones isn't a good idea in such situations.

3.3. Mobile Phone Computing

With the continuous improvement of the hardware associated with mobile computing, today's mobile devices makes it possible to implement complicated and useful functionalities and services. The increased functionalities have created a major need among consumers, with primarily uses are for communication, education, entertainment and location services.

Many experts e.g. [25] have commented on the growth of mobile technology. In 2013 it was announced that for the first time in history, smartphones and tablet computers are becoming the most popular form of computer, surpassing the sales of desktop computers [25]. Ryan Reith, of Worldwide Mobile Device Trackers, stated that the growth of mobile devices and the decline of traditional desktop computers are due to the device prices and portability [25].

Despite the success of mobile devices in the market, the processing and memory power capabilities are not as capable as the power available in desktop computers, with the latest iPhones containing a CPU clock speed of 1 gigahertz and a Random Access Memory (RAM) of 1 Gigabytes [44]. Thus, as a developer there are many limitations to consider when designing mobile applications. The battery life is also a limitation on the developer, with today's battery life lastly for about 225 hours on standby [51]. This will need to be considered as high performance tasks in a mobile application can affect the battery life.

Although internet access is available from smartphones, the speed is not yet very sufficient as the speed accessible on desktop computers. Therefore, it is vital to take the network speed into account when implementing the solution as functionalities may possibly need to download data, in order to perform effectively. Ideally, the size of the data should be kept to minimal to avoid external costs associated with the user's service providers. On the other hand, Wi-Fi is common these days and various Wi-Fi hotspots are available, especially in large city centres. Wi-Fi offers a lot more, in terms of speed, than mobile internet. However, since the users are expected to move around with their devices, taking advantage of the high speed offered by Wi-Fi is simply not feasible in this current age of technology.

Since the early days, the most useful improvement on mobile devices has been the screen size. As of 2011, the average screen size of smartphones varies from 3.5 to 4.5 inches [24]. The limitation of the screen size is turning out to be one of the most challenging problem in this project, in relates to the amount of content that is feasible to display on a mobile device screen. This links to the facts found in the reading of *'what makes a good app'* stated in Section 3.5.1, that cluttering the screen with excessive amount of content can affect the appearance of the application, thus have influence on the user experience.

3.4. Human computer interaction

Human Computer Interaction (HCI) is the subject of the relationship between people and computer systems. The design of the user interface is one of the main concerns associated with HCI. This section discusses some of the issues that need to be taken into consideration when designing the user interface.

3.4.1. HCI

There four key stages in user interface design [1]:

1) *Understand and specify the context of use*

   This concerns with the choice of user interface elements and functions, such as the use of user input via keyboard or selection,in relations to the context in which the users are likely to be within when using the application.

2) *Specify user requirements*

This concern with capturing the user requirements. It also involves considering the factors related to the users such as the characteristics of the users and their requirements with regard to the proposed system.

3) *Designs*

This stage involves the creation of designs that satisfy the user requirements. This stage extends beyond the design itself, the testing of the designs produced by developing prototypes in order to gain feedback from potential users. Various types of prototypes exist: low fidelity and high fidelity [4]. Each type has its own advantages, with the latter allowing the usability to be assessed while the former is quick, inexpensive and can provide valuable feedback in the early stages of development.

4) *User based assessments*

This stage involves asking potential users to interact with the prototype designs. The result of the assessments can provide valuable feedback with regard to deciding whether the prototypes meet the user requirements. The assessments can help to identify navigational issues within the application, the features or aspects of the application that the users don't like and the parts of the application where more general usability issues exist. Reiteration of the designs can then help to solve the issues discovered from the assessments hence resulting a better system, in terms of usability.

3.4.2. Methodology

As discussed in Section 2.2, the chosen software methodology is the incremental methodology. Following the findings of the process of interface design, as discussed in Section 3.4.1, it was thought that this methodology had to be adapted in order to accommodate the four HCI user interface design stages. Figure 3.1 shows the revised incremental methodology plan.

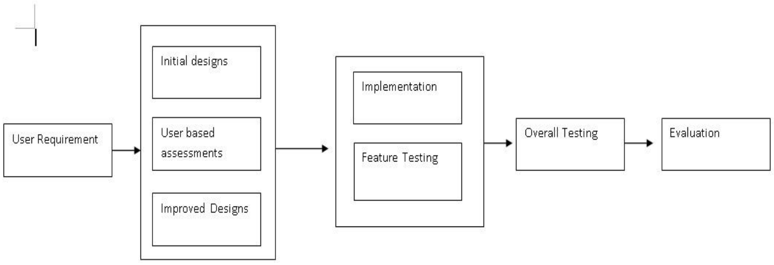


Figure 3.1. The altered incremental methodology

3.4.3. Conclusion

In conclusion, it is vital to consider the users and their characteristics in order to design a successful system as research [47] shows that successful systems have been linked to consumer based assessments. On the other hand, research by Subrayaman et al [48] shows that the results from the assessments clearly depend on the number of participants. As the number of participants increase, the satisfaction of the system reduces. However, Kujala and Mantyla [49] concluded that it is worth carrying out the assessments as the benefits outweigh the effort and cost associated with the assessments. In addition, the usability of the application is also key to implementing a successful system. In terms of user experience, the limited screen size does no favour at all. It is important to reflect this by decreasing information complexity in order to fit everything within the parameters of the devices' screen. Ultimately, the emphasis should always be on the user, a fact that should be treated as a building block to implementing a successful system.

3.5. Mobile Application

Mobile application, also known as 'apps', are on a huge rise. In 2013, Apple announced 40 billion apps download since 2008, with more than half of that recorded in 2012 [26]. This section debates what makes a good app, followed by two of the main mobile application platform.

3.5.1. Mobile apps

For the users, the main aspect of a mobile application is the user interface. The appearance of the application and the extent to which the application operates is very crucial in the success of the application [11]. The first impression is essential with regards to capturing the users' interests. As with website applications, it is important that the user interface of a mobile application is attractive as it contributes toward the ease of use. Recent study by Google and Peep Laja has discovered that visual appearance can be judged in around 50 milliseconds [27] [28]. This shows that applications have around 50 ms to create a good first impression.

In order to implement a good application, it is essential to make the flow of the navigation within an application as fluid as possible. As Matt Brain of TWN Mobile stated [11], the use of standard icons and buttons that users are familiar from the use of other applications, can help to achieve a good navigation within an application [11]. Users of computing devices and mobile applications are already familiar with the back, cancel and next buttons. Hence the use of these buttons will help the user to make the correct decision in according to the task the user is trying to achieve.

Moreover, a good application is achieved by developers keeping in mind the mobile device limitations, in particularly the small touch screen [10]. The content of an application needs to be laid out so that the user interface isn't cluttered with too many design elements. The use of categories to organise and separate content helps to achieve this. However, one must consider the complexity of accessing the separate content and its influence on the flow of the navigation within the application.

3.5.2. Mobile application platform

This section introduces the mobile application platforms that were explored, in order to make a decision on which is the most appropriate platform to select. Currently, there are various mobile application platforms. The decision was made to focus on the two main leaders of the market, Android and Apple iOS, over the past years both have been fighting for the market dominance [17]. We decided to compare these two platforms as they clearly dominate the market.

3.5.2.1. Android

Android was developed by Android Inc. later bought by Google in 2005. The Linux based operating system is principally used for designing and implementing mobile applications for touchscreen mobile devices such as smartphones and tablets [12]. Mobile applications in Androids are developed in Java programming language using the official Android SDK [13]. As well as the operating system, the platform also includes the middleware and key applications. Moreover, the platform utilises the Linux version 2.6, primarily for core system services. The core system services include security, memory management, process management, network stack and the driver model [14].

Android is the most popular mobile application platform, with 75% of the smartphone market share in 2012 [13]. More than 600,000 mobile applications have been implemented in the open source platform with over 20 billion downloads [16].

The platform is activated on over 500 million devices [29]. Each mobile device comes with different hardware capabilities and sizes. Screen sizes of each device vary, hence an application on one device may appear differently on another. Although, the Android SDK provides the tools which support developers to adapt the user interface so that it looks its best on each device [12], an implementation of one mobile application has to be developed for four different devices. In other words, one single Android mobile application has to be developed for four different screen sizes. This is because there are four different screen sizes for mobile devices that run on the Android platform [50].

3.5.2.2. Apple iOS

Apple iOS is a mobile platform developed by Apple Inc. The platform was developed in 2007 for implementing mobile applications for the iPhone and iPod, later extended to support Apple's newly invention, the iPad and AppleTV [30]. Mobile applications are developed in Objective – C programming language using Apple's SDK. Unlike Android, which relies on Linux 2.6 for core system services, iOS has its own system framework, one which can be perceived as a set of layers. The platform consists of four layers: the Core OS layer, the Core Services layer, the Media layer and the Cocoa Touch layer. The high level layers are primarily to assist with the mobile application. On the other hand, the lower level layers provide core system services.

Apple is considered as the 2nd most popular mobile application platform, with 14.9% of the smartphone market share in 2012 [31]. Apple mobile applications are distributed through the official Apple's 'App Store', which has over 700,000 mobile applications and more than 30 billion downloads [16]. In contrast to 500m Android devices, in September 2012, Apple announced that 400m iOS devices have been sold [32], with Horace Dediu of Asymco projecting that Apple will sell 1 billion by 2015 [33].

There are, however, some limitations imposed on developers. Since iOS mobile applications will reflect Apple as a brand, Apple has strict guidelines that developers have to comply to in order to meet Apple's design standards and have their applications published in the 'App Store'. Each application is reviewed by Apple's developers [34] before permission to distribute to the general public is obtained. This means that it takes a lot longer to publish iOS applications compared to Android applications.

3.5.2.3. Conclusion

After spending a great deal of time exploring Android and Apple iOS mobile application platform, the decision was made to select the iOS platform for this project. The decision was heavily based on the Apple's strict design and implementation guidelines and on the fact that iOS offered far better benefit from the developer's point of view [17].This means that as a developer the guidelines can help to implement a better mobile application.

Some would argue that Apple's strict guidelines are a nuisance, however, the guidelines help to overcome the HCI issues associated with the project and that complying with the guidelines results in a much better mobile application. Furthermore, this could justify Jude Venn's, of Cuttlefish Multimedia, opinion that iOS users are more likely to spend money on mobile applications compared to those on Android [17], due to the quality of the applications implemented in iOS. Moreover, the guidelines can help to reflect and encounter the facts of 'what makes a good app' discussed in Section 3.4.2.

Finally, Android requires four implementation of a single application as Android phones can be of four different screen sizes. On the other hand, Apples phones come in two different screen sizes, however, only the content view screens need to be recreated for the two screen sizes. Other contents such as tableview are adjusted, automatically, to fit the screen sizes. Therefore, it is not feasible to implement an Android application for four different devices due to the time constraints associated with the project.

3.6. Similar applications

This section discusses the findings from the analysis of similar applications that aims to do the same job. Two applications were discovered and analysed, both implemented in iOS and analysed on an iPhone. It was vital to examine similar applications for different cities around the world in order to gain some experience in the visual appearance.

3.6.1. Cologne Guide

The Cologne Guide is the official tourist guide for the city of Cologne. The app, created by Koln Tourismus, was implemented for overseas tourists [19]. The travel guide provides extensive information about the city such as tourist attractions, events, places to shop and hotels.

The main screen contains a list of different features that the application provides such as accommodations and shopping. However, one can argue that the use of a list is inappropriate as most users will be on the move in a busy crowded environment and having users, especially users who are unfamiliar with the application, looking through a list is clearly unsuitable.

Since a list is hard to scan through to quickly find a particular item, the ease of finding something of interest is not feasible as users are required to take a number of steps to navigate through menus in order to get to where they want to be. The app provides a search functionality which is suppose to promote ease of use and to reduce the time in which it takes to find something of interest. However, the search functionality itself is not clear and requires a lot of effort to use.

3.6.2. Deira City Centre

The Deira City Centre app is the official mobile application for the city centre of Deira. The application developers claim the application is designed *'to be a mobile shopping assistant, helping customers to get the very best out of their visit'* to the city centre [35]. The application contains various functionalities including store directory, dining directory, parking reminders and a map of the malls.

The *Home* screen contains various icons which represent different features of the application such as shopping, what's on, promotions and parking. The use of icons is a noteworthy decision, in regard to HCI, as it is easier to look for a particular thing of interest than looking through a list, especially while on the move in a busy environment.

However, once selecting a particular icon of interest, i.e. shopping, the screen that appears next is very confusing. There are a number of options to choose from, which do not reflect any meaning of what they might do, hence a potential place for individuals to get lost within the application. Moreover, the map is very simple. There is no support for providing the current location of users or directions to a particular place of interest.

The final drawback of the application is the choice of colour. The application doesn't look attractive and certainly doesn't make the impression, as Google and Peep Laja found in their study, in 50 milliseconds. At first sight, the application does not make the impression but it does after used for a while.

3.6.3. Conclusion

The analysis of the similar applications has been beneficial to this project as it has provided some ideas on the design and layout of the proposed solution. Both applications tend to do the same job, with one offering more features than the other. The drawbacks of both applications have provided the opportunity to avoid making the same mistake when implementing the solution for this project.

3.7. Global Positioning System

Global Positioning System (GPS) is a navigation system made up of 24 satellites. Launched by the U.S. Department of Defence, it was primarily for military use. However, in the 1980's, it became available for civilian use. This system provides the exact location of the user, a piece of information that could be used to work out the distance to another point as well as the speed. This section discusses how GPS work and the errors associated with GPS location discovery.

3.7.1. GPS

The structure of GPS consists of three sections [20]:

- *The space segment* - concerned with the constellation of the GPS satellite.

- *The control segment* - concerned with the maintenance of the master control station
- *The user segment* – concerned with the user and their GPS equipment

The process of determining the user's exact position involves measuring the distance between the GPS receiver and the satellites that are associated by the GPS receiver. The distance is calculated by taking into account the time period that it takes for the signal to travel from the satellite to the GPS receiver.

On the assumption that signals in space travel at the speed of light, in order to calculate the distances, the receiver multiplies the travelling time of the signal by the speed of light. The distance measurements are calculated for 4 satellites in order to calculate a 3 dimensional position: latitude, longitude and altitude [20].

To calculate the travel time period, the receiver must identify when the signal left the satellite and when the signal reached the receiver. Identifying the arrival of the signal is simple, the receiver uses its internal clock to authenticate the time when the signal arrives. However, the main question arises as to how the receiver identifies when the signal left the satellite. GPS receivers are synchronised in coordination with the satellites, thus generate identical digital code. Once the GPS device receives the code from the corresponding satellites, the device can recall when it released the same code by searching its memory bank. From this, the receiver can calculate when the signal left the satellite [20]. It is left to work out the GPS location, the calculation at this point, is a problem of geometry. Since the receiver has identified where each four satellites are and its distance from each satellites, the final calculation of computing the location is completed through trilateration [20].

3.7.2. GPS errors

GPS coordinates are subject to errors due to environmental influences [22] that reduces the accuracy of the GPS position calculated [20]. The environmental influences follow [20]:

- *Atmospheric effects*
  This is concerned with the surroundings of the ionosphere and the troposphere that interferes with the signal, so the signal's speed doesn't correspond to the speed of light. Sunspot activity is also known to alter the signal speed.
- *Measurement noise and distortion of the signal*
  This is concerned with distortion caused by electrical interference or errors from the GPS device itself.
- *Ephemeris data*

This is concerned with errors from the information provided about the satellite orbits. A receiver may have inaccurate data about the location of the satellites hence the positions are calculated using inaccurate data and therefore errors in the final position computed.

- *Small variations in atomic clocks*

  This is concerned with drifts in the atomic clocks based on the satellites. Drifts in the clock can translate to very large position errors, with a fault of one nanosecond in the clocks cause errors of 0.3 meters with regard to the user's position.

- *Selective Availability*

  The Department of Defense implemented a security system as part of the system as a security measure. The security system itself is known to cause GPS position errors [22].

3.7.4. Conclusion

In conclusion, although the Global Positioning System is an extremely valuable feature in providing GPS location, there are many errors associated with the system. The research of GPS has been beneficial to the project as it has provided the fact that GPS locations are not always accurate when first collected. Therefore extra precautions will be taken whilst gathering GPS location of various amenities. Although GPS errors can be reduced, by using methods such as differential correction [20], this not feasible due to the costs associated with the equipment, software and data required to perform the methods and the time limitations associated with the project. On the other hand, GPS location are still very accurate, with many GPS devices maintain an accuracy of 15 meters [52].

3.8. Summary

This chapter has discussed the background reading embarked in order to find the most feasible approach to complete this project. The research revealed today's existing systems are currently utilised and how an opportunity is available to implement a much practical system with the use of mobile devices, that aims to overcome the problems associated in the existing systems. Technologies and methodologies to create the solution were also researched, leading to the selection of an appropriate mobile platform and methodology to develop the system. Finally, mobile limitations and the HCI aspects were explored to gain knowledge of the issues associated with the users and the design of the system.

4. Development

4.1. Introduction

This chapter will discuss the planning that was carried out prior to the implementation of the application. As stated in Section 3.4.2*,* user interface design consists of four stages. The task of planning will ensure that all four stages are covered. It was vital to generate a plan that would outline the main stages of the implementation. The use of the chosen methodology also required extensive planning, as discussed in Section 2.2.

4.2 User Requirements

With the results of the research and the analysis of existing systems, it was feasible to draw up a list of requirements for the proposed solution. This list was exhaustive and therefore a review of the proposed requirements was carried out. The result of the review provided a list of useful features that are thought to be beneficial to the intended users. Moreover, a list of futile features was exposed that are thought to be unfeasible and unbeneficial to the users whist utilising the application. The following list states the final requirements of the proposed solution:

- *Places to shop in Leeds city centre*
  Show a list of places to shop with the ability to call a specific shopping amenity, view the location of a particular shop on a map, visit the shop's website and provide directions to a shop from the user's current location.
- *Places to eat in Leeds city centre*
  Show a list of places to eat with the ability to call the dining place, view the location of a dining place on a map, visit their website and provide directions to a specific dining place from the user's current location
- *Places of interest in Leeds city centre*
  Show a list of the places to see with the ability to call a particular place of interest, show the location of a particular place of interest on a map, visit their website and provide directions to a place of interest from the user's current location.
- *A map of Leeds city centre*
  This map would notify individuals of their current location.
- *A feature for parking in the city centre*
  This feature would provide the ability to call the car park's reception and provide directions to a particular car park from the user's current location.

- *A feature to provide some travel information*

  This feature would provide information of three different travelling modes: train, bus and taxis. Furthermore, the application will allow individual to check the latest train and bus times as well provide the ability to request for a taxi.

- *Latest weather conditions*

  This feature would provide the latest weather conditions in the City Centre.

4.3 Designs

The user requirements gathered provide a clear idea of the features that the application is expected to support. With the proposed requirements agreed, it was viable to produce initial designs for the user interface. The Human Computer Interaction (HCI) issues, discussed in Section 3.4, as well as Apple's Human Interface Guidelines, had major influence on the design decisions. It was important to tailor the designs in order to ensure that the HCI issues are encountered and all the guidelines are met.

- *Vertical scrolling*

  Vertical scrolling is to be discarded throughout the whole application. This feature causes misperception and is thought to affect the navigation of the proposed application, thus create negative impacts on the user experience of the application

- *Reduce user text input*

  The application should solely rely on user's selection where possible in order to reduce users' text input. This decision was influenced by the fact that reducing user's text input results in a better experience as text inputs have proven to be problematic. Moreover, in the context that the application is likely to be used, user text input is not practical.

- *Image and icon size*

  It is important to set the image sizes to a viable dimension. This decision was heavily influenced on the acknowledged fact of the limited screen sizes and upon the decision to avoid the use of vertical scrolling. Images have to be set to certain dimensions such that it does not distract the users from accomplishing their intended task. Moreover, the size of the various tab bar icons had to comply with Apple's guidelines [40]. The guidelines instruct that tab bar icons must be set to 57 pixels by 57 pixels. These limitations are beyond the developer's control however, having said, it is believed that the limitations will help to design an aesthetic user interface.

- *Reduce user's cognitive burden*

  Reducing the users' cognitive load has major influence on the choice of user interface elements. The decision to use standard user interface elements was made upon the fact that individuals are already familiar with the appearance and behaviour of the standard elements. It is though that this will help to promote cognition, of the user interface elements and to some extent support ease of use**.** On the other hand, the use of unfamiliar user interface elements is not usable since it can lead to confusion thus affect the efficiency of accomplishing an intended task.

Initial designs

The initial designs consist of two main stages: the wireframe and the colour designs.

Wireframe

User interface wireframe is a graphical proposal that characterises the layout of a user interface as well as visualising the navigation within the application. With the absence of style, colour and graphics, the main focus of this approach is on the features, navigation and content of the application. Figure 4.1 illustrates the blueprint of the Leeds City Museum information 'scene'. The figure shows various user interface elements such as text areas, buttons and tabs. The wireframe were created using a 'WireFrameSketcher' which provides iPhone stencils, user interface widgets and icons that can be used to mock various user interfaces. The process of creating the wireframe has given a general idea on the layout of the application. Furthermore, it has provided the opportunity to assess whether the content of the application isn't excessively overloaded.

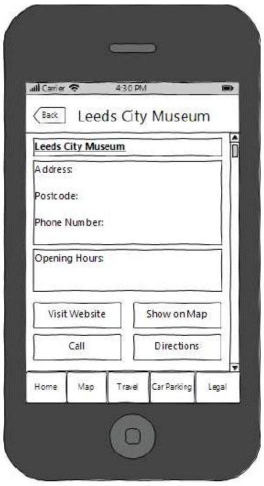


Figure 4.1. Wireframe of the Leeds City Museum information scene

Colour designs

The colour designs offer much more in terms of the aesthetic appearance. The main focus on this design approach is the colours, style and graphics. The most important aspect was the choice of colours, having been aware that it is vital to ensure that the appearance of the application incorporates with its main purpose. Furthermore, since the selection of the appropriate colours relates to the visual appearance of the application, it would mean that the process of making a good first impression is dependent on the choice of the colours. The designs were created using Photoshop CS5. Each user interface elements, such as the buttons, the tab icons, the background, were designed during the pencil and colour blending tools. Figure 4.2 shows the initial colour design of the home 'scene'.

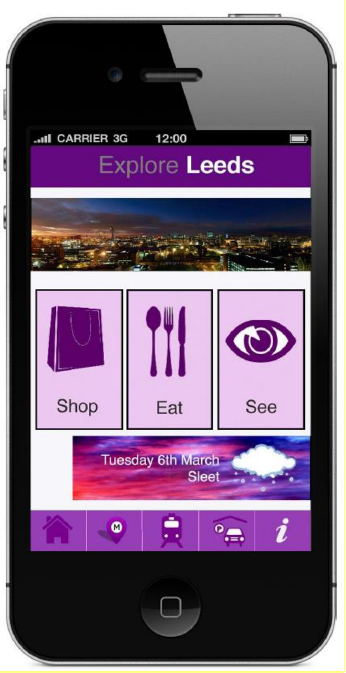


Figure 4.2. The initial design of the Home scene

4.4 User based assessment

Due to the nature of the HCI user interface design process, as discussed in Section 3.4.1, and to gain valuable feedback on the designs, the next phase was to perform a user based assessment. This phase involved using the initial designs as a form of a paper based prototype, a model that could be used to assess a new system. User based assessments expose problems within systems. In relation to the proposed mobile application, it is thought to expose navigational issues. The opportunity to discover issues within the proposed application at an early stage of the project justifies the decision to perform the as assessments.

Originally, the wireframe were set to be used as the paper based prototype for the assessments. However, it is impossible to visualise the experience without the use of colours, style and graphics and therefore it wasn't appropriate and beneficial to perform the assessments with the wireframe. Based on this assumption, it was categorical that the colour designs would be deployed for the user based assessment.

The initial designs were assessed systematically. In order to retrieve useful feedback on the designs, it was important to select participants that reflect the characteristics of the target group. Five participants were chosen as research has shown most usability issues can be detected using four to six individuals [41]. The selected participants were comprised of experienced and inexperienced users of smartphones. The purpose of the assessments was to build a picture of how well participants use the application and to discover problems that exist within the application.

Participants were asked to perform a given list of tasks (Appendix D) and to discuss the good and bad points of completing such tasks, with regard to effectiveness, utility and learnability. At completion, participants were given the opportunity to offer their opinions on how the designs can be improved. This approach was used as it was thought it would provide much more reliable and descriptive feedback from individuals that the application is primarily targeted at. Due to the nature of the questions asked, the results of the assessments were in form of qualitative data. Since the prototype was paper based, it was challenging to assess the designs in a numerical approach at this stage. Having said, the theoretical prototype doesn't provide the similar experience as to a real working system hence it is arguable that the use of a numerical approach at this stage isn't appropriate, which justifies the decision to use such nature of questions.

The result of the user based assessment exposed many unexpected issues. Firstly, 60% of the participants thought that the home screen layout is overcrowded and disliked the arrangements of the weather feature and the image of Leeds nightlife. Various navigational issues were also discovered. For a while, participants were lost after using the travel and the parking features as returning to the home screen was proving to be problematic, due to the unrecognisable icons. The fault for the unrecognisable icons was the choice of colours of the icons, as the colour chosen for the icons was similar to the background colour hence participants found it hard to recognise the meaning of such icons. Similarly, the colour selected for the back buttons was also criticised.

It became apparent that many participants had difficulties in selecting the three buttons on the home screen, with a few displaying their frustrations. Surprisingly, I think that the users had difficulties in selecting the three buttons.

The user based assessment has proven to be very beneficial to the design and the development of the proposed application as it has exposed several arrangement issues which are considered as a major threat to the success of the application. Furthermore, navigational issues have also been exposed therefore the decision to perform the user based assessment has been constructive as it has helped to avoid issues arising at latter stages where they can be very problematical to solve and indeed, time consuming.

4.5. Designs improvements

The next stage involved modifying the initial designs in order to make improvements to the application. Furthermore, it was also an opportunity to solve the issues that arise from the assessments. A few modifications were completed to originate the final designs.

- *Home screen*

  The home screen was modified slightly. The arrangement of the weather feature and the Leeds nightlife image has been exchanged. This modification has responded to the criticism of the home screen arrangement as it has generated a much cleaner interface which rationalises the decision to perform the modification of the home screen.
- *Icons revamp*

  The tab bar icons were revamped. As discussed in Section 4.4, many of the participants were unable to recognise the tab bar icons in the initial designs. This is the cause for the lengthy period for completing a task. The newly revamped icons appear to be similar to standard icons and therefore individuals would be able to recognise the meanings of the revamped icons. Furthermore, a text has been positioned underneath each icon, which promotes the meaning of its icon. The decision to place textual representations underneath each icon was to aid inexperience individuals of smartphones. Overall, the revamp of the icons is thought to have positive influence on the efficiency of completing an intended task.
- *Back button revamp in table view*

  The back button positioned in the table view was also revamped. The colours chosen for the back button was too similar to the background, therefore individuals found it difficult to distinct the back button from the background. The decision to revamp the back button is thought to provide an easier experience of navigating through the proposed application.

4.6. Implementation planning

Since the chosen methodology requires extensive planning prior to implementation, as discussed in Section 2.2, it was essential to plan each iteration before starting the implementation of the mobile application. An iteration has to benefit the system to an extent that the system expands after each iteration. It was thought that the feasible way to do this is to break down the requirements of the proposed solution into a series of builds. Each build corresponds to an iteration. The process of dividing the requirements involved assigning each requirement to a build. It was important to plan each iteration to an extent that each iteration weights similarly, in regard to the amount of work required. This process was proving to be difficult as some requirements needed prerequisite features prior to its implementation. After several attempts, the final implementation plan was originated as show in Figure 4.3.

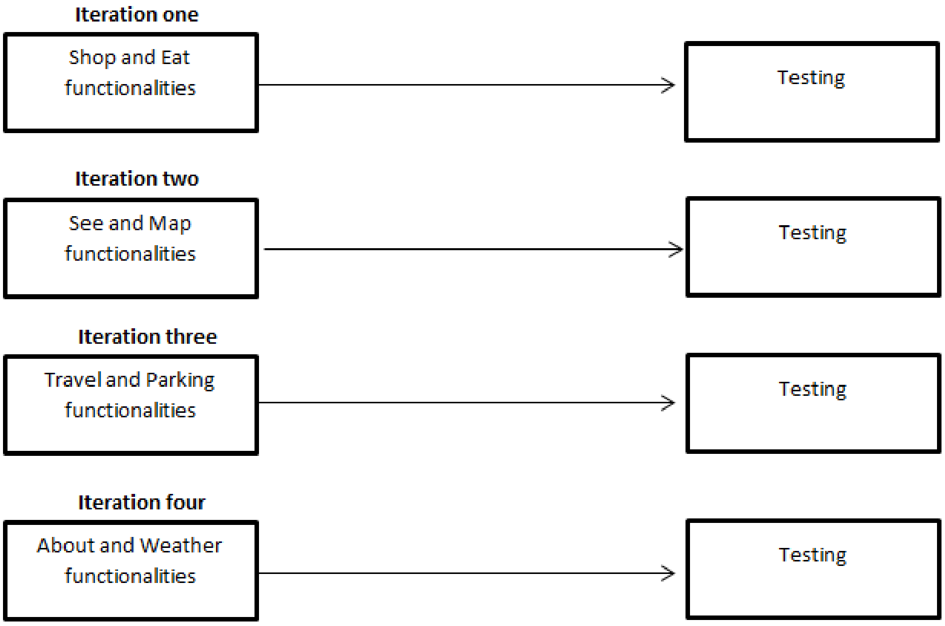

Iteration one
Shop and Eat functionalities
Testing
Iteration two
See and Map functionalities
Testing
Iteration three
Travel and Parking functionalities
Testing
Iteration four
About and Weather functionalities
Testing

Figure 4.3: The implementation plan showing the work for each iterations

4.7. Conclusion

With the user requirements and the designs created and finalised, the foundations of the proposed mobile application have been laid. The gathering of the user requirements has helped to produce the initial designs, in which the opportunity was accepted to gain feedback upon. This opportunity has generated a theoretical design which has assisted to visualise the appearance of the application. With the designs and implementation planning completed, the next stage of the project is to perform the implementation of iteration one.

5. Implementation

5.1. Introduction

This chapter will discuss the implementation of the proposed mobile application, designed and planned in the previous chapter. It will introduce the general process of implementing a mobile application in iOS, followed by discussion relating to each iterations of the implementation and challenges that were encountered during the process.

5.2. Implementation Process

The process of developing any mobile application starts from the application design. With the designs in place, the development of the application is much clear as the translation of the designs to the actual code help overcome major issues and questions that arise during the implementation.

5.2.1. Architecture and design patterns

Regardless of the structure and the form of the mobile application, there are many designs patterns and techniques that are important to acknowledge prior to the implementation of the application. The key pattern is the Model-View-Controller (MVC). This pattern oversees the overall form of the application. The purpose of the MVC is to allocate objects to a specific role within the application as well as to delineate the process that objects experience while communicating with each other. The roles include the model, view and the controller. The model encapsulates the application's data and specifies the methods that can manipulate particular data. The view is essentially an object that the users are presented with, i.e. the user interface. Its major role within the MVC architecture is to display data and react to users' action in order to manipulate the data. The controller notifies the view and the model of any changes within the data or the view. Figure 5.1 illustrates the Model-View-Controller architecture.

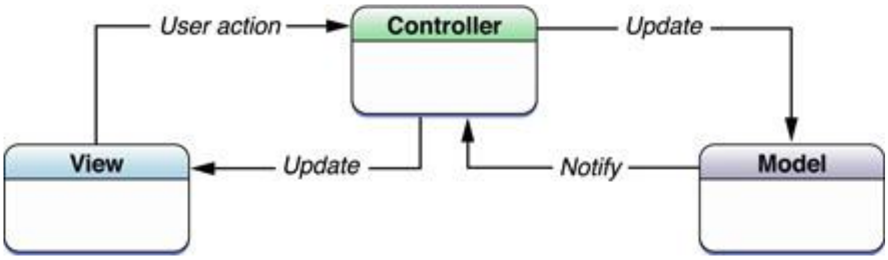


Figure 5.1. The Model-View-Controller architecture

The process of creating an iOS application depends on the form of the proposed application. Nearly all of the development of iOS applications are completed in Xcode, an integrated development environment for implementing iOS based applications. The main areas that are of concern when implementing mobile applications are as:

- ***Form of user interface***

  The selection of the basic view and view controller depends on the complexity and the form of the proposed application. It is essential that developers have an idea of the structure of the application. This justifies the need for the designs generated in the previous chapter. Furthermore, selecting the most suitable view controller makes the implementation of the application easier.

- ***Targeted device***

  This area mainly focuses on the type of devices that the application is proposed for. An iOS application can be implemented for iPhones and iPads. Both devices entail different view controllers and style of coding. It is also possible to create for both devices. However, only one application needs to be created to run on all iPhones whereas for Android based devices, four different implementation of the application is required, as mentioned in Section 3.5.3.1. Furthermore, Xcode provide the functionality to run the application on a simulator. The simulator must match the targeted device in order to demonstrate the performance and the visual appearance when deploying and running the application on a specific device.

  In relation to the project, I thought that it is highly unlikely that individuals would carry around an iPad in a busy place due to social context and the dimensions of the device. Therefore, the decision was made to create the application for iPhone use. This decision also considered the temporal limitation associated with the project.

- ***Use of storyboards***

  Storyboards help developers to visualise the application and the flow of the navigation within the application. This feature helps to manage connections and shifts between view controllers thus streamlining the development.

  In order to address the HCI problems associated with user interfaces, I thought that it is appropriate to use storyboards to implement the proposed application. The storyboard would visualise the navigation of the application. Although, many of the navigation issues were discovered during the user based assessments in the previous chapter, this form of visualisation would help to expose any further navigation issues in the application that may have been missed out during the user assessments, as documented in Section 4.4. This justifies the decision to use storyboards as part of the implementation.

5.2.2. Interface builder

Xcode provides an integrated interface builder to create and visualise the graphical user interface of an iOS application. Developers have access to various palettes and collections of user interface objects such as text fields, sliders, web view objects. The user interfaces can be constructed by dragging user interface objects onto the view controller. Many of the objects may perhaps emit an action which is coded in the view controller's corresponding implementation file such as the *'Aspecto's* implementation file shown in Figure 5.7. A connection between the object and its action code is created in order to emit the action. The process of initialising the connections before runtime improves the performance of the mobile application as well as simplifying the development process.

5.2.3. Memory management

Memory management is the process of allocating and de-allocating memory in devices during runtime. The memory limitation of mobile devices [25] makes this practise even more significant. The volume of memory required entirely depends on the type of the application, an application that requires less memory demonstrates a well-coded implementation. During the implementation of an iOS application, memory management can be accomplished in two ways: manual retain-release and Automatic Reference Counting (ARC).

Manual retain-release is essentially the developer managing the object's lifecycle throughout the execution of the application. Allocation and release of objects are specified within the code. This method requires manual effort and places a heavy burden on the developer as ineffective memory management can result in serious consequences.

ARC releases the burden from the developer. In ARC, the same method is used except that the system takes control of allocating and de-allocating of memory by inserting relevant code during runtime. ARC is not a garbage collection, the key mechanism of ARC is the ability to free an object once strong references to the particular objects are withdrawn.

In relation to this project, it was thought that in order to overcome the memory limitations of mobile devices, the ARC is the most appropriate memory management approach to adopt. This approach would ensure memory within the application and the device is managed efficiently and avoids memory problems. Furthermore, ARC would assist in maintaining and improving the performance of the proposed application during runtime.

5.3. Iteration one

5.3.1. Home screen

As planned in the previous chapter, the aim of the first iteration was to create the home screen, the shopping and the eating features. The implementation of the home screen aims to establish a solid platform of the proposed system for supplementary implementations to be developed upon such as the shop and the eat features. This justifies the reason to implement the home screen within the first iteration.

Due to the nature of the proposed mobile application, the home screen was embedded into a tab bar controller. Tab bar controllers help to arrange the mobile application so that different data is organised into separate views to avoid information overload and aid simplicity, in regard to discovering data that is of interest to the user. Furthermore, the tab bar controller was combined with the navigation controller. Navigation controllers control a stack of view controllers to provide a hierarchical navigation throughout the application. This justifies the decision to embed the application within a navigation controller as it is thought to deliver easier and simpler navigation.

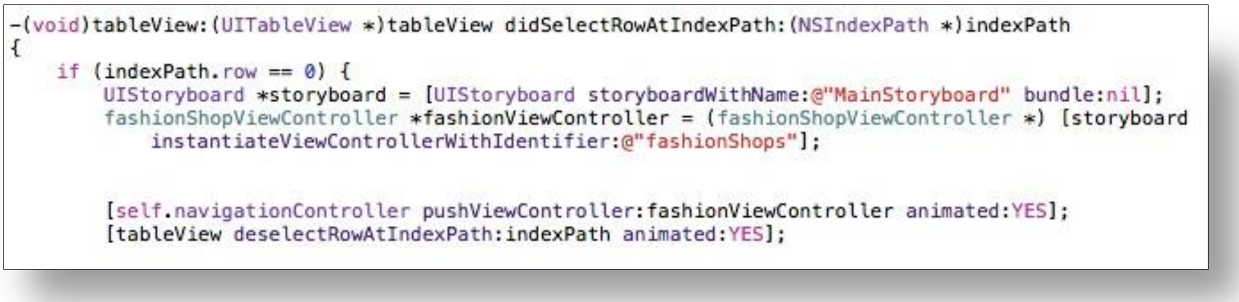
The implementation of the home screen also involved introducing custom navigation and tab bars. The initialisation of the custom navigation bar was straightforward, setting the background of the custom navigation bar to an image generated in Photoshop. On the other hand, the initialisation of the custom tab bar was a little tricky. Two set of images were essential for each tab: one for the selected action and one for the unselected action. Furthermore, the coding for each tabs were implemented within the corresponding implementation file such as the 'map view controller' implementation file. The code specifies the selected and unselected images as well as the implementation of selecting a specific tab.

Additionally, the home screen consisted of a picture carousel. The aim of this feature is to change images after a certain interval. There were two precise approaches to implement this feature. The principal approach was essentially a procedural programming style, with the use an if statement, a list of images and timer to constantly generate transition of images chosen randomly. The alternative was to make use of the carousel library. The library obtained from GitHub [47] contains the *NSCarousel* class. The class provides a simple implementation of the picture exchanging action with various effects that can be set to the developer's choice, such as cylindrical and flat. The *NSCarousel* class was applied as it was felt to be the most appropriate and efficient approach to use.

5.3.2. Shop, eat and see features

Iteration one also provided the shop, eat and the see features. With the main platform of the mobile application established, the implementation of the shop, eat and the see features were fairly straightforward, in regard to the infrastructure of the mobile application.

The implementation of all three features involved the use of table views. In iOS, table views present data in a list of rows. The motive behind the use of table view was that it allows navigation through data in a hierarchical manner, thus reinforces a simpler and cleaner navigation experience. Rows are identifiable and accessed via indexation, numbered from *0 to n -1*. Hierarchically, all three features' table views are of a depth of two. Programmatically, the main implementation was the coding of the transition between the selection of a particular cell and its corresponding view. This implementation was completed with the use of *if statements* and the *NSIndexPath* class, as illustrated in Figure 5.2.



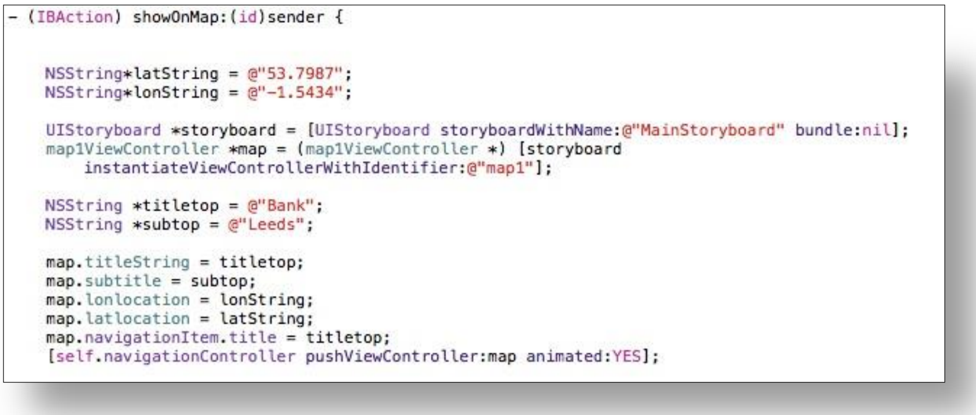
```
-(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (indexPath.row == 0) {
        UIStoryboard *storyboard = [UIStoryboard storyboardWithName:@"MainStoryboard" bundle:nil];
        fashionShopViewController *fashionViewController = (fashionShopViewController *) [storyboard
            instantiateViewControllerWithIdentifier:@"fashionShops"];


        [self.navigationController pushViewController:fashionViewController animated:YES];
        [tableView deselectRowAtIndexPath:indexPath animated:YES];
```

Figure 5.2. The implementation of the 'didSelectionRowAtIndexPath' action

The *'indexPath'* method executes when a user selects a row in the table view. Figure 5.2 shows a *'if statement'* which defines the action that would execute if '*indexPath.row == 0',* in other words if the users select row zero. The next bit of code essentially creates an instance of the *'fashionShopViewController'*, an object which represents a view that displays when the '*Fashion'* row is selected. The *'instantiateViewControllerWithIdentifier'* construct defines the unique name that identifies the *'fashion'* shop interface, built using the '*interface builder'*, so that the application can identify which view to display. Finally, the '*fashionShopViewController'* object, instantiated as '*fashionViewController'*, is passed on to the navigation controller. Similar code was used to define the action of selecting the other rows, each which uniquely instantiates and identifies its corresponding view object and user interface.

A content view controller was used to implement the individual amenities' information view. The information view consisted of many user interface elements such as textboxes and buttons. These elements were programmatically initialised, with memory allocated for each user interface object. The information was manually inputted into the textboxes through Xcode's interface builder. Four buttons were also implemented as part of the information view: *show on map*, *directions*, *call* and *visit website*. The actions for each buttons were coded in the corresponding implementation files which were linked to relevant buttons, Figure 5.7.

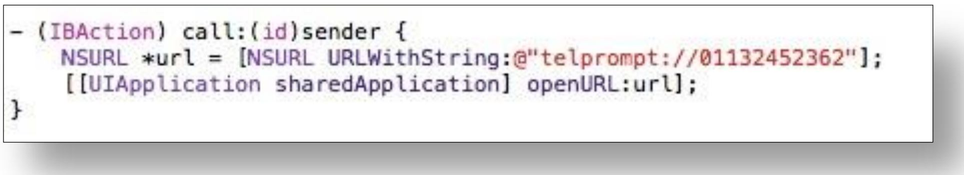
The '*Show on Map*' button shows the location of a specific place on a map, Figure 5.3. The Figure shows the implementation of the '*Show on Map'* button, the example given is for the *'Bank Fashion'* shop. An instance of the '*map1ViewController'* was created which required four arguments to show the location via an annotation pin: *'titleString', 'subtitle', 'lonlocation'* and '*latlocation*'. Four '*NSString'* objects were created to hold string values representing the arguments. In order to display the map on the screen, the '*map'* instance was passed onto the navigation controller.



```
- (IBAction) showOnMap:(id)sender {


    NSString*latString = @"53.7987";
    NSString*lonString = @"-1.5434";

    UIStoryboard *storyboard = [UIStoryboard storyboardWithName:@"MainStoryboard" bundle:nil];
    map1ViewController *map = (map1ViewController *) [storyboard
        instantiateViewControllerWithIdentifier:@"map1"];

    NSString *titletop = @"Bank";
    NSString *subtop = @"Leeds";

    map.titleString = titletop;
    map.subtitle = subtop;
    map.lonlocation = lonString;
    map.latlocation = latString;
    map.navigationItem.title = titletop;
    [self.navigationController pushViewController:map animated:YES];
```

Figure 5.3. The implementation of the 'Show on Map' button

The '*Call'* method dials a specific number specified as a NSURL object, as shown in Figure 5.4. The implementation is fairly simple, creating an instance of the '*NSURL'* class which represents the phone number and passing the object onto the UIApplication. The UIApplication object represents the root object of the application and in this case invokes the call.

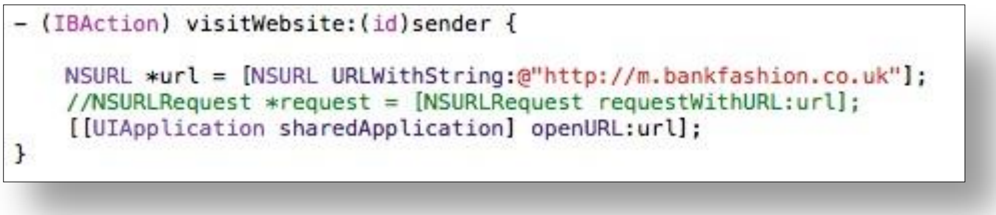


```
- (IBAction) call:(id)sender {
    NSURL *url = [NSURL URLWithString:@"telprompt://01132452362"];
    [[UIApplication sharedApplication] openURL:url];
}
```

Figure 5.4. The implementation of the 'Call' method

The 'Visit Website' method directs the application to a specified website, as show in Figure 5.5. The implementation is similar to the 'Call' method except that the NSURL object represents a HTTP URL.

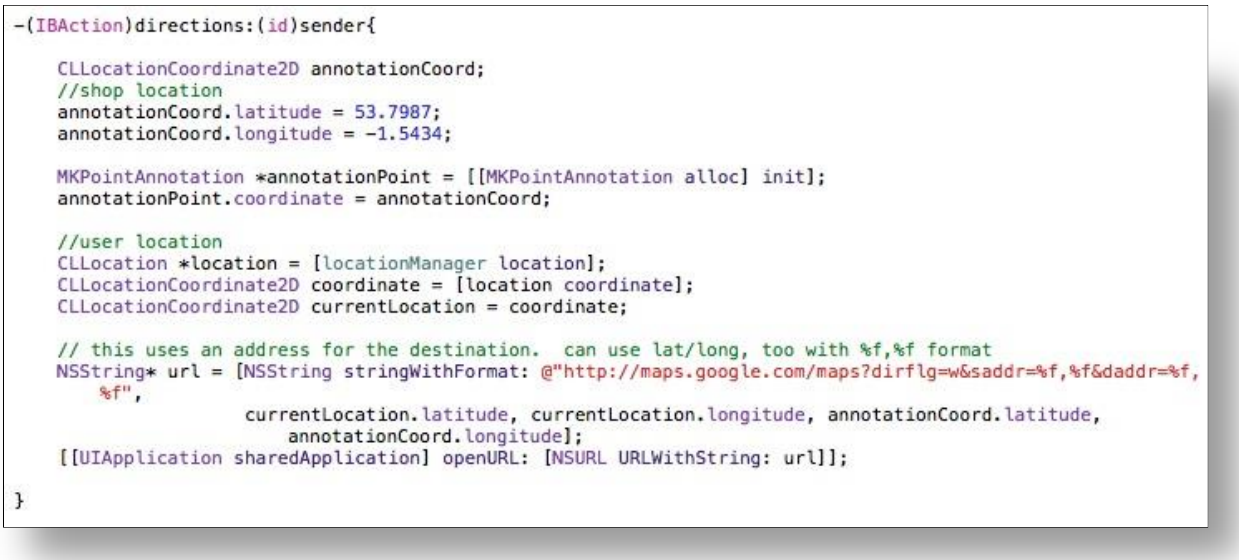
The UIApplication invokes Safari web browser, uploading the HTTP address specified which in this case is the ‘Bank Fashion’ website.



```
- (IBAction) visitWebsite:(id)sender {

    NSURL *url = [NSURL URLWithString:@"http://m.bankfashion.co.uk"];
    //NSURLRequest *request = [NSURLRequest requestWithURL:url];
    [[UIApplication sharedApplication] openURL:url];
}
```

Figure 5.5. The implementation of the ‘Visit Website’ method

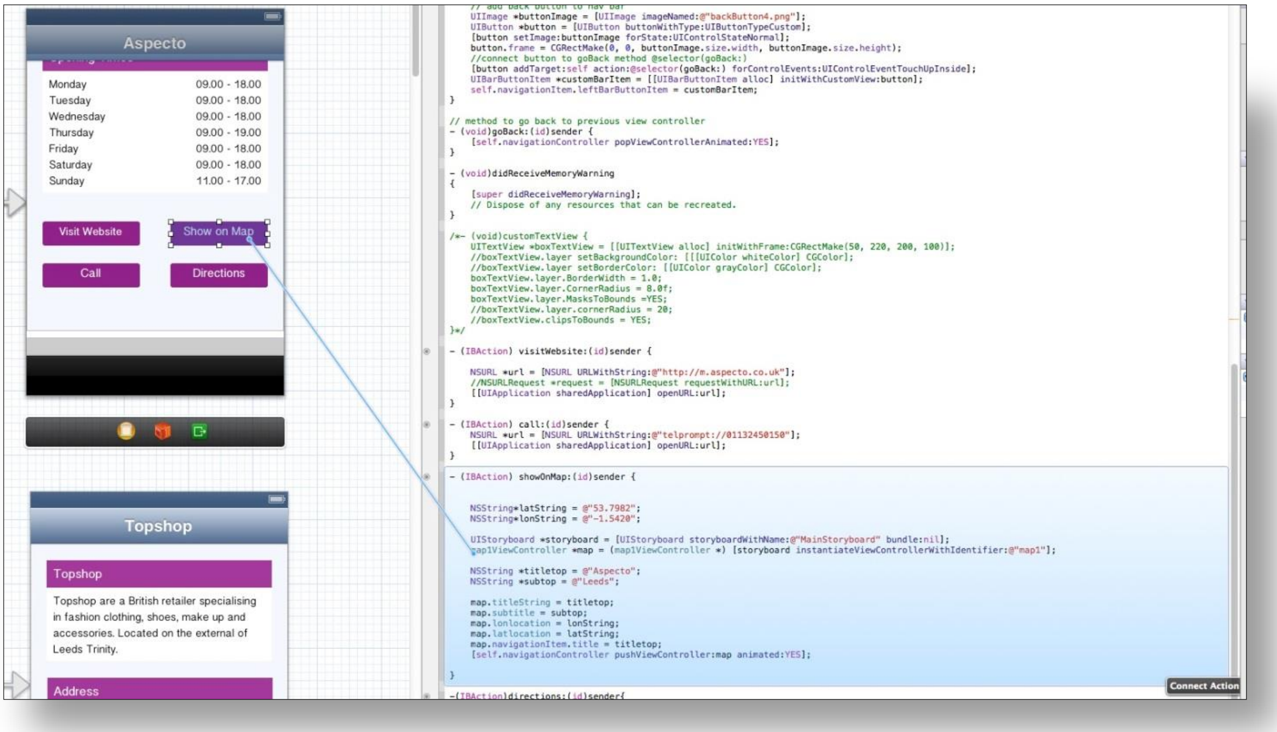
The ‘*Directions´* method provides directions from the user’s current location to a particular place specified by GPS coordinates, as shown in Figure 5.6. The GPS coordinates are represented as *‘CLLocationCordinate2D’* data types which are passed onto an NSURL object. The ‘*CLLocationManager’* is used to retrieve the current location of the user. The method uses Google Map’s URL Scheme to send a request for directions. The *‘NSURL’* object is passed on to the *UIApplication* which invokes the Google Map application. Using the arguments passed, Google Map provides the directions between the two locations.



```
-(IBAction)directions:(id)sender{

    CLLocationCoordinate2D annotationCoord;
    //shop location
    annotationCoord.latitude = 53.7987;
    annotationCoord.longitude = -1.5434;

    MKPointAnnotation *annotationPoint = [[MKPointAnnotation alloc] init];
    annotationPoint.coordinate = annotationCoord;

    //user location
    CLLocation *location = [locationManager location];
    CLLocationCoordinate2D coordinate = [location coordinate];
    CLLocationCoordinate2D currentLocation = coordinate;

    // this uses an address for the destination.  can use lat/long, too with %f,%f format
    NSString* url = [NSString stringWithFormat: @"http://maps.google.com/maps?dirflg=w&saddr=%f,%f&daddr=%f,
        %f",
                     currentLocation.latitude, currentLocation.longitude, annotationCoord.latitude,
                        annotationCoord.longitude];
    [[UIApplication sharedApplication] openURL: [NSURL URLWithString: url]];

}
```

Figure 5.6. The implementation of the ‘Directions’ method

The action for each button has to be connected to its corresponding button. Other programming languages, such as Java which is used for Android development, provide a low – level approach, where actions are linked to buttons explicitly. Xcode provides a high-level approach, which implicitly generates the code for the connections. This involves dragging a link between the button and its corresponding action, as show in Figure 5.7. This process was applied for all of the buttons in the application.



Monday	09.00 - 18.00
Tuesday	09.00 - 18.00
Wednesday	09.00 - 18.00
Thursday	09.00 - 19.00
Friday	09.00 - 18.00
Saturday	09.00 - 18.00
Sunday	11.00 - 17.00

Topshop are a British retailer specialising in fashion clothing, shoes, make up and accessories. Located on the external of Leeds Trinity.

```
    // add back button to nav bar
    UIImage *buttonImage = [UIImage imageNamed:@"backButton4.png"];
    UIButton *button = [UIButton buttonWithType:UIButtonTypeCustom];
    [button setImage:buttonImage forState:UIControlStateNormal];
    button.frame = CGRectMake(0, 0, buttonImage.size.width, buttonImage.size.height);
    //connect button to goBack method @selector(goBack:)
    [button addTarget:self action:@selector(goBack:) forControlEvents:UIControlEventTouchUpInside];
    UIBarButtonItem *customBarItem = [[UIBarButtonItem alloc] initWithCustomView:button];
    self.navigationItem.leftBarButtonItem = customBarItem;
}

// method to go back to previous view controller
- (void)goBack:(id)sender {
    [self.navigationController popViewControllerAnimated:YES];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

/*- (void)customTextView {
    UITextView *boxTextView = [[UITextView alloc] initWithFrame:CGRectMake(50, 220, 200, 100)];
    //boxTextView.layer setBackgroundColor: [[[UIColor whiteColor] CGColor];
    //boxTextView.layer setBorderColor: [[UIColor grayColor] CGColor];
    boxTextView.layer.BorderWidth = 1.0;
    boxTextView.layer.CornerRadius = 8.0f;
    boxTextView.layer.MasksToBounds =YES;
    //boxTextView.layer.cornerRadius = 20;
    //boxTextView.clipsToBounds = YES;
}*/

- (IBAction) visitWebsite:(id)sender {

    NSURL *url = [NSURL URLWithString:@"http://m.aspecto.co.uk"];
    //NSURLRequest *request = [NSURLRequest requestWithURL:url];
    [[UIApplication sharedApplication] openURL:url];
}

- (IBAction) call:(id)sender {
    NSURL *url = [NSURL URLWithString:@"telprompt://01132450150"];
    [[UIApplication sharedApplication] openURL:url];
}

- (IBAction) showOnMap:(id)sender {


    NSString*latString = @"53.7982";
    NSString*lonString = @"-1.5420";

    UIStoryboard *storyboard = [UIStoryboard storyboardWithName:@"MainStoryboard" bundle:nil];
    map1ViewController *map = (map1ViewController *) [storyboard instantiateViewControllerWithIdentifier:@"map1"];

    NSString *titletop = @"Aspecto";
    NSString *subtop = @"Leeds";

    map.titleString = titletop;
    map.subtitle = subtop;
    map.lonlocation = lonString;
    map.latlocation = latString;
    map.navigationItem.title = titletop;
    [self.navigationController pushViewController:map animated:YES];

}

-(IBAction)directions:(id)sender{
```

Figure 5.7. The connection between the 'Show on Map' button in the Aspecto view and the 'showOnMap' method in Aspecto's implementation file.

Each view in the application has its own header and implementation files. The header file contains the variables and methods declaration and the implementation file contains the implementation of the methods defined in the header file. Each shop, eat and see places' header and implementation files were similar. The only differences were the content itself, the website URL, the telephone URL and the GPS coordinates.

5.3.3. Changes to implementation plan

After the implementation of the shop feature, it become apparent that the eat and see features are precisely similar to the shop feature. After a critical analysis of the implementation plan proposed in the previous chapter, I thought it would be feasible to implement the see feature in iteration one along with the shop and the eat features. The revised implementation plan is illustrated in Figure 5.2.

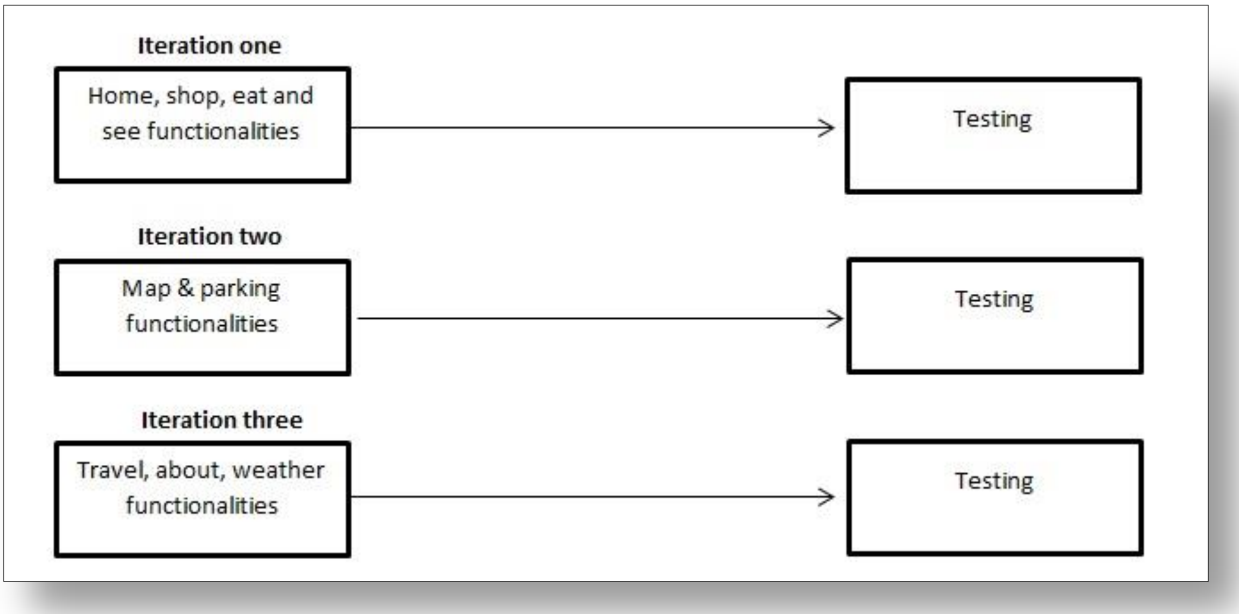

Iteration one
Home, shop, eat and see functionalities
Testing
Iteration two
Map & parking functionalities
Testing
Iteration three
Travel, about, weather functionalities
Testing

Figure 5.8. Illustration of the revised implementation plan

5.3.4. Iteration one testing

Due to the nature of the chosen methodology it was essential to conduct testing of each feature after its completion. The main focus of the testing was to ensure that all features are operating correctly, with regard to the outcome of the actions when selecting and clicking buttons and other user interface elements. In order to perform the testing, a table was derived (Appendix E). The table consist of four columns: 'procedure', 'expected result', 'pass or fail' and 'actual results'. The '*procedure'* column defines the actions performed during the testing, with the '*expected result'* column specifying the anticipated result of the consequence of performing such actions. Moreover, the '*pass or fail'* column defines whether the outcome of the actions corresponds to the expected result and the '*actual results/ comment'* column describes the result of the actions performed along with any failures during the testing.

- ***Home screen***

  Button not connected - The shop button on the home screen didn't respond when clicked upon. The fault behind this problem was that the button wasn't connected to its corresponding action.

- ***Shop***

  Content size error – Many of the errors were associated with the content size of the individual amenity's view. During development the content size was set to 200 pixels. It became apparent, during the testing, that 200 pixels is not sufficient as two of the four buttons were out of sight. In order to solve this problem the content size was increased to 280 pixels. The trial and error method was applied, increasing the size by 20 pixels until an appropriate size originated. The change in the content size had to be applied throughout the feature, since all of the amenities' view controller content size was set to 200.

  Lost in device error - The lost in device error is associated with the specific situation that user's may end up in after selecting the *call* button. This error was discovered when testing the call buttons in order to identify whether the app response to such action. The response to the selection of the button was operating correctly however, when the call had finished, the device didn't direct itself back to the application. It is thought that such situation can result in inexperienced users being confused and lost. The problem was solved by changing the call function '*tel://*' to '*telprompt://*'. The use of the '*telprompt*' function has allowed the device to shift back to the application after the call has ended. Since the call button is provided within every amenity's view controller, the change had to be applied to every implementation file associated with the shop features.

  Directions error – A minor error existed in the directions utility. The execution of the request for directions from Google was successful however, the directions provided were for driving. The error existed in the URL scheme as the *direction mode* was set to '*driving*'. Although the directions were correct, it is thought that users are more likely to request for the directions while on foot hence the *direction mode* was set to '*walking*'. Again, since the directions are accessible from every amenity's view controller, the changes had to be applied throughout the shop feature.

- ***Eat***

  Back navigation button error – The back navigation button error existed in the *Da Mario's Pizzeria* view controller. Upon selection of the back button, the application exhibited no response. The error was due to the back button not being connected to an action. This error was blamed on human errors, which occur infrequently. To solve the problem, the button was connected to its corresponding action.

Website cannot open error – The website cannot open error existed in the McDonalds view controller. This error prompted a *'website cannot open as it cannot be found on the server'* message in Safari. The error was caused by a grammar mistake in the URL address, again blamed on human error. The error was solved by spell-checking the URL. Since the error existed in one view controller, the change only had to be applied to that particular view controller.

- ***See***

  Directions not available error – During testing, the *'directions not available'* error occurred. The error simply means that directions for the particular route are not available. This error occurred due to invalid longitude and latitude. In order to solve this problem, the legitimacy of the longitude and latitude were verified in Google Maps. This showed that the coordinates are not valid and hence must have been overlooked when verifying the GPS coordinates. The trial and error method was applied until valid GPS coordinates were obtained. After updating the GPS coordinate programmatically, the error was resolved. The change in the GPS coordinates was only applied for the Leeds Art Gallery.

5.4. Iteration two

5.4.1. Map functionality

The second iteration involves the implementation of the map feature. This feature shows the location of the user along with numerous map pins that represent the location of the shop, eat and see amenities.

The map was initialised in its implementation file by using the *MKMapView* object. This object offers a map interface embedded inside the application and supports standard map interactions such as two - fingers swiping and standard dragging, the interactions result in the adjustment of the zoom level and change in the map position respectively. In order to show the user's current location, a representation of the current location of the user had to be retrieved. The retrieval of the current location involved the use of the *CLLocationManager* class which provided the location in GPS coordinates format. This feature is thought to help individuals in situations where they are lost in the city centre and would like to find out where they currently are. The decision was made to implement the map feature under the second tab as it is easily accessible and therefore reduces the user's burden in regard to finding the correct information efficiently.

Google Maps was chosen as oppose to Yahoo Maps. Google Map is very popular among smartphone users, especially for users of iPhones. The reputation of Google Map has not unnoticed, 10M downloads in less than 48 hours [39]. The popularity of Google Map suggests that users are likely to be familiar with the functions provided by Google Map as opposed to an alternative Yahoo Maps. Since users are familiar with Google Map from previous usage of the application, it is thought to make the use of the proposed application a lot easier since it doesn't place any burden upon the user, in regard to learnability. Such fact justifies the decision to utilise Google Map.

Along with marking the user's current location, the map also displays annotation of the shop, eat and see amenities. Each amenity is represented through an '*annotationPoint'* object. Initialisation of the annotation point object involved specifying the location of the amenity through its GPS coordinates. A title and a subtitle were also passed on to the annotation point, the latter describes the category of the particular amenity and the other shows the name of the particular place. In Figure 5.3, the map on the left shows the location of '*Aspecto*'. The map on the right shows the result of selecting such annotation point. As Teevan et al [9] concluded, although users tend to search for information relating to their destination 60% of the time, it means that 40% of the time individuals tend look for information relating to their current location. This justifies the reason to add annotation points of the amenities as users can see the places closer to their location by interpreting the map.

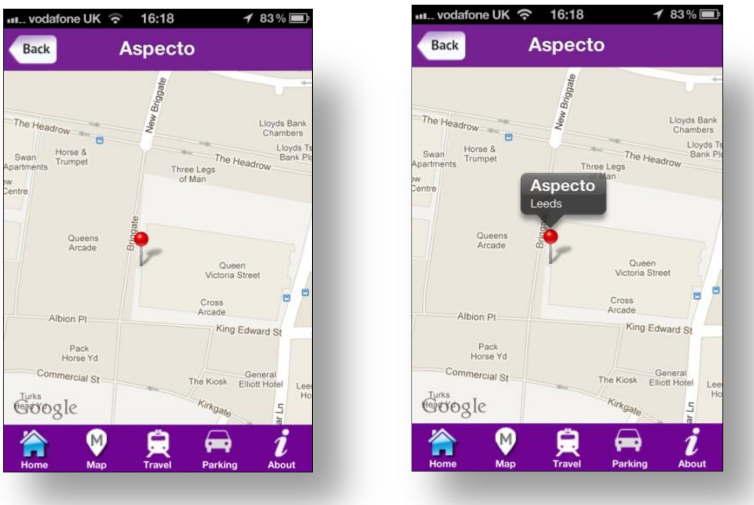


Figure 5.9. The map feature showing the location of Aspecto.

5.4.2. Parking

Iteration two also delivered the parking feature. The purpose of this feature is to provide users with a list of car parks around the city centre. Each car parking facility has its own view, presenting various information regarding car parks. Each view also consisted of four buttons which provides useful information to the users when selected.

The implementation of the parking feature involved utilising the 'UI*TableView'* class in order to provide navigation in a hierarchal manner. Selection of a specific row invokes its corresponding content view controller, an action programmatically coded. The content view, initialised by the use of Xcode's interface builder, composes of textboxes and buttons. The buttons are programmatically linked to their corresponding actions.

The main focus of this feature was to provide numerical information that represents the approximate distance between the user's current location and each individual car park. This involves calculating the distance between two GPS coordinates, the GPS coordinates of the users' current location and of the individual car parks. There are two key mathematical methodologies that help to deliver this feature: The Pythagorean Theorem and The Great Circle method.

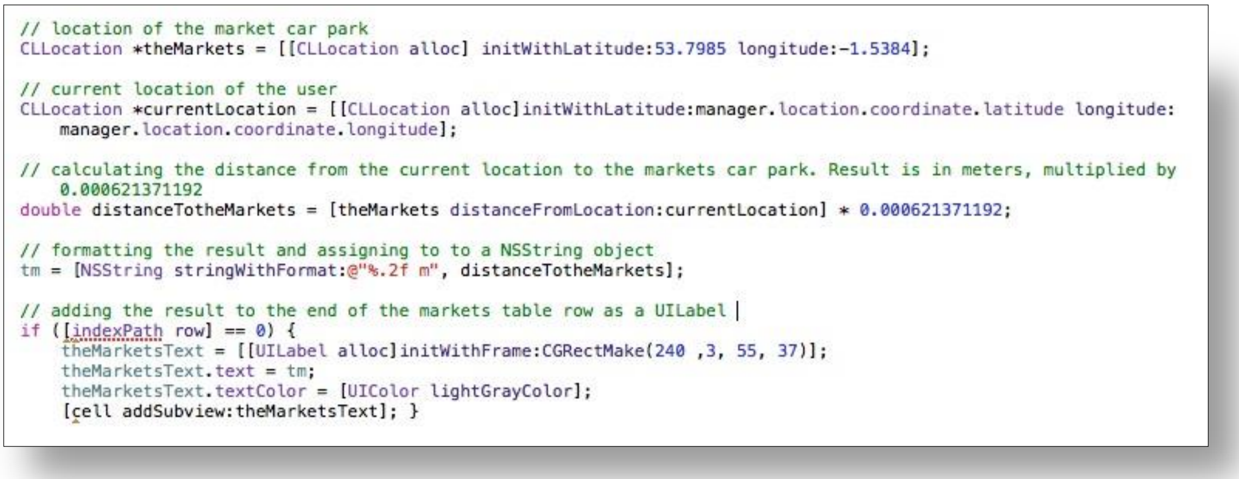
The Pythagorean Theorem is the simplest out of the two methodologies. In order to calculate the distance using this method, each GPS coordinate has to be in Universal Transverse Mercator (UTM) coordinate system, Universal Transverse Mercator. UTM is another form of a coordinate system, defining a position on the Earth. UTM are popular among hikes and are generally used by search and rescue operators. UTM provide earth positions by two sets of numbers, the first is the easting which expresses how far the position is from east and the second is the northing which expresses how far the position is from the north. With the two sets of coordinates the Pythagorean Theorem is applied, which results in a numerical representation. This result represents the distance between the two points.

The Pythagorean Theorem is relatively simple and is only precise for short distances [42]. It gets a lot more complicated when calculating longer distances due to the curve of the earth's surface and therefore requires advanced trigonometry formulation to retrieve more accurate calculations. The Great Circle method is used to calculate distances over a long range. The method is performed using the same coordinate system, but a difference formula (Figure 5.10) is applied. Since potential users are able to use the proposed application anywhere on the Earth prior to visiting Leeds, the decision was taken to use The Great Circle method since it offers much more accuracy. The distance calculation was coded so that, on selection of the travel tab, the calculation function would execute. This would avoid any lag during the start-up of the application. The calculation is performed for all five car parks, with the result given in meters. At run time, the results are converted to miles by multiplying each value by 0.000621371192. With the use of the ‘UIL*abel’* class, the calculated distances were displayed at the end of each row.

The implementation of this is demonstrated in Figure 5.11. The ‘*distanceFromLocation’* method defines The Great Circle method.

$$c = \operatorname{acos}\left(\frac{\sin(\mathrm{Lat2}) - \sin(\mathrm{Lat1}) \times \cos(\mathrm{d})}{\cos(\mathrm{Lat1}) \times \sin(\mathrm{d})}\right)$$

Figure 5.10. The Great Circle formula for calculating distances [43]



```
// location of the market car park
CLLocation *theMarkets = [[CLLocation alloc] initWithLatitude:53.7985 longitude:-1.5384];

// current location of the user
CLLocation *currentLocation = [[CLLocation alloc]initWithLatitude:manager.location.coordinate.latitude longitude:
    manager.location.coordinate.longitude];

// calculating the distance from the current location to the markets car park. Result is in meters, multiplied by
    0.000621371192
double distanceTotheMarkets = [theMarkets distanceFromLocation:currentLocation] * 0.000621371192;

// formatting the result and assigning to to a NSString object
tm = [NSString stringWithFormat:@"%.2f m", distanceTotheMarkets];

// adding the result to the end of the markets table row as a UILabel |
if ([indexPath row] == 0) {
    theMarketsText = [[UILabel alloc]initWithFrame:CGRectMake(240 ,3, 55, 37)];
    theMarketsText.text = tm;
    theMarketsText.textColor = [UIColor lightGrayColor];
    [cell addSubview:theMarketsText]; }
```

Figure 5.11. The implementation of the approximate distance from the user’s current location to ‘The Markets’ car park

5.4.3. Iteration two testing

- ***Map***

  User location error – The *user location* error occurred when selecting the *Map* tab. Although, the error doesn’t interrupt the application or the user, it was important to consider this error as it prevents users from perceiving their current location. In order to solve the problem, the ‘*showUserLocation’* function was set to a Boolean value, *YES*. This change was applied within one implementation file.

- ***Parking***

  Directions – The directions error within the parking utility was merely the fact that the directions provided were for walking. It is thought that since the parking utility is more likely to be used while users in their vehicles, the most appropriate direction mode would be *driving*. Also, the routes provided for the walking mode contain streets that are not passable by vehicles. This is important especially for directions within the city centre where many restrictions are likely to be in place in order to reduce traffic. To solve the problem, the direction mode within the URL scheme was set to driving. Since the problem was associated with every car park's view controller, the changes had to be applied throughout the individual car park's implementation file.

5.5. Iteration three

5.5.1. Travel

The final iteration involved the implementation of the travel feature. The key purpose of this feature is to provide information regarding to trains, buses and taxis. As proposed in the previous chapter, the feature would provide the ability to check the latest train and bus times. Furthermore, the taxis section would provide relevant information relating to the taxis in the city centre.

In order to separate the content into categories to aid ease of use, the *tableview* class was used to implement the feature. The selection of the train row would invoke another view which presented two cells: Leeds Train Station and Latest Train Times. Furthermore, the selection of the two cells would invoke their corresponding view. The Leeds Train Station view generally provides directions to the train station among other services such as '*show on map*' and '*call*'. A similar approach was taken in order to implement the bus feature.

The main focus on the travel feature was to provide users the ability to check for the latest train and bus times. Originally, web services provided by The National Rail and Metro were supposed to be used to provide users with the ability to check for the latest train and bus times. Plans were in place to build the travel feature around the web services. However, due to legal reasons, licenses had to be obtained from the corresponding operators and upon the request for the license it became clear that the process of obtaining such licenses is a lot more problematic that thought and therefore the implementation of this feature was turning out to be very challenging. Although it was emphasised that such web services would be used for the educational purposes and not for profit, unfortunately both requests were rejected. Therefore, the proposed travel feature had to be revised. After analysis, it became apparent that the most suitable approach to implement this feature was to embed a web view within the application. The web view would display the corresponding train and bus operator's website where users can check the latest train and bus times. This involved embedding a web view class with in the content view.

The final aspect of the travel feature was taxis. This feature presents users with the taxis companies that operate within the city centre. Users have the ability to ring for a taxi and also see the location of the corresponding taxis firm's office.

5.5.2. About

The About section of the application was also implemented under the third and the final iteration. Since the application will be used by the general public and that many of the services used were of third party such as the weather service, it was important to implement this section in order to clarify the rights of the individuals using the application and the individual implementing the application and also credit corporation as necessary. Essentially, the decision to implement this feature is to explain the purpose of the application and also to minimise legal disputes. This section involved the use of a content view controller and consisted of mainly text boxes. Due to the limitation of the screen, a scroll view was embedded in order to avoid cluttering the screen and to separate information in an appropriate manner.

5.5.3. Weather

The final implementation proposed within the third iteration is the local weather feature. The need for the weather feature was important as the analysis of existing system demonstrated that most travel guides have some kind of weather update, so it was thought that majority of the users would expect to see the latest weather updates.

From the exploration of many weather web services, it became apparent that the Yahoo and the Wunderground weather service are the most appropriate. This decision was based on the fact that a free weather service was required due to many limitations associated with the project and that both services are known for providing accurate and valid weather updates. Yahoo distributes a weather RSS feed that provides up to date weather information for a specific location. The feeds are dynamically generated upon request which follows a basic HTTP GET approach. The response gained from such request is a XML document.

The structure of such request follows:

http://weather.yahooapis.com/forecastrss?w=26042&u=c

The request starts with a basic URL followed by two parameters. The *w* parameter is to specify the *WOEID* which essentially represents the location of the weather forecast. This parameter can be retrieved by using the WOEID converter provided by Yahoo. The second parameter is to specify the degree unit.

The XML document returned as part of the response contains four different elements, to which the *item* element is of our interest. This element contains the data that provides the actual weather forecast. The next phase of the process involved parsing the XML document for the item element. Due to the complexity of the XML document, this process was extremely difficult as XML parsing in iOS isn't straightforward. An XML parser obtained from the internet [46] was used to perform such parsing of the XML document. This involved inspecting the XML document for tags that are of my interest, the tags that contain the actual weather information. I had to code the method to search for specified tags and store the retrieved tags in a dictionary. Once the dictionary is initialised with the retrieved data, I iterated through it to retrieve the data contained within the tags. Finally, it was left was to present the data. Since the home screen is the first screen that users encounter, it was thought to be the most appropriate screen to display such information. In order, to maintain consistency in the layout of the application, the data was presented as a HTML object hence the web view class was utilised.

5.5.4. Iteration three testing

- ***About***

  Scroll error – The scroll error is associated with the scrolling of the A*bout* view.. Due to the screen size limitation of the devices, the content had to be presented in a scroll view. A scroll view authorises developers to present information that goes beyond the size of the device and allows users to scroll within the content. During testing, it became apparent that it was impossible to scroll through the content. The problem was solved by enabling scroll within the content.

- ***Travel and Weather***

  No errors were discovered during the testing of the travel and the weather features.

5.7. Conclusion

The process of implementing the proposed solution to the project has been documented, along with the various challenges faced and key decisions made. In total 199 files were programmed to implement the application. The testing process after each iterations was discussed. The next stage is test the overall application, in terms of performance and system resources usage.

6. Testing

6.1. Introduction

This chapter will discuss the overall testing of the application implemented in the previous chapter. Although the user evaluation is vital for such application, it was important to assess the performance of the application in terms of CPU and memory usage in order to assess whether the application doesn't hog the performance of the device.

Mobile devices are associated with many limitations, particularly the low processing power and the limited memory capability as mentioned in Section 3.3.. Therefore, it is important to ensure that the implemented application doesn't take up too much of the resources as such consequences can be critical.

In order to test the device's performance, a performance analyst tool was used. This tool, called *'Instruments'*, is also used for analysis and testing purposes. The tool dynamically traces the CPU and memory usage, among other things. Many additional tools are available for device performance tracking however, without a doubt, Instrument was the most appropriate to utilised. This is because, since the tool is released by Apple for OS X and iOS code, it is thought that such a tool would provide a more accurate performance analysis than other alternative tools that are essentially generic. Furthermore, Instruments provides the ability to track one or more process over a period of time. This justifies the decision to select Instruments to analyse the performance.

6.2 Methodology

Originally, the plan was to test the performance of the application in order to determine the impacts that the application has on the device. However, in order to determine whether the application is more efficient than a similar application, the revised plan is to test the performance of the implemented application and a similar application so that the results can be compared.

The Cologne application, discussed in Section 3.6.7, is selected as the similar application to which the Explore Leeds application will be compared to. The reason to select this application is due to the reason that the Cologne application was implemented to solve the same problem.

In order to ensure that the performance and the resource usage testing is fair, the existing usage of the device will be equivalent when running both tests. For each test, the testing settings in Instrument will be configured such that the tool monitors the Explore Leeds process in the first test and the Cologne process in the second.

6.3. Result

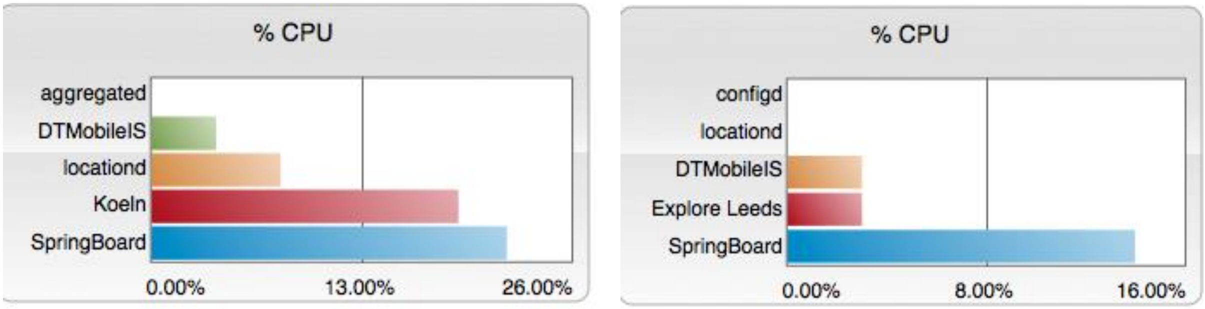


Figure 6.1. The bar charts show the percentage of the CPU usage for the Cologne ('Koeln') application (left) and the Explore Leeds application (right).

Figure 6.1 presents the result of the CPU usage testing. The bar chart to the left shows the result of the CPU usage for the Cologne (*'Koeln')* mobile application. The CPU usage for the application at the time was recorded at 20% of the overall CPU usage. This means that one application has occupied one fifth of the CPU availability which is thought to be unacceptable as other processes have less CPU availability to execute with. The bar chart to the right represents the result of the CPU usage for the Explore Leeds mobile application. The CPU usage for such application was recorded at approximately 3%. This result shows that there is a huge difference in the CPU usage between the Cologne and the Explore Leeds applications. Clearly, this shows that the Leeds application is likely to use less CPU time and hence in the long run save battery power. Furthermore, the use of less CPU usage will ensure such a device will run smoothly and not suffer from slow responses and crashes as applications have more CPU availability to use, in cases where the demand for CPU use from an specific application rises unexpectedly.

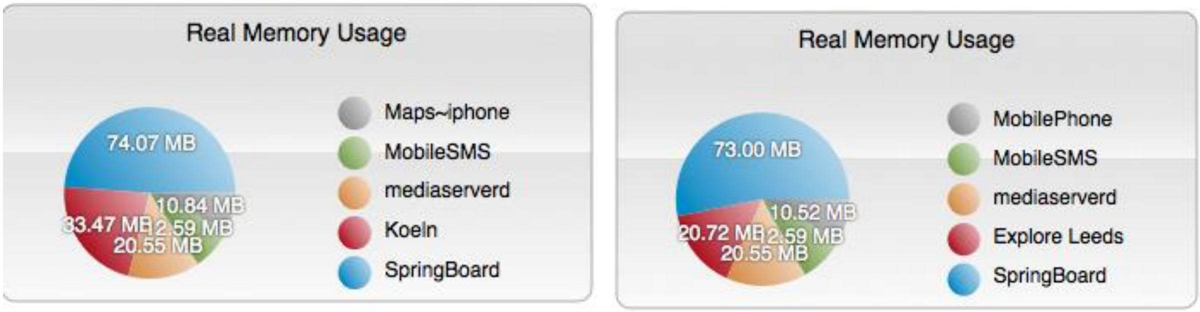


Figure 6.2. The bar charts show the memory usage for the Cologone ('Koeln') application (left) and the Explore Leeds application (right).

Figure 6.2 presents the result of the memory usage testing. The pie chart on the left shows the result of the memory usage for the Cologne (*'Koein')* mobile application. The memory usage recorded for the application was 33.47 megabytes. The pie chart to the right represents the result of the memory usage for the Leeds mobile application. The memory usage was 20.78 megabytes. Comparing the two results, the Explore Leeds application uses far less memory than the Cologne mobile application. As for the solution implementation, it means that the Explore Leeds application will run efficiently in regard to the memory usage. However, this comparison is not fair and invalid as the Cologne application has more shops and offers more features than the Explore Leeds application.

The Explore Leeds mobile application has proven to be much more efficient with regard to CPU usage hence provides more benefit to the device and its performance than the Cologne application. The reason for this might be due to the amount of data that many of the similar applications have to download during the execution of the application. This load can degrade the overall performance as it affects the CPU and memory power. As for the memory, both applications cannot be compared as mentioned earlier. Furthermore, the limited speed of the network connections put further impact on the performance of the overall application as download speed can have influence on the how efficient the application performs.

6.4. Conclusion

This chapter has discussed the overall testing of the Explore Leeds mobile application with regard to the CPU and memory usage. Furthermore, the result has been compared to the result obtained from the testing of the similar application. The comparisons show that the Explore Leeds mobile application performs more efficiently to the Cologne mobile application. The next stage is to test whether the application is usable by its intended users.

7. Evaluation

7.1. Introduction

This chapter will evaluate the implemented solution to the project. This will be accomplished by analysing the results from two different user evaluation methods: controlled observation and interview evaluation and a field study evaluation. Discussions will scrutinise such results and conclude the findings gained from the results.

7.2. User evaluation

User evaluation is considered to be very important to mobile application evaluation due to the user emphasised HCI design. Evaluations such as controlled observation help to highlight usability problems and evaluate whether the application satisfies the user requirements. Since user evaluation covers numerous areas, for the purpose of this project, the main focus of the user evaluation will be on the following criteria, as discussed in Section 1.8:

- User interface: Is the system usable? Can users figure out the layout and structure of the application? Are all the features apparent?
- Technology: Does the application work? Does it ever break or fail? Is it efficient?
- Context: When, where and how is the application used? Is the application suitable for the context of use?
- Usefulness: Does the user want to use such application? Do they find it useful?
- Improvement: How can the application be improved?

In order to evaluate the criteria above, two user evaluation methods were selected: controlled based observation and interview evaluation and field study evaluation.

7.3. Methodology

7.3.1. Controlled based observation and interview

'Controlled based observation and interview' is split into two fragments. The first fragment is the controlled based observation. The controlled based fragment comprises of users carrying out a predefined list of tasks (Appendix F1). The predefined list of tasks takes participants through a real life scenario that potential users are likely to experience while using the application. Furthermore, the list of tasks is believed to help users figure out the purpose of the application. Whilst performing the tasks, the participants were observed. The intention was to see how participants performed, in

terms of how well they executed the tasks. In order to observe the participants, a table was derived (Appendix F2). The table consists of four columns:*' task*', *'completion success', 'number of clicks', 'comments'.* The *'task'* column defines a number which corresponds to a specific task on the task list. The *'completion success'* column consists of either Y or N, a *'Y'* expresses that the task was executed by the participant successfully without any help and a *'N'* demonstrates that the assistance was required in order to perform the particular task. The *'number of clicks'* column defines a number, this number identifies the number of clicks performed by the participants, in order to successfully perform a particular task. Finally, the *'comment'* column comprises of relevant statements relating to problems the participants faced along with additional remarks that are thought to be relevant. Controlled based observations provide the ability to control the participants in regards to the tasks they perform and when such tasks are performed.

The second fragment is the interview. The interview fragment consists of interrogating participants in order to discuss their experience of using the application while performing the tasks. The interview questions (Appendix F3) were designed in a way to ensure that each criterion, mentioned in Section 7.2, were assessed. This ensured that individual participants' views on each criterion were obtained and therefore the participant's claims can be used in order to ascertain judgements about the application and most importantly, evaluate the application. The interview questions were both open and closed ones and hence the act of performing the interviews resulted in quantitative and qualitative responses. Participants were asked questions and responses were documented. The controlled based observation and interview were performed in the same setting. It was important to ensure that all participants performed the tasks in the same environment as different environments can impact the user experience hence have influence on the responses. Keeping consistency in the environment can ensure that the conclusions made from the responses actually reflect the experience of using the application.

7.3.1.1. Results

Due to the time limitation associated with the project, only 10 participants were selected. Thus, it was important to select the participants wisely in order to ensure that the evaluation consists of a representative for each target group.

Observation

The average statistical method was applied to the result, in order to analyse and interpret the quantitative sample obtained. The number of clicks sample was imported into a table upon which the average was calculated for two different purposes. The first purpose is to show the average number of clicks per task and is thought to demonstrate the efficiency of performing each task. This calculation can help to determine the tasks that are proving to be problematic to perform. The second purpose is to show the average number of clicks for inexperienced and experienced participants and is thought to establish the efficiency and the ease of use of the overall application. Overall, the decision to perform this calculation was to help response to the criteria, mentioned in Section 7.2, in regard to efficiency.

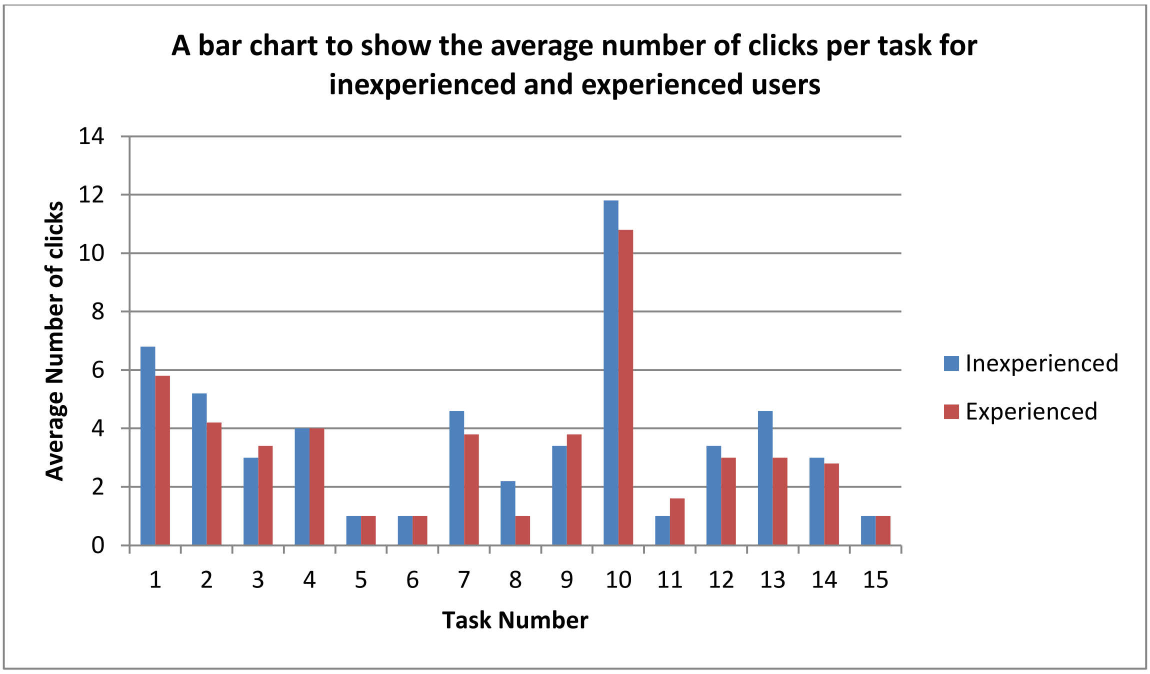


Figure 7.1. Comparison of the average number of clicks per task for inexperienced and experienced participants

The chart represents the average number of clicks per task for two types of participants: inexperienced and experience. Inexperienced participants are thought of users who have none or limited experience of using smartphones. A high average number of clicks imply that a task was difficult to carry out and a low average number of clicks imply that the task was simple to perform. Interpreting the graph in Figure 7.1, it can be clearly seen that task 10 was the most difficult task for both set of participants. During the observation, it was surprising to see that all of the participants used the *map* feature via the tab bar rather than the *'Show on Map´* feature presented on the cinema's detailed view under the *See* feature. The optimal solution for completing task 10 is four clicks, accomplished through the use of the '*Show on Map'* feature. The average number of clicks recorded was 11.89 and 10 for inexperienced and experienced participants respectively. This result suggests that the completion of task 10 wasn't straightforward and acquaints that the area requires improvement. The failure to recognise the ' *Show on Map'* feature is blamed on the fact that the participants were using the application for the first time hence further use would help to overcome this problem. Furthermore, both experienced and inexperienced participants suffered the same experience which implies that it is not the case of being an inexperience user but the case of failing to recognise what the feature does and again, further use of the application would assist recognition of this feature.

From the observation and the analysis of the result, one surprising finding emerged. Since experienced users of smartphone mobile application are able comprehend the navigation of an application more easily than inexperienced users, during the observation most experienced participants selected the back navigation button to return to the home screen as appose to selecting the '*Home'* tab bar button. This justifies the infrequent variation in task 11. Overall, the chart demonstrates that many of the tasks were relatively simple to carry out hence implies that the application is efficient and easy to use.

Interview

The quantifiable data obtained from questions one to four of the interview (Appendix F3), again were subject to average calculation for different purposes. The first purpose is to show the average rating for each question. Since each question correspond to one of the criteria, mentioned in Section 7.2, the result is thought to illustrate the extent to which each criteria is met. The calculations will help to show the level of satisfaction and domain knowledge held by the participants. In order to demonstrate the result, a graph was generated, Figure 7.2.

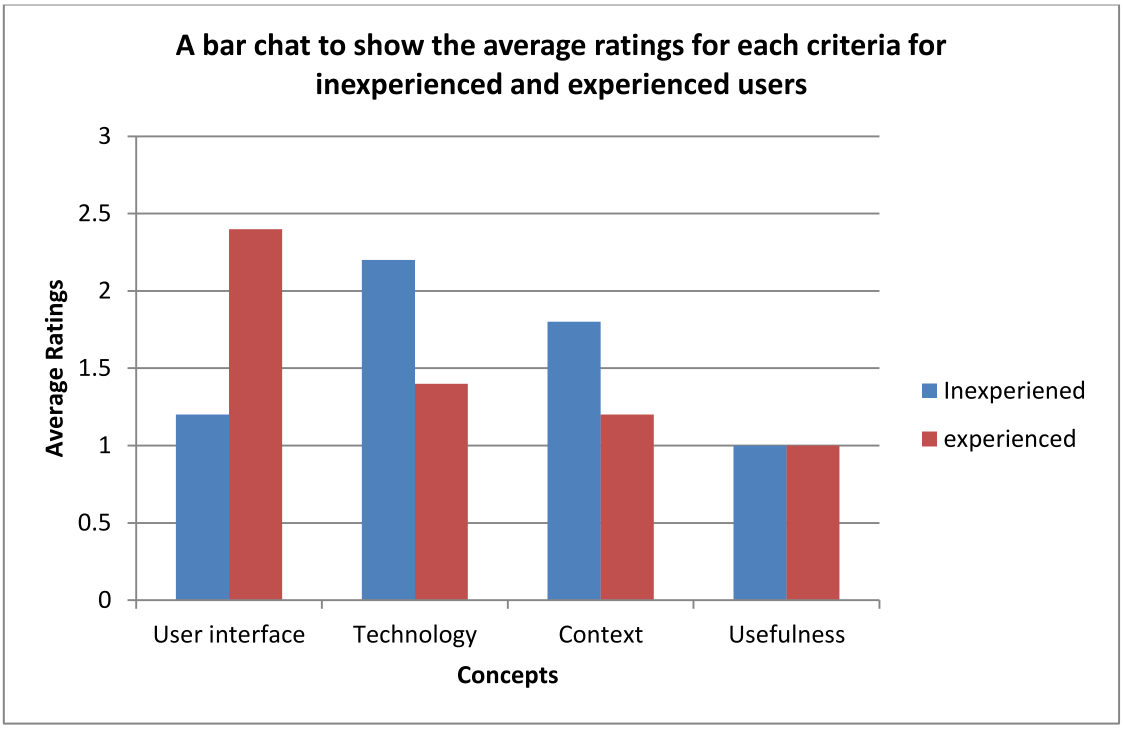


Figure 7.2. Comparison of the average rating for each criteria for inexperienced and experienced participants

The chart represents the average ratings per criteria for both type of participants: inexperienced and experienced. It is clearly evident that inexperienced participants were more satisfied with the aesthetic user interface compared to experienced participants. This result was expected as experienced participants are likely to have used numerous mobile applications in the past and it is likely that they have experienced an enhanced visual user interface. As for the technology criteria, the result is obtained wasn't surprising, with inexperienced participants rating a higher average than the experienced. This is understandable since experienced participants are already familiar with many concepts of a mobile application such as the navigation and hence are likely to be more efficient whilst performing the tasks. Both experienced and inexperienced participants agreed that application was useful, scoring an average of one. For the project, this justifies that the application is better than the existing systems and without a doubt there was a need for such application.

The later questions gave the participants the opportunity to provide their feedback on the features that they found useful, issues they faced while performing the tasks and how the application could be improved. Overall, the feedback obtained was positive. Majority of the participants found the directions feature as the most useful feature within the application and it is thought to be the one that makes the application very useful to the users. Due to the fact that Leeds City Centre tolerates a no free parking zone, many participants also found the parking feature beneficial.

On the other hand, participants forwarded their difficulties in performing task 10. As mentioned earlier, this result was also apparent in the observation where the number of clicks averaged 11 clicks, which is considerable higher than the optimal solution of 4 clicks.

Finally, as for the changes that could be made to improve the application, majority of the participants felt that providing the ability to search the application for something of their interest would help. Upon finding of the opinions, it is thought that the search box could actually improve the usability and the efficiency of the application as it can be treated as an alternative way to find information as appose to navigating through the application. This improvement would be very useful in situations where the user rushes and it is inappropriate to navigate through the application. With the increased popularity of the use of social networks, some participants felt that the application should integrate social networking. Carrying out this improvement is bound to affect the performance of the application with regard to the CPU, memory and network usage. Finally, comments from nearly all of the participants expressed that the application is limited in terms of the number of places to eat, eat and see features. These comments are understandable however, the limited number of places was down to the time limitation associated with the project.

7.3.2. Field Study evaluation

The field study evaluation was applied as part of the evaluation for the implemented mobile application. Field studies are completed in the context in which the application is intended to operate. The main focus is to illustrate the experience of using the application in its proposed surroundings.

Certain variables affect the outcome of the evaluation method, which are related to the environment and the context where the application is used. Characteristics of the surroundings such as lighting level, weather and the influence from walking can have impact on the usability of the application [37]. Some may argue that, although such variables are problematic to control, it is possible to manage this issue by eradicating the variables upon the analysis and conclusion of the results. However, it is arguable that this procedure can induce impractical outcomes hence the result would not reflect the real world practice.

To perform the field study evaluation, two participants were selected. It was important to select the most appropriate participants in order to ensure that both are representative of the target group and to acquire data that will allow the assessment of the influences that the application has upon new and experienced visitors of the City Centre. Field study evaluations are known for having a drawback in that it is difficult to recruit the correct participants [41], however this has not been the case during the evaluation of the application. The characteristics of the participants are described as follows:

- ***Participant 1***: an inexperienced user of smartphones that has never visited Leeds City Centre. It is thought that this participant would provide invaluable feedback for the application.
- ***Participant 2***: an experienced user of smartphones. A first year student who has been adapting to life in Leeds and has little knowledge of the city centre.

Participants were asked to visit the city centre on a day and time of their convenience. Although, this instruction is certain to have impact on the outcome of the evaluation, it was not viable to provide a fixed day and time since the outcome would not reflect the real world use. Participants were asked to use a mixture of mobile search and the application upon the need for location or local based information such as phone number, directions or the locality of a particular amenity. The mobile search approach involves the use of a smartphone's web browser. This approach differs to the experience on desktop computers [38], as mentioned in Section 1.2, due to the limitations of the mobile devices [24], as mentioned in Section 3.3. The key aim here was to assess whether the application actually makes a difference, which justifies the decision to ask the participants to use a combination of the approaches. After the visits, the participants were asked a series of closed and open questions (Appendix F4) in order discuss their experience. The questions were written to so that each criteria, mentioned in Section 7.2, were assessed in regard to both, the experience of using the local mobile approach and the mobile application approach.

Upon selecting the most appropriate evaluation methods, the expert review method was considered [37]. This method involves receiving valuable feedback from experts, whom have knowledge of such area. However, the decision to carry out the field study evaluation was certain as it is better and more practical to involve real users since they would provide realistic feedback as opposite to expert reviews [37]. Hence it is thought that the feedback provided will help to conclude whether the application essentially works in practise.

7.3.2.1. Results

The result obtained from the interview consisted of quantitative and qualitative data. As it can be comprehended from Appendix F4, each criteria, mentioned in Section 7.2, is judged for local mobile search experience and the mobile application experience. Participants were asked to rate each concept twice, once for each approach. The aim is to see whether the application makes a difference and the result would help to evaluate the whether the mobile application is better than the existing approach. To show comparison of the results obtained from participant 1 and participant 2, a bar chart was generated, Figure 7.3.

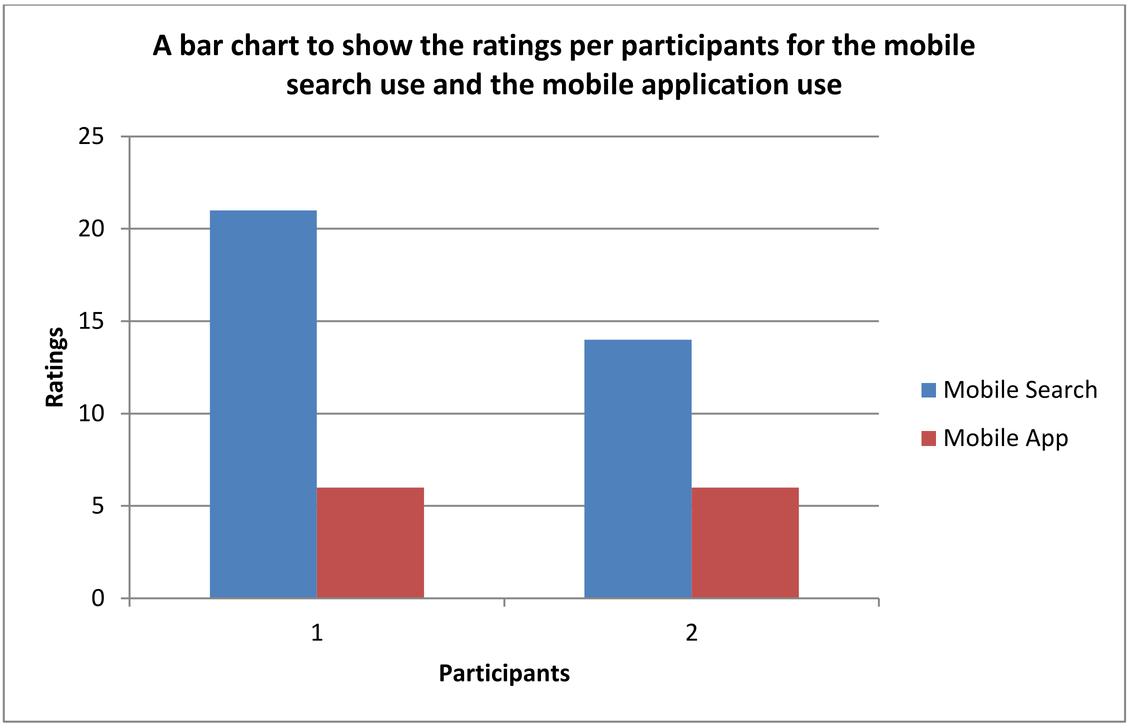


Figure 7.3. The rating between the local mobile experience and the mobile application experience of both participants

The chart shows the ratings per participant for each of the two approaches: the mobile search approach and the mobile application approach. A high rating implies unpleasant and frustrating experience and a low rating signifies satisfying and helpful experience. It is evident that both participants suffered from the use of the mobile search as the ratings demonstrate, with the inexperienced participant experiencing the most dissatisfied at a rating of 21. Although, participant two is classified as an experienced user of smartphones, surprisingly the experience of the mobile search was higher than expected at a rating of 14. On the other hand, the ratings for the mobile application expressed a contradictory experience. Both participants rated the experience at 6. These results were unexpected and have clearly demonstrated that the mobile application is proving to be better than the existing approach.

The result was interpreted in alternative way so that it allowed the comparison of the local mobile search approach with the mobile application approach with regard to each of the criteria, mentioned in Section 7.2. The chart demonstrates the result, Figure 7.4.

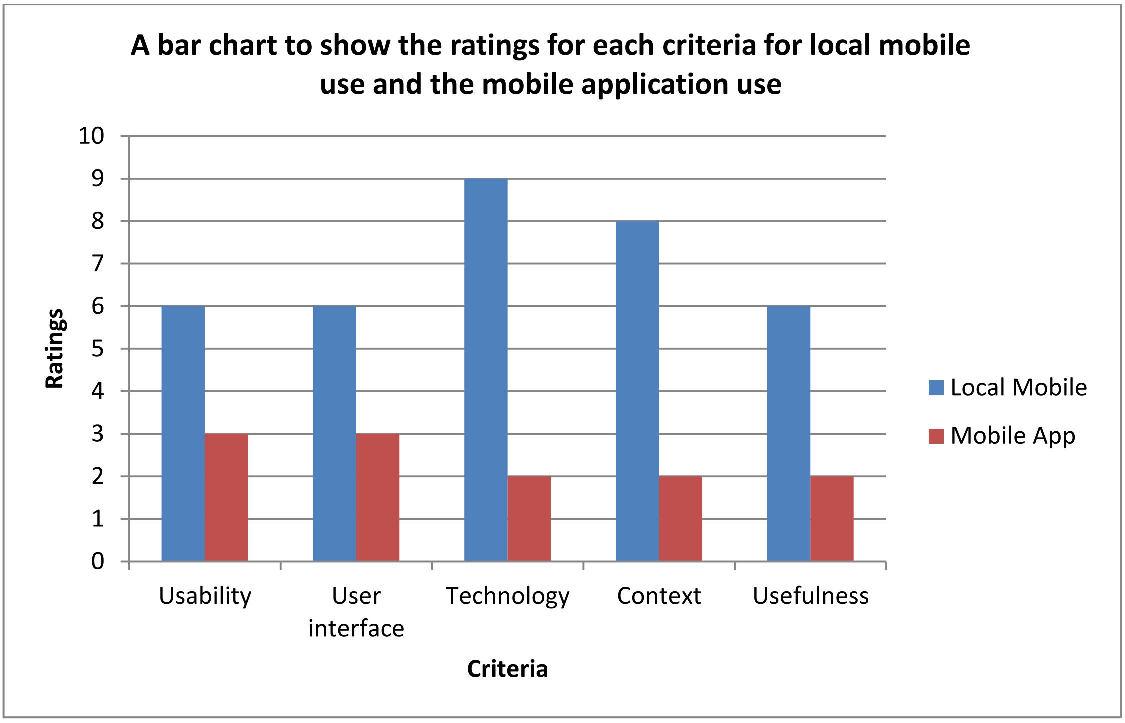


Figure 7.4. The ratings for each criterion for local mobile search use and mobile application use

The chart represents the ratings for each criterion for the local mobile search approach and the mobile application approach. As mentioned earlier, low ratings demonstrate a good experience. As illustrated, the mobile application scored a much better rating than the local mobile search approach in each of the criteria. The result proves that the participants regarded the mobile application as easy to use and useful. As for the technology criteria, the local mobile search relies heavily on internet access and with the limited internet speed available, at the time of writing, one can imagine the experience to be time consuming. This justifies the high ratings scored. Furthermore the context criterion, the local mobile search suffered vastly in this area possibly due to the fact that the approach requires excessive amount of user input and such activity is not practical especially within a busy surrounding. The alternative interpretation of the result strengthens the proof that the mobile application is better than the existing approach.

The additional questions asked were an opportunity was obtain valuable feedback on the mobile application. The main focus here was on the features participants found useful, any issues they faced and potential improvements that could enhanced the application. In terms of the useful features, both participants found the directions and the *Map* feature very useful. These views were expected since it was known that the two participants are not very familiar with the surroundings. Participant one forwarded her concern with the closeness of the shop, eat and see button on the home screen. She thought that the buttons are too adjacent together and while using the application, the also participant had to be aware of the situation around her such as looking out for vehicles and individuals in the surroundings. This design error is thought to be frustrating.

Finally, as for the enhancements that could be made to improve the application, both participants thought that the implementation of a search box would be very useful. Both participants felt that the search box in the local mobile search provided more freedom as to what they wanted to search.

7.4. Justification

Based on the findings obtained from applying the evaluation methodologies, it is fair to judge that the mobile application is a success among its intended users. Although some issues exist within the application, the majority of the features have proved to be very useful. Comparing the experiences of the local mobile search and the mobile application justifies the reason why the application is a success and is a good solution to the problem. Furthermore, the application is usable by both inexperienced and experienced users which demonstrate that the HCI issues have been addressed appropriately.

7.5. Conclusion

This chapter has evaluated the solution with regard to the users. Discussions upon the methodologies used to perform the user evaluation and the results have been made. Finally, the chapter justifies whether the mobile application is a success among its intended users.

8. Conclusion

8.1. Introduction

This chapter concludes the project by discussing the extent to which the project as a whole is considered successful. The chapter discusses whether the project has met its aim, objectives and minimum requirements.

8.2. Aim, objectives and minimum requirements

8.2.1. Aim

The aim of the project was to implement a Leeds City Centre guide that offers a context sensitive interface for visitors of the City Centre. This aim has been achieved by implementing an iOS mobile application which can be executed on iOS based devices.

8.2.2. Objectives

- *Research and collect relevant information that are essential in the process of addressing the problem*
  During the research performed prior to the production of the designs and the implementation of the solution, various contexts concerning the interest area of the project was scrutinised, as discussed in Chapter 3. Areas including Human Computer Interaction, mobile phone computing, existing systems and Global Positioning Systems were investigated.
- *Select an appropriate mobile platform*
  During the research conducted various mobile platform were considered. However, in order to ensure that the application is available to the majority of people, two most popular platforms were considered for development of the mobile application. Each had its own advantages however, it was important to consider how the HCI issues would be addressed. For this constraint, the iOS mobile platform was chosen due its strict design and implementation guidelines which are thought to lead developer's to the creation of a much better quality application and to help overcome many of the HCI issues associated with mobile phone interfaces.

- *Understand and familiarise with the programming construct and design patterns of the chosen platform*

Since having no knowledge of the chosen iOS mobile platform prior to the implementation, it was important to understand and familiarise with the platform's programming constructs and design approaches that allow the development of a mobile application. Architectural design patterns and techniques, as discussed in Section 5.2, were explored and online tutorials were completed which led to the development of a simple '*Hello World!'* application. The decision to consider this activity as a project objective and the action of meeting this objective has helped significantly during the implementation of the mobile application.

- *Select an appropriate software development methodology*

  The project management section of this project involved researching various software methodologies, followed by the selection of the methodology that was deemed to be the most appropriate. As many of the methodologies produce a solution toward the end of the development cycle, the methodologies were not appropriate for the development of mobile applications. Also, since the chosen mobile platform was relatively unknown to the developer, it was important to test as early as possible in order to ensure errors are fixed earlier in the development stage where they are smaller and efficient to fix. These reasons justify the decision to select the incremental methodology which was adjusted to consider the HCI interface design process, as discussed in Section 3.4.1 and Section 4.6.
- *Generate potential designs for mobile application*

  In order to comply with the HCI interface design process, it was important to generate potential designs for the proposed application. The designs were subject to user testing upon which the result was used to materialise the final designs, as discussed in Chapter 4. The decision to produce the designs made the implementation stage a lot simpler as the backbone of the application was already planned out, as discussed in Section 5.2.
- *Implement a mobile application that offers services accessible to the users*

  This objective has been met with the production of the Explore Leeds mobile application. In order to meet such objective, the overall process involved the establishment of the designs and the actual implementation itself, which is detailed in Chapter 4 and Chapter 5 respectively.
- *Test and evaluate the application with the main focus on the usability, accuracy and the performance*

  The final solution has been tested throughout the process with the focus been on the usability, accuracy and the performance. The testing of the accuracy within the application was performed after each iterations, as discussed in Chapter 5. The performance of the application with regard to the CPU and memory usage has also been performed and moreover compared to the performance of a similar application, as discussed in Section 6.3. More importantly, the usability of the application has been assessed with the use of two software evaluation methodologies, as discussed in Section 7.3.

- *Evaluate the application against the project aim and minimum requirements in order to comprehend whether the problem has been addressed*
  The evaluation of the application against the project aim and minimum requirements has been achieved, as discussed in Section 8.2.1 and Section 8.2.3 respectively.

8.2.3. Minimum requirements

- *A working mobile application that displays a map of Leeds City Centre*
  This requirement has been completed with the implementation of the Explore Leeds mobile application which displays a map of Leeds City Centre under the *Map* section of the tab bar. The use of Google Maps has helped to achieve this requirement.
- *Provide indication of the user's position*
  Following the implementation of the Leeds City Centre map requirement, this requirement was met by adding a pin onto the map which indicate the users of their current position.
- *Display a single predefined shop on the map*
  The implementation of a red pin on the map of Leeds City Centre ensured that this requirement was met. Furthermore, the implementation of each amenity's detailed view gave the opportunity to implement a '*Show on Map'* button. This button shows the location of the corresponding amenity on the map.
- *To provide visual and text based directions from the user's current position to the predefined shop*
  This requirement has been met through the implementation of the *directions* button on each shop's detailed view. The directions are provided textually and visually with the use of the Google Maps application.

8.3. Project Management

8.3.1. Schedule

Amazingly, the schedule produced at the start of the project was followed throughout. Initially, it was predicated that the original schedule would be subject to changes due to external influences. Issues did arise as predicated, discussed in Chapter 5, but it didn't affect the schedule considerably, mainly due to the fact that any time lost was made up on the following day. Originally, 6 hours a day was a minimum target set. However, as the project advanced into its latter stages especially during the implementation, as a precaution, it was decided to increase the minimum target to 8 hours. This decision has demonstrates to be significant as the influence from the increased hours were clearly visible and it is absolutely certain that things would have gone differently if it was for this decision.

8.3.2. Methodology

The custom incremental methodology formulated in Section 2.2, Section 3.4.2 and Section 4.6 was followed throughout the project. The incremental methodology was customised to integrate the process of interface design, as discussed in Section 3.4.2. The first stage of the methodology was the designs, followed by the HCI design process, as documented in Chapter 4. The subsequent stage was the actual implementation. To get the most out of this methodology it was important to plan out the implementation stages prior to the progression to this stage, as discussed in Section 4.6. During the implementation, a major change affected the planning of the iterations, as discussed in Section 5.3. The series of builds have been documented in Sections 5.3, 5.4 and 5.5. Testing was performed after each iteration as documented in Sections 5.3, 5.4 and 5.4. Finally, the subsequent stages were the overall testing, discussed in Chapter 6 and the evaluation, documented in Chapter 7.

The selection and customisation of this methodology contributed significantly to the success of the overall project. Planning of the implementation and the process of each stage were the main aspects which are thought to play an immense role to the success of this project. The planning of the implementation gave a clear idea of what is to be developed and when and without the planning of the iterations, one would have been lost, wondering what to do next and hence lose incredible amount of time. It is fair to say that the selection of the methodology helped to manage the time constraints associated with project as it allowed efficient time management and planning.

And as for the above reasons, the decision to select this methodology, without a doubt, was appropriate and the decision played a very significant role to the success of the project.

8.4. Further Extensions

The extensions to the project were to implement features that provide shopping, eating and seeing amenities, provide up to date train and bus travel information and provide information services such as the weather information. Due to the extensive planning of the project, it was possible to implement all of the extensions discussed in Section 1.6. The shopping, eating and seeing amenities were implemented within iteration one, as discussed in Section 5.3. The train and bus travel information was implemented in iteration two and the weather was implemented in iteration three as discussed in Section 5.7. Furthermore, a picture carousel was also implemented on the home screen as part of the further extensions. This feature, documented in Section 5.3, was unplanned.

8.5. Future Extensions

Due to the time limitations associated with the project, some aspects of the application were not feasible to implement. In terms of the existing features, the car parking features could be extended to a extent that users are able to sort the list out according to distance, opening periods and prices. Some may argue that the list could be ordered according to the number of car parking spaces available. However, this information is not yet available and cannot be provided due to third party interferences. From the result of the user evaluation, the option of a search box was a popular suggestion among participants and one that is thought to build the gap between local mobile search and the Explore Leeds mobile application.

In terms of additional features, it is fair to say that the current mobile application is showing signs of potential. Therefore, a proposal for future extension is to implement the complete application by adding all the amenities within the city centre, covering all the shops, dining places and places to see. This may seem simple however, adding more amenities to the application is not feasible due to the complexity of the application development process. A solution to this problem is to introduce the process of dynamically presenting relevant information on to the screen. In order to do this, every aspect of the application has to be centralised and the information regarding each amenity has to be stored into a centralised database. With the use of PHP and SQL, the information can then be generated dynamically depending on the information requested by the users. One must consider the efficiency while implementing such extension. Furthermore, as the application is then centralised, one can extend to this by providing the facilities for users to add their own amenity which are thought to be missing in the application. The centralisation of the application will ensure that the introduction of a user's amenity will update the application so that others users can benefit from the changes made, in this case the added amenity.

8.6. Conclusion

This chapter has shown that the project aim and objectives have been met and that the project has exceeded the minimum requirements. Discussion on the extent to which the chosen methodology was successful was made and finally, future works on the application were suggested. Based on the achievement of the aim and the objectives and the surpassing of the minimum requirements, this project as a whole can only be pondered to have been a success.

References

[1] Steve Love (2005) Understanding mobile human-computer interaction, 1st edn., Amsterdam: Elsevier Butterworth-Heinmann.

[2] Bob Hughes, Mike Cotterell (2009) Software Project Management, 5th edn., : McGraw-Hill Education.

[3] Anabela Sarmento (2005) Issues of Human Computer Interaction, : IRM Press.

[4] Yvonne Rogers (2011) Interaction design: beyond human-computer interaction, 3rd edn., Hoboken, N.J: Wiley.

[5] Ian Sommerville (2006) Software Engineering, 8th edn., : Addison Wesley.

[6] Paul Malish (2012) Leeds City Guide, Available at: http://leeds-city-guide.com/ (Accessed: 4th February 2012).

[7] Leeds, Live it, Love it (2013) Leeds, Live it, Love it &gt; Visitors &gt; Home, Available at: http://www.leedsliveitloveit.com/visitors/home (Accessed: 4th February 2012).

[8] Valentino Lee (2004) Mobile applications: architecture, design, and development, 1st edn., Upper Saddle River, N.J: Pearson Education.

[9] Jaime Teevan, Amy Karlson, Shahriyar Amini, A.J. Bernheim Brush, John Krumm / Microsoft Research, Carnegie Mellon University (2011) Understanding the Importance of Location, Time, and People in Mobile Local Search Behavior, Stockholm, Sweden: ACM.

[10] Kuo-Ying Huang (2009) 'Challenges in Human-Computer Interaction Design for Mobile Devices', Proceedings of the World Congress on Engineering and Computer Science, 1(), pp. 52-57.

[11] Matt Brian (2011) Mobile Apps: A look at what makes an app popular - TNW Mobile, Available at: http://thenextweb.com/mobile/2011/07/16/mobile-apps-a-look-at-what-makes-a-good-app-great/ (Accessed: 30th January 2013).

[12] Android (2013) Android, the world's most popular mobile platform , Available at: http://developer.android.com/about/index.html (Accessed: 30th January 2013).

[13] International Data Corporation (2013) Android Marks Fourth Anniversary Since Launch with 75.0% Market Share in Third Quarter, According to IDC - prUS23771812, Available at:

http://www.idc.com/getdoc.jsp?containerId=prUS23771812#.USZUUVu1nTe (Accessed: 30th January 2013).

[14] Engineers Garage (2013) Android, Available at: http://www.engineersgarage.com/articles/what-is-android-introduction (Accessed: 30th January 2013).

[15] Engineers Garage (2013) Android, Available at: http://www.engineersgarage.com/articles/what-is-android-introduction?page=2 (Accessed: 30th January 2013).

[16] TechCrunch (2013) Apple Has Sold Over 400 Million iOS Devices: 9.5% Growth Since March | TechCrunch, Available at: http://techcrunch.com/2012/09/12/apple-has-sold-over-400-million-ios-devices-9-5-growth-since-march/ (Accessed: 30th January 2013).

[17] Damien McFerran (2012) Do Android and iOS present equal opportunities for developers? Damien McFerran investigates, : Know Your Mobile.

[18] NetMarketShare (2013) *Mobile market share,* Available at: *http://www.netmarketshare.com/mobile-market share?qprid=8&qptimeframe=M&qpsp=167&qpmr=100&qpdt=1&qpcustomd=1&qpct=3* (Accessed: 30th January 2013).

[19] Cologne-Tourism (2013) *Cologne Tourism: Cologne Guide - The official guide to Cologne from KölnTourismus - APP,* Available at: *http://www.cologne-tourism.com/service/koeln-guide.html* (Accessed: 30th January 2013).

[20] Diana Cooksey (2011) *Understanding the Global Positioning System,* Available at: *http://www.montana.edu/gps/understd.html#What_is_GPS* (Accessed: 6th February 2013).

[21] Apple (2013) *iOS Human Interface Guidelines,* Available at: *https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/AppDesign/AppDesign.html#//apple_ref/doc/uid/TP40006556-CH19-SW1* (Accessed: 9th February 2013).

[22] William R. Teague, Michael Garner, Michael B. Daniels, H. Don Scott (1999) *GPS/map Position Coordinate Issues: GPS Position Accuracy*, University of Arkansas, U.S. Department of Agriculture, and county governments cooperating: Cooperative Extension Service.

[23] Faridan J. 2011. A Guide to Selecting Software Development Methodologies, March 2011

[24] Balasubramanyam Seshan (2013) *Top 10 Smartphones with Large Screen Size,* Available at: http://www.ibtimes.com/top-10-smartphones-large-screen-size-374664 (Accessed: 28th February 2013).

[25] Ed Arnold (2013) *Tablet and Smartphone Growth Flies Past Traditional Computer Market,* Available at: http://www.bizjournals.com/memphis/news/2013/02/26/tablet-and-smartphone-growth-flies.html (Accessed: 27th February 2013).

[26] Don Reisinger (2013) 'Apple App Store hits 40 billion downloads; 20 billion in 2012, alone', *CNET,* 7th January.

[27] Peep Laja (2012) *First Impressions Matter: The Importance of Great Visual Design*: ConversionXL.

[28] Alexandre N. Tuch, Eva E. Presslaber, Markus St□cklin, Klaus Opwis, Javier A. Bargas-Avila (2012) *The Role of Visual Complexity and Prototypicality Regarding First Impression of Websites: Working Towards Understanding Aesthetic Judgments*, Zurich, Switzerland: University of Basel,Center for Cognitive Psychology and Methodology.

[29] Stephen Shankland (2012) 'Google: 500 million Android Devices Activated', *CNET,* 12th September.

[30] Global Mobile Academy (2013) *Understand iOS Platform ,* Available at: http://globalmobileacademy.com/understanding-ios-platform/ (Accessed: 26th February 2013).

[31] Fred Donovan (2012) *Android Surges 75 Percent Mobile OS Market Share Says IDC,* Available at: http://www.fiercemobileit.com/story/android-surges-75-percent-mobile-os-market-share-says-idc/2012-11-02 (Accessed: 26th February 2013).

[32] Paul Sloan (2012) 'Apple by the numbers: 84M iPads, 400M iOS devices, 350M iPods sold', *CNET,* 12th September.

[33] Sam Oliver (2012) *Apple on pace to sell 1 billion iOS devices by 2015,* Available at: http://appleinsider.com/articles/12/09/17/apple_on_pace_to_sell_1_billion_ios_devices_by_2015 (Accessed: 26th February 2013).

[34] Apple Inc. (2013) *Publishing an App in the App Store,* Available at: http://developer.apple.com/library/ios/#documentation/General/Conceptual/ApplicationDevelopmentOverview/DeliverYourApppontheAppStore/DeliverYourAppontheAppStore.html (Accessed: 26th February 2013).

[35] Apple Inc. (2013) *Deira City Centre Official,* Available at: *https://itunes.apple.com/gb/app/deira-city-centre-official/id564767582?mt=8* (Accessed: 26th February 2013).

[36] Suniwe (2008) *Project Approach,* Available at: *http://www.projects.staffs.ac.uk/suniwe/project/projectapproach.html#Why+an+Iterative+Approach%3F* (Accessed: 26th February 2013).

[37] Joy Goodman, Stephen Brewster and Philip Gray (2004) Using Field Experiments to Evaluate Mobile Guides, Glasgow, UK: N/A.

[38] Karen Church and Barry Smyth (2009) Understanding the Intent Behind Mobile Information Needs, Sanibel Island, Florida, USA: Association for Computing Machinery.

[39] Rachel King (2012) 'Google Maps for iOS surpassed 10M downloads in less than 48 hours', Between the Lines, (), pp. [Online]. Available at: http://www.zdnet.com/google-maps-for-ios-surpassed-10m-downloads-in-less-than-48-hours-7000008907/ (Accessed: 19th April 2013).

[40] Apple (2013) Custom Icon and Image Creation Guidelines, Available at: http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/IconsImages/IconsImages.html (Accessed: 4th April 2013).

[41] Ian Brace (2008) Questionnaire Design: How to Plan, Structure and Write Survey Material for Effective Market Research , 2nd edn., Great Britain : Market Research in Practice.

[42] Mark Well (2013) 10.6. Calculating Distance Between Two Sets of Coordinates, Available at: 10.6. Calculating Distance between Two Sets of Coordinates (Accessed: 3rd April 2013).

[43] B. Green (2010) GPS Waypoint Navigation with Fixed-Wing Platform , Available at: http://dasl.mem.drexel.edu/alumni/bGreen/www.pages.drexel.edu/_weg22/gpsMagpie/gpsFixedWing.html (Accessed: 3rd April 2013).

[44] Anand Lal Shimpi & Brian Klug (2012) 'iPhone 5 Memory Size and Speed Revealed', AnandTech, (), pp. [Online]. Available at: http://www.anandtech.com/show/6297/iphone-5-memory-size-and-speed-revealed-1gb-lpddr21066 (Accessed: 19th April 2013).

[45] Hoshi Takanor (2013) XmlPullParser , Available at: https://github.com/hoshi-takanori/XmlPullParser (Accessed: 30th March 2013).

[46] Nick Lockwood (2013) iCarousel , Available at: https://github.com/nicklockwood/iCarousel (Accessed: 14th March 2013).

[47] Mark Keil and Erran Carmel (1995) 'Customer-Developer Links in Software Development', Communications of the ACM, 38(5), pp. 33-44.

[48] R Subrayaman, F.L. Weisstein, M.S Krishnan (2010) 'User Participation in Software Development Projects', Communications of the ACM, 53(3), pp. 137-141.

[49] Sari Kujala and Martti Mantyla (2000) 'Is User Involvement Harmful or Useful in the Early Stages of Product Development?', CHI 2000 Extended Abstracts, ACM Press, (), pp. 285 - 286.

[50] Android (2013) Supporting Different Screens, Available at: http://developer.android.com/training/basics/supporting-devices/screens.html (Accessed: 14th February 2013).

[51] CompareBatteryLife (2013) iPhone 5 Battery Life Review, Available at: http://comparebatterylife.com/ (Accessed: 9th February 2013).

[52] Desmond Lee (2013) What is GPS? Available at: http://www8.garmin.com/aboutGPS/ (Accessed: 26th February 2013).

Appendix A: Personal Reflection

This project has been, without a doubt, the hardest and challenging task undertaken.

My desire to learn and try new things everyday was the main influence behind the reason to commence this project. Having read advices from previous FYP reports, many students recommended embarking upon an area that is of interests rather than something new. My weakness of finding it difficult to say no resulted in overlooking such advice and embarking on something that I do not have knowledge of. This decision demonstrates to be a success as the process has allowed me to finally discover an area of interest which I intend to pursue further.

Initially, I was spending 5 days a week, 6 hours each day, on the project. However during the implementation, as a precaution, this increased to 6 days a week, 8 hours per day. The schedule created at the start of the project wasn't, surprisingly, affected by any delays. Along with the Gantt chart, I also created a table view of the schedule which defined specific predicted start and finish dates for each task. I believe such table view schedule was extremely helpful and one that helped me to manage time efficiently. Each predicated finish date was treated as a deadline, which gave more motivation to complete each task on time.

The project itself consists of numerous files; research papers, write up reports, images and programming files. To ensure back up of data, I constantly backed my work up on three different devices; the DEC10 machines, a USB and a laptop. This decision was valuable as I occasionally left my USB at home, in which case, the back up on the DEC10 machines were used.

I consider the implementation of the solution as an aspect that I truly enjoyed. I relished the challenges that I faced during the implementation, which required my problem solving skills, something that I enjoy putting into practise. I believe that the success of the planning for both, the project and the implementation, were the aspect of the project that went really well and contributed significant to the completion of the project. I consider the planning as a guide, leading me forward, when I was in doubt about what is to be done next. Without the planning, I would have lost valuable time and left confused.

Initially, I started using my personal ‘Project Log’ under the ‘Research Project’ module on the VLE. I posted one post per day; each outlining ‘what was done today?’, ‘what is to be done tomorrow?’ and any problems that I faced on a particular day. This was very helpful in tracking my progress and helped me to plan out the next day to ensure time was spent efficiently. However; due to the workload required for the implementation stage, I forgot about the blog and didn't keep up with the process which I am sure would have been very helpful during my write up.

I think that some aspect of my user evaluation went really bad. I believe that the field study undertaken was planned incorrectly and the decision to use two participants was illogical. Although, some results helped to conclude the solution, I believe using more participants would have helped to gain a more valid conclusion.

Lastly, the project heavily relied on the use of the Apple Mac machines in DEC10. These machines required compulsory updates in order to start the implementation. Due to the external issues associated with the School of Computing Support and the University's finance department, the updates were taking longer than expected. Luckily, I had a back up plan which was deployed.

This project has taught me some invaluable skills and knowledge. I have learnt the process of evaluating research literature, such as research papers which has enhanced my analytical skills. Furthermore, the project has enhanced my organisation and time keeping skills. Communicating with various people during my project has helped to improve my communication skills along with the write up of the report. All which I am sure will help me during my career and personal life.

I suggest the following recommendations for future final year project students:

- Select a project that is new and of interest to you. Do not worry about the complexity of the project. The proud feeling of producing a solution that you thought was impossible at the start of the project is a feeling you want to grasp.
- Throughout the project keep constant track of your progress, with the use of a blog. This will be extreme helpful during your write up.
- If you require permission to carry out a particular task or if the project relies on support from others, get in touch with them as soon as possible. Keep in touch with them, notifying them of your progress. Most important of all, have a back up plan.
- Before starting your project, make a list of the all the tasks that need to be done. Create an initial schedule. Assign realistic predicated start and finish dates and treat these as you treat coursework deadlines. This will help to motivate you and keep you going.
- Plan your implementation and be prepared for unexpected delays and errors.
- Keep a back up of every single file associated with the project on a number of devices.
- The write up is definitely a daunting experience. If you get into a situation where you don't know what to write, move on and write what you can. You can always come back and add more to where you left.

Appendix B: Resources and external materials

- The parsing of the XML data obtained from the weather feed was based on code obtained from: https://github.com/hoshi-takanori/XmlPullParser [45]. This code provided the mean to parse the XML data and permission for use has been granted.
- The Carousel Libaray used for the picture carousel on the home screen was based on code obtained from: https://github.com/nicklockwood/iCarousel [46]. This helped to implement the picture carousel on the home screen.

Appendix C: Ethical Issues

This section describes the ethical issues that were encountered during the project. It also describes how the issues were dealt with.

The user based assessments on the initial designs involved asking participants to perform a given list of tasks which led to questions relating to the usability of the prototype system. Before each session, participants were notified of the purpose of the assessment. They were provided with an '*Information Sheet'* which informed them what would happen if they took part, what type of data is sought from them and how the data would be used. If they wished to take part, before commencing each session, they were given a consent form to sign. They were also notified that they are free to stop during any time of the assessment. The agreement of the consent form confirmed that they agree to take part and agree that their data could be used for the project and for future research. To ensure confidentiality, the responses were presented as anonymous and the participants' names have not been disclosed.

The user evaluation consisted of 12 participants. The evaluation involved performing a given list of task on a mobile application. Two of the participants had to use the application in its intended context of use. Before the evaluation, the participants were informed of the purpose of the evaluation and how the data would be used. They were also notified that they may stop for a break or withdraw from the entire test at any point. To ensure anonymity, the results of the evaluation have been presented as anonymous and the participants' personal details have not been disclosed.

Appendix D: User Based Assessments

Introduction

The purpose of this activity is to gain feedback on the initial designs for a proposed mobile application. You are asked to perform the tasks below using a paper based prototype. Once you have carried out the tasks, please comment on the usability, utility and learnability. Comments on the good and bad aspects of the designs and improvements that could be made to the designs are appreciated. The responses and feedback gained from this help to discover problems that exist in the prototypes. Appropriate improvements to the designs will also be made from the responses.

All data collected from this activity will be confidential and individuals can be kept anonymous, if required. You may withdraw from the entire test at any point.

Task list

1. You are presented with the Home screen. Access the shop section, select one of the fashion shops
2. Go back to the Home screen
3. You're a bit lost, find out where you are on the map
4. Return to the home screen
5. You want to park your car, find a car park
6. Return to the home screen
7. You want something to eat. Ring 'Akbars' to book a table.
8. Go back to the home screen
9. You need to pick up a friend from Leeds Train Station, find directions to the station
10. Return to the home screen.

Appendix E: Iterative Functionality Testing Sample

Procedure	Expected Result	Pass / Fail (P/F)	Actual results / Comments
Shop button pressed	List of category of shops should appear – shop view controller	P	List of category presented – No problems
Fashion row selected	List of fashion shops should appear – fashion view controller	P	List of fashion shops presented – No problems
Aspecto row selected	Aspecto's detailed view should appear showing shop description, address, opening times and four buttons	F	Aspecto detailed view presented however, the scroll view doesn't show 2 of the buttons. Bug fixed by changing the content size of the scroll view. Re-tested and works perfectly fine.
Visit website button pressed	App should direct users to safari and load up http://m.aspecto	P	Aspecto's website presented in safari – no bugs
Show on map button pressed	Load up map with a red pin showing the location of the Aspecto store	P	Map view showing Aspecto's store presented
Call button pressed	Should dial phone number of Aspecto's Leeds store	F	The app rings the number perfectly. However; when the call ends it doesn't direct the users back to the app. Fixed issue and re tested. Works perfectly fine.
Directions button pressed	Load up directions from user's current location to Aspecto store, in Google maps	F	Directions loaded up but directions provided are for cars. Fixed issue by changing transit-type to walking. Re tested gave no issues.

Appendix F1: User evaluation

Introduction

The purpose of this activity is to evaluate the 'Explore Leeds' mobile application. You are asked to perform the tasks below. During the test, you will be observed. Once you have carried out the tasks, you will be asked a series of questions. The responses and feedback gained from this will aid to evaluate the usability of the application.

All data collected from this activity will be confidential and individuals can be kept anonymous, if required. You may stop for a break or withdraw from the entire test at any point.

Task list

1. Open the app and find the nearest car park to park your car. Find a car park that has 24 hours parking.
2. Access the shop section, select one of the fashion shops and visit their website
3. Go back to the app and find the directions to the shop
4. Go back to the app and return to the home screen
5. You're a bit lost, find out where you are on the map
6. Return to the home screen
7. You want something to eat, call Akbars to book a table
8. Your table will be ready in 15 minutes so you make your way over to Akbars but you don't know the way. Find directions to Akbars.
9. Go back to the app and return to the home screen
10. You want to watch a movie, find out where the cinema is on the map
11. Go back to the home screen
12. You find that it is snowing in Leeds and as a safety precaution; you decide to catch the train home. Find out when the next train is.
13. Now find directions to the train station.
14. Go back to the app and return to the home screen.
15. Close the app.

Appendix F2: User observation

Task	Completion success? (Y / N)	Number of clicks	Comments
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

Appendix F3: Evaluation Interview

1. **On a scale of 1 – 5, how would you rate the following statement?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

- The application is easy to use

2. **On a scale of 1 – 5, how would you rate the following statement?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

- I can find the information that I need without any major issues

3. **On a scale of 1 – 5, how would you rate the following statement?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

- The user interface is pleasing

4. **On a scale of 1 – 5, how would you rate the following statement?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

- The application, as a whole, would be very useful for me when I visit the city centre, in the future

5. If you disagree/strongly disagree to the statements above, why?

6. **Did you have any issues when carrying out the tasks?** *(i.e. unable to select a button)*

7. **Was there something missing that you was expecting to see?** *(e.g. features, images)*

8. Which features did you find very useful?

9. Do you have any suggestions on how the app could be improved?

Appendix F4: Wild study Evaluation Questions

For questions 1 – 5 please leave two ratings, one for the use of local mobile search and one for the use of the mobile application.

1. **On a scale of 1 – 5, how would you rate the following statements?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

   - The application was easy to use

     Local mobile search: Mobile Application:

2. **On a scale of 1 – 5, how would you rate the following statements?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

   - I was able to find something that I wanted to know efficiently

     Local mobile search: Mobile Application:

3. **On a scale of 1 – 5, how would you rate the following statements?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

   - The user interface was pleasing

     Local mobile search: Mobile Application:

4. **On a scale of 1 – 5, how would you rate the following statements?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

   - The application enhanced my experience of visiting the city centre

     Local mobile search: Mobile Application:

5. **On a scale of 1 – 5, how would you rate the following statements?** *(1 – strongly agree, 2 – agree, 3 – neither, 4 – disagree, 5 – strong disagree)*

- I would use this application again

  Local mobile search: Mobile Application:

6. **If you disagree / strongly disagree with any statements above, why?**

7. **When did you use the application?** *(i.e. when sat down or when walking around the city centre)*

8. **Did you have any issues when using the application?**

9. **Which features did you like the most?**

10. **Do you have any suggestions on how the app could be improved?**