

Final Report

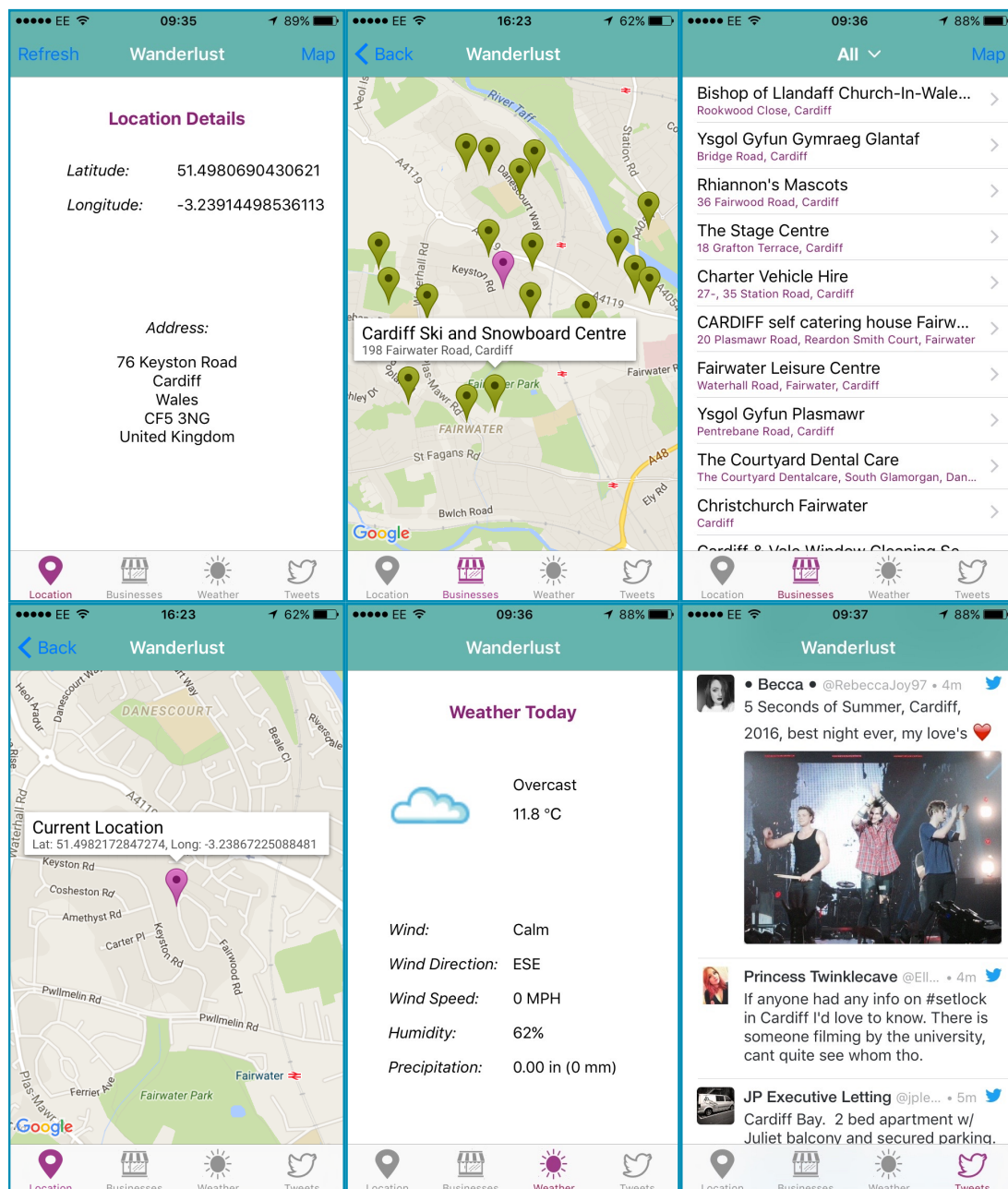
Project 164 - Location Based iOS Application

Author - Kelsey Dowling

Supervisor - Martin Chorley

Moderator - Alia Abdelmoty

CM3203 - One Semester Individual Project (40 Credits)



Abstract

The UK tourism industry continues to grow year after year, with more travellers deciding to holiday in Britain, exploring the countless historic landmarks and cultural features the country has to offer. Currently there is not an effective means of providing these tourists with a simple but useful way of finding information about their unfamiliar surroundings.

As a result, an iOS mobile application has been developed in this project using the Swift programming language, to provide users with location based information. Details such as a list of businesses located nearby, the weather where they currently are and Tweets sent about their location, were all provided to facilitate ease of activity planning and navigation in areas not known to the users. In order to implement these features, various APIs were used including Google Places, Weather Underground and Twitter. The main aim of this research study was to deduce whether it would be feasible to create an application that took the user's location as input and processed it to provide users with helpful information about their surroundings, given they were new to the area.

The application was subjected to three types of testing: lab-based functional testing, field testing and user testing. The results of the test phase indicated that users enjoyed using the mobile application and believed it to be very useful. All test participants indicated they would use the mobile application if it were commercially available. Further work is needed to overcome a few limitations of the project however, before it could be made publicly available, such as increasing the number of businesses returned by the Google Places API request. In conclusion, the study indicated it would be feasible to create an application with such features and that it should be well received if it were launched commercially in the future.

Acknowledgements

- Thank you to Martin Chorley for supervising this project, and providing unwavering academic support over its duration. The feedback and advice he provided was very insightful, contributing to the project's success.
- Thank you to the data providers Google, Weather Underground and Twitter, for making their APIs free to use; without them it would have been impossible to complete the project over the given time period.
- Thank you to the test participants who agreed to test the mobile application: Dean McDevitt, Sadie Pulman, Nicole Oduwaiye, Ellyn Berman-Jones, Mike Atherton, Ryan Jones, Abbie Hey-Smith, and Adam Davies; their insight and feedback was incredibly valuable.
- Finally, thank you to my loving partner, family and friends who have supported me over the course of my studies. Their kindness and encouragement has always allowed me to believe I can achieve anything I set my mind to.

Contents

Introduction	6
Aims and Goals of the Project	6
Intended Audience	8
Project Approach	8
Developmental Project Schedule	9
Project Constraints	10
Project Assumptions	11
Importance of the Project	11
Project Outcomes	11
Background and Research	12
Research Questions	12
Types of Location Technologies	13
Mobile Device Platforms	14
Similar Applications in the Market	17
Elements to Include in My Application	20
APIs To Be Used In The Project	20
Specification & Design	24
User Requirements	24
User Interface Designs	27
Navigational Flow of Application	32
Application Use Cases	33
System Architecture	36
Implementation	39
GPS Location	39
Google Maps GMS Map	40
Google Places API Request	41

Weather Underground API Request	42
Tweets About the Location	42
Error Handling	43
Testing	45
Lab Testing	45
Field Testing	50
User Testing	54
Results and Evaluation	56
Main Findings of the Study	56
Alternative Explanations for Project Outcome	57
Limitations of the Project	58
Future Work	59
Conclusions	60
Reflections	61
Bibliography	63
Appendices	66

Introduction

Tourism in the UK is a growing industry with more people visiting the country year after year. VisitBritain's tourism forecast for 2016 is for 36.7 million visits, an increase of 3.8% on 2015, and £22.9bn in visitor spending, an increase of 4.2% on 2015 (Visit Britain, 2016). This project consisted of implementing an application that could support this influx of tourists during their many visits to the country.

For this project, it was intended that a mobile application be developed to provide users with information based on their current location. The application accesses geographical and contextual information about the user's device to provide them with information using the device's location. The information the application retrieves is then processed to provide the user with a variety of details related to their position. Types of data provided include the user's current address and co-ordinates, which is then pin plotted onto a map of the area. The application also displays weather information for that location as well as businesses in the surrounding area. As a result of the project schedule progressing well, the application's functionality was able to be developed further, allowing users to retrieve Tweets sent about their location in addition to the application's other features.

Aims and Goals of the Project

The project had a number of aims and objectives which have been split up into primary and secondary objectives for the project.

Primary Aims & Objectives

The main aim of the project was that it needed to be able to gather geographical data about user's location and use this to display a variety of information to the user about their current location. The primary aspects of this aim included:

- **Co-ordinates** – the application needed to be able to access the user's location co-ordinates and plot these on a map of the area. It also needed to access these details in order to provide the user with the rest of the location-specific information.
- **Weather Information** – the application needed to use the device's location details to provide the user with information about the weather in that particular area.

- **Businesses Nearby** – the application also needed to use the device's location details to inform the user of businesses located near them which may have been of interest such as shops, restaurants and hotels.

Another aim of the project was to design and implement an effective user interface for the application to ensure it was easy to use and displayed information suitably. The main aspects of this objective included:

- **Easy to Use Interface** – the application's user interface needed to be as intuitive as possible to ensure it was easy to use. This meant having a logical navigation system in place to allow users to transition between different screens sensibly and with ease, in order to view different pieces of information about the location.
- **Information Displayed Suitably** – the application also needed to display all information in the user interface clearly and with suitable formatting to ensure information was easy to read and accessible for the user. This meant having their location plotted on a map, businesses nearby in a list, etc.

Secondary Aims & Objectives

The application's functionality could be developed and extended to include more information related to the user's location, such as Tweets sent about their location and Instagram photographs taken near that area.

- **Tweets Sent About the Location** – the application may also be able to display Tweets sent about the location to the user. This could provide them with an interesting insight into the thoughts and behaviours of people in that area as well as events that may be going on around them. This was however a secondary objective of the project, so would only be developed if there was sufficient time during the course of the project to implement such a feature.
- **Instagram Photos Taken Near the Location** – the application may also be able to display a range of photographs taken near the user's current location. This could provide them with exciting sights they may wish to see or landmarks near the areas which may be of interest to them.

Intended Audience

The intended audience for the project were predominantly tourists and people travelling to new areas, as the application aimed to provide details of businesses nearby and weather conditions. This is the type of information people visiting a new place would not be familiar with so the application could assist them greatly in this respect. As a result, the application could potentially have a very wide and varied user base as most people travel during their lifetime so could make use of an application of this nature.

Another audience the application focussed on were students. Upon arriving at university new students are just getting to know a particular area and are unaware of their surroundings. This application could help them settle into their new homes and gain familiarity with the climate and businesses their new city contains. It could also enable them to find interesting student hangouts, places to eat out and meet new friends.

The majority of people seem to fall into one of these two categories at some point during their lifetimes so it was important that the application be able to accommodate as many different types of users as possible. Therefore, it was crucial the application was easy to use and information was displayed logically so people with varying levels of technical expertise could utilise its services with ease.

Project Approach

A phased approach was used for this project, by splitting it up into its 5 main areas: research, design, implementation, testing and reflection. This approach is similar to that of the traditional Waterfall development methodology (Jim Bowes, 2014), however it was adapted slightly to suit the specifics of this project.

The research and discovery phase of the project involved completing research into what the best direction for the project would be. This included looking at available location technologies and mobile application platforms, as well as other applications in the market, to determinate the most appropriate path for the project. It also included investigating which APIs the application would be able to make use of to provide users with the various types location specific information. By the end of this phase I had a clear idea of what form the application would take and what tools I could utilise to develop the application.

The second phase of the project was the specification and design phase. This entailed creating mock ups of the user interface as well as outlining the various functional and non-functional user requirements of the system. At the completion of this phase it was clear what the project would involve and what the project priorities were in terms of functionality. It was also evident what the style and layout of the basic user interface would be as well as the navigational flow for the application.

Next came the implementation phase for the project, which consisted of following the design details outlined in the second phase to deliver a working prototype application. This phase essentially involved coding the features and functionality of the application in the chosen language. I anticipated some issues or trouble shooting would occur during this phase which could have caused the project to overrun its schedule slightly, however extra time was allocated into the plan to allow for this. At the closure of this phase a basic, working prototype was created which was ready to be tested and improvements made.

This allowed the movement of the project into the testing phase. The application prototype was then tested using functional testing and usability testing techniques. This ensured the application functioned as it should in accordance with the user requirements set out in the design phase, and that it was fit for purpose. This highlighted whether users were happy that the information provided was useful and presented well in comparison with other similar applications.

The final phase of the project was the reflection phase. This was a crucial time for the project as it was here that the application was evaluated, and lessons learned from the project were realised. It was evident which requirements were met and which areas could have been developed further, as well as what aspects of the project were successful. This reflection phase enabled me to identify the growth I had developed over the course of the project, in addition to areas for future development in my personal project management style which will prove invaluable when undertaking projects in the future. At the conclusion of this phase, the project was considered to be completed.

Developmental Project Schedule

A schedule for the project was created and followed throughout the course of the project. This has been presented in the form of a Gantt chart and can be viewed in Appendix 1 of this document.

Project Constraints

There were various constraints associated with the project. One of these constraints was that various external APIs were used in order to provide the user with information on the current weather conditions and businesses located near them. As a result, if something happened to these data providers the application would no longer function correctly. This was unlikely however, as the APIs used in this project were selected due to their provider's dominance in the market and because they were well established data providers in their respective fields. As such they should have been reliable sources of information which were unlikely to be shut down or out of business any time over the course of the project. This risk was mitigated however by implementing each feature independently of one another. This ensured that if one service was temporarily unavailable, for example Twitter, the application's other features would still function. The Twitter page would simply be temporarily useless to the user until the service was restored.

Another major constraint of the project was its duration. The schedule for research, design, implementation, testing and reflection of the project was 12 weeks which is an extremely short time. As a result, it was imperative that I managed the project carefully and tried to abide by the timeline set out in the Gantt chart as closely as possible (see Appendix 1 for full details). Due to the limited time scale, it was important that the project priorities were kept in order and crucial functionality was implemented first, to ensure a working prototype containing most of the basic features could be demonstrated by the end of the project.

The biggest constraint for the project however, was my lack of experience in developing a mobile application. This was the first time I had created a mobile application, which could have resulted in some complications arising during the project. In order to reduce this risk, extra time was scheduled at the end of the project to allow these problems to be worked through. I also had the support of my supervisor to guide me and offer advice where necessary. Therefore, this opportunity allowed me to develop many new skills and gain knowledge in a domain I had no previous experience in, which I believe has allowed me to strengthen my character and professionalism for the future.

Project Assumptions

The predominant assumption for the project was that users would be in possession of a smart device such as a smartphone or tablet and would also have location enabled technology on their device. The device would also have to be connected to the internet in order to allow the application to send requests to the various APIs, allowing the application to get and display information to the users appropriately.

Importance of the Project

I believe this project is significantly important as it has created a new mobile application which could be used to provide its users with a wealth of different information pertaining to their location. Based on research into the mobile applications market, I felt there was a gap in the market for this project.

Alternative applications focussed on either providing weather information or listing businesses in the area, none however aimed to combine this information to make it useful for tourists. Other applications also provided route planning services whereas this project provided users with much more. It also gave them details of the climate they could expect and the types of businesses around. Additionally it allowed them to gain a feel for the attitudes and customs of the people in that particular location, through viewing Tweets sent by users about their current position. This enabled them to obtain inside information on what the place was like to visit so they knew exactly what to expect of the people and the area. No previous application found during market research provided this combination of functionality, meaning there was potential for the application to be very successful, filling a gap in the market.

Project Outcomes

The project's main outcome was to create a fully functioning mobile application which allowed users to get their location data, display their address and plot their current location on a map. It also aimed to provide them with a list of nearby businesses as well as the current weather conditions at their location. The secondary outcome was to understand if this application could be used by tourists effectively, and whether the application could be extended further to include other information such as photos taken of the location. The application's intelligence could also be improved in future, with the data retrieved from the various APIs being carefully selected to provide contextually aware information. Examples of this are; only showing businesses to the user which are currently open, or only showing details of ice cream parlours on a hot day.

Background and Research

There were a variety of applications available at the start of the project that provided users with location information. However, most only displayed one type of information to the user, such as how far away from the nearest bank they were. None seemed to have the multifaceted approach this project aimed to develop providing users with location, weather conditions, businesses and an overview of the attitudes of people in that particular location. For this project to be successful, various areas of the project needed to be researched to understand how best the application could be developed and implemented.

Research Questions

The main research question surrounding the project was therefore: would it be feasible to create an application that takes the user's location as input and processes it to provide them with useful information about their surroundings, given they are new to the area. To understand this, a test application was developed, providing the user with some basic information about their surroundings, with various test participants used to evaluate the application's effectiveness. Research into this consisted of 4 main sections:

1. Location Technologies Currently Available

- What location technologies were currently available?
- What were the advantages and disadvantages of each technology?
- Which technology would be best suited to this project?

2. Available Platforms

- What different mobile device platforms were available for implementation?
- What were the advantages and disadvantages of each platform?
- Which platform was best suited to the project?

3. Similar Applications

- What similar applications were currently in the market?
- What were the advantages and disadvantages of each application?
- How did these applications differ from this project's application?
- What common elements did these applications have?
- Could any of these useful common elements be implemented in this project?

4. Available APIs

- What APIs were available for providing the information this application requires?
- What were the advantages and disadvantages of each API?
- Which APIs were best suited for use in this project?

Types of Location Technologies

The first research section to tackle was that of location technologies. There were a variety of different types of location technologies available, so it was important to evaluate each technology's advantages and disadvantages in order to determine which was best suited to this project. The main three technologies identified were GPS (Global Positioning System), GSM localisation, and NLBS (Near Location Based Services).

GPS Technology

GPS (Global Positioning System) technology relies on receiving information from 4 of the 24 satellites in orbit above the Earth. These satellites transmit information to the Earth which is picked up by GPS receivers in devices such as smartphones. The receivers use triangulation methods on the data they obtain, in order to calculate the receiver's distance from the satellite. With distance measures from multiple satellites, the receiver can determine the user's position and display this on a map for example (Garmin, 2016).

Advantages of using GPS technology are that most modern devices today contain GPS receivers, so users do not have to go out and purchase new devices in order to use applications which utilise GPS technology. It is also fairly accurate, usually between 5 - 20 metres depending on environmental conditions. A disadvantage of GPS technology however, is that for trilateration to work correctly the receiver must have a clear line of sight to the satellites. This means GPS is not well suited for use indoors in environments such as shopping centres but works very well for use outside.

GSM Localisation

GSM localisation technology is the finding of a mobile device in relation to its cell site or base station in order to find out the location of a user. In this type of network based tracking, the service provider's network infrastructure is used to calculate the location of the GSM enabled device. The geographical position of the device is found using various techniques like Time Difference of Arrival (TDOA) or Enhanced Observed Time Difference (E-OTD).

The biggest advantage of network based techniques is that they can be implemented without intruding on the private space of the person or device that is being tracked. The technique does however require working closely with the service provider, as the process includes installation of various software and hardware in the network infrastructure. This can be a big disadvantage of using this technology, as it is often difficult to gain the cooperation of service providers (Dark Government, 2012).

Near Location Based Services (NLBS)

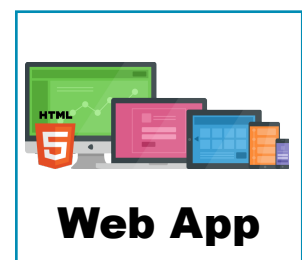
Near LBS are local-range technologies like Bluetooth, WLAN, infrared, RFID and Near Field Communication (NFC) technologies, which are used to match devices to nearby services. This allows users to access information based on their surrounding environment. NLBS are especially useful for inside use in closed premises such as shopping centres (MediaBUZZ Pte Ltd, 2012). This would work for example using RFID tags. Each store or business could have its own RFID tag located inside and users would use readers to find the locations of these stores. An advantage of this technology is it is very precise down to even sometimes centimetres of accuracy, but is more suited to smaller scale projects which are indoors, such as creating a tour of a museum or for locating stores within a shopping centre.

Technology Most Suited To The Project

After research and analysis it seemed the location technology most suited for use in this project was GPS. This was because most modern devices contain GPS enabled receivers, so users would not have to go out and purchase new devices specifically to use this application. Most would already have these capabilities available through their smartphones or tablets. The application was also designed for use outside where GPS technology is the most accurate, hence it was the most suitable option for this project. Getting service provider cooperation for such a small project was also infeasible so GSM localisation was not suitable. NLBS are more suited to smaller scale projects for use indoors, hence this was also not as well suited to the project.

Mobile Device Platforms

The second research area to be explored was that of mobile device platforms. There were 4 main mobile device platforms that could have been utilised in this project: iOS, Android, Windows, or a Web Application. Each platform had its own unique characteristics as well as advantages and disadvantages which had to be considered when deciding which platform was best.



iOS Platform

Apple's iOS platform allows new applications to be programmed in predominantly two languages. The first is the Objective-C programming language, an object oriented language built on top of the existing C language. The second is Swift, a programming language created for iOS, OS X, watchOS and tvOS by Apple.

After research it was clear that if an iOS platform was chosen for the project, the Swift programming language should be used over the Objective-C option. Swift has a variety of advantages that make it the better development language, including that it is much easier to read and maintain. Swift code more closely resembles natural English and does not require line terminations using semicolons or parentheses to surround conditional expressions inside if and else statements (Apple, 2016e), making it much easier to work with from a development perspective. Objective-C is also held back due to its reliance on the C programming language. C requires programmers to maintain two code files in order to improve build times and efficiency of application creation, whereas Swift does not, giving it an advantage in terms of development (Paul Solt, 2015).

Android

Android applications are developed using the Java programming language. Most developers usually have some degree of previous experience with this language, making it a favourable choice of platform. It is also highly customisable, making it a favourite for many developers who enjoy having less restrictions than that of iOS's closed system environment. However, this can result in a reduced user interface experience, with novice users feeling overwhelmed by too many customisable options. This means Android applications often struggle to provide the user friendly interfaces boasted by their iOS counterparts (Bryan Wilde, 2013).

Windows

Microsoft has been very successful in the PC market in the past but so far have not managed to convert this success into results in the mobile applications market, with less than a 5% market share in the mobile development market worldwide (Bryan Wilde, 2013). Windows mobile applications can be developed in a number of languages, predominantly Visual C++, Visual C# and Visual Basic. At the start of the project I had no previous experience using these languages as well as no experience using or owning a Windows mobile phone. As a result, I felt developing the project using this platform would have been a big risk. It would also have limited the potential user base for the application due to Windows' user base being significantly smaller than that of the iOS and Android market.

Web Application

Web applications are programs that are stored on remote servers and delivered to users over the internet through a browser interface (Tech Target, 2011). Web applications are typically built using a combination of HTML and CSS, sometimes with Javascript and other scripting languages being used to implement specific features depending on what the application is trying to achieve. A significant benefit of creating a web application is that it can be used by various platforms, as opposed to a native application which will only work on the platform it has been designed for. Therefore a web application has the potential to reach a very vast user base. However, web applications can be slower and less responsive than native applications due to them being run in a browser, whereas native applications work well with the device's built in features, meaning they perform better on the device (Priya Viswanathan, 2015).

Platform Most Suited To The Project

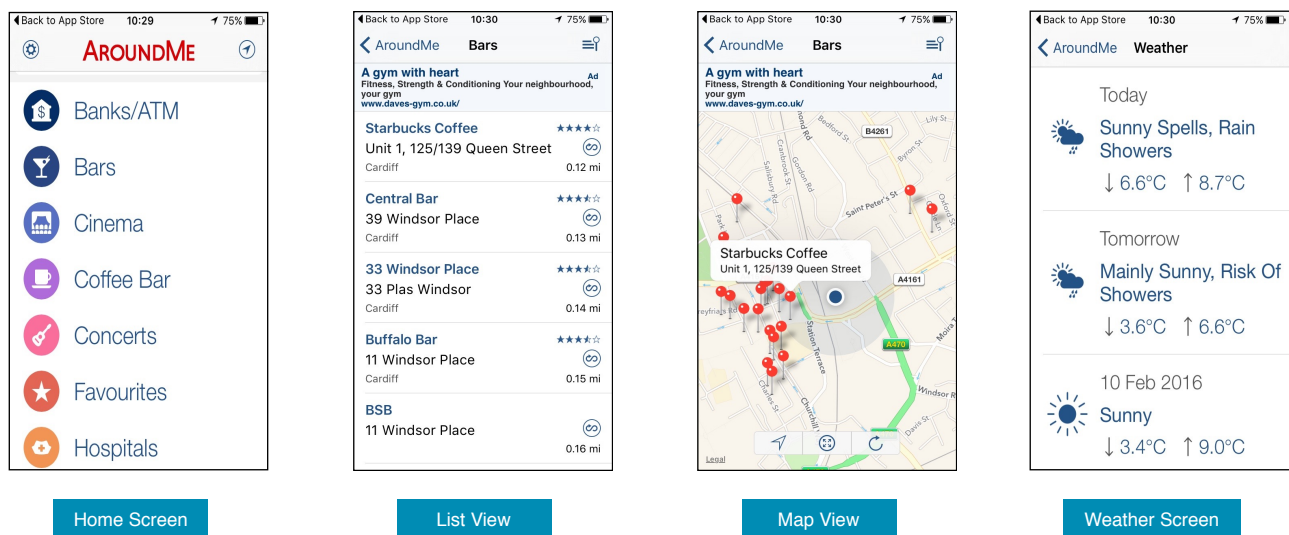
The platform best suited for use in this project was the iOS platform due to its user friendly nature. The interfaces created in iOS applications allow users to interact with applications very intuitively, which was important for this project as it needed to be used easily by a large variety of users. I also own an iPhone and iPad so had substantial familiarity using these devices and applications. I felt this enabled me to develop an application consistent with the iOS style present in standard applications. I was also keen to challenge myself and use this project as a development opportunity to learn a new language. Therefore I felt it would be beneficial to implement the application using Swift in iOS, as opposed to using Java to create an Android application or using HTML and CSS to create a web application, as I already possessed some experience developing for these platforms.

Similar Applications in the Market

The third research section was to identify similar applications in the market, and to evaluate how they differed from the application developed in this project. This also helped with some design elements of the application, as some commonly used features or functionality which worked well could also be incorporated in the development of this project. Various applications were found in Apple's App Store:

Around Me

The most similar application identified during the project's preliminary research was called 'Around Me' (Flying Code Ltd, 2016). This application worked by using location co-ordinates from the user's device. It then processed them to provide the user with a complete list of all the businesses in the category the selected by the user. Categories included bars, restaurants, hospitals, hotels and several others. It also provided some basic weather information to the user.

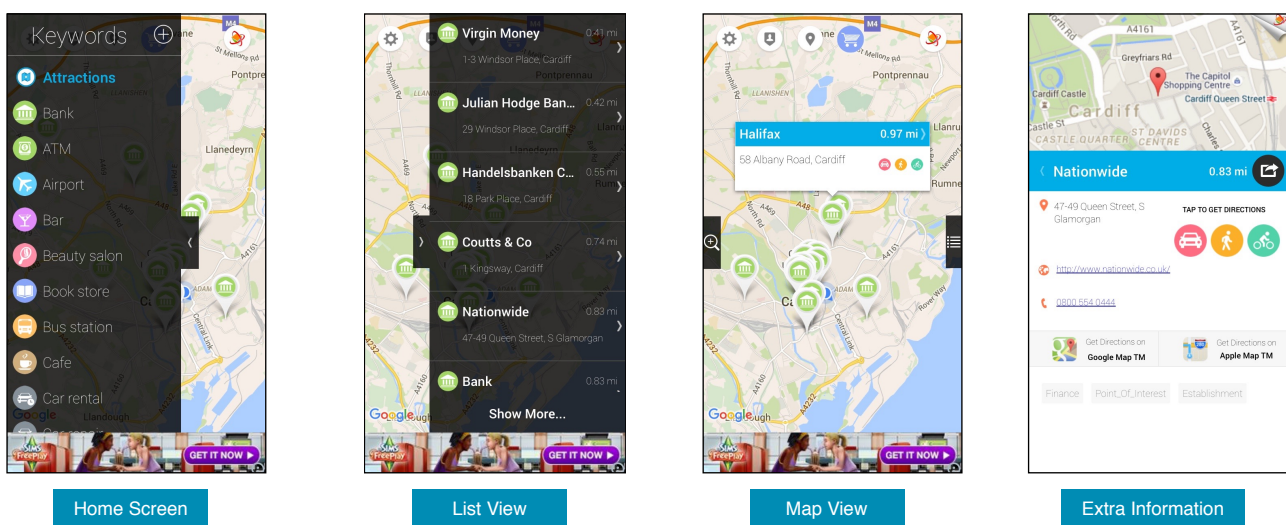


An advantage of the application was that it allowed users to filter which businesses they wished to view by selecting a category, then only displaying businesses that fit that description. However this could also be seen as a disadvantage as users may not have wished to specify which category of businesses they wanted to view. They may have wanted to simply browse all the businesses nearby including shops, restaurants, and others rather than only view bars nearby for example. Another good aspect of the application was that it allowed users to view the businesses that fitted their search in two views: a list view or map view. This was something that was taken into account and developed when this project's user interface was designed.

A disadvantage of this application however, was that sometimes businesses were displayed that did not fit into the category that had been selected. As you can see from the 'Map View' screenshot on the previous page, the category 'Bars' was been selected, however 'Starbucks' was displayed as one of the businesses nearby, which did not fit the category selected. Therefore it was important to note that if this project were to implement category selection and filtering, a better categorisation system should have been implemented to ensure only suitable businesses were displayed to the user.

Find Near Me

The 'Find Near Me' application was also similar to the application developed in this project in some respects, mainly in that it provided the user with a list of businesses (XLabz Technologies Pvt. Ltd, 2016). It used a similar approach to that of the previously mentioned 'Around Me' application in terms of functionality and user interface design. Both applications required the user to select a business category type, then filtered the results displayed based on the user's selection. It is clear from the screen shots below that the layout of the applications were similar and seemed to work effectively in the context of the application. If filtering based on categories were employed in this project a similar UI design could be adopted.

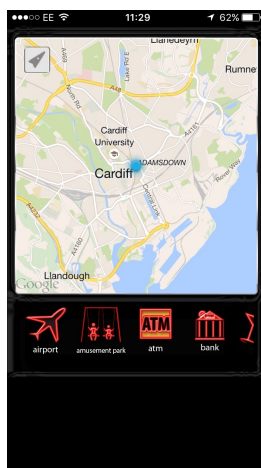


An advantage of the 'Find Near Me' application was that it provided some useful extra information about the place/business when it was selected from the list or map. This was a useful feature that was then implemented in this project. A disadvantage however, was that the information provided was quite limited. This could therefore have been expanded to also include useful information such as opening times to the user. Another good feature of this application was that it included a link to either Google Maps or Apple Maps which the user could select, then be redirected to these other applications to receive direction information.

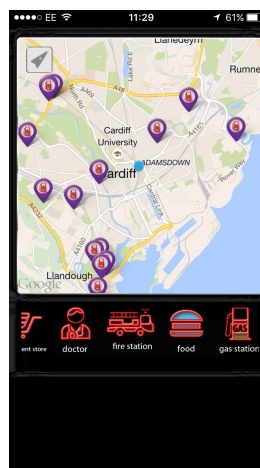
Nearby Places

The 'Nearby Places' application again provided the user with a list of businesses based on a categorisation selection. It then simply plotted the businesses that fell into the category selected onto a map. The functionality of the application was extremely limited however, as it did not provide the user with any other ways to interact with the information displayed. It did not allow users to be directed to another source such as Google for directions (as in the case of the 'Find Near Me' application), nor did it display any extra information about the business aside from the business name.

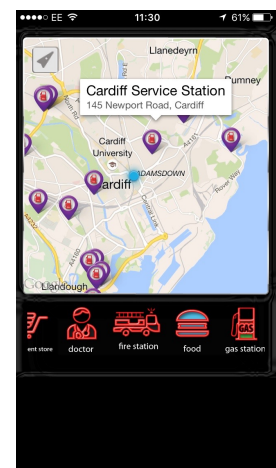
This differed significantly from the application being created in this project in that 'Nearby Places' only provided the most minimal amount of information on local businesses. It did not have the extended functionality of this project's application as it did not provide the user with any weather information or Tweets sent from near that location. It also only displayed businesses based on the category chosen, whereas the application developed in this project aimed to provide the user with a variety of information about their location, with little input from the user. This project also aimed to provide the user with a much more enhanced and enjoyable application experience in terms of user interface design and the degree of interactivity.



Home Screen



Map View



Information Provided

Elements to Include in My Application

Research into the market of other GPS based location applications allowed me to identify a number of features that I wanted to implement in this project. It enabled me to identify features that worked well, as well as various disadvantages of different approaches. Firstly I thought it would be beneficial to allow users to view results in both a list and map view, as was demonstrated in the 'Around Me' application, as this gave the users flexibility to choose the view that suited them best. Some people prefer to view information in a very visual way such as on a map representation, while others prefer simplistic list structures to read which businesses are nearby.

Secondly, I also thought it would be worthwhile to allow users to select the business they were interested in from the list, then have the application display some extra information or details about the business, such as its address and the distance the user is away from that particular business. This would enable them to then make informed decisions based on this information as to what they would like to do. For example they may wish to choose a restaurant which is located closer to them when it is raining, rather than one which is 3 miles away.

Finally, I thought it was incredibly useful if the application could provide links to external services and applications which could provide them with directions. As the application was most suited to providing information to people new to the area, it was understandable that they may wish to be provided with directions to a business of their choice. Providing links to Google Maps for example, could really improve the applications appeal to potential users.

APIs To Be Used In The Project

The final area of research for this project was to investigate and understand which APIs were available that the application could utilise and manipulate in order to provide users with the most helpful information possible. APIs needed to be identified to provide users with both a list of businesses nearby and weather information. Various weather information providers were discovered which may have been suitable for this project:

Weather Underground

The Weather Underground API contains 3 different plans each with their own capabilities. The plan most suited to this project was the “Stratus plan” as this contained the basic information that needed to be displayed on the weather page of the mobile application. It can provide details of the current temperature, humidity, wind speed, wind direction and more in a given location when the latitude and longitude is provided in a “Conditions” request. Developers can also choose which level of API usage they require. For the purposes of this project the lowest usage plan sufficed, which provided the application with access to the API for 500 free calls per day. Another significant advantage of using this API was that it contained clear documentation with example requests shown to assist developers with integrating this API into their applications (The Weather Company, LLC, 2016).

World Weather Online

The World Weather Online provides users with a variety of different weather APIs; the one most suited to this project was the “Local Weather API”. This API provides support for 3 million towns and cities, indicating its vast use and seeming reliability. World Weather Online also provides developers with helpful documentation on how to construct a suitable request and integrate this functionality into their application successfully. This was a very big advantage for this project as I had not previously integrated API data into an application before due to being new to mobile development. Therefore, having support from this API would have proved invaluable. The Weather Online API also offers a free or premium subscription. Since this project was for research purposes only a free service was desirable, however the usage rates for the free service with this API were quite low with only 250 requests per day being accepted (World Weather Online, 2016).

Forecast io

The Forecast io API is an excellent weather API to use, as a wealth of detail is provided to the developer for use in their application. However, this could also be seen as a disadvantage in this instance, as being a novice user meant it may have been difficult to locate and mine the basic information I wished to display. Using so much data may have overcomplicated the simplistic design I was trying to achieve in my application. However a big advantage of this API is the amenable API call rates of 1,000 calls per day. This would have certainly been enough to support the application developed as part of this research project (The Dark Sky Company, LLC, 2016).

There were two main APIs for consideration in this project that could provide data on businesses in the area: Foursquare and Google Places.

Foursquare

The Foursquare API allows developers to make 5,000 requests per hour to the API for information. This was an extremely big advantage as it meant I could be almost certain that the application would not run out of API call requests, even during vigorous use of the API such as during the testing phase of the project. Regular API calls would be being made at this time as the functionality of the application would be being tested repeatedly to ensure it worked correctly, so it was important the number of API calls allowed was reasonable. There were two types of requests that would have been suitable for returning a list of businesses nearby: either a search request or an explore request (Foursquare, 2016). An explore request is better at answering the question “what are some popular coffeeshops in this area?”, whereas search requests are better at “where is the nearest Joe’s Coffee?” or “where am I right now?”.

Therefore it would have been important, if this API was chosen, to evaluate which request would have been most suitable for use in this project. A significant disadvantage of using the Foursquare API was that the data stored is crowd sourced, meaning that although it could provide often very up-to-date listings, an overwhelming number of duplicate or “fake” listings could have occurred which may have cluttered the application and made it look very unprofessional.

Google Places

A significant benefit the Google Places API can boast is that its coverage of places is excellent, with Google having the biggest businesses database out of the current business APIs. Another useful aspect of the Google Places API was that a suitable number of requests to the API could be made: the free limit is currently set at 150,000 requests per day by the mobile application. Google is also a huge company meaning their API reliability and speed is superb, with the reputation of the company indicating that was very unlikely that the API would experience much downtime or response lagging. One disadvantage however was that performing a Google Places search request would only return 20 results, limiting the amount of information available to users (Google, 2016c). This limitation was not a problem during the course of this project as it was only a research project, so was being used only for proof of concept. However, if the app were to be launched commercially in future this would need to be addressed.

APIs Most Suited To The Project

The weather API most suited for use in the project was Weather Underground due to its developer friendly documentation and suitable allowance of API requests per day. This API provided all the information necessary for display on the weather page of the mobile application as stipulated in the design section of this report. Information such as the current temperature, humidity and wind speed could be processed and retrieved quite easily when following the guidelines provided by the API owners. Forecast io, whilst also providing all the necessary information to display on the weather page, supplied too much data. Using this API would have therefore made it difficult as a novice user to mine and find the useful data and discard the extra details included in the API responses. This discouraged me from using this API.

The businesses API most suited for use in this project was the Google Places API. It had better coverage than the Foursquare API in terms of the number of businesses stored in their databases. Another important advantage Google had over the Foursquare API was that the data stored in the Google Places API was largely from licensed providers, meaning the data they had was generally more reliable than the crowd provided data of the Foursquare API. Using crowdsourced data in the application could have caused the results displayed to seem unprofessional and inaccurate in some instances, which made the Foursquare API unsuitable for this application.

Specification & Design

The specification and design section of this report illustrates what the system will look like once completed in terms of its user interface, and how it will function in terms of the various user requirements of the system. This section also outlines the navigational flow of the application, detailing the transition between screens users experience when utilising the various different areas of functionality. Justification for design choices are also highlighted throughout the section.

User Requirements

First when moving into the design phase of the project, user requirements had to be identified to ensure that all important features and functionality of the system were included. During the implementation phase, these requirements could then be used as a guide on which features should be implemented first and were crucial for the success of the system (the functional requirements), and which were beneficial to have but not essential for the system to function appropriately (the non-functional requirements).

Functional Requirements

Below are a set of functional user requirements for the system. These requirements had to be implemented if the system was to function appropriately, containing all the desired features for the users.

1) Gather GPS Information About The Device	
Acceptance Criteria	On the touch of a button the application should gather GPS information about the device's location including its latitude, longitude and address, then present this to the user.
Justification	This is because users need to be able to find out where they are and have this information displayed to them by the application as they may not know their exact location because they are not familiar with the area.
2) Plot Device's Location On A Map Of The Area	
Acceptance Criteria	A map should be displayed to the user when a 'View on Map' option is chosen. The map should contain a pin indicating the user's current position.
Justification	Users can then view their current location in the context of the surrounding area, which will help them navigate and gain an awareness of where they are in relation to other places in the city. This will also help users who like to view information in a diagrammatic way as opposed to reading the address in text.

3) Gather Information About Businesses In The Area Using The Device's Location	
Acceptance Criteria	The application should use the device's location information (obtained in requirement 1) to perform a request to the API for which businesses are located nearby and receive a response.
Justification	This information is crucial to the functionality of the application. This requirement must be met in order to provide the user with information on the surrounding businesses in the area.
4) Display Information On Businesses Nearby To The User In A List Format	
Acceptance Criteria	The application should process the information received from the API and display it to the user in a suitable list format.
Justification	This will enable users to view the response in a clear, concise format, making it easy to read and digest.
5) Plot Businesses Nearby On A Map Display For The User To View	
Acceptance Criteria	The application should plot the locations of the various businesses returned in the API response, on a map for the user to view if they should wish.
Justification	This will enable users to view information on the location of the businesses in a diagrammatic way, making it easier to read for people who process information more effectively using a visual representation of data.
6) Gather Information About The Weather Conditions In The Area Using The Device's Location	
Acceptance Criteria	The application should use the device's location information (obtained in requirement 1) to perform a request to the API for what the weather conditions are like in the users current location and receive a response from the API.
Justification	This information is crucial to the functionality of the application. This requirement must be met in order to provide the user with information on the weather in their current location.
7) Display Weather Information To The User In An Appropriate Format	
Acceptance Criteria	The application should use the weather information received in the API response (obtained in requirement 6) and process this in order to present the user with suitable information about the current weather conditions near them. This should include the current temperature, a brief description and a summary icon, for example a sun on a sunny day or a cloud when it is cloudy.
Justification	This information is crucial to the functionality of the application. This requirement must be met in order to provide the user with information on the weather in their current location in a clear and concise manner.

8) Allow The User To Re-Gather Their Location Using A Refresh Button	
Acceptance Criteria	The application should allow users to re-gather their GPS information using a refresh button of some kind. This will enable them to gather fresh data on the user's location if they have moved away from the position they were in when they originally opened the application.
Justification	This is a very important feature if the application is to be user-friendly and effective. It is impractical for the user to have to quit the application and begin again if they wish to re-gather their location information. Therefore a refresh button is necessary.

9) Allow The User To Navigate The Application Screen With Ease	
Acceptance Criteria	A suitable navigational flow should be implemented in the application, that allows users to move throughout the various screens of the application easily and logically.
Justification	The user must be able to switch between the different screens with ease to enable them to view the desired content. Otherwise if they cannot navigate to the weather details screen for example, then they will be unable to access this part of the application's functionality.

Non-Functional Requirements

Below are a set of non-functional user requirements for the system. These requirements indicate desired aspects of the system but do not specifically relate to any exact system functionality.

1) Application Has A High Level Of Usability	
Acceptance Criteria	The application's layout, style and design should be created and implemented, following a set of heuristics to ensure the user interface is easy to use for all types of users. Neilson's Usability Heuristics have been used to design the applications interface to ensure this requirement should be met to a high standard.
Justification	It is essential that the application's interface be intuitive and easy to use for all types of users. As the audience for the application will mostly be tourists, anyone could be using the application to assist them. At some point during their lives, most people are unaware of their surroundings sometimes so may find themselves in need of using this application's features. Therefore, it is very important that user's can utilise the application's functionality simply.

2) Application Is Compatible With All Types Of iOS Device	
Acceptance Criteria	The application should be compatible with many different iOS devices such as iPhones and iPads. The layout and data in the application should therefore be presented in a suitable format on various devices and should not be hard coded to only display correctly on one type of device such as the iPhone 5 for example.
Justification	The application is designed to be used on iOS mobile devices, as there are many types of mobiles using iOS it would be impractical to make the system optimal on the test device only. As many types of users will be using this application and they will all possess different devices, it is important that the application is suitable for use on many different types of iOS device so as to accommodate as many users as possible.

3) Application Is Reliable	
Acceptance Criteria	The application should be suitably reliable, providing users with access to the information they desire provided all the necessary components are available such as GPS signal and internet access.
Justification	If the application is not reliable, users will not want to use it as they will be unable to access the details they need when they require them. It is important that the application is reliable as tourists may be relying on the application as their only way of finding a particular place. As a result, they may get lost if the application does not retain a suitable degree of reliability.

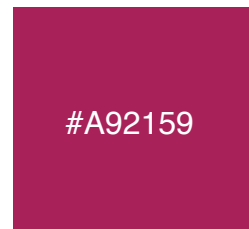
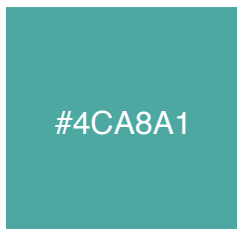
User Interface Designs

To ensure a clear picture of how the application would look was kept in mind during the implementation phase, it was vital that preliminary designs for the various application screens were created. The following are initial designs for the style, layout and colour scheme which would later be employed in the project application.

Throughout the design process an awareness of Nielsen's Usability Heuristics (Jakob Nielsen, 1995) was considered to ensure an effective, user-friendly interface was created, fostering ease of use for all user abilities. It was also important to note that this application was being created for the iOS platform which has its own guidelines on how applications should be designed to ensure the application's suitability to the expectations of its users. These guidelines were also considered during the design phase of this project (Apple, 2016a).

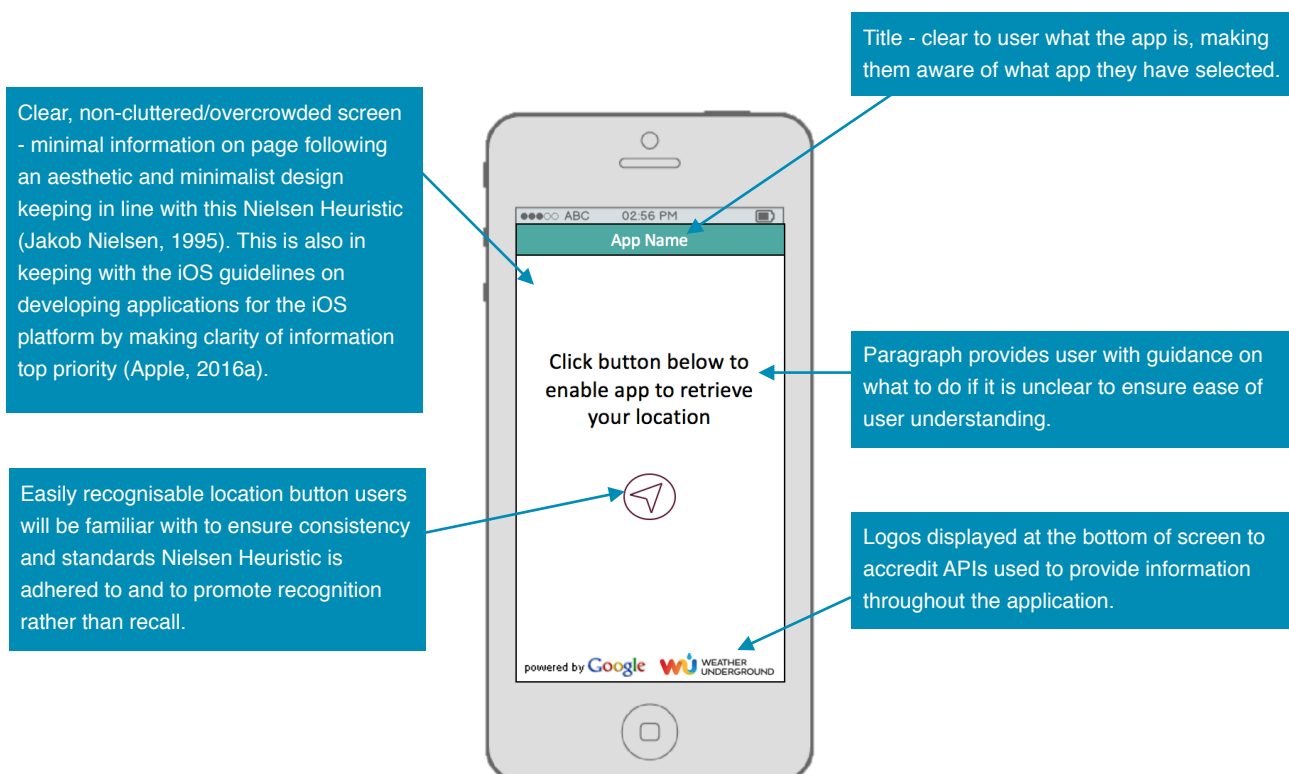
Colour Scheme

A professional and interesting colour scheme comprising of 3 main colours was chosen for the application in order to create a visually stimulating appearance for users. These colours were used throughout the application to create a continuous style and brand for the application.



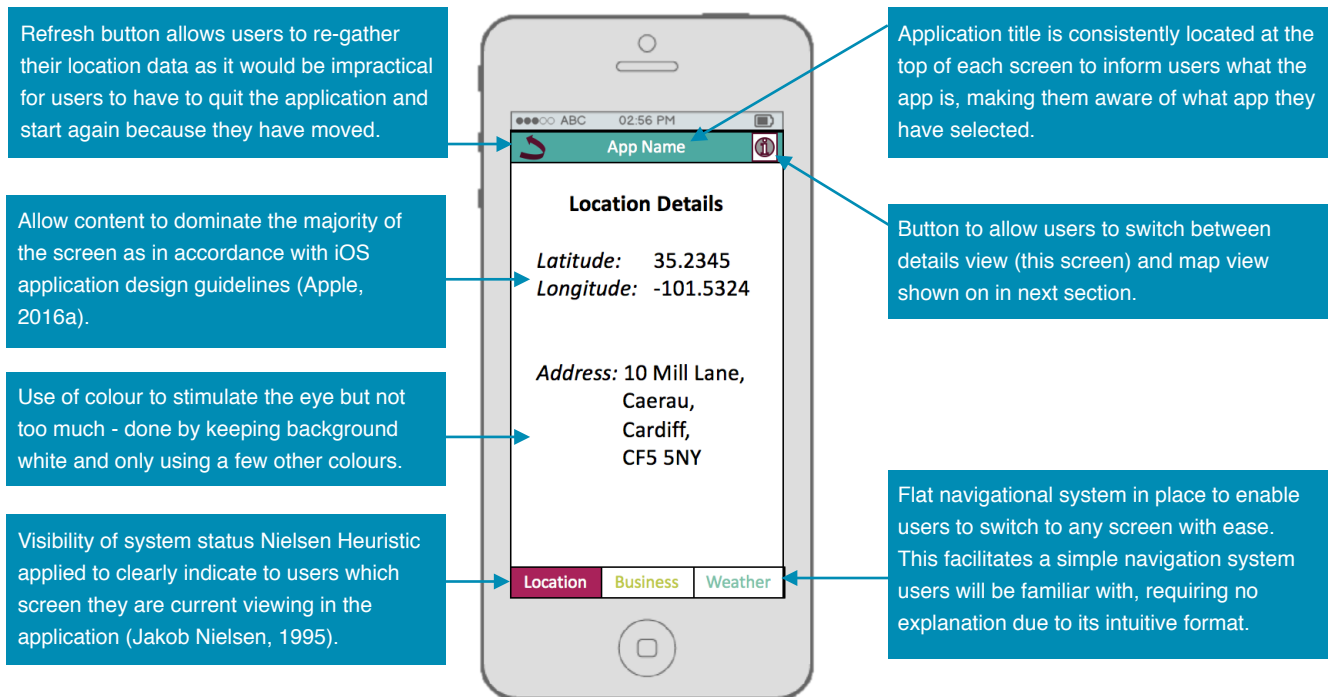
Initial Screen

The following design illustrates how the initial screen would look after completion of the implementation phase. The user is taken to this screen when the application first opens.



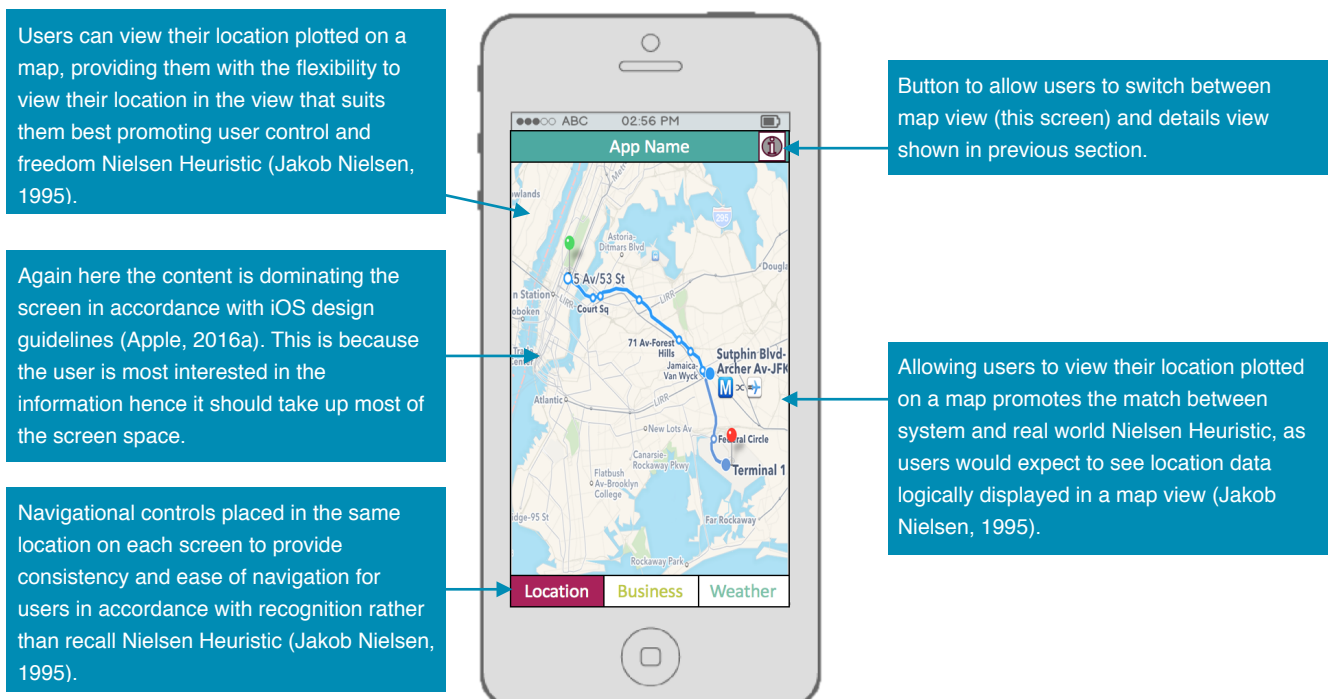
Location Details Screen

Users would be directed to this screen after selecting the “Get Location Information” button on the previous screen. Here they would be provided with various pieces of information about their location, including their latitude, longitude and address.



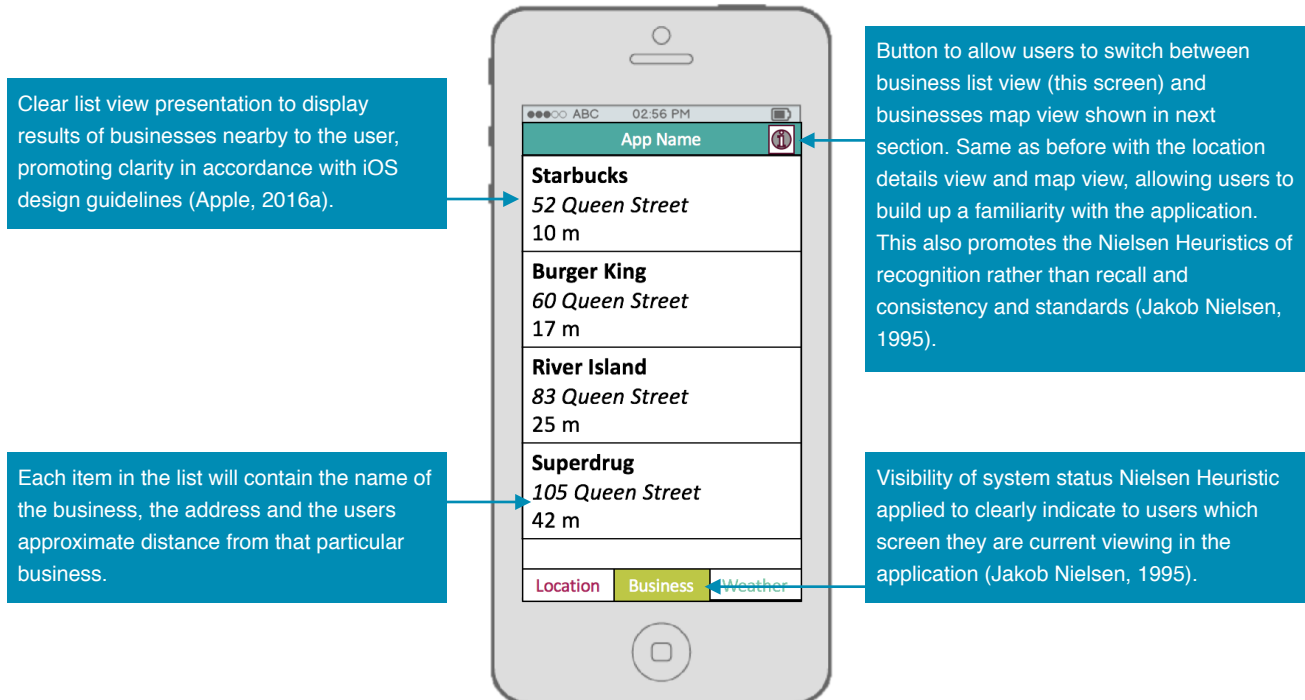
Location on Map Screen

By selecting the button located in the top right of the screen, users could switch between the previous text based location details view and this map based view.



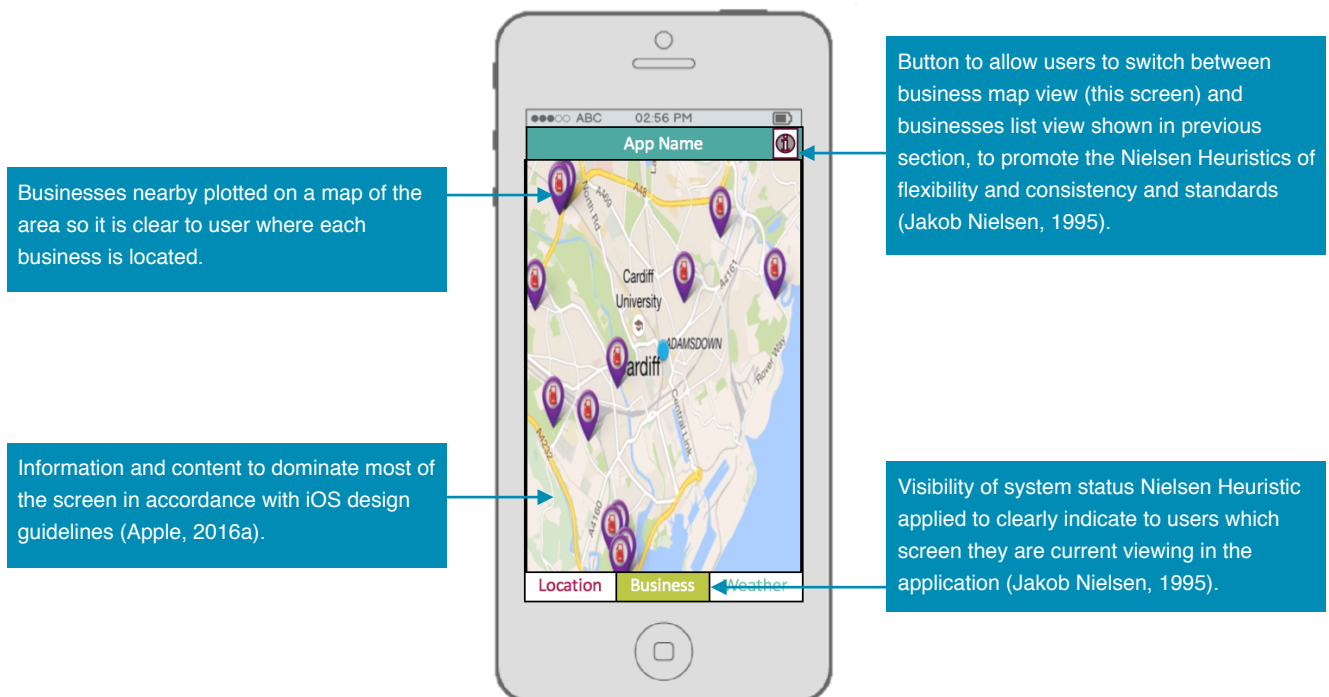
Businesses in List View Screen

Users could choose to view the businesses near them in a list view or map style view. This screenshot demonstrates how the application would look in a list view.



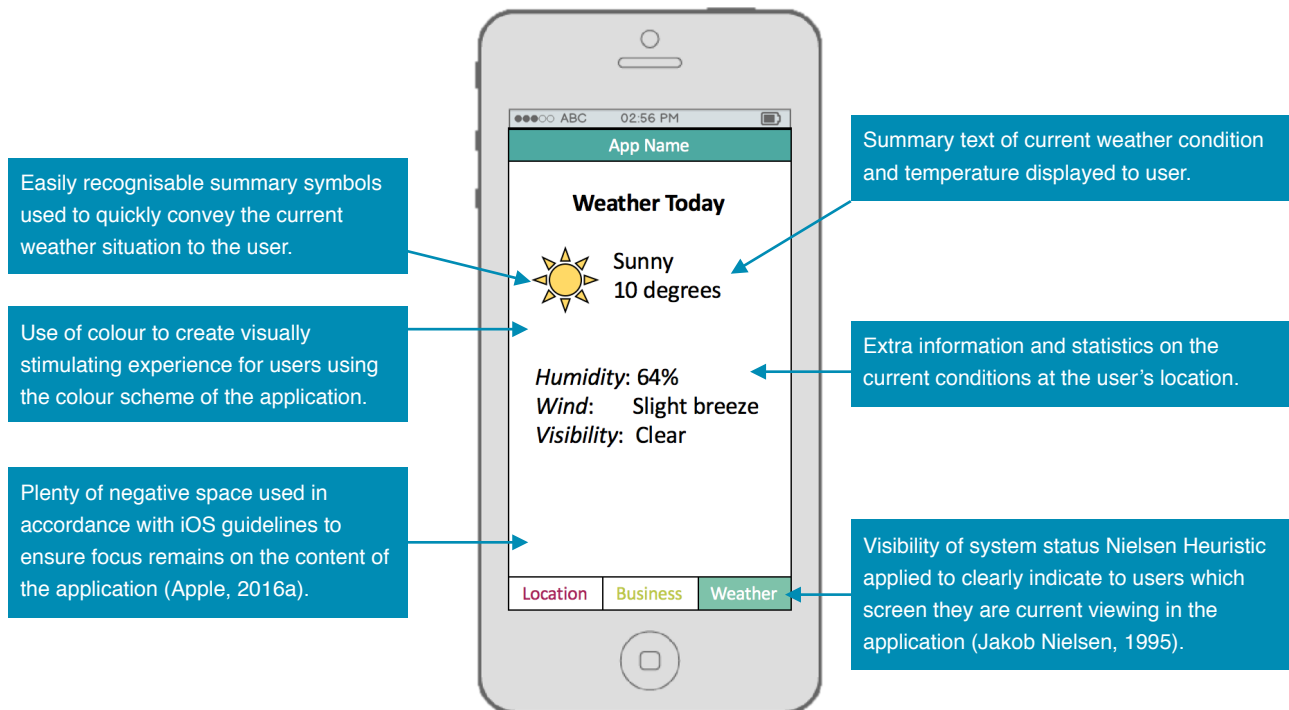
Businesses in Map View Screen

This screenshot indicates how the application would look when the user selects the button in the top right of the screen, to switch to view the businesses in map view as opposed to the previously shown list view.



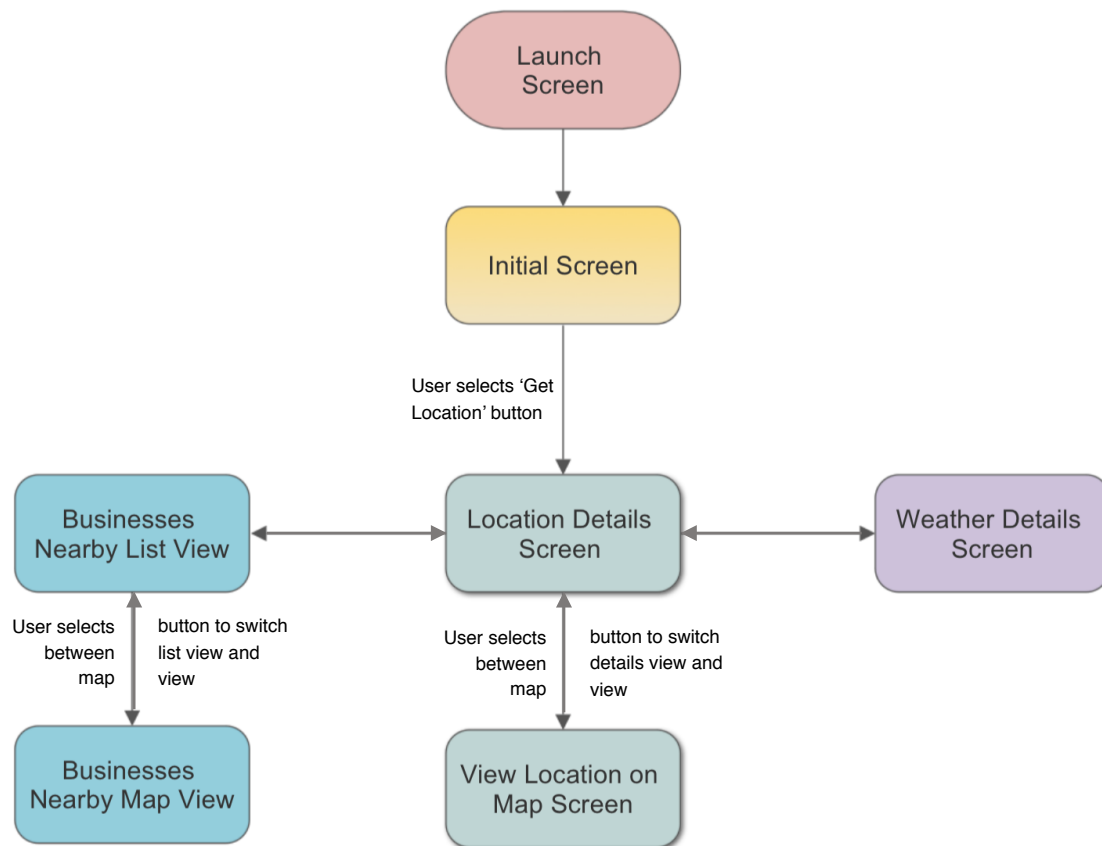
Weather Information Screen

This screen shot demonstrates how the application would look when viewing information about the current weather conditions.



Navigational Flow of Application

The navigational flow for the application is demonstrated in the diagram below. This diagram attempts to indicate the relationship between the various screens of the applications interface as shown in the previous user interface design section. Diagram below has been created using online tool, Smartdraw (Smart Draw, 2016).



There are 3 main navigational styles that can be adopted in an iOS application: hierarchical, flat and content driven. This project implemented a predominantly flat navigational structure, meaning users could navigate directly from one primary category to another because all primary categories were accessible from the main screen. This application had 3 primary categories: Location Details, Businesses and Weather Information. Essentially, users would be able to navigate directly to any of these 3 screens from any of the other screens they were currently viewing.

Application Use Cases

The next step of the design phase consisted of developing use cases for the application. Use cases assisted during the design phase of the project by helping me to understand what the necessary components of the application needed to be and how they needed to engage with each other successfully. This enabled the application to deliver the user requirements set out at the beginning of the design phase of this project. The following use cases were identified for this project:

Use Case ID: 1		Use Case Title: Start Application	
Use Case Description		Use case defines how the user starts up the application on their device.	
Actors		Tourist, Student or Person new to the area (Application Users).	
Triggers		User selects application.	
Pre-Conditions		The device has the application installed. The device is connected to the internet. The device has GPS technology and services enabled.	
Basic Flow		1. The user navigates to the application on their device. 2. The user selects the application to launch it on their device. 3. The application should then open and be running on the user's device.	
Exception Flow		If the application does not have access to the internet or does not have GPS enabled technology the application will still launch but will be unable to display information to the user.	
Post-Conditions		Initial screen is displayed to user.	

Use Case ID: 2		Use Case Title: Gather GPS Information	
Use Case Description		Use case defines how the GPS location information should be gathered.	
Actors		Tourist, Student or Person new to the area (Application Users).	
Triggers		User selects 'Get Location' button on initial screen.	
Pre-Conditions		The device is connected to GPS satellites.	
Basic Flow		1. The user selects the 'Get Location' button on the initial screen. 2. The application then detects the button has been pressed and gets user's current GPS information from the Location Manager. 3. The application then displays the user's location information on the Location Details screen of the application.	
Exception Flow		If there is no GPS connection to satellites the application will not gather any location information.	
Post-Conditions		User's latitude, longitude and address are displayed to them on the Location Details screen of the application.	

Use Case ID: 3		Use Case Title: Plot Location on Map
Use Case Description	Use case defines how the application displays the users location on a map.	
Actors	Tourist, Student or Person new to the area (Application Users).	
Triggers	User selects button to view their location in map view.	
Pre-Conditions	The application has gathered and stored the user's GPS information.	
Basic Flow	<ol style="list-style-type: none"> 1. The user selects the 'map view' button on the Location Details screen. 2. The application then detects the button has been pressed and plots the user's stored location on a map. 3. The application then displays the map to the user with a pin plotted in their location on the Map View screen. 	
Exception Flow	If there is no stored location information the application will not plot the user's location on a map.	
Post-Conditions	User's position will be pinned on a map of the area.	

Use Case ID: 4		Use Case Title: Gather Businesses Nearby
Use Case Description	Use case defines how the application displays gathers information on businesses nearby.	
Actors	Tourist, Student or Person new to the area (Application Users).	
Triggers	User selects button to view the businesses nearby.	
Pre-Conditions	The application has gathered and stored the user's GPS information. The device is connected to the internet.	
Basic Flow	<ol style="list-style-type: none"> 1. The user selects the 'Businesses tab' button on the navigation bar. 2. The application then detects the button has been pressed and sends a request to the API to retrieve information on the businesses near the user's stored location. 3. The application receives a response containing the requested data. 4. This response data is then processed and displayed in a list view on the Businesses in List View Screen. 	
Exception Flow	If the device is not connected to the internet the request cannot be sent to the API so no information on businesses will be displayed to the user.	
Post-Conditions	The user will be presented with a list of businesses nearby in a list view.	

Use Case ID: 5		Use Case Title: Show Businesses on Map	
Use Case Description	Use case defines how the application displays information on businesses nearby in a map view.		
Actors	Tourist, Student or Person new to the area (Application Users).		
Triggers	User selects button to view the businesses nearby in map view.		
Pre-Conditions	The application has gathered and stored the businesses located nearby.		
Basic Flow	<div><div>1.</div><div>The user selects the ‘View Businesses on Map View’ button in the top right.</div><div>2.</div><div>The application then detects the button has been pressed and gets the coordinates of each businesses location and plots each location on the map.</div><div>3.</div><div>The application then displays the businesses nearby on the map.</div></div>		
Exception Flow	If the device is has not stored a list of businesses nearby these cannot be plotted on a map so no businesses will be displayed on the map.		
Post-Conditions	The user will be presented with a map containing pins where the various businesses nearby are located.		

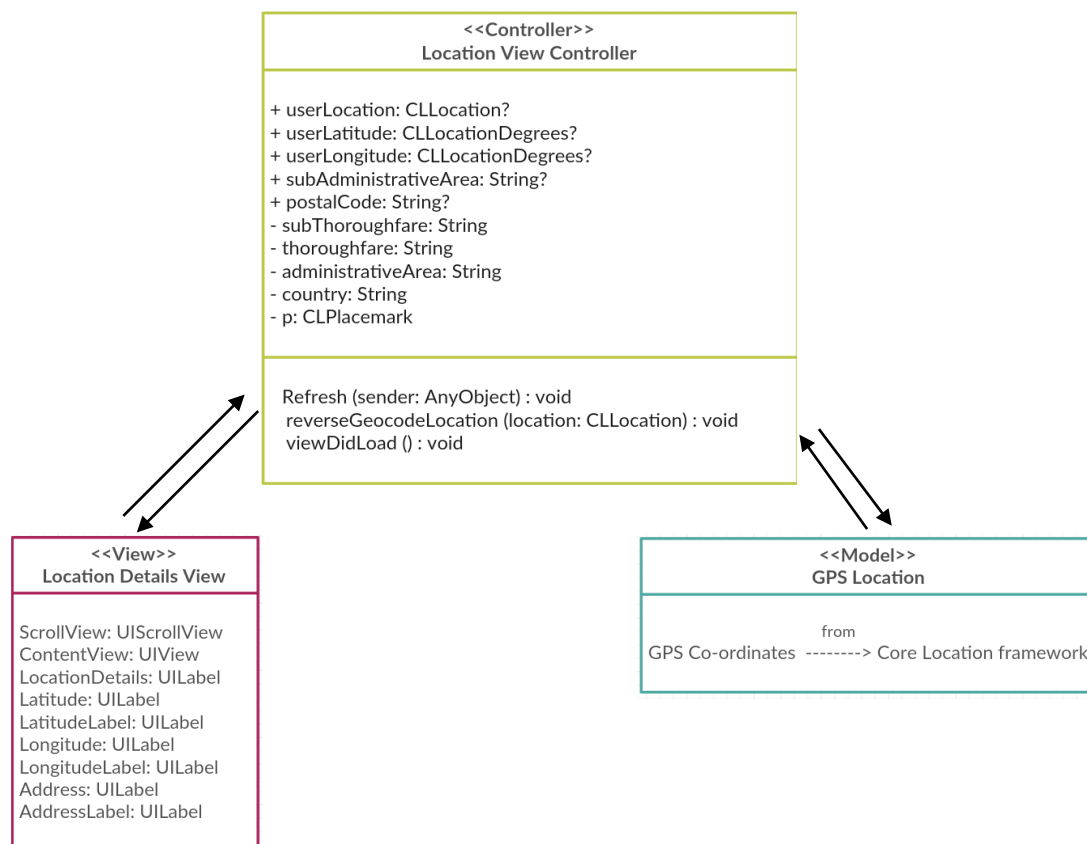
Use Case ID: 6	Use Case Title: Gather Weather Information
Use Case Description	Use case defines how the application gathers and displays information on weather conditions in their current location.
Actors	Tourist, Student or Person new to the area (Application Users).
Triggers	User selects button to view the weather information at their location.
Pre-Conditions	The application has gathered and stored the user’s location information. The device is connected to the internet.
Basic Flow	<ol style="list-style-type: none">1. The user selects the ‘View Weather Information’ button on navigation bar.2. The application then detects the button has been pressed and sends a request to the weather API to gather information on the current conditions at the user’s location.3. The application then receives a response containing the requested information on the conditions at the given location.4. This data is processed and displayed to the user on the Weather Screen.
Exception Flow	If the device is not connected to the internet the request cannot be sent to the API so no information on weather will be displayed to the user.
Post-Conditions	The user will be presented with a weather summary containing details such as the current temperature, a summary description of the conditions and the wind and current humidity in the area the user is currently located at.

System Architecture

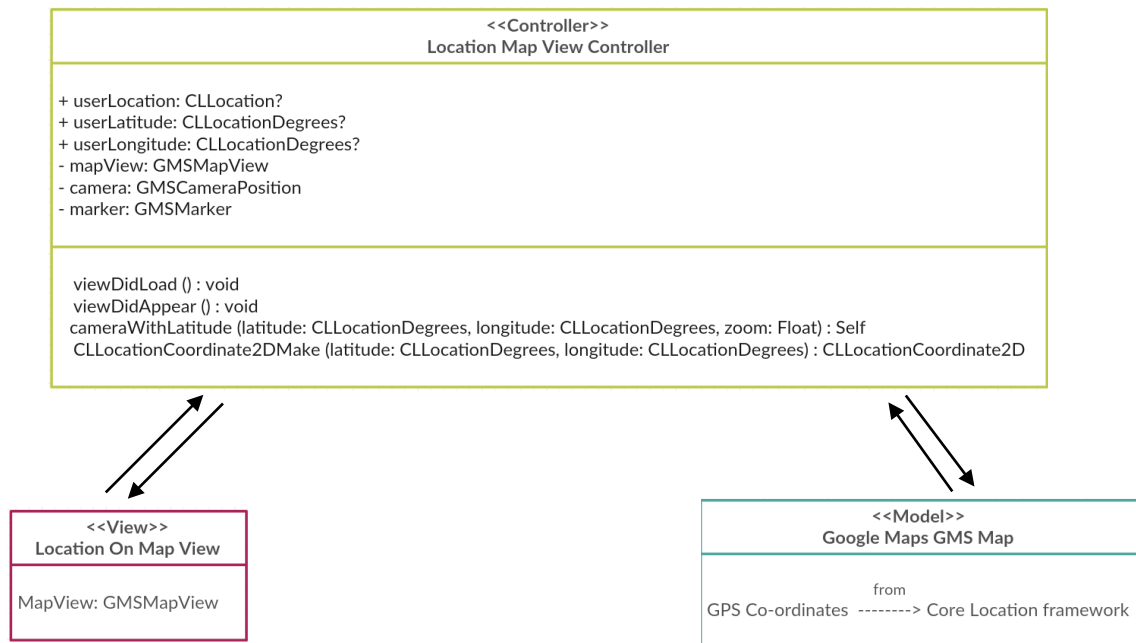
The system architecture of the iOS application developed in this project followed the Model-View-Controller design pattern (Apple, 2016b). The Model aspect corresponded to the data the application would use and display to its users such as information provided by the various APIs. The View part of the design pattern related to the user interface elements the user would be able to interact with such as buttons, labels and tables. The Controller component linked to the application's various view controllers, indicating how the functions and methods would transport data from the Model to the View.

A high level Model-View-Controller diagram corresponding to each of the application's 4 primary aspects of functionality has been included below in order to illustrate the relationships between the components of the system. These diagrams were created using an online tool called Creately (Creately, 2016).

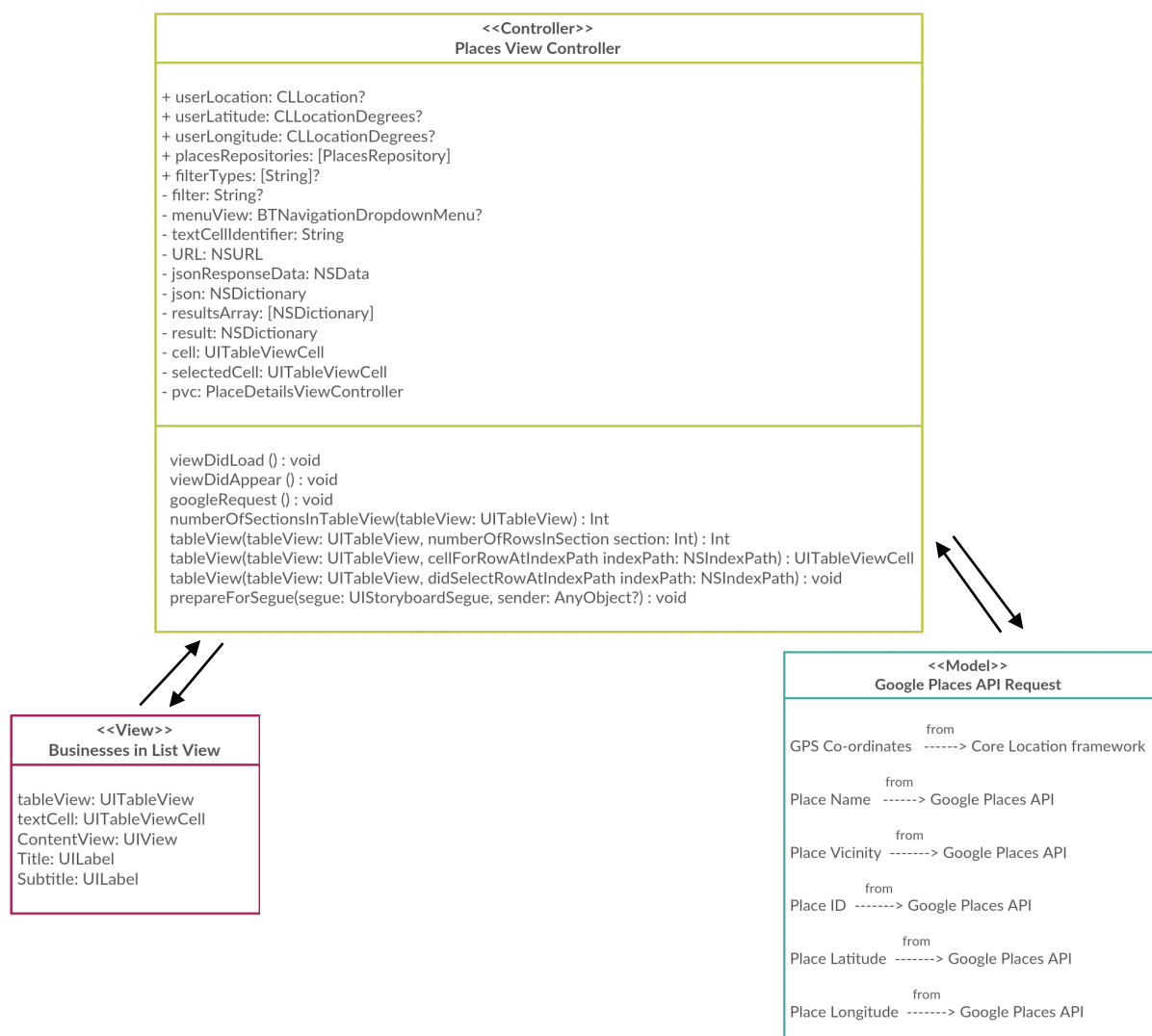
MVC for GPS Location



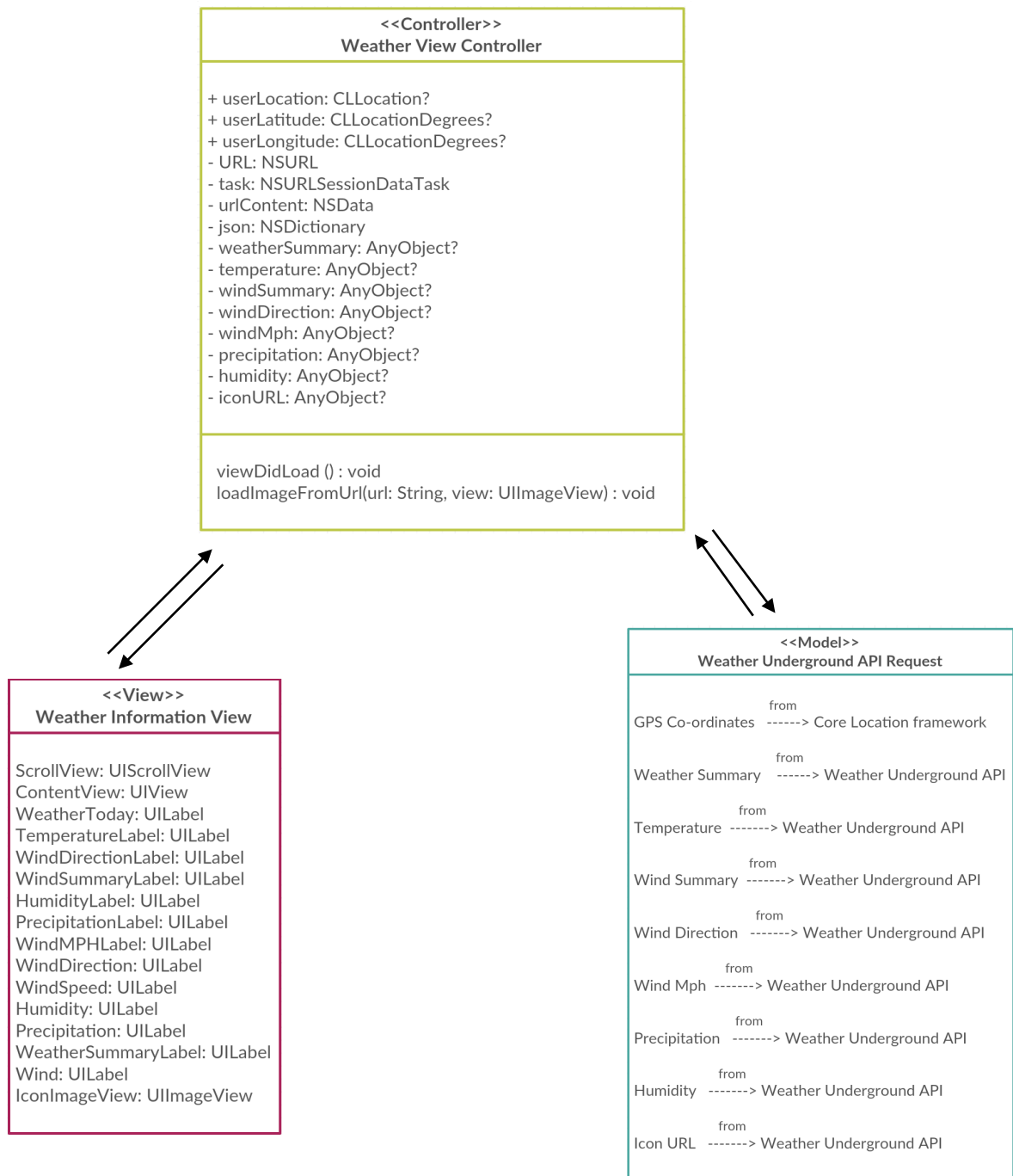
MVC for Google Maps GMS Map



MVC for Google Places API Request



MVC for Weather Underground API Request



Implementation

At the start of the implementation phase of this project, it was identified through the concepts outlined in the design section that there were 4 main aspects of functionality that needed to be included in the application for it to meet the identified requirements. The implementation of each of the 4 main aspects of functionality included in the application have been listed below and described in detail in the rest of this section of the report:

- GPS Location
- Google Maps GMS Map
- Google Places API Request
- Weather Underground API Request

Throughout the development of this project the Apple IDE (Integrated Developer Environment) Xcode version 7.3 was used to implement all aspects of the project.

GPS Location

The first and most crucial piece of functionality implemented for this project was the gathering of the device's GPS location co-ordinates. This was clearly the most important aspect of this project, as without this it would have been impossible to implement the other features specified in this project's aims and objectives. This was because all the other features of the application relied on having access to the device's location in order to provide the user with various types of useful location based information. Therefore, it was of the utmost importance this was implemented appropriately.

In order to get access to the device's GPS co-ordinates, Apple's Core Location Framework was imported into the "Link Binary With Libraries" section under the "Build Phases" tab of the application. I then created a key in the info.plist file called "NSLocationWhenInUseUsageDescription" with the description provided being shown to users when requesting access to location services (see error handling section for more details). After the application was granted access to the location services, an instance of the CLLocationManager class was created. This enabled the locationManager to begin monitoring the user's location after the startUpdatingLocation() function had been called. The locationManager's didUpdateLocations() function would then notify the application when the user's location changed. This enabled the application to monitor the device's location information such as its latitude and longitude.

This information was then displayed on the Locations View Controller, informing the user of their current latitude and longitude as well as their approximate address. The approximate address was gathered by the application using the Core Location Framework's CLGeocoder class. CLGeocoder's reverseGeocodeLocation() function was passed the user's current location as an argument. It then simply returned their approximate address which could then be displayed in a label on the user interface.

Google Maps GMS Map

The second piece of functionality in the application, plotting locations on a map, was somewhat straightforward to implement. It was decided to use a GMS map view to display locations on a map in the user interface as this was required under the terms of use for the Google Places API. The API stated that if business results obtained from the Google Places API were going to be plotted on a map, the map used had to be a GMS map view. Therefore, to keep the functionality consistent across the application it was decided to plot all locations not just the business locations on a GMS map view.

In order to plot the user's location on a map I first had to integrate the application with Google's Maps API. This integration was handled using the CocoaPods dependency manager (CocoaPods, 2016). Once a pod had been created and installed within the application, an instance of a GMS map view could then be created. Once a GMS map view had been added to the user interface, the camera position needed to be set to zoom the map into the user's current location. This was implemented using the GMSCameraPosition class' cameraWithLatitude() function:

```
let camera = GMSCameraPosition.cameraWithLatitude(userLatitude!, longitude: userLongitude!, zoom: 15)
self.mapView.camera = camera
```

Once the Google map was centralised around the user's current location, a simple GMS marker could then be added to the map using the following implementation:

```
let marker = GMSMarker()
marker.position = CLLocationCoordinate2DMake(userLatitude!, userLongitude!)
marker.title = "Current Location"
marker.snippet = "Lat: \(userLatitude!), Long: \(userLongitude!)"
marker.icon = GMSMarker.markerImageWithColor(UIColor.init(red: 188.0/255.0, green: 199.0/255.0,
blue: 71.0/255.0, alpha: 1.0))
marker.map = mapView
```

The same implementation could then be followed when plotting markers into the positions of the various businesses around the user on the businesses map view page of the application.

Google Places API Request

The Google Places API request functionality was more complex than the previous functionality employed in the application (Google, 2016c). This was because it involved sending an API request to the Google Places API then mining the data returned in the JSON response to allow it to be displayed suitably in a list table view. Sending the request and receiving a response was quite a simple task, an NSURL was simply sent to the Google Places API as shown below:

```
let URL = NSURL(string: "https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=\n(userLatitude!),\\(userLongitude!)&radius=1000&type=\n(filter!)&language=en&key=AIzaSyBFtAKuQm5siS9YzwgF1fj3mEwh91526YE")!
```

The JSON response received was then unwrapped into an NSDictionary using the NSJSONSerialization class. Scrutinising the NSDictionary representation of the data indicated that all the necessary information was contained in the key entitled “results”. These result items were then unwrapped into an array of dictionaries in order to access the data required.

```
if let jsonResponseData = NSData(contentsOfURL: URL) {
    if let json = (try? NSJSONSerialization.JSONObjectWithData(jsonResponseData, options: [])) as?
    NSDictionary {
        if let resultsArray = json["results"] as? [NSDictionary] {
            for result in resultsArray {
                placesRepositories.append(PlacesRepository(json: result))
            }
        }
    }
}
```

I then iterated over the the array of result items using a for loop, constructing a PlacesRepository object for each result and finally adding it to the placesRepositories array. A PlacesRepository class was constructed which was initialised inside the for loop, taking the NSDictionary as input then extracting the specified values that corresponded to the properties outlined in the PlacesRepository class. The results contained in the placesRepositories array then formed the list of businesses to be displayed on the businesses list table view controller.

Weather Underground API Request

Implementing the Weather Underground API request was somewhat easier after gathering some experience working with JSON responses in the previous functionality component (The Weather Company, LLC, 2016). This JSON API response was much simpler to unwrap, making it easier to extract the required elements of information to be displayed on the weather page of the application. The data contained on the weather page included details such as a weather summary string, temperature, wind direction, wind speed, relative humidity, precipitation today string and finally a weather summary icon. These snippets of information were unpacked from the JSON response as shown in the example below:

```
let temperature = json["current_observation"]!["temp_c"]!  
let windMph = json["current_observation"]!["wind_mph"]!  
let iconURL = json["current_observation"]!["icon_url"]!  where the json variable is an NSDictionary
```

With this API there was no need to create a separate WeatherRepository class as the JSON response received was much more straightforward and could be unpacked with ease to extract the useful sections of data returned by the API. These variables were then assigned to labels on the user interface, presenting this useful weather data to the user.

Tweets About the Location

As a result of the implementation phase of the project progressing well, I was able to also include an additional feature stipulated in my secondary objectives for the project, Tweets about the user's location. This was implemented by adding Fabric for iOS into the application to handle the Twitter integration in a simple and easy to manage manner (Fabric, 2016). To use Fabric for iOS in the application I simply had to download the Fabric for iOS Mac application and follow the simple instructions it provided. Firstly Fabric instructed me to add a "new run script phase" under the "build phases section of the project". Once this was completed I could then copy the Twitter Framework files provided in the Fabric Mac application and paste them into my Xcode project to allow them to be used. Finally adding the "import Fabric" and "import TwitterKit" statements into the App Delegate file of the project meant the integration was complete.

Displaying Tweets sent about the user's location was then very simple. The Tweets view controller was changed to a type of `TWTRTimelineViewController`, a pre-existing TwitterKit view controller. This meant when Tweets were searched and displayed on the view controller it would automatically format them appropriately in accordance with Twitter's guidelines. To perform a search, a connection to the Twitter client needed to be instantiated initially. This was done by creating a `TWTRAPIClient()` which allowed requests or searches to be sent to Twitter after this connection had been made (Twitter, 2016a). The search was then completed using the `TWTRSearchTimelineDataSource` by providing it with a simple search query of the current area the user was located in (Twitter, 2016b). The application would then return a list of Tweets sent containing that area name, providing users with an interesting insight into the opinions and attitudes of the people around them.

Error Handling

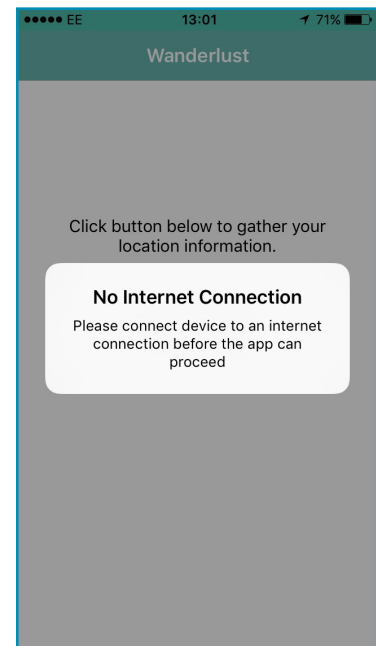
The application included various error handling techniques to ensure the user's device had all the requirements necessary to use the mobile application. The two main requirements the application needed to operate correctly were internet access and authorisation to use the device's location services. Without these two crucial requirements, the functionality of the application would not work as expected, causing the application to crash or react unexpectedly. As a result, it was important that suitable error handling was implemented to deal with the lack of these important system requirements. In keeping with the error handling style iOS users were accustomed to, it was decided the most user friendly way to handle these losses of location privileges or drop outs of internet connection, were to use `UIAlertControllers`. A `UIAlertController` displayed an alert message to the user when certain conditions occurred throughout the application.

Lack of Internet Connection

Firstly, connection to the internet was an essential requirement of this application, and without it requests could not be sent to the various APIs. This meant users would be presented with no useful location based data, rendering the application almost futile. In order to ensure the application could monitor the device's connectivity at all times, a reachability listener was set up in the App Delegate file of the application. This listener used an imported Apple Reachability class to continuously check the device's connectivity status (Apple, 2015d). If the status changed to `NotReachable` at any point during the running of the application, a `UIAlert` was displayed to the user informing them of their lack of connectivity and asking them to reconnect to the internet before trying to continue using the app.

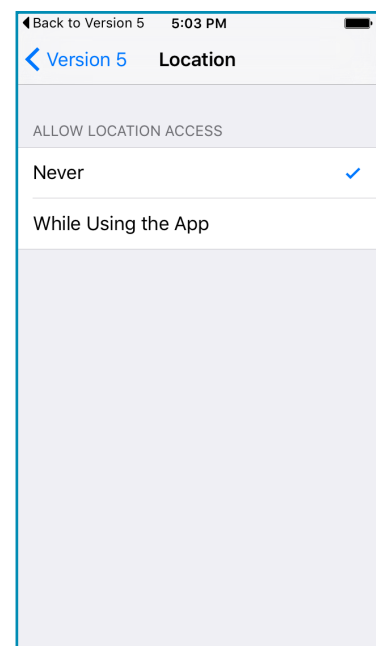
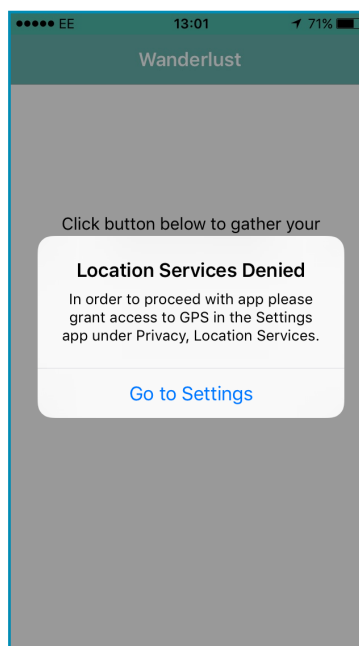
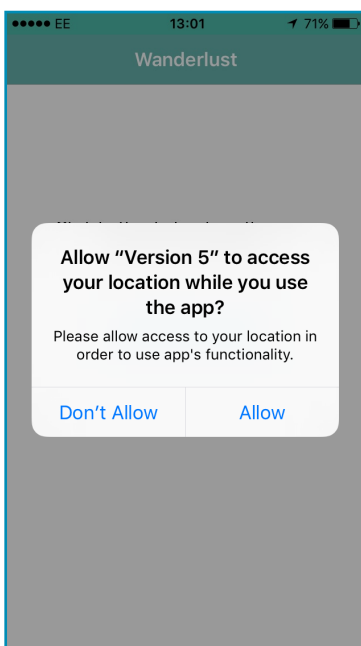
The UIAlert would then remain on the screen until connectivity was restored. This alert would therefore prevent users from continuing to use the application until a stable internet connection was maintained.

```
if (remoteHostStatus.rawValue == NotReachable.rawValue) {  
    alertController = UIAlertController(title: "No Internet Connection",  
message: "Please connect device to an internet connection before the app  
can proceed", preferredStyle: .Alert)  
}
```



Denied Access to Location Services

The second essential requirement of this application was that location services were enabled by users to allow the application to access the device's location. If users denied access to location services the application could not function correctly. As a result a UIAlert would be displayed to the user informing them that they needed to change their location privacy settings for the application, in the Settings Application. A link to the Settings application of their device was then provided, allowing them to quickly rectify this issue and continue using the application's excellent functionality (Natasha, 2015).



Testing

In order to ensure the application was thoroughly tested, 3 different kinds of testing were completed. Firstly, initial lab tests were carried out on the application using Xcode's built in Simulator. This enabled the first stage of testing to be carried out ensuring the application worked as expected under controlled conditions before moving onto the next test phase. The second phase of testing was field testing, where the application was run on an iOS device in various locations to ensure it functioned as expected. Finally the application was tested by various users in order to ensure the application was fit for purpose and easy to use. The user testing phase enabled me to evaluate the application's usability as well as to assess how intuitive an experience it provided for users.

Lab Testing

In order to test the application in a lab based setting, 6 unit tests were created to test each aspect of the program's functionality. Each test was also run 5 times with a different location being used each time to ensure the functionality worked appropriately regardless of location used. For each test I used the same 5 locations, these were Fairwood Road in Fairwater, The Parade in Cardiff City Centre, Red Dragon Centre in Cardiff Bay, Wenvoe in The Vale of Glamorgan, and Cosmeston Lake in Penarth. The following test cases indicate exactly which steps were taken throughout the test and the outcome received by each action. The test cases below only include the test being run on one location. The additional 4 locations of each test can be found in the appendix of this report.

Test Case ID: 1.1		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) into the Simulator		
3	Select ‘Get GPS Location’ button		

Test Case ID: 1.1		Test Case Title: Gather GPS Information	
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:		Pass	

Test Case ID: 2.1		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Map’ button in top right of Location Details View Controller		
5	Check if the device’s location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:	Pass		

Test Case ID: 3.1		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
6	Select ‘All’ dropdown menu and choose a category to filter businesses		‘Petrol Station’ category selected from filter
7	Check businesses now provided in list table view comply with chosen category	Pass	1 petrol station is displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 4.1		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		

Test Case ID: 4.1		Test Case Title: Gather Businesses Map View	
2	Program Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) into the Simulator		
3	Select 'Get GPS Location' button		
4	Select 'Businesses' icon in second space on tab bar located at the bottom of the view		
5	Select 'Map' button in the top right corner of the view		
6	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user's current location
7	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:		Pass	

Test Case ID: 5.1		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) into the Simulator		
3	Select ‘Get GPS Location’ button		

Test Case ID: 5.1		Test Case Title: Gather Weather Information	
4	Select 'Weather' icon in third space on tab bar located at the bottom of the view		
5	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:		Pass	

Test Case ID: 6.1		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
5	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Cardiff location
Additional Notes:			
Test Result:	Pass		

Field Testing

As part of the field testing for my application I physically took the device to each of the 5 locations used during the lab work testing. The same 6 test cases as used in the lab testing were again run in each location to check if the outcome was the same when tested in a real world environment. The following test cases indicate exactly which steps were taken throughout the test and the outcome received by each action. The test cases below only include the test being run on one location. The additional 4 locations of each test can be found in the appendix of this report.

Test Case ID: 7.1		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application physically on device		
3	Select ‘Get GPS Location’ button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 8.1		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		

Test Case ID: 8.1		Test Case Title: Plot Location on Map	
2	Select 'Get GPS Location' button		
3	Select 'Map' button in top right of Location Details View Controller		
4	Check if the device's location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:		Pass	

Test Case ID: 9.1		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none"> Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) 		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
5	Select ‘All’ dropdown menu and choose a category to filter businesses		‘Petrol Station’ category selected from filter
6	Check businesses now provided in list table view comply with chosen category	Pass	1 petrol station is displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 10.1		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Select ‘Map’ button in the top right corner of the view		
5	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location
6	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 11.1		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater)		
Test Step:	Step Description:	Result:	Notes:

Test Case ID: 11.1		Test Case Title: Gather Weather Information	
1	Run 'Wanderlust' application in Simulator		
2	Select 'Get GPS Location' button		
3	Select 'Weather' icon in third space on tab bar located at the bottom of the view		
4	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:		Pass	

Test Case ID: 12.1		Test Case Title: Gather Tweets About Location	
Test Case Description:		The purpose of this test case is to check the application can correctly acquire tweets about the user's current GPS location and display this to the user in a real world situation.	
Test Case Parameters:		In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none"> Latitude: 51.49826 , Longitude: -3.23881 (Fairwood Road, Fairwater) 	
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Select 'Get GPS Location' button		
3	Select 'Tweets' icon in fourth space on tab bar located at the bottom of the view		
4	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user's Cardiff location
Additional Notes:			
Test Result:		Pass	

User Testing

Approach

In order to gather information on the opinions of users for the application, the final step of the testing phase was to complete user testing. The first part of this process was to create an information page to provide the test participants with. This gave them an insight into the aims of the project, the purpose of this type of testing, and how the test would be carried out, allowing them to become aware of the context of the application before testing its features. I then set about creating a simple questionnaire users could complete once they had used the application, in order to gather data on their opinions. This questionnaire can be seen in Appendix 4 of this report. A total of 8 participants completed the user test, with the test being completed as follows:

- I installed the mobile application onto my device.
- I provided the test participant with the device.
- I gave the test participant the information sheet and asked them to take a few minutes to read through it and familiarise themselves with the project.
- I told them they would have 5 minutes to use the application however they wished.
- Once they finished using the application, I asked them to complete the questionnaire as honestly as possible.

Results

The user testing was completed successfully, with no test participants discovering any faults during the course of the test. This is a very good indicator of how robust the application is, as even during use in a non-structured way the application was able to run as smoothly as during controlled lab testing.

Appendix 5 contains the completed feedback questionnaires received from each test participant if further detail is required. I will now however provide a summary of the feedback received. The responses provided from the user feedback were very positive overall. Test participants believed the application to be extremely easy to use with 75% giving the application a score of 10 on the usability scale. The other 25% gave the application a score of 9 for usability, indicating all participants believe the app to be significantly user friendly. This was echoed during their use of the device as no users appeared to become frustrated whilst using the application. Also all users agreed that they would certainly use the application if it were available commercially, with 3 participants even asking for a copy of the application to put onto their devices, indicating their positive experience and satisfaction with its features.

When users were asked which aspects of the application they liked, responses included that providing access to Google directions was very useful for users as well as that “the Twitter feature provided interesting insight from other people in the local area”. Users also enjoyed the simplicity of the user experience, stating that “the app is really user friendly”. They also liked “the fact that the app includes both factual information about the businesses, such as opening times and description, but also that it has personal opinions in the form of reviews”.

Users also provided interesting constructive criticisms and suggestions of extra features that could be included in the application. Responses included that it could be useful to have a search bar feature, as well as simple categorisations for the businesses shown in the nearby area. They also stated it could be useful to allow the user to input the radius of the search instead of the fixed 1km radius the application currently has. One very common response was that the dropdown menu filter on the businesses list could be made more obvious to users. One user also stated that it may be useful for the weather section to contain information on what the weather will be in the coming hours, as opposed to just providing details of the current conditions.

Users also had some interesting novel ideas of extra features that could be included in a future version of the application such as “a Facebook check-in location could be added to share locations and integrate the app with social media”. Another user suggested that “the app could be improved by giving the user the flexibility to change your location, rather than being restricted to your current location, as this could be useful for planning a holiday or going to a new city”.

Results and Evaluation

In the results and evaluation section of this report, a critical appraisal has been completed indicating to what extent the research question of this project has been answered. The research question was: would it be feasible to create an application that takes the user's location as input and processes it to provide them with useful information about their surroundings, given they are new to the area.

Main Findings of the Study

In order to extensively evaluate the success of this project in terms of answering its research question, it was important to consider two aspects of the project. Firstly it was critical to assess to what extent the mobile application actually worked. In order to ensure each feature of the application worked correctly, 2 types of functional testing were carried out: lab testing and field testing. Based on the results of these extensive tests documented and outlined in the previous section of this report, it was found that all functionality of the application worked correctly. This indicated that it would be feasible from a technological perspective to develop and implement an application with the features described in this report. This was because to some extent the application developed in this project has proven these capabilities are possible, as the test results demonstrated the application's competence at meeting all of the project's primary, and one of its secondary objectives.

The second aspect to consider was to understand whether users would want to use this application and whether it would suit their needs if they were a tourist visiting a new area. To assess this part of the research question, user testing was completed to gather opinions from potential users on whether they would use this application if it were commercially available. Responses from the user testing can be seen in the previous section of this report; however, an overwhelming positive response indicated that 100% of users tested said they would use this application if it were available. This seemed to indicate that there is indeed a gap in the marketplace for such an application and that it would be beneficial to pursue the project further in future.

Therefore the results of this study found that it would indeed be feasible to develop an application that takes the user's location as input and processes it to provide them with useful information about their surroundings, given that they are new to the area. The results also showed there is indeed demand for such an application because 100% of test participants stated they would use this application if available.

Alternative Explanations for Project Outcome

There were no alternative explanations in terms of how well the mobile application's functionality worked by the end of the project. The lab testing and field testing carried out was comprehensive and thorough, indicating that each aspect of the application functioned as expected. The test cases constructed were pass or fail meaning there could be no misinterpretation of the results of the functional testing. There were potential alternative explanations, however, on the outcome of the user testing for the project: only a small number of test participants were used to evaluate whether there is a demand for such an application in the current application marketplace.

The feedback received in this project indicated that there most certainly is demand for this type of application, with all test participants agreeing that they would use this application if it were commercially available. In order to gain a high degree of accuracy in answering this question and capturing the opinion of the general public as a whole, this sample would need to be increased to perhaps 100 people. This would ensure the outcome received really is the opinion of the majority, as opposed to just the group selected to complete this project's user testing. The user testing should also contain participants with a range of ages and technology capabilities in order to correctly assess the application's usability. The test participants used in this project were all of a similar age and possessed similar technological capabilities meaning they may have given the application a higher usability score than those with less experience using mobile devices.

Furthermore, the test participants utilised in the user testing were my family members and friends. Even though I instructed honest feedback, it is possible that they gave slightly higher scores and less critical feedback because they were biased in my favour.

Limitations of the Project

There were several limitations associated with this project; the main limitation was that the Google Places API response only returns 20 results in the businesses nearby list. Users would require a complete list of all businesses in the 1km radius specified, so while only showing 20 results was acceptable for this research project, if the application were to be launched in future, this issue would need to be rectified. Another issue to be considered if the application were available on the AppStore was the limited number of Google Places and Weather Underground API calls that can be made, which are 150,000 and 500 respectively. If the application were to become highly successful, this would not be sufficient, so a paid subscription with higher API call limits would need to be purchased.

A personal limitation which impacted the project was my lack of digital design skills and unfamiliarity using design software such as Photoshop. During the Design phase of the project, I indicated several button icons I wanted to design and include in the final version of the application. I was able to implement icons for the essential user interface elements, such as the tab bar icons, but could not include graphical icons for the 'Get GPS location', 'Refresh' and 'Show Map' buttons. Due to my lack of experience, even designing a simple icon took a significant amount of time, so in the 12 short weeks available for the project, the remaining icon designs were a low priority in comparison to other features that needed to be implemented.

Another significant impediment to the project was its strict timescale of only 12 weeks. Keeping to such a narrow time frame meant that only the functionality stated in the aims and objectives section could be included, with little scope to pursue anything extra. However, due to good time management throughout the project, I was able to implement one of the secondary objectives of displaying a Twitter feed, containing Tweets about the user's location. If the time scale was more flexible, the user experience could have been enhanced to include some of the features stated in the following Future Work section.

Future Work

Despite the overall success of the project and its many accomplishments in terms of achieving most of the project's aims and objectives, there were still a number of areas in the application which I would improve upon if development were to continue in the future.

There were a number of existing aspects which I would spend time improving including the weather page. The weather functionality in the mobile application only displayed basic weather conditions to the user such as temperature, humidity and wind speed. However, in future it could be useful for the application to also show users the weather conditions for the next few hours in addition to current conditions. This would enable them to plan their activities in a more informed manner. It could also be beneficial to the user if more business categories were included in the dropdown menu on the businesses list page. This would provide users with more control to find information on a larger range of business types. Another feature which would give users more control to find the information most relevant to them would be to add a search field, allowing users to search for a specific business. This was suggested by a few users during the user testing feedback so seemed to be a feature many potential users would like to see included.

Test participants also suggested it would be helpful to allow users to edit the radius of the businesses returned in the businesses nearby feature of the application, as opposed to the fixed 1km radius that had been implemented. Finally, future improvements on the project would also need to address the limitations of using the Google Places API. More research would need to be done to investigate whether it is possible to return more than 20 results of businesses nearby using the Google Places API. Alternatively, an API with similar features would need to be identified and implemented which allows developers to return a greater number of query results for businesses in the area.

There is also a great deal of new functionality which could be added to the application as future work. Many of the following ideas were suggested by users during the testing phase of the project, some of which were excellent and novel which I think the application could have benefitted from employing:

- Add the user's current distance from each place listed in the business list view.
- Add feature so users can input a location, not just use their current location as this could allow flexibility to plan weekends away.
- Application could be available in different languages.
- A Facebook check-in location feature could be added to integrate social media.

Conclusions

In summary, it is reasonable to conclude that it is feasible to develop a mobile application which takes the user's location as input and provides them with a wealth of useful information pertaining to that location, given they are new to the area. The results of the testing phase indicate it is possible to include features in the application such as displaying Tweets sent about their location, presenting weather information and businesses in the surrounding area. From a technological perspective, it can be verified that this combination of features could be included in a mobile application and function appropriately, indicated through every feature passing its functional tests in the testing phase of this project. It would also be possible to include many other features that utilise location data if the project were to continue in the future, such as displaying Instagram photos of the user's location.

The responses from the user testing were incredibly positive with every user stating they would use the application if it were commercially available. This indicates there is demand for such an application and that users would be interested to use the application if further work was completed on the project. Despite the fact that the functional tests completed were thorough and that agreeable feedback was received during the testing phase, additional testing would need to be completed before committing more time to launching the project in future. This is because only a small number of test participants were used. As a result the tests would need to be completed on a larger population before being able to reliably conclude that the application would be well received.

Reflections

Over the course of the project I have developed various new skills as well as improving upon existing knowledge and expertise in a number of areas. Firstly, at the start of the project I had no knowledge or experience of what was required to design, implement and test a mobile application. I had never created a mobile application before which proved to be a steep learning curve in the months to come. Throughout the project copious amounts of research was carried out to understand how to approach the project and implement each feature required to meet the project's objectives. I am extremely glad I decided to pursue a project in the realm of mobile development as it has allowed me to challenge myself in a completely new area of technology I had not previously explored.

The project has also allowed me to develop significant knowledge and understanding of the Swift programming language used to create iOS based applications. Completing countless tutorials and reading endless pages of documentation has enabled me to grasp the very important concepts and challenges working in the Swift language brings. A feature of the Swift language I had not yet been exposed to as a developer in any other language was the concept of optionals. However after reading Apple's developer documentation, I was able to utilise this feature countless times throughout my application in order to implement different areas of the application's functionality. Working in an iOS environment also allowed me to learn the basics of UI design for iOS applications through following Apple's iOS Human Interface Guidelines (Apple, 2016a). Creating iOS interfaces made me realise the importance of letting the content dominate each view and using plenty of negative space to provide clarity for the user. I learnt that using too many colours or logos could overwhelm the little space available on mobile screens, causing users' attention to be distracted from the heart of the application: its content.

I feel pursuing the project has enabled me to become a competent iOS developer with the basic knowledge needed to create applications successfully. With Apple's huge user base I believe these skills will be invaluable in the future with many businesses creating mobile application versions of their websites in order to appeal to the masses. Therefore, these skills will be sought out by many employers, giving me a significant advantage over applicants who do not possess the knowledge of iOS development and the Swift programming language I have obtained.

The project has also taught me the importance of user feedback and testing when working on a project such as this. Users often have very interesting ideas and insights into what features could set the application above the rest and make it desirable to users. Therefore, in future I have learnt that it may be beneficial to gather user opinions and suggestions through a survey completed at the beginning of the project. This way the best and most feasible ideas could be included in the first revision of the project, as opposed to including them in the future work proposals.

Finally an important non-technical lesson learnt through completing this project has been the significance of strict time management. It is crucial that a detailed schedule is made and followed as closely as possible if the project is to be successful in achieving its aims. I have also learnt that it is necessary to consider that additional time may be required to rectify unforeseen problems over the course of the project. Therefore, it is important, at the time of schedule creation, to evaluate the severity and likelihood of risks to the project, adding an appropriate amount of time to deal with these situations into the plan. This should prevent issues from impacting the success of the project. It is also important that the plan's time scales are amended if issues do arise, in order to ensure that the plan can still be followed and the project aims are completed on time. Time management is an excellent transferable skill that will be of great use to me in my future career when developing or managing any type of project.

Bibliography

Adobe Systems Incorporated. (2016). *Photoshop: start your free trial*. Available: <https://creative.adobe.com/products/download/photoshop?promoid=GXUTM>. Last accessed 29th March 2016.

Apple (a). (2016). *Designing for iOS*. Available: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>. Last accessed 11th Feb 2016.

Apple (b). (2015). *Model-View-Controller*. Available: <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>. Last accessed 25th April 2016.

Apple (c). (2016). *NSNotificationCenter*. Available: https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSNotificationCenter_Class/index.html. Last accessed 19th March 2016.

Apple (d). (2015). *Reachability*. Available: <https://developer.apple.com/library/ios/samplecode/Reachability/Introduction/Intro.html>. Last accessed 16th April 2016.

Apple (e). (2016). *Swift. A modern programming language that is safe, fast, and interactive*. Available: <https://developer.apple.com/swift/>. Last accessed 24th Jan 2016.

Apple (f). (2016). *Template Icons*. Available: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/BarIcons.html#/apple_ref/doc/uid/TP40006556-CH21-SW1. Last accessed 28th Feb 2016.

Bryan Wilde. (2013). *iOS vs Android vs Windows: The Battle of the Mobile OS*. Available: <https://blog.udemy.com/ios-vs-android-vs-windows/>. Last accessed 23rd Jan 2016.

Chris Ching. (2014). *Starting an iOS Tab Bar App with UITabBarController*. Available: <http://codewithchris.com/ios-tab-bar-app/>. Last accessed 16th Feb 2016.

CocoaPods. (2016). *What is CocoaPods?*. Available: CocoaPods.org. Last accessed 17th Feb 2016.

Creately. (2016). -. Available: https://creately.com/app/?tempID=gc7qvpsj1&login_type=demo#. Last accessed 22nd April 2016.

Dark Government. (2012). *How GSM Mobile Phone Tracking Works*. Available: <http://www.darkgovernment.com/news/how-gsm-mobile-phone-tracking-works/>. Last accessed 20th Jan 2016.

The Dark Sky Company, LLC. (2016). *The Dark Sky Forecast API*. Available: <https://developer.forecast.io>. Last accessed 5th Feb 2016.

Fabric. (2016). *The Tools You Need to Build the Best Apps*. Available: <https://get.fabric.io/ios?locale=en-us>. Last accessed 7th March 2016.

Flying Code Ltd. (2016). *Around Me*. Available: <https://itunes.apple.com/gb/app/aroundme/id290051590?mt=8>. Last accessed 26th Jan 2016.

Foursquare. (2016). *Search for Places in an Area*. Available: <https://developer.foursquare.com/start/search>. Last accessed 5th Feb 2016.

Garmin. (2016). *What is GPS?* Available: <http://www8.garmin.com/aboutGPS/>. Last accessed 20th Jan 2016.

Google (a). (2016). *Get Started with the Places API for iOS*. Available: <https://developers.google.com/places/ios-api/start>. Last accessed 17th Feb 2016.

Google (b). (2016). *Place Details*. Available: <https://developers.google.com/places/web-service/details>. Last accessed 1st March 2016.

Google (c). (2016). *Place Search*. Available: <https://developers.google.com/places/web-service/search>. Last accessed 24th Feb 2016.

Jakob Nielsen. (1995). *10 Usability Heuristics for User Interface Design*. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Last accessed 9th Feb 2016.

Jared Davidson. (2014). *How to Pass Data from View Controller*. Available: <https://www.youtube.com/watch?v=guSYMPaXLaw>. Last accessed 11th Feb 2016.

Jim Bowes. (2014). *Agile vs Waterfall: Comparing project management methods*. Available: <https://manifesto.co.uk/agile-vs-waterfall-comparing-project-management-methodologies/>. Last accessed 3rd May 2016.

Joshua Greene . (2015). *How to Use CocoaPods with Swift*. Available: <http://www.raywenderlich.com/97014/use-cocoapods-with-swift>. Last accessed 18th Feb 2016.

Juan Carlos Sánchez. (2015). *Swift 2 examples – #1 Loading an image from a URL*. Available: <https://softwarejuancarlos.com/2015/11/27/swift-2-examples-loading-an-image-from-a-url/>. Last accessed 25th Feb 2016.

MediaBUZZ Pte Ltd. (2012). *Location-based service (LBS): Categories of Methods, Techniques and Technologies*. Available: <http://www.mediabuzz.com.sg/buzzwords-march-12/location-based-service-lbs-categories-of-methods-techniques-and-technologies>. Last accessed 20th Jan 2016.

Natasha. (2015). *iOS: You're Doing Settings Wrong*. Available: <https://www.natashatherobot.com/ios-taking-the-user-to-settings/>. Last accessed 19th April 2016.

Olivier Cuenot. (2016). *Json Parser Online*. Available: <http://json.parser.online.fr>. Last accessed 24th Feb 2016.

Paul Solt. (2015). *Swift vs. Objective-C: 10 reasons the future favors Swift*. Available: <http://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html>. Last accessed 22nd Jan 2016.

Priya Viswanathan. (2015). *The Pros and Cons of Native Apps and Mobile Web Apps*. Available: <http://mobiledevices.about.com/od/additionalresources/qt/The-Pros-And-Cons-Of-Native-Apps-And-Mobile-Web-Apps.htm>. Last accessed 27th April 2015.

Rob Percival. (2016). *App: Location Aware*. Available: <https://www.udemy.com/the-complete-ios-9-developer-course/learn/v4/t/lecture/3193432?start=1190>. Last accessed 15th Feb 2016.

Ron Kliffer. (2015). *Google Maps iOS SDK Tutorial: Getting Started*. Available: <http://www.raywenderlich.com/109888/google-maps-ios-sdk-tutorial>. Last accessed 20th Feb 2016.

Simon NG. (2012). *iOS Programming 101: How To Customize Tab Bar Background and Appearance*. Available: <http://www.appcoda.com/ios-programming-how-to-customize-tab-bar-background-appearance/>. Last accessed 30th April 2016.

Smart Draw. (2016). - . Available: <https://cloud.smartdraw.com>. Last accessed 12th Feb 2016.

Tech Target. (2011). *Web application (Web app)*. Available: <http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>. Last accessed 27th April 2016.

Twitter (a). (2016). *Access Twitter's REST API*. Available: <https://docs.fabric.io/ios/twitter/access-rest-api.html#construct-a-twitter-api-client>. Last accessed 7th March 2016.

Twitter (b). (2016). *Show Timelines*. Available: <https://docs.fabric.io/ios/twitter/show-timelines.html>. Last accessed 7th March 2016.

Visit Britain. (2016). *2016 forecast*. Available: <https://www.visitbritain.org/2016-forecast>. Last accessed 16th Jan 2016.

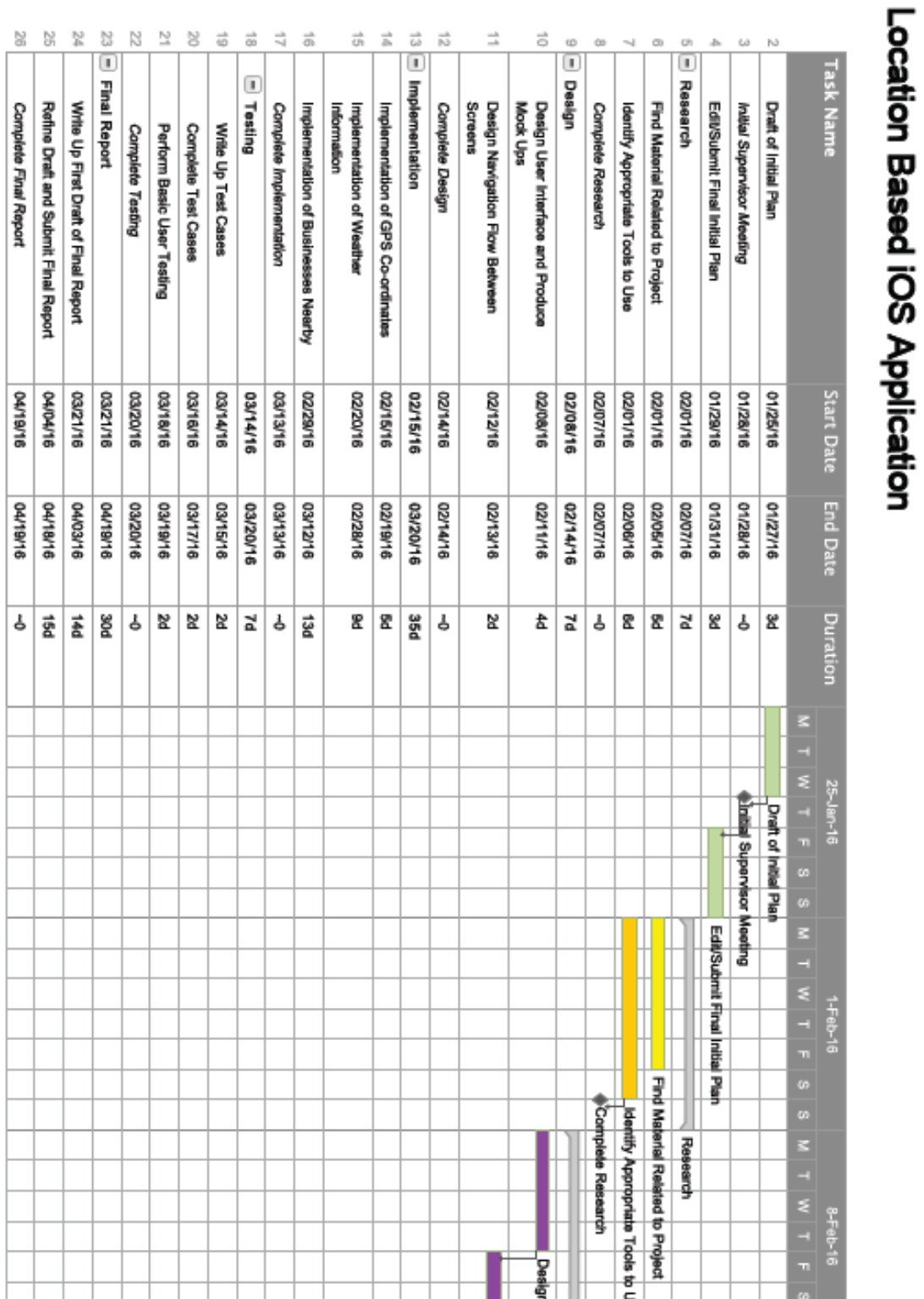
The Weather Company, LLC. (2016). *Weather API: Introduction*. Available: <https://www.wunderground.com/weather/api/d/docs?d=data/index&MR=1>. Last accessed 2nd Feb 2016.

World Weather Online. (2016). *Weather API*. Available: <http://developer.worldweatheronline.com/api/>. Last accessed 3rd Feb 2016.

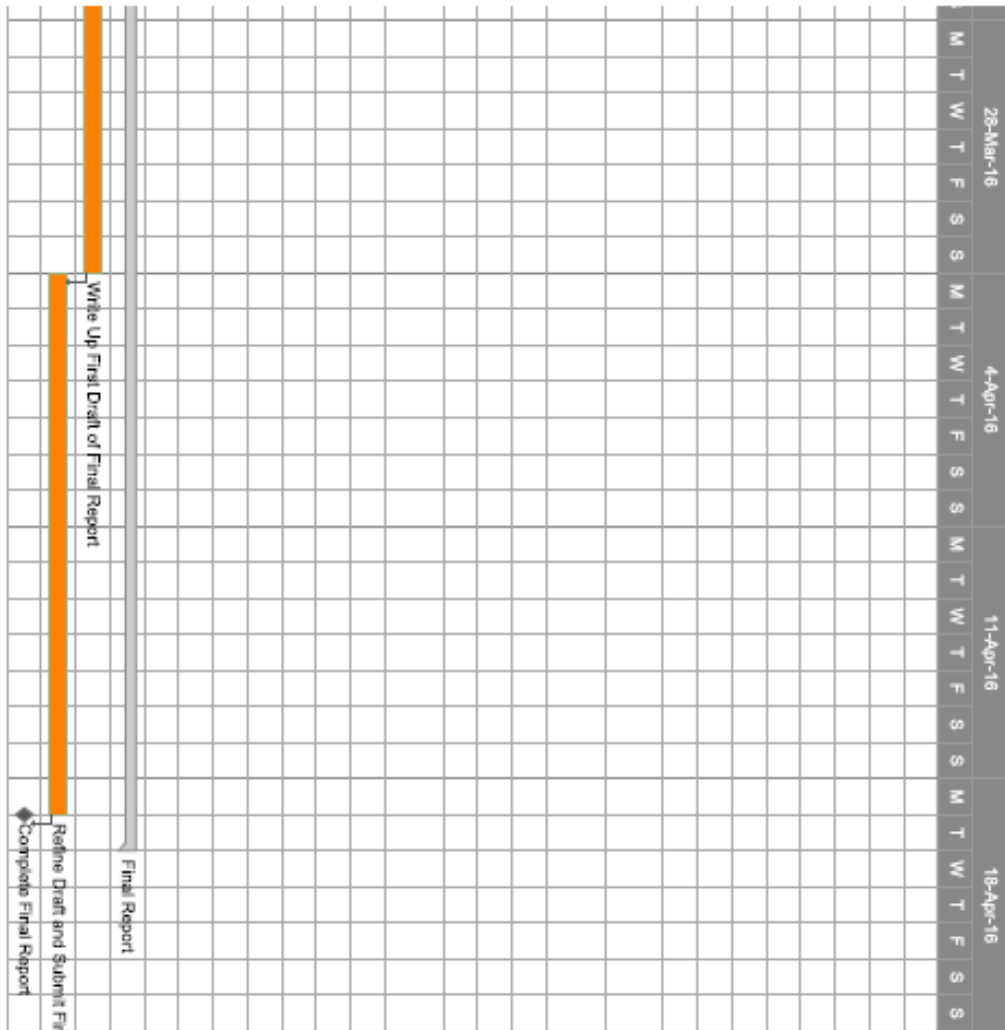
XLabz Technologies Pvt. Ltd. (2016). *Find Near Me for iPhone - Find nearby ATM's, Taxi, Hotels & everything around you*. Available: <https://itunes.apple.com/gb/app/find-near-me-for-iphone-find/id353369769?mt=8>. Last accessed 26th Jan 2016.

Appendices

Appendix 1: Project Schedule Gantt Chart







Appendix 2: Lab Test Cases

Lab Test Cases 1.2 - 1.5

Test Case ID: 1.2		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 1.3		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay) into the Simulator		
3	Select ‘Get GPS Location’ button		

Test Case ID: 1.3		Test Case Title: Gather GPS Information	
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:		Pass	

Test Case ID: 1.4		Test Case Title: Gather GPS Information	
Test Case Description:		The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device's GPS location information.	
Test Case Parameters:		In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none"> Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) 	
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Program Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) into the Simulator		
3	Select 'Get GPS Location' button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:		Pass	

Test Case ID: 1.5		Test Case Title: Gather GPS Information	
Test Case Description:		The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device's GPS location information.	
Test Case Parameters:		In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none"> Latitude: 51.44715 , Longitude: -3.26644 (Cosmeston Lake, Penarth) 	

Test Case ID: 1.5		Test Case Title: Gather GPS Information	
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Program Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth) into the Simulator		
3	Select 'Get GPS Location' button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:	When observing address data for Cosmeston Lake, Penarth, minimal address details are shown. Application included only the county, postcode and country (no street name etc) this is because the location is a park so will not have a street name.		
Test Result:	Pass		

Lab Test Cases 2.2 - 2.5

Test Case ID: 2.2		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">• Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Map’ button in top right of Location Details View Controller		
5	Check if the device’s location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude

Test Case ID: 2.2		Test Case Title: Plot Location on Map	
Additional Notes:			
Test Result:	Pass		

Test Case ID: 2.3		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device's GPS co-ordinates and display them on a map.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Program Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay) into the Simulator		
3	Select 'Get GPS Location' button		
4	Select 'Map' button in top right of Location Details View Controller		
5	Check if the device's location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:	Pass		

Test Case ID: 2.4		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">• Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:

Test Case ID: 2.4		Test Case Title: Plot Location on Map	
1	Run 'Wanderlust' application in Simulator		
2	Program Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) into the Simulator		
3	Select 'Get GPS Location' button		
4	Select 'Map' button in top right of Location Details View Controller		
5	Check if the device's location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:		Pass	

Test Case ID: 2.5		Test Case Title: Plot Location on Map	
Test Case Description:		The purpose of this test case is to check the application can correctly acquire the device's GPS co-ordinates and display them on a map.	
Test Case Parameters:		In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none"> Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth) 	
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Program Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth) into the Simulator		
3	Select 'Get GPS Location' button		
4	Select 'Map' button in top right of Location Details View Controller		
5	Check if the device's location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			

Test Case ID: 2.5		Test Case Title: Plot Location on Map	
Test Result:	Pass		

Lab Test Cases 3.2 - 3.5

Test Case ID: 3.2		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view. The test also checks the list can be filtered using a dropdown menu and the results updated.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
6	Select ‘All’ dropdown menu and choose a category to filter businesses		‘Hotels’ category selected from filter
7	Check businesses now provided in list table view comply with chosen category	Pass	20 hotels are displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 3.3		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device's GPS co-ordinates then display them in a list table view. The test also checks the list can be filtered using a dropdown menu and the results updated.		

Test Case ID: 3.3		Test Case Title: Gather Businesses Nearby List View	
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Centre, Cardiff Bay) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
6	Select ‘All’ dropdown menu and choose a category to filter businesses		‘Supermarkets’ category selected from filter
7	Check businesses now provided in list table view comply with chosen category	Pass	4 supermarkets are displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 3.4		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view. The test also checks the list can be filtered using a dropdown menu and the results updated.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		

Test Case ID: 3.4		Test Case Title: Gather Businesses Nearby List View	
2	Program Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) into the Simulator		
3	Select 'Get GPS Location' button		
4	Select 'Businesses' icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
6	Select 'All' dropdown menu and choose a category to filter businesses		'Schools' category selected from filter
7	Check businesses now provided in list table view comply with chosen category	Pass	1 school is displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 3.5		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view. The test also checks the list can be filtered using a dropdown menu and the results updated.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lakes, Penarth) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view

Test Case ID: 3.5		Test Case Title: Gather Businesses Nearby List View	
6	Select 'All' dropdown menu and choose a category to filter businesses		'Bars' category selected from filter
7	Check businesses now provided in list table view comply with chosen category	Pass	2 bars are displayed in list table view
Additional Notes:			
Test Result:		Pass	

Lab Test Cases 4.2 - 4.5

Test Case ID: 4.2		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided		
6	Select ‘Map’ button in the top right corner of the view		
7	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location

Test Case ID: 4.2		Test Case Title: Gather Businesses Map View	
8	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 4.3		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Centre, Cardiff Bay) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided		
6	Select ‘Map’ button in the top right corner of the view		
7	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location

Test Case ID: 4.3		Test Case Title: Gather Businesses Map View	
8	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:		Pass	

Test Case ID: 4.4		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided		
6	Select ‘Map’ button in the top right corner of the view		
7	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location

Test Case ID: 4.4		Test Case Title: Gather Businesses Map View	
8	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:		Pass	

Test Case ID: 4.5		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lakes, Penarth) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
5	Check if a list of businesses nearby is provided		
6	Select ‘Map’ button in the top right corner of the view		
7	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location

Test Case ID: 4.5		Test Case Title: Gather Businesses Map View	
8	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:	Pass		

Lab Test Cases 5.2 - 5.5

Test Case ID: 5.2		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Weather’ icon in third space on tab bar located at the bottom of the view		
5	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:	Pass		

Test Case ID: 5.3		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Centre, Cardiff Bay) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Weather’ icon in third space on tab bar located at the bottom of the view		
5	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:	Pass		

Test Case ID: 5.4		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) into the Simulator		
3	Select ‘Get GPS Location’ button		

Test Case ID: 5.4		Test Case Title: Gather Weather Information	
4	Select 'Weather' icon in third space on tab bar located at the bottom of the view		
5	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:		Pass	

Test Case ID: 5.5		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lakes, Penarth) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Weather’ icon in third space on tab bar located at the bottom of the view		
5	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:	Pass		

Lab Test Cases 6.2 - 6.5

Test Case ID: 6.2		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">• Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
5	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Cardiff location
Additional Notes:			
Test Result:	Pass		

Test Case ID: 6.3		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">• Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		

Test Case ID: 6.3		Test Case Title: Gather Tweets About Location	
2	Program Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Centre, Cardiff Bay) into the Simulator		
3	Select 'Get GPS Location' button		
4	Select 'Tweets' icon in fourth space on tab bar located at the bottom of the view		
5	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user's Cardiff location
Additional Notes:			
Test Result:		Pass	

Test Case ID: 6.4		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">• Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
5	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Vale of Glamorgan location
Additional Notes:			
Test Result:	Pass		

Test Case ID: 6.5		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test using the co-ordinates of 5 different locations to ensure the application can find the correct area. The co-ordinates used for this test are: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Program Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lakes, Penarth) into the Simulator		
3	Select ‘Get GPS Location’ button		
4	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
5	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Vale of Glamorgan location
Additional Notes:			
Test Result:	Pass		

Appendix 3: Field Test Cases

Field Test Cases 7.2 - 7.5

Test Case ID: 7.2		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application physically on device		
3	Select ‘Get GPS Location’ button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 7.3		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application physically on device		
3	Select ‘Get GPS Location’ button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address

Test Case ID: 7.3		Test Case Title: Gather GPS Information	
Additional Notes:			
Test Result:	Pass		

Test Case ID: 7.4		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application physically on device		
3	Select ‘Get GPS Location’ button		
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 7.5		Test Case Title: Gather GPS Information	
Test Case Description:	The purpose of this test case is to check the application loads up successfully and correctly, acquires then displays the device’s GPS location information in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application physically on device		
3	Select ‘Get GPS Location’ button		

Test Case ID: 7.5		Test Case Title: Gather GPS Information	
4	Observe if Location Details View Controller displays the correct information	Pass	Correct latitude and longitude is displayed on view controller, along with correct approximate address
Additional Notes:			
Test Result:		Pass	

Field Test Cases 8.2 - 8.5

Test Case ID: 8.2		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Map’ button in top right of Location Details View Controller		
4	Check if the device’s location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:	Pass		

Test Case ID: 8.3		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:

Test Case ID: 8.3		Test Case Title: Plot Location on Map	
1	Run 'Wanderlust' application in Simulator		
2	Select 'Get GPS Location' button		
3	Select 'Map' button in top right of Location Details View Controller		
4	Check if the device's location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:		Pass	

Test Case ID: 8.4		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Map’ button in top right of Location Details View Controller		
4	Check if the device’s location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:	Pass		

Test Case ID: 8.5		Test Case Title: Plot Location on Map	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire the device’s GPS co-ordinates and display them on a map in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Map’ button in top right of Location Details View Controller		
4	Check if the device’s location is plotted correctly on a map	Pass	Map is displayed with a marker added in the position corresponding to the inputted latitude and longitude
Additional Notes:			
Test Result:	Pass		

Field Test Cases 9.2 - 9.5

Test Case ID: 9.2		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view

Test Case ID: 9.2		Test Case Title: Gather Businesses Nearby List View	
5	Select 'All' dropdown menu and choose a category to filter businesses		'Hotels' category selected from filter
6	Check businesses now provided in list table view comply with chosen category	Pass	20 hotels are displayed in list table view
Additional Notes:			
Test Result:		Pass	

Test Case ID: 9.3		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
5	Select ‘All’ dropdown menu and choose a category to filter businesses		‘Supermarkets’ category selected from filter
6	Check businesses now provided in list table view comply with chosen category	Pass	4 supermarkets are displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 9.4		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
5	Select ‘All’ dropdown menu and choose a category to filter businesses		‘Schools’ category selected from filter
6	Check businesses now provided in list table view comply with chosen category	Pass	1 school is displayed in list table view
Additional Notes:			
Test Result:	Pass		

Test Case ID: 9.5		Test Case Title: Gather Businesses Nearby List View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a list table view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		

Test Case ID: 9.5		Test Case Title: Gather Businesses Nearby List View	
4	Check if a list of businesses nearby is provided	Pass	List of businesses nearby is displayed in a list table view
5	Select 'All' dropdown menu and choose a category to filter businesses		'Bars' category selected from filter
6	Check businesses now provided in list table view comply with chosen category	Pass	2 bars are displayed in list table view
Additional Notes:			
Test Result:		Pass	

Field Test Cases 10.2 - 10.5

Test Case ID: 10.2		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Select ‘Map’ button in the top right corner of the view		
5	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location
6	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address

Test Case ID: 10.2		Test Case Title: Gather Businesses Map View	
Additional Notes:			
Test Result:	Pass		

Test Case ID: 10.3		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Select ‘Map’ button in the top right corner of the view		
5	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location
6	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 10.4		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Businesses’ icon in second space on tab bar located at the bottom of the view		
4	Select ‘Map’ button in the top right corner of the view		
5	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user’s current location
6	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:	Pass		

Test Case ID: 10.5		Test Case Title: Gather Businesses Map View	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire businesses nearby using the device’s GPS co-ordinates then display them in a map view in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:

Test Case ID: 10.5		Test Case Title: Gather Businesses Map View	
1	Run 'Wanderlust' application in Simulator		
2	Select 'Get GPS Location' button		
3	Select 'Businesses' icon in second space on tab bar located at the bottom of the view		
4	Select 'Map' button in the top right corner of the view		
5	Check if businesses nearby are plotted on a map	Pass	Businesses nearby are plotted on a map view along with a marker indicating user's current location
6	Click on marker for a business to check information window appears	Pass	When marker is selected information window appears displaying name of business and approximate address
Additional Notes:			
Test Result:		Pass	

Field Test Cases 11.2 - 11.5

Test Case ID: 11.2		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Weather’ icon in third space on tab bar located at the bottom of the view		

Test Case ID: 11.2		Test Case Title: Gather Weather Information	
4	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:		Pass	

Test Case ID: 11.3		Test Case Title: Gather Weather Information	
Test Case Description:		The purpose of this test case is to check the application can correctly acquire weather information for the user's current GPS location and display this to the user in a real world situation.	
Test Case Parameters:		In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none"> Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay) 	
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Select 'Get GPS Location' button		
3	Select 'Weather' icon in third space on tab bar located at the bottom of the view		
4	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:		Pass	

Test Case ID: 11.4		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Weather’ icon in third space on tab bar located at the bottom of the view		
4	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc
Additional Notes:			
Test Result:	Pass		

Test Case ID: 11.5		Test Case Title: Gather Weather Information	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire weather information for the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Weather’ icon in third space on tab bar located at the bottom of the view		
4	Observe if Weather View Controller displays appropriate weather information	Pass	Appropriate weather information is displayed including a summary icon and current temperature etc

Test Case ID: 11.5		Test Case Title: Gather Weather Information	
Additional Notes:			
Test Result:	Pass		

Field Test Cases 12.2 - 12.5

Test Case ID: 12.2		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.48468 , Longitude: -3.16889 (The Parade, Cardiff City Centre)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
4	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Cardiff location
Additional Notes:			
Test Result:	Pass		

Test Case ID: 12.3		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">• Latitude: 51.46757 , Longitude: -3.16192 (Red Dragon Center, Cardiff Bay)		

Test Case ID: 12.3		Test Case Title: Gather Tweets About Location	
Test Step:	Step Description:	Result:	Notes:
1	Run 'Wanderlust' application in Simulator		
2	Select 'Get GPS Location' button		
3	Select 'Tweets' icon in fourth space on tab bar located at the bottom of the view		
4	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user's Cardiff location
Additional Notes:			
Test Result:	Pass		

Test Case ID: 12.4		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">Latitude: 51.44715 , Longitude: -3.26644 (Wenvoe, The Vale of Glamorgan)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
4	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Vale of Glamorgan location
Additional Notes:			
Test Result:	Pass		

Test Case ID: 12.5		Test Case Title: Gather Tweets About Location	
Test Case Description:	The purpose of this test case is to check the application can correctly acquire tweets about the user’s current GPS location and display this to the user in a real world situation.		
Test Case Parameters:	In order to thoroughly test this feature, I will run the test on the device whilst present at the following location: <ul style="list-style-type: none">• Latitude: 51.41576 , Longitude: -3.18298 (Cosmeston Lake, Penarth)		
Test Step:	Step Description:	Result:	Notes:
1	Run ‘Wanderlust’ application in Simulator		
2	Select ‘Get GPS Location’ button		
3	Select ‘Tweets’ icon in fourth space on tab bar located at the bottom of the view		
4	Observe if Tweets View Controller displays appropriate list of tweets	Pass	Appropriate tweets are displayed that mention the user’s Vale of Glamorgan location
Additional Notes:			
Test Result:	Pass		

Location Based iOS Application User Testing

Information Page

The Mobile Application

The aim of this project is to create an iOS mobile application that can access the device's GPS co-ordinates and process this data to provide users with information based on their current location. The types of data provided will include the user's current address and co-ordinates which should then be pin plotted onto a map of the area. The application should also display weather information for that location as well as businesses in the surrounding area with some basic information about each business. It should also provide a list of Tweets containing references to their current location. The intended audience for the project is predominantly tourists and people travelling to new areas as the application aims to provide details of businesses nearby and weather conditions. This is the type of information people visiting a new place will not be familiar with so the application could assist them in this respect.

The Purpose of the Study

The purpose of completing user testing on the application is to simply test the mobile app can function correctly in a non-controlled environment. Another reason for completing user testing is to check the practicality of creating such an application and releasing it for commercial use. It is important to ensure the app is easy to use and suits the needs of its users which is where feedback from test participants can be very useful. The feedback given will provide an insight into the application's success in terms of meeting the aims and objectives of this project.

The Test Procedure

The test procedure is very straightforward: you will firstly be provided with a device that has the application pre-installed for you, to ensure there is no risk to your own device. You should then simply run the application and use it as you wish, with no instruction being provided on how to operate it. You should use the application for 5 minutes. On completing this section of the user test, you should then complete the user feedback questionnaire as honestly as you can, providing constructive feedback on successful aspects of the application as well as areas in which you feel the application is lacking.

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 10

2. Could any functionality be added in order to improve the mobile application, if yes what?

3. Which features did you particularly like about the mobile application, if any?

4. Could any of the features of the mobile application be changed to improve it, if yes how?

5. Are there any other comments you would like to make about the mobile application?

6. If this mobile application was available commercially, would you use it? Yes / No

2

Appendix 5: Completed User Questionnaires

Test Participant 1

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 **10**

2. Could any functionality be added in order to improve the mobile application, if yes what?

Maybe to include a route to the destinations on the map section and also an option to expand or decrease the search area.

3. Which features did you particularly like about the mobile application, if any?

I really enjoyed the feature showing the closest locations to yourself.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

The options of filtering the results of things around you.

5. Any there any other comments you would like to make about the mobile application?

I think everything worked really well and I'd definitely download it if it was available commercially.

6. If this mobile application was available commercially, would you use it? **Yes** / No

Yes. I think it is really helpful and very easy to use.

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 10

2. Could any functionality be added in order to improve the mobile application, if yes what?

There could be a search bar for users to input specific requests, also the filter tab could be made more obvious.

3. Which features did you particularly like about the mobile application, if any?

I enjoyed the amount of information available about of each of the businesses. Also the fact that it provides access to google maps and directions is really useful. In addition, the Twitter feature provides interesting insight from other people in the local area.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

I don't think any improvements are necessary.

5. Any there any other comments you would like to make about the mobile application?

This is a very interesting idea that is executed really well.

6. If this mobile application was commercially available, would you use it? **Yes** / No

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 10

2. Could any functionality be added in order to improve the mobile application, if yes what?

It could be made more obvious that there are more categories in the dropdown menu and may be useful to have a search bar.

3. Which features did you particularly like about the mobile application, if any?

I liked the amount of information provided about businesses and that there are options to open their locations on maps and get directions. Once the category drop down bar is discovered it is useful to retrieve the locations of specific types of businesses. I also liked that it finds your location without the need to input the address yourself. The weather feature is also a nice touch and is useful if using the app to plan activities and or days out.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

I feel like there are enough features for this type of app, any more may make it cluttered and more complicated to use.

5. Are there any other comments you would like to make about the mobile application?

No further comments.

6. If this mobile application was available commercially, would you use it? **Yes** / No

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 10

2. Could any functionality be added in order to improve the mobile application, if yes what?

A Facebook check in location could be added to share locations and integrate social media.

3. Which features did you particularly like about the mobile application, if any?

I really like the simplicity and user experience. The app is really user friendly and I can see it being really helpful.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

I think it could be useful to separate the restaurants section into restaurants and takeaways to make it more specific.

5. Are there any other comments you would like to make about the mobile application?

I think the app is a really good idea and simple to use.

6. If this mobile application was commercially available, would you use it? **Yes** / No

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 **10**

2. Could any functionality be added to in order to improve the mobile application if yes what?

On the business list, make it more obvious that you are able to separate the businesses into different categories as this was not very clear, perhaps put "Filters: None" when they are all listed for example.

3. Which features did you particularly like about the mobile application, if any?

I liked that hat if you click on a business on the map it takes you to the page and provides the business information as well as accessing the page through the list view.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

Perhaps it would be useful if the app was available in different languages/could translate different business pages if they're not available in English.

5. Are there any other comments you would like to make about the mobile application?

Great application, I would definitely use this if i was on holiday/travelling as it provides useful information for discovering new places.

6. If this mobile application were available commercially, would you use it? **Yes** / No

2

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 **9** 10

2. Could any functionality be added to improve the mobile application, if yes what?

I didn't have any problems with the functionality of the app myself.

3. Which features did you particularly like about the mobile application, if any?

I liked the fact that the app includes both factual information about the businesses, such as opening times and description, but also that it has personal opinions in the form of reviews. It is very easy and straightforward to use, with a clear layout. The app provides enough information for those who are not familiar with the area to be aware of what services are available clearly.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

I like the simple basic design, however, the colour scheme in some areas could be improved as it appears slightly too basic.

5. Any further comments you would like to make about the mobile application?

I really like the app and would definitely download it for myself, because I like the idea of being provided with the facts and also the subjective reviews, especially for finding somewhere to eat.

6. If this mobile application was available commercially, would you use it? **Yes** / No

2

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 **9** 10

2. Could any functionality be added to improve the mobile application, if yes what?

I didn't have any problems with the functionality of the app myself.

3. Which features did you particularly like about the mobile application, if any?

I liked the fact that the app includes both factual information about the businesses, such as opening times and description, but also that it has personal opinions in the form of reviews. It is very easy and straightforward to use, with a clear layout. The app provides enough information for those who are not familiar with the area to be aware of what services are available clearly.

4. Could any of the features of the mobile application be changed to improve it, if yes how?

I like the simple basic design, however, the colour scheme in some areas could be improved as it appears slightly too basic.

5. Any further comments you would like to make about the mobile application?

I really like the app and would definitely download it for myself, because I like the idea of being provided with the facts and also the subjective reviews, especially for finding somewhere to eat.

6. If this mobile application was available commercially, would you use it? **Yes** / No

2

Location Based iOS Application User Testing

User Feedback Form

1. On a scale of 1 to 10, how would you rate the usability of the mobile application?
(1 = Unusable, 10 = Very easy to use)

1 2 3 4 5 6 7 8 9 10

2. Could any functionality be added in order to improve the mobile application, if yes what?

It could be useful to give the exact distance from the business on the app so people could see in numbers how far away it is from them.

There could also be a rain radar map that could show people how far away rain is.

3. Which features did you particularly like about the mobile application, if any?

I love the tweet function. It's really good to see what people around you are saying at that minute in time.

Also the list of local businesses are really useful as it is easy to see on the map which ones are closest.

4. Could any of features of the mobile application be changed to improve it, if yes how?

The weather section could maybe use a bit more information with what the weather would be in the next coming minutes.

5. Are there any other comments you would like to make about the mobile application?

I tweeted and it showed up on the app tweet section which was amazing!!

I would download this app - I think it would be a really useful thing to make exploring a city a lot easier and to see local information via the tweet section.

6. If this mobile application was available commercially, would you use it? **Yes** / No

2