



## **Final Report**

### **Project 118: 'Electronic Map of COMSC'**

**By Melissa Whiting, C1432295**

**CM3203 One Semester Individual Project, 40 Credits**

**Supervisor: Helen R Phillips**

**Moderator: Padraig Corcoran**

## **Abstract**

The aim of the project was to carry out a feasibility study in order to determine if it would be possible to create an application that could help both students and staff navigate around the Queen's building at Cardiff University more effectively. Research carried out into this subject area revealed that indoor navigation is still a problem that has not yet been fully overcome. Surveys carried out helped to shed light on how much of an issue indoor navigation is within the Queen's building. The feasibility study helped to identify that the most feasible solution to this problem would be a social media Bot. A prototype of the Bot has been implemented, making use of Javascript and the Facebook API. Due to a bug with the Facebook API some directions are in a list format, whereas others are in a plain text format. This is something that can be easily changed when the bug has been fixed by Facebook.

Various different tests were carried out on the prototype, determining it a success as it dramatically reduced the time it took individuals to find their desired destination. All users of the Bot said that they found it helpful and it is something that they would like to use themselves. The Bot works for various operating systems. However, it was found that for some iOS devices, the directions that were presented in a list format often had the last word or two missing. This was a minor issue as it did not affect the user's ability to find their desired location. As a result of this, it is recommended that Cardiff University consider developing a full version of the Bot for future use.

## **Acknowledgements**

I would like to thank my supervisor, Helen Phillips, for all of her support and guidance throughout the project. I would also like to thank my friends and family for their help in testing my prototype.

## Contents

1. Introduction .....	6
2. Background .....	8
2.1. Technologies We Use Today .....	8
2.1.1. GPS .....	8
2.1.2. Wireless Beacons with Bluetooth .....	9
2.1.3. RFID .....	9
2.2. How Blind People Currently Navigate .....	10
2.2.1. Guide Dogs .....	10
2.2.2. Human Guides .....	11
2.2.3. White Canes .....	11
2.3. The Challenges Associated with Indoor Navigation .....	11
2.4. Current Solution for Navigation within the Queen's Building .....	12
2.5. Existing Solutions for Navigation in a Similar Scenario .....	14
2.5.1. Sat Navs .....	14
2.5.2. Google Maps .....	15
2.5.3. Wayfindr .....	17
3. Feasibility Study .....	18
3.1. Potential Solution One – Mobile Application using iBeacon Technology .....	18
3.1.1. Technical Feasibility .....	18
3.1.2. Legal Feasibility .....	22
3.1.3. Operational Feasibility .....	23
3.1.4. Schedule Feasibility .....	26
3.1.5. Financial Feasibility .....	26
3.2. Potential Solution Two – Social Media 'Chat-Bot' .....	27
3.2.1. Technical Feasibility .....	27
3.2.2. Legal Feasibility .....	31
3.2.3. Operational Feasibility .....	32
3.2.4. Schedule Feasibility .....	34
3.2.5. Financial Feasibility .....	35
4. Implementation of a Prototype .....	36
4.1. Chosen Solution .....	36
4.2. Implementation .....	36

4.2.1. Example Route .....	37
4.2.2. Creating the Bot .....	37
5. Results and Evaluation .....	49
5.1. App vs No App.....	49
5.2. Usability Testing.....	50
5.3 Questionnaire .....	51
5.4. Changes Made as a Result of User Testing .....	52
5.5 Reliability Testing .....	52
6. Future Work .....	62
7. Conclusions .....	63
8. Reflection on Learning .....	65
9. Appendices.....	69
10. References .....	80

## 1. Introduction

The aim of this project is to carry out a feasibility study in order to determine if it would be possible to create an application that could help both students and staff navigate around the Queen's building at Cardiff University more effectively.

Navigation is a significant problem within the Queen's building at Cardiff University due to the lack of signs for way finding. Many new students, members of staff or visitors often find themselves lost and confused about how to reach their destination, be it a lecture room or a computer laboratory. For students, being lost can result in their late attendance at lectures or completely missing the lecture due to not being able to find the correct room. For many students with anxiety who find the room late, it could result in them deciding to skip the entire lecture as they do not want to face walking in late and drawing attention to themselves. Missing lectures can severely affect a students' academic performance and jeopardise their learning as they are missing out on invaluable contact time with course lecturers. For staff, not being able to locate the correct teaching room could result in turning up late to a lecture. This then affects the lecturers' ability to deliver all course content as lectures are only fifty minutes long, again affecting students as they are missing out on vital teaching.

This report will look more closely at the problem of indoor navigation by exploring and analysing existing solutions, evaluating the current way of navigating the Queen's building and surveying individuals to determine how severe the problem currently is. This report will include a feasibility study of two potential solutions to the problem (one, a live map application using iBeacon technology, and the other, a social media 'chat-bot' navigational planner). The project will also include implementation of a prototype of the most feasible solution, which will be tested and evaluated in order to determine if it solves the problem fully.

The main aims and goals of this project are as follows:

- Understand the problem by carrying out research into existing applications that solve navigational problems in similar scenarios, and by carrying out an analysis of the current solution to the navigational problem at university.
- Identify and carry out a feasibility study on two separate potential solutions in order to determine the most feasible solution to the problem.
- Carry out a survey to determine how severe the navigational problem is and identify the most common problems individuals face when visiting the building for the first time. The survey should also identify how their learning/ teaching is affected by poor navigational aids, give insight into how others see the problem being resolved, and highlight areas within the building where the current navigation system is confusing.
- Create a prototype of the most feasible solution in order to solve the problem of navigation within the Queen's building.
- Create and carry out user testing to determine how well the solution solves the problem.

The intended audience of such a solution would be both university staff and students who have a need for getting around the Queen's building, e.g. lecturers and students from within the School of Computer Science and Informatics.

The scope of this project is to solve the problem of navigation within the Queen's building by carrying out a feasibility study and then implementing a prototype which should allow users to give a start location and a destination. The solution should then provide the user with easy to follow instructions of how to get to the desired destination. The solution should enable the user to find their destination in under ten minutes, as this is the length of time that a student has to move rooms between lectures. However, there is potential in the future that such a solution could also be developed for the other schools within Cardiff University.

The waterfall model (Rouse 2007) of software development will be used throughout this project as each task needs to be carried out sequentially. Research and feasibility need to be carried out before implementation can start, and implementation needs to be carried out before testing and reflection.

This work is based on the assumption that all members of staff or students within the building have access to a mobile phone which can connect to the universities WiFi. It is also assumed that the WiFi covers all areas of the building.

## **2. Background**

The problem of navigation is not a new one. During the 1700's primitive navigation techniques were used to steer ships, where sailors had to use an accurate map along with their latitude and longitude in order to determine their approximate location (BBC 2009). Compasses, Davis Quadrants and Astrolabes were used to help navigate during this time.

Today we utilise a multitude of technologies to aid us with positioning, such as GPS, wireless beacons with Bluetooth and RFID.

This section will look at the different technologies we use today to aid us in navigating, the problems that visually impaired people face and what they currently use to help them navigate, the challenges associated with indoor navigation, the current way finding system used within the Queen's building and an analysis of existing tools used for navigation.

### **2.1. Technologies We Use Today**

'The Science and technology for positioning and navigation has experienced a dramatic evolution' (G, Seco-Granados et al. 2012, p. 1). Below is some information about the different technologies that we use to help with navigation today.

#### **2.1.1. GPS**

GPS, also known as the Global Positioning System, is a satellite navigation system which was originally developed by the U.S Department of Defence for military use (Garmin). GPS works as there are GPS satellites orbiting the Earth two times a day, each transmitting a different signal. A GPS device receives these signals and determines the distance from each satellite based on the length of time it takes to receive them, which is used to determine a users' exact location on Earth.

Some of the major advantages of using GPS are that it works in all weather conditions, anywhere in the world, twenty-four hours a day. GPS is accurate due to its parallel multi-channel design. GPS is also free. GPS devices such as Sat Navs, or even mobile phones, are widely used for navigation as they can provide detailed directions, making navigation easier for the user. GPS can also be embedded within wearable technology, such as watches, making it more accessible when needed and more integrated into the users' life.

However, there are also some drawbacks to using GPS, such as the accuracy of GPS can be affected by certain atmospheric factors, e.g. ionosphere and troposphere delays caused by the signal slowing down whilst passing through the atmosphere. GPS signals will not pass through solid objects such as buildings, so GPS cannot be used for indoor navigation. Getting directions on the GPS device relies on the user being in range of a satellite. Using mobile phones for navigation with GPS relies on the mobile phone having a cell connection (M. Orth 2015). As well as this, GPS accuracy can be affected by not updating the device, as road names etc change all the time.



### **2.1.2. Wireless Beacons with Bluetooth**

Bluetooth is a wireless technology that uses radio waves to exchange data over short distances (WebWise Team 2012; Bluetooth a). Bluetooth beacons make use of BLE (Bluetooth Low Energy), which was built for use by Internet of Things (IoT) devices (Bluetooth b). Bluetooth nodes are used in order to utilise Bluetooth with wireless beacons for navigation. These transmit messages to a users' device to uniquely identify themselves, which can then be used by applications in order to determine their current location.

One major advantage of Bluetooth with wireless beacons is that it can be used for navigation indoors (E. Gaudlitz 2015). Beacons are small in size, so they can be placed virtually anywhere. Beacons are cheap, and they also have low energy consumption so can run on button cell batteries for more than a year, making them a cost effective solution. Roman Foeckle, CEO of Onyx Beacons, says that 'basic beacon prices should quickly fall to \$10, so they'll be affordable to install en masse' (G. Gruman 2014). Accuracy of this method is up to one meter. Bluetooth is also very secure. Bluetooth 5 has more advanced location based capabilities. The range is quadrupled, speed is doubled and broadcast messaging is boosted by 800%. As well as this, this solution can be used for much more than just navigation. There is potential to use this solution for things such as tracking shoppers within a shopping centre in order to target them with personalised voucher codes depending on the types of shops that they visit. Many devices (iPhones, iPads, Android mobile phones) already have BLE technology inbuilt.

However, there are also some disadvantages associated with using wireless beacons with Bluetooth. One disadvantage is that additional hardware, such as the actual beacons, is required, increasing the cost of the solution. As well as this, client based solutions also requires an application to be built in order to use the beacon information (E. Gaudlitz 2016). The beacons also have a relatively small range of up to thirty meters, so it isn't a viable solution for larger areas.

### **2.1.3. RFID**

RFID, also known as Radio-Frequency Identification, provides unique identification for a given object (Technovelgy a). This object must then be scanned to get the relevant information. An RFID system consists of an antenna, a transceiver and a transponder (Technovelgy b). The antenna transmits a signal. When a tag detects the signal, the RFID chip turns on and transmits the information back to the antenna. RFID can be embedded within a name badge, for example, which could then be used to identify a persons' location within a building. RFID is more suited for object tracking than indoor navigation as it does not allow automatic positioning (E. Gaudlitz 2016).

One of the advantages of using RFID is that the tag does not need to be positioned relative to the scanner in order to be read, as it isn't reliant on line-of-sight, so they can work within a few feet of the scanner (Technovelgy f). RFID tags are now very cheap, being 'used as a throwaway inventory or control device' (Technovelgy a). One advantage associated with active RFID tags is that they have their own power source, which means that they can be further away from the antenna, up to 300 meters, and still receive a signal. One advantage associated with passive RFID tags is that they require no batteries, meaning that they can be smaller in size and have a virtually unlimited life span. The read time of an RFID tag is very

fast (<100 milliseconds), and multiple tags can be read at the same time. Tags do not have to be on the surface of an object to be read.

However, there are also disadvantages to using RFID. One disadvantage of RFID systems is that they require the use of special scanners and readers, which are expensive, meaning a higher solution cost and less accuracy as it is not possible to install them everywhere. RFID systems are also relatively easy to jam (Technovelgy e). RFID tags are difficult to remove from an object, as they may be too small or embedded. Some problems associated with active RFID tags are that they can be repeatedly read by a scanner, which could result in the battery tiring out. They also have a limited life span. Both reader and tag collisions can occur. Reader collisions occur when two or more reader signals overlap, and tag collisions occur when there are lots of tags in a small area at the same time. One concern with RFID systems is that anyone close enough with a scanner can activate and read an RFID device (Technovelgy c). This makes RFID systems not very secure or private. From the above example, someone could read the RFID tag within a name badge and gain unauthorised access into a building. Many civil liberties groups also have concerns 'about the use of RFIDs to track the movements of individuals' (Technovelgy d).

## **2.2. How Blind People Currently Navigate**

Whilst travelling, blind or visually impaired people use a mixture of their senses to help them navigate, such as touch, smell and sound. Memory can also be an important tool for navigating around specific areas that are familiar to a person (Sandys View 2015). For example, a blind or visually impaired person may rely on Braille signs to help identify a building, using their sense of touch. According to the American Foundation for the Blind, 'Braille is a system of raised dots that can be read with the fingers by people who are blind or who have low vision' (American Foundation for the Blind 2016). A blind or visually impaired person may use their sense of smell in order to identify what is nearby, such as a bakery. They may also use audio announcements on public transport to determine when to exit the vehicle. There are three main methods that blind or visually impaired people use to navigate around unfamiliar surroundings: Guide dogs, human guides or long white canes (Accessible Pedestrian Signals).

### **2.2.1. Guide Dogs**

Guide dogs are dogs that have been specially trained to obey their handlers' commands and ensure that their handler is kept out of harm's way. These dogs obey basic orientation commands such as left, right and forward, but will navigate around obstacles for their handler to avoid any danger. Contrary to common belief, it is the handler who makes all decisions whilst navigating.

### **2.2.2. Human Guides**

Many blind or visually impaired people use human guides whilst navigating. A human guide is someone with full sight, who they feel that they can trust. A blind or visually impaired person may hold onto the arm of their human guide who will then lead them around safely.

### **2.2.3. White Canes**

Many blind or visually impaired people choose to use a white cane in order to help with navigation. A white cane is a long white stick that may be swung out in front of the visually impaired person along the floor in order to help them determine if there are any obstacles in front of them. The cane is generally long enough to determine what is in front of the individual for the next few steps. For example, an individual may use a white cane to determine if they are near a crossing, as crossings generally have circular bumps embedded into the ground that can be detected with the cane, or platform edges at train stations. Using a white cane can also signal to others, such as motorists, that the person using the cane is visually impaired.

## **2.3. The Challenges Associated with Indoor Navigation**

Meir Druyan, an author at SPREO, once said that 'if you think outdoor navigation is hard, try to map and navigate indoors. It's an even more elusive challenge' (M, Druyan 2014).

As mentioned above, GPS signals will not go through solid objects, such as buildings, so GPS cannot be used for indoor navigation. One of the main challenges associated with indoor navigation is finding a technology that can be utilised for this purpose. Many people are now exploring the possibility of using both WiFi and Bluetooth for indoor navigation, but these technologies were 'built to transfer data, not to show location' (M, Druyan 2014).

Another problem associated with indoor navigation is accuracy. In order to be effective, navigation systems indoors need to have high positioning accuracy of roughly '2 meters' (M, Druyan 2014). Even if GPS did work within indoor environments, its accuracy is between 5-15 meters, meaning that it would not be nearly accurate enough to use effectively. As both WiFi and Bluetooth were built for transferring data, then it may not exactly be the most accurate technology to use for positioning.

Buildings can also have various levels, so another challenge associated with indoor navigation would be identifying areas where navigational help is required, such as stair cases and left and right corridors. These will be the areas where individuals are able to go the 'wrong' way if they have no clear instructions given to them.

## 2.4. Current Solution for Navigation within the Queen's Building

In the Queen's building there is currently only minimal signage that individuals can use to help orient and navigate around the campus. When someone first arrives at the campus it can be very confusing and difficult to find a specific room, given the various levels and buildings.

The following test was carried out in order to determine how difficult it is to navigate around the campus for individuals who have never been there before:

An individual who has never been to the Queen's building was taken to the security office. This individual was told only where they currently were and what room number they had to find. The time it took for the individual to find the room was timed. The individual was asked to find room S/2.21.

This individual took 19 minutes and 21 seconds to find the room, where it only takes an experienced third year individual 3 minutes and 8 seconds.

Below shows the results from all the tests that were carried out:

Start Location	End Location	Time Taken
Canteen	S/2.21	19 minutes 21 seconds
Canteen	N/4.07	2 minutes 46 seconds
Canteen	C/2.07	6 minutes 17 seconds

These results suggest that signage within the building is very weak and could be greatly improved. The individual involved within this test commented on the lack of signs within the building, even fire escape route signs, and that the signs that were present were not clear. (They were not sure whether signs meant go straight ahead or go up the stairs. The signs gave no hints at how to access other blocks etc)

A survey was then created to determine how severe the problem of navigation is within the building. Before creating the survey, ethical approval was received as the survey contained questions regarding disabilities. When creating the survey, care was taken to ensure anonymity was maintained. All participants were told what the purpose of the survey was, allowing them to decide if they wished to participate or not. The survey was created using SurveyMonkey, a web-based solution for creating free surveys (SurveyMonkey 2009). SurveyMonkey uses SSL encryption and multi-machine backup to keep data secure.

The survey contained the following questions:

- Have you ever found yourself lost or unsure of where to go within the building?
- How did you get lost? Why were you unsure of where to go?
- Do you think the existing signs for navigation within the building are useful?
- Do you have a disability?
- Does having a disability affect your navigation around the building in any way?
- Do you own a mobile phone?
- What OS does your device have?

- Would you rather use an application that is already installed on your device or a new application specifically for navigation within the building?

Out of the 40 people surveyed, 85% of people have found themselves lost or unsure of where to go within the Queen's building at some point during their time at university (see appendix A). This clearly shows that navigation is a major issue that needs to be addressed within the university building as it directly affects a student's ability to study. For example, some students arrived late, or completely missed lectures due to being unable to find the correct room. Cardiff Universities' School of Computer Science and Informatics is a likely stakeholder within this problem area.

When asked why they got lost or why they were unsure of where to go, a multitude of answers were generated. The most commented on instances were as follows:

- Crossing between buildings can be confusing. Cannot find where the buildings join. Eg, North and West buildings connect on different floor numbers.
- There are very few signs for navigation, and they do not direct you to other buildings. 57.50% of people surveyed said that they do not think the existing signs for navigation within the building are useful (see appendix B). Many students commented that they had not noticed navigational signs due to them being too small, and one student commented that they had to follow paper signs put up by a lecturer around the building to find the correct lecture room. However, one student noted that the new, big signs at specific staircases within the building are helping.
- No indication of how room numbers are assigned. They seem inconsistent. They are not intuitive. Some students commented that they try to find room numbers themselves by wandering around the building as the signs are more confusing.
- There are lots of staircases within the Queen's building, and no clear way of identifying which rooms are on what floors.
- Never explicitly told what each building is called.
- Staff at the university may not have adequate knowledge of the building, as they have trouble directing students to specific rooms.

These are the main issues that should be focussed on when trying to improve navigation within the Queen's building.

10% of participants identified themselves as having a disability (see appendix C). Only half of these individuals said that having a disability affected their navigation around the building. Some participants commented on the fact that having Aspergers Syndrome can result in them getting lost in unfamiliar settings. Others commented on the fact that having a hearing disability meant that when asking other students or members of staff for directions it can be difficult to listen to their responses. Some participants also offered their views on how they see the problem with navigation being solved. They suggested that a reference map of the building would be extremely helpful. Others suggested creating 3D interactive signs that could be placed around the building.

100% of participants said that they own a mobile phone (see appendix D). 57.50% of participants own mobile phones with an Android Operating System (OS). 37.50% of participants own a mobile phone with an iOS OS, and 5% of participants own a mobile phone with a Windows OS (see appendix E). 52.63% of participants said that they would rather use an existing application for navigation within the building over downloading and installing a new application just for this specific purpose (see appendix F).

One lecturer also mentioned that he still sometimes gets lost while trying to find his own room even though he has been at the university for a few years.

This information clearly shows that there is a need for improved navigation within the Queen's building, and that a mobile solution is definitely feasible. It also clearly identifies that students and new visitors are the most likely users of the potential solution.

## **2.5. Existing Solutions for Navigation in a Similar Scenario**

Below are some of the existing solutions to the navigation problem.

### **2.5.1. Sat Navs**

A Sat Nav, or Satellite Navigational System, is a GPS receiver which combines pre-programmed maps with GPS readings that determine the physical location of the Sat Nav on Earth to help guide users to their desired destinations (KNOWHOW 2012). Sat Navs are used for vehicle navigation. As Sat Navs use GPS they cannot be used for indoor navigation, making it an unsuitable solution to the problem at hand. We will be creating a prototype for an application that will aim to work in an indoor environment. Sat Navs can also be affected by tall buildings, which block the GPS signal, and storms which affect their accuracy (Teach-ICT). Sat Nav maps need to be constantly updated to ensure that a user is using the most up-to-date map that takes into account new road changes. If Sat Navs are not updated then users could be taken down unsuitable roads. Users of Sat Navs should use their common sense and not blindly rely on them. For example, in 2014 'a fuel tanker was stuck for three days in Wiltshire after getting stuck in a narrow country lane' (L, Kelion 2016) after relying on a Sat Nav for directions. Users should always keep a physical copy of a map for when Sat Navs become affected to ensure that they are not lost.

Sat Navs come in a variety of different forms, the most common being a portable Sat Nav that can be fixed to the windscreen of a car (Satnav). Users can input their desired destination by typing the address using the Sat Nav touch screen, making them intuitive and easy to use. The Sat Nav will then 'calculate' your route, often choosing the shortest path. The Sat Nav can be used to navigate to any location. This saves time on having to plan routes to unfamiliar locations as the Sat Nav does the route planning for you.

The Sat Nav provides a graphical representation in the form of a map of the roads that you are driving along. The location of the vehicle is updated. The screen also represents buildings etc from a birds-eye view, making it easier for users to visualise where they are or where they are going. Sat Navs can be very accurate. The screen also provides written

prompts, such as the next instruction or the speed limit. This is helpful to users as they don't have to remember the next instruction. The screen also provides street names, again helping the user to orientate themselves. The screen eliminates the need for a physical map which can be hard to read whilst driving, lowering the risk of getting lost or having an accident.

As well as the screen, most Sat Navs also provide voice directions, which can be useful for those who cannot multi-task and look at the Sat Nav screen whilst driving. The voice directions give clear instructions to the user, as well as distance information. Eg 'In two hundred yards turn left'. These instructions are clear and concise and therefore easy for the user to follow. Sat Navs can also account for the user going the wrong direction, as they can re-calculate the route a user needs to take. This is useful as it helps to ensure that the user will not get lost.

### **2.5.2. Google Maps**

Google Maps is an online map service developed by Google that can be accessed via web browser or mobile device (G tricks 2016; GCF Learn Free). It relies on aerial and satellite images that are taken every two weeks in order to stay up-to-date (M, Petronzio 2012), as well as mobile location services (R, Nightingale 2017). Google Maps uses a form of the Dijkstra shortest path algorithm in order to determine routes for navigation (M, Byrne 2015). It allows users to quickly and easily create routes and navigate to desired locations (Google Maps a). When creating a route, the user has multiple transportation choices such as Walking, Driving, Buses, Trains and Cycling. Google Maps will then calculate the route based on the mode of transportation chosen. It finds real-time data for each mode of transport, such as bus schedules and traffic congestions, which easily allows the user to find the quickest route that they should take and re-calculates routes on the fly to adapt to new data. As with Sat Nav, Google Map also provides turn-by-turn navigation.

Google Maps can also be used to find out information about various places of interest such as restaurants. Users can easily search by typing in the name of the establishment they're looking for and then view information such as opening hours, website, busy periods and reviews, helping them make an informed decision. Users can even book tables in restaurants from within the Google Maps app, making the process easier and more efficient for the user. 'Local guides' can be browsed to find the most popular bars and restaurants among residents in a local area. Google Maps isn't just a navigation service anymore.

Another way in which Google crowd sources data for their Maps is through Google Map Maker, which allows any user to submit data to Google Maps. However, this feature is now being incorporated fully into Google Maps directly. This allows all users to fact check and to add little details to Google Maps, making it a rich source of information for others.

Another aspect that is closely related to Google Maps is Google Street View. It allows users to view exact images of streets, so they can see how the area looks in real life. Google cars drive around the Earth and take photos which are then put on top of the base Google Map. However, these images can sometimes be a few years out of date, meaning an area may look different in real life.

Google Maps can also be used by users to create their own customised maps. Eg, adding locations to visit whilst travelling abroad. Users can also use Google Maps and Google Street View to explore museums using 'Art Project'.

Google Maps also provides indoor mapping. Indoor maps are created by submitting a floor plan of a building to Google, who then turns the floor plan into an interactive map. Users can then navigate around the map by zooming in and out to access the different floors within the building. However, Google is only currently creating indoor maps for 'high density locations such as airports, shopping centres, stadiums and public transport stops and stations' (Google Maps b), meaning that it is not possible to create an indoor map using Google Maps for the university building.

#### **2.5.2.1. Dijkstra's Algorithm**

Dijkstra's algorithm is a type of graph search algorithm that is often used in route planning as it finds the shortest path from the initial state to the goal state (P, Bharuch et al. SlideShare 2015, p.3). However, as it performs a blind search, running this algorithm can be time consuming (P, Bharuch et al. SlideShare 2015, p.29). All edges must also have a positive weighting, as the algorithm cannot handle negative weighted edges and, as a result, may not find the shortest path.

Using the shortest path algorithm may not be the best solution for navigation within the Queen's building. For example, the shortest path may be the path with the most obstacles, which can become dangerous. As well as this, the shortest path may not take into account a route for students with disabilities who need to use things such as elevators in order to help them to get to their destination.

#### **2.5.2.2. Pseudocode** (University of Washington Computer Science and Engineering.)

Below is the pseudocode for Dijkstra's Algorithm.

Initialise the cost of each node to infinite.

Initialise the cost of the source to 0.

While there are unknown nodes left in the graph.

    Select the unknown node A with the lowest cost.

    Mark A as known.

    For each node B adjacent to A

        B's cost = min(B's old cost, A's cost + cost of (A,B))



### **2.5.3. Wayfindr**

Wayfindr is an open standard for audio navigation using mobile devices (Wayfindr). Wayfindr CEO, Umesh Pandya said that Wayfindr wanted to create 'an open standard for everyone to use and benefit from and make the daily lives for visually impaired people better, by allowing them to navigate the world independently' (M, Burgess 2015).

Many visually impaired people rely on public transport to get around as they cannot learn to drive. Wayfindr was created in order to help visually impaired people navigate around public transport. For example, in 2015, a prototype of Wayfindr was tested in Euston Station in London (RSBC 2015). The prototype consisted of an application for a mobile device that would interact with Bluetooth emitting beacons installed throughout the station and then provide the user with audio directions. This solution helps to empower people with visual impairments, making them more independent.

One advantage of the open standard is that it gives venue owners and digital navigation services the tools to implement a high quality, consistent, audio navigation solution. However, Wayfindr are still very much in the development stage and this solution is not yet public, meaning that it cannot be used for solving the problem of navigation within the Queen's building.

In order to carry out a feasibility study and develop a prototype of an application to help both students and staff navigate around the building, research into what to include within a feasibility study must be undertaken. Research on the implementation of the two potential solutions must also be carried out. The potential solution must be measured in a quantifiable way to determine if it is successful in helping users to navigate around the building more effectively.

### **3. Feasibility Study**

Below is a detailed description of the two potential solutions to the navigation problem within the Queen's building and a feasibility study of each. The feasibility study will help to determine the most viable solution to take forward into the implementation stage of the project. The study will cover technical, legal, operational, schedule and financial feasibility of both potential solutions.

#### **3.1. Potential Solution One – Mobile Application using iBeacon Technology**

##### **3.1.1. Technical Feasibility**

'Apple's iBeacons technology is merely the start of a new approach to location-aware apps' (G. Gruman 2014). The iBeacon standard was developed by Apple in 2013 (J, Sieck et al. 2014). iBeacons are small Bluetooth Low Energy (BTLE) devices that can send out signals to mobile applications and calculate an individual's position as an approximation of the proximity between an iBeacon and a mobile device (iBeacon Insider 2014). Using iBeacons and the iBeacon standard as part of the potential solution is viable as Apple is very likely to stay in business and they provide lots of guidance and support for developers (Apple b).

The following information is sent as a signal from an iBeacon in order to identify it:

- A universally unique identifier (UUID) which is sixteen bytes long and identifies the iBeacon.
- A Major, which is two bytes long and can be used to identify or group iBeacons.
- A Minor, which is two bytes long and can be used to identify or group iBeacons (Apple a 2014).

Before an iBeacon can be used, it must first be calibrated. The calibration process is straightforward, and therefore it is easy to learn how to calibrate and deploy an iBeacon. In order to calibrate the iBeacon the following steps must be followed:

- Install the iBeacon and have it emitting a signal.
- Repeatedly sample the signal strength at a distance of one meter for a minimum of ten seconds using an iPhone.
- Collect the values that are reported in the CLBeacons RSSI property. This property is used to measure the distance between the iBeacon and a mobile device.
- Calculate an average of the values reported to be the measured power value.
- Apply the measured power value to the iBeacon.

To use iBeacons for indoor navigation within the Queen's building, iBeacons should be placed at several points throughout the building (for example, points where a user has to make a decision about which way to go next). iBeacons can be placed virtually anywhere due to them being able to run on coin cells, so they are not restricted by being placed near a

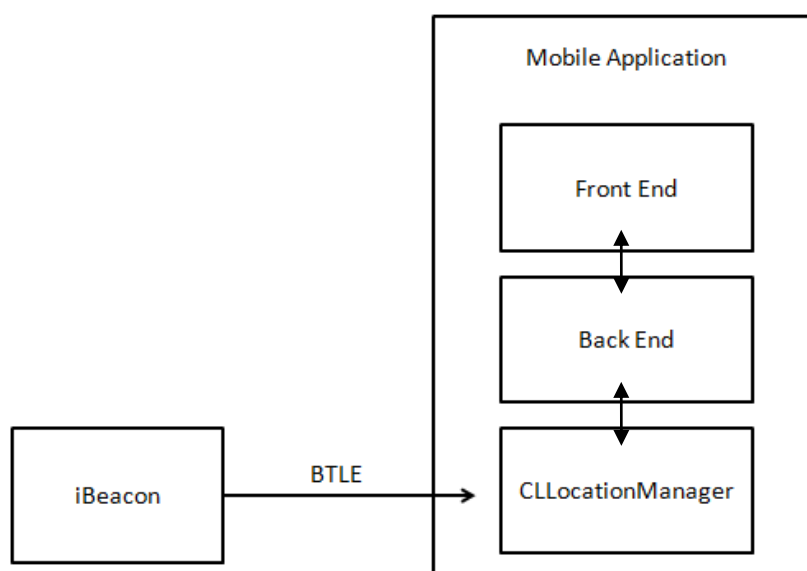
plug socket. These iBeacons will then send the information mentioned above as a signal to a mobile device using Bluetooth. In order to use iBeacons with a mobile application a CLLocationManager should be created. A mobile application built specifically for the purpose of indoor navigation within the Queen's building will use the CLLocationManager to interpret the incoming signal and update the application in response. The user interface for the mobile application will include a map of the building, and the user's location will be updated on the screen as the user makes their way around the building, making the application a live map. The application will allow a user to input a desired location, and it will then provide on screen directions, such as 'left' and 'right' in order to guide the user to the location, as well as a map so that users can visualise their route. (See appendix G, H, I and J for examples of some user interface designs of how the application may look.)

The user interface itself will be self intuitive, and will take inspiration from other map applications, such as Google Maps, making it familiar to the user, so it is easy for them to learn how to use it.

In order to determine how many iBeacons would be required to create an application that covers the whole of the Queen's building (see appendix K, where the Queen's building consists of the buildings highlighted), the Infsoft Beacon calculator was used (Infsoft a). The approximated area of the Queen's building is 9240m<sup>2</sup>. For a building of this size, the Infsoft beacon calculator suggests that between 140-160 beacons should be used to maintain accuracy.

To avoid writing two separate mobile applications for iOS and Android, Xamarin (Xamarin) could be used, which allows the creation of native applications for all platforms. However, it requires knowledge of C#, which is not easy to learn in a short amount of time.

Below is an example of the architecture of the system:



The diagram above shows how the iBeacon uses Bluetooth Low Energy to interact with the mobile application. The CLLocationManager interprets the information from the iBeacon, and then passes it onto the back-end of the application. The back-end of the application then uses this interpreted information to determine what directions to pass on to the user.

The directions are then passed to the front-end of the application, which the user interacts with.

In order to create this solution the following components are required:

Component:	Quantity:	Where from:	Comments:
iBeacon	140-160	<a href="https://www.amazon.co.uk/iBeacon-i4-Bluetooth-Programmable-Beacon/dp/B019G0VS64/ref=sr_1_1?ie=UTF8&amp;qid=1487439414&amp;sr=8-1-spons&amp;keywords=beacon&amp;psc=1">https://www.amazon.co.uk/iBeacon-i4-Bluetooth-Programmable-Beacon/dp/B019G0VS64/ref=sr_1_1?ie=UTF8&amp;qid=1487439414&amp;sr=8-1-spons&amp;keywords=beacon&amp;psc=1</a>	Each beacon comes with 1 cell battery.  Social Retail provides lots of online support for developing with their range of beacons. ( <a href="http://www.digitalsocialretail.com/help/">http://www.digitalsocialretail.com/help/</a> )
Cell batteries	560-640 for a year, as they need to be replaced every 3 months or so.	<a href="https://www.amazon.co.uk/Panasonic-CR2477-Lithium-Battery-Batteries/dp/B01GGW82OQ/ref=sr_1_15?ie=UTF8&amp;qid=1487508758&amp;sr=8-15&amp;keywords=CR2477+battery">https://www.amazon.co.uk/Panasonic-CR2477-Lithium-Battery-Batteries/dp/B01GGW82OQ/ref=sr_1_15?ie=UTF8&amp;qid=1487508758&amp;sr=8-15&amp;keywords=CR2477+battery</a>	CR2477 batteries are compatible with the above iBeacons.
iPhone 4S for calibration	1	<a href="https://www.amazon.co.uk/Apple-iPhone-4S-16GB-SIM-Free/dp/B005VTIHFA">https://www.amazon.co.uk/Apple-iPhone-4S-16GB-SIM-Free/dp/B005VTIHFA</a>	Cost: £73.72 for a previously used device, correct as of 19/02/2017.
Mobile application: - User interface - Way of dealing with information from iBeacon	1	N/A – Must make the mobile application from scratch.	Xamarin will be used, as it enables native coding for both iOS and Android. ( <a href="https://www.xamarin.com/">https://www.xamarin.com/</a> )

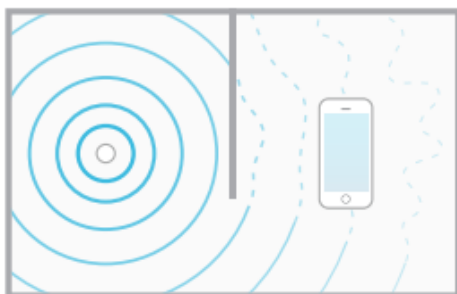
Both Apple and Android devices support iBeacon technology. However, Windows does not. This would be a problem for the 5% of Windows OS users within the university.

There are two functions that can be used to interact with iBeacons: Region monitoring and a ranging function. Region monitoring is used to send push notifications to a device when it enters or exits the 'region' of an iBeacon. In the future, this could be used to send 'special offers' from the canteen for example, when an individual passes by. The ranging function is used to determine the proximity of a mobile device to an iBeacon. Only the ranging function will be used for determining the users' location. Accuracy is calculated by determining the strength of the signal from the iBeacon to the mobile device. This distance is given in meters. The more accurate the distance, the higher the signal strength. A new set of

Application Programming Interfaces (API's) were introduced with iOS 7. These new API's allowed ranging, which allows the estimate of proximity to be determined as one of the following:

- Immediate – This signals that the mobile device is physically very close to the iBeacon. This has the highest level of confidence.
- Near – This signals that the mobile device is within a clear line of sight of the iBeacon (within 1 to 3 meters of the iBeacon).
- Far – This signals that the iBeacon can be detected but the confidence is low, so cannot determine whether the device is near or immediate.
- Unknown – This signals that the mobile device cannot be detected.

However, there are a number of restrictions when working with iBeacons. For example, as iBeacons are an Apple standard, the data format provided by Apple must be used. Bluetooth makes use of 2.4GHz signals. These types of signals are known for having problems with 'damping'. 'Damping' the signal refers to when the signal is reduced as a result of something coming between the iBeacon and the mobile device, affecting its performance. Walls and humans can be a cause of dampening. Dampening could definitely affect the created solution, as there are many different rooms and floors within the building, so lots of walls which could cause dampening. As well as this, it is a university building, so lots of human bodies, which could also affect the signal. The images below show how both walls (Figure 3 – Apple a 2014) and human bodies (Figure 4 – Apple a 2014) can cause dampening and the affect it can have on the signal and its accuracy:



**Figure 3:** Physical objects and materials can block signals, reducing the received signal strength.



**Figure 4:** Human bodies can also block signals.

As iBeacons rely on Bluetooth Low Energy, the solution will only work with iPhone 4S, iPod Touch (5<sup>th</sup> generation), iPad (3<sup>rd</sup> generation) or iPad Mini or later. Another restriction is that the Android OS doesn't support iBeacon detection when the application is closed. This means that the application must remain open so that the application can receive information from all iBeacons encountered, whereas the application doesn't need to be open in iOS, as the OS automatically looks for iBeacons if both the Bluetooth and location service is enabled. When an iBeacon is detected, a notification is pushed by the application. Having to keep Android mobile devices open for the solution to work could be a safety problem as 57.50% of students owned an Android device. This could become a problem whilst walking up and down stairs, as people may pay more attention to their mobile devices instead of concentrating on where they are going, making it more likely that collisions or trips could occur. Some students may not be able to multitask, and as a result may refrain from using the application.

Coin cells can work for up to two years, depending on the signal strength and advertising interval. However, if a higher performance is required from the iBeacon then the coin cell life will be reduced to just a few months. These batteries would have to be replaced fairly frequently, meaning some amount of 'down-time' would have to occur. This would also mean that someone would have to be in charge of changing the batteries. This will also make the solution more expensive.

If a user has not allowed background application refreshment, then the application will not receive notifications each time the mobile device moves within or out of range of an iBeacon. This could result in a user not getting the most accurate and up-to-date location on the mobile map, which could mean that the user takes the wrong direction and not find the destination.

To complete this project the following people will be required:

- Developer – As the developer, a strong technical ability is required. A knowledge of creating mobile applications and the programming language C# is required. This would be very hard to achieve with just 5 weeks of implementation.
- Testers.

The following factors could affect the creation of the potential solution on-budget and on time:

- Technical capabilities of the developer may prevent solution from being completed, or completed on time, as they may need additional training in the appropriate programming language.
- Illness of the developer may mean that deadlines are not met.
- Developer managing workload, along with workload of other modules may result in solution not being finished on time or to as high a standard as it could be.
- Cost of purchasing both beacons and batteries may result in being over-budget
- The unavailability of components could mean that implementation has to be postponed.
- The shipping time of components could mean that implementation has to be postponed.

This solution has the potential to be scaled to every university building. However, it would be very expensive.

Due to the developer having no prior knowledge of creating mobile applications or using the programming language C#, further training would be required in order to start work on the solution. Therefore it is not technically feasible to create this potential solution within the timeframe.

### **3.1.2. Legal Feasibility**

Many developers may not be aware that in order to call your device an iBeacon you must first acquire a license from Apple. User authentication is required in order to use the Core

Location API. Users can change the privacy settings of an application at any time, as all applications that use the iBeacon region API are present in Settings.

One legal aspect to consider is notification as unsolicited advertising (Wilde Beuger Solmecke 2014). There is a lot of debate about whether installing an application means consent. Users must actively opt-in to give their consent to receive proximity marketing notifications. These notifications must not affect a users' freedom of choice.

When using iBeacons there is no bidirectional communication between the iBeacon and the mobile application. So the iBeacon does not receive any information from its users. Only the mobile application will be sent data that will be processed. As this data constitutes as personal data, the Data Protection Act must be considered. Information regarding the purpose of the data processing must be given to individuals and they must give their consent for their personal data to be processed.

Creating this potential solution is legally feasible.

### **3.1.3. Operational Feasibility**

Below is a list of both the functional and non-functional requirements for the proposed solution:

#### **Functional Requirements:**

- The solution should provide the user with a visual representation of the indoors of the Queen's building.
- The solution should allow a user to see where they currently are on the visual representation.
- The solution should allow the user to input a desired location.
- The solution should provide the user with easy to follow instructions of how to get to the desired location.
- The solution should retrieve data from iBeacons within the building.
- The solution should process the data retrieved from the iBeacons in order to provide useful information to the user.
- The solution should be able to tell a user when they are going the wrong way.
- The solution should provide a list of room names so that a user can choose a location to navigate to.

#### **Non-functional Requirements:**

- The solution should provide users with directions within a few seconds of the user submitting their request.
- The solution should be easily expanded to include other buildings of the university within Cardiff.
- The solution should always be able to retrieve directions to a location
- The solution code should be easy to keep up-to-date (eg, room number changes etc).
- The solution should protect a users' private data.
- The solution should be intuitive and easy to use.
- The solution should work on both Android and iOS operating systems.

The proposed solution satisfies all of the above functional requirements. This solution will effectively resolve the problem with navigation within the Queen's Building as the user interface provides a map of the building, which will allow users to visualise where they are within the building as it updates in response to their movements due to the mobile application interacting with iBeacons. The user can also input a desired destination, and the application will then relay directions on how to get there. This will help to reduce confusion and the number of students who get lost within the building.

In order to support the operation of the proposed solution, the following tools are required:

- iPhone 4S or later to calibrate the iBeacons.
- Xamarin to edit the code for a mobile application native for both Android and iOS.
- Replacement coin cells to ensure that the iBeacons continue to work accurately.

In order to continue the successful operation of the proposed solution someone will need to be trained in iBeacon maintenance to ensure they continue to run accurately. As well as this, someone would need to be trained in the C# programming language so that the code can be edited and kept up-to-date. Anyone in charge of maintaining the application will also need to be trained on how to use the application correctly. The following documentation would be provided to the support team to ensure that they are prepared to deal with different aspects of the solution:

- Documentation provided by Apple on iBeacons.
- Documentation on Xamarin.
- Documentation on C#.

Users of the application would be able to access a guide which would detail how to install the application as well as how to use all features of the application successfully. The university could also provide training to all students on how to install and use the application correctly. The application itself could also have a 'walkthrough' when the user first interacts with it that explains all the key features included.

The university supports the idea of an application to improve navigation within the building. Both staff and students within the university are not happy with the current wayfinding system due to the signs being confusing and the room numbers not being intuitive. As shown from the tests carried out previously, it takes someone who has not been to the building before 19 minutes and 21 seconds to find a specific room when asked, whereas it only takes an experienced third year individual 3 minutes and 8 seconds. This proves that the current wayfinding system within the building is sub-par and that the proposed solution could help to improve this. The students who were surveyed also gave their opinions on how they saw the problem being overcome. Most of the students agreed that a mobile application of some kind would be the most appropriate solution to the problem, so the application would be widely used.

The operation of the proposed solution relies heavily on the iBeacons working and being compatible with the mobile application, as well as the potential users having mobile devices



that are compatible with the application. The proposed solution can be easily maintained. The batteries within the iBeacons only need to be changed every few months and the application code can be easily edited to change room names etc. There is also lots of online support for both the iBeacons and the programming language for the mobile application.

The application will be easy to use as the user interface will be self intuitive, as it takes inspiration from existing mapping applications, so it will be familiar to the user.

However, iBeacons are not intended to be used for specific location identification due to the issues around signal strength and the variables within the deployment environment, so this could affect the operation of the application. The proposed solution does not fit in with the development schedule as only 5 weeks have been set aside for implementation, including both the coding of the mobile application and the installing and setting up of iBeacons. As well as this, the application is not disposable as it will be costly to implement, especially if it were to be expanded to include the whole building. As a result of this, the proposed solution is not operationally feasible.

### **Risk Assessment**

Below is a risk assessment for the potential solution:

<b>Risk</b>	<b>Impact</b>	<b>Likelihood</b>	<b>Solution</b>
iBeacons are not in stock, so shipping is delayed, which will affect implementation of the potential solution.	High.	Low.	Purchase iBeacons from a different online vendor.
Coin cells are out of stock, so shipping is delayed, could affect replacing iBeacon batteries.	Low.	Low.	Purchase coin cells from a different online vendor.
iPhone 4S is out of stock, so shipping is delayed, meaning that iBeacon calibration is delayed.	Medium.	Low.	Purchase iPhone 4S from a different online vendor, or borrow an iPhone 4S from a willing individual.
Apple and other online communities are down, so hard to retrieve iBeacon support or coding support when needed.	Low.	Medium.	Try and find other suitable online resources or ask someone within the university for help.

Laptop stops working, which could affect coding.	Medium.	Medium.	Go to the university and use computers there.
Illness, which could affect implementation of potential solution.	High.	Low.	Work from home.
Get stuck during implementation, which could affect its completion.	Medium.	Medium.	Consult online resources and communities to try and resolve the problem, or talk to supervisor for advice.

#### 3.1.4. Schedule Feasibility

According to research carried out by AYTU and Kinvey (K, Rice. Kinvey 2013) where 100 iOS, Android and HTML5 developers were surveyed, it takes an average of 18 weeks to build a mobile application from scratch, not including getting the mobile application to communicate with the iBeacons. As well as this, each iBeacon would need to be separately installed and calibrated. As the individuals surveyed were professional developers, it would take someone with no mobile development experience considerably longer to create a mobile application from scratch. The amount of time allocated to implementation within the initial plan was 5 weeks, so creating a mobile application within this time frame is not feasible.

#### 3.1.5. Financial Feasibility

Potential Costs	Potential Benefits
<ul style="list-style-type: none"> <li>Cost of purchasing iBeacons (Need between 140-160): £2938.60 - £3358.40.</li> <li>Cost of replacing coin cells every few months: £1850.80 - £2115.20.</li> <li>Cost of iPhone 4S for calibration: £73.72 used.</li> <li>Cost of hiring someone to maintain iBeacons: Average wage for a 6 hour shift would be £45 (Minimum Wage)</li> <li>Cost of incentivising testers: Could give them pay for an hour of their time, so minimum wage £7.50.</li> <li>Cost of attending training courses to</li> </ul>	<ul style="list-style-type: none"> <li>Bulk order of iBeacons could make purchase cheaper.</li> <li>Bulk order of coin cells could make purchase cheaper.</li> <li>Increased revenue, additional sales attained from offer notifications for the university canteen.</li> <li>Improved navigation within the Queen's Building.</li> <li>Increase in percentage of students attending lectures.</li> <li>Decrease in percentage of students/lecturers late to lectures.</li> </ul>

improve technical skills: Some free courses in C# are available (eg, <a href="http://csharp.net-tutorials.com/">http://csharp.net-tutorials.com/</a> ). • Increased student/ lecturer dissatisfaction from fear of change.	
Maximum Total Cost: £5599.82	

The solution will cost £5599.82. The solution will give the university benefits of increasing their income through the use of advertisements and will ensure that students and staff can find their way around the building easily. The benefits do not out-weigh the costs, therefore the proposed solution is not financially feasible as the costs are never likely to be recovered.

## 3.2. Potential Solution Two – Social Media ‘Chat-Bot’

### 3.2.1. Technical Feasibility

In 2016 Facebook launched its’ Messenger platform which provides a way for developers to create Bots. Last year, Microsoft’s Satya Nadella said that ‘Bots are the new apps’ (D, Lee 2017).

Facebook Bots allow their creators to connect with anyone who uses the mobile Messenger platform through message threads (S, Rosenberg 2016). Anyone with a developers account can create their own Bot. When a Bot is created, it is native for all platforms that can use the Messenger platform, meaning that it works for iOS, Android and Windows. Using a Facebook Bot as part of the potential solution is viable as Facebook is highly likely to stay in business and they also provide lots of guidance and support and readily encourage developers to use their platform (Facebook for developers a 2016).

A Bot has three main capabilities:

- Send/ Receive API for images, text and CTA’s.
- Generic Message Templates.
- Welcome screen + null state CTA’s.

In order to create a Bot the following steps must be followed:

- Create a Facebook App and Page.
- Create a Messenger Bot for the above App and Page.
- Set up a server.
- Setup a Webhook to allow the Bot to receive messages from users.
- Generate a Page Access Token.
- Subscribe the App to the Page.
- Write code to send and receive messages using JavaScript.

This shows how easily a Bot can be created and deployed.

A Facebook Bot could be used as part of a navigational planner, where students and staff at the university add the Bot as a contact within the Messenger application using the Messenger Username. The Bot can be used 24/7. It can be used throughout the whole of the Queen's building as it uses WiFi/ a mobile network. A user will then be greeted with the Bots welcome screen where they can then click a call-to-action button in order to start the conversation. Users can then send a message to ask the Bot for directions to a desired location over WiFi (eg, I am at T/2.09 and I want to go to S/2.21). The Bot would then respond with a set of directions for the user to follow, as well as images that will act as visual clues along the way to reassure the user that they are going in the right direction. If a user gets lost, they can easily write another request to the Bot, who will then generate a new set of directions for the user to follow. Bots can use natural language processing in order to understand what a user has said. The user interface is intuitive and easy to use as it makes use of the Messenger platform. (See appendix L for an example of how the user interface will look.) Most potential users will have the Messenger application already installed on their mobile phone, meaning that there is no need for a separate application to be installed, and they will already know how to use and interact with it.

Below is an example of the architecture of the system:



The diagram above shows how the user sends a request for directions from their mobile device to the Facebook Bot over a Wifi or cellular (eg 4G) connection. The Bot then processes this request and sends back a list of directions to the users' device over WiFi.

In order to create this solution the following components are required:

Component:	Quantity:	Where from:	Comments:
Facebook Page	1	Facebook	To be created
Facebook App	1	Facebook	To be created
Bot	1	Facebook	To be created
Server	1	Heroku	To be created

To successfully launch a Bot the following steps are required:

- Developers must complete the pre-launch checklist. The pre-launch checklist is there to ensure that developers include a 'Get Started' CTA so that a user can easily start a conversation with the Bot even if they don't know what to say initially. A greeting should be included so that users know what service the Bot provides. Developers should ensure that their conversations are personalised, and also ensure that users have the option to turn off notifications at any time.
- Developers must test their Bot. Things such as responsiveness, fail states and user journeys should be considered.
- Developers must submit their App for review so that it can be tested by the Facebook team to ensure it meets their policies.
- Developers must then make their App public and publish the Facebook page (Facebook for developers b 2016).

There are a number of different ways in which a developer can interact with a user. However, each option has restrictions according to the policies associated with the Messenger platform. Developers can choose to interact with users using either standard messaging or subscription messaging (Facebook for developers c 2016). A user must initiate the conversation with a Bot. After initiating the conversation, using standard messaging, the developer has a twenty four hour period in which they can send and receive messages with the user. The developer can send one additional message to the user outside of the twenty four hour period. The user can block or mute a conversation with a Bot at any time. Using standard messaging requires the `pages_messaging` permission.

Subscription messaging allows a developer to send regular messages to users that they have subscribed to. Subscription messaging is not subject to the same restrictions as standard messaging. Subscription messages can be sent outside of the twenty four hour time window. However, subscription messaging cannot be used to send a user promotional content. These types of messages can be used for things such as managing personal productivity such as tasks from calendar events. For example, this could be used to integrate a user's timetable and calendar events, so that the Bot could tell them which room they need to go to next and ask them if they need directions. As well as using subscription messages for productivity, they can also be used for sending university news. Using subscription messages requires the `pages_messaging_subscription` and `pages_messaging` permissions.

There are also a number of tools that a developer can use in order to inform a potential user of what service the Bot provides, how it works and why it should be used. Developers can use Messenger Greetings to explain what the Bot does (Facebook for developers d 2016). However, this greeting disappears after the user starts interacting with the Bot, so instructions on how to use the Bot should not be detailed here. The Messenger Greeting can only be a maximum of one hundred and sixty characters, so it should be fairly concise. The Get Started CTA should follow from the Messenger Greeting. Developers can also create a Welcome Message which details the functionality of the Bot so that users know how to interact with it.

Developers can choose to send textual messages, photographs and structured messages. Sending too many textual messages at once can be overwhelming to the user, so a developer should try and keep messages concise and send a few smaller messages so that they are easier to follow and not too intimidating. Photo messages should be of a high quality. Structured messages provide templates for sending messages. There are three different types of structured messages that can be used:

- Button – The Button template allows the developer to add CTA's to their messages. All CTA's should only be between one or three words in length and under twenty characters. CTA's should start with a verb that describes the action that they are linked to.
- Generic – The Generic template allows for a combination of text, buttons and images. The Generic template should only be used for messages that are consistent.
- Carousels – The Carousel template is used for sending multiple similar messages at the same time. All messages on the carousel should be consistent.

A developer should ensure that they create fail state messages to let a user know that the Bot does not understand what they are asking. The developer should ensure that they create multiple messages for each fail state as they can be open to interpretation and a user may not understand one. These messages should help a user understand where they went wrong. As well as this, the guidelines also state that a developer should make their messages conversational by using contractions, addressing the user by name and writing in second person so that it feels more personal.

As the solution relies heavily on the use of WiFi it is also important to consider what might affect the WiFi within the Queen's building. Construction materials can potentially stop the WiFi signal from going as far as it can (Packetworks). Metal, concrete and brick are amongst the materials that can potentially affect the WiFi signal. As well as physical interference, a WiFi signal can also be affected by different types of frequencies. Any phone that uses a 2.4 GHz frequency could potentially affect WiFi signal. As well as this, other wireless devices, hearing aids and bodies can affect Wifi signal.

To complete this project the following people will be required:

- Developer - As the developer, a strong technical ability will be required. A knowledge of JavaScript is required. However, there is lots of online documentation on creating Bots and using JavaScript, so it could definitely be achieved in the 5 weeks scheduled for implementation.
- Testers.

The following factors could affect the creation of the potential solution on-budget and on time:

- Technical capabilities of the developer may prevent solution from being completed, or completed on time.
- Illness of the developer may mean that deadlines are not met.

- Developer managing workload, along with workload of other modules may result in solution not being finished on time or to as high a standard as it could be.
- Facebook not being 'online' could prevent the solution from being completed.

This solution has the potential to be scaled to every university building. Due to the high volume of documentation provided by Facebook for developers, as well as lots of online communities that can help with JavaScript queries, this solution is technically feasible.

### **3.2.2. Legal Feasibility**

Many users may not be aware that when engaging in conversation with a Bot, not only does the Bot 'read' the conversation, but so does the developers and Facebook as its servers log communications between users and Bots (B, Ondrisek 2016). A user must actively choose to engage in a conversation with a Bot, so they are free to make an informed choice. As with the previous potential solution, notification as unsolicited advertising must be considered. The user may also block or mute the Bot at any time.

In November 2014, The Electronic Frontier Foundation created a Security Messaging Scorecard in order to determine just how secure messaging applications really are. On the Messenger platform data is encrypted during transit, meaning that it cannot be intercepted and read. However, as mentioned above, the data can be read by Facebook. You cannot verify the identity of a contact, but most social media messaging platforms do not provide this. Your past communications are not secure if you lose your key. The code used on this platform is not open to independent review. The security design of Messenger is not properly documented. However, there has been a recent code audit. As this scorecard was created in 2014, and Bots have only recently been released, Facebook have obviously put more thought into their users' privacy and data protection.

Bots have access to the following user information:

- First and Last Names.
- Gender.
- Timezone.
- Locale.
- Age (using a workaround).
- Country (using a workaround).
- Username (using a workaround).
- GPS location (it has to be shared explicitly on request).

As this information is deemed as personal information, the Data Protection Act must again be considered for this potential solution. However, creating this potential solution is legally feasible.

### 3.2.3. Operational Feasibility

Below is a list of both the functional and non-functional requirements for the proposed solution:

#### Functional Requirements:

- The solution should allow the user to initiate a conversation.
- The solution should provide clear instructions on how to interact with it.
- The solution should allow the user to input a start location and a desired location.
- The solution should take a start location and desired location and determine a route.
- The solution should provide the user with easy to follow instructions of how to get to the desired location, including images to ensure that the user is on the right path.
- The solution should allow a user to tell it when they are lost.
- The solution should take a new start location and determine a new route when a user is lost.
- The solution should allow a user to block or mute the conversation at any time.

#### Non-functional Requirements:

- The solution should respond to a users request with directions within a few seconds.
- The solution should be easily expanded to include other buildings of the university of Cardiff.
- The solution should always be able to provide a user with directions.
- The solution code should be easy to keep up-to-date.
- The solution should protect a users' private data.
- The solution should be intuitive and easy to use.
- The solution should work on all OS platforms.

The proposed solution satisfies all of the above requirements. This solution will effectively resolve the problem with navigation within the Queen's building as it allows users to plan their journey before visiting the building, or use the application as they are navigating the building in order to retrieve directions. The images will help to ensure that the user knows they are going the correct way. The Bot should help to reduce the number of students who get lost within the building and the amount of time it takes for new students to get to their lectures.

In order to support the operation of the proposed solution the following tools are required:

- Facebook developers account to create and maintain the Bot and its code.

In order to continue the successful operation of the Bot someone will need to be trained in Node JS so that the code can be edited and kept up-to-date (eg when rooms change etc). Anyone in charge of maintaining the Facebook Bot will also need to be trained in how to use the Bot correctly themselves. The following documentation would be provided to the support team:

- Documentation provided by Facebook on creating and maintaining Bots.
- Documentation on Node JS and JavaScript.



Users of the application could be given a guide that explains how to interact with the Bot. The Bot itself can also help to explain how to interact and use its functionality. Again, the university could also provide training to all students on how to use the Bot correctly.

The university supports the idea of a social media Bot to help improve navigation within the building. As mentioned before, both staff and students within the university are not happy with the current wayfinding system due to the signs being confusing and the room numbers not being intuitive.

The operation of the Bot relies heavily on a WiFi connection and its potential users having mobile devices that are compatible with the Messenger application. The Bot can be easily maintained through the developer account and there is also a lot of online documentation available from Facebook, as well as other sources, for creating bots and coding with JavaScript and Node JS.

The application will be easy to use as the user interface is exactly the same throughout the Messenger application and it is self intuitive. As most of the potential users will already have the Messenger application installed, they will be familiar with its interface and will know how to interact with the Bot.

The proposed solution fits in with the development schedule as 5 weeks have been set aside for surveying the Queen's Building and the implementation of the Bot. The application is disposable as it is very cheap to implement. As a result of this, the proposed solution is operationally feasible.

### Risk Assessment

Below is a risk assessment for the potential solution:

Risk	Impact	Likelihood	Solution
The Facebook platform is down so it is impossible to access Bot documentation.	Medium.	Low.	Find other online resources for documentation on Bots.
Github is down so cannot code online.	High.	Low.	Code offline.
Server is down so application doesn't work.	High.	Low.	Find an alternative server to host on.
Facebook does not approve my Bot.	High.	Low.	Ensure all guidelines set out by Facebook are followed.
The Bot provides directions that are not very useful.	Medium.	Medium.	Provide different ways of saying each direction and allow

			for users to ask the Bot for clarification. Could also get testers to also verify if directions are unclear.
Directions provided by the Bot are out of date.	Medium.	Low.	Go and survey the university each time there are room changes etc and provide new directions.
Laptop stops working, which could affect coding.	Medium.	Medium.	Go to university and use computers there.
Illness which could affect implementation of potential solution.	High.	Low.	Work from home.
Get stuck during implementation which could affect its completion.	Medium.	Medium.	Consult online resources and communities to try and resolve the problem, talk to supervisor for advice.
The universities WiFi does not work.	High.	Low.	Users would have to use signs within the building to navigate. Could work with the university to make the signs better.

### 3.2.4. Schedule Feasibility

Facebooks' guide for setting up a Bot is called 'Getting Started' and it says that you can 'create a bot in 10 minutes using Node JS'. It seems unlikely that a developer with no previous experience of creating a Bot can create their own in such a small timeframe, but a number of online sources suggest the initial setup will only take a day. After that, it's down to the coding and how detailed you want the Bot to be. The review process for getting a Bot approved by Facebook so that it can be used by the public usually takes around twenty four hours. Only one Bot needs to be created as it is native to all OS that can use the Messenger platform. The amount of time allocated to implementation within the initial plan was 5 weeks, so it is feasible to create a Bot within this timeframe.

### 3.2.5. Financial Feasibility

Potential Costs	Potential Benefits
<ul style="list-style-type: none"><li>• Cost of incentivising testers: Could give them pay for an hour of their time, so minimum wage £7.50.</li><li>• Increased student/ lecturer dissatisfaction from fear of change.</li></ul>	<ul style="list-style-type: none"><li>• Increased revenue, additional sales attained from offer notifications for the university canteen.</li><li>• Improved navigation within the Queen's building.</li><li>• Increase in percentage of students attending lectures.</li><li>• Decrease in percentage of students/ lecturers late to lectures.</li></ul>
Maximum Total Cost: £7.50	

The solution will cost a maximum of £7.50 per application tester. The solution will give the university benefits of increasing their income through the use of advertisements and will ensure that students and staff can find their way around the building easily. The benefits definitely outweigh the cost; therefore the proposed solution is financially feasible as the costs can be easily recovered.

## **4. Implementation of a Prototype**

This section of the report will detail how the implementation of the prototype for the chosen solution was created.

### **4.1. Chosen Solution**

The potential solution that is deemed most feasible is the social media Bot. The Bot passed the technical, legal, operational, schedule and financial feasibility study, whereas the iBeacon potential solution only passed the legal feasibility study.

As the developer, the Bot is the most feasible solution technically as previous experience of using JavaScript has been acquired. Bots are an emerging technology and will only get bigger and better. Facebook are actively encouraging developers to create applications on their platform, meaning that this solution will also have a relatively small cost. As well as this, it also means that there is lots of documentation readily available to help developers. Creating a Bot using the Messenger platform also means that the Bot can be used on any OS that supports it, so this will not limit who can use the application. The majority of people will already have the Facebook Messenger application installed on their device, so it eliminates the need for a new application to be installed and it means that the user will already know how to use it. This solution is definitely achievable within the 5 weeks set aside for implementation and it will help users to navigate around the building more effectively.

We have chosen not to take the iBeacon mobile application further as it will be incredibly difficult to implement due to not having the required skills and it will definitely take longer than the 5 weeks allocated. The cost of creating this solution is also large in comparison to the Bot solution. As well as this, iBeacons will not work with the Windows OS, so this solution would not work for all university students, iBeacons are also susceptible to dampening, which could affect the workings of the solution. iBeacons are not intended to be used for specific location identification, so they may not work as planned.

### **4.2. Implementation**

The prototype will consist of the routes that are needed by first year Computer Science students, as they are the students who have the most to benefit from using the application.

From looking at the timetable of first year Computer Science students it was determined that the following rooms would be included within the prototype for the navigational Bot:

- C2.04
- C2.07
- S1.32
- S2.21
- S2.22
- N4.07

- T2.07
- T2.09
- Library
- Canteen
- Toilets

In order to create the directions to be used within the navigational Bot, the university building was surveyed and routes between each room were determined. Pictures were also taken so that they could be included to provide the user with a visual aid to determine if they are going in the right direction.

#### 4.2.1. Example Route

Here is an example of a route that will be included within the prototype from the canteen to the library. This route assumes that the user of the application is already outside of the canteen and that they wish to navigate to the library.

The following directions would then be given to the user in order to direct them to the library:

- 'Take the stairs that are in front of you, and go up 2 flights.'
- Go through the door just to your right. This is the library.'

#### 4.2.2. Creating the Bot

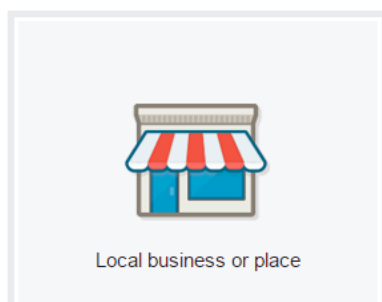
This section of the report details how the Bot will be created. To create the initial Bot the following tutorials were used (A, Jodlowski 2016; Facebook for developers e 2016). These tutorials explain most of the steps that need to be followed.

##### 4.2.2.1. Creating a Facebook App, Page and Messenger Bot

To create a page, a type of page must be selected. Below, the 'Brand or product' type of page was selected and the name 'Cardiff Queen's Building Navigational Aid' was assigned to it:

#### Create a Page

Give your brand, business or cause a voice on Facebook and connect with the people who matter to you.  
It's free to set up. Just choose a Page type to get started.



**Brand or product**

App Page ▼

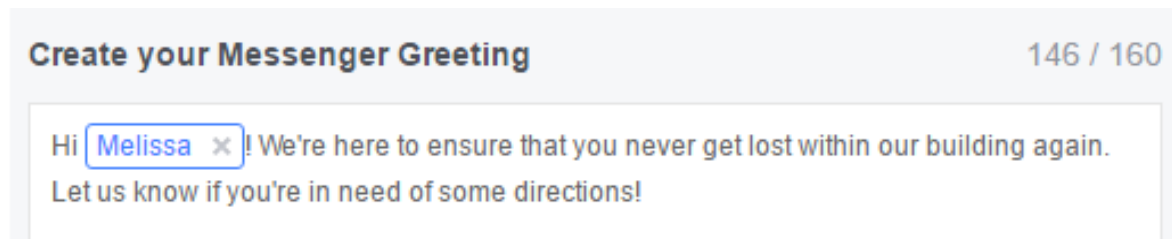
Cardiff Queen's Building Navigational Aid

By clicking Get Started, you agree to the [Facebook Pages Terms](#).

**Get Started**

After creating the initial page it was then possible to add a picture. This profile image will be used to identify the messenger application. The Cardiff University logo was selected so that the Bot was easily identifiable to Cardiff University students and staff (See appendix M).

A messenger greeting was also created. This is a way to communicate the functionality of the Bot to its users. Below you can see the messenger greeting that was created. The users name has also been used to provide a more personalised experience.



To create a Facebook app, an app ID must be generated by using the following URL: <https://developers.facebook.com/apps>. There is a thirty-two character limit when creating an app display name. Originally the name 'Cardiff University Navigation Aid' was to be used as the display name so that it could be easily identifiable. However, after learning about the display name restriction, it was decided that 'CU Navigational Aid' would be used as it fits within the character limit, yet it still instantly recognisable as Cardiff University is often shortened to 'CU'. A contact email address must be provided. The 'Apps for Messenger' category was selected in order to create a Bot (See appendix N).

#### **4.2.2.2. Generating a Token, Setting Up a Webhook and Subscribing the Webhook to the Page.**

In order to setup a webhook, server space must first be acquired. It was decided that Heroku should be used as it was proven to work in the tutorial that is being followed, and they offer free accounts and server space. In order to use Heroku, an account must be made by visiting <https://signup.heroku.com/login>.

Once logged in, a new application was created by selecting the 'New' button from the dashboard (see appendix O), and then the 'Create new App' button. No name was given to this application, so the name 'radiant-beyond-52321' was randomly generated.

Back in the Facebook App, a token must be generated in the token generation tab. The page for which we want to generate a token must be selected (see appendix P). After the page access token was generated, it was then copied to the clipboard.

Back in Heroku, the app 'radiant-beyond-52321' was selected. In the apps settings tab, 'Reveal Config Vars' was selected, and the page access token was pasted in as shown below:

## Config Variables

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

Hide Config Vars

PAGE\_ACCESS\_TOKEN

EAATbYxaZBE24BAMkySv2y6GgMv



Back in the Facebook app, the webhooks tab was selected. 'Setup Webhooks' was selected. The Callback URL for the application was given, along with a verify token. This will be defined in the setup code for the bot:

Edit Page Subscription

Callback URL

https://radiant-beyond-52321.herokuapp.com/webhook

Verify Token

testbot\_verify\_token

Cancel

Verify and Save

As well as the Callback URL, the subscription fields required are also selected (see appendix Q). Back on the Messenger tab, under the webhooks section, the Cardiff Queen's Building Navigational Aid page was selected to subscribe the webhook to as shown below:

Webhooks

Edit events

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Selected events: message\_deliveries, message\_echoes, message\_reads, messages, messaging\_account\_linking, messaging\_checkout\_updates, messaging\_game\_plays, messaging\_optins, messaging\_payments, messaging\_postbacks, messaging\_pre\_checkouts, messaging\_referrals

Complete

Select a page to subscribe your webhook to the page events

Select a Page

Subscribed pages: Cardiff Queen's Building Navigational Aid

A Github repository called 'facebookbot' was then created in order to complete the code for the project. This repository was then connected with the Heroku application (see appendix R).

#### 4.2.2.3. Writing Code to Send and Receive Messages

This section explains the code that was used for the initial setup of the Bot as detailed in the tutorial that is being followed.

```
var express = require('express');
var bodyParser = require('body-parser');
var request = require('request');
var app = express();

app.use(bodyParser.urlencoded({extended:false}));
app.use(bodyParser.json());
app.listen((process.env.PORT || 3000));

//Server frontpage – This allows us to test that the server is running correctly
app.get('/', function(req, res) {
    res.send('This is TestBot Server');
});

//Facebook Webhook – Used to verify that the application can be integrated with Facebook, given the verify token provided.
app.get('/webhook', function(req, res) {
    if (req.query['hub.verify_token'] === 'testbot_verify_token') {
        res.send(req.query['hub.challenge']);
    }
    else {
        res.send('Invalid verify token');
    }
});

//Handler receiving messages – Iterates over the messages received by the Bot, and sends the received text back to the user,
//making use of the sendMessage function below.
app.post('/webhook', function(req, res) {
    var events = req.body.entry[0].messaging;
    for (i = 0; i < events.length; i++) {
        var event = events[i];
        if (event.message && event.message.text) {
            sendMessage(event.sender.id, {text: "Echo: " + event.message.text});
        }
    }
    res.sendStatus(200);
});

//Generic function sending messages – Both the access token and the recipient ID is required in order to authorise the request.
function sendMessage(recipientId, message) {
    request({
        url:"https://graph.facebook.com/v2.6/me/messages",
        qs: { access_token: process.env.PAGE_ACCESS_TOKEN},
        method: 'POST',
        json: {
            recipient: {id: recipientId},
            message: message,
        }
    },
    function(error, response, body) {
        if (error) {
            console.log('Error sending message: ', error);
        }
        else if(response.body.error){
            console.log('Error: ', response.body.error);
        }
    });
};
```

This code was then committed to Github and deployed on the server.



Below shows that the Bot can correctly echo a message back to the user:



#### 4.2.2.4. Writing Code for Navigational Purposes

In order to utilise the Bot for navigational purposes the messenger platform documentation was consulted to determine which layout template would be most appropriate. The messenger platform has many different templates to consider, such as a button template, generic template, list template, receipt template, carousel template and an airline boarding pass template. It was decided that the most appropriate template would be the carousel template (Facebook for developers f 2016). The image within the carousel could be an image of the corridor etc so that the user of the Bot knows that they are going in the right direction. The text within the carousel would then provide the directions for the user to follow. The user would then swipe left or right with their finger in order to reveal the next or previous direction. This is the best choice for a layout as it is self intuitive, and users can easily and quickly navigate between the directions. (For an example of how the carousel template could look see appendix L).

However, the messenger platform documentation does not include any details about implementing the carousel message layout, and after much investigation it was found that the carousel layout for messenger is currently only being trialled in Thailand and Australia (D, Fuller 2017). This means that it is not possible to use this type of layout for the Bot at present.

The second choice for the template is the list template. The list template allows you to present information as a set of items one on top of the other (Facebook for developers g 2016). According to the online documentation there are 2 ways in which a list template can be formatted:

- Option 1: Where 1 item is more prominent than the other items (see appendix S).
- Option 2: Where there are no items more prominent than the other (see appendix T).

The list template has the following limitations:

- Must have at least 2 elements in a list.
- Can only have up to 4 elements in a list.
- The title has a character limit of 80 characters.
- The subtitle has a character limit of 80 characters.

The list template documentation also includes details about how to implement a list template, along with example code.

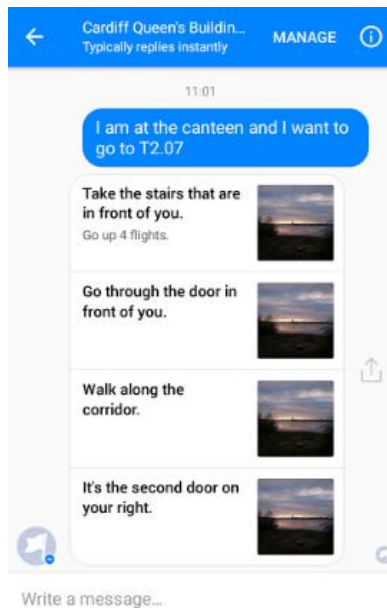
Below is the code that is used for creating each list within my Bot (This particular example is for the route from the canteen to T207. The image currently being used is a placeholder image):

```
function canteenT207(recipientId, text) {
  var imageUrl = https://scontent.cdninstagram.com/t51.2885-
15/s320x320/e35/c135.0.810/16465567_737180996447784_7352930134682238976.n.jpg;
  message = {
    "attachment": {
      "type": "template",
      "payload": {
        "template_type": "list",
        "top_element_style": "compact",
        "elements": [
          {
            "title": "Take the stairs that are in front of you.",
            "subtitle": "Go up 4 flights.",
            "image_url": imageUrl
          },
          {
            "title": "Go through the door in front of you.",
            "image_url": imageUrl
          },
          {
            "title": "Walk along the corridor.",
            "image_url": imageUrl
          },
          {
            "title": "It's the second door on your right.",
            "image_url": imageUrl
          }
        ]
      }
    }
  };
  sendMessage(recipientId, message);
};
```

The function is then called as follows:

```
app.post('/webhook', function (req, res) {
  var events = req.body.entry[0].messaging;
  for (i= 0; i < events.length; i++) {
    var event = events[i];
    if (event.message && event.message.text === "I am at the canteen and I want to go to T2.07") {
      canteenT207(event.sender.id, {text: "Here"});
    }
  }
  res.sendStatus(200);
});
```

When the user of the application types in "I am at the canteen and I want to go to T2.07, the function canteenT207 is called, and then the following is sent back to the user:



Users can easily scroll up and down the list of directions, and again the images will help the user to easily visualise where they are going and determine if they are heading in the correct direction.

However, some routes were larger than others and therefore required the use of more than 1 list in order to provide the user with all of the directions in a way that could be easily understood, due to the maximum number of elements in a list being set at 4. In order to achieve multiple lists, a separate function was created for each list part.

Below is the code for one such route from the canteen to C2.04:

```
function canteenC204A(recipientId, text) {
  var imageUrl = https://scontent.cdninstagram.com/t51.2885-
15/s320x320/e35/c135.0.810/16465567_737180996447784_7352930134682238976.n.jpg;
  message = {
    "attachment": {
      "type": "template",
      "payload": {
        "template_type": "list",
        "top_element_style": "compact",
        "elements": [
          {
            "title": "Take the stairs that are in front of you.",
            "subtitle": "Go up 3 flights.",
            "image_url": imageUrl
          },
          {
            "title": "Turn right and walk along the corridor.",
            "image_url": imageUrl
          },
          {
            "title": "At the end of the corridor take the stairs.",
            "subtitle": "Go up 2 flights.",
            "image_url": imageUrl
          },
          {

```

```

        "title": "Turn right and right again.",
        "image_url": imageUrl
    }
    ]
}
}
};
sendMessage(recipientId, message);
};

function canteenC204B(recipientId, text) {
    var imageUrl = https://scontent.cdninstagram.com/t51.2885-
15/s320x320/e35/c135.0.810/16465567_737180996447784_7352930134682238976.n.jpg;
    message = {
        "attachment": {
            "type": "template",
            "payload": {
                "template_type": "list",
                "top_element_style": "compact",
                "elements": [
                    {
                        "title": "Walk along this corridor.",
                        "image_url": imageUrl
                    },
                    {
                        "title": "It's the fourth door on your left.",
                        "image_url": imageUrl
                    }
                ]
            }
        }
    };
    sendMessage(recipientId, message);
};

```

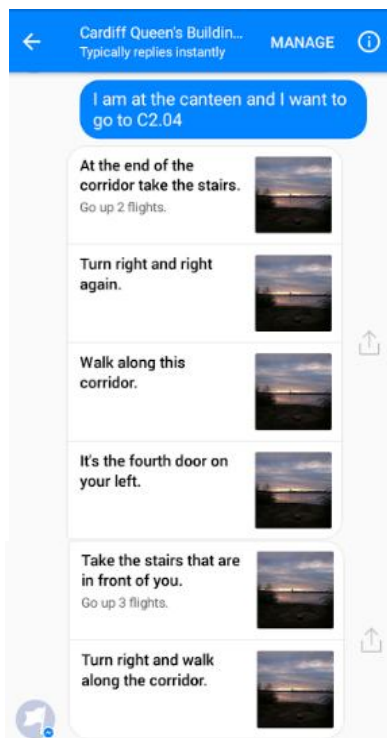
These two functions would've been called in the following way:

```

app.post('/webhook', function (req, res) {
    var events = req.body.entry[0].messaging;
    for (i= 0; i < events.length; i++) {
        var event = events[i];
        if (event.message && event.message.text === "I am at the canteen and I want to go to C2.04") {
            canteenC204A(event.sender.id, {text: "Here"});
            canteenC204B(event.sender.id, {text: "Here"});
        }
    }
    res.sendStatus(200);
});

```

When the user types in 'I am at the canteen and I want to go to C2.04, the functions canteenC204A and canteenC204B are called, and the following is sent back to the user:



As you can see, the directions have not come through to the user in the correct order, due to more than 1 list being used. This is obviously an issue as it can result in the user of the application getting lost and confused about where to go next.

After much investigation into why multiple lists come through in the wrong order, it was found that there is currently a bug with Facebook that has not yet been fixed, where multiple messages on a Bot are highly likely to come through in a random order due to it being asynchronous (Facebook for developers h 2016). Facebook suggested that developers try queuing their messages. However, many developers have responded to the post stating that queuing still does not solve the problem. We have also tried using wait functions to delay the code being executed before another piece of code has finished, but again this did not solve the problem. There is no news on when this bug will be fixed by Facebook.

As there is currently no way to guarantee that the Bot messages will be delivered in the correct order for the routes that require the use of more than 1 list, it was decided that until Facebook release a fix for this bug that the only alternative would be to give the directions for such routes as plain text. The plain text option will include no images, but the directions will at least be in the correct order.

Some routes, such as the above route from the canteen to C2.04, were then condensed so that they only needed to utilise one list. Below is the code for the condensed version of the route between the canteen and C2.04:

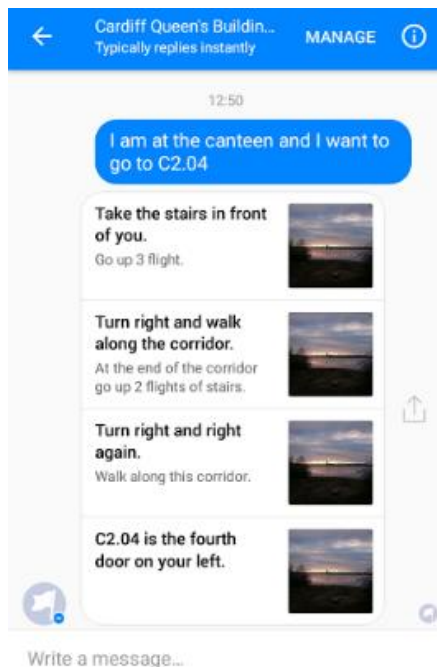
```
function canteenC204(recipientId, text) {
  var imageUrl = https://scontent.cdninstagram.com/t51.2885-
15/s320x320/e35/c135.0.810.810/16465567_737180996447784_7352930134682238976_n.jpg;
  message = {
    "attachment": {
      "type": "template",
      "payload": {
        "template_type": "list",
```

```

        "top_element_style": "compact",
        "elements": [
            {
                "title": "Take the stairs in front of you.",
                "subtitle": "Go up 3 flights.",
                "image_url": imageUrl
            },
            {
                "title": "Turn right and walk along the corridor.",
                "subtitle": "At the end of the corridor go up 2 flights of stairs.",
                "image_url": imageUrl
            },
            {
                "title": "Turn right and right again.",
                "subtitle": "Walk along this corridor.",
                "image_url": imageUrl
            },
            {
                "title": "C2.04 is the fourth door on your left.",
                "image_url": imageUrl
            }
        ]
    }
};
sendMessage(recipientId, message);
};

```

When the user now types “I am at the canteen and I want to go to C2.04”, this is the output that they will receive:



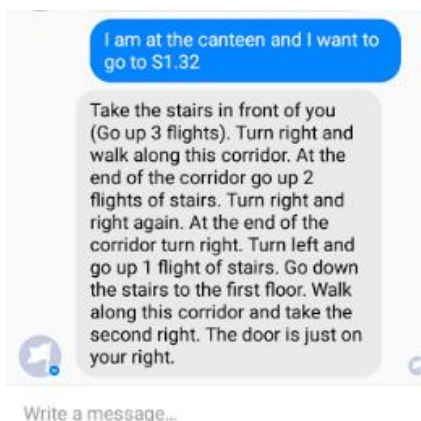
Other routes that could not be condensed were made completely plain text. Below is the code for the plain text version of the route from the canteen to S1.32:

```

app.post('/webhook', function(req, res) {
  var events = req.body.entry[0].messaging;
  for (i = 0; i < events.length; i++) {
    var event = events[i]
    if (event.message && event.message.text === "I am at the canteen and I want to go to S1.32") {
      sendMessage(event.sender.id, {text: "Take the stairs in front of you (Go up 3 flights). Turn
      right and walk along this corridor. At the end of the corridor go up 2 flights of stairs. Turn right
      and right again. At the end of the corridor turn right. Turn left and go up 1 flight of stairs. Go
      down the stairs to the first floor. Walk along this corridor and take the second right. The door
      is just on your right."});
    }
  }
  res.sendStatus(200);
});

```

When the user types "I am at the canteen and I want to go to S1.32", this is the output that the user will receive:



The rest of the routes were then implemented in the same way, using both lists and plain text where appropriate.

The Bot also accounts for any other message that a user may send which it cannot understand. The below code was used:

```

Else if (event.message) {

  sendMessage(event.sender.id, {text: "I'm sorry, I don't understand! Type 'help' to learn how to interact with me."

}

```

#### 4.2.2.5. Uploading and Using My Own Images

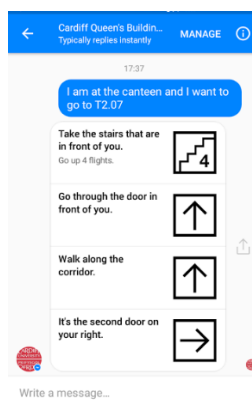
It was originally planned that the images used within the list of directions would be images of the corresponding corridor etc, as it would easily help users of the Bot to determine if they are going in the right direction. However, after taking these images it was found that the majority of them looked too similar to actually be helpful, especially with the merging of some directions (for example, in the title and subtitle section of each list element). As a result of this it was decided that the images should be plain vector symbols. These symbols are universally recognised and understood, so users would be able to determine what they mean with ease.

Only images that are public and accessible via a URL can be added to the Bot. In order to upload appropriate images to assist with effective navigation an Instagram account was created. Instagram allows its users to upload their own content which can then be reached via a URL. Each of the images that need to be utilised were uploaded to the Instagram account. (See appendix U to see how the Instagram account looks).

The code was then edited so that each route now contained the appropriate image URL's as shown below:

```
function canteenT207(recipientId, text) {
  message = {
    "attachment": {
      "type": "template",
      "payload": {
        "template_type": "list",
        "top_element_style": "compact",
        "elements": [{
          "title": "Take the stairs that are in front of you.",
          "subtitle": "Go up 4 flights.",
          "image_url": "https://scontent-fra3-1.cdninstagram.com/t51.2885-15/e35/17439000_1653569318272959_6790133369859473408_n.jpg",
          ...
        }
      ]
    }
  };
  sendMessage(recipientId, message);
};
```

This is how the completed routes that utilise lists now look:





## 5. Results and Evaluation

This section of the report will include details of the various tests that were carried out on the Bot and discuss how effective it is at solving the problem of navigation within the Queen's building.

The following tests will be carried out:

- A comparison between results from navigating the building with the app versus without the app in order to determine if the Bot does help reduce the time taken.
- A usability test to determine how easy it is for users to interact with the Bot. This test will help determine if the Bot allows users to find their desired destination in the desired timeframe.
- A follow-up questionnaire to gain insight into what the test participants thought of the Bot and if they would actually use it themselves.
- Reliability testing based on the original requirements for the Bot to determine if the Bot accomplishes everything it set out to achieve.

### 5.1. App vs No App

The results from the initial testing where an individual who had never been to the Queen's building was asked to find a number of locations using only the signs placed around the university building can also be found in the table below. Another individual who had never been to the Queen's building was also asked to find the same locations using the Bot. The table below summarises the findings from this test:

Start Location	End Location	Time taken without Bot	Time taken with Bot	List or plain text	Average Experienced Time
Canteen	S/2.21	19 minutes 21 seconds	4 minutes 40 seconds	Plain text	3 minutes 8 seconds
Canteen	N/4.07	2 minutes 46 seconds	2 minutes 25 seconds	List	1 minute 58 seconds
Canteen	C/2.07	6 minutes 17 seconds	3 minutes 2 seconds	List	1 minute 24 seconds

As you can see, the individual using the Bot found each location in a shorter amount of time than the individual who was just using the buildings signs. This suggests that the Bot does help with navigation in the Queen's building. The individual using the Bot found each location in under 10 minutes (the amount of time that a student has to move rooms between lectures), meaning the Bot is successful at ensuring that students do attend lectures on time. The time taken with the Bot was much closer to the time that an average student who already knows the building would take in order to find their desired locations.

## 5.2. Usability Testing

Usability testing was carried out in order to determine how easy users found interacting with the given solution. Each user was given a sheet of paper (see appendix V) which contained the following instructions:

- How to install the messenger application (if not already installed on the device).
- How to find the Bot on the messenger application.
- A list of rooms that they were asked to find.
- Instructions on what to do if the welcome message is not displayed

As the users interacted with the Bot, and tried to find each location, they were observed. The time taken for each user to find each location was recorded, along with any questions or comments they made. Below is a summary of the results:

	Canteen – S1.32 (Plain text)	S1.32 – N4.07 (Plain text)	N4.07 – T2.07 (List)	T2.07 – C2.04 (List)	Comments
User A	2 minutes 51 seconds	2 minutes 30 seconds	2 minutes 41 seconds	1 minute 27 seconds	Confusion over corridor between central and Trevithick building. Confusion over floor 2.5, should make it more consistent with the rest of the directions. Easy to use.
User B	3 minutes 51 seconds	3 minutes 13 seconds	2 minutes 5 seconds	1 minute 50 seconds	Confusion over floor 2.5, should change to be more consistent with the rest of the directions. Easy to use.
User C	3 minutes 27 seconds	2 minutes 48 seconds	2 minutes 54 seconds	2 minutes 29 seconds	Easy to use. Confusion over corridor between central and Trevithick building Would be better if all directions used list format.

As the results show, each user was able to find each location in under the 10 minutes that students have in order to change rooms between lectures. This suggests that the Bot is a success as each user was easily able to find each location in time, meaning that it would be helpful in ensuring that students attend all lectures and on time. This will help to ensure that students do not miss valuable contact time with lecturers.

Whilst carrying out this testing, all users commented on how easy the Bot was to use, as it is part of the Facebook messenger application. All users already had the messenger

application installed on their device and were comfortable with using it. They also thought that the Bot was a good idea and that it would be something that they would use if they were in the same position as many first year students often find themselves in. The testers also made some constructive comments throughout the testing, which will be acted upon for the final version of the prototype of the Bot. Some users found that it was confusing when some stairs were referred to as floor 2.5, rather than sticking to the number of flights of stairs to go up or down. Whereas some users also commented on confusion on which corridor to take in order to travel between the central building and the Trevithick building.

However, finding participants was difficult due to needing individuals who do not know their way around the Queen's building. This was even more of an issue due to non-university students not being allowed into the school during the evenings or weekends. Ideally, further testing would be carried out on a much larger number of people in order to establish how reliable and useful it is.

### **5.3. Questionnaire**

After carrying out the testing, each user was given a short questionnaire (see appendix W). The questionnaire contained the following questions:

- Type of operating system the mobile device uses.
- If Facebook messenger was already installed on the device being used.
- If the Bot was easy to find.
- If the welcome message was displayed correctly.
- If the navigational directions were received within seconds.
- If the navigational directions were easy to follow.
- If the user found the desired location in a suitable amount of time.
- If the Bot was easy to use and interact with.
- If the user would consider using this type of product in the future.

All individuals who took part in the user testing above already had the Messenger application installed on their devices. Each individual stated that the Bot was easy to find in Messenger as they just had to type its name into the search bar, the same as if they were starting a conversation with any friend. As a result of the Bot being incorporated into the Messenger application, each individual thought that the Bot was easy to use and interact with. The welcome message was displayed correctly for each individual. This was helpful as the welcome message told them what the Bot was capable of and how to interact with it. Each individual stated that the directions were received within a few seconds after sending their request and that said directions were easy to understand and follow in order to reach their required destination. As shown above, each individual was able to reach their desired destinations within the required amount of time. All individuals said that they thought the Bot was a good idea and that it would be something that they would consider making use of.

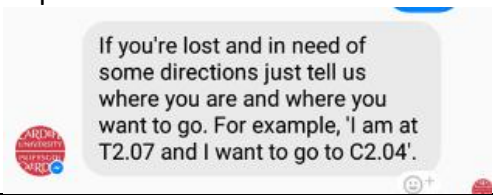
#### 5.4. Changes Made as a Result of User Testing

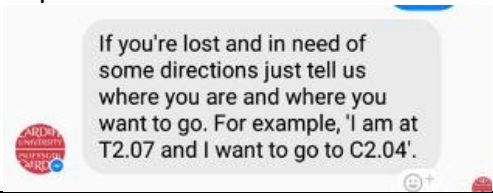
As a result of the user testing, the following changes were made to the Bot:

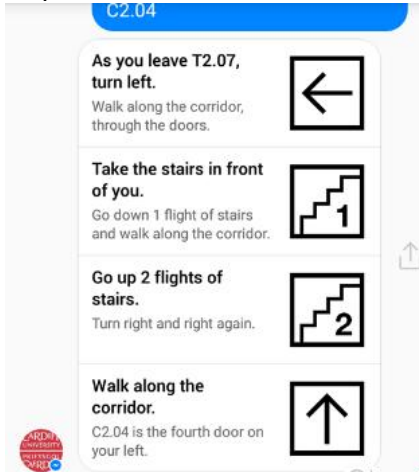
- Where the Bot originally says 'go up to floor 2.5' etc within the south building, the Bot now says things like 'go up 4 flights of stairs', making it more consistent with the rest of the Bot, and easier for the user to understand.
- The corridor connecting central and Trevithick building has been clarified, so that users of the Bot know which corridor to take in order to travel between them.

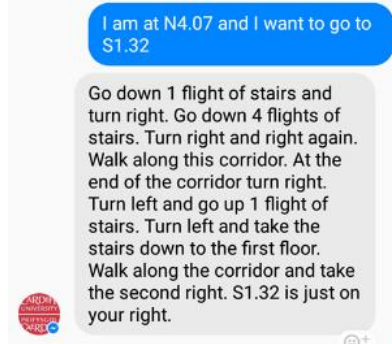
#### 5.5. Reliability Testing

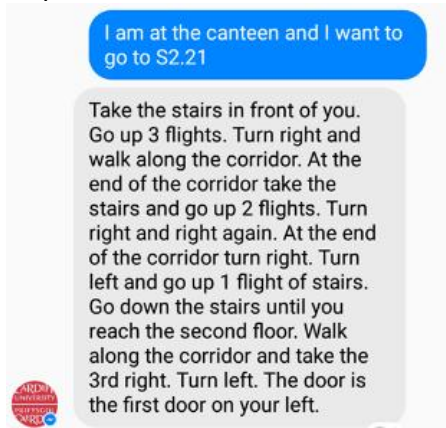
Below are some of the tests that have been carried out in order to ensure that the Bot works reliably. These tests have been designed with the requirements for the Bot in mind.

Test Case ID: 1		Test Purpose: Check that the Bot responds within 2 seconds.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'Help' and press the send button.	<div>The Bot should send the following response within 2 seconds:</div> <div></div>	PASS 10/10
Comments: N/A			

Test Case ID: 2		Test Purpose: Check that the Bot responds at all.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'Help' and press the send button.	<div>The Bot should send the following response:</div> <div></div>	PASS 10/10
Comments: N/A			

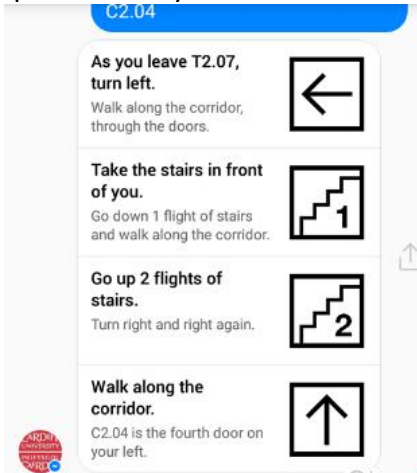
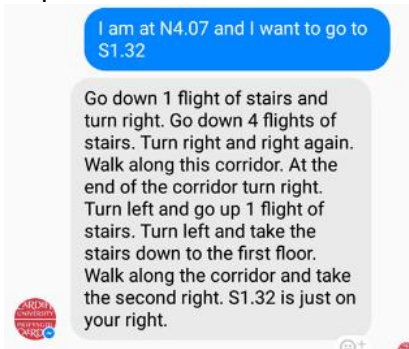
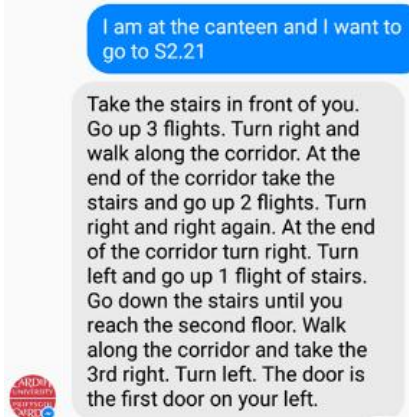
Test Case ID: 3	Test Purpose: Check that the Bot responds with accurate directions for the following request: 'I am at T2.07 and I want to go to C2.04'.		
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at T2.07 and I want to go to C2.04' and press the send button.	<p>The Bot should send the below response:</p> 	PASS 10/10
Comments: N/A			

Test Case ID: 4		Test Purpose: Check that the Bot responds with accurate directions for the following request: 'I am at N4.07 and I want to go to S1.32'.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at N4.07 and I want to go to S1.32' and press the send button.	<p>The Bot should send the below response:</p> 	PASS 10/10
Comments: N/A			

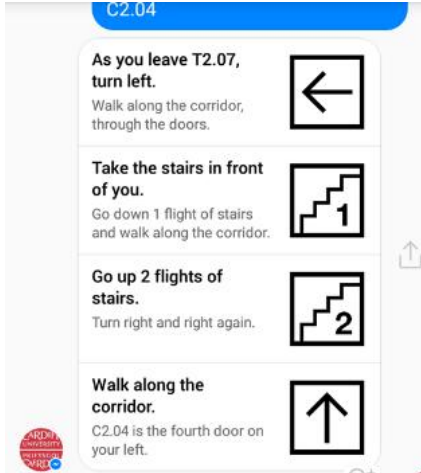
Test Case ID: 5		Test Purpose: Check that the Bot responds with accurate directions for the following request: 'I am at the canteen and I want to go to S2.21'.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at the canteen and I want to go to S2.21' and press the send button.	<p>The Bot should send the below response:</p> 	PASS 10/10
Comments: N/A			

Test Case ID: 6		Test Purpose: Check that the Bot responds with accurate directions for the following request: 'I am at C2.04 and I want to go to the toilet'.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at C2.04 and I want to go to the toilet' and press the send button.	<div>The Bot should send the below response:</div> <div><div>I am at C2.04 and I want to go to the toilet</div><div><div>Turn right as you leave the room. Walk to the end of the corridor.</div><div>The toilets are right in front of you.</div></div></div>	PASS 10/10
Comments: N/A			

Test Case ID: 7	Test Purpose: Check that the Bot responds with directions that are spelt correctly for the four routes as tested above.
Environment: Mobile device with Messenger application.	
Preconditions: The Bot has been installed and the conversation is open.	
Test Case Steps:	

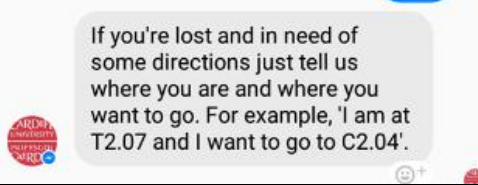
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at T2.07 and I want to go to C2.04' and press the send button.	<p>The Bot should send the below response and all words should be spelt correctly :</p> 	PASS 10/10
2	Type 'I am at N4.07 and I want to go to S1.32' and press the send button.	<p>The Bot should send the below response:</p> 	PASS 10/10
3	Type 'I am at the canteen and I want to go to S2.21' and press the send button.	<p>The Bot should send the below response:</p> 	PASS 10/10
4	Type 'I am at C2.04 and I want to go to the toilet' and press the send button.	<p>The Bot should send the below response:</p>	PASS 10/10

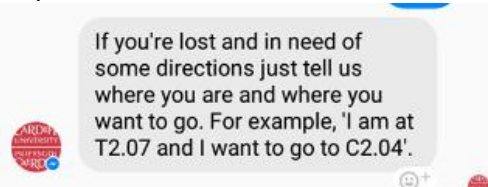
			
Comments: N/A			

Test Case ID: 8		Test Purpose: Check that the Bot list directions are displayed correctly.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at T2.07 and I want to go to C2.04' and press the send button.	<p>The Bot should send the below response (all text and pictures should be displayed correctly):</p> 	PASS 7/10 FAIL 3/10
Comments: The ones that failed often had the last word cut in half. Each one that has failed has been an iOS device. Although, it does not happen to all iOS devices.			

Test Case ID: 9		Test Purpose: Check that the Bot allows the user to initiate a conversation.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Open the conversation.	The Bot should display the following welcome message: 'Hi *NAME*! We're here to ensure that you never get lost within our building again. Let us know if you're in need of some	PASS 10/10



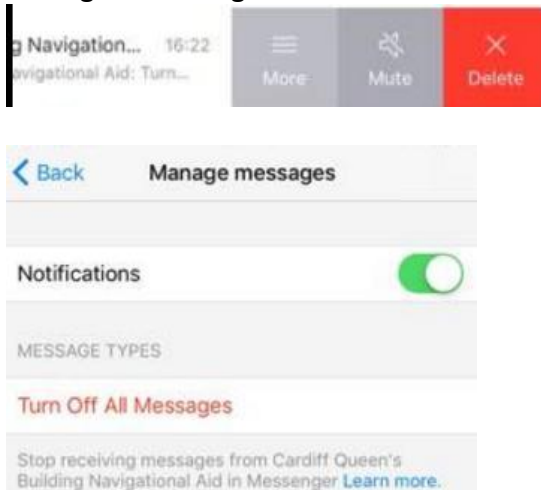
		directions! Type 'help' 😊'.	
2	Type 'help' and press the send button.	<p>The Bot should send the following response:</p> 	PASS 10/10
Comments: N/A			

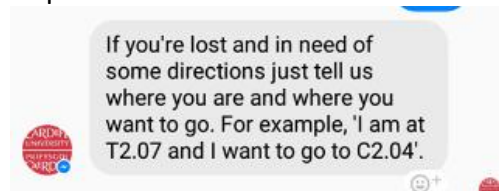
Test Case ID: 10		Test Purpose: Check that the Bot gives the user clear instructions on how to use the Bot.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'help' and press the send button.	<p>The Bot should send the following response:</p> <div></div> <p>The user should know how to correctly send a request to the Bot.</p>	PASS 10/10
Comments: N/A			

Test Case ID: 11		Test Purpose: Check that the Bot allows the user to input a start and end location in order to ask for directions.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at T2.07 and I want to go to C2.04' and press the send button.	The Bot should send the below response:	PASS 10/10

Comments: N/A			

Test Case ID: 12		Test Purpose: Check that the Bot allows the user to enter a new request when they are lost, and then the Bot responds with a new set of directions.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'I am at the canteen and I want to go to S1.32' and press the send button.	<div>The Bot should send the below response:</div> <div></div>	PASS 10/10
2	Type 'I am at N4.07 and I want to go to S1.32'.	<div>The Bot should send the below response:</div> <div></div>	PASS 10/10
Comments: N/A			

Test Case ID: 13		Test Purpose: Check that the Bot allows the user to block or mute the conversation.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Attempt to mute or block the conversation in Messenger.	<p>The following should be displayed if muting or blocking is enabled:</p>  <p>(These will vary slightly depending on the operating system being used. The above is an example of the iOS operating system.)</p>	PASS 10/10
Comments: N/A			

Test Case ID: 14		Test Purpose: Check that the Bot works on all operating systems.	
Environment: Mobile device with Messenger application.			
Preconditions: The Bot has been installed and the conversation is open.			
Test Case Steps:			
Step No.	Procedure	Response	Pass/Fail
1	Type 'help' and press the send button.	<div>The Bot should send the below response:</div> <div></div>	<div>PASS 4/4 Android</div> <div>PASS 4/4 iOS</div> <div>PASS 2/2 Windows</div> <div>Total: PASS 10/10</div>
Comments: N/A			

For 7 out of 10 people the Bot successfully worked as intended. However, for 3 out of 10 people the Bot did not work as intended. All of these users were iOS users. However, it did not happen for all iOS users. The only problem which meant that the Bot was not working as intended was that the last word or two were cropped from directions that utilised the list format. This could be down to the different screen sizing and resolutions associated with Apple devices. This is a relatively minor issue, as often users can work out what the missing word was supposed to read. The iOS users who carried out user testing at the university still managed to find their desired locations within the 10 minute time frame. However, this is something that will be changed in the future. The wording of the list directions can be changed so that there are fewer words, which will result in no words being cropped off. Bar this minor issue, the testing successfully shows that the Bot works across a number of different OS correctly.

There was one user for which the Bot would not work for initially. This user was the project supervisor. This was a strange anomaly, and the only plausible explanation at the time was that it may have something to do with their privacy settings on Facebook. However, it did start to work for my supervisor towards the end of the project, so the likely cause of this issue was a bad internet connection. Further testing, using individuals with varying levels of account privacy, is needed in this area to determine the full cause.

Again, more testing is required on a larger number of people to ensure that the Bot is reliable.

One of the biggest strengths of the Bot is that it can be easily expandable. Extra routes and locations can be added quickly with ease. Minimal skills would be required (basic knowledge of javascript and working with the Facebook API) in order to add these extra locations to the Bot, as there is already a defined structure to the code, so this format can be followed. Even when the School of Computer Science and Informatics moves to a new location within Cardiff, the Bot can be updated to accommodate these changes. Surveying the building would take some time, but again, adding these changes to the code should not be too time consuming or difficult.

The Bot itself should not be affected by the changes that Facebook makes to its services. However, if the Bot is affected, then all the changes that Facebook make will be fully documented on the Facebook for Developers website. This documentation will mean that changing the Bot to work in accordance to these changes made by Facebook will be easy to do. One weakness with the Bot is that there are some bugs with the Facebook API which are beyond our control to fix. Bugs like these mean that the Bot does not fully work how it was intended. Facebook are constantly investigating and fixing bugs that individuals find with their API. Once the bug affecting the order of multiple lists being sent is fixed, then the Bot can be changed so that all routes use this format. This will make the Bot more consistent, and will make the instructions easier for individuals to understand and follow.

Another strength of this solution is that many individuals already have the Messenger application installed on their mobile devices. As a result of this, it means that individuals are more likely to understand and feel comfortable when interacting with a Bot. This also means that individuals do not need to install a separate application on their device in order to access the Bot. This makes it more likely that they would use such a Bot. Whilst carrying out user testing, all individuals said that they thought the Bot was a good idea and that it would

be something that they would like to use in order to help navigate around unfamiliar buildings.

The Bot dramatically cuts down the time it takes individuals to find locations within an unknown building. As a result of this, the Bot can be deemed as successful as it successfully achieves the main goal of the project. It also means that students are more likely to find their desired location on time, so they are less likely to skip, or arrive late to, a lecture.

One of the biggest weaknesses of the Bot in its current state is that the Heroku server shuts down after 30 minutes of inactivity. As a result of using Heroku, the Bot can only handle a limited number of requests. This means that the Bot could be potentially unavailable at times, which could affect students who make use of the Bot. However, the plan is to move the Bot to the school's own server, which will ensure that the Bot is available 24/7 to all students who wish to use it.

Another strength of the Bot is that it can also be used by visually impaired and blind people. Visually impaired people have the option to either increase the font size within the messenger application or increase the font size of their whole mobile device to make reading the text easier. Whereas blind people, as well as visually impaired people, can make use of screen readers. Screen readers are programs that can enable such users to read text that is displayed in a digital format using either a speech synthesizer or a Braille display (American Foundation for the Blind 2016.). Facebook's help centre contains lots of information on how screen readers can be used with the messenger application in order to read and respond to messages (Facebook Help Centre 2017).

The Bot was relatively quick to create, taking only a number of weeks to implement the prototype. A full working version of this Bot could be quickly created by another individual with the relevant skills within a month. The most time consuming part of creating the full version of the Bot would be surveying the university building in order to determine suitable routes to each desired location. The code itself would be fairly easy to extend as the format is already created.

Making use of Javascript in order to implement the Bot was the correct choice as Javascript is heavily documented online, and there are lots of useful web forums to make use of if there are any problems. It is also very easy to make use of the Facebook API using Javascript. The Facebook for Developers website also gives lots of code examples using Javascript.

It was originally decided that the waterfall model of software development would be utilised. However, as the project moved forward it was quickly realised that the iterative model was probably more realistic an approach. Throughout the project it was decided that when finding problems with the code that more research should be carried out in order to overcome them. We also revisited the implementation section again after carrying out user testing in order to make changes to the code that individuals suggested. As a result of this, we entered the research, development and testing stage a number of times throughout the project before we were satisfied with the prototype. Using the iterative model was a strength as it meant that continuous improvement of the prototype could take place and ensure that problems with the code could be solved.

## **6. Future Work**

This section of the report suggests some ways in which the navigational Bot can be improved upon and expanded in the near future.

The most obvious improvement to be made to the prototype is to make it inclusive of all years and degrees that make use of the Queen's building. This would involve surveying the whole of the building and making routes between each room within it. This would mean that the application can be used by all students who have lectures within the building. The application could also be expanded to include every building at Cardiff University, making it an application that could be used by every student during their time at Cardiff University.

Another change that could be made is moving the application to the schools' server. This would ensure that the application is always up and running. Due to the Bot currently only being a prototype, the Heroku server space that is currently being used shuts off after 30 minutes of inactivity, whereas to be put to use it would need to be online 24/7.

The application could be expanded to be inclusive of people who have difficulties navigating stairs (those with mobility disabilities) etc, by having an option that allows users to select routes that include only elevators, or routes that have the least number of stairs. This could be a preference that is saved during users initial interactions with the Bot.

The application could be further expanded to allow a user to input their timetable. This would allow the Bot to pre-empt when a user will be in need of navigational support and ask the user if they require assistance. This would help to make the application more efficient and less time consuming for a user to interact with.

Once a fix for the bug with the Facebook API becomes available, then all plain text directions will also need to be changed to become lists. This will make the application more consistent and it will make all of the directions easier for the user to understand.

After carrying out reliability testing, it was found that for some iOS devices, the last word or two of directions that were presented in the list format were cut off. In the future, the wording could be slightly changed so that there are fewer words, which would result in no words being cut off. This would ensure that all users receive the information that they require fully.

## 7. Conclusions

The aim of this project was to carry out a feasibility study in order to determine if it would be possible to create an application that could help both students and staff navigate around the Queen's building at Cardiff University more effectively.

The main goals of this project were to:

- Understand the problem by carrying out research into existing applications that solve navigational problems in similar scenarios, and by carrying out an analysis of the current solution to the navigational problem at university.
- Identify and carry out a feasibility study on two separate potential solutions in order to determine the most feasible solution to the problem.
- Carry out a survey to determine how severe the navigational problem is and identify the most common problems individuals face when visiting the building for the first time. The survey should also identify how their learning/ teaching is affected by poor navigational aids, give insight into how others see the problem being resolved, and highlight areas within the building where the current navigation system is confusing.
- Create a prototype of the most feasible solution in order to solve the problem of navigation within the Queen's building.
- Create and carry out user testing to determine how well the solution solves the problem.

All of the main goals were achieved throughout this project. Research into existing solutions showed that the problem of navigation is not a new one, and indoor navigation is still very much a growing area of interest as it has not yet been perfected. An analysis of the current solution to the navigational problem at Cardiff university highlighted how difficult navigation within the Queen's building can be. It helped to shed light on the severe lack of navigational signs within the building, as well as fire escape route signs, which are essential for the safety of all staff and students.

The survey showed that 85% of students who attend the School of Computer Science and Informatics have found themselves lost, which highlights what a big problem navigation is within this building at Cardiff University. Many said that they find navigating between the different buildings difficult as they connect on different floor numbers. As well as this, many also commented on the severe lack of useful navigational signs within the building. Many students also find that the room numbers are not laid out in an intuitive way and that they seem inconsistent throughout the building. This is a big problem as failure to attend lectures or missing parts of lectures can really affect a student's ability to learn and keep up with all course content. From a lecturers perspective, this is a problem as turning up late to lectures can result in not having enough time to fully deliver all course content required within the number of teaching slots allocated. Many students said that they could see the problem being solved using a mobile solution, as all students surveyed had a mobile device.

A feasibility study was carried out on two potential mobile solutions (one, a live map application using iBeacon technology, the other, a social media 'chat-bot' navigational planner). Both potential solutions had their own respective strengths and weaknesses. However, it was determined that the most feasible solution would be the Bot. The Bot

passed the feasibility study as the Bot could be implemented making use of Javascript and online documentation relating to the Facebook API. Users also have to initiate a conversation with the Bot, meaning that they need to consent to Facebook's privacy terms and conditions. The Bot also passed the feasibility study as it can be up and running in a relatively short amount of time and within the timeline created for the project. The Bot is also free to implement with the Facebook API and free to host on the Heroku server.

A prototype of the messenger Bot was created using Javascript and the Facebook API. This prototype included routes for all lecture rooms that the current first year students utilise, ensuring that the prototype could be of use to first year students, as they are the students who are most likely to benefit from having the application available to them. The prototype was fully created within the time allocated to its creation. A number of problems were encountered throughout implementation. However, the problem of lists being sent in a random order was down to a Facebook bug. As a result of this, some directions were in a list format, whereas others were in a plain text format. This is something that can be easily changed to make the Bot more consistent and easy for users to understand when the bug is fixed by Facebook.

The testing proved that the Bot was successful as it dramatically improved the timing in which individuals who were new to the building found their desired locations. The testing also helped to show that the Bot worked on a number of different OS's, meaning that it can be used by all students at Cardiff University, regardless of their device. However, the testing did help to highlight that some of the directions were cropped on some iOS devices due to the different screen sizes of such devices. This is a relatively minor issue, and it did not affect individual's abilities to find the desired location. This is a relatively simple problem that can easily be fixed in the future. The comments made by individuals during the testing also helped to reiterate the fact that a Bot is something that they would like to use in order to help them navigate around unknown places.

In conclusion, it is possible to create a mobile solution that could help with navigation within the Queen's building at Cardiff University. The most feasible mobile solution is the use of a Bot. Creating a prototype of the Bot was possible, and it helped to show that creating a full version of the Bot is feasible and relatively easy to implement. The Bot was proven to be extremely helpful for navigating around the building and solving the problem of navigation, as all individuals who used the Bot were able to find their desired location in very little time. Creating a full version of the Bot is definitely something that Cardiff University should consider in order to help new students feel more comfortable and settle in during their first weeks at university.



## 8. Reflection on Learning

This section of the report will contain an evaluation of my approach to the project as a whole.

Undertaking this project has helped me to develop my time management skills. Creating the initial plan was useful as it challenged me to think about what tasks needed to be completed and how long these tasks may take. I successfully used some of my existing knowledge of previous software development tasks in order to determine how much time would be needed for implementation for this project. Using this knowledge was efficient as it meant that I was able to allocate the correct amount of time to carry out each task. Creating an initial plan is something that I will always do when undertaking a software development project in the future as it helps to set realistic expectations about what can be done in time, whilst also helping to identify all tasks that need to be completed. I referred back to my initial plan throughout the project to determine if I was on track and to see what tasks still needed to be completed. Referring back to my initial plan helped me to focus and not waste time on unnecessary tasks.

As well as this, I also had to balance my personal and work time. In order to do this, I decided that I would treat each week day as a university day and work on my dissertation throughout this time. I then used my weekends for personal time. This was effective as it meant that I had a defined structure, but also time set aside where I could take a break from work. Having a break from the same piece of work is important, so it will be something I continue to do in the future. It helps me to come back to problems with fresh eyes and potentially new ideas on how to solve them. I also had to manage my time around other university modules and coursework. In order to do this, I prioritised the coursework for the other module during the week that it was due in to ensure that it was complete to a high standard. This was effective as it meant that I was able to maximise my mark in the other module, while still continuing to work on my dissertation. Prioritising more urgent pieces of work is effective as it means that all pieces of work get done on time. As a result of this, it is something that I will continue to do in the future.

Although the work/ personal time balance is important, it was often important to also be flexible. When it came to the testing of the application, I was willing to work around the schedule of each person who volunteered to help test it, which included working on weekends or evenings.

However, whilst creating my initial plan I had decided to use the waterfall model of software development as I had thought that each task would be carried out and completed before the next. In reality, I feel that I have utilised the iterative model, as I created the prototype, tested it and then went back to make changes to the implementation. When planning a software project in the future, I will definitely be using the iterative model from the beginning, as it is a more realistic approach when building a piece of software, as changes often need to be made.

Throughout the project I have also had an opportunity to develop my programming skills. Before the start of this project I had little prior experience with using Javascript. However, I feel that I have now become more adept and confident with using this language, and it will be a skill that I continue to use and develop in the future, especially for web based programming tasks. I supported my learning of Javascript in a variety of ways throughout

this project. One such way was through the use of the tutorials, which offered explanations to some of the more complex sections of code in order to initially set up the Bot. These explanations helped clarify what each section of the code was responsible for. I also made use of online forums such as stack overflow when I encountered programming problems that proved difficult to solve. Using these forums was helpful and effective as it allowed me to utilise the experience and expertise of many Javascript programmers worldwide and learn more about the language. As a result of using these forums I was able to recognise and find a way to overcome the problem of the Bot messages being delivered in a random order. I found out that Javascript is an asynchronous language that runs functions as soon as possible. This was something that I did not know before, but it will be something I keep in mind whilst programming in this language in the future.

I also had the opportunity to learn how to set up a server. Prior to undertaking this project I had never set up server space before. Following a tutorial made the process fairly easy. Setting up server space with Heroku was efficient as it was free to use and it easily allowed me to link my Github account so that the code for my Bot could automatically be deployed. This was a valuable opportunity as I will definitely be reusing this knowledge in the future. However, in the future I may consider paying for a full subscription, as the free version of Heroku only allows for a limited number of calls and automatically switches off after thirty minutes of inactivity. If I were to create and release a full version of this Bot I would consider moving the code to the school's server, which is on twenty-four seven. This would mean that students could utilise the Bot at any time.

Prior to undertaking this project I had limited experience of using Github. I used Github in order to keep my code all in one place and to deploy my code to the Heroku server. This was effective as it meant that my code could automatically be deployed by Heroku. It was also efficient as it meant that I could access my code from any location and allowed me to become more flexible. It also meant that my code was backed up so that even if I had computer problems my code was safe. Using Github also allows you to keep track of changes made to the code. Knowledge in Github is something that many employers express interest in, as many software development teams make use of it in order to collaborate on coding projects, so it will be something that will be of use to me in the future.

Throughout this project I have also learned how to work with the Facebook API. During a group project last year, I had limited exposure to the Facebook API, as I was only concerned with testing the application that we produced. In order to understand how to work with the Facebook API, I used the documentation provided by Facebook for developers that concerned the messenger application. This, along with various bits of information included within the tutorials, was extremely useful as they provided all information necessary in order to create applications with the Facebook API successfully. The documentation also included information about releasing the Bot for public use. Taking in this information was effective as it meant that I could create my Bot with these restrictions in mind, ensuring that my Bot could be released to the public. One downside about working with the Facebook API is that it is constantly evolving. These changes could potentially affect the workings of the Bot in the future. For example, there is a current bug with the Facebook API which means that messages that are in the form of lists are not received in the correct order that they are sent. This bug means that for navigational purposes lists cannot be utilised for every route that the Bot knows of. When this bug is fixed it will be possible for all routes to utilise the list

format, making the Bot more consistent and user friendly. I plan on using the knowledge I have gained from working with the Facebook API in order to create more Bots for other purposes in the future.

I have also utilised and developed my problem solving skills through completing this project. I came up with two separate ideas on how the problem of navigation within the Queen's building could be solved. In order to do this I had to consider many variables, such as how feasible a mobile solution could be, its cost, how long it would take and if I had the technical skills required in order to implement the solution. Considering each of these points in a feasibility study required me to carry out in depth research into my potential solutions in order to determine how feasible each would be. Considering these variables allowed me to be impartial and think logically about which solution would actually be the best solution to implement. I would definitely use a feasibility study in future software development briefs.

I also had to overcome many programming problems. As mentioned above, I had to overcome the problem of the Bot messages being sent in the wrong order. In order to overcome this I tried many different things, such as creating a wait function to stop the function from running. However, this was unsuccessful, and it was only after a lot of research that I found out that there was a Facebook bug. In order to work around this problem, I decided that it would be best to send some messages as plain text, rather than in the list format. Deciding to send these messages in plain text meant that the directions were not mixed up and that the user of the Bot would still be able to find the location that they were looking for.

As a result of carrying out this project I have also developed my research skills. I utilised many resources such as online journals and websites in order to carry out effective research. Reliable resources are important so that your work can be deemed credible. Knowing the difference between reliable resources and non-reliable resources is something that will be of use to me in the future. For example, trying to justify and carry out a feasibility study for potential software solutions to a potential client.

Throughout this project I have also had the opportunity to learn about how the blind and other visually impaired people navigate. As a result of this I also considered how technology could be used to help them with navigation. This was effective as it means that I can be more considerate of this in the future. For example, I can implement software which can also be utilised by the visually impaired by either having a larger text size or by allowing the software to be read by a screen reader. This knowledge has also given me more of an appreciation for the daily struggles that the blind and visually disabled face.

My communication skills have also developed as a result of carrying out this project. In order to successfully manage and complete this project I have had to communicate with a number of people. I have had to communicate with my supervisor to ensure that I am carrying out the project in a way that is satisfactory and that I am including all necessary elements within my project report. I have had to communicate in a number of different ways, such as formal meetings and less formal emails. Communicating in a formal way is a necessary skill when entering the world of work, as formal communication is a must in industry, be it with supervisors, managers or clients. Note keeping has also been a necessary part of this project for me. Keeping notes regarding tasks to complete, conversations with my supervisor or notes about my progress with implementation meant that I was far more

easily to communicate my experience throughout this project when it came to writing the final report. Note keeping was also effective as it meant that I knew exactly what I needed to achieve for the week, as I often made to-do lists. Note keeping is something that I have and will still aim to do throughout any software development project as it allows me to maintain a clear focus on the goals.

I communicated with various testers for my application. In order to successfully communicate with the testers and explain what it was that was required of them I decided that I would create an information sheet. This sheet provided the tester with all essential information that they required about the test, such as setting up their device to use the Bot etc. This sheet helped me to ensure that the tester was given all necessary information, which I may have forgotten to do if simply talking to each individual. This helped to lower the chance of human error interfering with the results of each test. Using the information sheet also allowed me to include screen shots. Using screen shots was effective as it meant that the information was more clear and easier for the testers to understand and follow. In order to gain useful feedback from the testers I decided to create a questionnaire. Whilst creating the questionnaire I thought about all necessary information that I wished to require as a result of testing and included questions in order to gain this. I made the questions predominately multiple choice so that it was quick and easy for the testers to fill out. However, I also provided comment sections so that the testers could provide a more in-depth answer where required. Using multiple choice questions meant that the data gained was easier to collate and analyse. My skills in changing my communication method when communicating with different people can easily be utilised in the future. My skills in creating questionnaires that collect relevant data is something that can be used when testing software in the future.

Whilst carrying out this project I have acted under the following assumptions:

- The university WiFi works in all areas of the building/ the mobile phone has access to mobile data.
- Everyone within the building has access to a mobile device.

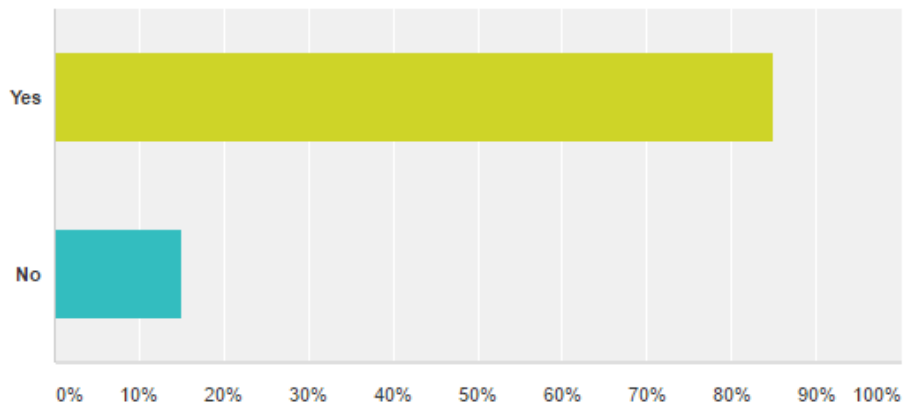
Acting under these assumptions means that a student would be able to utilise my Bot throughout the whole building. Working under these assumptions meant that creating a mobile application in order to solve the problem of navigation within the Queen's building was possible.

## 9. Appendices

A)

**Have you ever found yourself lost or unsure of where to go within the building?**

Answered: 40 Skipped: 0

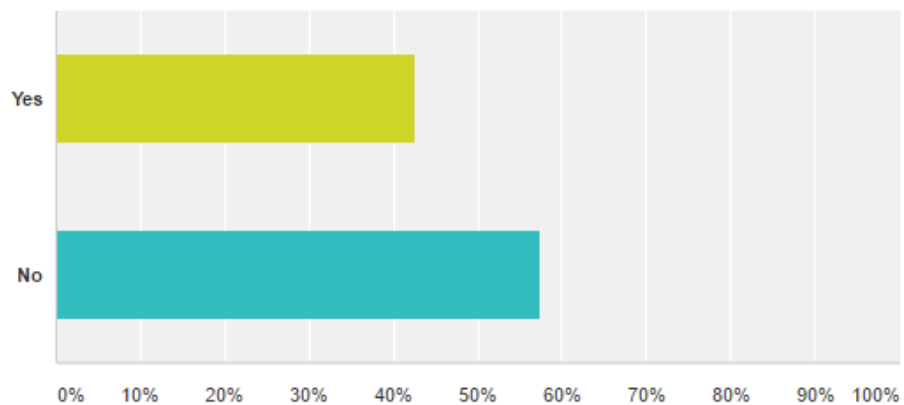


*SurveyMonkey – Correct as of 10/02/2017*

B)

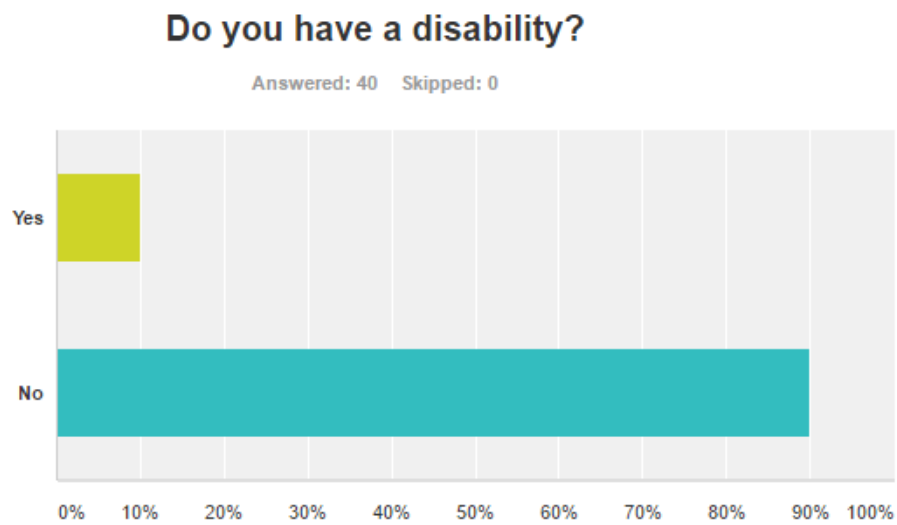
**Do you think the existing signs for navigation within the building are useful?**

Answered: 40 Skipped: 0



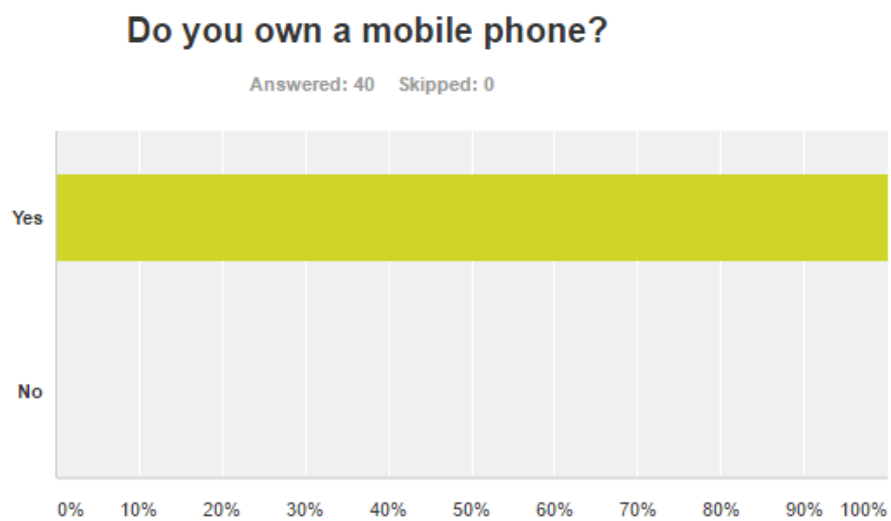
*SurveyMonkey – Correct as of 10/02/2017*

c)



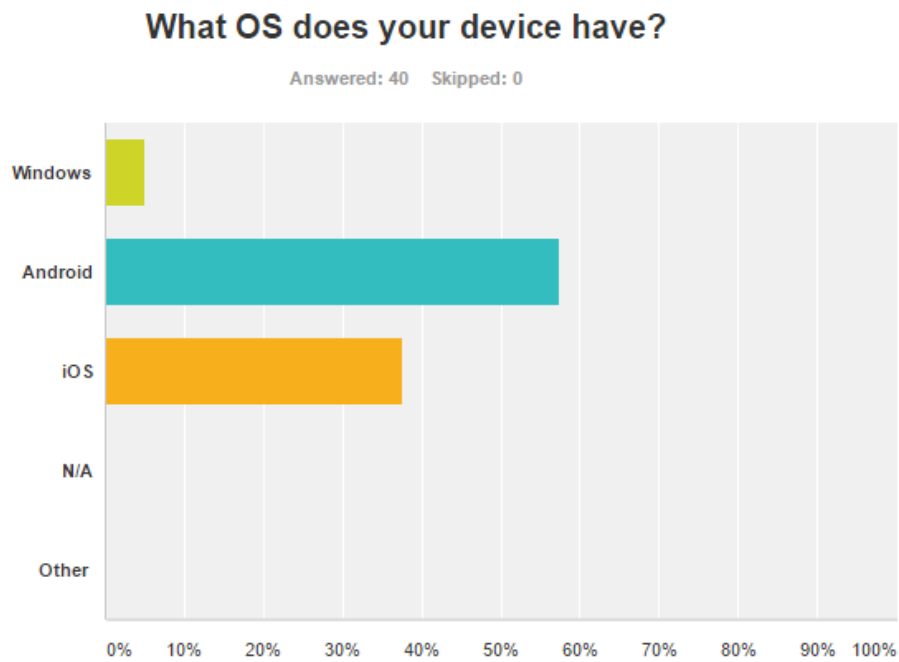
SurveyMonkey – Correct as of 10/02/2017

D)



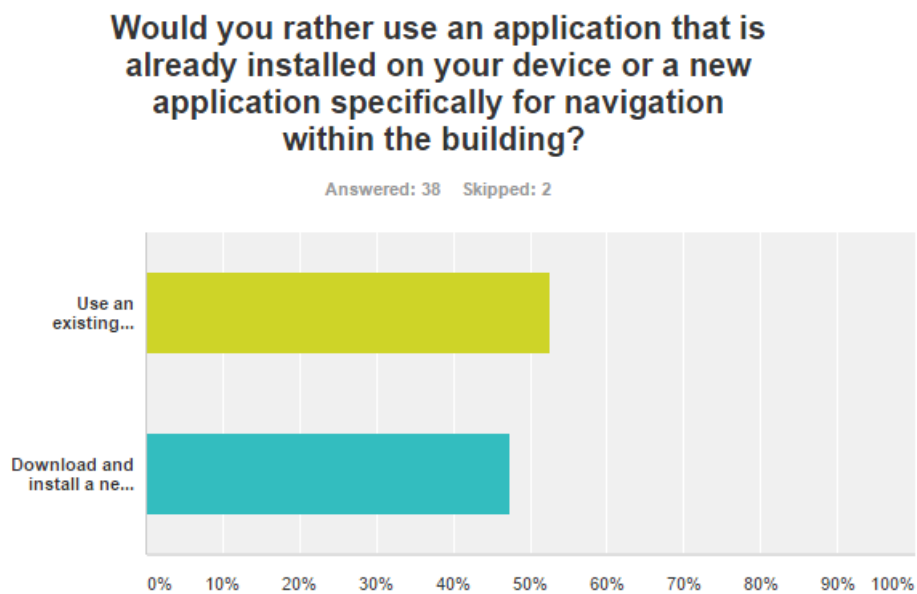
SurveyMonkey – Correct as of 10/02/2017

E)



SurveyMonkey – Correct as of 10/02/2017

F)



SurveyMonkey – Correct as of 10/02/2017

G)

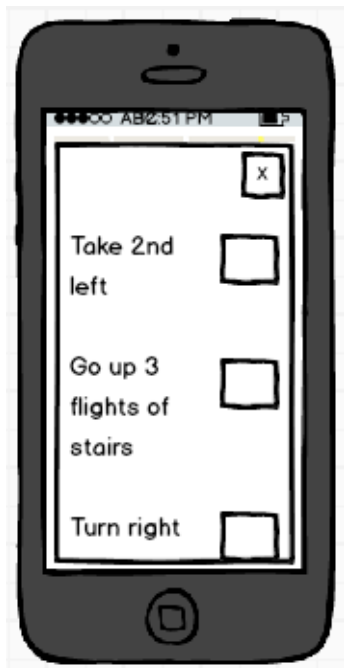


H)

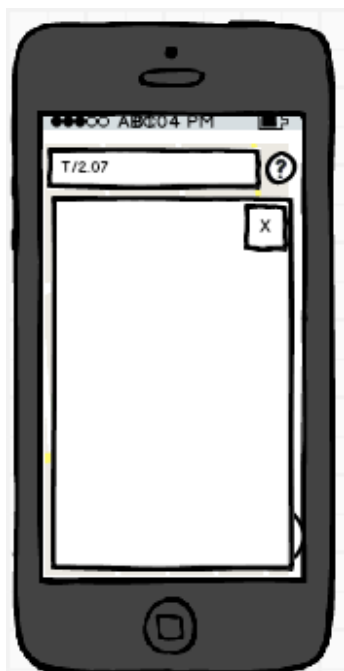




I)



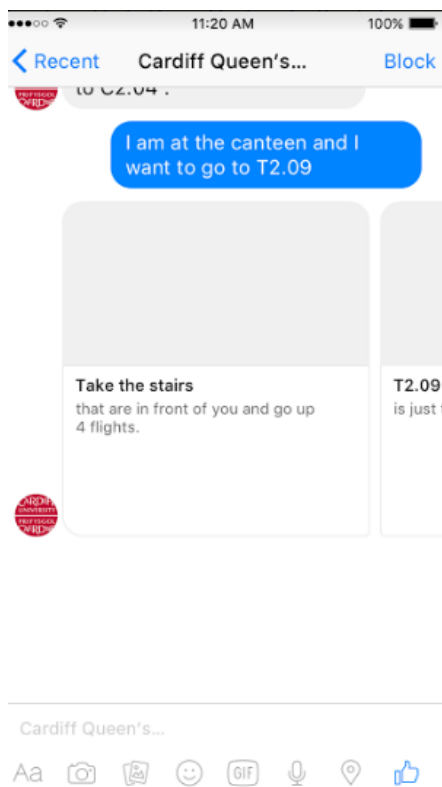
J)



K)



L)



M)



N)

### Create a New App ID

Get started integrating Facebook into your app or website

Display Name

CU Navigational Aid

Contact Email

melissawhiting11@googlemail.com

Category

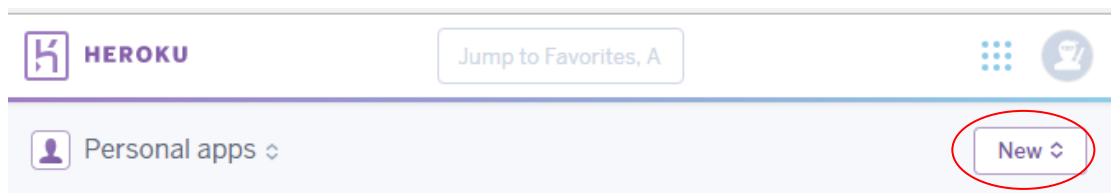
Apps for Messenger

By proceeding, you agree to the Facebook Platform Policies

Cancel

Create App ID

O)



P)

### Token Generation

Page token is required to start using the APIs. This page token will have all messenger permissions even if your app is not approved to use them yet, though in this case you will be able to message only app admins. You can also generate page tokens for the pages you don't own using Facebook Login.

Page

Select a Page ▼

Page Access Token

You must select a Page to generate an access token.

[Create a new page](#)

Q)

## Webhooks

New Subscription

Page Subscription

×

Callback URL

https://radiant-beyond-52321.herokuapp.com/webhook

Edit

Fields

message\_deliveries, message\_echoes, message\_reads, messages, messaging\_account\_linking, messaging\_checkout\_updates, messaging\_game\_plays, messaging\_optins, messaging\_payments, messaging\_postbacks, messaging\_pre\_checkouts, messaging\_referrals

Edit

To configure payment subscriptions, go to [Web Payments](#).

R)

Deployment method



Heroku Git  
Use Heroku CLI



GitHub  
Connected



Dropbox  
Connect to Dropbox

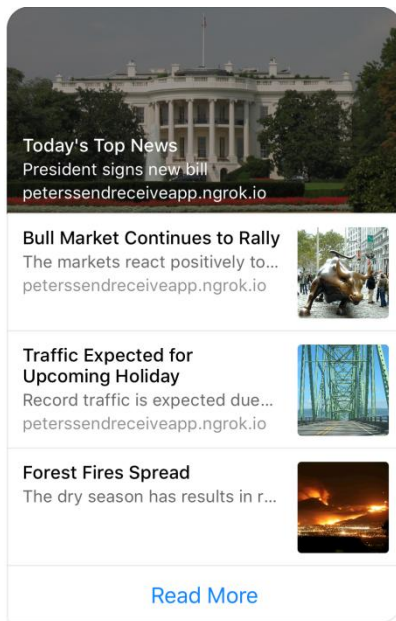
App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [melissa12/facebookbot](#)

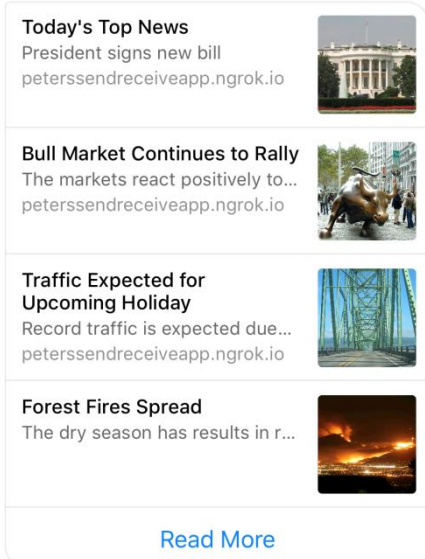
- ✓ Releases in the [activity feed](#) link to GitHub to view commit diffs
- ✓ Automatically deploys from master

S)



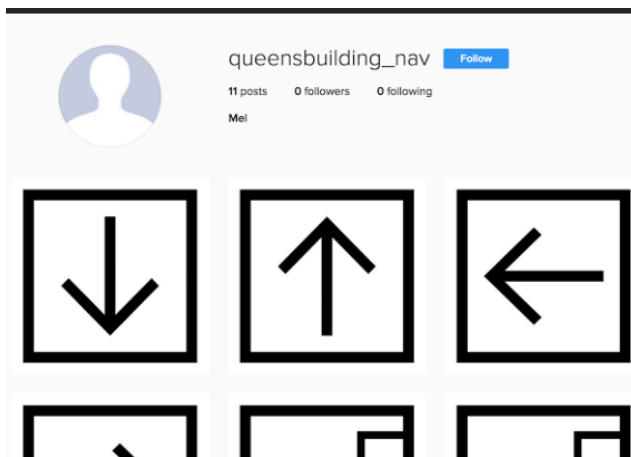
Facebook for developers g 2016

T)



Facebook for developers g 2016

U)



V)

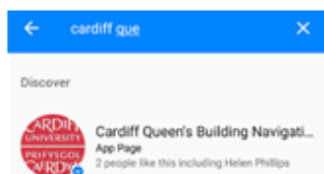
#### Before the testing

Installing the messenger app if you don't already have it on your device:

Go to the play store (or app store, depending on your device OS). Once in the store, search for 'Messenger'. Install this application. (If you are unsure if the application you are installing is the correct application, then please ask the observer to verify that it is).

Finding the Bot on the messenger app:

Once the 'Messenger' application is installed, open the app and click on the top search bar. Enter 'Cardiff Queen's Building Navigational Aid' and select the result as shown below:



A welcome message should be displayed.

What to do if the welcome message isn't displayed:

If the welcome message isn't displayed then type and send 'help' to the Bot. The Bot should then provide you with details about how to use the Bot for navigational purposes.

You are required to find the following rooms:

- S1.32
- N4.07
- T2.07

W)

## Questionnaire

Please circle your answer from the choice below.

What type of OS is your mobile?

iOS

Android

Windows

Other

Did you already have ~~facebook~~ messenger installed on your device?

Yes

No

Was the chat ~~bot~~ easy to find on your device?

Yes

No

Was the welcome message displayed? Did you know how to interact with it?

Yes

No

Did you receive directions from the ~~bot~~ within seconds?

Yes

No

Were the directions easy to understand?

Yes

No

Comments:

Did you find the desired location? How long did it take you?

Yes

No

If yes, time taken:

Overall, was the ~~bot~~ easy to use?

Yes

No

Comments:

If you were a first year university student, could you see this being useful and something you would make use of?

Yes

No

Comments:

## 10. References

Accessible Pedestrian Signals. *Travel Tools and Techniques of People Who are Blind or Who Have Low Vision*. [Online]. Available at: [http://www.apsguide.org/chapter2\\_travel.cfm](http://www.apsguide.org/chapter2_travel.cfm) [Accessed: 03/02/2017].

A, Jodlowski. X-TEAM. 2016. *How To Get Started With Facebook Messenger Bots*. [Online]. Available at: <https://x-team.com/blog/how-to-get-started-with-facebook-messenger-bots/>. [Accessed: 01/03/2017].

American Foundation for the Blind. 2016. *What Is Braille?* [Online]. Available at: <http://www.afb.org/info/living-with-vision-loss/braille/what-is-braille/123> [Accessed: 03/02/2017].

American Foundation for the Blind. 2016. *Screen Readers*. [Online]. Available at: <http://www.afb.org/prodBrowseCatResults.aspx?CatID=49> [Accessed: 20/04/2017].

Apple a. 2014. *Getting Started with iBeacon*. [Online]. Available at: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf> [Accessed: 16/02/2017]

Apple b. *iBeacon for Developers – Apple Developer*. [Online]. Available at: <https://developer.apple.com/ibeacon/> [Accessed: 18/02/2017].

BBC. 2009. *A History of Navigation*. [Online]. Available at: [http://www.bbc.co.uk/history/interactive/animations/navigation/index\\_embed.shtml](http://www.bbc.co.uk/history/interactive/animations/navigation/index_embed.shtml) [Accessed: 28/01/2017].

Bluetooth a. *What is Bluetooth?* [Online]. Available at: <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works> [Accessed: 30/01/2017].

Bluetooth b. *Bluetooth Low Energy*. [Online]. Available at: <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy> [Accessed: 30/01/2017].

B, Ondrisek. Medium. 2016. *Privacy and Data Security of Chatbots*. [Online]. Available at: <https://medium.com/@electrobabe/privacy-and-data-security-of-chatbots-6ab87773aadcf#.jcutn2iyd> [Accessed: 22/02/2017].

D, Fuller. Android Headlines. 2017. *Facebook Runs Limited Test Of Carousel Ads In Messenger*. [Online]. Available at: <https://www.androidheadlines.com/2017/01/facebook-runs-limited-test-of-carousel-ads-in-messenger.html> [Accessed: 23/03/2017].

D, Lee. BBC. 2017. *This 14-year-old made the best Facebook Messenger chatbot*. [Online]. Available at: <http://www.bbc.co.uk/news/technology-39013950> [Accessed: 20/02/2017].



E. Gaudlitz, Infsoft. 2015. *Indoor navigation & indoor positioning using Bluetooth*. [Online]. Available at: <https://www.infsoft.com/blog-en/articleid/41/indoor-navigation-indoor-positioning-using-bluetooth> [Accessed: 30/01/2017].

E. Gaudlitz, Infsoft. 2016. *Indoor tracking using beacons or RFID – what are the differences?* [Online]. Available at: <https://www.infsoft.com/blog-en/articleid/60/indoor-tracking-using-bacons-or-rfid-what-are-the-differences> [Accessed: 30/01/2017].

Facebook for developers a. 2016. *Messenger Platform*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform> [Accessed: 20/02/2017].

Facebook for developers b. 2016. *Launch*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform/product-overview/launch> [Accessed: 21/02/2017].

Facebook for developers c. 2016. *Platform Policy Overview*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform/policy-overview> [Accessed: 21/02/2017].

Facebook for developers d. 2016. *Platform Guidelines*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform/guidelines> [Accessed: 21/02/2017].

Facebook for developers e. 2016. *Quick Start*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform/guides/quick-start/> [Accessed: 02/03/2017].

Facebook for developers f. 2016. *Message Templates*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform/send-api-reference/templates> [Accessed: 23/03/2017].

Facebook for developers g. 2016. *List Template*. [Online]. Available at: <https://developers.facebook.com/docs/messenger-platform/send-api-reference/list-template> [Accessed: 23/03/2017].

Facebook for developers h. 2016. *Messages not sent in order – Facebook for Developers*. [Online]. Available at: <https://developers.facebook.com/bugs/565416400306038> [Accessed: 24/03/2017].

Facebook Help Centre. 2017. *Accessibility for People with Disabilities*. [Online]. Available at: <https://www.facebook.com/help/141636465971794> [Accessed: 20/04/2017].

Garmin. *About GPS*. [Online]. Available at: <http://www8.garmin.com/aboutGPS/> [Accessed: 29/01/2017].

GCF Learn Free. *What is Google Maps?* [Online]. Available at: <http://www.gcflearnfree.org/google-maps/what-is-google-maps/1/> [Accessed: 10/02/2017].

G. Gruman, Infoworld. 2014. *What you need to know about using Bluetooth beacons.* [Online]. Available at: <http://www.infoworld.com/article/2608498/mobile-apps/what-you-need-to-know-about-using-bluetooth-beacons.html?height=921.6&width=921.6> [Accessed: 30/01/2017].

Google Maps a. *About – Google Maps.* [Online]. Available at: <https://www.google.com/maps/about/> [Accessed: 10/02/2017].

Google Maps b. *Indoor Maps – About – Google Maps.* [Online]. Available at: [https://www.google.com/intl/en\\_uk/maps/about/partners/indoormaps/](https://www.google.com/intl/en_uk/maps/about/partners/indoormaps/) [Accessed: 10/02/2017].

G, Seco-Granados et al. 2012. *Challenges in Indoor Global Navigation Satellite Systems.* [Online]. Cite Seer X, IEEE Signal Processing Magazine 17 February 2012, p. 1. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.476.7790&rep=rep1&type=pdf> [Accessed: 05/02/2017].

G tricks. 2016. *Difference Between Google Maps, Google Street View and Google Earth.* [Online]. Available at: <http://www.gtricks.com/google-maps-tricks/google-maps-vs-street-view-vs-google-earth/> [Accessed: 10/02/2017].

iBeacon Insider. 2014. *What is iBeacon? A Guide to Beacons.* [Online]. Available at: <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/> [Accessed: 16/02/2017].

Infsoft a. *Indoor Positioning, Tracking and Indoor Navigation with Beacons.* [Online]. Available at: <https://www.infsoft.com/technology/hardware/beacons> [Accessed: 18/02/2017].

J, Sieck et al. IEEE Xplore Digital Library. 2014. *Location-Based Services with iBeacon Technology.* [Online]. Available at: <http://ieeexplore.ieee.org.abc.cardiff.ac.uk/xpls/icp.jsp?arnumber=7102480> [Accessed: 16/02/2017].

KNOWHOW. 2012. *SAT NAV AND GPS – ARE THEY THE SAME THING?* [Online]. Available at: <http://knowhow.com/article.dhtml?articleReference=1866&country=uk> [Accessed: 10/02/2017].

K, Rice. Kinvey. 2013. *How Long Does it Take to Build a Mobile App?* [Online]. Available at: <https://www.kinvey.com/how-long-to-build-an-app-infographic/> [Accessed: 16/02/2017].

L, Kelion. BBC. 2016. *Sat-nav maps update aims to prevent stuck lorries.* [Online]. Available at: <http://www.bbc.co.uk/news/technology-37703556> [Accessed: 10/02/2017].

M, Burgess. Wired. 2015. *Euston's new beacons will help blind people navigate the Tube*. [Online]. Available at: <http://www.wired.co.uk/article/wayfindr-navigation-visually-impaired-euston> [Accessed: 13/02/2017].

M, Byrne. Motherboard. 2015. *The Simple, Elegant Algorithm That Makes Google Maps Possible*. [Online]. Available at: [https://motherboard.vice.com/en\\_us/article/the-simple-elegant-algorithm-that-makes-google-maps-possible](https://motherboard.vice.com/en_us/article/the-simple-elegant-algorithm-that-makes-google-maps-possible) [Accessed: 10/02/2017].

M, Druyan. Spreo. 2014. *Indoor Navigation Challenges Series – Chapter 1: Finding an accurate location indoors*. [Online]. Available at: <http://spreo.co/indoor-navigation-challenges-series-chapter-1-finding-an-accurate-location-indoors/> [Accessed: 05/02/2017].

Minimum Wage. *National Minimum Wage 2017 Rates*. [Online]. Available at: <http://www.minimum-wage.co.uk/> [Accessed: 19/02/2017].

M. Orth, Techwalla. 2015. *Advantages and Disadvantages of GPS Systems*. [Online]. Available at: <https://www.techwalla.com/articles/advantages-and-disadvantages-of-gps-systems> [Accessed: 29/01/2017].

M, Petronzio. Mashable. 2012. *11 Fascinating Facts About Google Maps*. [Online]. Available at: [http://mashable.com/2012/08/22/google-maps-facts/#\\_9aWB5079Zql](http://mashable.com/2012/08/22/google-maps-facts/#_9aWB5079Zql) [Accessed: 10/02/2017].

M, Rouse. SearchSoftwareQuality. 2007. *Waterfall model*. [Online]. Available at: <http://searchsoftwarequality.techtarget.com/definition/waterfall-model> [Accessed: 27/01/2017].

Packetworks. *Common Causes of WiFi Interference*. [Online]. Available at: <http://packetworks.net/blog/common-causes-of-wifi-interference> [Accessed: 21/02/2017].

P, Bharuch et al. SlideShare. 2015. *SCHOOL MANAGEMENT SYSTEM*. [Online]. Available at: <http://www.slideshare.net/faisal2204/dijkstras-algorithm-47571607> [Accessed: 13/02/2017].

R, Nightingale. Make Use Of. 2017. *How Does Google Maps Work?* [Online]. Available at: <http://www.makeuseof.com/tag/technology-explained-google-maps-work/> [Accessed: 10/02/2017].

RSBC. 2015. *WAYFINDR*. [Online]. Available at: <http://www.rsbc.org.uk/tech-hub/wayfindr/> [Accessed: 13/02/2017].

Sandys View, The Chicago Lighthouse. 2015. *How Do Blind and Visually Impaired People Get Around?* [Online]. Available at: <http://chicagolighthouse.org/sandys-view/getting-around/> [Accessed: 03/02/2017].

satnav.co.uk. *How it Works*. [Online]. Available at: <http://www.satnav.co.uk/how-it-works.php> [Accessed: 10/02/2017].

S, Nunn. Grow. 2015. One Million Facebook Video Calls in Two Days: Social Media News. [Online]. Available at: <http://www.growonlinemarketing.co.uk/one-million-facebook-video-calls-in-two-days-social-media-news/> [Accessed: 23/03/2017].

S, Rosenberg. Facebook for developers. 2016. *How To Build Bots for Messenger*. [Online]. Available at: <https://developers.facebook.com/blog/post/2016/04/12/bots-for-messenger/> [Accessed: 20/02/2017].

SurveyMonkey. 2009. *Everything You Wanted to Know, But Were Afraid to Ask*. [Online]. Available at: <https://www.surveymonkey.co.uk/mp/aboutus/> [Accessed: 10/02/2017].

Teach-ICT. 11. *Navigation satellites – pros and cons*. [Online]. Available at: [http://www.teach-ict.com/as\\_a2\\_ict\\_new/ocr/A2\\_G063/333\\_networks\\_coms/satellite\\_systems/miniweb/pg11.htm](http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/333_networks_coms/satellite_systems/miniweb/pg11.htm) [Accessed: 10/02/2017]

Technovelgy a. *What is RFID?* [Online]. Available at: <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=1> [Accessed: 30/01/2017].

Technovelgy b. *How RFID Works*. [Online]. Available at: <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=2> [Accessed: 30/01/2017].

Technovelgy c. *Is RFID Technology Secure and Private?* [Online]. Available at: <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=5> [Accessed: 30/01/2017].

Technovelgy d. *Are There Concerns About How RFID Will Be Used? (Update)*. [Online]. Available at: <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=6> [Accessed: 30/01/2017].

Technovelgy e. *Problems With RFID*. [Online]. Available at: <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=20> [Accessed: 30/01/2017].

Technovelgy f. *Advantages of RFID Versus Barcodes*. [Online]. Available at: <http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=60> [Accessed: 30/01/2017].

University of Washington Computer Science and Engineering. *Dijkstra's Algorithm Continued*. [Online]. Available at:

<https://courses.cs.washington.edu/courses/cse326/06au/lectures/lect22.pdf> [Accessed: 13/02/2017].

Wayfindr. *Empowering vision impaired people*. [Online]. Available at: <https://www.wayfindr.net/> [Accessed: 13/02/2017].

WebWise Team, BBC. 2012. *What is bluetooth?* [Online]. Available at: <http://www.bbc.co.uk/webwise/guides/about-bluetooth> [Accessed: 30/01/2017].

Wilde Beuger Solmecke. 2014. *The legal issues surrounding the iBeacon*. [Online]. Available at: <https://www.wbs-law.de/eng/commercial-law/legal-issues-surrounding-ibeacon-54413/> [Accessed: 16/02/2017].

Xamarin. *Mobile App Development and App Creation Software*. [Online]. Available at: <https://www.xamarin.com/> [Accessed: 19/02/2017].