

**Honours Project (CCIS) Final Report**

**2018-2019**

**Submitted for the Degree of:**

**BSc Computing**

**Project Title: LOCATION AWARENESS WITHIN BUILDINGS USING AN INSTALLED WI-FI NETWORK AND ANDROID**

**Name: DARYL MCALLISTER**

**Programme: COMPUTING**

**Matriculation Number: S1222204**

**Project Supervisor: IAIN LAMBIE**

**Second Marker: Richard Holden**

**Word count : 10,932**

**“Except where explicitly stated, all work in this report, including the appendices, is my own original work and has not been submitted elsewhere in fulfilment of the requirement of this or any other award”**

**Signed by Student:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_**

# Abstract

Mobile application development has seen huge increases in recent years due to the front row seat that mobile devices have taken in modern life and the constant push to streamline it. Location services have been one of the area’s driving trends; with the ability to accurately locate an individual indoors being a large new emerging request from users and has been investigated for some time - with most technologies requiring specific hardware or being too inaccurate. Indoor positioning systems are becoming an important investment for many organizations or groups of individuals, one example being students for finding class rooms quickly. The ability to base an indoor positioning system to function with existing technology and installed infrastructure would help benefit these groups if this was a feasible technology.

This “Develop and Test” project investigated how accurate an indoor positioning system can be using only WIFI signals from router points, on an android mobile device. To evaluate the project an android application was developed that could show a user’s position on a virtual map of the building. The accuracy of the application was determined from field testing known locations against the applications location points. The results from the testing provided evidence that supports that a WiFi based indoor location system is feasible to an extent with an average overall accuracy of within 10.55 meters which could be developed further in the future to provide more accurate results.

# Acknowledgements

I would like to thank Iain Lambie, project supervisor, for the help and guidance to bring this project to fruition through both development and write-ups.

Additionally, I would also like to show my appreciation to Aaron and Lola Whitters for their role in proof reading and providing support to the project.

Contents

[1. Introduction 5](#_Toc6327896)

[1.1. Project Background 5](#_Toc6327897)

[1.2. Project Outline 8](#_Toc6327898)

[1.2.1. Project Aim and Method 8](#_Toc6327899)

[1.2.2. Objectives 9](#_Toc6327900)

[1.3. Hypothesis 9](#_Toc6327901)

[1.4. Project Structure 10](#_Toc6327902)

[Literature Review and Technology Assessment 10](#_Toc6327903)

[Execution 10](#_Toc6327904)

[Evaluation and Discussions 10](#_Toc6327905)

[Conclusions 10](#_Toc6327906)

[2. Literature Review and Technology Assessment 11](#_Toc6327907)

[2.1. Examine the accuracy of other forms of indoor positioning 11](#_Toc6327908)

[2.1.1. Bluetooth Low Energy Beacons 11](#_Toc6327909)

[2.1.2. Ultrasonic 12](#_Toc6327910)

[2.2. Investigate Wi-Fi and Signal Technology 13](#_Toc6327911)

[2.2.1. Wi-Fi and Routers Applications 13](#_Toc6327912)

[2.2.2. RSSI fingerprinting 14](#_Toc6327913)

[2.2.3. Trilateration 14](#_Toc6327914)

[2.3. Investigate the Android Platform 15](#_Toc6327915)

[2.3.1. WiFi on Android 15](#_Toc6327916)

[2.3.2. Android Development 16](#_Toc6327917)

[3. Execution 17](#_Toc6327918)

[3.1. Lifecycle 17](#_Toc6327919)

[3.2. Analysis and Design 17](#_Toc6327920)

[3.3. Implementation 18](#_Toc6327921)

[3.3.1. Lifecycle Iterations 19](#_Toc6327922)

[3.3.2. Indoor Map Display 19](#_Toc6327923)

[3.3.3. Algorithm Implementation 20](#_Toc6327924)

[WiFi Scan and Distance calculation 20](#_Toc6327925)

[Data Bulking and Trilateration 21](#_Toc6327926)

[3.4. Application Testing 23](#_Toc6327927)

[4. Evaluation 24](#_Toc6327928)

[4.1. Research Testing Methodology 24](#_Toc6327929)

[4.2. Results Examination 25](#_Toc6327930)

[4.2.1. XYZ Accuracy 25](#_Toc6327931)

[4.2.2. Room positioning and reliability 26](#_Toc6327932)

[4.3. Comparisons to other technologies 27](#_Toc6327933)

[4.4. Conclusions 28](#_Toc6327934)

[5. Conclusions and further work 29](#_Toc6327935)

[6. References 31](#_Toc6327936)

[7.0 Appendixes 35](#_Toc6327937)

# Table of Figures

[Figure 1 - PokemonGo Location Game (Niantic, 2018) 5](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328283)

[Figure 2- BLE beacon positioning system (infsoft, 2015) 6](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328284)

[Figure 3 - Trilateration of signals from router points 7](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328285)

[Figure 4 - Comparison of real-life distance and calculated distance in a testing of a BLE beacon indoor positioning system. (Herrera Vargas, 2016). 11](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328286)

[Figure 5 - Diagram of an ultrasonic indoor positioning system (Chakraborty, 2000) 12](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328287)

[Figure 6 - Comparison table of capabilities of some of the most recent amendments to the 802.11 standard for Wi-Fi (Farnell element14, 2017). 13](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328288)

[Figure 7 - Trilateration Diagram 14](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328289)

[Figure 8 - Lifecycle Diagram 17](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328290)

[Figure 9 - Initial wireframes 18](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328291)

[Figure 10 - MapWize application example 19](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328292)

[Figure 11 - Projects distance formula 21](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328293)

[Figure 12 - Graphic representation of 3D trilateration 22](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328294)

[Figure 13 - Initial testing environment 23](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328295)

[Figure 14 - Screenshot of testing 24](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328296)

[Figure 15 - Graph showing floor guesses by floor 26](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328297)

[Figure 16 - Heatmap of floor router coverage 26](file:///C:\Users\Daryl\Downloads\aaaatest.docx#_Toc6328298)

[Table 1 - Error margin by floor 25](#_Toc6328299)

[Table 2 - Floor accuracy by floor 25](#_Toc6328300)

[Table 3 - room accuracy by floor 27](#_Toc6328301)

# List of Abbreviations

BLE – Bluetooth Low Energy

GHz – Gigahertz

GPS – Global Positioning System

IDE – Integrated Development Environment

IEEE - Institute of Electrical and Electronics Engineers

IOT – Internet of Things

NFC – Near-field Communication

OS – Operating System

RF – Radio Frequency

RFID – Radio-frequency Identification

RSSI – Received Signal Strength Indication

UI – User Interface

USD – United States Doller

RTT – Round Trip Time

FPL – Free Path Loss

# 1. Introduction

## 1.1. Project Background

Mobile devices have become more prevalent in our modern life with almost 92% of the world’s population now owning a mobile device (Grimus & Ebner, 2015) and the use of smartphones has become so integrated , it is one of the first and last things we use each day with 36% - 40% of users checking their phone within 5 minutes of waking up and going to bed (Montag et al., 2015). The ever presence of these devices has led to a constant strive to develop applications that can help streamline everyday life and make as many services available to users as possible. This potential for integrated use of mobile applications was even shown by head of state figures such as the then President Obama, who’s pledging to have two public services available through mobile phones at his speech in 2012 (Rakestraw et al., 2012) was a boost for the industry. The boom of the mobile market has given rise to a more apparent application trends and their integration into everyday life. This boom has led to a huge sector for developing applications with many methods implementation ways that can be used to implement them on different platforms. This sector is also seeing several driving trends that are becoming more demanded by users

One of these trends in application development and the market currently is location positioning and location-based services (Fedorychak, 2018). While not a new technology, with the first appearance in 2001 (Michael, 2004), it is the advancement in technology and growth of the mobile market that has given way to the ability to integrate location positioning into many applications, from direction applications like Google maps to even gaming such as the handheld game PokemonGo, as shown in figure 1. The wide implementation of location services into many products has led to one third of all mobile searches being location related (Ramaswamy, 2016).



Figure 1 - PokemonGo Location Game (Niantic, 2018)

Location services have also found implementation in applications that are not directly linked to location. Social media applications have adopted location services to show users their friends in proximity or even filter content to show relevant nearby information (Scassa, Sattler 2011). Adoption of location services has remained in the outdoor or world position. These integrations have created a major market because of the convenience they cause for users whether it is getting them from point A to point B or giving them a glance at how near they are to a desired goal. In terms of games they have also found the novelty of exploration in location games helps with enjoyment and learning (Wake, 2013) but in recent years the market for location based systems has seen a large increase in interest for indoor localization as well; this can be due to the performance and abilities of outdoor positioning being already established (Mautz, 2012) and the market already currently utilizing it. Indoor location positioning is a natural advancement in positioning technology that has generated interest from users and researchers helping shift potential from outdoor to indoor positioning (Nhlanhla, Adeyeye-Oshin et al. 2017)

The current potential for indoor positioning is also leading its growth as it opens the ability for a diverse range of applications with functionality that cover a large area to be developed. Companies in the Internet of Things(IoT) area can utilize it for asset tracking (Hui Liu, Darabi, Banerjee & Jing Liu, 2007), universities could implement a system that provides positioning for students – particularly those with disabilities i.e. to find the most accessible route (G. Delnevo et al., 2018) or in a landmark setting where visitors can be shown their position in an area and nearby points of interest. Further potentials for indoor positioning could be the progression from standard positioning in a building to the ability to show users paths to a specified destination or even the ability to calculate the time it will take to get between two points in a building. The progression of the technology has many knock-on effects that it can solve such as cutting down time getting lost in airport terminals or getting between classes at an educational institute. While systems for the previous examples exist in static maps readily available, it is an analogue system that many people still find difficult to access and it can be time consuming to locate themselves on or to find the best route.

Even with the ability to adhere this technology to many helpful everyday systems, the emergence of systems like these has been slow due to the problems that currently affect indoor localization - the largest problem being a reliable set technology that is easily applied. Global Positioning System (GPS) is the main technology used for world positioning (Montenbruck, Ramos-Bosch 2008) and is used by Google to position users in their Google Maps application (Garude, Haldikar 2014). While GPS can also obtain a position indoors it’s usually obscured by walls and building material which make it inaccurate (Farid, Nordin & Ismail, 2013), this has required other technologies to be developed to allow a more accurate indoor positioning system to be feasible.

The various current technologies include Bluetooth sensors, Ultrasonic positioning, Radio Frequency Identification (RFID), and Wi-Fi (He Xu et al., 2017). RFID currently has numerous implementations but the consistent set up is to have the individual wear RFID tags since most modern mobile devices use Near-Field Communication (NFC) instead which has a range of 4 inches (Chandler, 2012). The RFID tags would interact with scanners in the building which relay back to a server from which the device would be connected, giving a user their position (Y Ortakci, E Demiral, U Atila & I R Karas, 2015). This method brings downsides which have been witnessed in its already widespread use - in areas such as supermarkets, it may be unfeasible to provide every individual in the building the required tags to wear and it can become costly. Amazon has set up a supermarket that tracks products and individuals in a store to provide a checkout free experience using cameras and RIFD tags (Likhitha, Anusha et al. 2018). This shows the real-life implications are possible despite the downsides but the accuracy of such a system relies on many RFID scanners placed within the building which is expensive and not viable for smaller enterprises to adopt for indoor localization as this does not encompass the labour costs. Amazon’s store also uses the CCTV to aid the RIFD detections which would produce a complex and bulky system to maintain. One of the reasons RFID tags have wide spread use, especially in the agriculture area, is due to the robustness of the tags in a harsh environment (Ruiz-Garcia, Lunadei 2011) yet such robustness would not be needed indoors.



Figure 2- BLE beacon positioning system (infsoft, 2015)

The largest growing technology for indoor positioning is Bluetooth Low Energy (BLE) beacons. This developing technology has seen uptake due to its use of BLE which can have high throughput and be more power efficient compared to other communication protocols (Rondón, Gidlund & Landernäs, 2017). Usual set-ups for beacon indoor positioning systems as exemplified in figure 2 are consisted of multiple beacons in an area that will interact with a device which may use a database to receive the positions of the beacons and triangulate its position (infsoft, 2015). Many companies are providing software to develop systems like this due to its prevalence and ability to be used beyond positioning systems. Problems still arise with positioning systems based on this concept due to excessive costs and labour to fit a building with beacons. Bluetooth also must be active to allow this system to work whereas only 60% of mobile users have their Bluetooth active (Coombs, 2017), meaning users can be missed out in this system. Another factor holding back beacons from dominating the indoor positioning development productions is that beacons have security issues that have yet to be ironed out like piggybacking (Gąsiorek, 2016) where users tailgate another person’s data. This goes against current trends to ensure all produced applications are secure (KHARCHENKO, 2017).

These costly and labour-intensive technologies have led to investigations into the possible use of an established system such as Wi-Fi signals from pre-installed router points in a building which would connect to a device instead of using proprietary hardware. Making use of the potentials of Wi-Fi technology with such a system would save organizations valuable time and money because of the use of an already functioning systems to provide the indoor localization. This would allow the integration of indoor positioning in any building that had a router network by allowing the Wi-Fi signals to be received to a device and location determined through a trilateration technique (Bianca Bobescu & Marian Alexandru, 2015) such as the one shown in figure 3. Most major public buildings have Wi-Fi installed due to continuing public expectations of readily available Wi-Fi with 40% of people considering no Wi-Fi as a deal breaker when booking a hotel (boostandco, 2017). This expectation means the adoption of such a technology could be accomplished due to the existence of a popular platform which could solve the problem of costly and sometimes inaccurate indoor positioning that other systems currently provide. While this shows a clear benefit to a Wi-Fi signal-based system compared to other technologies there will still have to be investigations into the accuracy of this technology to ensure it is a workable method of positioning an individual indoors as well as ensuring it can be as accurate as other forms of indoor positioning.

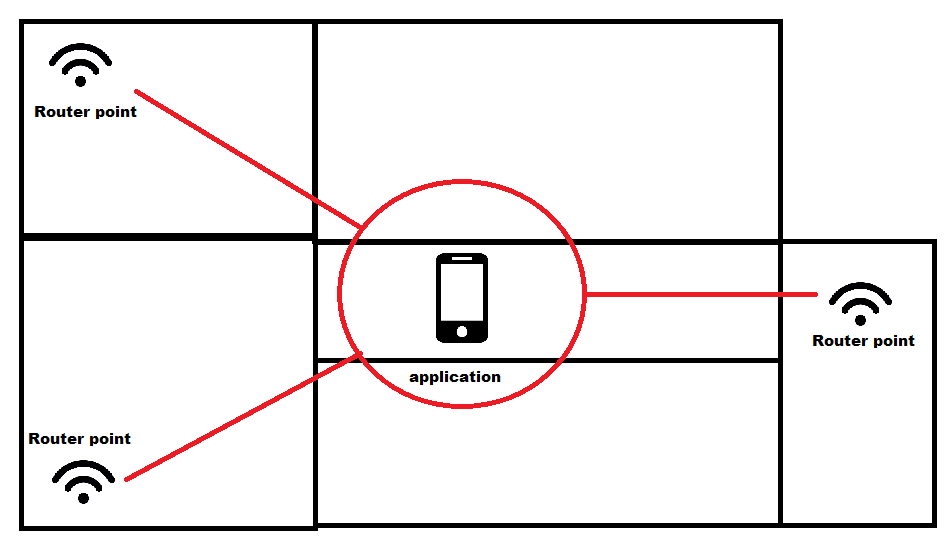


Figure 3 - Trilateration of signals from router points

## 1.2. Project Outline

This section further details the explicit aim of this project and the research question. It will also provide the list of research objectives that will build up the literature review and provide a hypothesis.

### 1.2.1. Project Aim and Method

This project aims to determine the feasibility of an indoor positioning system using received Wi-Fi signals from router points and the ability to use it for mass market Android applications with the results having the potential to open the existing technology in new directions without the addition of specialised hardware. The project was executed by developing an application and testing it to allow investigation of the potential feasibility.

The application developed was android based and allows a user to view the test building and determine their current location within it by using the WiFi signals from the existing router infrastructure. The location aspects of the project made it clear that android would be ideal as there is a plethora of android devices and it has a large support base with open source-esque quality. The testing was initially devised as group user testing but was changed as it was out of scope for the project, instead the application was tested within the test building with results taking the foreground. The test environment chosen was a university building due to the real-life relevance such an environment would have. The project only sought to test the application on the top three floors of the building in the evening time as this allowed for less interference giving the testing results more weight on the technology which was more relevant to the project. Testing sought to look at the technologies ability to position a user accurately in both floor number and XY position on that floor, additional abilities such as room positioning was also marked for supplementary evidence.

The research question that will drive this project is:

**How accurate can an indoor positioning system be using the received Wi-Fi signals from access points by an Android mobile device?**

### 1.2.2. Objectives

This section highlights the secondary research objectives that will make up this project. These objectives will be further explored in more detail within this report.

**Objectives to be completed of a main literature review:**

SO1: Examine the accuracy of other forms of indoor positioning

The project’s examination of the feasibility of Wi-Fi accuracy must be done against the accuracy of other systems such as the BLE or RFID. The accuracy and attributes of those systems should be examined to compare against the Wi-Fi indoor positioning.

SO2: Investigate Wi-Fi and signal technology.

Due to the project’s reliance on Wi-Fi signals, it is important to research an overview of Wi-Fi and network technology to gain an understanding of the systems that will allow the application to determine the device’s location.

SO3: Investigate the Android Platform

An understanding of Android systems is required for the development of the project. Investigations into WiFi on android technology and software related configurations are important for the project since it relies on the development of a WiFi Android application. All parts of this investigation will be used to assist implementation.

## 1.3. Hypothesis

Testing the hypothesis would involve testing the application in specified areas and mark how accurate the application is in determining their position, then comparing this accuracy against other forms of indoor positioning. The hypothesis for this project is:

The feasibility of using received Wi-Fi signals from access point by an Android device for an indoor positioning system will be accurate to an extent but will be less so than other forms of indoor positioning systems while also coming with its own forms of restrictions.

Justification for this hypothesis comes from the ability of Wi-Fi signals to fluctuate (Bai, Wu et al. 2014) which may affect the positioning of the application. This may not affect the accuracy as much however as the presence of multiple access points sending signals may offset this.

## 1.4. Project Structure

This section gives a brief description and overview of the following sections covered within this project and their contribution to the project overall.

### Literature Review and Technology Assessment

The literature review and technology assessment will look at previous studies and resources that relate to the secondary research objectives. This section hopes to identify related materials that can be used to further this project’s development and evaluations.

### Execution

This section will detail the execution that was required to develop the artefact that will be used to test the research question. It will detail the lifecycle used to develop it including the analysis, design and implementation of the artefact.

### Evaluation and Discussions

The evaluation and discussions section will outline the testing of the application against the research question and the results that were gained from this. These results will then be examined and compared against researched technologies to help answer the key issues and draw conclusions.

### Conclusions

The conclusions section aims to conclude on the aforementioned sections and give insight into enhancements to the project that could occur, and the future implementations of the technology as explored.

# 2. Literature Review and Technology Assessment

This section attempts to achieve an understanding which will be used to complete the project and gather the findings for the research question. The literature review section is undertaken by examining relevant work and research produced by others that will provide knowledge that can be carried forward into the primary research.

The literature review will explore the following research objectives (mentioned in section 1.2.3):

* Examine the accuracy of other forms of indoor positioning
* Investigate Wi-Fi and Router technology
* Investigate Android application development.

## 2.1. Examine the accuracy of other forms of indoor positioning

The project background briefly overviewed downfalls of current existing indoor positioning systems but it is also crucial to understand them further as well as examine how accurate they can be so that comparisons can be made between them and Wi-Fi positioning indoors. Examining other forms of positioning will give a clear indication of their ability and accuracy which will be useful for determining this technology’s potential feasibility once compared.

### 2.1.1. Bluetooth Low Energy Beacons

As previously mentioned in the project background, BLE beacons are a fast-evolving technology that has seen uptake by a wide variety of sectors from retail to aviation with “84% of global airports… running a pilot or implementing beacon technology by 2019” (unacast, 2016). Part of this adoption can be due to the expanded possibilities of BLE beacons, but this examination will focus on their indoor positioning aspects. BLE beacons work by emitting packets of data which can be detected by a smartphone where a connection is established (Park, Noh et al., 2016). The most common positioning method being received signal strengths of the beacons surrounding the device which are used to fingerprint the devices location based on them. The accuracy of beacons has been tested in many studies with one finding that accuracy was 74% with a margin of error of 1 meter (Huh, Seo, 2017). This was conducted in a single room with a floor space of only 28.8 meters² and a placement of 7 beacons in this area. The high density of beacons in such a small area skews the results and is not a true representation of how it would be used in a commercial area. In a more reflective examination of beacons where the testing environment was a 50m² apartment and had only two beacons separated by 5 meters, it was found that distances from the beacons that were greater than 3 meters gave results that were incorrect by over 2 meters, the full extent of which can be seen in figure 4 (Herrera Vargas, 2016). A reason the second and more realistic study found less accurate results could be that BLE beacons’ initial conception was designed for proximity detection rather than exact distance (lemberg, 2014). The beacons also can have lag in connections because of the power conservation as they were designed to be able to run only on batteries which was possible due to BLE’s low power consumption (lemberg, 2014). While this is useful for cable-less operating it makes the BLE beacons delayed in detecting and positioning a moving individual and the repositioning of an individual who has just stopped moving.

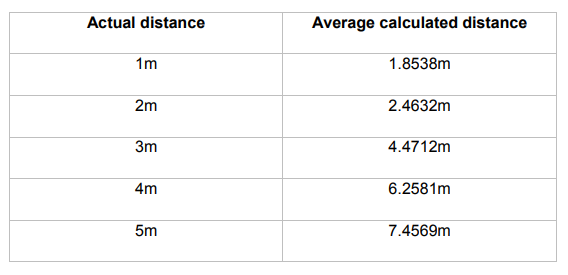


Figure 4 - Comparison of real-life distance and calculated distance in a testing of a BLE beacon indoor positioning system. (Herrera Vargas, 2016).

The BLE beacons are highly accurate when a large density of them are present but become less so when there are fewer in an area. This gives companies the decision to spend more on hardware installation or present a less accurate system to their customers. While beacons are getting cheaper, in high numbers the cost can inflate especially in maintenance costs. This paired with the delay of positioning means that beacons can have inaccuracies and downsides that this project hopes the investigation of Wi-Fi will provide an alternate to. The potential for Wi-Fi positioning to have a comparable accuracy but for a much cheaper implementation may make it a much better choice that should be investigated.

### 2.1.2. Ultrasonic

Ultrasonic is seen as a more complicated implementation of an indoor positioning system and there are many examples of implementation which have been researched. The most common researched is a process where individuals will carry nodes that have both radio frequency (RF) and ultrasonic capabilities. The nodes use the RF capabilities to synchronise with installed hardware which when synchronised will listen for an ultrasonic pulse from the node and when received can be used to calculate the individual’s position (Priyantha, Chakraborty et al., 2000). Figure 5 shows a diagram of this, but an RF Transmitter is used to initially give the receiver and transmitter an RF identity (Chakraborty, 2000). Initially from assumptions it would be logical to think such system would be obsolete since ultrasonic waves cannot penetrate walls and thick infrastructure (Gifford, 2018) but this may be one of ultrasonic technologies strong points. The inability to penetrate walls means that the room an individual is in is clearer cut and the system will provide a definite room location while with BLE beacons the signal attenuation can confuse individuals between rooms or floors (Gifford, 2018). Studies showed that system set ups like this can provide results with “a median error of less than 5cm” when used to track a moving model train around a room with 5 receiving sensors (Smith, Balakrishnan et al., 2004). Indoor positioning with accuracy so high is impressive especially when tracking a moving object, this can be due to the fast disperse and receive that it has, due to quick properties of ultrasonic sound waves. Downfalls become apparent when research into scaled versions takes place but due to the limits and propriety of the hardware, a lack of full-scale research has been done. A reason for this can be found in the main benefit of the system, the inability to penetrate walls; this is because enclosed area requires its own sensors which in larger builds can become expensive. This hardware is often bespoke, being made for such a system which inflates the cost and the time required for manufacturing the hardware. Much like the BLE beacons, a room may require several receiving nodes installed and in rooms with tall ceilings this becomes a large issue (Gifford, 2018). These pitfalls hinder ultrasonic from becoming more refined and gives way for newer technologies to take its place. Even if a system could be implemented on scale it still suffers downfalls that are like RFID where individuals must carry a device which also emits the ultrasonic pulse. This adds more hardware that is required for the system and while investigations were made into the potential of cell devices receiving ultrasonic pulses from transmitting beacons, it was less accurate (Smith, Balakrishnan et al., 2004).

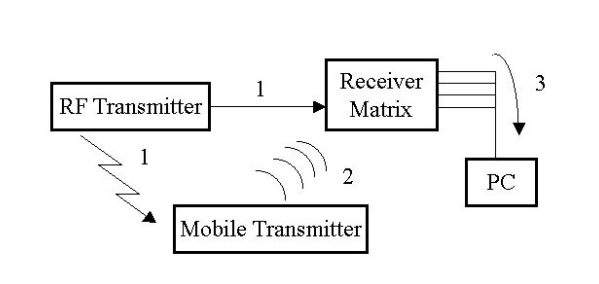


Figure 5 - Diagram of an ultrasonic indoor positioning system (Chakraborty, 2000)

While easier to position users within a specific room and accurate at plotting their point, on scale an ultrasonic system is unfeasible as it is bespoke and cumbersome. These limitations may be answered by Wi-Fi based positioning that uses existing router hardware which would be easier for large implementations and be able to have wide spread use.

## 2.2. Investigate Wi-Fi and Signal Technology

The project’s aim to find the feasibility of Wi-Fi indoor positioning dictates that an examination of the core concepts of Wi-Fi and router technology is explored as well as the methods of processing them. This research will give merit of the potential of the technology and explore how positioning may be determined from it.

### 2.2.1. Wi-Fi and Routers Applications

Since the introduction of the 802.11 standard by the Institute of Electrical and Electronics Engineers (IEEE) in 1997 (Tambe, 2015) and its initial introduction to homes in 1999 (Thomas, 2014) Wi-Fi has boomed and “there are now more Wi-Fi devices in use than there are people on earth” (Farnell element14, 2017). This growth has led to affordable and integrated hardware that is an often-expected facility provided by companies or necessary utility in households. Growing demands and further developments of the technology have provided improvements to the 802.11 standards; the improvements can be seen in figure 6: (Farnell element14, 2017).

Figure 6 shows that each release of the 802.11 standard improves upon the previous and that 802.11n, one of the most used in terms of commercial hardware, provides increases in speed and more importantly - range. This has opened the potential use for Wi-Fi from internet connection to other areas such as enabling wireless intranets to be utilized or the significant role Wi-Fi plays in the IoT. A 2017 study into the potential to create a low-cost home automation system using Wi-Fi and IoT found that it was possible to do so with less than 100 USD (Vikram, Harish et al., 2017). The findings of this study show that the ability to utilize Wi-Fi in unpredicted ways is possible and that its ability to allow the communication of different devices on a network can be done so affordably. The study used a smartphone as the base hub for controlling wireless sensors on the network, this relates to the proposed project as an Android smartphone will be used to receive Wi-Fi signals for positioning. The study does fail to mention potential downsides to the open communications of Wi-Fi such as the susceptibility to interference. Wi-Fi signals produced by routers and other transmitters are still a radio technology and can be affected by interference such as multiple wireless networks that use the same frequency near one another. An example of this could be a multiple office building with separate networks. The more populated an area is by objects can also affect the reliability of Wi-Fi signals, especially when metal objects are close to the transmitter (Wiegand, n.d.). This interference is known and attempted to be compensated for when installing wireless networks by having the access points higher and not obstructed. Bluetooth also uses radio waves and experiences the same possibility for interference meaning Wi-Fi is not alone when considering this downside. The examination of Wi-Fi and its potential shows that the increases in the standards have allowed further applications of Wi-Fi to exist which gives relevance for this project. While interference can occur, this is true with any radio technology and common place abilities to limit this are well known.

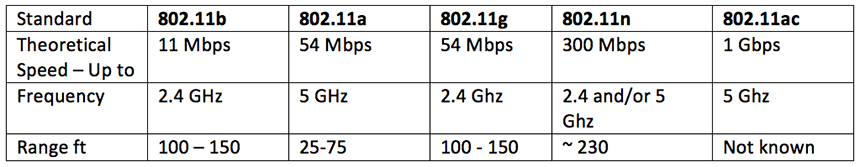


Figure - Comparison table of capabilities of some of the most recent amendments to the 802.11 standard for Wi-Fi (Farnell element14, 2017).

### 2.2.2. RSSI fingerprinting

Received Signal Strength Indicator (RSSI) is often used in localization technology to “estimate the distance between the transmitting and receiving devices” (Zhang, Yuan et al., 2015), this distance could be used to determine a location but often alone is not enough due to the ability to receive signals at different angles from the transmitting device. RSSI can relate to any type of signal strength including Wi-Fi or Bluetooth. The introduction of fingerprinting makes the location determining more possible. Typically, location fingerprinting has two stages, an offline and online stage (Alikhani, Amirinanloo et al., 2017). In the offline stage the RSSI and location of known reference points are stored in a database (Alikhani, Amirinanloo et al., 2017). In the online stage, pattern matching algorithms are used with the fingerprints and the user’s data to determine a location (Alikhani, Amirinanloo et al., 2017). RSSI fingerprinting has been used with Wi-Fi signals and GPS before in a 2008 research paper to implement an outdoor localization system (Li, Quader et al., 2008). The paper aimed to increase accuracy in urban areas, as GPS is less effective when blocked by buildings by using Wi-Fi from surrounding buildings to narrow down the GPS position. Initially the accuracy was poor but upon refinement of the fingerprinting algorithm it was able to accurately position within 35 meters and had potential for further investigation (Li, Quader et al., 2008). While it is useful in obtaining accurate results, RSSI fingerprinting does come with setbacks. The possible requirement for different algorithms to be run for more accurate results makes the RSSI fingerprinting resource heavy when running on older devices (Xia, Liu et al., 2017). This also means that RSSI fingerprinting may be slower on devices with less computation power which is an aspect that this project should consider when choosing how to determine a localization method. The potential for a higher accuracy location position still makes this a beneficial method of producing localization.

### 2.2.3. Trilateration

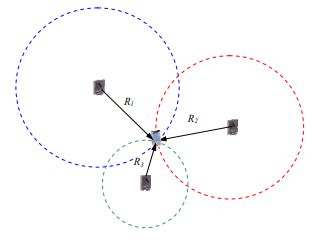


Figure - Trilateration Diagram

Compared to fingerprinting, trilateration is a simpler approach to determining position. Trilateration is performed by having a minimum of three points that have known positioning, with the distance to each being calculated, the distance is used to determine radius circles around the three points and where they meet is the determined location (Li, Quader et al., 2008). A clear example of this can be seen in figure 7 (Li, Quader et al., 2008). In instances where the signals received are of signal strength rather than an exact distance the distance will have to be calculated to be triangulated (Li, Quader et al., 2008). Trilateration is the geometrical approach to positioning that requires less computation than RSSI fingerprinting as there is no need to acquire a database of fingerprints beforehand to refer to, instead the position can be determined from the position of the signal points. A 2016 conference proceeding explored the use of trilateration in an ultrasound positioning system and found that the method provided good basis for future research as the static and dynamic position testing had less errors (Nguyen, Huynh,2016). The study shows the potential for trilateration to be used in positioning systems, especially with the benefit of less set-up. On the other hand, trilateration requires a minimum of three signal points meaning it is not possible to use this method in implementations with less points. While this is restrictive it can be noted that most systems so far work with implementations that have a basis of multiple signal points to get an accurate position, so this downfall of trilateration is less noticeable. This point is also true in that larger areas where a positioning system is needed will have the need for more signal points or in the case of this project, more router points to cover the entire area. Keeping these considerations in mind, trilateration is still a more viable option for this project due to the simplicity and less computing power it needs but the choice between RSSI fingerprinting and trilateration may be more blurred as both may be used for a potentially more accurate outcome.

## 2.3. Investigate the Android Platform

Trends in Android development were mentioned in the project background, but it is beneficial to examine them in more detail with emphasis on WiFi signals implementations and interactions. This will help the development of this project and give insight into the ability for WiFi integration on an android device.

### 2.3.1. WiFi on Android

Mobile devices have a large dependency on WiFi as it is still one of the primary ways they receive internet even with new standards such as 5G being integrated (Fogg, 2018). The integration of WiFi to access the internet in mobile phones has become ubiquitous in our everyday life. With 85.9% of “global mobile OS market share, in terms of sale to ends users” being to Android (Statista,2018). The plausibility for an android and WiFi based positioning system has the potential for a wide audience using interactions of android and WiFi capabilities, especially in positioning, is to be investigated. The design and implementation of a WiFi based food industry application is outlined in a Philippines study (Banacia, Dindo Fernando et al.,2014). The paper outlines the creation of a WiFi based android application where it states the inclusion of WiFi allowed the application to be created which may have required significant infrastructure otherwise and allowed customers to interact with the system. This paper is useful as it highlights that an existing WiFi infrastructure can allow other opportunities to be integrated which gives merit to an indoor positioning system. A 2015 investigation into WiFi and other positioning systems (Chouchang Yang, Huai-rong Shao, 2015) explored the ability of access points to be able to provide distances from received signal strength and the cost for indoor positioning systems based on WiFi. It found that installed WiFi infrastructures could potentially be used to implement an indoor positioning system at a low cost due to the minimal effort required to implement the system compared to other positioning system that require additional equipment, which the findings from previous sections. The paper also explores formulas used to convert RSSI to distances through algorithms like free path loss or time shift property. The use of these formulas may be important for the development of this application to obtain distances from access points.

Understanding previous attempts at android applications using Wi-Fi and their benefits and issues is important to this project as it is necessary to take these into consideration for the development of its own artefact that will be used for testing the research question.

### 2.3.2. Android Development

Higher demand for secure applications on Android platforms is a current industry trend (KHARCHENKO, 2017). It is recommended that developers now spend more time when testing their applications because security can “play a vital role in the success/failure of an app” (Inukollu, Keshamoni et al., 2014). With this trend growing, it has led its way to numerous processes and best practises being used to satisfy customers’ wants as well as limiting the number of applications with vulnerabilities. The Android developer documentations have grown to include best practice guides on security, testing and other areas (Google, 2018). WiFi usage by applications is becoming more regulated on the system with permissions having to be permitted by the user to allow an application to use the communication technology (Google,2018). WiFi Security is also linked into other trends in Android Development such as location services, as previously mentioned. Emphasis has been on keeping data such as geo-location or router history contained and to always inform users of any tracking of data, even if it is not stored and instead used in live processes (Future of Privacy Form, Centre for Democracy & Technology, 2011). The potential for security to be compromised by improper use of data means that care will be required when producing the application for this project. The substantial number of Java users on Android will help support a secure development as the resources are more abundant with Java being the most popular language since 2013 according to the TIOBE index (TIOBE, 2018).

Examining the trends in Android development has shed light on the necessity for security especially given its links into other trends such as positioning technology which is the main aspect of this project. Investigations have also shown that Java is still the main development language on Android.

# 3. Execution

This section will provide an account and justification for development of the artefact as well as the chosen development styles, lifecycle and methods as identified by the literature review.

## 3.1. Lifecycle

The lifecycle of the development was based upon a Waterfall model that included an iterative implementation stage. It is understood that most software-based projects use an Agile to development with 80% of USA federal government projects being described as Agile in 2017 (Viechnicki and Kelkar, 2017). This was decided to be the wrong approach for this project due to the reliance of Agile to be applied to teams (Choudhary, Rakesh 2016) and group developments whereas this is not the case for this project. The project also contains many write-ups which does not lend well to Agile development due to its large focus on incremental delivery. An iterative Waterfall model, shown in figure 8, was chosen for its ability to allow a clear planning stage and methodical stages like traditional Waterfall models but has added benefits of revisiting previous stages to allow for constant iterative and incremental implementation where the system is built upon (Mihai Liviu 2014).

This allowed the development to follow the steps of investigation and analysis of the project where requirements where determined. Following this, an iterative development of design, implement and testing occurred where systems are incrementally expanded. This ensured the system is robust before adding to it and limiting the bugs in the final stage - deployment. In this project, deployment will be the final project testing and drawing of conclusions. Each iteration built upon the previous and improved it before reaching the final artefact.

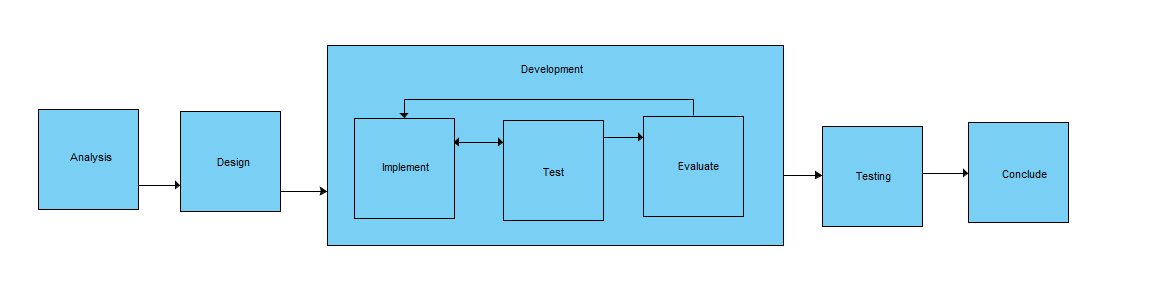


Figure - Lifecycle Diagram

## 3.2. Analysis and Design

The initial stage was to analyse the system and using knowledge from the literature review, to determine the problems that may be faced during execution and how to achieve the artefact of an android application that can position using WiFi.

The initial concerns for the project were outlined in a brainstorming document seen in appendix 8. This highlighted routes for development and was crucial to the project as it allowed surface exploration of development. This identification also aided the requirements specification outlining. Investigations in the literature review section also helped with these identified problems especially the WiFi investigations which narrowed down an algorithm to use. It is clear from the online support that android studio is the most supported platform for Android development. The features and ability to use emulators helped make it most powerful development environment. This also helped decide the development language as Java for the same reasons, it was the most supported.

Analysis of the requirement specifications as seen in appendix 1 helped give a clear definition of the applications functionality and helped to create goals to work towards in each iteration of the implementation. The requirements gathered were not all achievable as the requirement of having updates of the user’s location was not implemented due to google changing the premised number of possible WiFi scans for security and battery.

The design stage of the application was built upon from the analysis stage with large focus on functionality as this was the main aspect that would be used to test against the research question. There was however, scope for having a clear UI for the application due to the need to have a user’s position displayed upon a map and initial wireframes were implemented as seen in figure 9. The ability to visualize what the application may look like also helped to provide flesh to the functionality as it could be looked at how this could be achieved. The design did not vary greatly from refined wireframes as seen in appendix 2 and the final UI as seen in appendix 3. This clear design was important to the project as it helped to keep development time down in later stages. Design diagrams were also produced for the same reasons and can be seen in appendix 2 with the production of diagrams such as a use case or sequence diagrams providing an insight into the project that helped ensure the focus was clear as limited resources for the project meant that slippage in time or man power was not possible.



Figure - Initial wireframes

## 3.3. Implementation

This section will give further justification to implementation methods and choices as well as explain requirement elements such as the positioning algorithm and the front-end map display.

### 3.3.1. Lifecycle Iterations

The project’s lifecycle iterations were broken down so development could be gradual and would build upon the working application without worrying about tackling every requirement from the start.

The initial iterations of development regarded getting basic functionality working small scale. It would have been difficult to attempt to implement, test and evaluate the application during the initial stages in the final test building of the university so a smaller environment as seen in appendix 4 was set up. This consisted of a five room flat with three routers positioned within. This allowed first iterations to attempt to be able to determine a user’s position by testing algorithms and using a basic 2D graphic for simplicity.

Through each iteration, functionality was built upon and a refined algorithm was determined with the testing finally moved to the testing building.

### 3.3.2. Indoor Map Display



Figure - MapWize application example

The UI of the map was implemented to fulfil requirements that allow a user to see their current position within the building. This was an important aspect of determining the feasibility of the technology since the inability to translate a received position onto a map would affect the potential of the technology. The implementation of an indoor map was originally implemented as a 2D graphic during the first iterations of the project as shown in appendix 4. This served the initial iterations purpose of getting basic functionality working but it soon became clear that moving to a larger environment would be difficult if graphics had to be produced for each floor.

It was determined the solution to implement a map for an indoor positioning that allows simplistic transference to larger spaces would be with a framework. The framework which would allow this was MapWize. Looking into MapWize, it was an obvious choice for the project as it was a dedicated indoor mapping platform (Mapwize.io, 2019). MapWize provides the ability to set up blueprints and overlays as indoor maps called venues which can be shown when a user zooms in close enough to them. An example of a MapWize application is shown in figure 10. An individual creates an account and configures an indoor location known as a venue through the web-application then can imported with the MapWize framework which then allows a normal mapview to use the MapWize style. The way this is done is very simple, as shown in the following line of code:

**mapView**.setStyleUrl(**"http://outdoor.mapwize.io/styles/mapwize/style.json?key="** + AccountManager.*getInstance*().getApiKey());

The framework uses API keys to import your venue which can then be used to place it on the map. For this project a venue was created after the initial iterations which was of the same five room flat allowing continued testing within the area to ensure that the application worked the same way. Once this was deemed acceptable the final venue was modelled with the software by importing blueprints of the building for each of the floors that would be tested and mapping out the routers for each floor.

### 3.3.3. Algorithm Implementation

The implementation of the algorithm was the largest focus during the implementation stage and required copious amounts of time to ensure the accurate results were being implemented so as not to affect the potential outcome. The findings of the literature review helped to provide the possible algorithm that would suit the project more which was trilateration. This was chosen over RSSI fingerprinting as fingerprinting could require machine learning to ensure it was implemented accurately which would be difficult on a mobile and was out of scope to have a server process it. Trilateration was also found to be the simplest method which was in line with the project as the alternative of WiFi positioning was to include the potential to look at how easy it could be implemented and to be run on a host of devices.

#### Router Set Up

The choice of trilateration meant that before implementation could be done for the application, gathering of router points Latitude and Longitude positions within the building had to be gathered as well as the MAC address which would act as an identifier. This was done for the test building and was done small scale during the initial iterations for the smaller environment to simulate how it would work in the test building. The application would set up a data structure containing a custom RouterMap class object that had the position and MAC stored in it. This was useful to ensure that access to the data did not have to be written or read from a database and was stored in memory, which while can be resource heavy, was not measured in this project. The collection of known router points was used to ensure that any other detected routers were ignored and once an area was scanned, the nearby routers could be determined by their MAC address then their coordinates received. This was necessary as trilateration required known points to be able to position a user against them.

### WiFi Scan and Distance calculation

Implementation of a system that could scan WiFi signals near the device and be able to calculate a distance with them was required to be able to get the data received to the mobile device for trilateration. The original approach to having the WiFi scan was to implement threads and a timer to schedule the device to initiate a scan everyone half second or every second but this was not possible to do due to the scan method. The method for scanning was to use the WiFi manager class provided in android which is the primary API for obtaining and dealing with WiFi. The WiFi Manager provides a method called StartScan which can be seen in the code snippet below.

registerReceiver(**wifiReceiver**, **new** IntentFilter(WifiManager.***SCAN\_RESULTS\_AVAILABLE\_ACTION***));  
**wifiManager**.startScan(); *//Start scan is deprecated and will be eventually removed. Should be changed if application is to continue.*

The WiFi Managers StartScan method is setup by registering a receive method that will be called once the scan completed asynchronously. The method instructs the device to scan for access points. This is seen as the primary way to obtain surrounding access points on android, so it was the method chosen due to newer introduced methods such as Round-Trip Time scanning (RTT) - where a ping is sent to points around and the ping back is recorded with the data - are still being refined. RTT was considered for the project as it can also provide a distance to the point without external calculation and can be seen as accurate even while still being refined RTT requires specific routers to work which would have limited the project’s available testing areas and would not be a representation of being able to install the solution in everyday life. The choice for the older method of StartScan did not come without difficulties as the introduction of RTT has deprecated StartScan and has imposed restrictions on the number of scans allowed by the system to four scans every two minutes. This would be too slow to feasibly implement application updates of the users position as the time spaces between positions would be inaccurate as they have already occurred. This trade off was necessary to ensure that the application could still position as intended.

Once the application received a collection of access points, they were then filtered to ensure the ones that would have the distance calculated was part of the premised router points of the test environment as previously setup. The code for calculating distance, which can be seen below, was based on the free pass loss (FPL) formula, a formula used to predict the loss of signal strength across free open space (Wolff, 2019) . The formula is used as a key stone in determining distance and can be changed to account for more dense areas as the formula in figure 11 shows.



Figure - Projects distance formula

**public double** calculateDistance(**double** signalLevelInDb, **double** freqInMHz) {  
 **double** exp = (27.55 - (20 \* Math.*log10*(freqInMHz)) + Math.*abs*(signalLevelInDb)) / 20.0;  
 **return** (Math.*pow*(10.0, exp)) / 100000;  
}

The unreliability of WiFi signals in populated areas was seen during the literature review stage and so the original FPL formula was altered and would help to account for walls and signal interference within the test area but as the formula was originally developed for open space, inaccuracies would still be likely to occur, and the changing environment of different walls and objects would not be accounted in the formula which affects the final accuracy of the application.

### Data Bulking and Trilateration

The trilateration method used was 3D n-point trilateration which would attempt to position the user within a 3D space and rather than the traditional method as seen in the literature review, could use a variable number of points (n).

The trilateration could be achieved with a minimum of two points, however this could be unpredictable and in earlier tests would result in several iterations of the method having to be performed severely affecting the time of which a user’s position could be relayed back to them. This was due to trilateration’s optimum points for 2D being three and above and four or more for 3D. With the hanging of the application in areas of the test building due to the lower density of the router, it was determined that data bulking had to occur to achieve the minimum optimums. The application implemented a method that would check the number of routers, if it was less than four then it would attempt to bulk up the data by adding new points which had their positions and distances determined by the average of the previous results. This aspect was the least impactful on the application as it minimally influenced the results rather than fabricating points from random data. While this method was not ideal for the accuracy, it was deemed acceptable as the areas with less than four router connections available was small across the entire test base as seen in appendix 9.

The complexity of 3D trilateration comes from the additional plane which turns from overlapping circles and becomes spheres with more computation. Attempting to formulate this within Java would be difficult and require understandings of complicated mathematics, for this reason it was logical for to instead use a mathematical library instead. This library was an MIT licenced library built upon the Apache Commons Math library and implemented a non-liner least squares method for trilateration. Least squares approach is where a line of best fit is determined by using squares regression (Kenton, 2019). The non-linear version used applies to a version of the algorithm that is applied without known parameters, in this case, the known position.

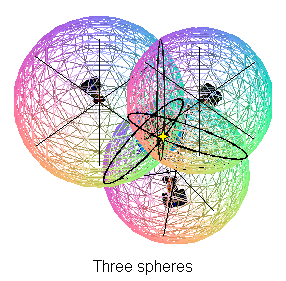


Figure - Graphic representation of 3D trilateration

**try**{ *//to ensure program does not break  
  
 //Solver which solves trilateration* NonLinearLeastSquaresSolver solver = **new** NonLinearLeastSquaresSolver(**new** TrilaterationFunction(positions, distances), **new** LevenbergMarquardtOptimizer());  
 LeastSquaresOptimizer.Optimum optimum = solver.solve();  
 centroid = optimum.getPoint().toArray();  
}

The above code snippet is the implementation of the trilateration function provided by the library that takes in the positions and distances. The library allowed for the cut of much larger development time that would be required to implement a custom version of 3D trilateration and ensured that the project was successful. The trilateration function would take in the distances and positions then determine the best fit for the position which would be returned. This variable would become the coordinates for a user’s position.

The trilateration of a user’s longitude and latitude remained unchanged after being produced by the algorithm but due to the floor position having to be a whole number, an additional step was required to determine a user’s floor. A method was created, ConvertToWholeFloor, which was used to convert a floating-point number to be used as a floor and was only called if this was the case. The method consisted of If-Else statements the classified the floor into three potential brackets as this was the number of floors for the testing environment. The cut off numbers for the brackets to classify the floors were determined by looking at many different trilateration outputs on different floors and in different areas to determine where a fitting bracket for each floor was. Once the number was converted, a user’s position was fully formed and would be displayed back to them.

## 3.4. Application Testing

This section will layout the testing of the application for bugs and to ensure functionality and how this was achieved within the iterations.

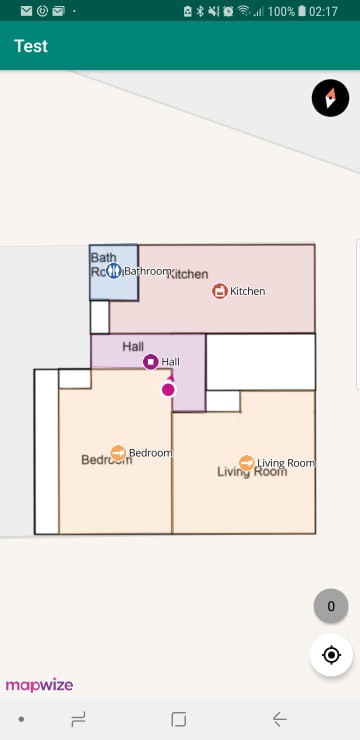


Figure - Initial testing environment

The initial testing for the application, as outlined, was in an environment smaller than the final testing environment to ensure that it was controlled and allowed for a much quicker development. The initial environment was a five-room flat. Figure 13 shows a screenshot of an early iteration where testing for functionality occurred within the smaller environment. This was an important phase as all router points could be reached and the application launched quickly to be tested and troubleshooted without problems. The approach taken to testing the application within this area was to determine the correct response or output that would be required from the application, with the application then being troubleshooted if it was not correct. This informal testing was commonplace for this project as the final artefact that would be produced would not be for users and instead to test the research question, so usability bugs were not covered in the scope unlike functionality bugs. The transference to the larger testing area used the same approach to test the application for the correct output with positions and rooms known to the developer with the output being tested for what is expected.

# 4. Evaluation

This section will detail testing the application against the research question as well as the findings from the results. It will also include an evaluation of the system which will be used with results to examine the findings against both the hypothesis and research question. The key issues that this section hopes to conclude upon are:

* Is an indoor positioning system based on received WiFi signals by an android mobile device a feasible technology?
* When compared to other forms of indoor positioning systems, does WiFi provide a comparable experience?

They two key elements come from both the research question of this project and the hypothesis suggested in the introduction. The hypothesis highlighted the potential that WiFi may be a feasible indoor positioning system but only to an extent and less so than other forms of indoor positioning systems than those of the literature review.

## 4.1. Research Testing Methodology

The testing of the application against the research question originally envisioned user testing in the planning of the project as it was intended for the project to include scope for how the application was suitable for users, ethical approval was even granted for this form of testing and can be seen in appendix 10 for an in-depth overview of this. This approach has been refactored as it was determined that the application’s usability was not in the scope of the project as the project was to determine the feasibility of WiFi as an indoor position technology with the application only assisting. The testing took approach of having a single tester use the application within the test environment to test known locations against what the application predicted with the data being analysed and compared.

The final testing location was a university building which was chosen for its realistic setting the technology could be used in along with existing router infrastructure. The variance in wall types, people and objects helped to give the testing area a robustness to full test the project. It was decided that the top three floors would be the full testing area as it gave enough floors to test the application.



Figure - Screenshot of testing

The testing positions were determined beforehand with each floor having 8 testing positions, as shown in figure 14, spread across them with conscious effort to ensure that all parts of each floor would be determined. The XY position on each floor for these points was recorded and testing consisted of the tester going to each and using the application to note the position, floor and room the application positioned the user in. The tester also took note of how accurate the floor guess was and how many attempts it took for the application to provide a guess, to look at reliability.

## 4.2. Results Examination

The results from the testing of the application can be examined in various angles to get an overview of how feasible a system is including the results of accuracy for XYZ positions, reliability and ability to position a user within a room. All the results will be examined to get the best conclusion on the ability of the system. The implementation of an application that can provide results to be examined has been an advantage to the project as it has allowed an exploration of tangible results which will be able to give justification for the outcomes of this report and will be able to compare the research question and key issues.

### 4.2.1. XYZ Accuracy

|  |  |
| --- | --- |
| **Error Margin Area** | **Average Result** |
| 5th Floor | 12.35 meters |
| 6th Floor | 6.40 meters |
| 7th Floor | 12.90 Meters |
| Overall | 10.55 Meters |

Table - Error margin by floor

The XYZ accuracy relates to the positioning of a user on a floor (Z) and their physical location (X, Y) which has a large amount of weight on the feasibility of such a system as a large part of any positioning system is the ability to provide an accurate position. The results from the XY accuracy testing are listed in table 1 which is a refined view of the results from appendix 7.

Table 1 shows the average error margin in the actual and guessed XY positions for each floor and overall. The results show at that the system was able to position a user within an average of 10.55 meters of their actual point over all tests. In consideration this is good for the application’s accuracy as it provides a near similar accuracy to GPS which is considered between 5-10 meters of accuracy. The issue is more apparent in further consideration as GPS is dealing with much larger outdoor spaces, the inaccuracy of 10.55 meters indoors can mean a user’s room position is inaccurate. Breaking down the accuracy onto floors it can be found that the worst floor for positioning was the 7th floor. It had the largest error margin of 12.90 meters. The best floor by almost half of the other results is the 6th floor which had the smallest error margin of 6.40 meters. The differences in floors varies which was unexpected in the project but has a pattern in that the two edge case floors have a considerable worse accuracy than the floor in the middle.

|  |  |
| --- | --- |
|  |  |
| Total Floor Guesses | 24 |
| Correct Guesses | 17 |
| Wrong Guesses | 7 |
| % Accuracy | 70.83% |

Table - Floor accuracy by floor

Figure 15 and table 2 relate to the accuracy of the application to guess an individual’s floor (Z) correctly. The table shows an overall accuracy of 70.83% for the system to guess floors correctly. This result is quite accurate and more so than was expected of the application as it is correctly guessing the floor over half the number of times. Breaking this down again shows that the 7th floor had the worst results with only 4 out of 8 tests (50%) giving a correct floor guess and that the 6th floor once again had the best results with 7 out of 8 (87.5%) having a correct guess.

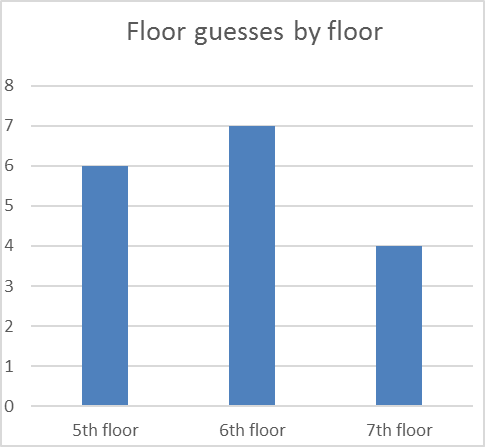


Figure - Graph showing floor guesses by floor

The floor and XY positions have a tread in both sets of results with the 7th floor being the worst for positioning and the 6th floor being the best. The results help show that both edge cases of 5th and 7th floor preformed worse than the 6th floor. It is possible to have a thesis of why this occurred; it is possible that the 6th floor had the benefit of having access to both the floor above and below’s router points to help give a more accurate result with back up to this being that the more positions for trilateration to use the more accurate it can be. For the 7th and 5th floor, they would be limited in their ability to receive router points from other floors as the loss path on the WiFi signals occurs quickly when passing through barriers such as floors and ceilings. This considered with the accuracy data could potentially mean that on a larger scale the accuracy of the application would improve as there would be more router points for the application to use. While the accuracy in this setting is only within 10.55 meters (XY) this could fall even lower giving much more feasibility to the technology.

### 4.2.2. Room positioning and reliability

The reliability of the application was also looked at while testing to comment on how the system could be used in a real-life situation to ensure that even if the system was accurate, it was feasible to be used to gain those accurate results. The initial plan for testing of this included more in-depth aspects where the application constantly updated, and reliability could check on the update of positions but due to the limitations imposed on the scanning frequency, it was not possible to do this. As it was believed this was still in scope of the project, the testing of reliability was changed to take on a smaller role where the number of times it took to position a user in a location was recorded as WiFi router coverage does not cover the entirety of every floor as seen in the router heat map in figure 16. The reliability was continued as it would help determine what the technology could produce if the situation was less than optimum as it may be in potential uses for it. The results of the testing show that all the tests were able to produce a result to the user on the first attempt (accurate or not). This is positive for the application and the project overall as it means the technology is very feasible for the reliability factor and can be relied upon in real world uses to give users results when they require it which would be essential. The downside of the results produced is that they still do not include factors of users moving with the results updating as all testing was done standing still in a location due to the scan limitations. This would have to be explored in further depth with an alternative way of updating a user’s location to give a better idea of the reliability on a moving device as the current testing and its low weight result has shown a limitation of the implemented solution.

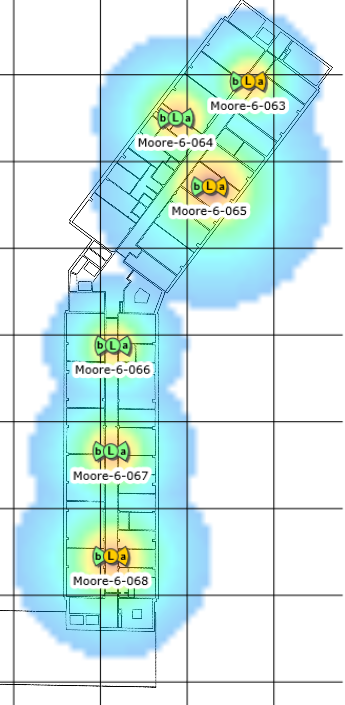


Figure - Heatmap of floor router coverage

|  |  |
| --- | --- |
| **Location** | **Accuracy Percentage** |
| 5th Floor | 50% |
| 6th Floor | 37.5% |
| 7th Floor | 12.5% |
| Overall | 33.33% |

Table - room accuracy by floor

Room positioning was also tested for the application to help supplement the findings of accuracy. Whilst this testing and its results is directly related to the specific testing environment, it is still useful to the project to be able to give tangible accuracy to the 10.55-meter difference. The actual room that the tester was in was compared with the room that the application predicted it was in. The recorded result was either correct or incorrect as another recording of the data would not translate to a real-world implementation. Table 3 shows the results for the predicted room per floor and for overall. The results overall show that accuracy for the technology in terms of correct room positioning is low. 33.33% makes up the overall room accuracy which is an issue for the potential of the technology since it relies heavily on not only accurate positioning but the ability to position a user within an area correctly. While the 10.55 meters difference has the same accuracy as GPS and is low, when compared with the ability to position a user in a room, it shows that this difference is large for the space that was tested. This will be due to larger open spaces such as outside, the difference is minimal where as in a building this is a clear difference. As stated before, there is scope for the error margin to be reduced with further testing in a larger testing zone with routers, but room accuracy may remain the same due to buildings having smaller rooms.

## 4.3. Comparisons to other technologies

Feasibility of this technology not only comes from its own ability to position individuals but also the larger look at how it fairs against other technologies, which were explored in the literature review, and the comparisons that can be drawn. This is an important compassion for the project as technologies that have better accuracy may be a better choice making the feasibility of WiFi less so than determined.

The direct competitor to WiFi positioning would be the newly popular Bluetooth Low Energy beacons which, when explored in the literature, was found to be able to produce accurate results to 1 meter (Huh, Seo, 2017). This was further investigated and found that results could be inaccurate by over 2 meters in other situations (Herrera Vargas, 2016). The variance in accuracies in the Herrera and Vargas study found that the density of beacons available affected the results that were produced. This was a similar finding of this project where the 6th floor that had access to more router points was able to provide a much more accurate result. While the results of beacons are much more accurate even in less dense areas, further investigation to compare the results would have to be done to compare accuracy per density but increasing router density could potentially close the difference between BLE and WiFi. Cost should also be considered as it would be unfeasible to use the technology in real settings if it cost too much compared to other options. Even with BLE beacons becoming as affordable as router points, they are still an additional cost on top of routers as well as an additional labour to install whereas an area would already possess WiFi routers. The factor of additional cost and that accuracy in both systems is dependent on density of receivers, it can be said that WiFi is still a competitor to BLE beacons due to their existing infrastructure and ease of set up.

Much like the comparisons against BLE beacons, ultrasonic has impressive accuracy results in smaller rooms but as explored in the literature review, scale tests have been insufficient or lacking due to the complexity of the system and the lack of support at a scale level. It is less possible to directly compare WiFi and ultrasonic in that regard as WiFi worked in the scale tests with no composite results from ultrasonic. This matter gives much greater feasibility to WiFi as it has less complexity compared to ultrasonic and has the potential to be used at scale areas depending on density.

## 4.4. Conclusions

The project set out with the goals to investigate WiFi as a potential indoor positioning system using received router signals and a mobile device and has managed to gain results and conclusions to two of the key issues surrounding this problem question. The development of the application allowed these investigations to be more extensive and practical than just study comparisons but also came with its own limitations in the form of limiting ability to investigate reliability and that the explored method is only one of several potentially better or worse implementations. The contribution of the artefacts based on the question is that it can serve as proof of concept as a functioning indoor positioning system that works in a test area. The criticisms of such artefact are, as previously stated, the changing scan methods by android that may change the way the application has to work as well as change the method entirely.

The results found that the application could position a user on a floor fairly accurately and have an XY position on that floor accurate to within an average of 10.55 meters. While impressive and potentially able to be reduced with a larger testing area and larger density router position, issues become more apparent when room accuracy is considered due to the low 33.33% accuracy for rooms that the application was able to provide due to 10.55 meters being a larger error margin indoors. The reliability of the technology is a redeeming factor but must still be fully tested to ensure this is correct. The results conclude that WiFi is feasible to an extent but requires refinements and further investigation to ensure that error margin is reduced. Comparisons against other forms of indoor positioning systems also support this as it shares similar downsides to those investigated but has less accuracy than them. It does have distinct positives against them in that it is far simpler to set up and requires no additional infrastructure in areas already equipped with wireless capabilities. These factors considered give merit that it is feasible when compared to other solutions as it has its own weighted positives and negatives.

# 5. Conclusions and further work

The aim of this project was to investigate the potential for an indoor positioning system based on received WiFi signals from router points by an android device. The method for doing so was to build an artefact that would help test against the research question to provide results that could be examined and compared against other forms of indoor positioning technologies.

The literature review provided insight into other forms of indoor positioning systems and their attributes which was essential for drawing comparisons against the technology to determine feasibility against other forms and to set base standards for its own results. The literature review also provided bases for the direction the project should take for positioning using WiFi signals in its exploration of trilateration and RSSI fingerprinting. This was important as it also gave merit to what algorithm the application would use to be implemented. The drawback of the literature review in this area was that investigations into both RSSI fingerprinting and trilateration found that both had uses and merits with a much less clear-cut difference.

The execution of the project was a sizeable part of the project that required development of the android application through an iterative Waterfall Lifecyle with the use of a framework to implement a map system specialised for indoor positioning systems and a trilateration library used to help determine a user’s position indoors. The analysis and design of the application allowed the ability to ensure the application would focus on the relevant issues the project wished to investigate and to help speed up the implementation stage to allow more time spent on drawing conclusions from the results and comparisons. The artefact was successfully implemented and was reliable as shown in the evaluation section but was only one implementation of a possible way for the technology with many other possible implementations that may reap different results than the findings of this project. The testing of the application against the research question was done in a potentially realistic building to give a better representation of how the technology would fair deployed in a commercial setting.

The conclusions from the testing of the application against the research question support the original hypothesis in its prediction that it will be accurate only to an extent and less than other forms of indoor positioning as it had accuracy to within 10.55 meters compared to a few meters for other technologies. While it was also correct in determining that it comes with its own form of restrictions, findings also concluded that WiFi positioning has other positives that existing technologies do not, such as ease to set up and small cost due to existing infrastructure. It was also able to position users on floors fairly accurate with 70% overall. Taking a weighted view of the benefits and downsides of the results and conclusions, it can be said that the accuracy is a limiting factor for the feasibility of an indoor positioning system based on WiFi, but the benefits make it a worthwhile contender that should be further explored.

The projects investigation of WiFi as an indoor positioning system was also limited in its resources and table scale to which it could look at and examine this technology. Refinements or enhancements of the project could occur in the future and may provide notable and better results. These include that a different positioning algorithm other than trilateration may be used to position with a building such as RSSI finger printing. This would be useful in the future as the fingerprints would act as further reference points potentially allowing a much more accurate position to be calculated, or for machine learning algorithms to execute more complicated mathematics to help reduce the error margin. Further enhancements could also be looking at a method other than scanning from the WiFi manager on android to produce local access points in the effort to be able to implement a continually updating positioning system that would allow a much more in-depth and correct look at the reliability of such a system that is to be produced. The future direction for this project and its enhancements could be the possibility of investigating an overall refinement of the indoor positioning system that could then be used in commercial settings or public areas to help groups identified in the introduction such as students or tourists. WiFi based indoor positioning systems would help these individuals navigate and explore areas they are familiar with helping further integrate location services ubiquitously into everyday life.

# 6. References

BAI, Y.B., WU, S., RETSCHER, G., KEALY, A., HOLDEN, L., TOMKO, M., BORRIAK, A., HU, B., WU, H.R. and ZHANG, K., (2014). A new method for improving Wi-Fi-based indoor positioning accuracy. Journal of Location Based Services, 8(3), pp. 135-147.

BANACIA, A., DINDO FERNANDO, M., REQUILLO, A. and RUBI, N., Nov 2014Design and implementation of an Android application using WiFi-enabled devices for the food servicing industry, Nov 2014, IEEE, pp. 1-4.

BIANCA BOBESCU & MARIAN ALEXANDRU, (2015). MOBILE INDOOR POSITIONING USING WI-FI LOCALIZATION. Review of the Air Force Academy. (1), pp. 119.

BOOSTANDCO, (2017), six Wi-Fi trends that show why it’s the internet’s growth business. Available from: <https://boostandco.com/news/wifi-trends-2017/> [viewed 11/10/18].

CHAKRABORTY, A., (2000). A Distributed Architecture for Mobile,  
Location-Dependent Applications, Massachusetts Institute of Technology.

CHANDLER,N., (2012). What's the difference between RFID and NFC? March, Available from: <https://electronics.howstuffworks.com/difference-between-rfid-and-nfc.htm>. [Viewed April, 5, 2018].

CHOUCHANG YANG and HUAI-RONG SHAO, 2015. WiFi-based indoor positioning. *IEEE Communications Magazine,***53**(3), pp. 150-157.

CHOUDHARY, B. and RAKESH, S.K., (2016). An approach using agile method for software development, 2016, IEEE, pp. 155-158.

COOMBS, J., (2017). More Straight Goods on Location: Bluetooth and Location Permission Opt-in Rates. Jun 8, [viewed April 5, 2018]. Available from: <https://m.rover.io/more-straight-goods-on-location-bluetooth-and-location-permission-opt-in-rates-cd4361acf3bc>.

FARID, Z., NORDIN, R. & ISMAIL, M., (2013). Recent Advances in Wireless Indoor Localization Techniques and System. Journal of Computer Networks and Communications. 2013, pp. 1-12

FARNELL ELEMENT14, 1 November, (2017), Wi-Fi Architecture, Implementation and Applications. Available: <https://uk.farnell.com/wi-fi-architecture-implementation-and-applications> [Viewed 29/10/18].

FEDORYCHAK, V., (2018). Top 9 Mobile App Development Trends for 2018. January, 4, [viewed April, 5, 2018]. Available from: <https://lvivity.com/app-development-trends-2018>.

FOGG, I., November 2018. *The State of Wifi vs Mobile Network Experience as 5G Arrives.*OpenSignal: VP Analysis.

FURLAN, A., September 4, (2018)-last update, Android Development Tools List. Available: <http://www.businessofapps.com/guide/android-development-tools/> [Viewed 22/10/18].

FUTURE OF PRIVACY FORM and CENTER FOR DEMOCRACY & TECHNOLOGY, (2011). Best Practices for Mobile Application Developers. Future of Privacy Form. [Viewed 23/10/18].

G. DELNEVO, L. MONTI, F. VIGNOLA, P. SALOMONI&S. MIRRI. , (2018). 2018 15th IEEE annual consumer communications & networking conference (CCNC) In: Anonymous 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC) 1-6.

GARUDE, M. and HALDIKAR, N., (2014). Real Time Position Tracking System Using Google Maps API V3. International Journal of Scientific and Research Publications, 4(9).

GĄSIOREK, A., (2016). Beacons Are Vulnerable; It’s Time We Made Beacons Secure. April 11, Available from: <https://kontakt.io/blog/beacon-security/>. [Viewed April 5, 2018].

GOOGLE, (2018), App security best practices. Available: <https://developer.android.com/topic/security/best-practices> [Viewed 23/10/18].

GOYAL, V., BHATHEJA, A. and AHUJA, D., (2016). Survey Paper on Android Vs. IOS. International Journal of Latest Engineering Research and Applications, 1(3), pp. 1.

GRIMUS, M. & EBNER, M., (2015). Learning and teaching with mobile devices. International Journal of Mobile and Blended Learning. 7(2), pp. 17-32.

HE XU, YE DING, PENG LI, RUCHUAN WANG & YIZHU LI, (2017). An RFID Indoor Positioning Algorithm Based on Bayesian Probability and K-Nearest Neighbor. Sensors. 17(8), pp. 1806.

HERRERA VARGAS, M., (2016). Indoor navigation using Bluetooth Low Energy (BLE) beacons, Turku University of Applied Sciences.

HUH, J. and SEO, K., (2017). An Indoor Location-Based Control System Using Bluetooth Beacons for IoT Systems. Sensors, 17(12),

HUI LIU, DARABI, H., BANERJEE, P. & JING LIU, (2007). Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 37(6), pp. 1067-1080.

INFSOFT (2015). BLE beacon positioning system. [image] Available at: https://www.infsoft.com/blog-en/articleid/41/indoor-navigation-indoor-positioning-using-bluetooth# [viewed 5 Apr. 2018].

INUKOLLU, V.N., KESHAMONI, D.D., KANG, T. and INUKOLLU, M., (2014). Factors Influencing Quality of Mobile Apps:Role of Mobile App Development Life Cycle. International Journal of Software engineering & Applications, 5.

ISTQBEXAMCERTIFICATION.COM (2017). Incremental Life Cycle Model. [image] Available at: http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/ [Accessed 3 Apr. 2018].

Kenton, W. (2019). *How the Least Squares Statistical Method Works*. [online] Investopedia. Available at: https://www.investopedia.com/terms/l/least-squares.asp [Accessed 12 Apr. 2019].

KHARCHENKO, N., (2017). 5 Hot Android App Development Trends. February 8, Available from: <https://www.upwork.com/hiring/for-clients/5-hot-android-app-development-trends/>. [Viewed April 5, 2018].

L. RAKESTRAW, T., V. EUNNI, R. & R. KASUGANTI, R., (2012). The mobile apps industry: A case study. Journal of Business Cases and Applications. 9, pp. 11-21

LEMBERG. (2014). iBeacon: The Pros And Cons &amp; Where to Begin With Programming. Dec 09 2014, Available from: <https://blog.lemberg.co.uk/ibeacon-pros-and-cons-where-begin-programming>. [Viewed April 5, 2018].

LI, B., QUADER, I.J. and DEMPSTER, A.G., (2008). On outdoor positioning with Wi-Fi. Journal of Global Positioning Systems, 7(1), pp. 18-26.

LIKHITHA, C., ANUSHA, N.K., ANNAPURNA, R. and PROF. SAVITA B PATIL .U G STUDENT, (2018). Just Walk Out Technology Using RFID, Computer Vision and Sensor Fusion. International Journal of Innovative Research in Science, Engineering and Technology, 7(6).

Mapwize.io. (2019). *Indoor mapping & Wayfinding for Smart Buildings*. [online] Available at: https://www.mapwize.io/ [Accessed 11 Apr. 2019].

Mapwize.io. (2019). *Indoor mapping & Wayfinding for Smart Buildings*. [online] Available at: https://www.mapwize.io/ [Accessed 11 Apr. 2019].

MAUTZ, R., (2012). Indoor Positioning Technologies. ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry

MICHAEL, K., (2004). Location-Based Services: a vehicle for IT&T convergence. Location-Based Services: a vehicle for IT&T convergence.

MOHAMED, I. and PATEL, D., Apr (2015). Android vs iOS Security: A Comparative Study, Apr 2015, IEEE, pp. 725-730.

MONTAG, C., KANNEN, C., LACHMANN, B., SARIYSKA, R., DUKE, É, REUTER, M. & MARKOWETZ, A., (2015). The importance of analogue zeitgebers to reduce digital addictive tendencies in the 21st century. Addictive Behaviours Reports. 2, pp. 23-27.

MONTENBRUCK, O. and RAMOS-BOSCH, P., (2008). Precision real-time navigation of LEO satellites using global positioning system measurements. GPS Solutions, 12(3), pp. 187-198.

NHLANHLA, M., ADEYEYE-OSHIN, M. and SAKPERE, W., (2017). A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies. South African Computer Journal, 29(3).

NIANTIC (2018). Screenshot of location-based game PokemonGo. [image] Available at: https://www.pokemongo.com [viewed 5 Apr. 2018].

PARK, H., NOH, J. and CHO, S., (2016). Three-dimensional positioning system using Bluetooth low-energy beacons. International Journal of Distributed Sensor Networks, 12(10).

PERZYŃSKI, T., PIETRUSZCZAK, D. and ZIÓŁEK, G., (2018). Analysis of selected operating systems in mobile devices. AUTOBUSY – Technika, Eksploatacja, Systemy Transportowe, 19(6), pp. 649-654.

PRIYANTHA, N., CHAKRABORTY, A. and BALAKRISHNAN, H., (2000) The Cricket location-support system, Aug 1, 2000, ACM, pp. 32-43.

RAMASWAMY, S., (2016). Ads and analytics innovations for a mobile-first world. Available: <https://www.blog.google/products/ads/ads-and-analytics-innovations-for-a-mobile-first-world/> [viewed 09/10/18].

RONDÓN, R., GIDLUND, M. & LANDERNÄS, K., (2017). Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications. International Journal of Wireless Information Networks. 24(3), pp. 278-290.

RUIZ-GARCIA, L. and LUNADEI, L., (2011). The role of RFID in agriculture: Applications, limitations and challenges. Computers and Electronics in Agriculture, 79(1), pp. 42-50.

SCASSA, T. and SATTLER, A., (2011). Location-based services and privacy. Canadian Journal of Law & Technology, 9(1-2), pp. 99.

SMITH, A., BALAKRISHNAN, H., GORACZKO, M. and PRIYANTHA, N., (2004).Tracking Moving Devices with the Cricket Location System, Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services 2004, ACM, pp. 190-202.

STATISTA, August, (2018)-last update, Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018. Available: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/> [viewed 22/10/18].

TAMBE, S., (2015). wireless technology in networks. International Journal of Scientific and Research Publications, 5(7).

THOMAS, J.,(2014)-lastupdate,The History of WiFi. Available: <https://purple.ai/blogs/history-wifi/> [Viewed 29/10/18].

TIOBE, (2018) -last update, TIOBE Index. Available: <https://www.tiobe.com/tiobe-index/> [Viewed 25/10/18].

UNACAST, (2016). Proximity marketing  
in airports & transportation. The Proxbook Report: proximity.directory.

Viechnicki, P. and Kelkar, M. (2017). *Agile by the numbers*. [online] Deloitte Insights. Available at: https://www2.deloitte.com/insights/us/en/industry/public-sector/agile-in-government-by-the-numbers.html#endnote-5 [Accessed 11 Apr. 2019].

VIKRAM, N., HARISH, K.S., NIHAAL, M.S., UMESH, R. and KUMAR, S.A.A., Jan (2017). A Low Cost Home Automation System Using Wi-Fi Based Wireless Sensor Network Incorporating Internet of Things (IoT), Jan 2017, IEEE, pp. 174-178.

WAKE, J., (2013). Mobile, location-based games for learning. Developing, deploying and evaluating mobile game technology in education, University of Bergen.

WELLS, G., (2015). The Future of iOS Development: Evaluating the Swift Programming Language. Technical Services Quarterly, 32(2), pp. 230-232.

WIEGAND, N., (n.d.). The Effects of Metal on Wireless Routers. Available: <https://smallbusiness.chron.com/effects-metal-wireless-routers-73559.html> [Viewed 29/10/2018].

Wolff, C. (2019). *Radar Basics - Free-Space Path Loss*. [online] Radartutorial.eu. Available at: http://www.radartutorial.eu/01.basics/Free-Space%20Path%20Loss.en.html [Accessed 12 Apr. 2019].

WUKKADADA, B., NAMBIAR, R. and NAIR, A., (2015). Mobile Operating System: Analysis and Comparison of Android and iOS. International Journal of Computing and Technology, 2(7), pp. 77.

XIA, S., LIU, Y., YUAN, G., ZHU, M. and WANG, Z., (2017). Indoor Fingerprint Positioning Based on Wi-Fi: An Overview. ISPRS International Journal of Geo-Information, 6(5), pp. 135.

Y ORTAKCI, E DEMIRAL, U ATILA & I R KARAS, (2015). Indoor navigation design integrated with smart phones and rfid devices. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. II(2), pp. 223-226.

ZANDBERGEN, A. (2012). Comparison of WiFi positioning on two mobile devices. Journal of Location Based Services, 6(1), pp.35-50.

ZHANG, A., YUAN, Y., WU, Q., ZHU, S. and DENG, J., (2015). Wireless Localization Based on RSSI Fingerprint Feature Vector. International Journal of Distributed Sensor Networks, 11(11).

# 7.0 Appendixes

## Appendix 1 – Requirements Specification

### Functional Requirements

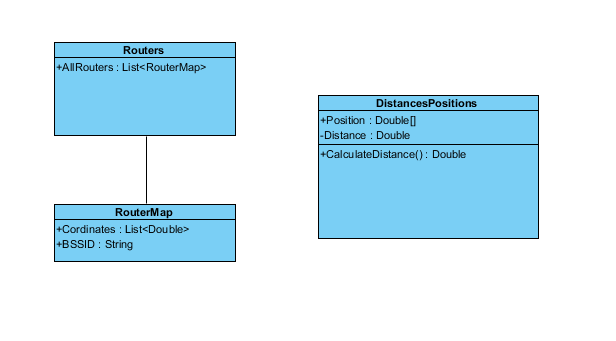
* Indoor Positioning Application Produced
  + Application will be able to position a user within doors
  + Ability for a user to see their current location within the building graphically
  + Application will automatically update the provided location for a user
  + The positioning will show both the XY coordinates and the floor of the building a user is on
  + WiFi signals must be the technology used to determine a user’s indoor position
* Map Based User Interface Produced
  + The Map view will have the layout of the building including room numbers

### Non-Functional Requirements

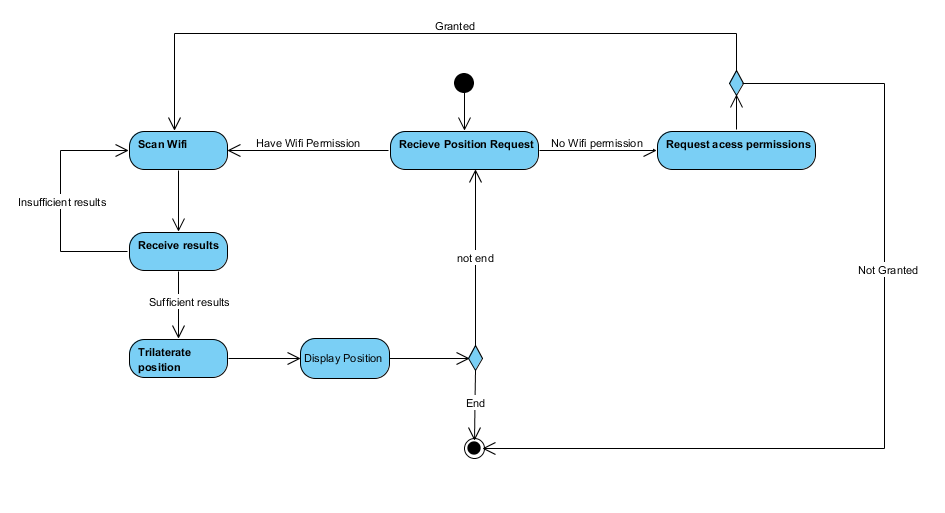
* Application must run on android mobile devices
* Application will run on an android mobile device

## Appendix 2 - Design Documents

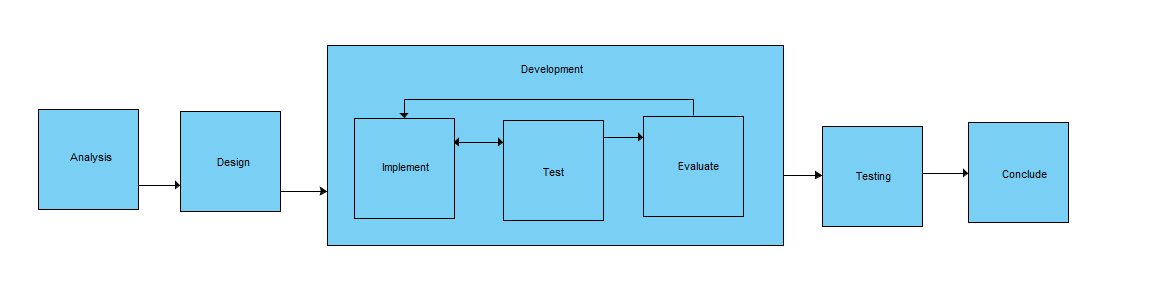
### Class Diagram



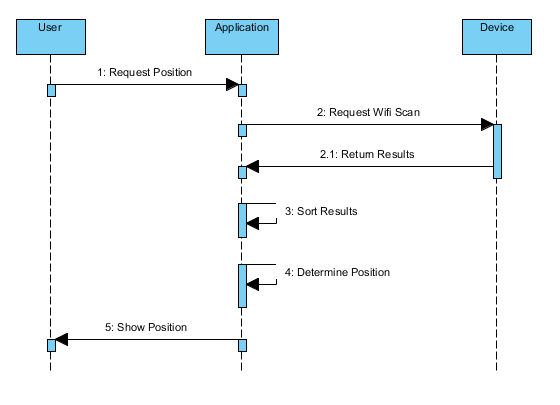
### Activity Diagram



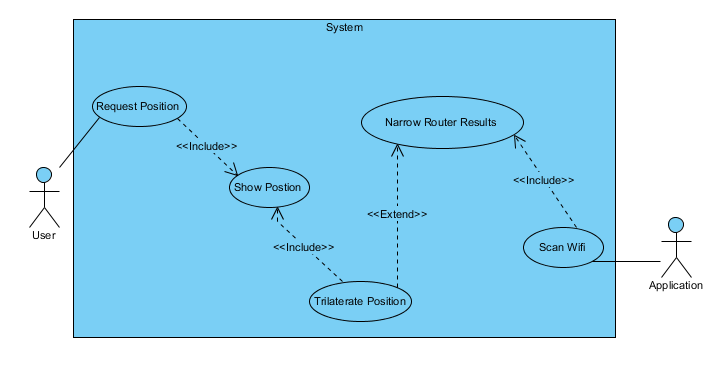
### Lifecycle Diagram



### Sequence Diagram



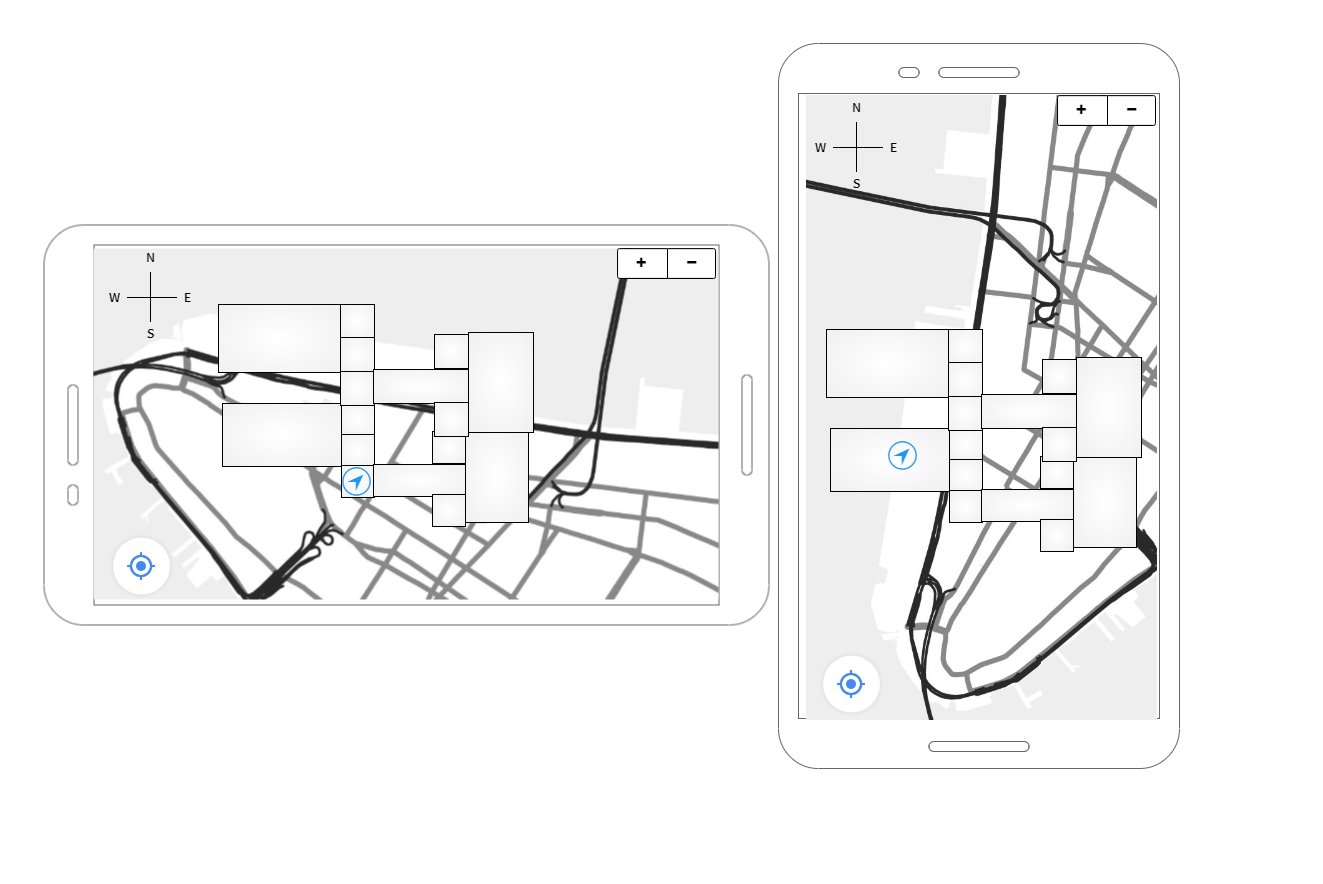
### Use Case



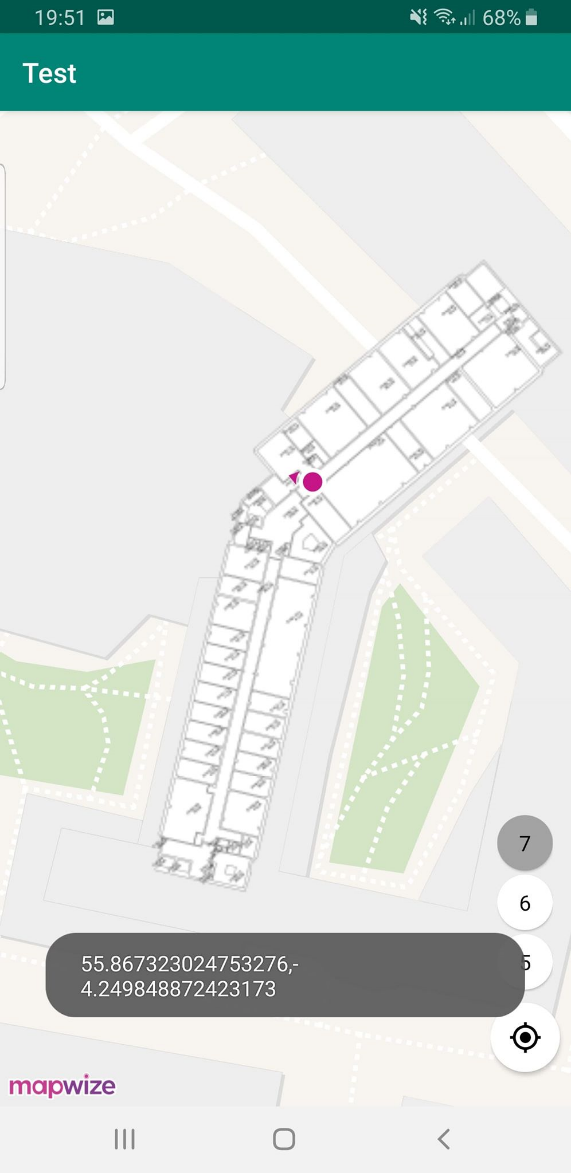
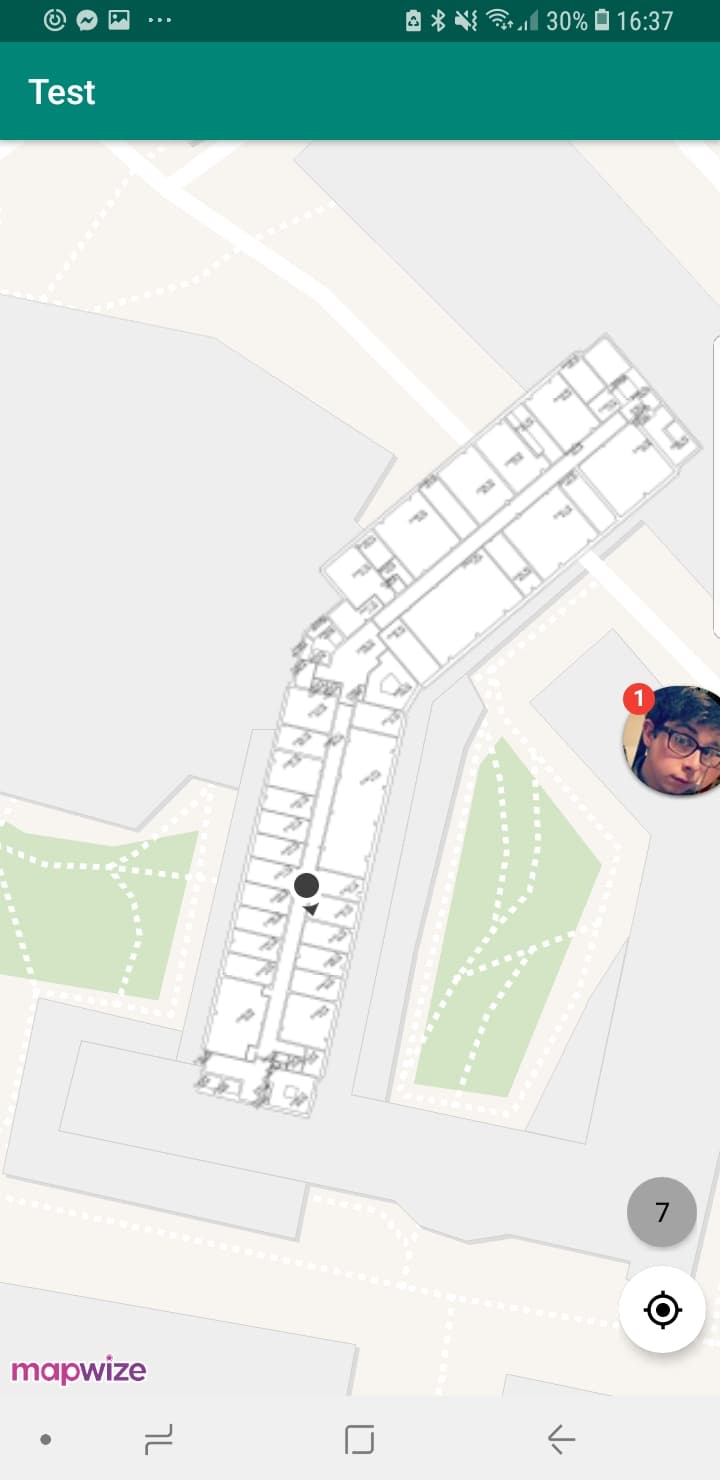
### Initial Wireframes



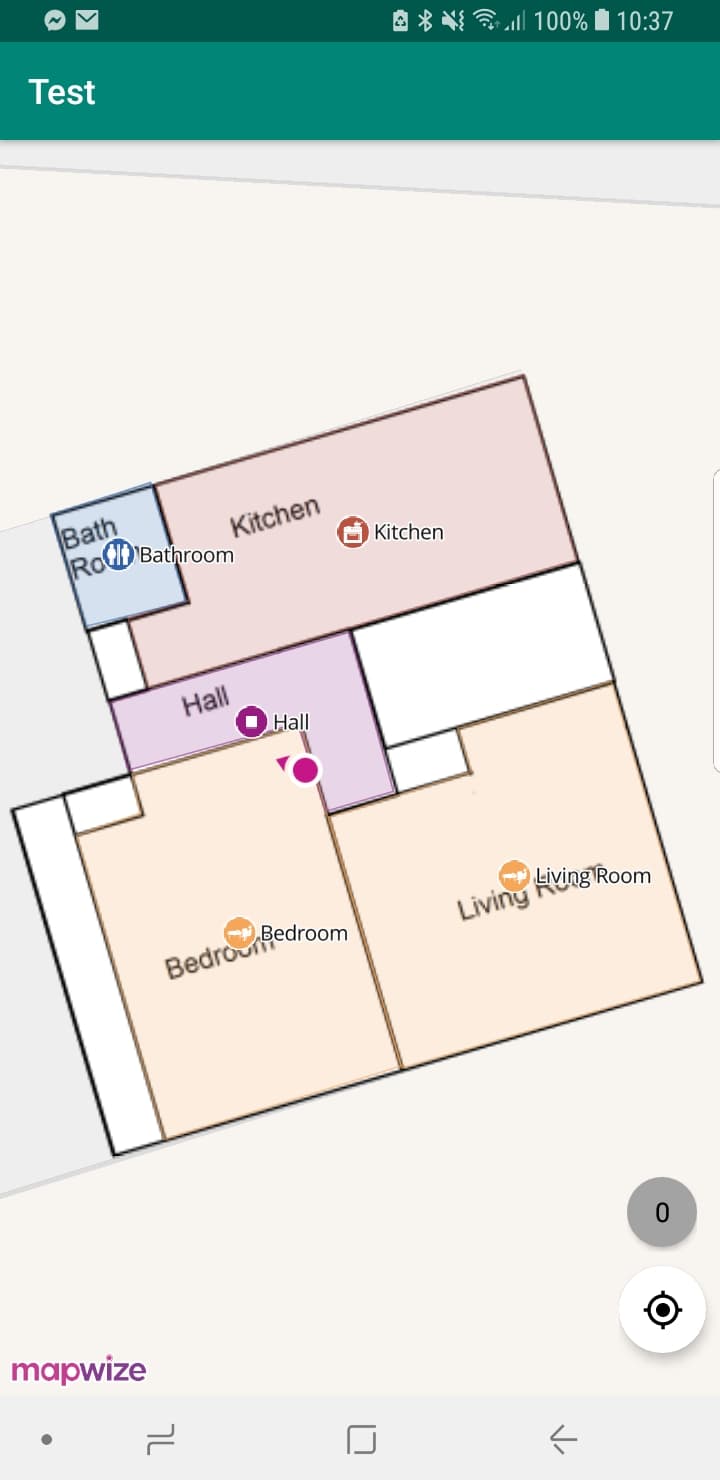
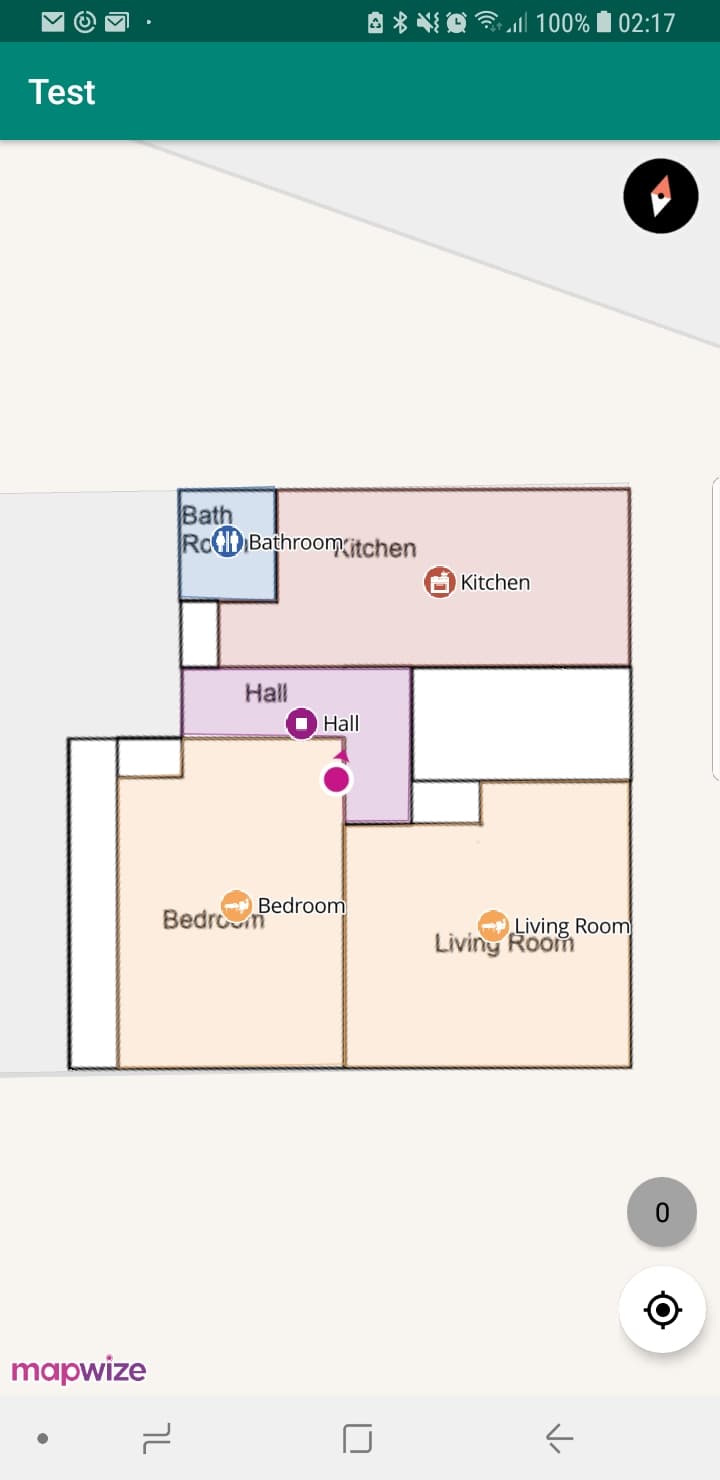
### Refined Wireframes



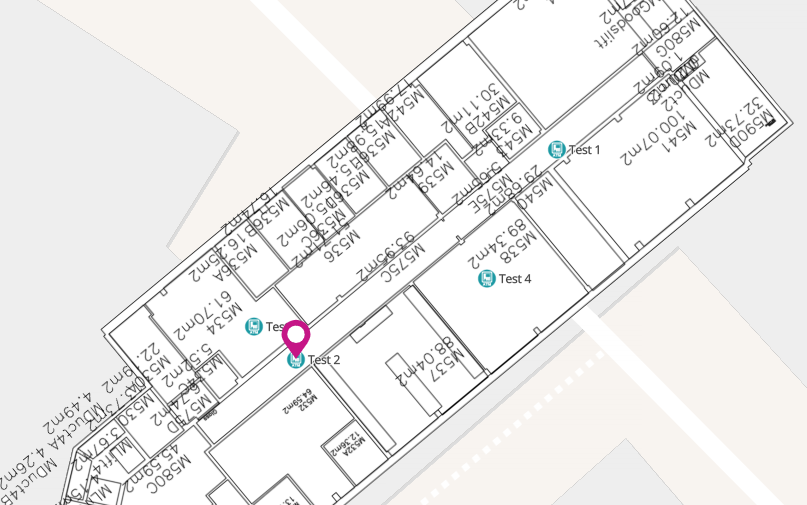
## Appendix 3 - Final UI Screens



## Appendix 4 - Initial Test Environment Screenshots

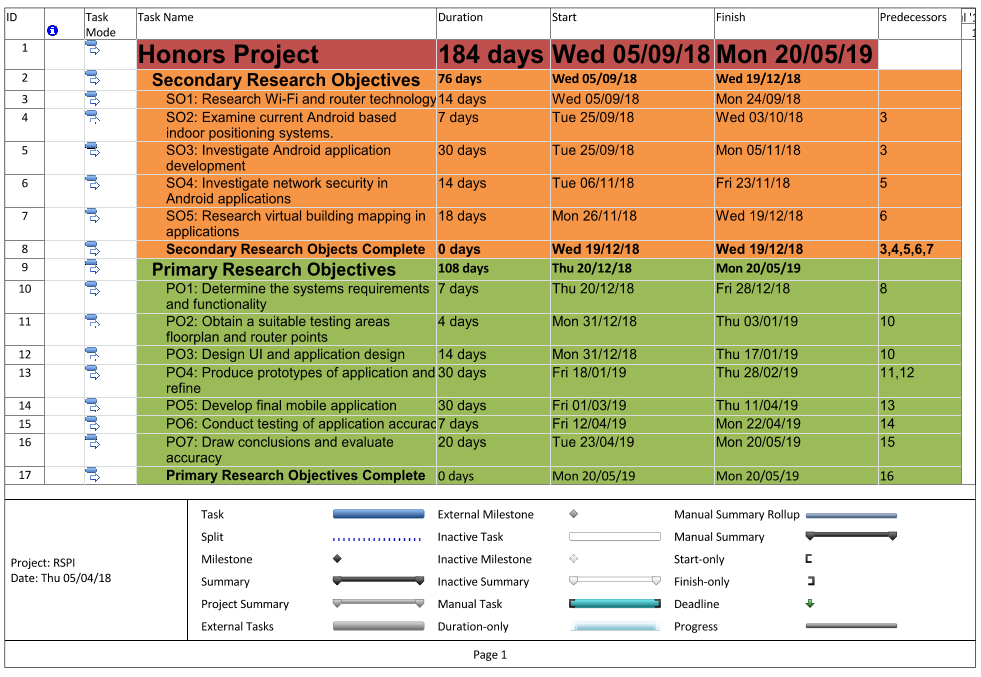


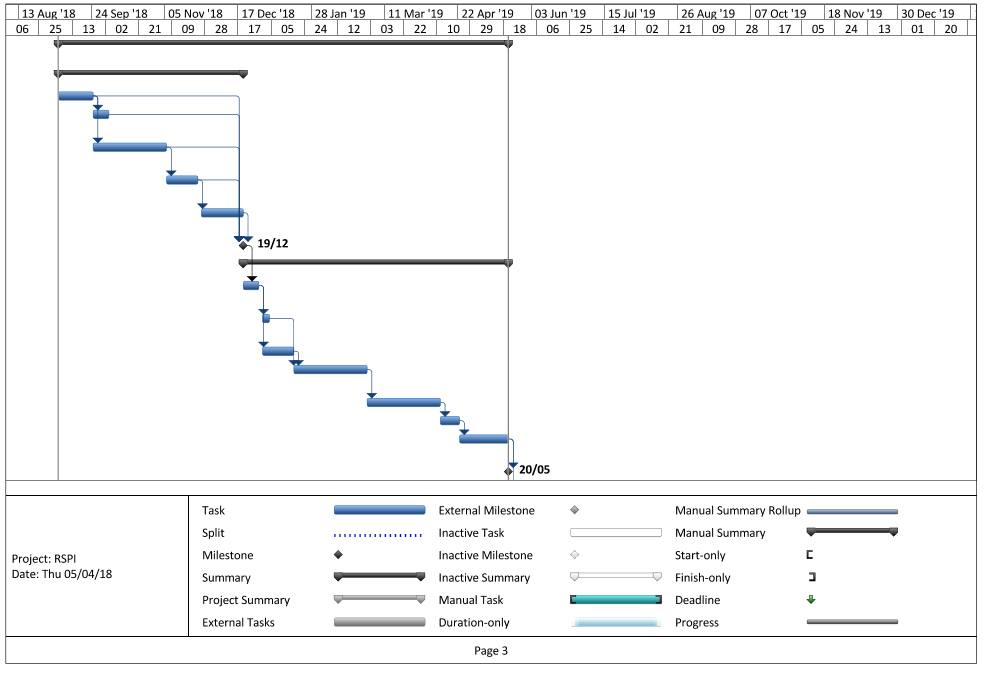
## Appendix 5 – Final Area Testing Screens



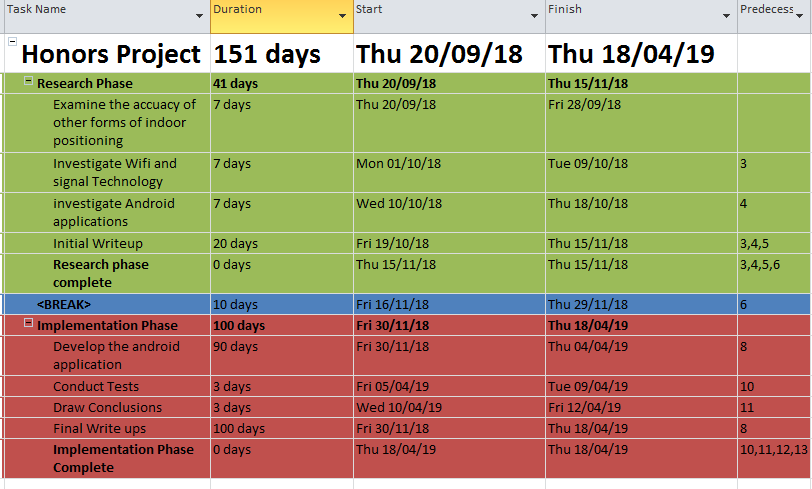
## Appendix 6 – Gantt Charts

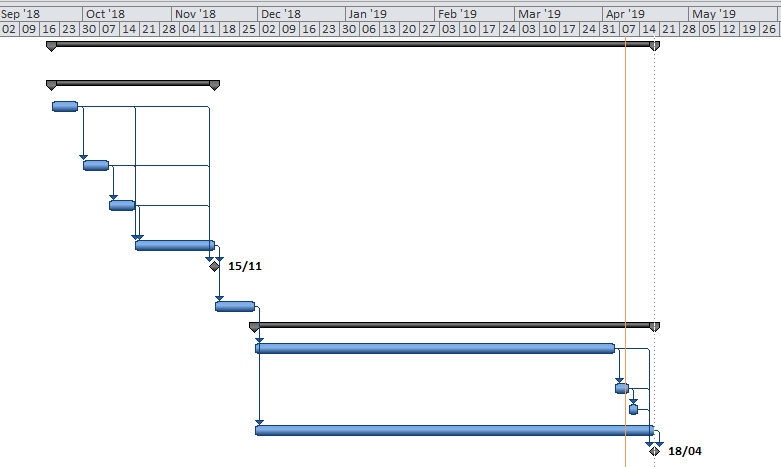
### Gantt Chart Initial



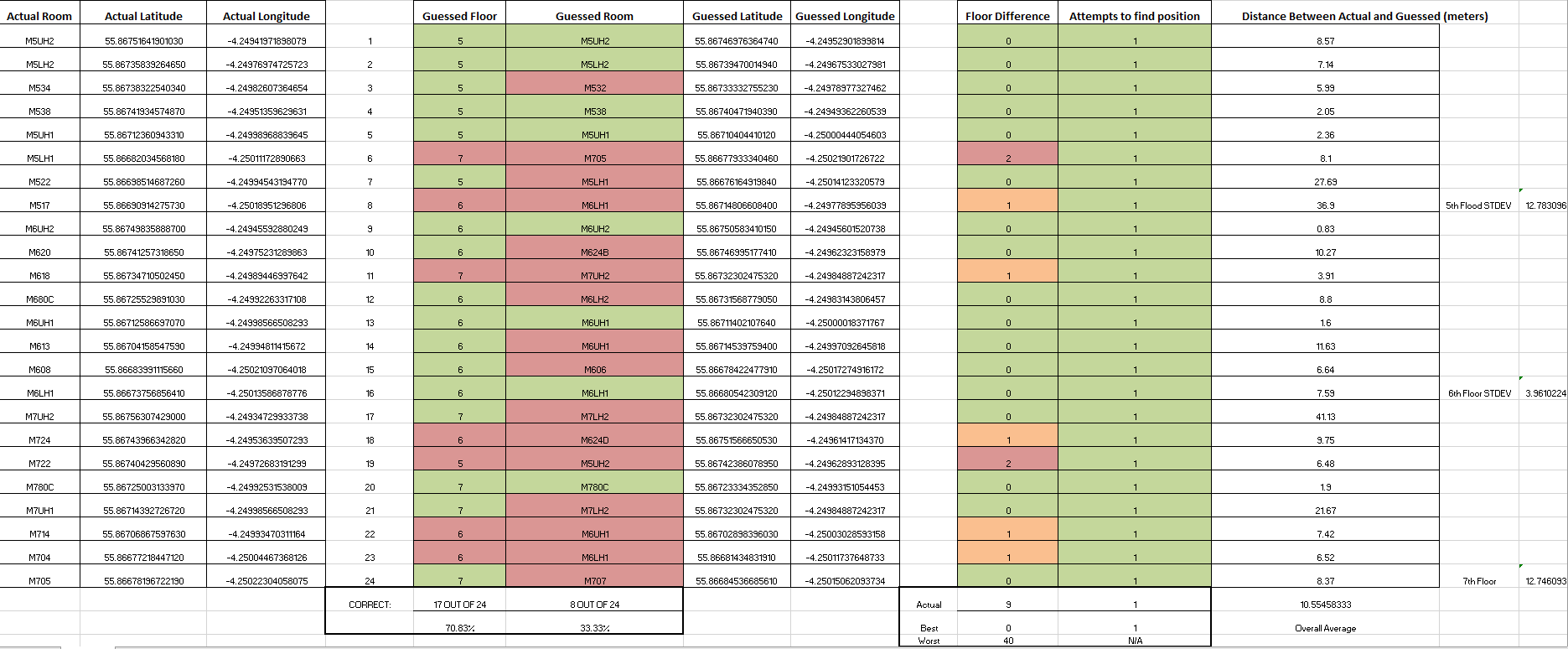


### Gantt Chart Final





## Appendix 7 – Raw Test Data



## Appendix 8 – Mind Map for SystemDescription: https://i.gyazo.com/36532a446c09468c89be8e59a70ac0f0.png

## Appendix 9 – Code

Code can be viewed on github: <https://github.com/mcllsti/Indoor-Positioning-WiFi-Android>

### MainActivity.Java

**package** honors.testing.test;  
  
*/\*\*  
 \* Daryl McAllister  
 \* S1222204  
 \* Indoor Positioning System  
 \* Honors Project  
 \* Iain Lambie  
 \*/***import** android.Manifest;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.content.pm.PackageManager;  
**import** android.location.Location;  
**import** android.net.wifi.ScanResult;  
**import** android.net.wifi.WifiManager;  
**import** android.net.wifi.rtt.WifiRttManager;  
**import** android.support.annotation.NonNull;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
  
**import** com.lemmingapex.trilateration.NonLinearLeastSquaresSolver;  
**import** com.lemmingapex.trilateration.TrilaterationFunction;  
**import** com.mapbox.mapboxsdk.Mapbox;  
**import** com.mapbox.mapboxsdk.maps.MapView;  
  
**import** org.apache.commons.math3.exception.TooManyEvaluationsException;  
**import** org.apache.commons.math3.fitting.leastsquares.GaussNewtonOptimizer;  
**import** org.apache.commons.math3.fitting.leastsquares.LeastSquaresOptimizer;  
**import** org.apache.commons.math3.fitting.leastsquares.LevenbergMarquardtOptimizer;  
  
**import** java.text.DecimalFormat;  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.HashMap;  
**import** java.util.List;  
**import** java.util.Timer;  
**import** java.util.TimerTask;  
  
**import** io.indoorlocation.core.IndoorLocation;  
**import** io.indoorlocation.manual.ManualIndoorLocationProvider;  
**import** io.mapwize.mapwizeformapbox.AccountManager;  
**import** io.mapwize.mapwizeformapbox.api.LatLngFloor;  
**import** io.mapwize.mapwizeformapbox.api.Venue;  
**import** io.mapwize.mapwizeformapbox.map.FollowUserMode;  
**import** io.mapwize.mapwizeformapbox.map.MapOptions;  
**import** io.mapwize.mapwizeformapbox.map.MapwizePlugin;  
**import** io.mapwize.mapwizeformapbox.map.MapwizePluginFactory;  
**import** pub.devrel.easypermissions.AfterPermissionGranted;  
**import** pub.devrel.easypermissions.EasyPermissions;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** MapView **mapView**;  
 **private** MapwizePlugin **mapwizePlugin**;  
 **private** ManualIndoorLocationProvider **manualIndoorLocationProvider**;  
 **private final int REQUEST\_LOCATION\_PERMISSION** = 1;  
 **double floor** = 0.0;  
 **private** WifiManager **wifiManager**;  
 **private** List<ScanResult> **results**;  
 **private** ArrayList<Double> **cordinates** = **new** ArrayList<>();  
 Venue **venue**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 *//Mapwize is built on mapbox so must instilize this* Mapbox.*getInstance*(**this**, **"pk.mapwize"**);  
 setContentView(R.layout.***activity\_main***);  
 requestLocationPermission();  
  
 *//Getting our map view* **mapView** = findViewById(R.id.***mapview***);  
 **mapView**.onCreate(savedInstanceState);  
  
 *//Setting style in acocrdance with mapwize* **mapView**.setStyleUrl(**"http://outdoor.mapwize.io/styles/mapwize/style.json?key="** + AccountManager.*getInstance*().getApiKey());  
  
  
 *//Setting the venue our indoor map will be based on* MapOptions opts = **new** MapOptions.Builder().centerOnVenue(**"5c5dc21da17ef7002dffd5e3"**).restrictContentToVenue(**"5c5dc21da17ef7002dffd5e3"**).floor(5.0).build();  
 **mapwizePlugin** = MapwizePluginFactory.*create*(**mapView**, opts);  
 **venue** = **mapwizePlugin**.getVenue();  
  
 *//Setting up wifimanager for scanning* **wifiManager** = (WifiManager) getApplicationContext().getSystemService(Context.***WIFI\_SERVICE***);  
 **if** (!**wifiManager**.isWifiEnabled()) {  
 Toast.*makeText*(**this**, **"WiFi is disabled ... We need to enable it"**, Toast.***LENGTH\_LONG***).show();  
 **wifiManager**.setWifiEnabled(**true**);  
 }  
  
  
 Toast.*makeText*(**this**, **"Scanning WiFi ..."**, Toast.***LENGTH\_SHORT***).show();  
 scanWifi(); *//scan to start positioning* }  
  
 */\*\*  
 \* Takes a none round number and coverts it to a rounded state to be used to position a user on a floor  
 \* Returns the whole floor number depending if the trilaterated number is within certain regions  
 \** ***@param currentGuess*** *Non whole number floor  
 \** ***@return*** *\*/* **public double** ConvertToWholeFloor(**double** currentGuess){  
  
 **if**(currentGuess < 5.667){  
 **return** 5.0;  
 }  
 **else if**(currentGuess >= 5.667 && currentGuess < 6.6902){  
 **return** 6.0;  
 }  
 **else if**(currentGuess >= 6.692){  
 **return** 7.0;  
 }  
 **return** 0.0;  
 }  
  
 */\*\*  
 \* Updates map with determined position of a user  
 \*/* **public void** loadedmap()  
 {  
  
 setupLocationProvider(); *//needed for positioning  
  
 //ensuring we have a whole number floor* **if**(**cordinates**.size() > 2)  
 {  
 **if**(Arrays.*asList*(6.0,5.0,7.0).contains(**cordinates**.get(2))){  
 **floor** = **cordinates**.get(2);  
 }  
 **else** {  
 **floor** = ConvertToWholeFloor(**cordinates**.get(2));  
 }  
 }  
  
  
  
 *//set the location from cordinates and floor* IndoorLocation indoorLocation = **new** IndoorLocation(**manualIndoorLocationProvider**.getName(), **cordinates**.get(0), **cordinates**.get(1), **floor**, System.*currentTimeMillis*());  
  
 **manualIndoorLocationProvider**.setIndoorLocation(indoorLocation);  
 **mapwizePlugin**.setFollowUserMode(FollowUserMode.***FOLLOW\_USER***);  
  
 *//Output used in testing to see cordinates* String coords = Double.*toString*(indoorLocation.getLatitude()) + **","** + Double.*toString*(indoorLocation.getLongitude());  
 Toast.*makeText*(**this**,coords,Toast.***LENGTH\_LONG***).show();  
  
 *//Ensure map switches to correct floor user is on* **mapwizePlugin**.setFloor(indoorLocation.getFloor());  
 }  
  
 */\*\*  
 \* ScanWifi method will be used almost exclusively to start a wifi scan  
 \*/* **private void** scanWifi() {  
  
 *//Clearing results to ensure accurate scan* **if**(**results** != **null**){  
 **results**.clear();  
 }  
  
 registerReceiver(**wifiReceiver**, **new** IntentFilter(WifiManager.***SCAN\_RESULTS\_AVAILABLE\_ACTION***));  
 **wifiManager**.startScan(); *//Start scan is deprecated and will be eventually removed. Should be changed if application is to continue.  
 //****TODO: Find another method to scan for future project development. StartScans deprecated status comes with limitations such as scanning frequency limits*** }  
  
 */\*\*  
 \* Used by Mapwize for location providing and indor locationing  
 \*/* **private void** setupLocationProvider() {  
 **manualIndoorLocationProvider** = **new** ManualIndoorLocationProvider();  
 **mapwizePlugin**.setLocationProvider(**manualIndoorLocationProvider**);  
 }  
  
  
 */\*\*  
 \* Proforms trilateration on N points providing within a ArrayList  
 \*  
 \* Created with the following resources:  
 \* https://github.com/lemmingapex/trilateration  
 \*  
 \*  
 \** ***@param list*** *list of positions and distances  
 \** ***@return*** *\*/* **public double**[] trilateration(ArrayList<DistancesPositions> list)  
 {  
 *//setting up our distances and positions in a more assesiable format* **double**[] distances = **new double**[list.size()];  
 **double**[][] positions = **new double**[list.size()][list.get(0).**position**.**length**];  
 **int** i = 0;  
 **for**(DistancesPositions e : list)  
 {  
 distances[i] = e.getDistance();  
 positions[i] = e.getPositionArray();  
 i++;  
 }  
  
  
  
 **double**[] centroid;  
  
 **try**{ *//to ensure program does not break  
  
 //Solver which solves trilateration* NonLinearLeastSquaresSolver solver = **new** NonLinearLeastSquaresSolver(**new** TrilaterationFunction(positions, distances), **new** LevenbergMarquardtOptimizer());  
 LeastSquaresOptimizer.Optimum optimum = solver.solve();  
 centroid = optimum.getPoint().toArray();  
  
 }  
 **catch**(Exception e){  
 *//If trilateration cannot be done then return an averaged position* centroid = list.get(list.size()-1).**position**;  
 }  
 **return** centroid;  
  
 }  
  
 */\*\*  
 \* Brocast reciever is an class which is called upon a completed scan  
 \*/* BroadcastReceiver **wifiReceiver** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
  
 unregisterReceiver(**this**); *//unregistering in case of future scans* **results** = **wifiManager**.getScanResults(); *//getting results* ArrayList<RouterMap> allRouters = setUp(); *//setting up our Uni router infestructure* ArrayList<DistancesPositions> listy = **new** ArrayList<DistancesPositions>();  
 **double**[][] positions = **new double**[20][2]; *//Never be more than 20 posisitions* String checker = **wifiManager**.getConnectionInfo().getSSID();  
 **int** i = 0;  
  
 *//For each of our results check if it is on our network and if so and is part of our acceptable routers  
 //add its position and distance to our list for trilateration* **for** (ScanResult scanResult : **results**) {  
 **if** (checker.equals(**'"'** + scanResult.**SSID** + **'"'**)) {  
  
 **try** {  
 positions[i] = allRouters.stream().filter(item -> item.getBSSID().equals(scanResult.**BSSID**)).findFirst().get().getCordinates();  
  
 listy.add(**new** DistancesPositions((**double**) scanResult.**level**, scanResult.**frequency**, positions[i]));  
  
 } **catch** (Exception e) {  
  
 }  
  
 **if** (!(positions[i][0] == 0 && positions[i][1] == 0)) { *//used to sync positions and distances* i++;  
 }  
 }  
 }  
  
 **if** (listy.size() < 2) {  
 *//If there is less than 2 router points we cannot proceed so we have to scan again* Toast.*makeText*(getApplicationContext(), **"Not enough router points to determine position "** , Toast.***LENGTH\_SHORT***).show();  
 Timer timer = **new** Timer();  
 timer.schedule(**new** TimerTask() {  
 @Override  
 **public void** run() {  
 scanWifi(); *//This must change as there is limit on scans due to deprecated* **return**; *//get out of method* }  
 }, 10000);  
 }  
  
 **else if**(listy.size() > 1 && listy.size() < 4){  
 listy = BulkResults(listy); *//have to bulk our results if we have less than 4 for trilateration to work* }  
  
 **double**[] centeroid = trilateration(listy);  
 **if** (centeroid != **null**) { *//as long as not null* **for** (**int** j = 0; j < centeroid.**length**; j++) {  
 **cordinates**.add(centeroid[j]);  
 }  
 loadedmap();  
 }  
 } };  
  
 */\*\*  
 \* Used to bulk our results by adding false results that are determined by averages of previous results which should not affect our overall acuracy.  
 \** ***@param listy*** *our current list that needs bulking  
 \** ***@return*** *\*/* **private** ArrayList<DistancesPositions> BulkResults(ArrayList<DistancesPositions> listy) {  
 **if**(listy.size() == 2)  
 {  
 *//If there is 2 then we need to add another 2 for trilateration* DistancesPositions x = listy.get(0);  
 x.**position**[0] = ((x.**position**[0] + listy.get(1).**position**[0])/2);  
 x.**position**[1] = ((x.**position**[1] + listy.get(1).**position**[1])/2);  
 x.**position**[2] = ((x.**position**[2] + listy.get(1).**position**[2])/2);  
 x.**distance** = ((x.**distance** + listy.get(1).**distance**)/2);  
 listy.add(x);  
  
 x = listy.get(0);  
 x.**position**[0] = ((x.**position**[0] + listy.get(1).**position**[0] + listy.get(2).**position**[0])/3);  
 x.**position**[1] = ((x.**position**[1] + listy.get(1).**position**[1] + listy.get(2).**position**[1])/3);  
 x.**position**[2] = ((x.**position**[2] + listy.get(1).**position**[2] + listy.get(2).**position**[2])/3);  
 x.**distance** = ((x.**distance** + listy.get(1).**distance** + listy.get(2).**distance**)/3);  
 listy.add(x);  
  
 }  
 **else**{  
 *//if there is three we only need one* DistancesPositions x = listy.get(0);  
 x.**position**[0] = ((x.**position**[0] + listy.get(1).**position**[0])/2);  
 x.**position**[1] = ((x.**position**[1] + listy.get(1).**position**[1])/2);  
 x.**position**[2] = ((x.**position**[2] + listy.get(1).**position**[2])/2);  
 x.**distance** = ((x.**distance** + listy.get(1).**distance**)/2);  
 listy.add(x);  
 }  
  
 **return** listy;  
 }  
  
  
 @Override  
 **protected void** onStart() { *//LIFECYCLE METHOD* **super**.onStart();  
 **mapView**.onStart();  
 }  
  
 @Override  
 **public void** onResume() { *//LIFECYCLE METHOD* **super**.onResume();  
 **mapView**.onResume();  
 }  
  
 @Override  
 **public void** onPause() { *//LIFECYCLE METHOD* **super**.onPause();  
 **mapView**.onPause();  
 }  
  
 @Override  
 **protected void** onStop() { *//LIFECYCLE METHOD* **super**.onStop();  
 **mapView**.onStop();  
 }  
  
 @Override  
 **public void** onLowMemory() { *//LIFECYCLE METHOD* **super**.onLowMemory();  
 **mapView**.onLowMemory();  
 }  
  
 @Override  
 **protected void** onDestroy() { *//LIFECYCLE METHOD* **super**.onDestroy();  
 **mapView**.onDestroy();  
 }  
  
 @Override  
 **protected void** onSaveInstanceState(Bundle outState) { *//LIFECYCLE METHOD* **super**.onSaveInstanceState(outState);  
 **mapView**.onSaveInstanceState(outState);  
 }  
  
 */\*\*  
 \* Sets up an Arraylist with our known University Wifi Infistructure  
 \** ***@return*** *\*/* **public** ArrayList<RouterMap> setUp()  
 {  
 ArrayList<RouterMap> allRouters = **new** ArrayList<RouterMap>();  
  
 *///SEVENTH FLOOR* **double**[][] Sevenpositions = **new double**[][] {  
 { 55.8675254490689, -4.249418377876283,7},  
 { 55.867424613296336,-4.249635636806489,7 },  
 { 55.867323024753276,-4.249848872423173,7 },  
 { 55.867104044101254, -4.250004440546037,7 },  
 { 55.86691140030752, -4.2500822246074685,7 },  
 { 55.86677933340469, -4.250219017267228,7 }  
 };  
  
 RouterMap SevenOne = **new** RouterMap(Sevenpositions[0],**"00:b7:71:97:dc:ad"**);  
 RouterMap Seventwo = **new** RouterMap(Sevenpositions[1],**"00:b7:71:aa:4d:0d"**);  
 RouterMap Seventhree = **new** RouterMap(Sevenpositions[2],**"00:b7:71:97:8f:ad"**);  
 RouterMap Sevenfour = **new** RouterMap(Sevenpositions[3],**"00:b7:71:a5:e4:6d"**);  
 RouterMap Sevenfive = **new** RouterMap(Sevenpositions[4],**"00:b7:71:ac:02:ad"**);  
 RouterMap Sevensix = **new** RouterMap(Sevenpositions[5],**"00:b7:71:a3:9d:c2"**);  
  
 allRouters.add(SevenOne);  
 allRouters.add(Seventwo);  
 allRouters.add(Seventhree);  
 allRouters.add(Sevenfour);  
 allRouters.add(Sevenfive);  
 allRouters.add(Sevensix);  
  
 *//////////////////////////////  
  
 //SIXTH FLOOR* **double**[][] Sixpositions = **new double**[][] {  
 { 55.867542756675164, -4.249380826950074,6},  
 { 55.86751566650539,-4.24961417913437,6 },  
 { 55.86737833410411,-4.249572604894639,6 },  
 { 55.86714806608404, -4.249978959560395,6 },  
 { 55.86695542250868, -4.250056743621827,6 },  
 { 55.86678911615361, -4.250126481056214,6 }  
 };  
  
 RouterMap SixOne = **new** RouterMap(Sixpositions[0],**"00:b7:71:99:14:ed"**);  
 RouterMap Sixtwo = **new** RouterMap(Sixpositions[1],**"00:b7:71:97:dd:8d"**);  
 RouterMap Sixthree = **new** RouterMap(Sixpositions[2],**"00:b7:71:96:cd:8d"**);  
 RouterMap Sixfour = **new** RouterMap(Sixpositions[3],**"00:b7:71:97:ca:22"**);  
 RouterMap Sixfive = **new** RouterMap(Sixpositions[4],**"00:b7:71:97:ca:82"**);  
 RouterMap Sixsix = **new** RouterMap(Sixpositions[5],**"00:b7:71:99:2f:2d"**);  
  
 allRouters.add(SixOne);  
 allRouters.add(Sixtwo);  
 allRouters.add(Sixthree);  
 allRouters.add(Sixfour);  
 allRouters.add(Sixfive);  
 allRouters.add(Sixsix);  
  
 *//////////////////////////////  
  
 //FIFTH FLOOR* **double**[][] FivePositions = **new double**[][] {  
 { 55.86751566650539, -4.249429106712342,5},  
 { 55.86742386078959,-4.249628931283952,5 },  
 { 55.867293676902484,-4.249779134988786,5 },  
 { 55.867114579281164, -4.250004440546037,5 },  
 { 55.866948273607605, -4.250066131353379,5 },  
 { 55.86676164919841, -4.250141233205796,5 }  
 };  
  
 RouterMap FiveOne = **new** RouterMap(FivePositions[0],**"00:b7:71:aa:42:2d"**);  
 RouterMap Fivetwo = **new** RouterMap(FivePositions[1],**"00:b7:71:aa:41:8d"**);  
 RouterMap Fivethree = **new** RouterMap(FivePositions[2],**"00:b7:71:aa:6f:82"**);  
 RouterMap Fivefour = **new** RouterMap(FivePositions[3],**"00:b7:71:99:13:cd"**);  
 RouterMap Fivefive = **new** RouterMap(FivePositions[4],**"00:b7:71:99:15:4d"**);  
 RouterMap Fivesix = **new** RouterMap(FivePositions[5],**"00:b7:71:97:de:ad"**);  
  
 allRouters.add(FiveOne);  
 allRouters.add(Fivetwo);  
 allRouters.add(Fivethree);  
 allRouters.add(Fivefour);  
 allRouters.add(Fivefive);  
 allRouters.add(Fivesix);  
  
  
 **return** allRouters;  
 }  
  
 */\*\*  
 \* Overriding permissions to allow us to acess WIFI and locations  
 \** ***@param requestCode*** *\** ***@param permissions*** *\** ***@param grantResults*** *\*/* @Override  
 **public void** onRequestPermissionsResult(**int** requestCode, String[] permissions, **int**[] grantResults) {  
 **super**.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
 *// Forward results to EasyPermissions* EasyPermissions.*onRequestPermissionsResult*(requestCode, permissions, grantResults, **this**);  
 }  
  
 */\*\*  
 \* Used to request location permissions on newer devices for our positioning and wifi  
 \*/* @AfterPermissionGranted(**REQUEST\_LOCATION\_PERMISSION**)  
 **public void** requestLocationPermission() {  
 String[] perms = {Manifest.permission.***ACCESS\_FINE\_LOCATION***};  
 **if** (EasyPermissions.*hasPermissions*(**this**, perms)) {  
 Toast.*makeText*(**this**, **"Permission already granted"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 EasyPermissions.*requestPermissions*(**this**, **"Please grant the location permission"**, **REQUEST\_LOCATION\_PERMISSION**, perms);  
 }  
 }}

DistancesPositions.java

**package** honors.testing.test;

*/\*\**

*\* Daryl McAllister*

*\* S1222204*

*\* Indoor Positioning System*

*\* Honors Project*

*\* Iain Lambie*

*\*/*

*/\*\**

*\* Class used to map a distance to a cordinate position*

*\*/*

**public class** DistancesPositions {

*/\*\**

*\* Overloaded Constructor*

*\** ***@param Distance*** *distance to position*

*\** ***@param Position*** *latLon position*

*\*/*

**public** DistancesPositions(**double** Distance, **double**[] Position){

**distance** = Distance;

**position** = Position;

}

*/\*\**

*\* Overloaded Constructor that determines a position upon creation*

*\** ***@param level*** *level of signal to router point*

*\** ***@param frequency*** *frequence in int Mhz of signal*

*\** ***@param Position*** *LatLon position*

*\*/*

**public** DistancesPositions(**double** level, **int** frequency, **double**[] Position){

**position** = Position;

**distance** = calculateDistance(level,frequency);

}

**double distance**;

**double**[] **position**;

*/\*\**

*\* Calcualtes the distance using Signal level and Frequency*

*\* Generally inaccurate as many factors can affect the outcome.*

*\**

*\* Code created using help from the following resources:*

*\* https://stackoverflow.com/questions/42512697/wifi-rssi-signal-to-distance-calculation-method-with-javascript*

*\* https://gist.github.com/eklimcz/446b56c0cb9cfe61d575*

*\* https://www.electronicdesign.com/communications/understanding-wireless-range-calculations*

*\**

*\** ***@param signalLevelInDb*** *Signal level of the RSSI*

*\** ***@param freqInMHz*** *Frequency of the signal*

*\** ***@return***

*\*/*

**public double** calculateDistance(**double** signalLevelInDb, **double** freqInMHz) {

**double** exp = (27.55 - (20 \* Math.*log10*(freqInMHz)) + Math.*abs*(signalLevelInDb)) / 20.0;

**return** (Math.*pow*(10.0, exp)) / 100000;

}

**public double** getDistance() {

**return distance**;

}

**public double**[] getPositionArray() {

**return position**;}}

### RouterMap.Java

**package** honors.testing.test;

*/\*\**

*\* Daryl McAllister*

*\* S1222204*

*\* Indoor Positioning System*

*\* Honors Project*

*\* Iain Lambie*

*\*/*

**import** java.util.ArrayList;

**import** java.util.List;

*/\*\**

*\*RouterMap Class*

*\* Used to match a set of LatLon cordinates to router using the routers BSSID*

*\*/*

**public class** RouterMap {

**private double**[] **cordinates**;

**private** String **BSSID**;

*/\*\**

*\* Overloaded Constructor*

*\** ***@param cord*** *array containing cordinates*

*\** ***@param bssid*** *String of BSSID*

*\*/*

**public** RouterMap(**double**[] cord, String bssid)

{

**this**.**cordinates** = cord;

**this**.**BSSID** = bssid;

}

**public double**[] getCordinates() {

**return this**.**cordinates**;

}

**public** String getBSSID() {

**return this**.**BSSID**;

}

}

### MainActivity.XML

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**tools:context=".MainActivity"**>

<**com.mapbox.mapboxsdk.maps.MapView**

**android:id="@+id/mapview"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"** />

</**android.support.constraint.ConstraintLayout**>

## Appendix 10 – Ethical Approval