Indoor Positioning System using existing smartphone technologies

By
Toufiqur R. Chowdhury

Supervised by
Mr. Asif Malik

A dissertation submitted in partial fulfilment of the
University of Greenwich's BEng Software Engineering

In the
School of Computing and Mathematical Sciences

April 2014

Word Count
[10199]

Keywords
[Indoor Positioning System, IPS, Machine Learning, Artificial Intelligence, Pattern Recognition, Algorithm, Fingerprinting, Triangulation, Indoor Navigation, Smartphone, Wi-Fi, Geomagnetic Field, Sensor Fusion, Inertial Navigation]

Declaration of Authorship

I, Toufiqur Rahaman Chowdhury, declare that this dissertation titled, Indoor Positioning System using existing smartphone technologies' and the work presented in it are my own. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University.
2. Where any part of this dissertation has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
3. Where I have consulted the published work of others, this is always clearly attributed.
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
5. I have acknowledged all main sources of help.
6. Where the dissertation is based on work done by myself jointly with others, I have distinctly identified what others have done and what I have contributed myself.

Inspirational quotes

“I have not failed. I've just found 10,000 ways that won't work.”
— Thomas Edison

“Success is not final, failure is not fatal: it is the courage to continue that counts.”
— Winston Churchill

Abstract

Global Positioning System (GPS) is an amazing technology that helped change human lifestyle immensely over the past few years. Contrary to the great benefits of GPS, there is a huge drawback of GPS when it comes to positioning and navigating indoors as the GPS signal can not or barely can penetrate building walls and other objects inside the buildings making it almost unusable in the indoor conditions. The modern smartphones are powerhouses and come with a plethora of sensors built into them. Even though these sensors are not primarily developed to be used for localization purposes, some of the sensors have the potential to be used for localization. This project explores the possibilities of building a self-sufficient Indoor Positioning System (IPS), which is affordable and easy to deploy, by harnessing the power of the modern smartphones and the sensors come with them. An application has also been developed that proves that it is possible to develop such a system.

Acknowledgement

Thanks to both of Mr. Asif Malik and Mr. Markus Wolf for continuous support and inspiration with the project.

Thanks to my classmate and friends (Tuhin, Ashif, Hassan and others) for encouragement and support.

Thanks to Jewel Marandy (a friend) for letting me use his Google Nexus 7 in many occasions.

Table of Contents

Abbreviations

1. IPS - Indoor Positioning System
2. GPS - Global Positioning System
3. GFS - Geomagnetic Field Strength
4. RSS - Received Signal Strength
5. RSSI - Received Signal Strength Indicator
6. RP - Reference Point
7. KNN - K-Nearest Neighbour
8. WKNN - Weighted K-Nearest Neighbour
9. SVM - Support Vector Machine
10. MVP - Minimum Viable Product
11. OS - Operating System
12. ADT - Android™ Development Toolkit
13. KML - Keyhole Markup Language
14. JavaML - Java Machine Learning Library

Chapter 1

Introduction

Global Positioning System (GPS) is an amazing technology that helped change human lifestyle immensely over the past few years. Contrary to the great benefits of GPS, there is a huge drawback of GPS when it comes to positioning and navigating indoors as the GPS signal can not or barely can penetrate building walls and other objects inside the buildings making it almost unusable in the indoor conditions. Another notable issue with GPS is that it is barely usable in critical weather conditions. A lot of research is being conducted with an ultimate goal to find an alternative solution for indoor positioning and navigation. Some of the researchers have been successful in building systems that works almost perfectly. Some of them even achieved centimetre level positioning accuracy error. Most of these technologies require a pre-deployment of infrastructures containing expensive hardware. This leads to not only an expensive deployment of the system but also makes maintenance expensive. Furthermore, due to the utilization of all these new technologies and/or infrastructures, the system requires the users to have a supported device containing very specific chips.

Smartphones have become a norm amongst most of the cities and towns all over the world. These smartphones are not only a powerhouses with multi-core processing power and advanced operating systems, they also come equipped with loads of sensors and chips inside. Although, none of these chips were primarily built with the intention for triangulation and positioning purposes, some of them can be used to gain a usable level of positioning accuracy. This makes smartphone technologies a viable alternative in building a much cheaper system with the possibility of reaching a much larger audience without requiring any additional equipment for the users.

1.1 Aim

The aim of this project is to explore the possibilities of utilizing one or more of these existing smartphone technologies to develop a mobile application for Indoor Positioning System and also explore the possibilities of building and deploying indoor maps and a routing system.

1.2 Objectives

1.2.1 Background research and idea development

1.2.1.1 Understanding how IPS system works and explore existing solutions
1.2.1.2 Search for and read relevant journals and publications [4.0]
1.2.1.3 Read books [4.0]
1.2.1.4 Search for and research existing solutions [4.0]
1.2.1.5 Develop the minimum requirements for a MVP [3.0]
1.2.1.6 Analyse the best suitable solution [2.0]

1.2.2 Software design and planning

1.2.2.1 Develop a plan for the software implementation and any required database using requirements for the MVP and design diagrammatic schemas
1.2.2.2 Design database schema and normalize to at least 2NF [3.0]
1.2.2.3 Explore design considerations and software design patterns [2.0]
1.2.2.4 Design diagrammatic schemas of software structure and model [7.0]
1.2.2.5 Develop a test plan for the system [2.0]

1.2.3 Minimum Viable Product (MVP) development

1.2.3.1 Development of the MVP software following a suitable software development methodology
1.2.3.2 Experiment with existing solutions [10.0]
1.2.3.3 Experiment with various fingerprinting algorithms [10.0]
1.2.3.4 Implementation one or more the solutions and/or algorithms experimented earlier [7.0]
1.2.3.5 Design and develop indoor map using suitable tools [3.0]
1.2.3.6 Develop a routing system for the indoor map [7.0]
1.2.3.7 Software testing [4.0]
1.2.3.6 Documentation [4.0]
1.2.3.8 Maintenance and debugging [7.0]
1.2.3.9 Improvements and further development possibilities [7.0]

1.2.4 Evaluation

1.2.4.1 Evaluate the system by testing it in real life using a test subject environment location
1.2.4.2 Take fingerprint data from the subject location [3.0]
1.2.4.3 Test positioning [1.0]
1.2.4.4 Test map routing & navigation [1.0]

1.2.5 Deployment

1.2.5.1 Compile application for demonstration [1.0]
1.2.5.2 Report Writing [45.0]

1.3 Development method

There are many different software development methodologies available. Waterfall, Agile and various other methods are popular among developers, as they have proven to be reliable and more suitable in many different circumstances. In contrary, this project is a proof of concept and requires a lot of experimentation. Therefore, it is required that a method is to be selected that provides with the advantage of changing requirements easily during the development process. Due to this nature of the project, the Software Prototyping methodology has been selected as the primary development method.

The way Software Prototyping works is that it attempts to reduce inherent project risk by breaking down a project into smaller segments and providing more ease-of-change during the development process. Furthermore, one of the largest advantages of following Software Prototyping is that the major segments (and sub-segments, if necessary) can be developed using different development methods and at the end put them all together. Therefore, the entire development of the application has been broken down into two major segments: 1) Building an indoor map with a custom routing and navigation system and 2) implementation of location fingerprinting algorithm. Afterwards, these two segments can be broken down into multiple smaller sub-segments and to be implemented, if necessary. Initially, small-scale mock-ups of the major segments and sub-segments, where necessary, will be developed following an iterative development cycle until each prototype evolves to meet their specific requirements and tested to a satisfactory level.

1.4 Project Management

Aside from following the development methods and executing the project plan in a timely manner, all necessary designs will be developed. All Story boards, Database Schema's and UML diagrams will be drawn up allowing for a basic program structure to be created before the programming begins. The designs will help make all the necessary technical decision and also make changes along the process of the development of the Campus Explorer application. Since, there will be a lot of experiments while developing the application, lot of changes will be required along the way and the design will be there to guide through the changes. A schedule of work has been developed from the objectives of the project transformed into a Gantt chart. The work schedule will be listed in the Appendix section of this report.

Chapter 2

Literature Review

2.1 Introduction

The aim of this literature review is to explore and investigate the possibilities of Indoor Positioning System using existing smartphone technologies. The discussion will also include comparison with any available alternative technologies; especially focussing on whether it is possible to build a working system without pre-deployment of expensive infrastructures.

2.2 Current Situation

"The increasing demand for location-based services inside buildings has made indoor positioning a significant research topic. The applications of indoor positioning are many, for instance, indoor navigation for people or robots, inventory tracking, locating patients in a hospital, guiding blind people, tracking small children or elderly individuals, location based advertising, ambient intelligence etc." (Jekabsons and Kairish et al., 2011, pp. 131--137).

Indoor Positioning System (IPS), unlike Global Positioning System (GPS), is yet to be standardized. Various IPS technologies are being researched, experimented and developed by many different organizations. Currently no particular technology exists that works perfectly in most of the general use case scenarios. More on this will be discussed in the problems section of this report. Being said that, in order to develop an application based on IPS, the first thing needed to done was to choose at least one particular technology from all the existing solutions that suites the majority of the objectives of this project. The rest of the literature review will discuss most viable technologies that are currently available and/or being developed.

2.3 Smartphones In IPS

Smartphones in modern lifestyle is not just a tool anymore; it is an essential part of modern life and culture. Smartphones do not only provide various ways of communications but also provide entertainment, a limited amount of creativity and document authoring and so on. In contrary, majority of these smartphone operating systems are far more advanced compared to a desktop computer operating system of less than a decade old. Majority of recent smartphones are equipped with multi-core processors with Gigabytes of RAM. Furthermore, all of these smartphones comes with plethora of sensors built-into, an advantage that no computer systems ever had. All these horsepower and the sensors are just sitting there doing almost nothing, majority of the time.

This processing power can be harnessed to develop far more complicated and feature rich applications. Even smartphones can be used to build systems that have never been imagined before. A recent initiative called *HTC Power To Give*[1] by HTC proved just that. The idea of this initiative is to use thousands of smartphones to build a supercomputer by harnessing the processing power.

[1] For more information about HTC Power To Give: http://www.htc.com/www/go/power-to-give/

2.3.1 Potentials Of smartphone Sensors

The most common sensors in recent smartphones aside from GPS are Wi-Fi, Accelerometer, Gyroscope and Magnetometer or commonly known as Compass. Each of these sensors holds great potential in building an indoor positioning system.

Fingerprinting Or Triangulation Using Wi-Fi

Many researchers have been experimented and build IPS using Wi-Fi signals alone. Redpin, an open source IPS, is a great example of such system. In contrary, majority of the modern city houses have Wi-Fi routers. This makes Wi-Fi highly viable technology to be considered in an IPS. Not to mentions, it is also very affordable technology. In the words of Chen and Li:

"Wi-Fi provides local wireless access to a fixed network that is low cost, widely deployed and whose indoor coverage is still rapidly increasing. Using existing infrastructure for positioning is a very attractive option." (Chen and Li et al., 2013, pp. 11085--11096)

Another great advantage of Wi-Fi is the wide area coverage. Li and Gallagher say:

"The low cost and wide coverage of such methods are the main advantages." (Li and Gallagher et al., 2012, pp. 1--9).

More about Redpin will be discussed in the Related Work section of this report.

Sensor Fusion Using Accelerometer And Magnetometer

FootPath is another open source project that uses mainly accelerometer and magnetometer/compass for indoor localization. In the words of the creators of FootPath:

"We present FootPath, a self-contained, map-based indoor navigation system. Using only the accelerometer and the compass readily available in modern smartphones we accurately localize a user on her route, and provide her with turn-by-turn instructions to her destination. To compensate for inaccuracies in step detection and heading estimation, we match the detected steps onto the expected route using sequence alignment algorithms from the field of bioinformatics." (Link and Smith et al., 2011, pp. 1--8)

More about FootPath will be discussed in the Related Work section of this report.

Magnetometer in smartphones is mainly used to determine heading by pointing to the north. Using this sensor it is possible to measure the level of Earth's Geomagnetic Field Strength (GFS). Earth's GFS is yet another possible candidate for research in building IPS. Several researchers have suggested the use of Earth's GFS for indoor positioning and navigation purposes. These researchers believe that it is possible to build an IPS using the Earth's GFS alone. According to Li and Gallagher:

"for indoor applications, accurate heading determination is difficult due to the presence of magnetic field anomalies. Here location fingerprinting methodology can take advantage of these anomalies. In fact, the more significant the local anomalies, the more unique the magnetic "fingerprint". In general, in each fingerprint, the more elements, the better for positioning." (Li and Gallagher et al., 2012, pp. 1--9)

A company called IndoorAtlas Limited has been working on an IPS; called IndoorAtlas; based on Earth's GFS. The company is planning on releasing the system commercially. The have currently released a beta version of the system. More will be discussed about it IndoorAtlas in the Related Work section of this report.

Therefore, it is clear that building an IPS, using existing smartphone technologies, is completely possible and it is viable cheaper alternative to infrastructure based systems.

2.4 The Basics Of Indoor Positioning

Before comparing different technologies and choosing a particular technology, it is vital to understand how Indoor Positioning Systems generally work and the variation of technologies available.

"The indoor positioning technologies can be classified into three categories: technologies based on signals-of-opportunity, technologies based on pre-deployed infrastructure, and others." (Li and Gallagher et al., 2012, pp. 1--9).

2.4.1 Signals-Of-Opportunity

According to Li and Gallagher, signals-of-opportunity are the signals that are around and have not been intended for the purpose of localization. These kinds of signals include Wi-Fi, Radio Frequency (RF), AM & FM radio signals, Television (TV), Earths geomagnetic field strength and even cellular signals. Li and Gallagher also mentions that these signals are designed for other purposes, and the reality of the harsh signal propagation environment, using these technologies for positioning to achieve high accuracy is a very difficult, if not impossible. In contrary, since these technologies do not require pre-deployment of any infrastructure, they are extremely cheap and widely available.

2.4.2 Infrastructure Based Systems

Technologies that require pre-deployment of infrastructures specifically for localization are known as infrastructure based technology. Positioning systems using infrared, ultrasound and ultra wide band, as well as RF-based systems are considered as infrastructure based system. While these technologies are generally very expensive and provide very limited range, in most cases they offer very high level of accuracy; according to Li and Gallagher.

2.4.3 Other Technologies

There are few other technologies available that do not make use of signals at all. Inertial Navigation systems and Vision-based systems are among the few have been explored by the researchers. The inertial navigation systems uses users' height, movement speed and step count in order to estimate how far and which direction they have walked towards.

2.5 Triangulation Versus Fingerprinting

In terms of localization technique IPS's can be developed using two different methods: triangulation based IPS and fingerprint based IPS.

The triangulation method in IPS comes from the Global Positioning System (GPS). The basic idea of triangulation is to pin point a receivers location based on the signals and the data received at a particular time and the relative location of other transmitters. *"In general, a GPS receiver uses information from the GPS satellites orbiting the earth to calculate its current location. The GPS system contains 27 satellites that continually orbit the earth, transmitting information to would-be receiver"* (Milette and Stroud, 2012, p. 4)

In case of IPS, the bacons; such as: Wi-Fi or Bluetooth access points; are the transmitters, which constantly transmit signals. *"Wi-Fi triangulation's goal is to map RSSI as a function of distance. This method requires a steep linear characterization curve in order to be properly implemented."* (Quan and Navarro et al., 2010)

Fingerprinting is the most commonly used localization technique among the IPS systems available. *"A methodology known as 'fingerprinting' is widely used where signal propagation is unpredictable or where direct line-of-sight propagation is not typical (including presence of multipath).* ***The low cost and wide coverage of such methods are the main advantages."*** (Li and Gallagher et al., 2012, pp. 1--9).

Fingerprinting takes a different approach to triangulation. Instead of attempting to pin point the location of the receiver by calculations based on the location of the transmitters, the location is estimated by matching the Received Signal Strength (RSS) of the available signals against pre-recorded data from the fingerprint database. More about the fingerprint database will be discussed later in this chapter.

While triangulation approach of GPS provides great results in pin pointing location and the implementation of the Wi-Fi triangulation system is relatively simple, in case of IPS, it is less reliable due the variation of RSS in the indoor environments. *"Due to the time-variation property of radio propagation in indoor WLAN environment (e.g., multi-path effect, RSS shadowing, and adjacent channel interference), the fingerprint-based positioning has been more preferred in practical use."* (Tian and Tang et al., 2013, pp. 1--8).

"For outdoor environments, a Global Navigation Satellite System (GNSS) such as Global Positioning System (GPS) is ideal. However, GPS is not suitable for indoor environments, as the satellite signals cannot penetrate walls or roof of buildings. Furthermore, when assisted-GPS techniques are used, the position may have errors of tens of meters." (Chen and Li et al., 2013, pp. 11085--11096)

That is why fingerprinting is preferred by researchers and is being widely used in many indoor positioning systems.

2.6 Building Fingerprint Database

Creating a fingerprint database is one of the most crucial parts of developing an indoor positing system. In contrary, the creation of fingerprint database and it maintenance is not a trivial task.

"The major problem of Wi-Fi fingerprint-based positioning technology is the signal strength fingerprint database creation and maintenance. The significant temporal variation of received signal strength (RSS) is the main factor responsible for the positioning error." (Chen and Li et al., 2013, pp. 11085--11096)

In order to create the fingerprint database, firstly, a radio map is created. The map consists of a collection of pre-selected reference points (RP). *"Wi-Fi Fingerprinting creates a radio map of a given area based on the RSSI data from several access points and generates a probability distribution of RSSI values for a given (x,y) location."* (Quan and Navarro et al., 2010)

The number of RP's required varies depending on the total area and the required rate of accuracy. Generally, the higher the number of RP's available the higher the possibility of achieving better accuracy rate. The RSS's at each of the RPs are measured and entered into the database along with the MAC address of the access points. In the fingerprint technique, *"the measurements (including RSSs and MAC addresses of the APs) are compared with the data in the database using a matching algorithm"* (Chen and Li et al., 2013, pp. 11085--11096).

2.7 Localization Algorithms

Localization algorithm plays the most significant role in an IPS. Use of a wrong algorithm can lead to completely disastrous results. There are a wide variety of machine learning algorithms and techniques available for localization purposes. K-Nearest Neighbour (KNN), Weighted K-Nearest Neighbour (WKNN), Support Vector Machine (SVM), Naïve Bayes Classifier, K-D Tree and Weka classifier are among the most popular ones. Each of these algorithms takes slightly different approach for classification.

The RSS values are crucial and sensitive in the indoor positioning systems. The signal strength at the indoor locations fluctuates often. The variation is caused by different reasons; such as: change in the decoration of a room or use of electronic devices. The Wi-Fi signal strength is highly affected by the interference of the electronic devices. Even the presence of a human being can cause decreased RSS values due to the higher level of water in the human body. While KNN/WKNN, K-D Tree and SVM are among the most common used algorithms in localization systems, due to this varying nature of indoor signals, they do not always work as expected. Many researchers take different approaches in order to get around this very critical issue. While some researchers attempt to construct a custom algorithm, some others propose hybrid algorithms constructed from two or more of the localization algorithms.

Among the first group of researcher, Shin and Lee proposed a custom algorithms that enhances the Weighted K-Nearest Neighbour algorithm. According to them, KNN is not suitable for IPS. They say: *"The K-nearest neighbour fingerprinting algorithm uses a fixed number of neighbours, which reduces positioning accuracy."* (Shin and Lee et al., 2012, pp. 574--577). Therefore, they proposed: *"a novel-fingerprinting algorithm, the enhanced weighted K-nearest neighbour (EWKNN) algorithm, which improves accuracy by changing the number of considered neighbours"*. According to them this improved algorithm resulted in a much higher accuracy level.

Among the second group of researchers, Del Mundo and Macatangga proposed a hybrid algorithm: *"Many algorithms proposed for indoor positioning using Wi-Fi based networks, those that rely on hybrid-based approach have been demonstrated to outperform those based on deterministic, probabilistic, and pattern recognition in terms of accuracy."* (Del Mundo and Macatangga, 2012, pp. 107--112). They claimed to have achieved an impressive 89.47% accuracy rate in their test system.

2.8 Indoor Maps And Routing

Building an indoor map for the client application is an essential part of the IPS. In fact, a map is the only thing that is visible to the end-users of the system. Building an indoor map is time consuming. It takes planning and lots of considerations, especially for the indoor locations. Two major facts must be considered while building an indoor map. Firstly, the areas that are to be visible in the map and publicly available must be identified and thoroughly considered. Secondly, developing a route map is essential for navigation purposes.

There are various different approaches to building an indoor map. Whether the indoor map is to be developed, as 2-dimensional (2D) or 3-dimensional (3D) map is also crucial to the development of the map. While 2D maps are relatively simple to develop and much lighter for the client applications, a well-built 3D map can result in a much richer user experience.

Google Maps Engine[2] is a great online tool for creating 2D indoor maps. It allows creating an overlay of indoor map on top of the Google Maps with a What You See Is What You Get (WYSIWYG) interface, which then can be exported as a KML (an XML format) file. This KML file can be used with any other map services as it does not rely on the Google Maps and contains coordinates for each of the nodes.

[2] For more information about Google Maps Engine: https://mapsengine.google.com/

2.9 Cost

The total cost of implementation and maintenance is one of the major factors while choosing a technology for building an IPS. While infrastructure based IPSs are proved to be more reliable and accurate, these systems are expensive both in deployment and maintenance. Researchers Li and Gallagher words just proves the point:

“Deploying new infrastructure is costly, and hence the coverage is often very limited – typically “hot spot” mode. However, if a reliable and accurate positioning result is required, such technologies typically have to be used.” (Li and Gallagher et al., 2012, pp. 1--9)

In contrary, the great advantage of using existing smartphone technologies for IPS is the low cost implementation and maintenance. Jekabsons and Kairish say:

“Indoor positioning using the Wireless Ethernet IEEE 802.11 (Wi-Fi) standard that has a distinct advantage of low cost over other indoor wireless technologies – it has relatively cheap equipment and in many areas usually a Wi-Fi network already exists as a part of the communication infrastructure avoiding expensive and time-consuming infrastructure deployment.” (Jekabsons and Kairish et al., 2011, pp. 131--137)

For an IPS based on or using Wi-Fi signals, expansion of coverage is easy and affordable. If a particular area is not covered by Wi-Fi; it is possible to buy Wi-Fi routers, which are extremely affordable and available in most of the computer stores.

On the other hand, IPSs based on sensor fusion does not require any signals. Therefore, expansion of coverage does not require any installation of new hardware.

Again, the entire Earth is covered by its geomagnetic field. Therefore, no hardware is required either for expansion of IPSs based on Earth’s GFS.

2.10 Issues

As with any technology there are some disadvantages of using smartphone technologies as well. Below is a brief discussion about the shortcomings of these technologies:

Fluctuation Of signals

One of the major problem with the IPSs based on signal-of-opportunity is the fluctuation in the RSSI values. Bilke and Sieck confirms:

"One disadvantage of this approach is that the signal propagation in the 2.4 GHz band cannot be fully predicted. Magnetic fields, which are distorted in buildings, can, however, be used in finger- printing techniques such as Wi-Fi." (Bilke and Sieck, 2013, pp. 195--208)

Issues Of Sensor Fusion Based IPS

IPSs like FootPath; that are based on sensor fusion has one of the great disadvantage is that the user must calibrate and train the system (the application) before use. This calibration and training phase need to be repeated whenever a different user wants to use the system.

Issues Regarding Earth's GFS

While Earth's GFS is available everywhere on Earth, it is prone to fluctuation as well. Any change on the structure of the building, including any change in the arrangement of furniture and equipment's; that has magnetic properties; such as metal equipment, lifts and refrigerators; may potentially change the RSSI value of GFS. While tiniest such change might not make a big difference, a reasonably large change may require the fingerprint data of the entire system to be taken each time.

2.11 Conclusion

In the light of above discussion, it can be said that IPS based on smartphone technologies and signal-of-opportunity is not only possible but also a viable alternative to expensive infrastructure-based systems. The high level accuracy of infrastructure-based systems may not always be necessary. This will be the case where, finding a room or a certain area is the ultimate goal of the system. IPS systems for shopping malls, hospitals, large buildings and universities are more likely to require only room level accuracy. Therefore, even with the low level of accuracy, the positioning systems based on signal-of-opportunity will be viable and more appropriate choice. This is where the use of existing smartphone technologies gives an edge to the IPSs.

Chapter 3

Related Work

3.1 Introduction

Indoor positioning technologies are yet to be standardised and more or less a proof of concept. This means that there are not many widely accepted solutions out there. Many large corporations are researching on it and some of them are more or less successful. Apple, Ericsson, Google and Nokia are among the leaders in this area of research and they are getting a lot attention in this regard. Most of the solutions are infrastructure based. This means using a solution from one of these solutions would require expensive hardware instalment. Not to mention that none of these companies has made their solutions free, let alone being open source. In contrary, the primary object of this project is to use existing solutions and build an affordable system. Therefore, this section will be based on products and/projects that are free to use or reasonable prices and/or open source.

3.2 Projects

Even though there are many different IPS technologies already exists or are being developed, not many are available for free, let alone being open source. Below are some of the technology/solutions available that are free to use and/or open source and have been tested before choosing a particular one, if any are to be chosen at all.

3.2.1 Redpin

While researching for project and in the hunt for an existing free to use and/or open source solution online, the one name came up again and again was Redpin. Redpin[3] is an Open Source IPS technology based on Wi-Fi fingerprinting. It happens to be completely free for any kind of use as well; which alone is a very compelling reason for Redpin being at the top of considered solutions list. While testing the Redpin system, it was possible to achieve a room level accuracy. In the test runs, the system was able to easily identify reference points as close as 5 metres, although there were a few minor hiccups from time to time. This is a big advantage of Redpin. In the words of the creators of Redpin:

[3] Redpin was developed with the goal of gaining at least room-level accuracy using only Wi-Fi signal strength. Redpin is open source and freely available online: http://redpin.org

"*Redpin is a fingerprint-based indoor localization system de- signed and built to run on mobile phones. The basic principles of our system are based on known systems like Place Lab or Radar. However, with Redpin it is possible to consider the signal-strength of GSM, Bluetooth, and Wi-Fi access points on a mobile phone. Moreover, we devised methods to omit the time-consuming training phase and instead incorporate a folksonomy-like approach where the users train the system while using it. Finally, this approach also enables the sys- tem to expeditiously adapt to changes in the environment, caused for example by replaced access points*." (Bolliger, 2008, pp. 55--60)

Below is a brief list of pros and cons of Redpin:

3.2.1.1 Pros of Redpin

1. Redpin makes use of multiple signals of opportunities. Such as Wi-Fi, Bluetooth and GSM. Thus, no infrastructure is required
2. Redpin is adaptive to the changes in the environment of the subject area. Such as: change in Wi-Fi access points or changes in furniture setup.
3. Possibility to develop a system only based on Wi-Fi or Bluetooth technology.
4. Provides room-level accuracy
5. This solution consists of two-tier applications: a Java based server application and a smartphone application as the client. This means all the fingerprint data can be managed from a single server.
6. It is possible to host the server either in a local server with access to the Wi-Fi network or hosted in a remote server.
7. Wide variety of platform support: Android, IOS, Symbian and Windows Mobile

3.2.1.2 Cons of Redpin

7. The largest disadvantage of Redpin is that it is only a proof of concept; which does seem to work but there is a lot to improve. The development and support has also been halted since August 2010. Resulting in a lot unresolved bugs in their support site.
8. The client application requires connection to the server for all the data and algorithmic calculations for fingerprinting, rendering the client almost useless without a server.
9. There is no routing and navigation solution available in the system.
10. The demo application uses an image based mapping system and the experience is cumbersome. This also means that it is not possible to even display a route from an external location to an indoor position.
11. IOS support is limited to only Jailbroken (hacked by the user) units.

3.2.2 IndoorAtlas

IndoorAtlas[4] was the second in the considered solutions list. This was the only product that is being public made available commercially. While IndoorAtlas is not an open source project, it does have a free edition with limited usage. Furthermore, currently it is in beta mode, available only for free users, and pricing has not been published so far (as of the writing of this report).

IndoorAtlas takes a completely different approach to the indoor positioning technologies currently available. It uses Earth's geomagnetic field strength for localizations purposes. Since, it is intended for commercial use, all the secrets behind the system have not been revealed by its creators. This means, it is not clear whether other technologies are utilized as well. Below is a brief list of pros and cons of IndoorAtlas:

3.2.2.1 Pros of IndoorAtlas

1. It mainly uses Earth's geomagnetic field strength; which is a signal of opportunity. Therefore, no infrastructure is required to be setup.
2. An API is available for developers to integrate indoor positioning system and the map into their applications.
3. Uses easy to create image based mapping system.
4. API supports routing and navigation.
5. The company claims, accuracy is typically less than 3 meters.
6. This is a product intended for commercial use. Therefore, there will always be support and improvement in the system. Furthermore, the creators of IndoorAtlas are researchers from the University of Oulu, Finland. This makes for a guaranteed further development of the product.
7. Supports users of both Android and IOS devices

3.2.2.2 Cons of IndoorAtlas

1. Requires an active Internet connection.
2. IndoorAtlas is a proprietary technology. Developers (clients of IndoorAtlas) have no access to the fingerprint data or any localization techniques.
3. Development is possible only using a very few number of supported mobile devices. Although, this does not affect the user-base.
4. In the test run of the system, using company provided application on a Google Nexus 7 tablet, it was not possible to receive any reliable accuracy level. In fact, most of the time application located the incorrect position.
5. Free edition users can use it for a very limited amount of time per month; which makes it viable for a product to be released immediately.

[4] IndoorAtlas is a beta technology currently freely available for beta users and will be commercially launched soon. For more information: http://indooratlas.com

3.2.3 FootPath

FootPath[5] is yet another Open Source project that attempts to solve the IPS problem with a very different approach. FootPath uses sensor fusion to detect steps and heading estimation for indoor navigation and positioning as an alternative to fingerprint based systems. In the words of the creator of FootPath:

"Our approach to this problem is twofold: (1) we use simple step detection and step heading estimation. (2) We match detected steps onto the expected route from the source to the destination using sequence alignment algorithms. Instead of a more general localization problem, we solve the localization problem on a specified route. This allows us to compensate for inaccuracies and give the user accurate turn-by-turn directions." (Link and Smith et al., 2011, pp. 1--8)

While experimenting and testing various technologies the FootPath application have been tested as well. Below is a compilation of pros and cons of FootPath:

3.2.3.1 Pros of FootPath

1. Does not rely on any kind of signals-of-opportunity; such as: Wi-Fi, Bluetooth
2. Does rely on any kind of infrastructure at all
3. Completely self-independent system/application. No server or Internet connection required for localization but does require Internet connection to load map from OpenStreetMap[6].
4. Supports a wide variety of Android OS based devices and the idea might be possible to implement on other platforms as well
5. It is possible to develop the system with offline maps.

3.2.3.2 Cons of FootPath

1. Requires calibration and testing for each user before it is usable
2. FootPath cannot pin point the user at first. The user must set the starting point manually.
3. The heading estimation might not always be correct. Simply because the user may walk faster or slower than the walking speed while in the calibration phase.
4. The mapping system completely relies on OpenStreetMap. That means mapping only a single floor is supported.
5. FootPath is not yet ready to be released as a product and requires a lot more research and experimentations.

---

[5] FootPath is an open source project, developed by the students of RWTH Aachen University as a part of their master's project. For more information: http://www.comsys.rwth-aachen.de/research/projects/footpath

[6] OpenStreetMap is an open license map created with the continuous collaboration by many people around the world. For more information: http://www.openstreetmap.org/

3.2.4 Indoor Position Tracker

Indoor Positioning Tracker[7] (IPT) is research project developed by a student at the University of Helsinki, Finland. The project demonstrates the possibility of indoor positioning based on Wi-Fi fingerprinting. Furthermore, IPT is a completely standalone application. This means all the fingerprint data and the calculation of localization algorithms are run from within the smartphone itself.

While in the test runs it did seem to locate the user to the nearest fingerprint from time to time, mostly it was inaccurate. More importantly, the system seems to be always confused and keep changing position in the map even though the user does not move at all. This is possibly due to the fluctuation of Wi-Fi signal strengths. A more customized implementation of the localization algorithm might possibly be able to the improve accuracy rate in this case. Even though, IPT has a lot of potential, it is still only a research project and far from being perfect.

3.2.5 Smart Campus

Smart Campus[8] is yet another project that came up while researching for this project. While there is a dedicated website available for it and there is a documentation section, the documentation section leave out a lot of details for getting started with the project. The demo application did not work very well and crashed very often for no particular reason.

3.3 Conclusion

All the five project discussed above are excellent instances of IPS based on smartphone technologies. Although, some of these projects are yet to be perfected and requires a lot of further development, they all prove one simple fact and that is smartphones have evolved over the past few years. Smartphones are capable of much more than just keeping notes, reminders and making phone calls. While the Redpin project promises a room-level accuracy, the Indoor Position Tracker application shows us that it is possible to run complex algorithms from within the application by harnessing the processing power of smartphones. On the other hand, even though IndoorAtlas attempts resolve the IPS problem with a very unique approach, it shows us great possibility of a perfected indoor positioning system by combining two or more of the technologies used by these projects. It is just a matter of time before researchers figure out a way to accomplish that.

[7] For more information about Indoor Position Tracker visit: https://github.com/TeroM/indoor-position-tracker

[8] For more information about Smart Campus visit: http://smartcampus.cs.aau.dk/

Chapter 4

Analysis

4.1 Introduction

This section will start by following up from the Related Work section by explaining which of the aforementioned project has been chosen for the implementation, if any, and why others have not been chosen. Then the main objectives, detailed in the Introduction, will be reviewed. Afterwards, this section will also discuss about which of the technologies have chosen for the development of Campus Explorer and why.

4.2 Choosing An IPS Implementation

Now that all the advantages and disadvantages of the related projects and products have been discussed, it is time to ascertain whether any of the projects are to be used in the implementation of the Campus Explorer.

In the experimental runs, the Redpin implementation gave the most promising results. In fact, the Redpin was able to live up to the promise by its creators and it delivered a room level positioning. Roughly, eight or more out of ten times Redpin was able to classify. While the most promising results generally means that this is to be the choice of implementation, that is quite not the case here. Firstly, the MapView did not work quite well in the test runs. The map, even though was just an image, went blank very often. Below is a screenshot of that illustrates the scenario:

Figure 1: Redpin MapView crashed

Another problem with Redpin is that it requires a Java based server application. This goes directly against the primary ideas and goals of this project. Right from the beginning of this project and the project presentation, having a self-sufficient mobile application, that can both have offline map and pin point the user without the need of an external server application, has been prioritised. It has been attempted to combine Redpin server and client applications into a single mobile application, simply because the server application is based on Java and Android client application is capable of running Java codes as well. Both the client and server applications have way too many interfaces and quite a complicated implementation. After several unsuccessful attempts to combine them, it has been decided that it would be much easier to build the Campus Explorer from the scratch, only if an implementation of a fingerprint classification algorithm is used.

In contrary, IndoorAtlas being a commercial project, it did seem to be quite a promising product. That is only until it has been tested using their demo application. The demo application supports fingerprint data collection using only supported devices, or that is what the company claims. Still it has been attempted to try out the application using the primary test device, the HTC One, and it was stuck in the sensor calibration phase. The system has also been tested using one of the (supposedly) supported devices, a Google Nexus 7 tablet. In this case, the application successfully calibrated sensors and it was also possible to collect fingerprint data using the Nexus 7. Unfortunately, the system was not able to locate the user most of the time, much like the Indoor Position Tracker application. Another reason for IndoorAtlas not being selected is that the client application is practically useless without an active Internet connection. Not to mention, IndoorAtlas is not open source and the positioning techniques are more or less kept unpublished by the company, possibly as a trade secret.

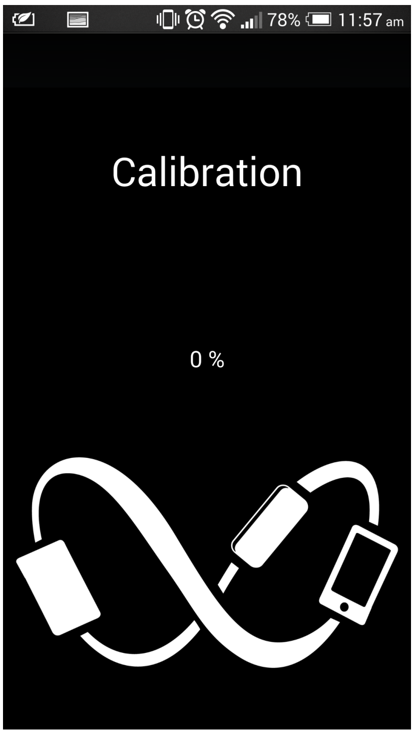


Figure 2: IndoorAtlas stuck at sensor calibration

Considering all the advantages and disadvantages, finally it has been decided to use the JavaML library for the implementation of fingerprint classification algorithm. The JavaML library is a great collection of implementation of Machine Learning algorithms. More about it is discussed later in the 6.3 Third party libraries section.

4.3 Choice of technologies And Development Plan

4.3.1 Three-Tier Application

After all the experimentation with existing solutions and considering the positive and negative sides, it has been decided to build the Campus Explorer application from the scratch. The prototype of the Campus Explorer will be developed as a three-tier client-server application. There will be a cloud database (tier-1) accessible through a web service (tier-2); which will be developed using PHP and MySQL. The web service will be developed as RESTFUL service and will output data in JSON format. The Campus Explorer Android application (tier-3) will be the client of this web service.

4.3.2 Local Data Storage

Even though the primary goal is to store all the fingerprint data and map data locally, for this prototype, all the data will be primarily stored in the cloud and there is a very solid reason for it. Storing all the fingerprint data centrally is required until all the fingerprint data has been collected and stored into the database. Once, all data is collected, it will be possible to extend the application so that the user can download all the required fingerprint data and store it locally, preferably in a database where possible, otherwise using text files. Since, Android has built-in support for SQLite database, it will be an easy task to accomplish that. This part will be for the further development segment of the development process of Campus Explorer.

4.3.3 Mapping

For similar reason to storing fingerprint data in the cloud, all the map data will be stored and used from the cloud as well. Google Maps Engine will be used to develop an indoor map, which then will be exported as independent KML files. The use of KML files will help keep the map data private and confidential. OpenStreetMap will be used as the map image server. The reason for the use of OpenStreetMap is that it is free to use and does not require any licensing or creation of an account. More importantly, OpenStreetMap allows downloading the map images. Therefore, it will be possible to download images of a certain map area and serve the map from an independent cloud service or even allow the user to download the entire map of the subject area into their smartphone.

OSMDroid library (along with the OSMDroidBonusPack, an extension to the OSMDroid) will be used to visualize the map using KML layer on top of the maps served by OpenStreetMap.

4.3.4 Routing And Navigation

Since, the indoor map (the KML layer) is more likely to be private and be unavailable from any public map server, the routing and navigation system will have to be developed as well. Therefore, a custom route manager will be developed using on the implementations of the Dijkstra's shortest path algorithm. This route manger will developed as an independent library or a separate Java package and then will be used alongside the OSMDroid library.

4.3.5 Fingerprint Algorithm

The fingerprint algorithm will be implemented using an open source library called JavaML, which provides implementations of various Machine Learning algorithms and supports Support Vector Machine (SVM) as well.

4.4 Conclusion

Now that all the required technologies have been selected, now is the turn to design and develop the application. In the following sections the design, implementation and evaluation of the system will be discussed.

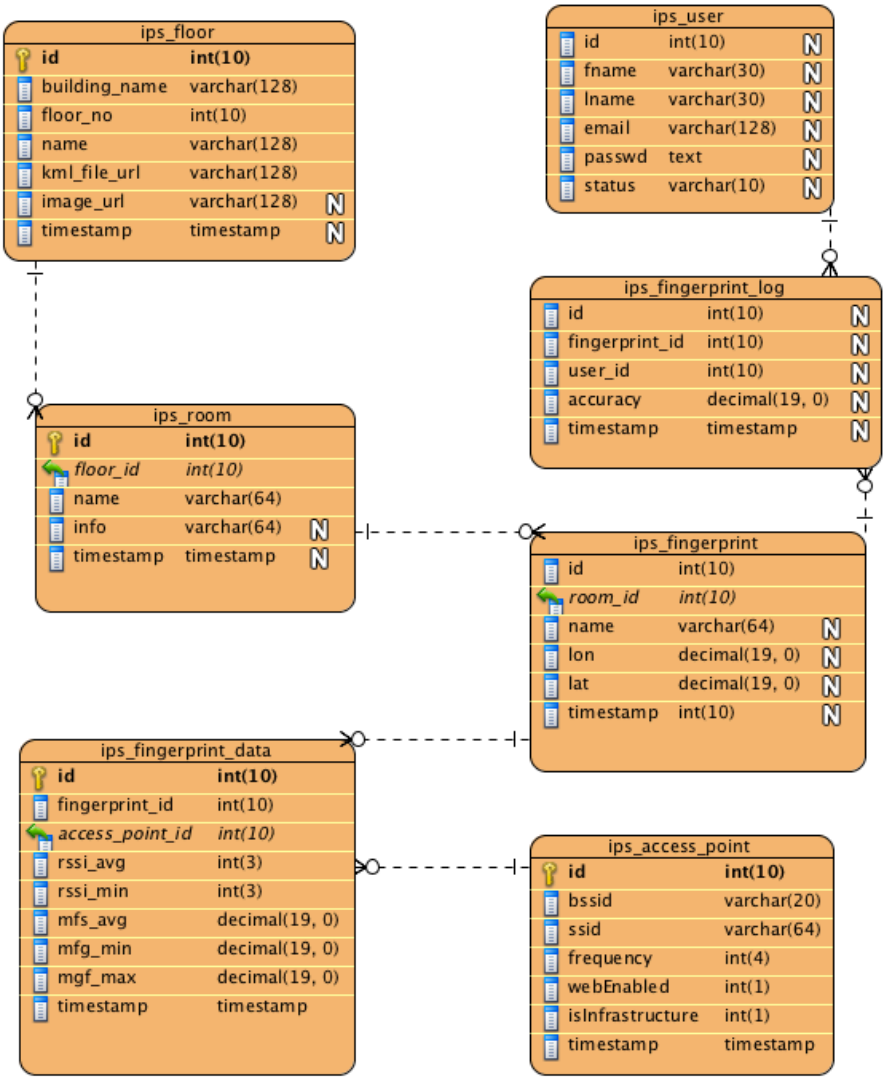
Chapter 5

Design Of Campus Explorer

The design of the Campus Explorer consists of two sections: database design and software design. Due to the nature of this project being an experimental project, all the design has been kept to this minimum and absolute necessary. This helps adapt to the changes.

Database design

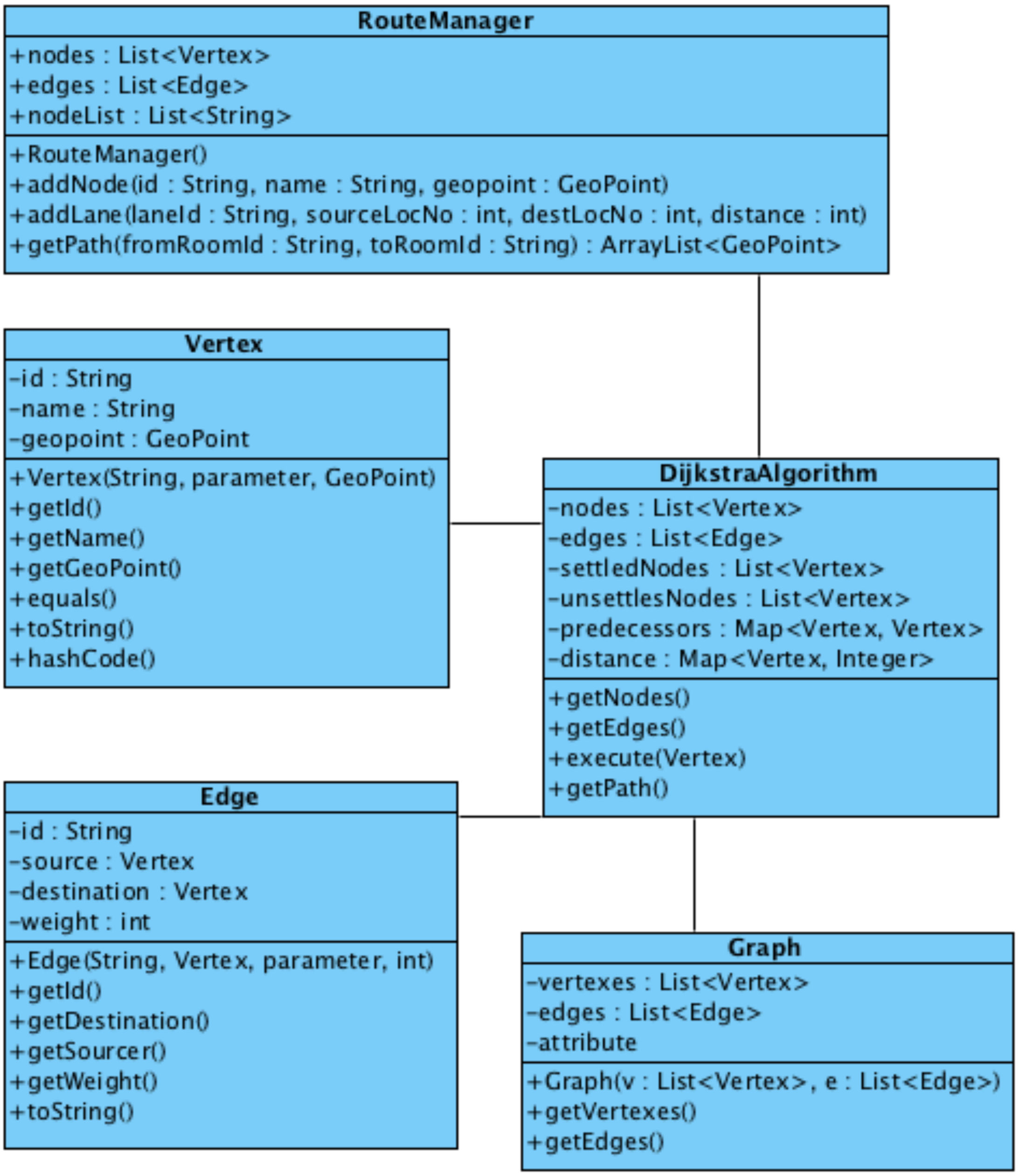
The design of the database for Campus Explorer is the most prominent aspect of the design section. The design of the database mainly focuses on building the fingerprint database. The implementation of the localization algorithm is critical to the design of the fingerprint database. Below is the design of the database for Campus Explorer:



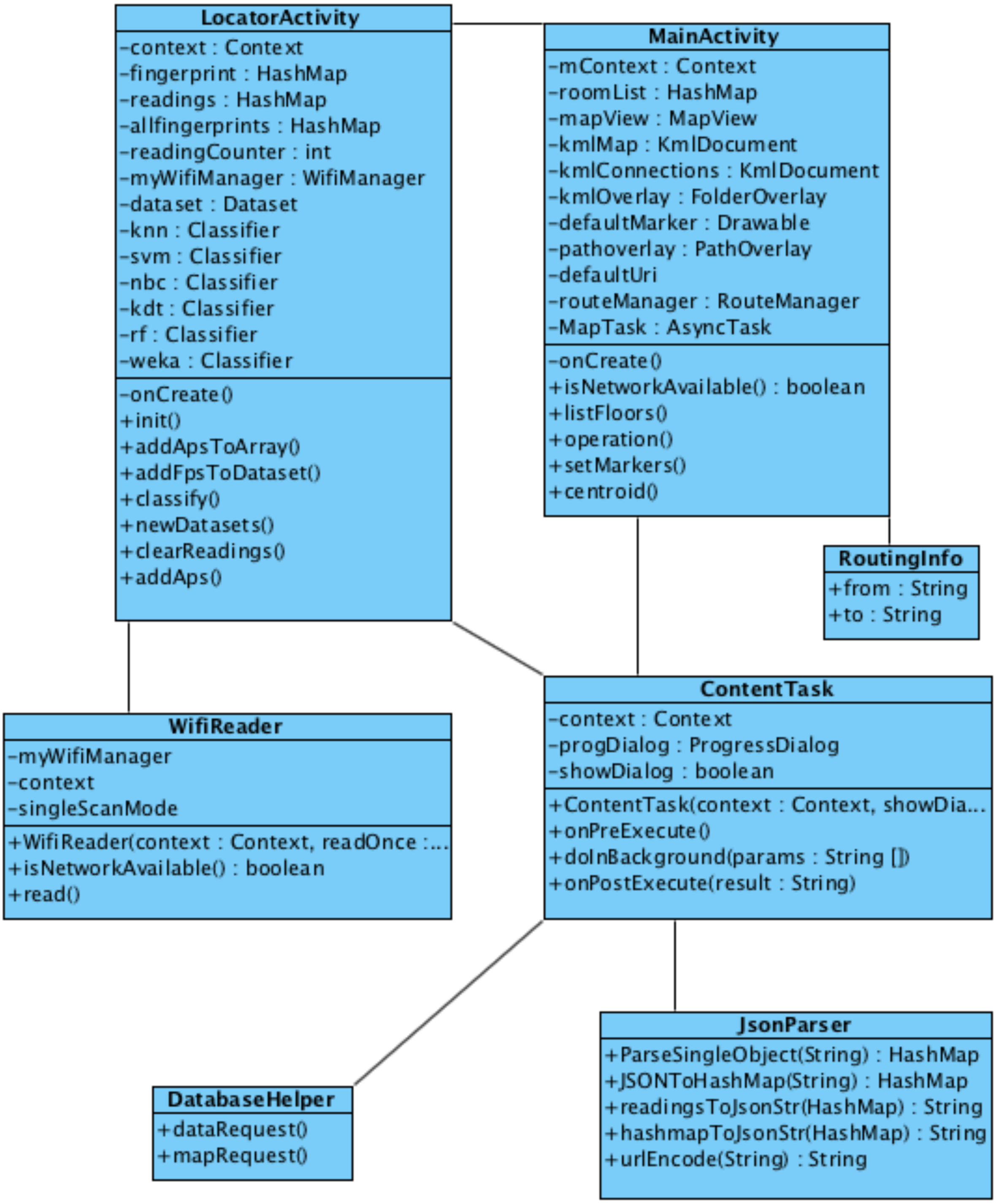
Software Design

The software design of the Campus Explorer consists of two main sections: the design of the custom route manager and the rest of the application.

The custom route manager is an implementation of the Dijkstra's algorithm. It has been implemented in an independent package (called com.toufiqurrahaman.dijkstra) using an online tutorial (more details about it in the Development of Campus Explorer section)



The rest of application consists of three Java packages: com.toufiqurrahaman.ce, com.toufiqurrahaman.ce.data and com.toufiqurrahaman.ce.task. The first package is the main application package that contains all the activities and other relevant classes. The second package is for data related classes and the third one is for background process (AsyncTask) for Internet connectivity purposes. Below is the conceptual class diagram for all the three packages combined:



Chapter 6

Development of Campus Explorer

6.1 Introduction

The Campus Explorer application has been developed following the Software Prototyping method. The application has been broken down into two major sections, as planned: map development and implementation of fingerprint algorithm. Rest of this section will walk through the step-by-step implementation of these sections.

6.2 Map Development

The map development section consists of two sub-sections: building an indoor map and development of the navigation system. Both of these sections have been developed using iterative prototyping method.

6.2.1 Building The Indoor Map

The indoor map has been developed using the Google Maps Engine. For this prototype application the first floor of the Greenwich campus of the University of Greenwich has been used as the subject location. In order to maintain confidentiality, only publicly available areas have been used and room names have been masked by an alias. By drawing an extra layer of indoor rooms, on top of the Google Maps, the indoor map has been developed. This layer of indoor map is then exported and downloaded as a single KML file and then stored in the cloud for application use. For demonstration purposes only the first floor has been used, otherwise, indoor map for each floor would be created individually and exported as separate KML files. The screenshot below illustrates how the indoor map has been developed:

The grey lines are the boundaries of the indoor locations (rooms, toilets etc.) and the bubbles implicate the entrances and exits of the buildings.

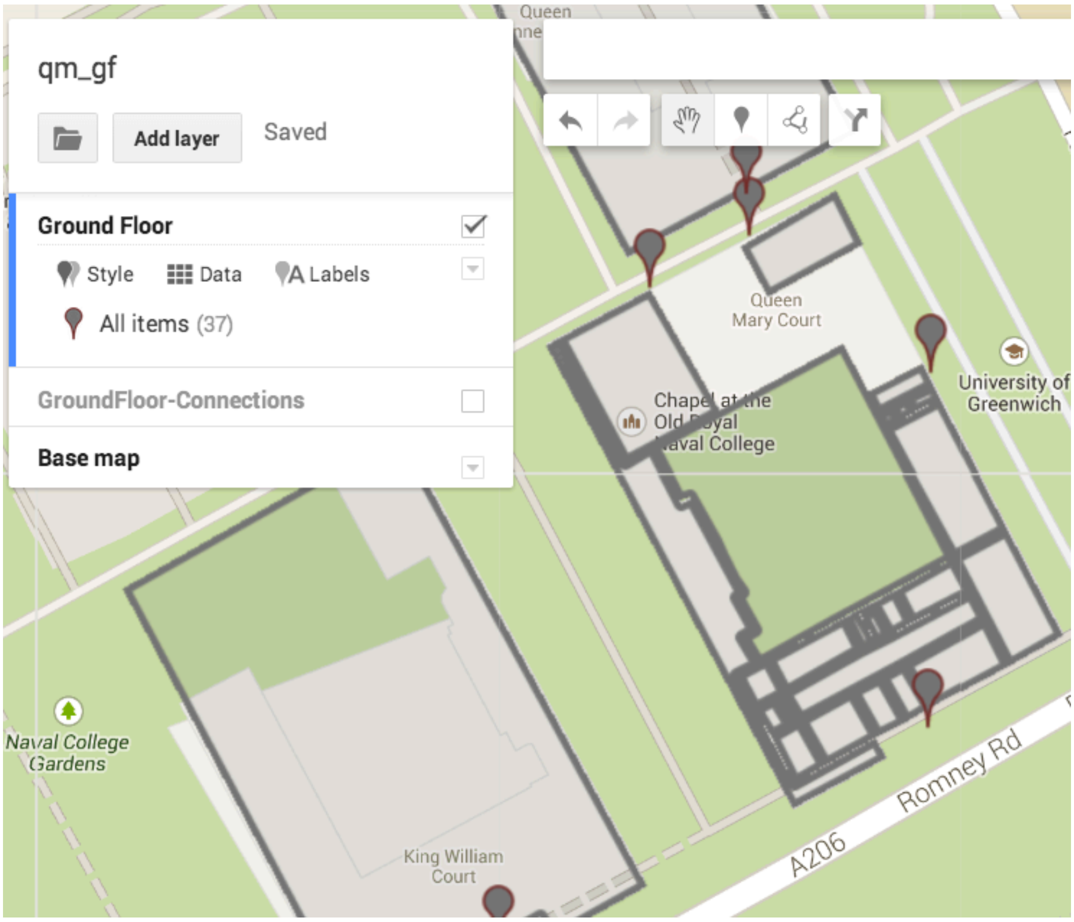


Figure 3: indoor map being developed

Exporting the map as KML file:

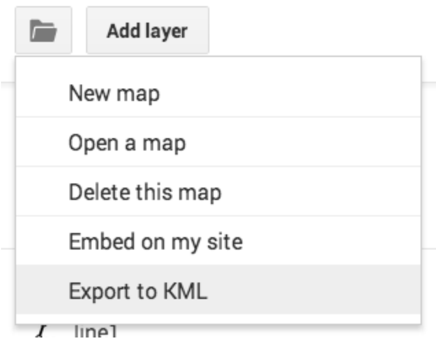


Figure 4: exporting the map layer as a KML file

Screenshots of the implementation of the map in the Campus Explorer application will be listed in the

6.4 Screenshots section.

6.2.2 Development Of Navigation System

6.2.2.1 Route Plan

In order to develop a routing and navigation system, a route plan for all rooms and point of interests needed to be developed. This is crucial to implementation of the routing system. The route plan has been developed using similar method to the development of the indoor map. Is this case, another extra layer has been created on top of the indoor map by drawing line connecting all the rooms, footpaths and entrances. The layer containing all the connections is then exported and downloaded as an individual KML file and then stored into the cloud as well. The screenshot below illustrates the process:

The black lines indicate the connection of the rooms. Note that, a node is created at the centre of each of the rooms and named after the room name.

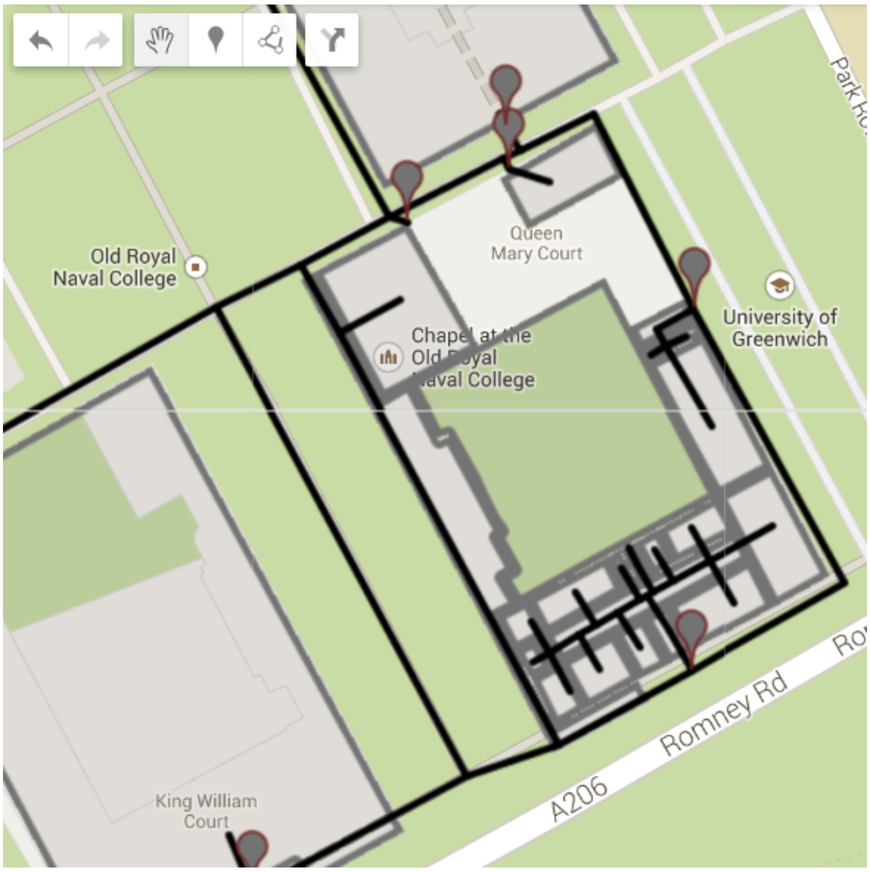


Figure 5: route plan is being created by drawing lines (connections) between places

6.2.2.2 Dijkstra's Shortest Path Algorithm

In order to make use of the route plan developed earlier, implementation of an algorithm is required that can create a route between two chosen locations and find the shortest path between them. In order to achieve this, an implementation of Dijkstra's Shortest Path algorithm has been used (details about the source of the code is listed later in the 6.3 Third party libraries section). By extending the implementation of this algorithm, a custom RouteManager has been developed. For simplification of the development, it has been assumed that all of the connections and pathways to the places are open for both way walking.

Using RouteManager

When implementing the RouteManager first an instance is created:

```
RouteManager routeManager = new RouteManager();
```

Then each of the nodes and lanes (connection between nodes) is added to the routeManager, one by one:

```
routeManager.addNode(id, name, geopoint);
routeManager.addLane("Lane_name", "uniqueId_from", "uniqueId_to", distance);
```

Since, all the connections are two-way pathways, two lanes are added for each connection by swapping the 'from' and 'to' nodes.

Once all the nodes and lanes are added for the each of the connections, the route manager can be used to find the shortest path between two nodes. Since, the centre of each of the room is a node as well, providing the room names of the starting room (or any other place) and destination room is necessary. To get the shortest path between two places the getPath() method is called by providing the node names of the start and destination nodes; which then returns a list of GeoPoints:

```
ArrayList<GeoPoint> waypoints = routeManager.getPath("from", "to");
```

The size of the waypoints list is zero means that there is no connection between two locations has been defined. Similarly, if the getPath() method returns null, then it means no connection has been found. Note that, all the calculations for finding the route are done locally, within the application. Therefore, all the results are instantaneous. Screenshots of the implementation of the RouteManager in the Campus Explorer application will be listed in the

6.4 Screenshots section.

6.3 Implementation Of Fingerprint Algorithm

The fingerprint algorithm has been implemented using the JavaML library. Among the many supported algorithms, the following six algorithms have been implemented and tested in the Campus Explorer.

1. K-Nearest Neighbour
2. Support Vector Machine
3. K-D Tree with K-Nearest Neighbour
4. Naïve Bayes Classifier
5. Weka Classifier
6. Rain Forest Classifier

Screenshots of the implantation will be enlisted later in this section.

6.4 Screenshots

6.4.1 Application at start-up (MainActivity.java)

The map data (KML files) is downloaded from the cloud at start-up and then the:

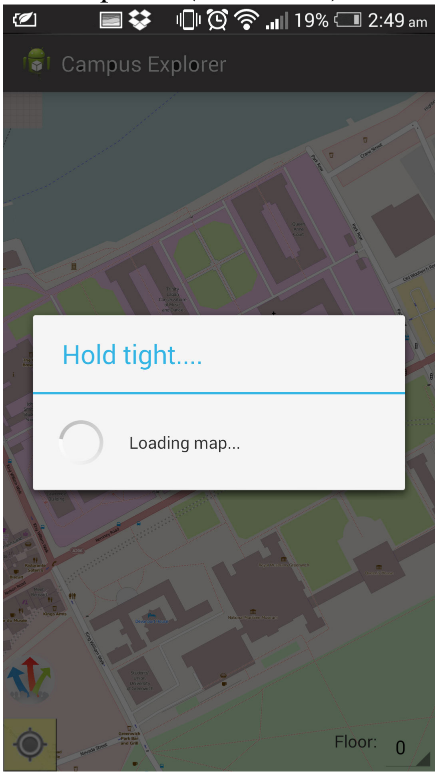


Figure 6: map data is being loaded

Figure 7: indoor map with POI marker for each location is displayed

6.4.2. Finding route between two locations:

Below are the screenshots illustrating the use of the RouteManager in the Campus Explorer application:

1. Choose starting point:

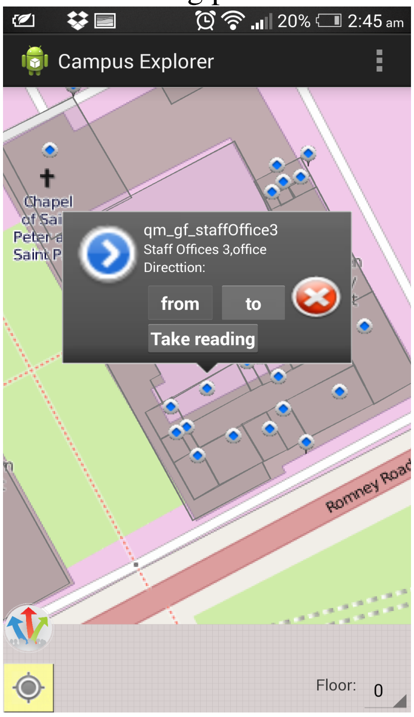


Figure 8: user selects a starting point by clicking on the POI marker

2. Choose destination point:

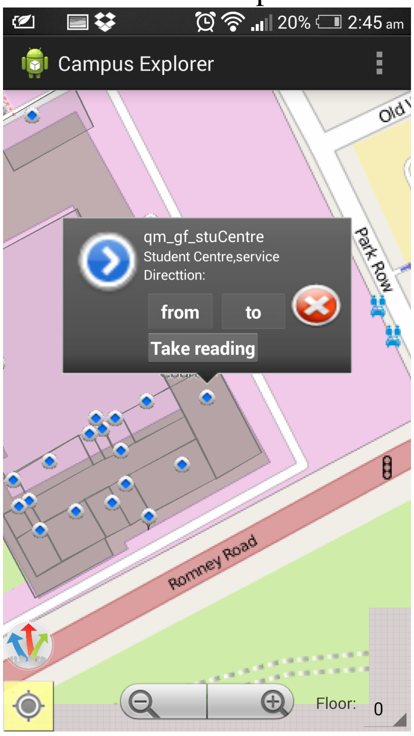


Figure 9: User selects destination point

3. Find route by clicking the find route button:

Figure 10: Red line indicates the route between the pre-selected locations

6.4.3 Fingerprinting (LocatorActivity.Java)

Collecting fingerprint data and uploading to the cloud:

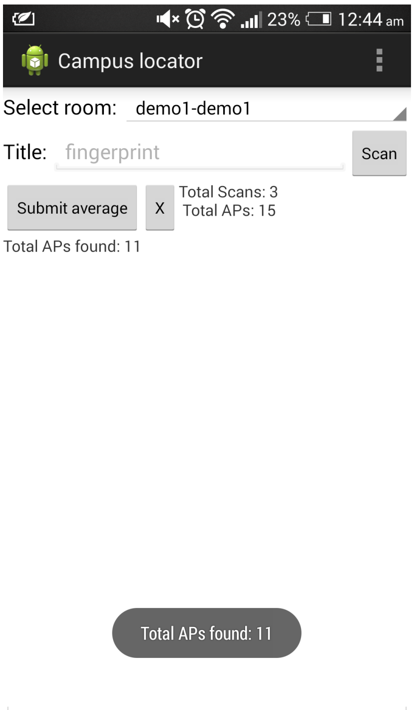


Figure 12: Scanning for Wi-Fi access points

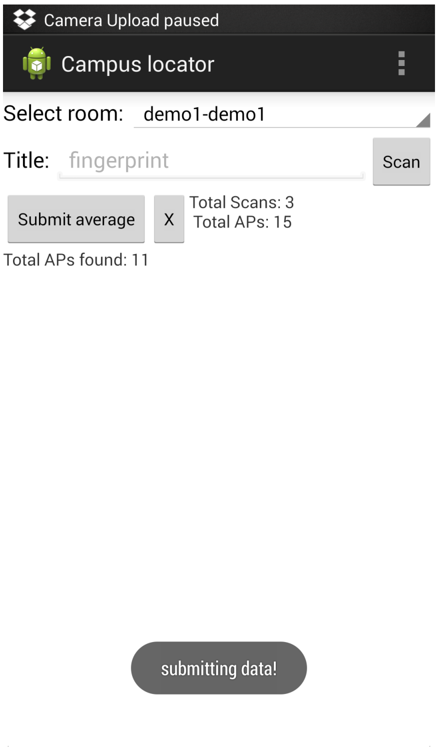


Figure 11: fingerprint data from the average of multiple scans is being uploaded

6.2 Development tools

In the development of Campus Explorer, many hardware and software tools have been used. Below is a brief discussion about the tools and how these tools have benefited the development process.

6.2.1 Software Tools

6.2.1.1 Eclipse

The Campus Explorer has been developed using the popular Eclipse IDE. Eclipse provides interface for developing in various different programming languages. It supports Java by default. For any other supported languages, one or more separate plugins are required to be installed. Since, Android™ is a framework base on Java and Java comes pre-packed with Eclipse by default, therefore, in order to enable support for Android™ [9], Google's™ [10] Android™ Development Toolkit (ADT) plugin was incorporated with Eclipse. This enabled development as well as compilation and debugging of applications based on Android™ platform using Eclipse.

6.2.1.2 GenyMotion

Even though, ADT plugin comes equipped with emulators for all the latest versions of Android™ , the emulator is quite frankly very slow. Therefore, a third party emulator called GenyMotion[11] has been used which several times faster than default emulators. In fact, in the primary development machine, the start-up time for Android™ is only 12 seconds compared to over 120 seconds for default emulator.

6.2.1.3 Google Maps Engine

Google Maps Engine[12] is an online tool that provides with an interface to create overlays on top of Google Maps. This tool has been used to create indoor map and indoor route map for the Campus Explorer application. The indoor map was then exported to KML format, which then was used with OSMDroid library.

6.2.1.4 Visual Paradigm

The Visual Paradigm application has been used for the design and development of database for Campus Explorer. It also has been used for designing the software as well. It provides a nice interface for creating and managing designs and has an excellent feature to generate the SQL queries for the creation and management of the database.

[9] Android is the registered Trademark of Google Inc., USA

[10] Google is the registered Trademark of Google Inc., USA

[11] GenyMotion is a third party Android™ emulator that can be downloaded free of charge for personal usage only. http://www.genymotion.com/

[12] Google Maps Engine can be accessed online, using Google account, here: http://mapsengine.google.com/

6.2.2 Hardware Tools

6.2.2.1 Development Machine

The primary device has been used for the development of the Campus Explorer was a MacBook Pro with 2.4 GHz Core i5 processor and 8 GB RAM and a SSD drive.

6.2.2.2 Mobile Devices

Alongside the emulator, the program was tested and used on two mobile devices, both of which were based on Android OS:

1. HTC One (2013 model)
2. Samsung Galaxy Note 10.1 N8010 (2012 model)
3. Google Nexus 7 (borrowed from a friend)

6.3 Third party libraries

The campus explorer application makes use of various open source third party libraries. Below is a list of the libraries used:

6.3.1 OSMDroid

OSMDroid[13] is an open source and completely free to use library for developing map applications. It is a replacement for the Google's default MapView class and provides with various features including indoor mapping and support for third party map providers.

6.3.2 OSMDroidBonusPack

OSMDroidBonusPack[14] is an extension to the OSMDroid library, which extends the OSMDroid with various extra features including support for Keyhole Markup Language (KML).

[13] For more information about OSMDroid library and download, visit: https://code.google.com/p/osmdroid/

[14] For more information about OSMDroidBonusPack and download, visit: https://code.google.com/p/osmbonuspack/

6.3.3 JavaML

Java Machine Learning Library (JavaML[15]) is an Open Source library that provides with the implementation of various Machine Learning algorithms. It also comes pre-packed with various libraries for some of these algorithm implementations, such as: Weka Classifier, LibSVM etc.

6.3.4 Dijkstra's Shortest Path Algorithm

For routing between indoor locations a custom route manager library has been developed which has been developed by extending the implementation of the algorithm from an article called 'Dijkstra's shortest path algorithm in Java - Tutorial'[16] by Lars Vogel published in November 2009.

[15] For more information about JavaML and download, visit: http://java-ml.sourceforge.net/

[16] The article 'Dijkstra's shortest path algorithm in Java – Tutorial' can be accessed online at http://www.vogella.com/tutorials/JavaAlgorithmsDijkstra/article.html

Chapter 7

Evaluation

The final prototype has been developed and majority of the objectives have been implemented. Some of the objectives have been left out, due to time limitation, as a scope for further development, such as: offline maps and local fingerprint data storage. The application has been designed and developed in a way so that these features can be implemented later on. Below is a brief reflective discussion on major sections of the application.

Map

The mapping system including the route manager works just as intended. The KML files can be programmed to load from local storage and the map data can be programmed do to the same as well. The route manager, the extended implementation of Dijkstra's Shortest Path algorithm, works as intended. Since all the execution runs locally, routing calculation happens almost instantaneously.

Fingerprint Classification Algorithm

While the fingerprint classification algorithms runs and classifies just in a fraction of time, the implemented algorithms do seemed to be less accurate individually. Therefore, it had been attempted to create a voting system where the location voted by majority of the algorithms is to be selected. Doing so the accuracy rate did seem to improve but it was far from usable state. This is due to the fact that Wi-Fi signal strengths fluctuate in every single scan. Even the number of received Access Points (AP) varies largely. In the test runs, the different between the first run and average of multiple runs was large both in terms of RSSI values and number of Aps. By improving the algorithm, the accuracy rate can be improved.

How can algorithm be improved?

One of the main reasons that the algorithms were not as successful in classification as intended was that these algorithms have not been optimized in their standards forms. Li and Zhang also confirms this:

"The K-nearest-neighbor fingerprinting algorithm uses a fixed number of neighbors, which reduces positioning accuracy. Here, we propose a novel fingerprinting algorithm, the enhanced weighted K-nearest neighbor (EWKNN) algorithm, which improves accuracy by changing the number of considered neighbors." (Lin and Zhang et al., 2009)
An algorithm will supposedly work as intended, in an IPS based on Wi-Fi fingerprinting, if a custom algorithm is developed. There are various considerations that need to be added when developing a custom algorithm. Below is a list of major facts that requires consideration while developing a custom algorithm:

1. Fluctuation of RSSI values
2. Higher RSSI value versus number of total Aps (Weighted KNN)
3. Exactly how many neighbours are to be considered?
4. Issues relating to APs with 2.4GHz Wi-Fi frequency

Chapter 8

Ethical Issues And Considerations

There are several things about this application that might need legal and social consideration. Due to the fact that the application needs to be tested on a subject area, supposedly whole or part of a building the University of Greenwich, it might be necessary to obtain legal approval from the university. Furthermore, all the map data and the structural information of the subject building will be available through the application for the end user if the application is made available for public use. This might be an issue if the university chooses to disapprove it.

The application will require at least one person to for testing purposes. Assuming testers will be among the supervisor, tutor and students there will be no requirements for clearance from the Research Ethics Committee.

By using KML files the indoor map data is kept private to the application only. Therefore, private property confidentiality issues can be avoided.

Contrary to the legal issues, this application might have some social impact among the students if the application is made available for them. This application will particularly be helpful for new students in finding the specific classrooms and possible for any visitor as well. It is assumed that there will be no negative social impacts.

Chapter 9

Conclusion

9.1 Development Methodology

As with any other software development projects, the development methodology and planning are the keys to project success. The use of wrong development methodology along can lead to a disastrous results and even failure of a project. After completing the project it can be said that in terms of choosing the right methodology; the chosen method, Software Prototyping, was very effective in the development of the project.

9.2 Project

The ultimate goal of this project was to investigate and prove that IPS systems can be developed using existing smartphone technologies and developing a self-sufficient mobile application. Considering this, the project has been completed with satisfactory results. The map has been developed in a way that it can be easily extended. The application can be further developed to enable compatibility of using locally stored maps. The application can be easily programmed to store the fingerprint data locally. What is more fascinating is that all the complex fingerprint classification and routing calculations we run locally within the application itself. Therefore, the need of an external server has been eliminated. It is impressive that even with all these complex calculations and the loading memory heavy map data, the application ran smoothly without crashes. While the algorithm needs to be improved, the development of a custom Wi-Fi fingerprint classification algorithm it self can be a much larger project that this project! Therefore, it can be said that the improvement of the algorithm resides in the further development process of the application.

Appendix A

References

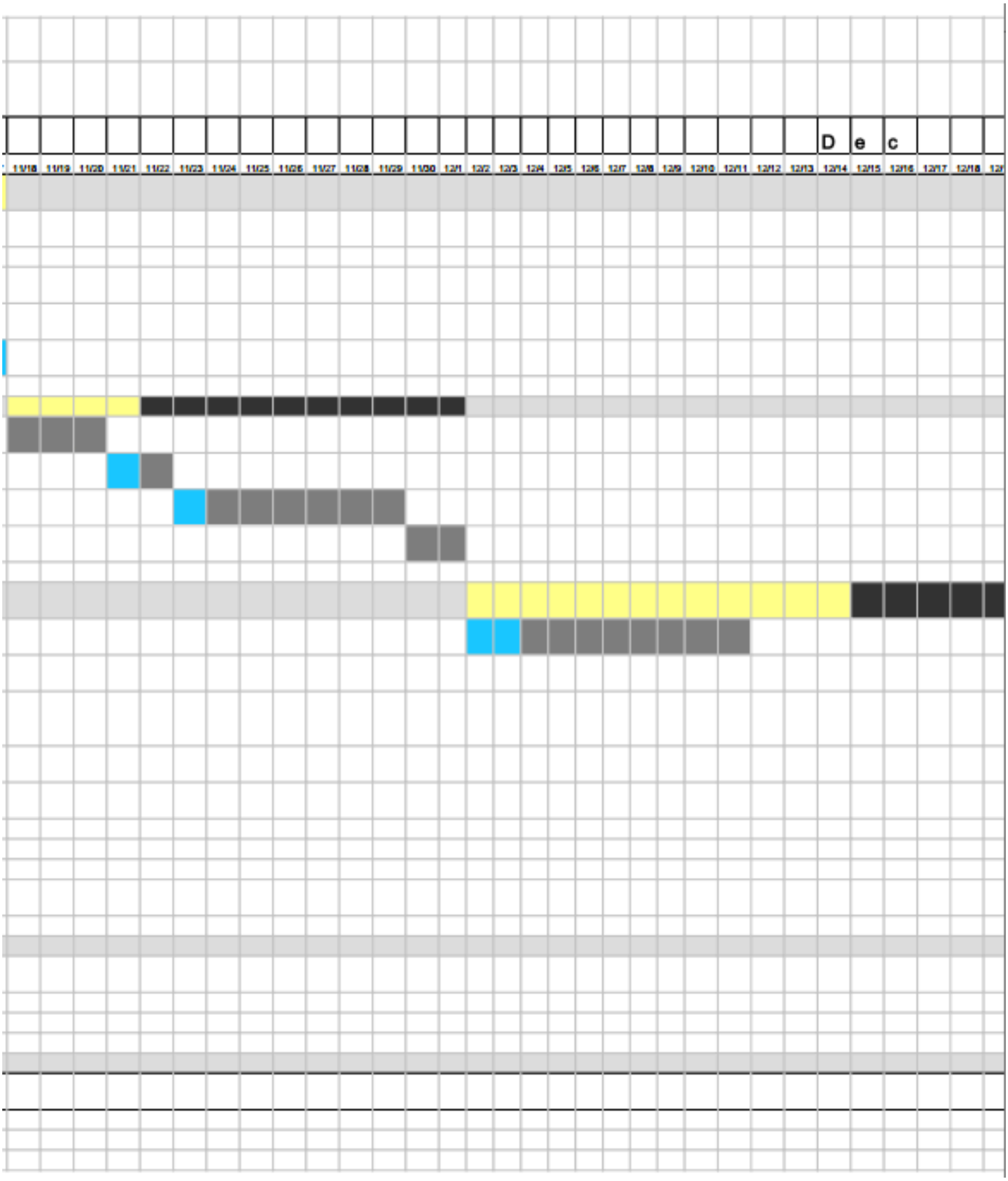
1. Abernethy, M., 2010. *Data mining with WEKA, Part 3: Nearest Neighbor and server-side library*. [online] Ibm.com. Available at: <http://www.ibm.com/developerworks/opensource/library/os-weka3/index.html>.
2. Abernethy, M., 2010. *Data mining with WEKA, Part 1: Introduction and regression*. [online] Ibm.com. Available at: <http://www.ibm.com/developerworks/opensource/library/os-weka1/index.html>.
3. Abernethy, M., 2010. *Data mining with WEKA, Part 2: Classification and clustering*. [online] Ibm.com. Available at: <http://www.ibm.com/developerworks/opensource/library/os-weka2/index.html>.
4. Alsehly, F., Arslan, T. and Sevak, Z., 2011. Indoor positioning with floor determination in multi story buildings. pp.1--7.
5. Anthony, S., 2012. *Indoor navigation on your smartphone, using the Earth's magnetic field — just like a pigeon | ExtremeTech*. [online] ExtremeTech. Available at: <http://www.extremetech.com/computing/132484-indoor-navigation-on-your-smartphone-using-the-earths-magnetic-field-just-like-a-homing-pigeon>.
6. Bilke, A. and Sieck, J., 2013. Using the Magnetic Field for Indoor Localisation on a Mobile Phone.*Springer*, pp.195--208.
7. Bolliger, P., 2008. Redpin-adaptive, zero-configuration indoor localization through user collaboration. pp.55--60.
8. Cavoukian, A. and Cameron, K., 2012. *Issues Involving the Unforeseen Uses of Pre-existing Architecture*. [online] Available at: <http://www.ipc.on.ca/images/Resources/wi-fi.pdf>.
9. Chen, L., Li, B., Zhao, K., Rizos, C. and Zheng, Z., 2013. An improved algorithm to generate a Wi-Fi fingerprint database for indoor positioning. *Sensors*, 13(8), pp.11085--11096.
10. Curran, K., 2012. Indoor Location Technologies.
11. Del Mundo, L. and Macatangga, R., 2012. Hybrid classifier for Wi-Fi fingerprinting system. pp.107--112.
12. Deng, Z., Yu, Y., Yuan, X., Wan, N. and Yang, L., 2013. Situation and development tendency of indoor positioning. *Communications, China*, 10(3), pp.42--55.
13. GOETZ, M. and ZIPF, A., 2012. Mapping the Indoor World. *GIM International*, pp.30-34.
14. Jekabsons, G., Kairish, V. and Zuravlyov, V., 2011. An Analysis of Wi-Fi Based Indoor Positioning Accuracy. *Scientific Journal of Riga Technical University. Computer Sciences*, 44(1), pp.131--137.
15. Kang, W., Nam, S., Han, Y. and Lee, S., 2012. Improved heading estimation for smartphone-based indoor positioning systems. pp.2449--2453.

16. Lee, J., Yoon, C., Park, H. and So, J., 2013. Analysis of Location Estimation Algorithms for Wifi Fingerprint-based Indoor Localization.
17. Li, B., Gallagher, T., Dempster, A. and Rizos, C., 2012. How feasible is the use of magnetic field alone for indoor positioning?. pp.1--9.
18. Lin, H., Zhang, Y., Griss, M., L, and a, I., 2009. Enhanced indoor locationing in a congested Wi-Fi environment.
19. Link, J., Smith, P., Viol, N. and Wehrle, K., 2011. Footpath: Accurate map-based indoor navigation using smartphones. pp.1--8.
20. Liu, H., Gan, Y., Yang, J., Sidhom, S., Wang, Y., Chen, Y. and Ye, F., 2012. Push the limit of WiFi based localization for smartphones. pp.305--316.
21. Meier, R., 2009. *Professional Android application development*. 1st ed. Indianapolis, IN: Wiley.
22. Meier, R., 2012. *Professional Android 4 application development*. 1st ed. Indianapolis, IN.: Wiley/[Wrox].
23. Milette, G. and Stroud, A., 2012. *Professional Android sensor programming*. 1st ed. Indianapolis: John Wiley & Sons.
24. Quan, M., Navarro, E. and Peuker, B., 2010. Wi-Fi Localization Using RSSI Fingerprinting.
25. Rizos, C., Roberts, G., Barnes, J. and Gambale, N., 2010. Experimental results of Locata: A high accuracy indoor positioning system. pp.1--7.
26. Schutzberg, A., 2013. *Ten Things You Need to Know About Indoor Positioning - Directions Magazine*. [online] Directionsmag.com. Available at: <http://www.directionsmag.com/articles/10-things-you-need-to-know-about-indoor-positioning/324602>.
27. Shala, U. and Rodriguez, A., 2011. Indoor positioning using sensor-fusion in android devices.
28. Shala, U. and Rodriguez, A., 2011. Indoor positioning using sensor-fusion in android devices.
29. Shetty, A., 2010. Weighted K-nearest neighbor algorithm as an object localization technique using passive RFID tags.
30. Shin, B., Lee, J., Lee, T. and Kim, H., 2012. Enhanced weighted K-nearest neighbor algorithm for indoor Wi-Fi positioning systems. 2, pp.574--577.
31. Tian, Z., Tang, X., Zhou, M. and Tan, Z., 2013. Fingerprint indoor positioning algorithm based on affinity propagation clustering. *EURASIP Journal on Wireless Communications and Networking*, 2013(1), pp.1--8.
32. Vicent, J., 2013. WIFI indoor positioning for mobile devices, an application for the UJI Smart Campus.
33. Vogel, L., 2009. *Dijkstra's shortest path algorithm in Java - Tutorial*. [online] Vogella.com. Available at: <http://www.vogella.com/tutorials/JavaAlgorithmsDijkstra/article.htm>.
34. Woodman, O. and Harle, R., 2008. Pedestrian localisation for indoor environments. pp.114--123.

Appendix B

Appendix A - Schedule of Work

	IPS using existing smartphone technologies		Start Date:	November 1, 2013																		
	Toufiqur Chowdhury (000598288), BEng Software Engineering, University of Greenwich																					
	Task	Start Date	End Date	Duration (days)	Percent Complete														N	o	v	
	Date:					11/1	11/2	11/3	11/4	11/5	11/6	11/7	11/8	11/9	11/10	11/11	11/12	11/13	11/14	11/15	11/16	11/17
17	1.0 Background research and idea development	2013-11-01	2013-11-17	17	100.00%																	
2	1.1 Search for and read relevant journals and publications	2013-11-01	2013-11-04	4	100.00%																	
	1.2 Read books	2013-11-05	2013-11-08	4	100.00%																	
	1.3 Search for and research existing solutions	2013-11-09	2013-11-12	4	100.00%																	
	1.4 Develop the minimum requirements for a MVP	2013-11-13	2013-11-15	3	100.00%																	
	1.5 Analyse the best suitable solution	2013-11-16	2013-11-17	2	100.00%																	
	2.0 Software design and planning	2013-11-18	2013-12-01	14	28.57%																	
	2.1 Design database schema and normalize to at least 2NF	2013-11-18	2013-11-20	3	25.00%																	
	2.2 Explore design considerations and software design patterns	2013-11-21	2013-11-22	2	50.00%																	
	2.3 Design diagrammatic schemas of software structure and model	2013-11-23	2013-11-29	7	25.00%																	
	2.4 Develop a test plan for the system	2013-11-30	2013-12-01	2	25.00%																	
	3.0 Minimum Viable Product (MVP) development	2013-12-02	2014-01-10	40	34.26%																	
	3.1 Experiment with existing solutions	2013-12-02	2013-12-11	10	25.00%																	
	3.2 Experiment with various fingerprinting algorithms	2013-12-23	2014-01-01	10	50.00%																	
	3.3 Implementation one or more the solutions and/or algorithms experimented earlier	2014-01-02	2014-01-08	7	25.00%																	
	3.4 Design and develop indoor map using suitable tools	2013-12-28	2013-12-30	3	25.00%																	
	3.5 Develop a routing system for the indoor map	2014-01-04	2014-01-10	7	25.00%																	
	3.6 Software testing	2014-01-11	2014-01-14	4	25.00%																	
	3.7 Documentation	2014-01-15	2014-01-18	4	25.00%																	
	3.8 Maintenance and debugging	2014-01-19	2014-01-25	7	25.00%																	
	3.9 Improvements and further development possibilities	2014-01-26	2014-02-01	7	25.00%																	
	5.0 Evaluation	2014-01-27	2014-01-31	5	25.00%																	
	5.1 Take fingerprint data from the subject location	2014-01-27	2014-01-29	3	60.00%																	
	5.2 Test positioning	2014-01-30	2014-01-30	1	20.00%																	
	5.2 Test map routing & navigation	2014-01-31	2014-01-31	1	20.00%																	
	6.0 Deployment	2014-02-01	2014-02-01	1	5.00%																	
	6.1 Compile application for demonstration	2014-02-01	2014-02-01	1	5.00%																	
	6.2 Report writing	2014-02-02	2014-03-18	45	95.00%																	


D e c
11/18 11/19 11/20 11/21 11/22 11/23 11/24 11/25 11/26 11/27 11/28 11/29 11/30 12/1 12/2 12/3 12/4 12/5 12/6 12/7 12/8 12/9 12/10 12/11 12/12 12/13 12/14 12/15 12/16 12/17 12/18