Islington College

BS.C (HONS) COMPUTER NETWORKING & IT SECURITY

Final Year Project

Wi-Fi Based Indoor Navigation System for Android Device.

Submitted by:

Name   : **Abhijeet Sapkota**

Year   : 3

ID   : 12065081

Group : L3N5

First Supervisor: Mr. Roshan Chaudhary

Second Supervisor: Mr Ashok Dhungana

Acknowledgement:

Abstract:

"Wi-Fi" a popular wireless networking technology, especially used for accessing internet and network connections. Beside this Wi-Fi technology has various other purposes. One of the system that can be developed using this technology is Wi-Fi based indoor navigation system. As GPS signal cannot be transmitted inside the building easily because of walls and other indoor objects, Wi-Fi signals are more ubiquitous and has higher signal strength than GPS satellite miles away from earth's surface. Every mobile device receives signal from Access Points (APs). The Received Signal Strength (RSS) values obtained from APs can be recorded in certain time interval and in different position. This technique is called Wireless Fingerprinting. Using this technique distance between an access point and mobile device can be determined.

For this project a Wi-Fi based Indoor Navigation System is developed for Islington College where students and other users inside the college premises can find their position using this system into their Android mobile. This system uses data received from mobile device's wireless adapter, compass and accelerometer to determine user's position. Firstly, a RSS logger application is developed using Android API which creates a file containing RSS values and for determining the position the recorded RSS values are compared with the real time RSS values received from the access point in mobile device. Position of the mobile device is displayed on map view user interface in the mobile.

Table of Contents:

Table of Figures:

1 Introduction:

1.1 Introduction of Subject Matter:

According to the research people spend 70 − 80% of their time in indoor environments such as shopping malls, hospitals, libraries, airports, universities/campuses (Strategy Analytics, 2014). These places has large buildings covering wide areas with different departments and sections and people new to this places (especially foreigners) may get lost. This aspect has made Location Based Services (LBS) a key interest to develop a location determining application to guide users throughout the building. Technologies advancement in robots has allowed them to locate their position in indoor areas like big warehouses in big factories/industries (Babu, 2014). But same application can be very useful for assisting people to determine their location in such indoor environment.

Early days in US GPS, (Global Positioning System) was developed especially for soldiers, airmen, sailors to find location during war. And in 1983 US Department of Defense allowed normal civilian to use satellite-based positioning system (Schneider, 2013). While using GPS in indoor environment, the system could not facilitate with accurate location because signal emitting from GPS satellite is unavailable or significantly weakened due to the attenuation or blockage of positioning signals by objects or interference with each other (Schutzberg, 2013). To solve this problem in indoor positioning existing Wi-Fi Access Points are used as a replacement of GPS satellite or in other words using wireless access points as "mini satellite" for navigation process.

Wi-Fi Access Points (AP) provides ubiquitous signal coverage inside buildings. Location can be determined at a room level by using Received Signal Strength (RSS) values extracted via continuous scanning of beacon packets transmitted by nearby APs. As today's smartphones are already equipped with WLAN adapters for wireless connections, only a software application is needed to monitor RSS values. The approach called Wireless/RSS fingerprinting is used for the development of indoor navigation system. This approach has two phases: i) Setup phase ii) Positioning phase.

In Setup phase, RSS fingerprints related with the respective AP are collected into a file or so called radiomap. Actually, radiomap is a mapping of multi-dimensional RSS values to physical coordinates (x, y). Later in positioning phase the currently received RSS fingerprints

are compared with the reference fingerprints in radiomap file and location of the user is determined associated with the matching fingerprint (Laoudias, et al., 2012). Like above described, if indoor navigation system is made using existing WLAN infrastructure time consumption and high installation costs of expensive equipment like custom transmitters and antennas are reduced.

1.2   Problem Domain:

Nepal is a developing country. Most of the technologies are yet to be implemented. Technology like Indoor Navigation is essential in places like hospitals, colleges, courts, supermarkets that helps those people who are new to these places. Although, there is a high trend of using smartphones with GPS technology installed in them they cannot be executed because of low bandwidth internet connection. 3G and GPRS are made available by different telecommunication companies but they cost more money. So, people do not use GPS for navigation in context of Nepal. Hence, navigation through available Wi-Fi signal inside building can bring a convenient way of finding the location and direction.

Concerning about the problem domain the best example would be a businessman, they have to travel all around the world for business. Where ever they go, either airports, train stations or different hotels for business conference, they will have hard time finding the location. And businessman would not want to waste their time in finding the conference room or ticket counter. So this system can help them save their time.

Most importantly Indoor Navigation could be used inside hospitals. There are different wards and rooms in hospital. In case of hospitals in Nepal, if anybody's relative is admitted into hospital, they often find difficulties finding the patient even if they know the room number. This system would help anyone visiting hospitals finding their related person.

1.3 Client Information:

1.3.1 Client Background:
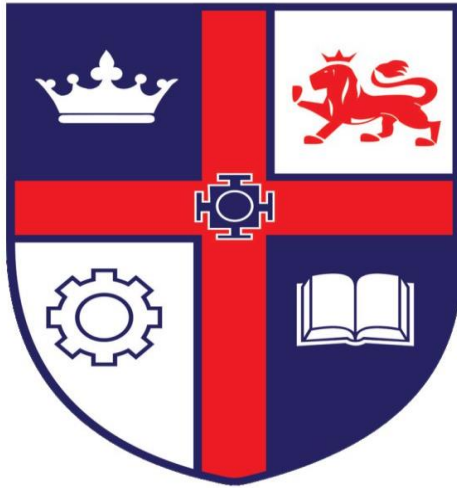
Islington College, formally known as Informatics College is a private education institution that provides IT and Business Education to Nepalese students. It is directly affiliated with London Metropolitan University and the first and only institution in Nepal that runs a UK university undergraduate program.

1.3.2 Client Perspective:

According to the IT Department head **"Sulabh Khanal",** Islington College has a big premises with lots of blocks and many classrooms and lecture rooms. Newly admitted students face hard time finding their study rooms. They have to ask college staff to search their rooms every time they have to attend classes in new class rooms. If Wi-Fi Based Indoor Navigation System for Android is developed, Students can easily find their respective classrooms and other locations inside college premises using their android devices. He also added that Islington College would be the first college in Nepal to use Wi-Fi Based Indoor Navigation System.

1.4 Aims and Objective:

The main aim of this project is to develop and android mobile application for Islington College that allows users to navigate themselves inside the college premises using their android smartphone by comparing real time Wi-Fi access points signal strength (RSS) and RSS fingerprint file.

Objectives are as follows:

- To provide an ease for college students and other people visiting college to navigate themselves inside college premises.
- To add a new facility in Islington College that has not been implemented in other college.
- To develop a system that has not been developed yet in Nepal.
- Research on the unique properties of pattern of the received signal strength in location fingerprints through extensive measurement process.
- To use all means of available resources to estimate accurate position of the mobile device as possible.

1.5 Report Structure:

This section describes how rest of the project is organized in this report.

2. Background:

This section provide the description of project background and context related to it. Contains the study of different article related to the project. Specify the client's perspective for the proposed system. This section provides the description of similar projects and their analysis. It also describes the technology used for the development of this project as well as provides information on relative projects.

3. Development:

Development part describes Methodology, Software Requirement Specification (SRS) and designing process for the project. Methodology depicts various approaches, tools and techniques used for the development of this project. SRS provides basic system overview, system architecture, and dependencies. Similarly at the end design part covers process flow diagram, context diagram and flow charts.

4. Testing, Result and Conclusion:

This section contains all the test process conducted during the implementation of the application in real scenario. Summary of the report describing the use of the system is found in conclusion part. And at the end mentioning some future task needed to be implemented in the system.

5. Social and Ethical Issues:

This section describes the legal issues that might arise after implementing this system. It clarifies that this system does not create any kind of social and ethical issues as no important data are required for the execution of the system.

6. References:

Lists of all the journals, books, websites and other resources are referenced in this section

2   Background/Context:

From day one, everything that could be done with a smartphone was possible without having the user install additional applications or programs. While old smartphones barely supported internet access, today's smartphones depends on internet for downloading all the applications. Higher needs and desires of users has led to customization and adaption of the large number of built in sensors to boost smartphone's application development.

The main goal of this project is to utilize available sensors of the smartphone to determine the location of the user without the use of additional hardware as required by GPS and some other sensor based approaches. Due to the lack of accurate positioning in the indoor environment multiple technique have been developed to solve the problem under different constraints. These techniques attempts to determine accurate positioning using Wi-Fi received signal strength (RSS), inertial sensors, etc.

This section describes the information channels considered for usages during this project as well as their functionality. Development platform and equipment used for this project is also described. Then a description of different coordinate systems used in this project will be presented.

2.1   Wi-Fi:

Wi-Fi is short for "Wireless Fidelity", a trademarked phrase that means IEEE 802.11 standard. Wi-Fi enabled device, such as laptops, smartphones, PDAs, uses radio waves to connect to the Internet via wireless network Access Point. (Beal, 2014)

Wireless network is just as same as two-way radio communication:

- Wireless adapter in Wi-Fi enabled devices convert data into radio signals and transmits it via an antenna.
- A wireless router then receives the signals form wireless devices and decodes it and then sends the information to the Internet using a physically wired, Ethernet connection.

A signal broadcasted by each wireless router is received by devices in the area. Every wireless devices have the capacity of measuring the signal strength and converts them into number known as Received Signal Strength Indicator (RSSI). Smartphones automatically preform this conversion to provide signal strength information to application running on them (Tisdale, 2014).

2.2 Sensors on Android:

There are three sensors commonly available in modern smartphones, *accelerometer, gyroscope* and *magnetometer*.

2.2.1 Accelerometer:

An accelerometer is a hardware sensor device used to detect a shake motion or measure proper acceleration of the device. Or in other words it is an instrument that measures time of rate of change in velocity with the respect to direction or magnitude (Francis, 2011). The accelerometer sensor of a smartphone allows it to measure the acceleration force in m/s^2 applied to it. Force of gravity is also included in acceleration measurement. (Aebi, 2012)
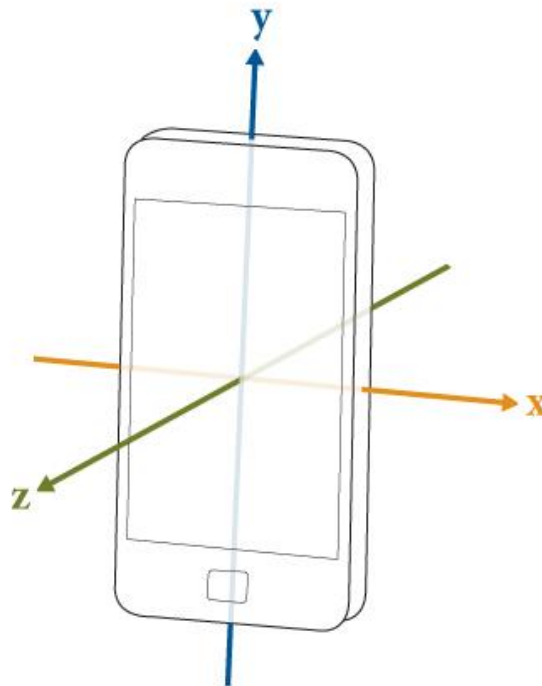
Figure 1: Co-ordinate system of smartphones. (Pacha, 2013 Sep 21)

Linear Acceleration:

A software based acceleration sensor that measures the rate of change of velocity as same as acceleration sensor but excluding the force of gravity. Therefore, linear acceleration can be defined as the rate of change in velocity of an object moving in one direction at given period of time. This sensor is used to count the step taken by user holding the device. (Karlsson & Karlsson, 2014)

2.2.2 Gyroscope:

In general Gyroscope is a device that measures orientation based on the basic principle of angular momentum. Figure 1 illustrate the mechanical gyroscope which contains a spinning rotor. (Wood,am, 2007)
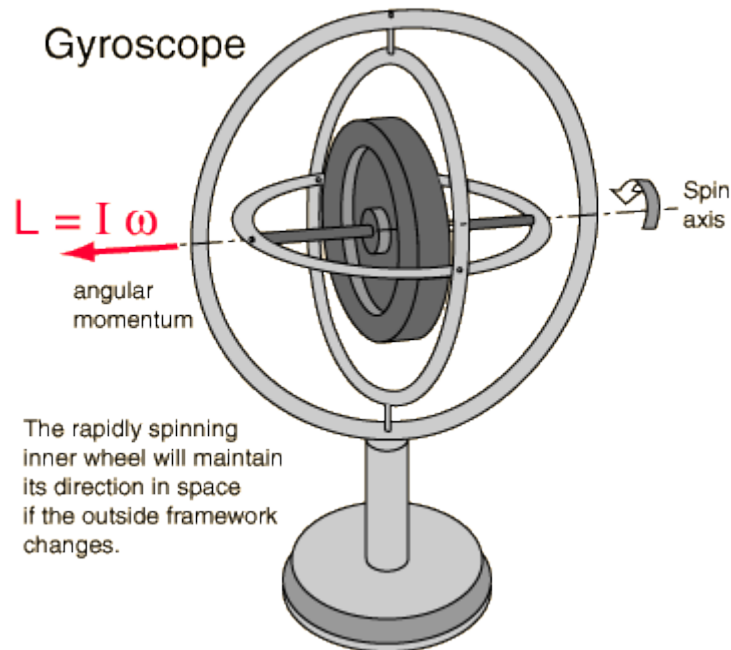
Figure 2: Mechanical Gyroscope

Figure 2 represents a MEMS (Micro Electro-Mechanical System) gyroscope from STMicroelectronics. It is the electronic gyroscope found in nowadays smartphones or tablets.
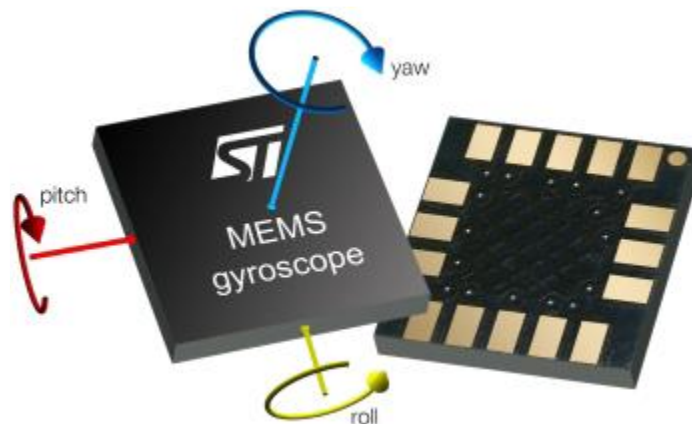
Figure 3: STMicroelectronics MEMS gyroscope

A gyroscope on an android powered device measures the rotation of a device around each of the three physical axis (x, y and z). It is used to obtain Azimuth, Pitch and Roll of the

smartphone and with Azimuth obtained a device can identify which direction the user is heading. (Robin, 2011)

2.3 Positioning Techniques:

2.3.1 Triangulation Estimation:

It is a trigonometric approach which determines unknown location based on two angles and distance between them. In case of sensor networks, two different reference nodes are needed to be located on a horizontal baseline for X-axis and two reference nodes are located at a vertical baseline for Y-axis. Then the distance between two reference nodes on the baseline is measured and stored in memory.
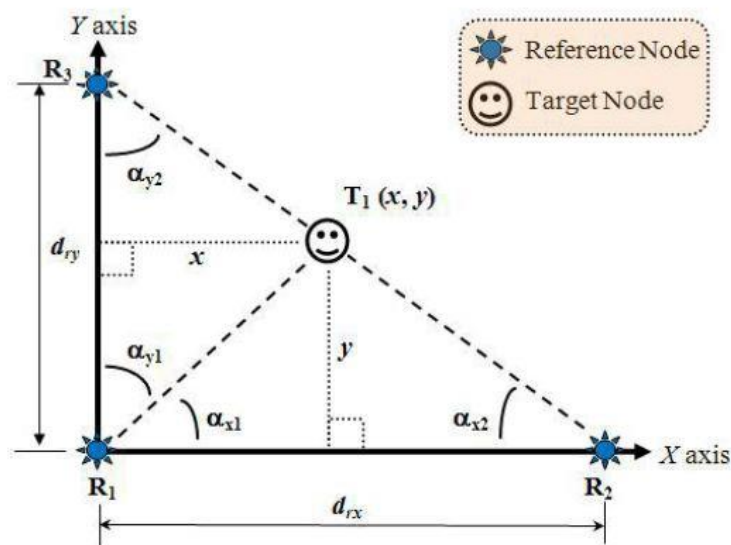
Figure 4: Location estimation based on triangulation estimation technique. (Pu, 2011)

As shown in Figure 3, two angles a_1 and a_2 are measured between baseline and the line that is formed by the reference node and the target. Reference node R1 and R3 from the baseline Y-axis, reference node R1 can be used to form the baseline of X-axis with the reference node R2. When the target node freely moves around the area, the location coordinate (x, y) of T1 target can be determined with the combination of R1 and R3 to determine x-coordinate and R1 and R2 to determine y-coordinate. (Baruch, 2011)

This approach is used in GPS satellites which rotate around earth in a precise orbit and transmit information signal to earth. A user device with GPS receiver receives the information from satellites and uses triangulation estimation to determine the exact location of the user.
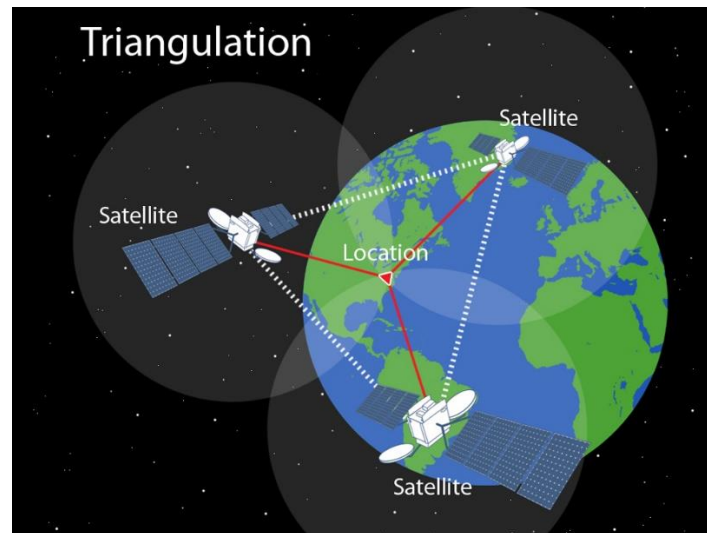
Figure 5: GPS Satellite nodes using Triangulation method. (National Geographic Society, 1996-2014)

Figure 4 shows the intersection point of the three GPS satellite which located the exact location of a GPS device on the earth surface. Triangulation method determines the location of an object by finding intersection point from three or more satellites. After measuring the angle of the known reference point triangulation is applied in the location process and estimates position of the GPS receiver marked by a red triangle on the figure. (Encyclopædia Britannica, Inc. , 2014 Feb 2014)

2.3.2 RSSI Similarity Matching Method:

The sensor nodes installed in the indoor environment has RSSI (Received Signal Strength Indicator) that evaluates the RSSI-based location estimation methods. As regards to the network infrastructure implementation, different wireless network sensors are configured to operate as a transmitter node and as a connection point between hardware components installed in the environment. A set of receivers having Radio Frequency Identification Device (RFID) tags, Bluetooth smart devices and Special Indoor Positioning and Communication System (IPCS) are arrayed throughout the coverage area to amount the RSSI values from the sensor nodes. Thus the RSSI values from receivers are sent to the server which hosts the algorithmic software that is capable of generating the target object's estimated location using the RSSI values.

In this project, the RSSI-based indoor localization system uses RSSI similarity matching approach which is done by measuring the RSSI values manually from each room and common area in the indoor environment thus, recording them to a file giving the associative

file-name to the rooms and common areas consequently sorting them onto a folder that makes up the RSSI sample map collection. Then, a database index is built for storing the vector index objects which represents the RSSI sample maps, and refining the speed of data retrieval operation while operating the location estimation process.

Location estimation process is evaluated by performing the matching operation between the vector indexes in the database index and the query vector built from the RSSI request of the target object. On contrary, the reference name of the vector index having the most alike element values to element values of the query vector will be selected as the name of the location where the target object is assessed to be.

AP's signals are triggered by the multipath contributions having occurred in indoor scenarios, like signal absorber or reflector building materials, humidity level and people blockage. Moreover, most of the existing wireless devices such as WLAN access points, ZigBee nodes are not designed to measure accurate RSSIs. (Zhu & Zheng, 2014)

2.4 Related Work:

2.4.1 WiFi-SLAM:

Currently developed Indoor Navigation System named "**WiFi-SLAM**" allows user's smartphone to locate its position in real time with accuracy of 2.5m only by reading Received Signal Strength (RSS) of Wi-Fi access points that are already installed inside the building. Instead of querying for the places or viewing wall maps, Indoor Positioning will provide greater ease to users for finding their location indoor. Developing a next generation location based mobile application will create a personal interaction to the step by step indoor navigation (WiFiSLAM, 2009).

WiFi-SLAM two different process to determine the location of a user Gaussian process to relate RSS fingerprints and human movement models as hidden variables. It analyses the signal strength of all the Wi-Fi access points along with the IDs of Wi-Fi networks. Then it matches the data to the reference data set either stored in the device or either accessed over the internet. This application also captures the human footsteps to make the location determination more accurate. Here one person walks around all the places in the building entering each room at least once. The algorithm developed to find the pattern of Wi-Fi fingerprints and footsteps for re-creating the pattern that the person covered in the whole building (Kuykendall, March 25, 2013).
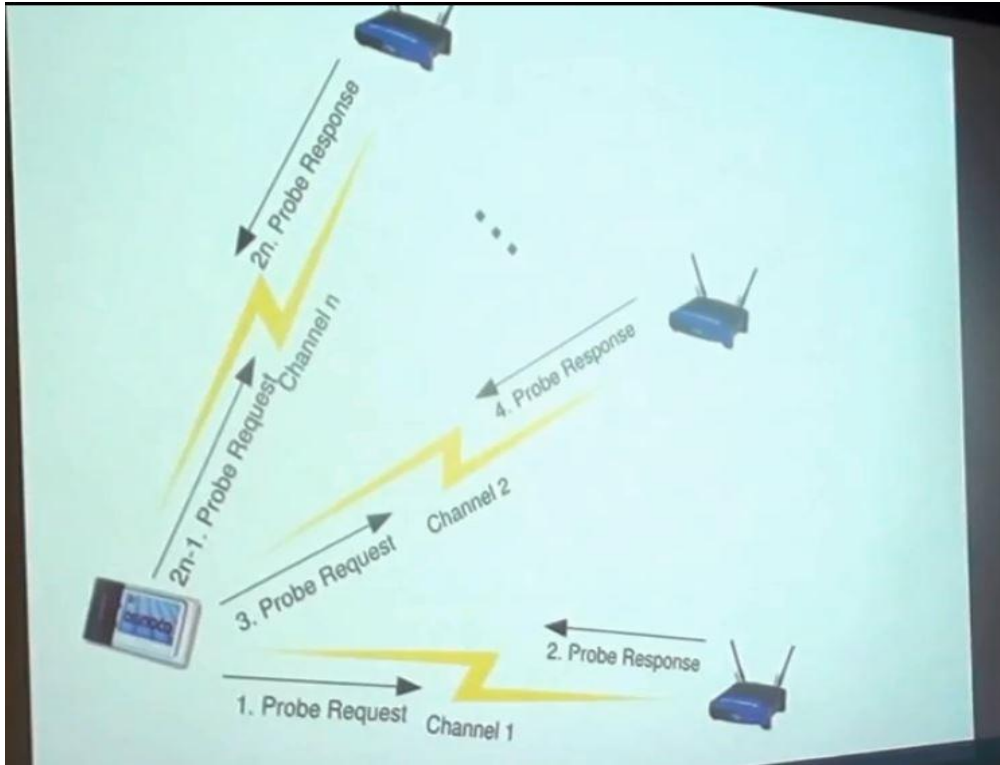
Figure 6: Time Difference of Arrival algorithm used by WIFI-SLAM (Burhum, Sep 14, 2012).

2.4.2 Non autonomous Indoor Navigation System

The first category of indoor navigation system is the non-autonomous, as the device running the indoor navigation system depends on the external factors i.e. technologies such as Wi-Fi, Bluetooth, etc. that are not residing inside the device, to compute the user's position.

iDocent is an application that uses this approach, which uses Wi-Fi triangulation to calculate the position of the device. If a user enters into a building his device inspects the whole area, detecting the access points. Though, other things such as getting the basic information about the access points, collecting their unique MAC address etc. will be properly documented and stored in a database. While retrieving these locations of the different access points, the device also calculates the signal strength for each of them. The system then inserts one of the entries per signal strength factor into another database table having the coordinates of the relevant access point. However In case of iDocent, these coordinates are signified on two-dimensional Cartesian coordinate system. (Aebi, 2012)

3   Development:

3.1   Methodology:

3.1.1   Software Development Life Cycle (SDLC):

Before starting the development process **Enhanced Waterfall model** was proposed as a suitable method. Although, this method helps in time management for completion of the system and evaluation can be done from time to time for specifying the continuous progress of the system it could not provide robustness in development process. Therefore, **Prototyping model** is used for application development in this project. In Prototype model, before proceeding to design and coding stage, a throw away prototype of the system is developed and introduced to the client. Although development of prototype system includes design and coding, it is performed in informal way. Once client interacts with the prototype and evaluates it against his/her requirement the system is developed in formal way with all the required documentation and procedures. Until the requirements of client is not meet, redevelopment of prototype is continuously done. This enable developers to understand the requirements of the clients (Lion Vision Info Tech Solutions, 2013).
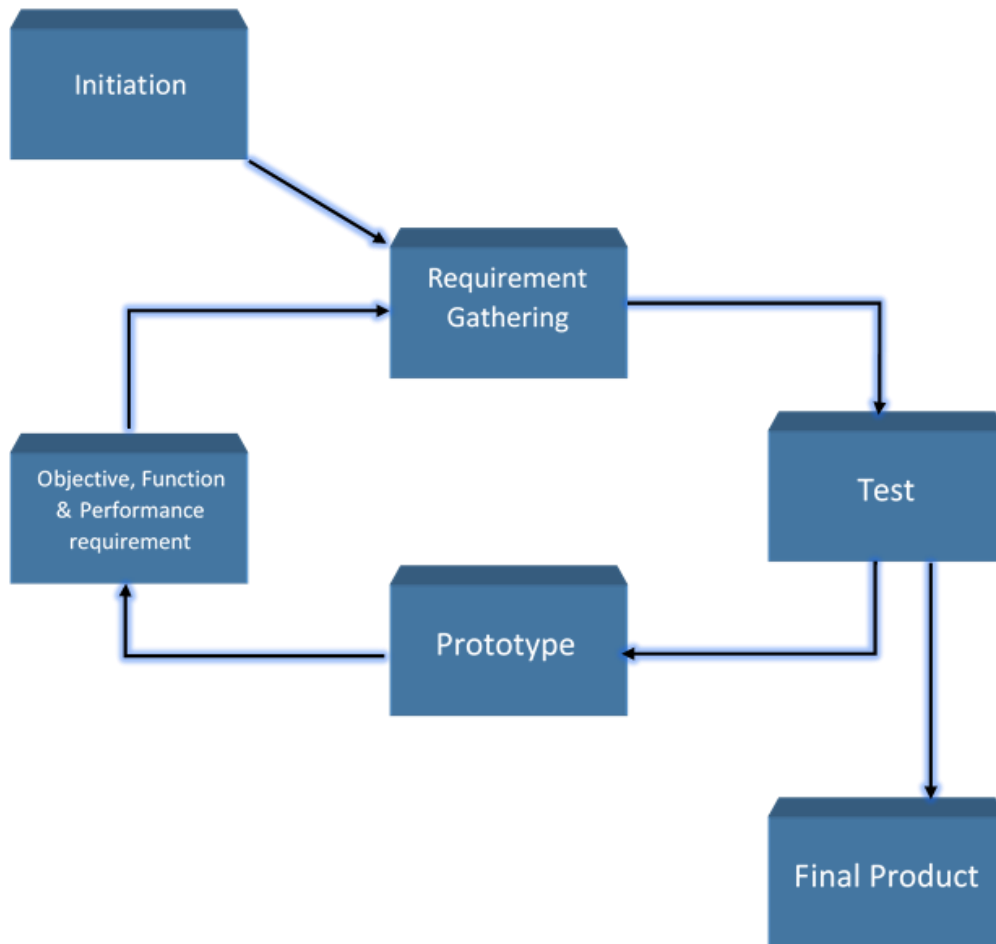
Figure 7: Prototyping Model

Since, the requirement of this project is unknown during early stages prototyping model would provide a basic idea instead of freezing the development process. Being a suitable software development model for large a sophisticated applications it is highly appropriate for the development of this project. This methodology helps in determining specific requirements of an application during development process and provides higher flexibility than other haphazard approach to develop any kind of system. Preferring this method it can assure for effective management of schedule or cost and determining Quality of Assurance (Choudhury, 2011 Dec 15). Hence, this methodology is proposed for the development of this project.

3.1.2 **Development Environment and Tools and Techniques:**

3.1.2.1 **Android:**

In association with Open Handset Alliance Google designed Android which is Linux-based operating system especially designed for smartphones featuring touchscreens. Google

released Android code as open source. Day by day the developer community of Android is getting larger.

Number of available applications in the Google Play Store from December 2009 to July 2014
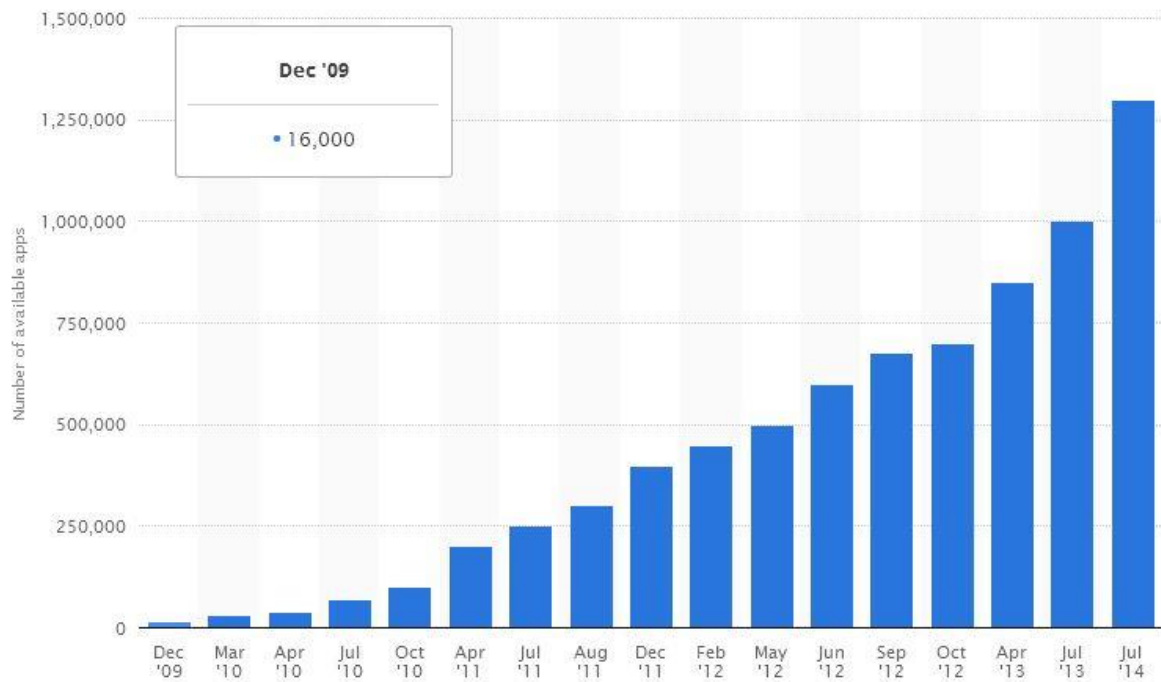
Figure 8: Available applications in Google Play Store (2009 - 2014)

Figure 9 illustrates the number of applications available in the Google Play Store dated from December 2009 to July 2014. According to Statista (http://www.statista.com) Google Play Store amounted 500,000 application in May 2012 and exceeded 1 million application in July 2013.

Figure 9: Top US Smartphone OS by Market Share

During June 2012, Android was the leader of the market in smartphones with 52 % (The Nielsen Company, 2012-12-20).

3.1.2.2   Eclipse IDE:

This application is developed in Eclipse (IDE) Integrated Development Environment. It is the most used IDE for Android Application development as it provides custom plugin called Android Developer Tools.

Figure 10: Eclipse integrated with ADT (Android Developer Tools)

This ADT plugin in Eclipse provides developer a powerful and integrated environment as well as provides Android SDK (Software Development Kit) containing a complete set of development tools for developing mobile applications for Android platform. Figure 12 shows an example of how the Eclipse IDE looks like with the tool provided by ADT plugin whose toolbar is on the top of the screen.
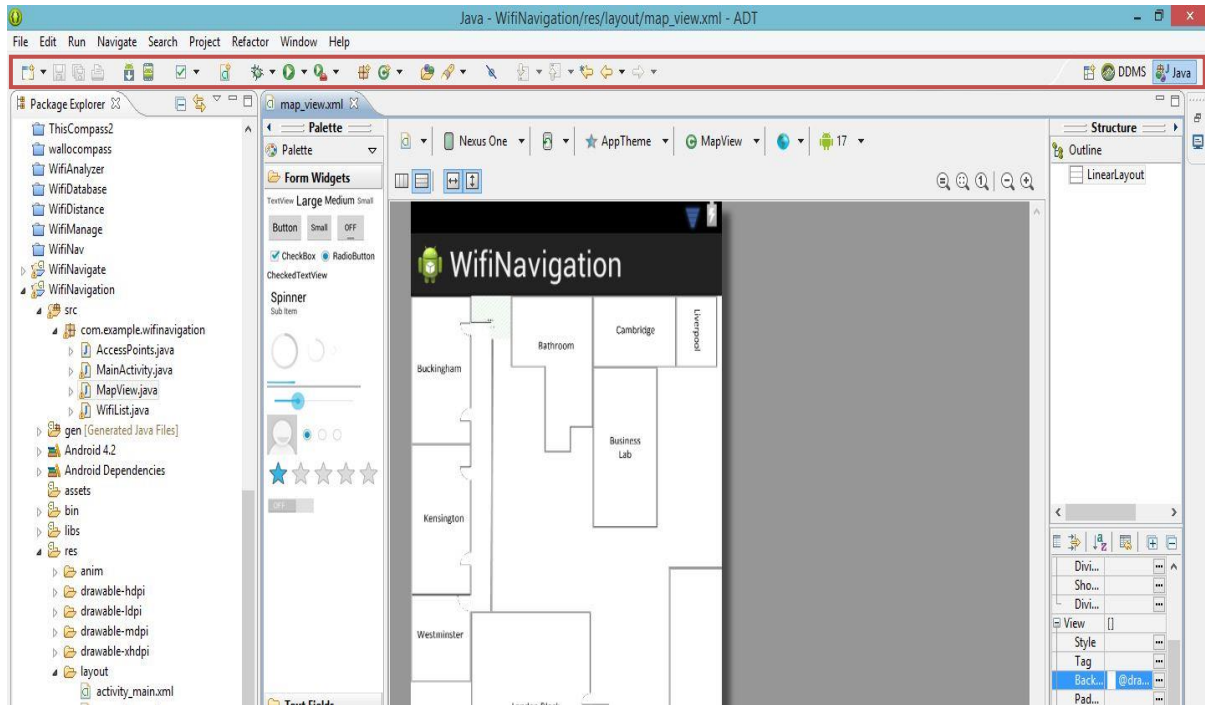
Figure 11: Eclipse IDE integrated with ADT plugin.

3.1.2.3   Android SDK:

It is a set of development tools used for developing applications for Android platform. The Android SDK includes following:

1. An Emulator for testing application.
2. Android OS tutorials
3. Debugger to check any error while running application.
4. Proper Documentation for Android Application Program Interfaces (APIs).
5. Required Libraries.
6. Sample source codes.
7. Android OS tutorials.

Since Android application is written in java code, Java Development Kit (JDK) is installed. (Janalta interactive Inc, 2010-2014)

3.1.2.4 JDK (Java Development Kit) 7:

It is a SDK for developing java application. It is required to install JDK for Android SDK to run the functions like debugging application, screenshots and application development (sublime1184, Nov 18 2009).

3.2 Software Requirement Specification:

3.2.1 Dependencies:

This section explains various dependencies required for the proper development and execution of this Android application.

3.2.1.1 Mobile Platform (Android Operating System):

Known to be as the most used open-source platform for smartphones, Android is an operative system developed and managed by Google Inc. which includes middleware and key applications. The increasing popularity and the fact of it being an open-source project has provided them a very large community of developers. The main reason for choosing Android as a development environment for this project rather than any other mobile platform like Windows Phone or iPhone OS is that Android is a Linux-based OS software. It stack is divided into four different layers that includes five groups:

1. **Application layer:**
   In this layer, application runs simultaneously. Moreover, it's used by end phone users and the language written in Java.
2. **Application framework:**
   This framework has been used to implement the standard structure of an application for Android OS.
3. **Libraries:**
   The available libraries are written in C/C++ and are called through Java interface.
4. **Android runtime:**
   The android runtime consist of two components where the first contains the core libraries which provides most of the functionality in Java core libraries and the second the virtual machine Dalvik which operates as a translator in between the application and the OS.
5. **The kernel:**
   Android uses Linux based kernel for its device drivers, memory management, process management and networking.
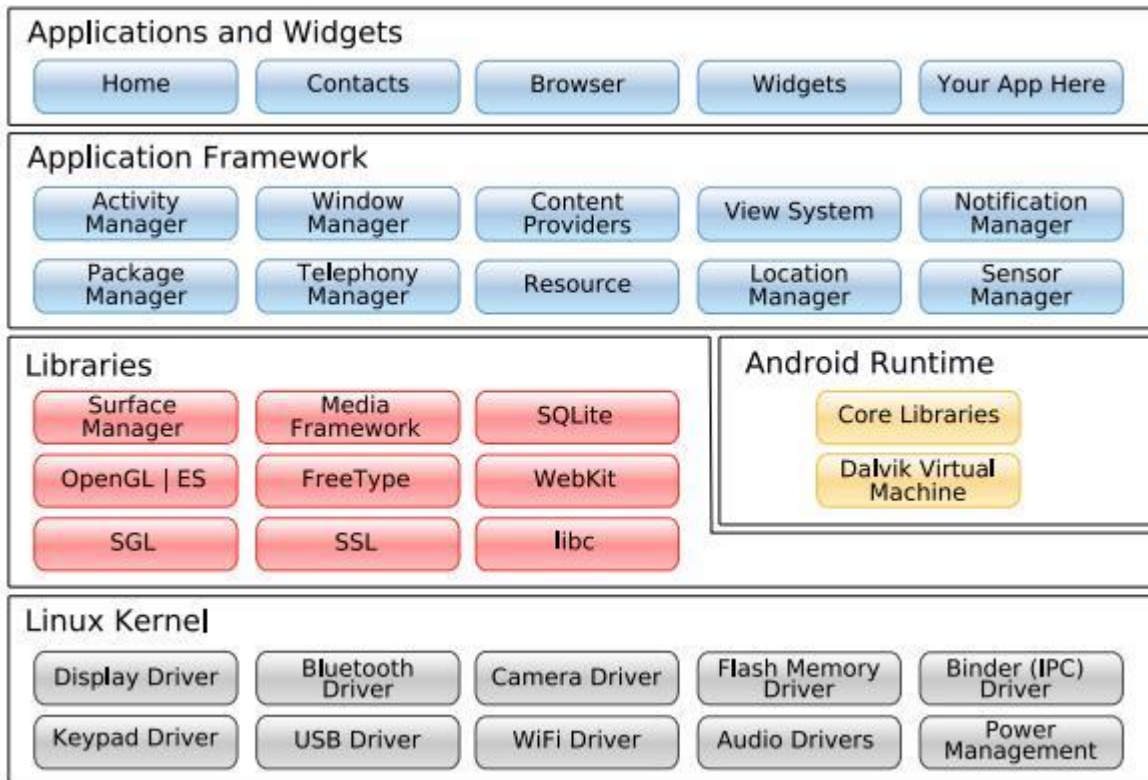
Figure 12: Android architecture diagram. (Kebomix, 2010)

3.2.1.2 Software Dependencies:

After installing Eclipse IDE for basic programming, ADT plugin should be added in order to perform android programming. ADT offers access to various features helping in development of Android applications. It provides GUI access to command line SDK tools as a UI designing tool for instant prototyping, designing of user interface (Android, 2014). Since Android application is developed in Java environment, JDK (Java Development Kit) 1.6 it needs to be installed without which Eclipse IDE will not function. Further Android SDK component includes different libraries and APIs required for the development of this android system.

3.2.1.3 Hardware Dependencies / Equipment:

Samsung Galaxy Core I8260:

Samsung Galaxy core is the next generation Samsung mobile device that uses android operating system version 4.1.2 (Jelly Bean). The main reason of choosing this device is the large numbers of built-in sensors into it as well as higher size RAM of 1 GB. Due to these specification the computation of program would easier.

Figure 13: Samsung Galaxy Core I8260 smartphone

Samsung Galaxy Core I8260 specification table	
Physical dimensions:	Height: 129.3 mm Width: 67.9 mm Depth: 9 mm Weight: 124 grams
Processor:	Cortex-A5 Dual-core 1.2 GHz
Operating System:	Android OS, version 4.1.2 (Jelly Bean)
Memory:	1 GB RAM
Display:	4.3 inches, 480 x 800 pixels
Connectivity:	GSM/HSPA Wi-Fi 802.11 b/g/n Bluetooth version 3.0 A2DP Micro-USB v2.0
Sensors:	Accelerometer, proximity, digital compass

Access Points:

Available Access Points used for this project are listed in Table 2 with their **vendor** detail
and **model ID**.

Name	SSID	BSSID (MAC Address)
AP1	Islington_Student	00:15:6d:5e:ec:7e
AP2	Business Lab	f8:d1:11:8c:16:42
AP3	dd-wrt	f8:d1:11:79:52:90

Table 1: Available Access Points
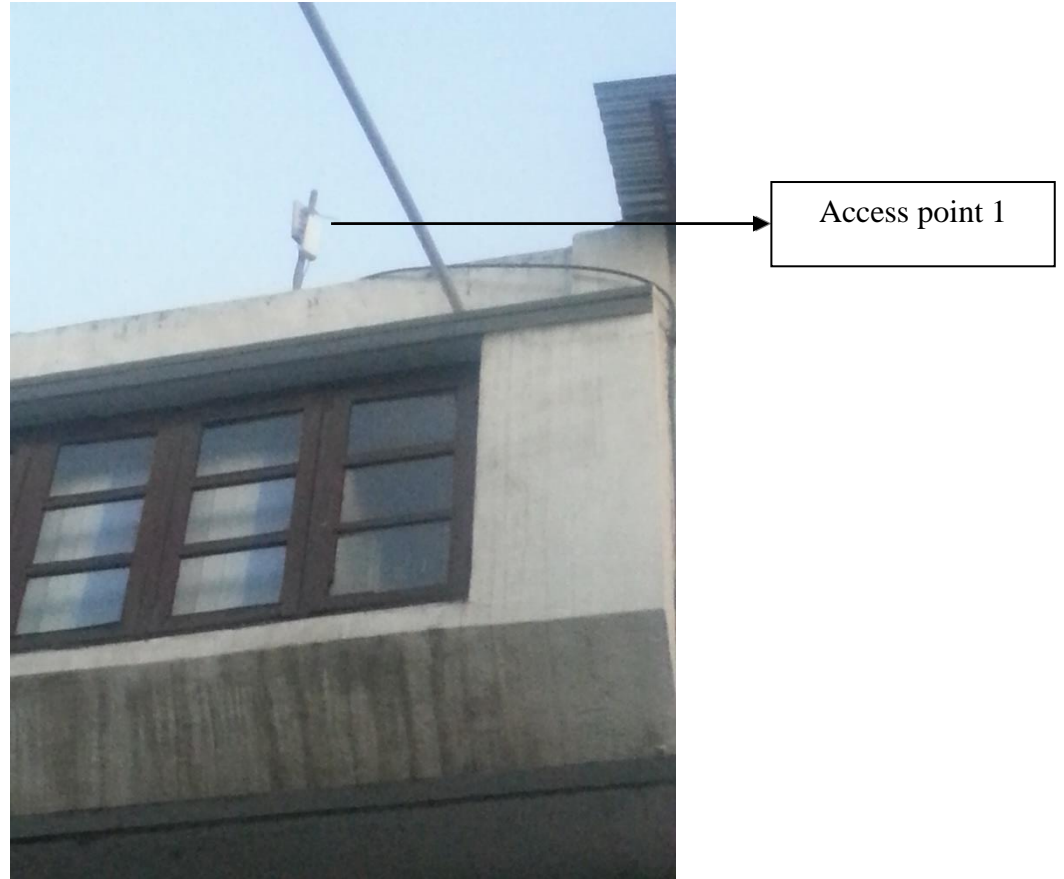
Figure 14: Access Point 1.
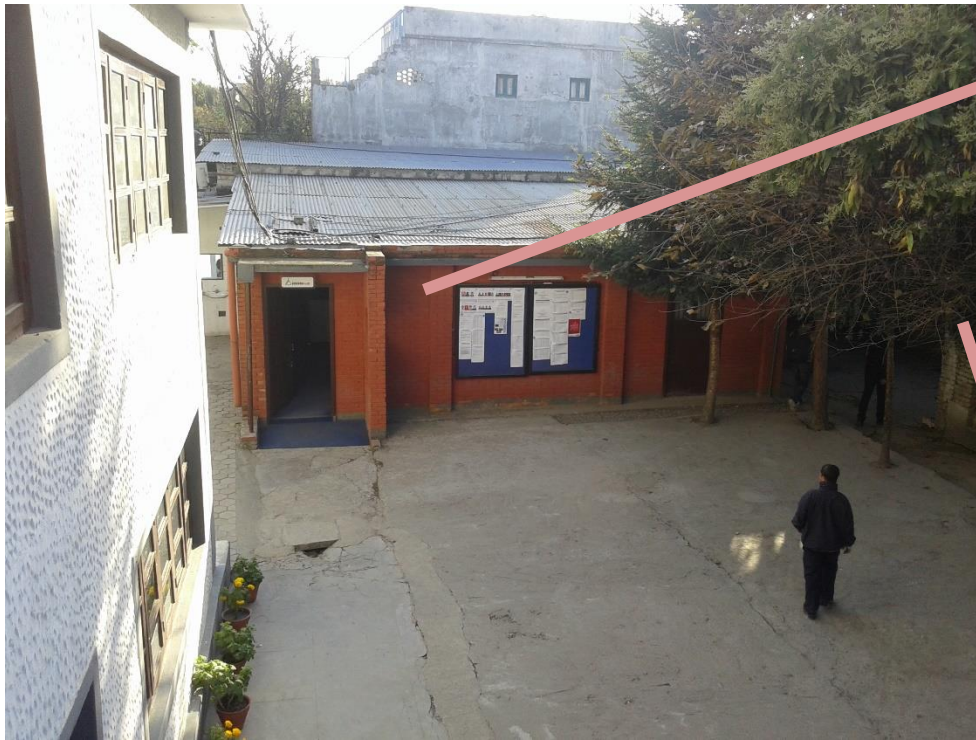
Access point 2 is inside this room

Figure 15: Access Point 2.

Access point 3

Figure 16: Access Point 3.

3.2.2   User Interface Requirements:

3.2.2.1   Administrator User Interface:

Administrator user interface contains two "**Edit Text**" field where he/she has to enter X and Y coordinate of the position manually during fingerprinting process. There are three **"Radio buttons"** each representing different Access Point. When a user press **"Start"** button the RSS scanning process is initiated. There is also a **"Stop"** button to immediately stop scanning process if required. A counter displays no of scans and at the end a notification pops up if the fingerprint is successfully written into the file.

3.2.2.2   Clients User Interface:

Similar to admin user interface, client user interface also contains three different **"Radio Buttons"** representing three different APs. Difference is that once the radio button is clicked RSS scanning process is automatically executed. For example if a user presses "AP1" radio button then signal strength from "AP1" is scanned for some time and collected RSS values are summed up and displayed in UI as well as when the client presses **"Get X and Y Axis"** button the application retrieves X and Y coordinates from the fingerprint file if any of the RSS value matches. When the client presses **"Find On Map"** button the application opens map view and displays current position of the mobile device on the map.

3.2.3   System Overview:

"Wi-Fi Based Indoor Navigation System" is a system developed using Android SDK and JDK. Is has two different user interface one for the administrator and other for the client. Figure 13 illustrates the user interface developed for administrator for generating Wi-Fi fingerprints and save them in three different files respective to three different APs.

Figure 17: Admin UI for generating RSSI fingerprints.

Similarly, Figure 14 demonstrate the user interface developed for clients to find their position in the college premises.
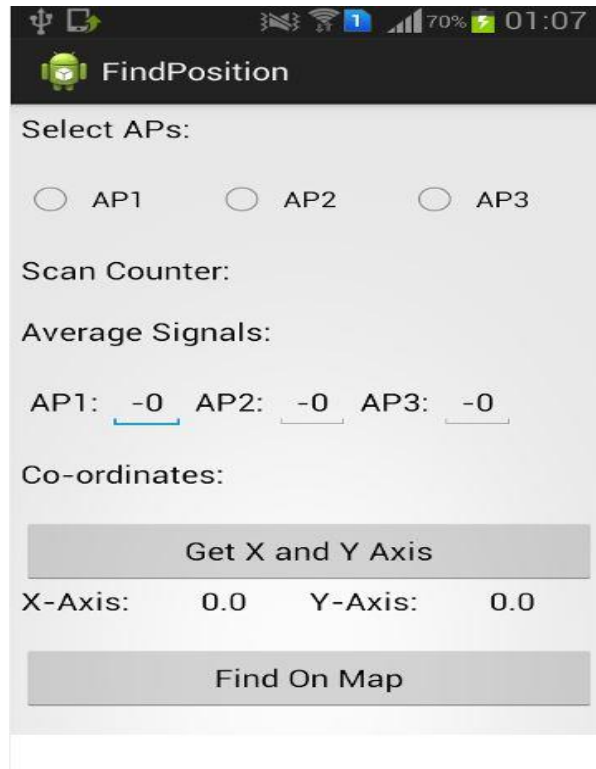
Figure 18: Client UI for scanning Average RSSI signal from three APs.

Figure 19: Map view after retrieving coordinates from RSSI fingerprint

3.2.3.1 System Architecture:

This system uses various method to achieve the aim and objective of the project. It has used the combination of various technologies like Wi-Fi Fingerprinting, motion detection and map view for the navigation process. Here fingerprinting technique is used to estimate the position of the client's mobile device whereas motion detection is used to count the step taken by the client holding the mobile device. Here the pointer is moved after every step taken by the user.

Training Phase:

An Android application developed using WifiManager API provided by Android SDK scans numbers of available APs and retrieves their SSIDs (Secured Set Identifiers), BSSIDs (Basic Service Identifiers) and corresponding (RSS) Received Signal Strengths in dBm. After scanning, all these information are saved to a file representing fingerprint data from different APs with their respective location.

Now this fingerprint database co-relates both training phase and operation phase because fingerprint collected in training phase is used in operating phase to determine the user's location. This construction of real time fingerprint data file for position estimation is followed by analysis for validating models for signal propagation. For the construction of fingerprint a mobile device is employed to receive broadcast signal from available APs periodically and measures the RSS values of each APs and finally calculates average RSS value and write it into fingerprint data file for each position. An android application for client to find a location is installed into their mobile device along with the fingerprint data file.
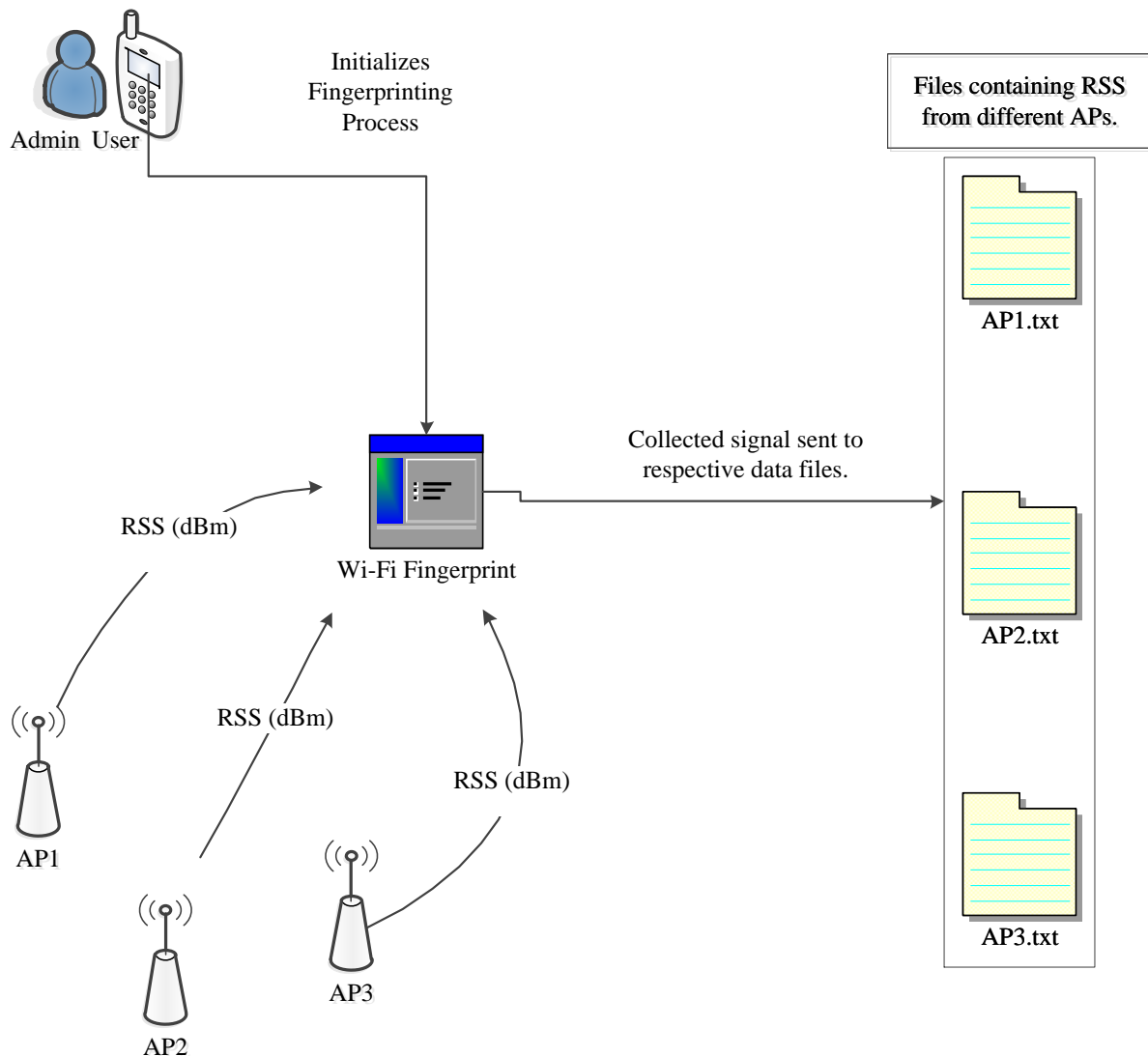
Figure 20: System architecture for wireless fingerprinting application (Training Phase).

Operating Phase:

In this phase, every mobile device whose location needs to be known performs position estimation process. Here the position is estimated based on the two different data sets. First is a previously constructed fingerprint data file from training phase. Second the set of RSS data measured by the user mobile device from every fixed APs.

For execution of position estimation process the application compares currently scanned average RSS values with fingerprint data file containing all the average RSS from N fixed APs. If average RSS value matches with fingerprint file the system returns X and Y co-ordinate from the file and plots the pointer in the map view in corresponding coordinate.
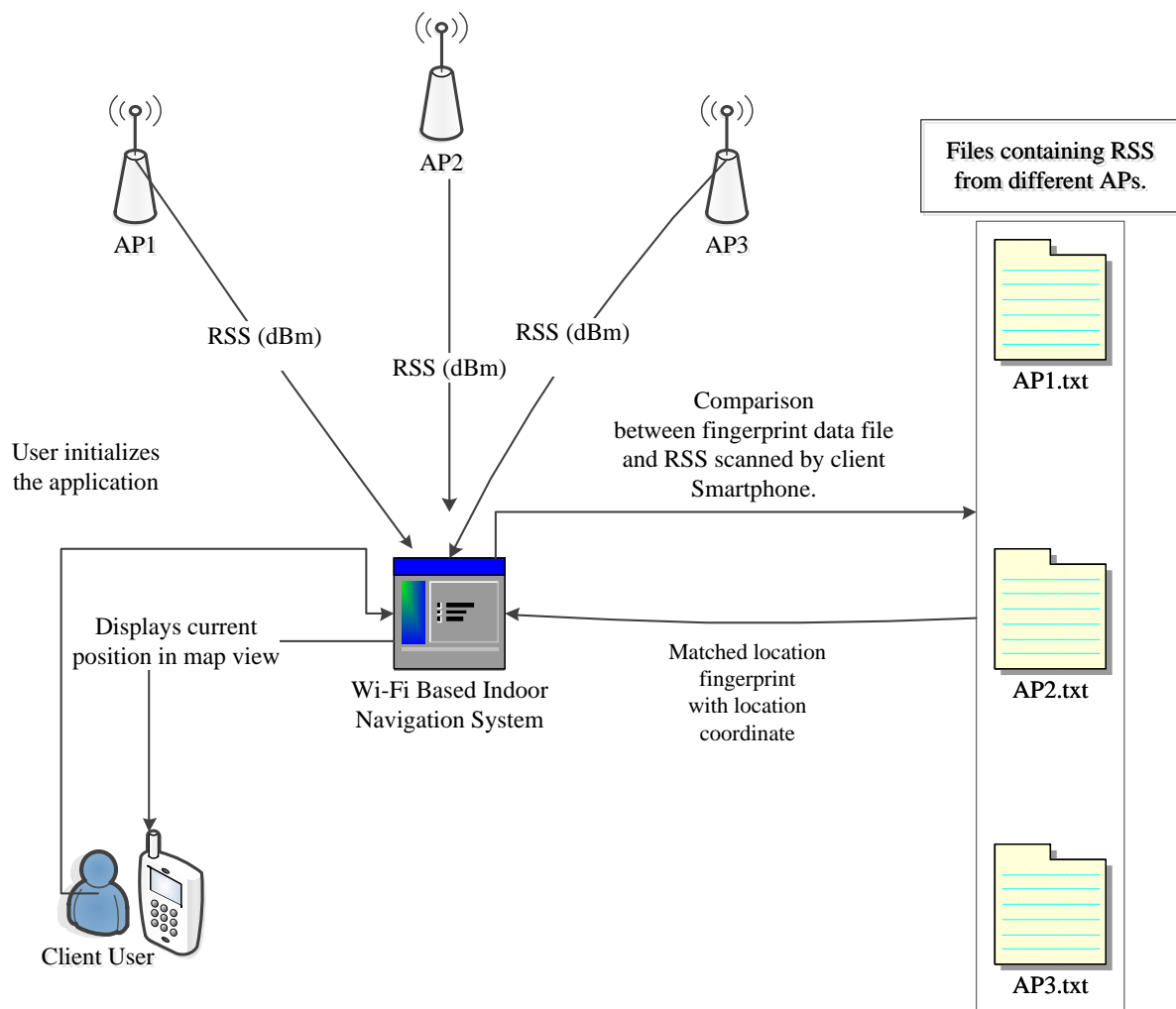
Figure 21: System architecture for position estimation and displaying current position in map view (Operating Phase).
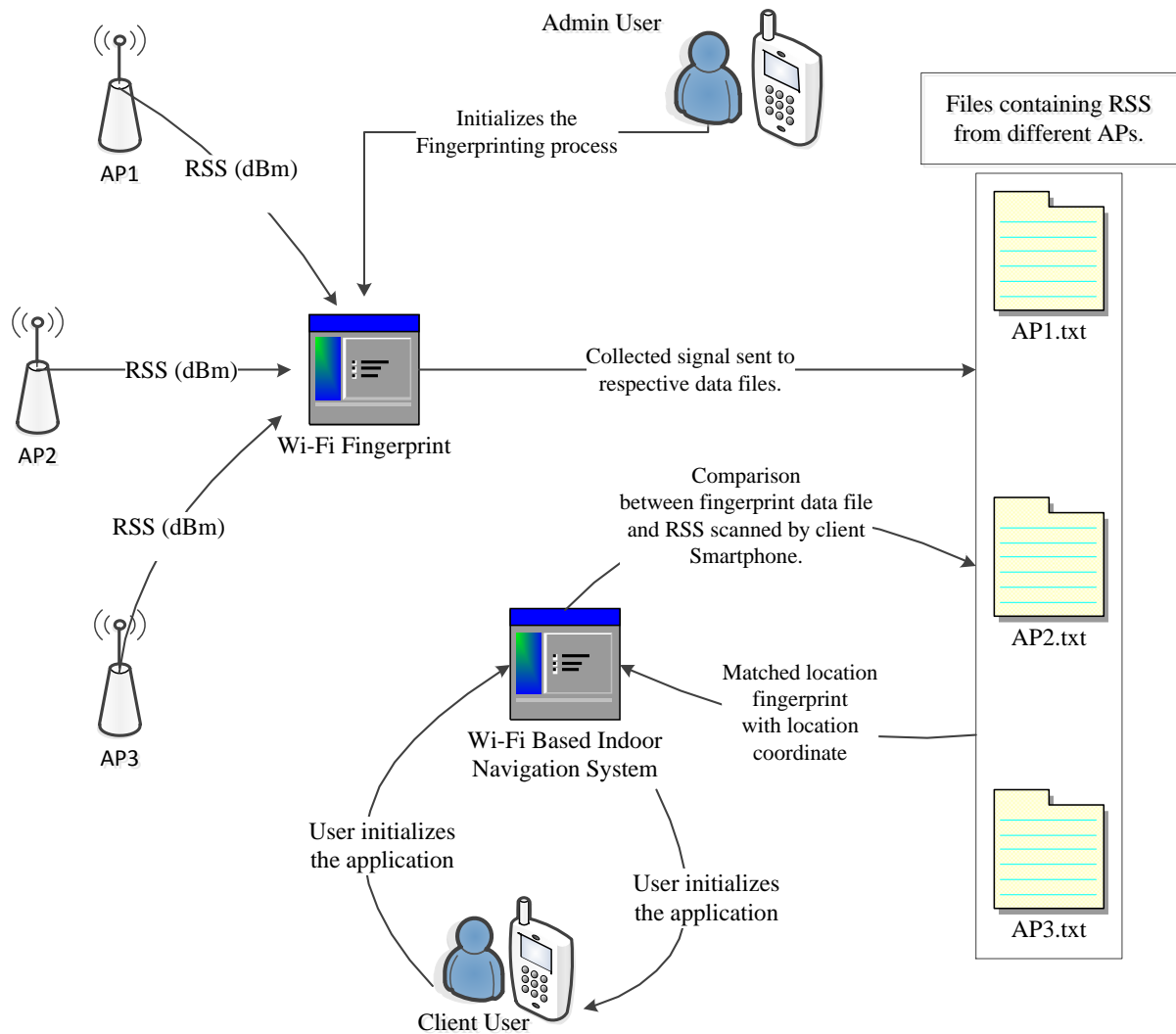
Figure 22: System Architecture for the system.

3.3 Design:

3.3.1 Process Flow:

3.3.1.1 Wi-Fi fingerprinting process flow for administrator:

Administrator	Manually sets X & Y co-ordinate of the position where scanning process is to be performed. → Selects different APs and starts signal (RSS) scanning process.
Undefined	Average RSS value along with X & Y co-ordinate and BSSID of AP is saved into fingerprint file for each AP. ← System scans signal for particular time and calculate average RSS value. Successful creation of fingerprint file is later used for determining position of client user.

3.3.1.2 Process flow for client to find their position:

User	User chooses access points and scans signal strength (RSS) for particular time.
Undefined	Sums up all RSS values and calculate average RSS → Stores average RSS values in different variables for each AP.
	If the fingerprint does not match, user is again asked to perform scanning. ← Application compares average RSS values with fingerprint file previously created by admin in training phase.
	Application retrieves X & Y co-ordinate from fingerprint file if average RSS value matches.

3.3.1.3 Process flow for map view and motion detection:

Undefined	Using X and Y co-ordinate retrieved from fingerprint, a map view displays current location of the system. → Points to the direction where the user or the mobile device is facing.
	The pointer moves towards the direction if a user takes any steps.
User	Successful creation of fingerprint file is later used for determining position of client user.
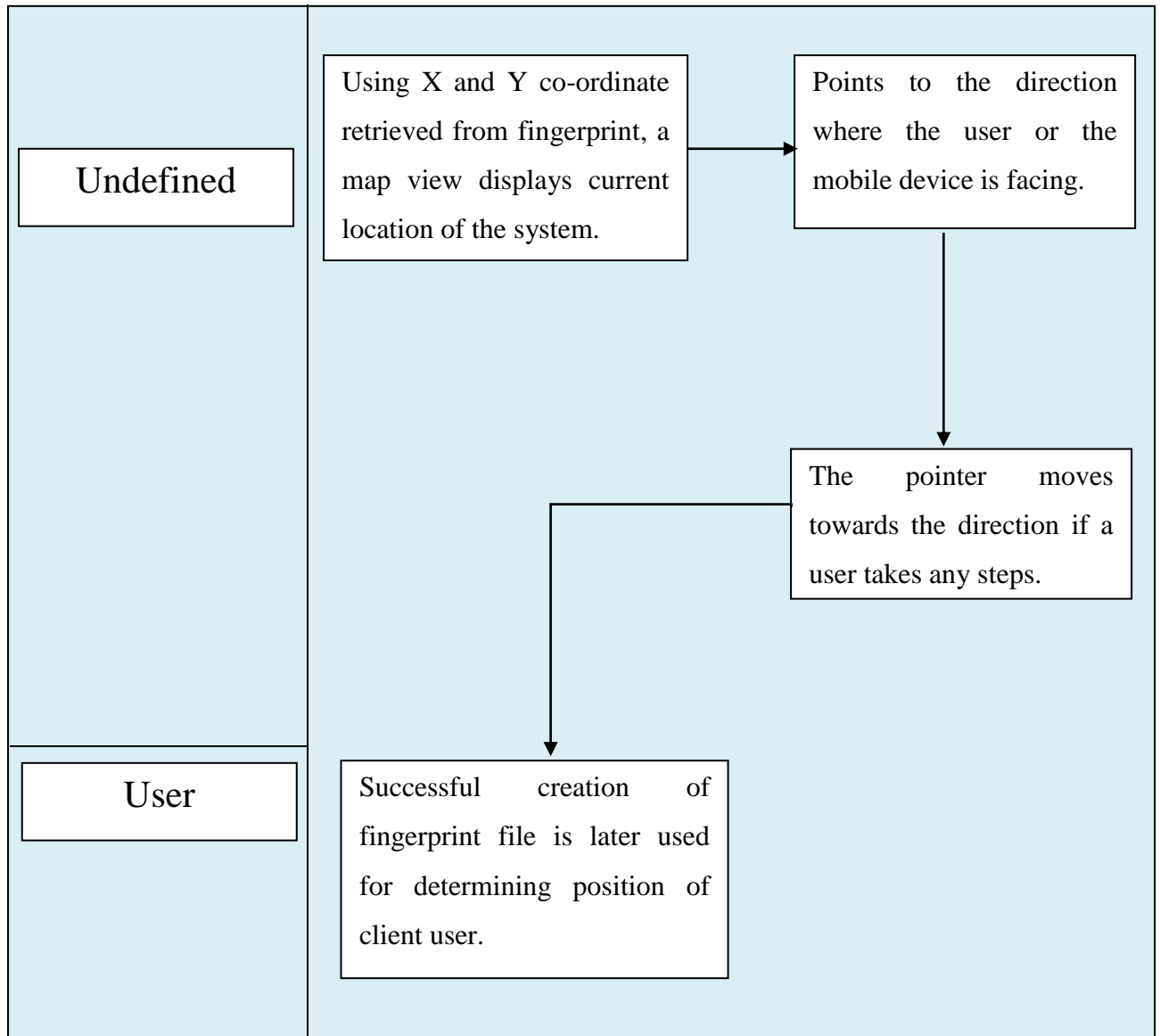
3.3.2   Context Diagrams:

3.3.2.1   Context Diagram for Wi-Fi Fingerprinting:

Administrator ——— Provide X and Y co-ordinates

Notifies after writing
data into the file.

Selects APs and
Start scanning
Process.

Wi-Fi
Fingerprinting
system

Provides average RSS values
after calculation
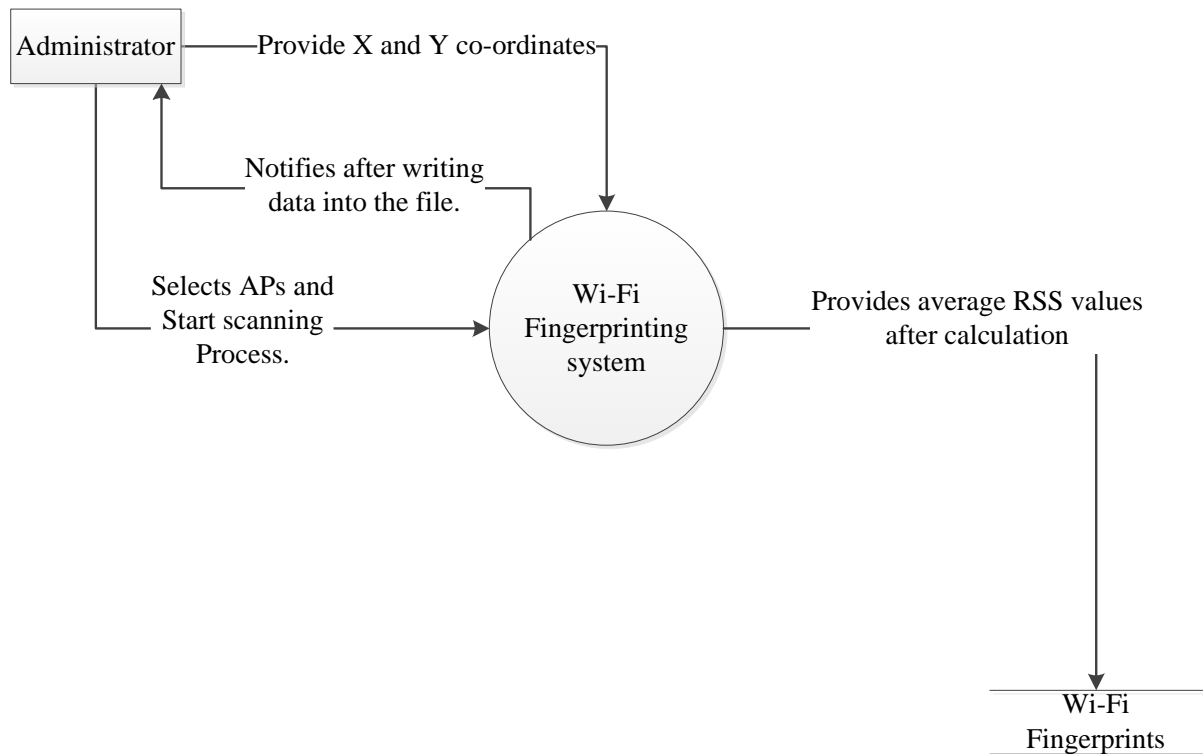
Wi-Fi
Fingerprints

Figure 23: Context Diagram for Wi-Fi Fingerprinting.

First of all, an administrator constructs RSS Fingerprints using the Wi-Fi Fingerprinting application where admin user provides X and Y co-ordinate to the system and scans RSS from selected APs. Later this information along with BSSID of APs are written into the file later used for position estimation of client user. After writing the data into file user gets the successful notification.

3.3.2.2   Context Diagram for Positioning System:
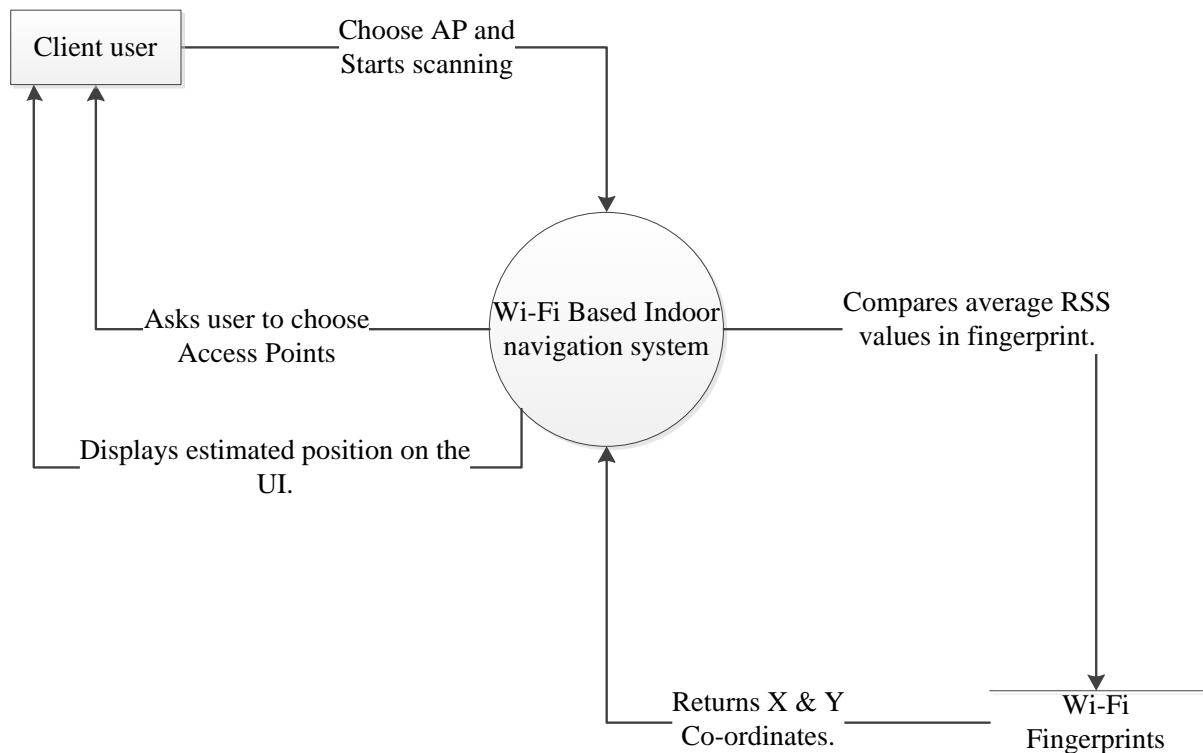
Figure 24: Context Diagram for Positioning System.

Before determining the position of the user they are asked to perform the same scanning process to calculate the average RSS value from selected APs. Those scanned RSS values are saved in different variables for each AP and compared with fingerprint file previously created by admin in training phase. If the RSS does not match with any data in the fingerprint user is again asked to perform the scanning. When the fingerprint is matched the application retrieves X and Y co-ordinate and the navigation pointer is placed into map view.

3.3.3 Use Case Diagram:
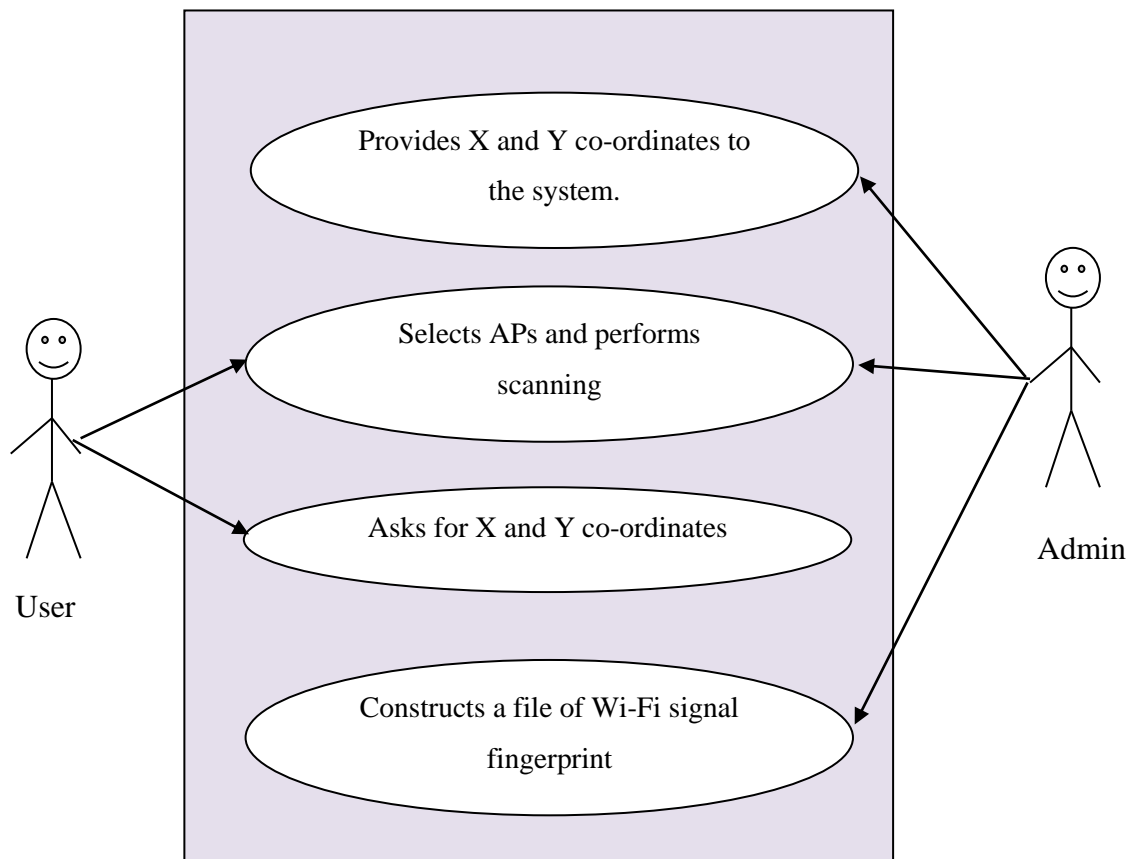
3.3.4   **Wi-Fi Fingerprint file format**

RSS | X Co-ordinate | Y Co-ordinate | BSSID

RSS fingerprint file format.
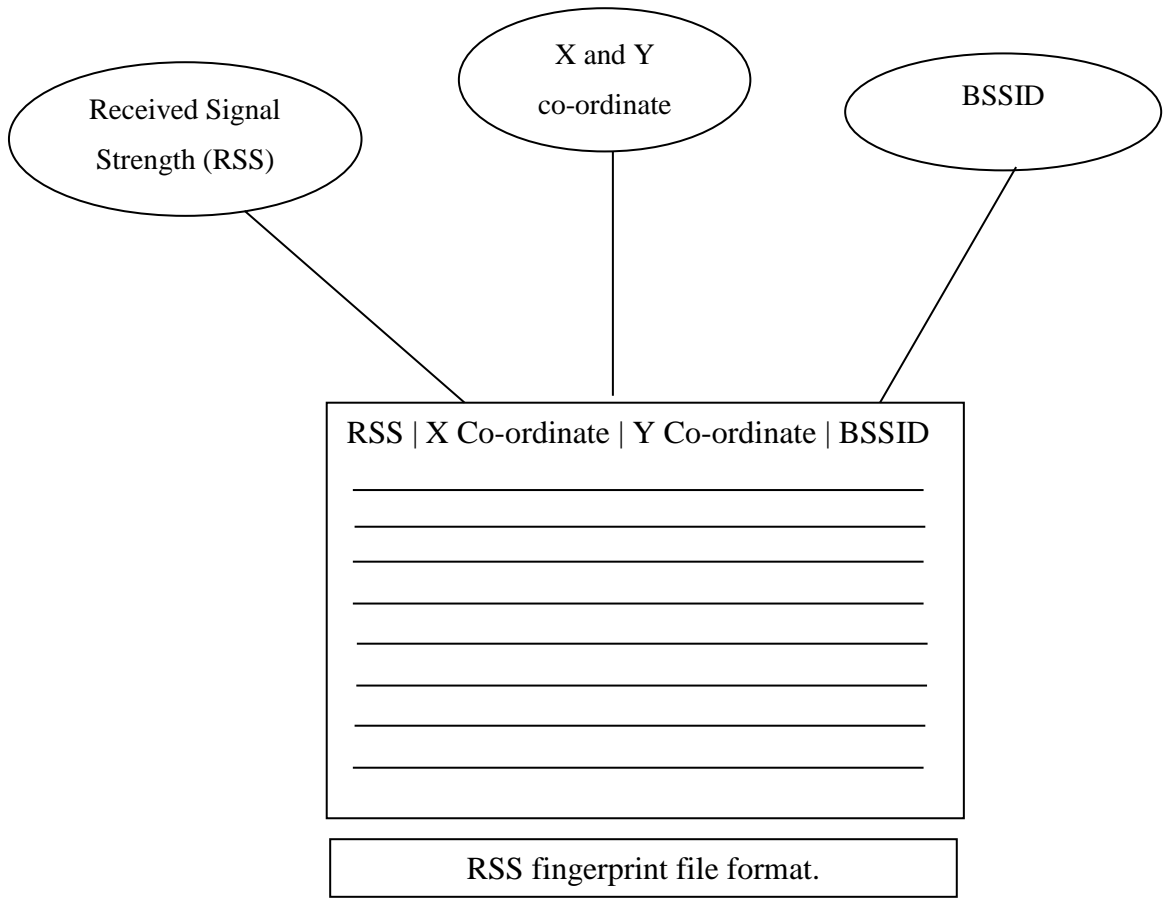
3.3.5 Flowchart:

Flowchart of various task of the system are represented below.
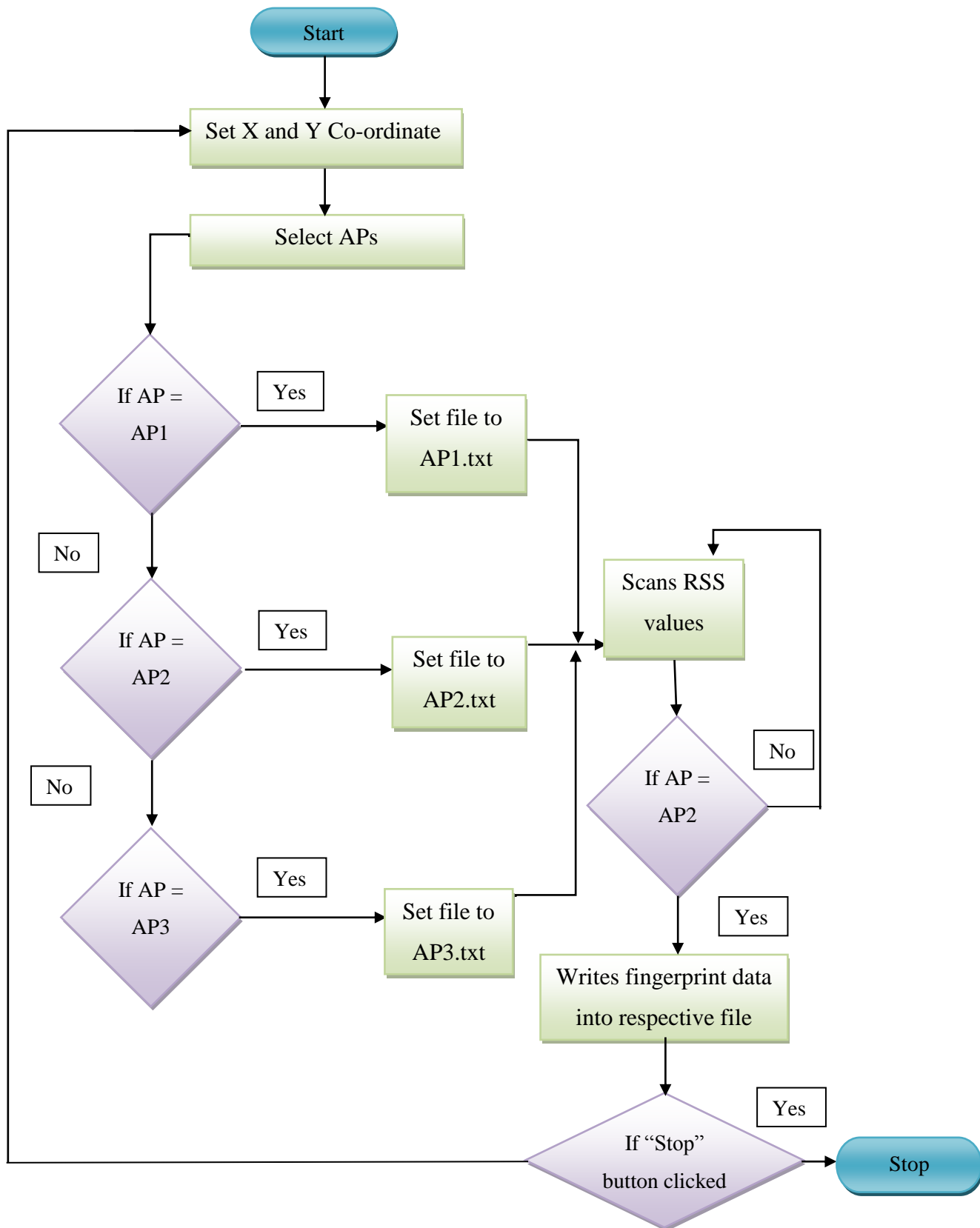
Flowchart for position estimate:
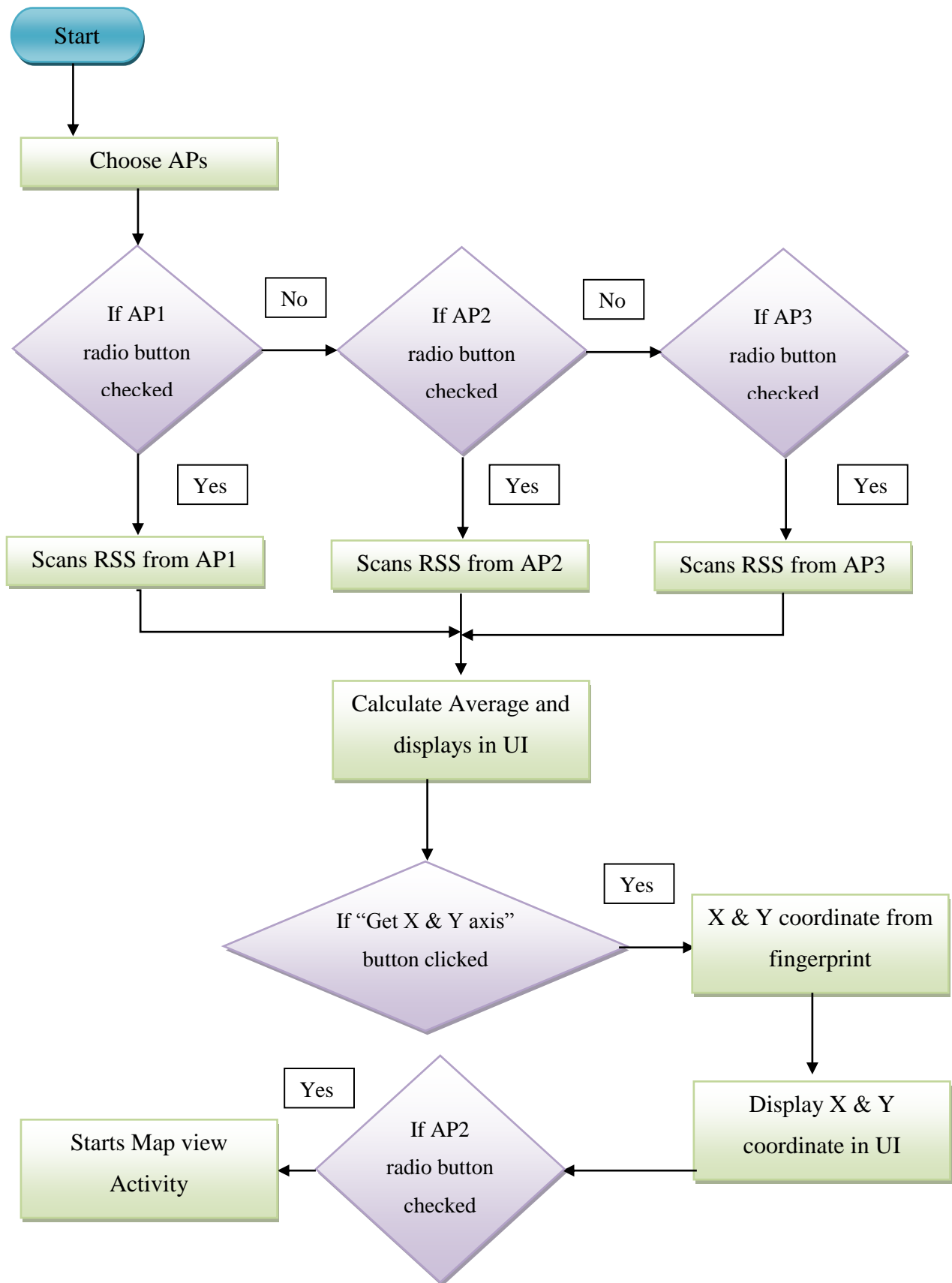
Flowchart for extraction of X and Y coordinate from fingerprint files:
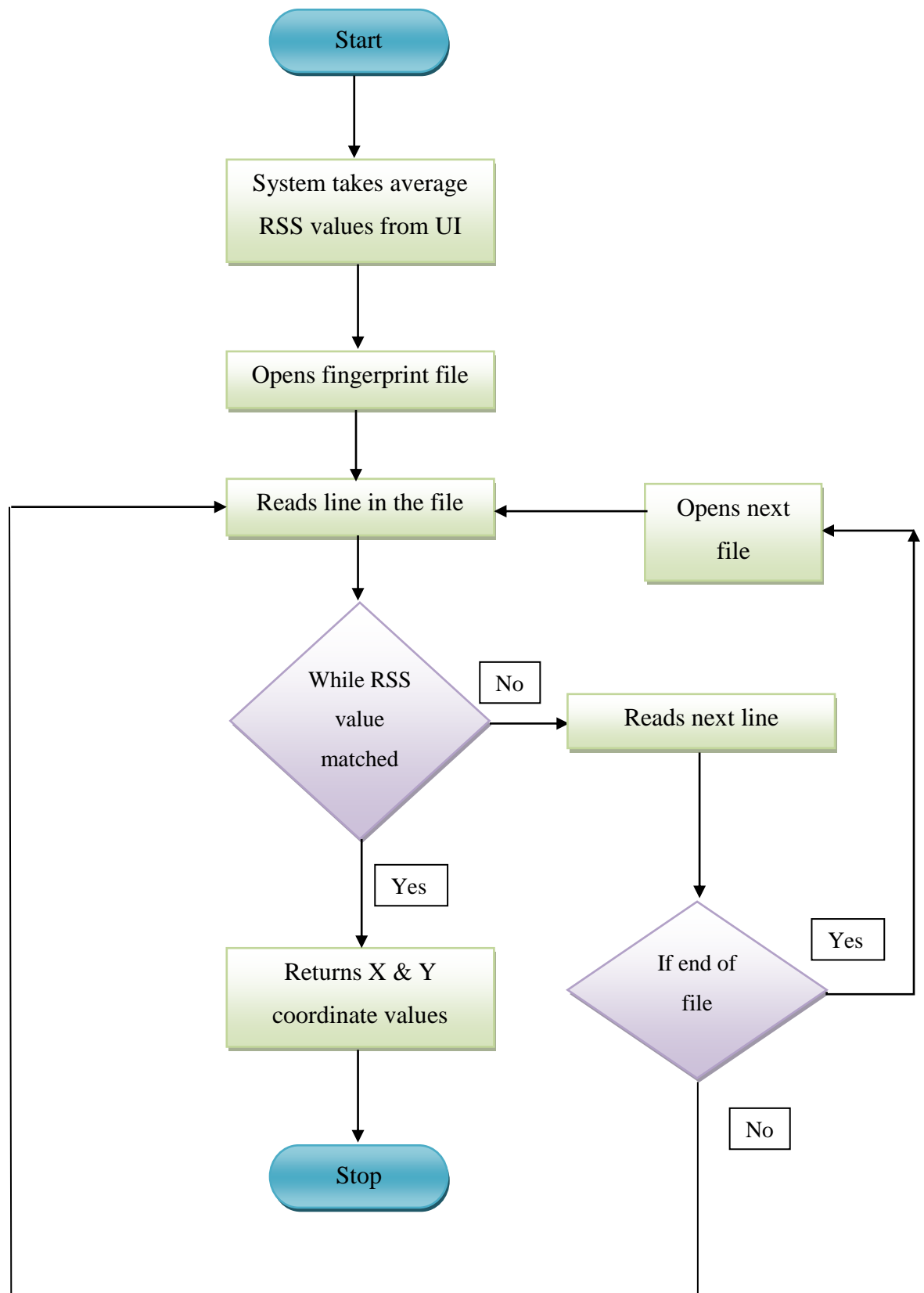
4   Testing, Results and Conclusion:

4.1   Unit Testing:

4.1.1   <u>Test Plan:</u>

4.1.1.1   Wi-Fi Fingerprinting test plan:

Test Cases:	Objectives:
1	To test whether an application runs on the mobile device.
2	To test whether application receives required information like (BSSID, SSID and RSS)from the AP
3	To test whether the edit text in an application allows the user to insert X and Y coordinate.
4	To test whether the system scans RSS from selected APs.
5	To test whether application sums RSS values and calculates average RSS.
6	To test whether application writes RSS fingerprint data into the file.

4.1.1.2   Wi-Fi based indoor navigation position estimation test plan:

Test Cases:	Objectives:
1	To test whether application runs on the mobile device.
2	To test whether application scans RSS from selected APs
3	To test whether application sums RSS and calculates average RSS and display on the user interface.
4	To test whether application reads the file and retrieves X and Y coordinate from the file.
5	To test whether the pointer is drawn on the map with respective coordinate
6	To test whether the estimated position is correct or not.
7	To test whether the pointer points towards the direction where the user is facing.
8	To test whether the pointer moves towards pointed direction after every step taken by user.

4.1.2 Test Log:

4.1.2.1 Wi-Fi Fingerprinting test plan:

Test Case 1	
Objective	To test whether an application runs on the mobile device.
Test Data	"Wi-Fi Fingerprint.apk" file is executed on the mobile device.
Expected Test Result	The application should run successfully.
Obtained Result	The application executed successfully.
Conclusion	Task successful.
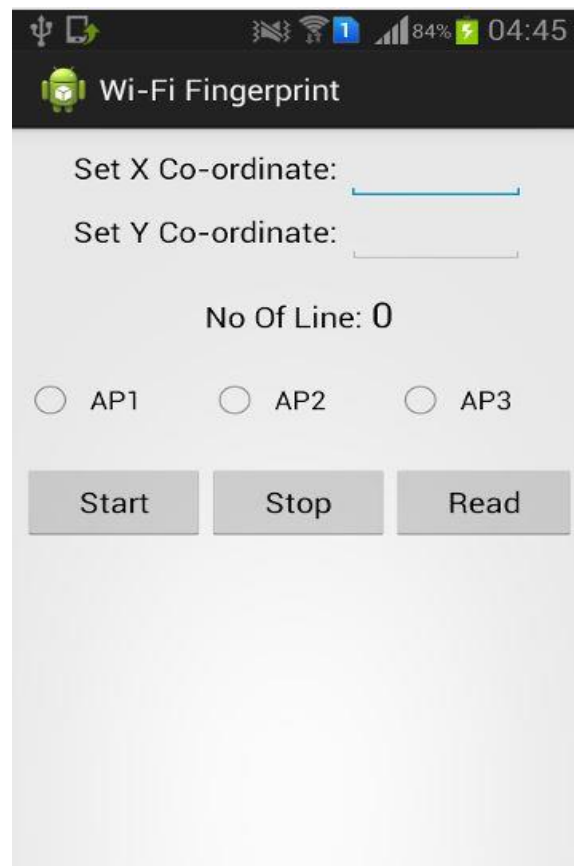
Test Case 2	
Objective	To test whether application receives required information like (BSSID, SSID and RSS)from the AP
Test Data	The user runs another activity to check the whether the application is receiving required information.
Expected Test Result	The application should receive required information
Obtained Result	The application received required information like (BSSID, SSID and RSS)from the AP
Conclusion	Task successful.
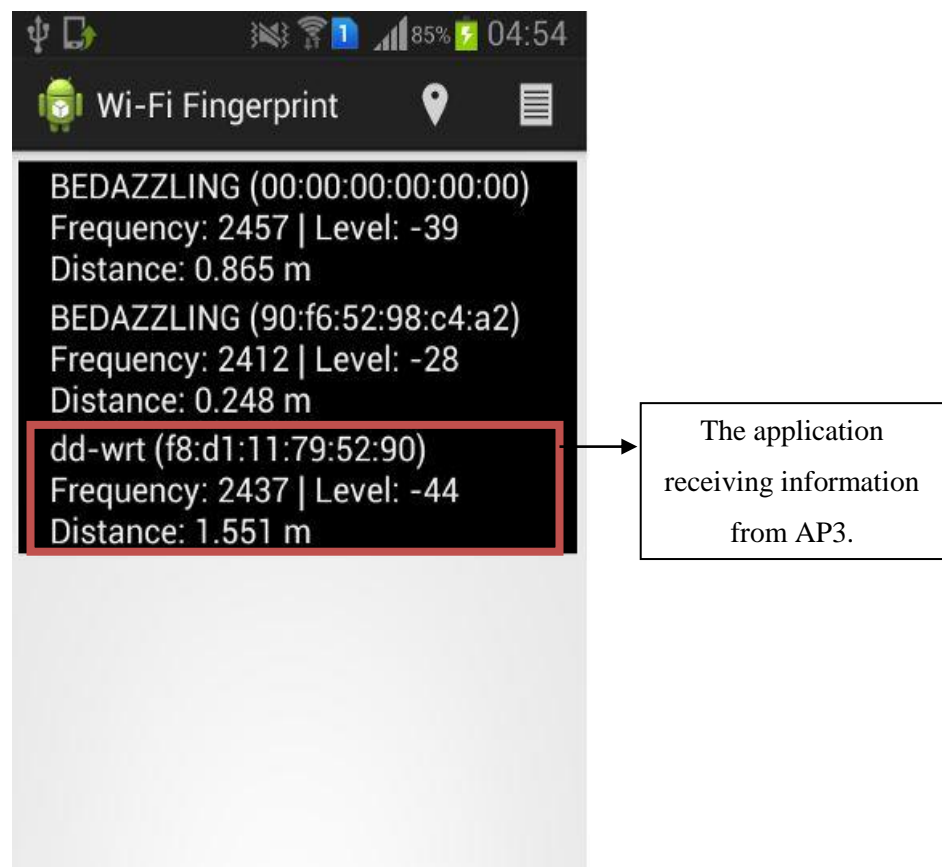
The application receiving information from AP3.

Test Case 3	
Objective	To test whether the edit text in an application allows the user to insert X and Y coordinate.
Test Data	The user sets X and Y coordinate
Expected Test Result	The application should allow the user to input X and Y coordinate.
Obtained Result	The application allowed the user to input X and Y coordinate.
Conclusion	Task successful.

Test Case 4	
Objective	To test whether the system scans RSS from selected APs.
Test Data	The user selects any AP and starts scanning.
Expected Test Result	The application should scans RSS from selected AP
Obtained Result	The application scanned RSS from selected AP
Conclusion	Task successful.

Test Case 5	
Objective	To test whether application sums RSS values and calculates average RSS.
Test Data	-
Expected Test Result	The application should sum all the RSS value and calculate Average RSS.
Obtained Result	The application summed all the RSS value and calculated Average RSS
Conclusion	Task successful.

Test Case 6	
Objective	To test whether application writes RSS fingerprint data into the file.
Test Data	Average RSS value, X and Y coordinate with AP's BSSID
Expected Test Result	The application should write the fingerprint data into the file and gives notification.
Obtained Result	The application wrote the fingerprint data into the file and gives notification.
Conclusion	Task successful.
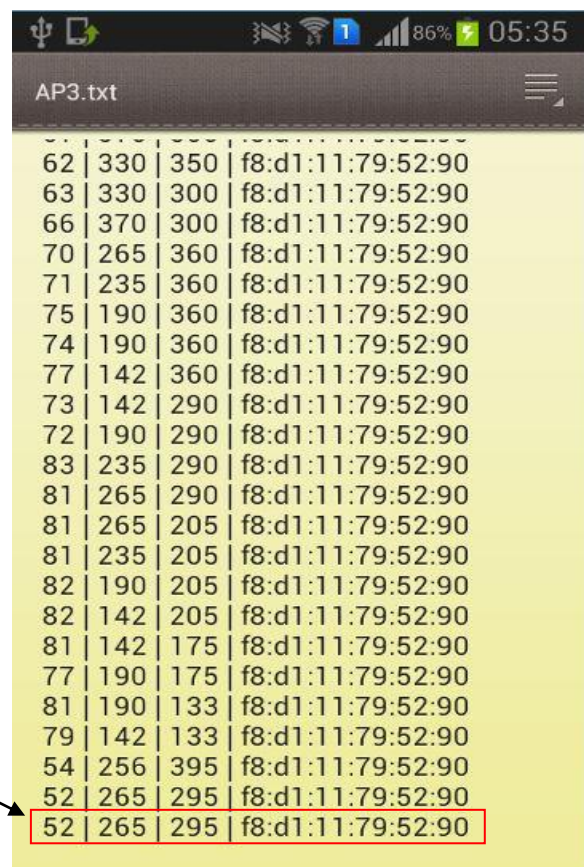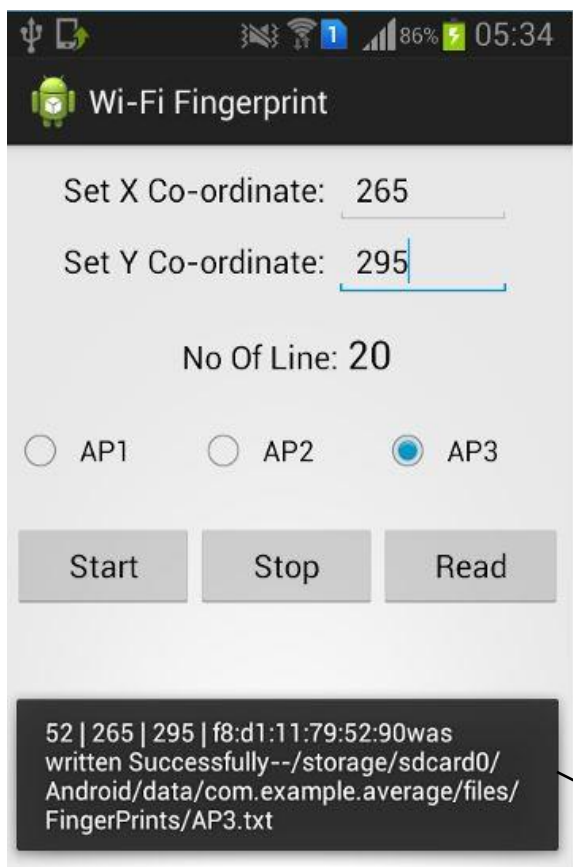
4.1.2.2   Wi-Fi based indoor navigation position estimation test plan:

Test Case 1	
Objective	To test whether an application runs on the mobile device.
Test Data	"Wi-Fi Navigation.apk" file is executed on the mobile device.
Expected Test Result	The application should run successfully.
Obtained Result	The application executed successfully.
Conclusion	Task successful.
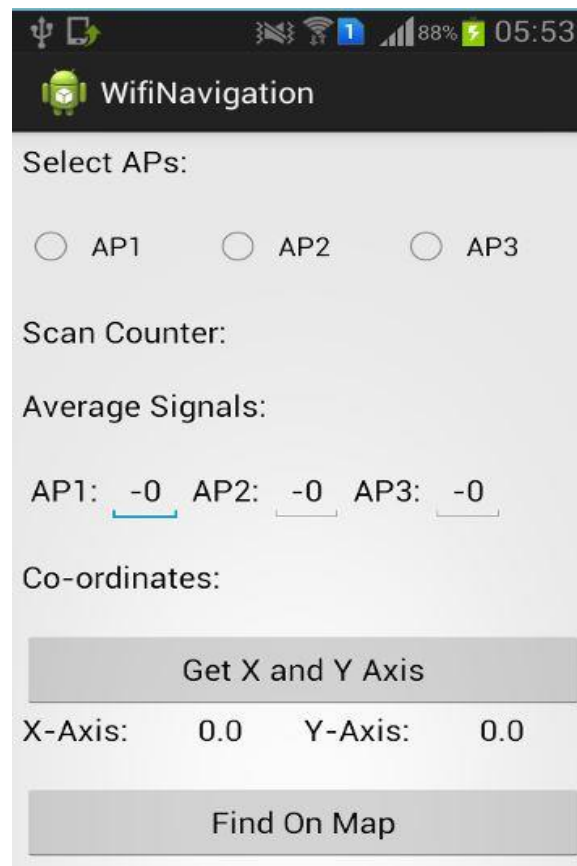
Test Case 2	
Objective	To test whether application scans RSS from selected APs
Test Data	Client selects any access points to scan RSS.
Expected Test Result	The application should scan RSS from selected AP.
Obtained Result	The application scanned RSS from selected AP
Conclusion	Task successful.

Test Case 3	
Objective	To test whether application sums RSS and calculates average RSS and display on the user interface.
Test Data	The average value calculated from summed up RSS value
Expected Test Result	The application should display average RSS on UI
Obtained Result	The application displayed averaged RSS on UI.
Conclusion	Task successful.

Test Case 4	
Objective	To test whether application reads the file and retrieves X and Y coordinate from the file.
Test Data	The average RSS value is compared with finger print file.
Expected Test Result	The application should display X and Y coordinate on the UI.
Obtained Result	The application displayed X and Y coordinate on the UI.
Conclusion	Task successful.

Test Case 5	
Objective	To test whether the pointer is drawn on the map with respective coordinate
Test Data	The value of X and Y coordinate is passed to Map view.
Expected Test Result	The application should display pointer on respective X and Y coordinate on the map.
Obtained Result	The application displayed pointer on respective X and Y coordinate on the map.
Conclusion	Task successful.

Test Case 6	
Objective	To test whether the estimated position is correct or not.
Test Data	--
Expected Test Result	The application should display pointer on correct position
Obtained Result	Although the retrieval of estimated position was successful from fingerprint, the accuracy of estimated position depends on RSS, due to the fluctuation of signal strength the position is not always correct.
Conclusion	Practically failed.

Test Case 7	
Objective	To test whether the pointer points towards the direction where the user is facing.
Test Data	When top of the user's phone points to the certain direction, the pointer on the map also faces the same direction.
Expected Test Result	For example: When the user's phone faces North the pointer should also face North.
Obtained Result	The pointer faced on the same direction as the top of user's mobile phone faced.
Conclusion	Task successful.

Test Case 8	
Objective	To test whether the pointer moves towards pointed direction after every step taken by user.
Test Data	The pointer changes its position as a user takes steps.
Expected Test Result	The pointer should change its position on every steps of the user.
Obtained Result	The pointer changed its position on every steps taken by user facing direction as same as the user.
Conclusion	Task successful.

4.2 Critical Evaluation:

All the testing made above are executed in an android device in real time. All the test result shows that the application worked in training phase and effort made was entirely fruitful. The application developed for admin is capable of creating RSSI fingerprint for position estimation. But the most important progress concerns the accuracy of the position. The orientation of the device is calculated with the fusion of accelerometer and gyroscope and magnetic field sensor. The linear acceleration is calculated by deducting the constant value of gravity and the displacement of device or motion is detected. The algorithm works properly and delivers accurate result in terms of motion detection and the direction faced by the user.

Wi-Fi based indoor navigation system however compares the RSS values in the fingerprint file and the RSS value scanned by the client in the real time. It is capable of extracting the position from the file. This system also has a negative impact when there is error in position estimation. The positioning error occurs when there is fluctuation in Received Signal Strength from the same AP. The fluctuation of signal occurs due to the obstruction of various objects in the area. Indeed the paper introduces a mechanism to construct the Wi-Fi RSSI fingerprint for admin user and introduces the process for estimating user position using the fingerprint file

However, if I have the opportunity and get access to the advance hardware to start this project again I would be studying the **Triangulation Fingerprinting** approach in detail so that I would be able to create an application that could determine the accurate position of the client user.

4.3   Future Work:

There are many required task that is needed to be done for this project. After adding those features this application would be more robust and effective for both Administrator while creating Wi-Fi Fingerprinting and for client while estimating their position. To improve the accuracy of these applications there are various approaches.

1. Improvement in linear acceleration by using rotation matrix calculation provided by Android SDK.

2. The most popular technique for positioning system called Kalman filter can be used to incorporate information from various.

3. If Kalman filter technique is used it can improve the accuracy of the position estimation due to its recursive nature of error estimation.

4. With higher computational configuration on handheld device, implementation of propagation based technique is possible to estimate the position of the mobile device using wireless network.

4.4   Conclusion:

Wi-Fi Based Indoor Navigation System is a positioning system developed for Android device in the context of Islington College. The system is develop to determine the position of the user within the premises of the college.

The main objective of this project as described in the "Aims and Objectives" part. While performing the research for this project, a basic concept of gaining level of accuracy in indoor navigation by using available sensors in android devices and developing a prototype application and research on available application of the technology.

In the research phase, various positioning algorithms were identified. These research showed that indoor positioning has different problems and different approaches and solutions. The basic hardware infrastructure required are external infrastructures like (Wi-Fi Access points, mobile devices.) and the built in sensors of mobile devices (accelerometer, gyroscope, etc.).The Inertial Navigation System (INS) is integrated with signal strength measurement called Wi-Fi fingerprinting used to estimate the position according to the known position of the Wi-Fi Access Points. The comparison between measured RSS and previously stored fingerprints determines the current position by selecting most similar values in the fingerprint.

Due to the fluctuation and variations of Wi-Fi signals the estimated position may be inaccurate. The prototype application was implemented in Android mobile. It saves the fingerprint file for the mapping coordinate, senses movement of the device with accelerometer, magnetometer and gyroscope, scans RSS signals from wireless network, calculates average RSS and retrieves coordinates from the finger print file. Later these coordinates of estimated position are transferred to the map activity to place the pointer on the map view.

5 Social and Ethical Issues:

This project is developed for the academic propose for the Final Year Project (FYP) for the course "B.Sc. in Computer Networking and IT Security" experimented in Islington College affiliated to London Metropolitan University. Since, it is developed for Islington College detail information is provided in "Client Information" part.

Different journals, articles, books and websites are taken as reference for the guidance in developing this project. The list of journals and other reference documents are listed in Reference section of this project. Additionally, the project concept and required resources are provided by Roshan Chaudhary my 1st supervisor, therefore his credit is also mentioned Acknowledge section. For the idea and concept of motion detection and overall android development concept, Abhinav Dahal lecturer of our college has also been acknowledged for his unforgettable guidance.

Reason for being ethical	Reasons for being unethical
Wandering students are able to locate themselves without wasting their time	Positioning data can be modified to create false positioning
May prevent crime from occurring.	Information of students can be used against them.
This application can be used as an advance technological infrastructure.	Many students feel difficulties in using this system as it require android device and devices cannot be afforded by all the students.

6   References:

Aebi, F., 2012. *Implementation of an Autonomous Indoor Navigation System on Android,* Switzerland: Software Engineering Group Department of Informatics University of Fribourg.

Android, 2014. *Android Developer Tools.* [Online]
Available at: http://developer.android.com/tools/help/adt.html
[Accessed 25 Oct 2014].

Babu, S., 2014. *Robots can now locate objects with RFID tags.* [Online]
Available at: http://techliveinfo.com/robots-can-now-locate-objects-rfid-tags/

Baruch, M., 2011. Indoor localization by WiFi. Issue IEEE International Conference, pp. 449-452.

Beal, V., 2014. *Wifi.* [Online]
Available at: http://www.webopedia.com/TERM/W/Wi_Fi.html

Burhum, R., Sep 14, 2012. *GeoMeetup - WiFiSLAM - Modern Innovations in Indoor Positioning (Joseph Huang).* [Online]
Available at: http://www.youtube.com/watch?v=OGdvjvla1Tc
[Accessed 31 March 2014].

Choudhury, A., 2011 Dec 15. *Prototyping Model.* [Online]
Available at: http://www.sdlc.ws/prototyping-model/
[Accessed 18 Oct 2014].

Encyclopædia Britannica, Inc. , 2014 Feb 2014. *GPS.* [Online]
Available at: http://www.britannica.com/EBchecked/topic/235395/GPS
[Accessed 15 Oct 2014].

Francis, J. W., 2011. *A quick tutorial on coding Android's accelerometer.* [Online]
Available at: http://www.techrepublic.com/blog/software-engineer/a-quick-tutorial-on-coding-androids-accelerometer/

Janalta interactive Inc, 2010-2014. *Android SDK.* [Online]
Available at: http://www.techopedia.com/definition/4220/android-sdk
[Accessed 18 Oct 2014].

Karlsson, F. & Karlsson, M., 2014. *Sensor Fused Indoor Positioning Using Dual Band WiFi Signal Measurements,* Sweden: Media-Tryck.

Kebomix, 2010. *Kebomix Blog.* [Online]
Available at: http://kebomix.wordpress.com/2010/08/17/android-system-architecture/
[Accessed 9 Oct 2014].

Kuykendall, K., March 25, 2013. *Geographical Information System.* [Online]
Available at: http://gis.stackexchange.com/questions/55389/how-does-wifislam-work
[Accessed 23 March 2014].

Laoudias, C. et al., 2012. *The Airplace Indoor Positioning Platform for Android Smartphones,* Washington, DC, USA ©2012 : IEEE Computer Society .

Lion Vision Info Tech Solutions, 2013. *Development Process.* [Online]
Available at: http://www.lionvisionits.com/Development-Process.aspx
[Accessed 18 Oct 2014].

National Geographic Society, 1996-2014. *Triangulation.* [Online]
Available at: education.nationalgeographic.com/education/media/triangulation/?ar_a=1
[Accessed 15 Oct 2014].

Pacha, A., 2013 Sep 21. *What's the best 3D angular co-ordinate system for working with smartphone apps.* [Online]
Available at: http://math.stackexchange.com/questions/381649/whats-the-best-3d-angular-co-ordinate-system-for-working-with-smartphone-apps
[Accessed 16 Oct 2014].

Pu, C. H., 2011. *Emerging communications for wireless sensor networks,* China: InTech.

Robin, L., 2011. *Consumer electronics turn to MEMS for gesture control, precision location.* [Online]
Available at: http://www.otg-tech.com/consumer-electronics-turn-to-mems-for-gesture-control-precision-location/
[Accessed 12 Oct 2014].

Schneider, D., 2013. *New Indoor Navigation Technologies.* [Online]
Available at: http://spectrum.ieee.org/telecom/wireless/new-indoor-navigation-technologies-work-where-gps-cant

Schutzberg, A., 2013. *Ten Things You Need To Know About Indoor Positioning.* [Online]
Available at: http://www.directionsmag.com/articles/10-things-you-need-to-know-about-indoor-positioning/324602

Strategy Analytics, 2014. [Online]
Available at: http://www.strategyanalytics.com/

sublime1184, Nov 18 2009. *modmymobile.* [Online]
Available at: http://modmymobile.com/forums/562-motorola-cliq-general/524986-setup-android-sdk-jdk.html
[Accessed 30 March 2014].

The Nielsen Company, 2012-12-20. *Nielsen Tops of 2012: Digital.* [Online]
Available at: http://www.nielsen.com/us/en/insights/news/2012/nielsen-tops-of-2012-digital.html
[Accessed 18 Oct 2014].

Tisdale, S., 2014. *Wi-Fi Signal Strength Testing Using Smartphone Apps.* [Online]
Available at: blog.adiglobal.us/wi-fi-signal-strength-testing-using-smartphone-apps/

WiFiSLAM, 2009. *WiFiSLAM Indoor GPS.* [Online]
Available at: https://angel.co/wifislam
[Accessed 14 March 2014].

Wood,am, O. J., 2007. *An introduction to inertial navigation.,* 15 JJ Thomson Avenue, United Kingdom: University of Cambridge.

Zhu, J. Y. & Zheng, A. X., 2014. *Spatio-temporal (S-T) Similarity Model for Constructing WIFI-based RSSI Fingerprinting Map for Indoor Localization.* [Online]
Available at: http://www.ipin2014.org/wp/SessionView.asp?code=8C&idx=1569953031
[Accessed 15 Nov 2014].

7 Appendix:

7.1 ACRONYMS:

AP	Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identifier
CCU	Cell Controller Unit
dBm	Power Ratio in Decibels
GHz	Giga Hertz
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineering
m/s	Meters per Second
PDA	Personal Digital Assistant
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SDK	Standard Development Kit
UI	User Interface
URL	Uniform Resource Locator
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

7.2 Source codes:

7.2.1 XML Files:

7.2.1.1 Wi-Fi Fingerprint:

activity_main.xml:

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".ScannerActivity" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:orientation="horizontal" >

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Set X Co-ordinate: "
                android:textAppearance="?android:attr/textAppearanceMedium"
/>

            <EditText
                android:id="@+id/X_CO"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:inputType="numberDecimal" >

                <requestFocus />
            </EditText>
        </LinearLayout>

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:orientation="horizontal" >

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Set Y Co-ordinate: "
```

```xml
                android:textAppearance="?android:attr/textAppearanceMedium"
/>

        <EditText
            android:id="@+id/Y_CO"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:inputType="numberDecimal" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:orientation="horizontal" >

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingTop="20dp"
            android:text="No Of Line: "
            android:textAppearance="?android:attr/textAppearanceMedium"
/>

        <TextView
            android:id="@+id/counter"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0"
            android:textAppearance="?android:attr/textAppearanceLarge"
/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="20dp" >

        <RadioGroup
            android:id="@+id/rGroup"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:paddingBottom="20dp" >

            <RadioButton
                android:id="@+id/AP1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="AP1" />

            <RadioButton
                android:id="@+id/AP2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="AP2" />

            <RadioButton
```

```xml
                android:id="@+id/AP3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="AP3" />
        </RadioGroup>
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal" >

    <Button
        android:id="@+id/scanBut"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Start" />

    <Button
        android:id="@+id/stopBut"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Stop" />

    <Button
        android:id="@+id/startReadActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Read" />
</LinearLayout>

</LinearLayout>
```

read_file.xml:

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical" >

            <TextView
                android:id="@+id/showText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:ems="10"
                android:text="Data" />

            <Button
                android:id="@+id/readBut"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Read File" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:paddingTop="20dp" >

                <RadioGroup
                    android:id="@+id/radioGroup1"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content" >

                    <RadioButton
                        android:id="@+id/AP1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:onClick="chooseFile"
                        android:text="AP1 File" />

                    <RadioButton
                        android:id="@+id/AP2"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:onClick="chooseFile"
                        android:text="AP2 File" />
```

```xml
            <RadioButton
                android:id="@+id/AP3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="chooseFile"
                android:text="AP3 File" />
        </RadioGroup>
    </LinearLayout>

    </LinearLayout>
  </ScrollView>

</LinearLayout>
```

Wi-Fi Fingerprint Manifest:

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.average"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.average.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity
            android:name=".ReadFile"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="com.example.average.ReadFile" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

7.2.1.2   WifiNavigation:

Find_position.xml:

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".FindPosition" >

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingBottom="20dp"
            android:text="Select APs:"
            android:textAppearance="?android:attr/textAppearanceMedium" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="20dp" >

        <RadioGroup
            android:id="@+id/radioGroup"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal" >

            <RadioButton
                android:id="@+id/AP1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:onClick="chooseAPs"
                android:text="AP1" />

            <RadioButton
                android:id="@+id/AP2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:onClick="chooseAPs"
                android:text="AP2" />

            <RadioButton
                android:id="@+id/AP3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```

```xml
                android:layout_weight="1"
                android:onClick="chooseAPs"
                android:text="AP3" />
        </RadioGroup>
    </LinearLayout>

    <TextView
        android:id="@+id/scanCounter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingBottom="20dp"
        android:text="Scan Counter: "
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingBottom="20dp"
            android:text="Average Signals:"
            android:textAppearance="?android:attr/textAppearanceMedium" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingBottom="20dp" >

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=" AP1:  "
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <!--
             <TextView
            android:id="@+id/ap1Signal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="-00"
            android:textAppearance="?android:attr/textAppearanceMedium" -->
        <EditText
            android:id="@+id/AP1Signal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:editable="false"
            android:text="-0" >
        </EditText>

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```xml
        android:text=" AP2:  "
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <!--
         <TextView
        android:id="@+id/ap2Signal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="-00"
        android:textAppearance="?android:attr/textAppearanceMedium" /
    -->

    <EditText
        android:id="@+id/AP2Signal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:editable="false"
        android:text="-0" >
    </EditText>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" AP3:  "
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <!--
        <TextView
        android:id="@+id/ap3Signal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="-00"
        android:textAppearance="?android:attr/textAppearanceMedium" /
    -->

    <EditText
        android:id="@+id/AP3Signal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:editable="false"
        android:text="-0" >
    </EditText>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingBottom="20dp"
        android:text="Co-ordinates:"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

```xml
<Button
    android:id="@+id/getAxis"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:text="Get X and Y Axis" />

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:orientation="horizontal"
    android:paddingBottom="20dp" >

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="X-Axis: "
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/x_axis"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="0.0"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Y-Axis: "
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/y_axis"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="0.0"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>

<Button
    android:id="@+id/find_On_Map"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:text="Find On Map" />

</LinearLayout>
```

activity_main.xml:

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="98dp"
        android:layout_marginTop="184dp"
        android:text="MapActivty" />

</RelativeLayout>
```

WifiNavigation_Find Manifest:

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.wifinavigation_find"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.wifinavigation_find.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity
            android:name="com.example.wifinavigation_find.FindPosition"
            android:label="@string/title_activity_find_position" >

            <intent-filter>
                <action
android:name="com.example.wifinavigation_find.FINDPOSITION" />
```

```xml
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>

        </activity>
        <activity
            android:name="com.example.wifinavigation_find.MapActivity"
            android:label="@string/title_activity_map" >
            <intent-filter>
                <action
android:name="com.example.wifinavigation_find.MAPACTIVITY" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>

        </activity>
    </application>

</manifest>
```

7.2.2   Java Files:

7.2.2.1   Wi-Fi Fingerprint:

MainActivity.java:

package com.example.average;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.List;

import com.example.average.R;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.net.wifi.ScanResult;

import android.net.wifi.WifiManager;

import android.os.Bundle;

import android.os.Handler;

import android.util.Log;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.RadioGroup.OnCheckedChangeListener;

import android.widget.TextView;

import android.widget.Toast;

```java
public class MainActivity extends Activity {

        WifiManager wifiManager;

        List<ScanResult> wifiScanList;

        TextView counterText;

        EditText x_co, y_co;

        String x_axis, y_axis;

        Button scanBut, stopBut, readFile;

        RadioGroup APS;

        String BSSID;

        File folder, dbFile;

        public Handler scanHandler;

        boolean run = true;

        int count = 0;

        int RSS = 0;

        int AvgRSS = 0;

        @Override
        protected void onCreate(Bundle savedInstanceState) {

                super.onCreate(savedInstanceState);

                setContentView(R.layout.activity_main);

                // calling wifi scanning service

                wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);


                folder = getExternalFilesDir("FingerPrints");


                // counting no of line inserted in file

                counterText = (TextView) findViewById(R.id.counter);

                // EditText where x and y co-ordinates are entered.

                x_co = (EditText) findViewById(R.id.X_CO);
```

```
y_co = (EditText) findViewById(R.id.Y_CO);

APS = (RadioGroup) findViewById(R.id.rGroup);

// button to start scanning

scanBut = (Button) findViewById(R.id.scanBut);

// button to stop scanning

stopBut = (Button) findViewById(R.id.stopBut);

// button to start ReadFile activity

readFile = (Button) findViewById(R.id.startReadActivity);

// calling scanHandler class

scanHandler = new Handler();

// scanHandler.postDelayed(scanTimer, 3000); // 3000 is for 3 seconds

// addValues();

scanBut.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {

                run = true;

                scanHandler.postDelayed(scanTimer, 1000); // 3000 is for 3

//seconds

                Toast.makeText(getApplicationContext(), "Scanning Started",

                        Toast.LENGTH_SHORT).show();

                Log.d("Start", "Scan started");

        }

});


stopBut.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {

                run = false;

                count = 0;

                RSS = 0;

                Toast.makeText(getApplicationContext(), "Scanning Stopped",

                        Toast.LENGTH_SHORT).show();
```

```
                Log.d("Stop", "Scan Stopped");

            }

        });


    readFile.setOnClickListener(new OnClickListener() {

        @Override

        public void onClick(View arg0) {

            Intent intent = new Intent("com.example.average.ReadFile");

            startActivity(intent);

        }

    });

    APS.setOnCheckedChangeListener(new OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(RadioGroup group, int Id) {

            switch (Id) {

            case R.id.AP1:

                BSSID = "00:15:6d:5e:ec:7e";

                dbFile = new File(folder, "AP1.txt");


                Toast.makeText(getApplicationContext(), "AP1
checked",
        Toast.LENGTH_SHORT).show();

                break;

            case R.id.AP2:

                BSSID = "f8:d1:11:8c:16:42";

                dbFile = new File(folder, "AP2.txt");

                Toast.makeText(getApplicationContext(), "AP2
checked",

                    Toast.LENGTH_SHORT).show();

                break;
```

```
                    case R.id.AP3:

                            BSSID = "f8:d1:11:79:52:90";

                            dbFile = new File(folder, "AP3.txt");

                            Toast.makeText(getApplicationContext(), "AP3
checked",
        Toast.LENGTH_SHORT).show();

                            break;

                    }

            }

        });

    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.activity_main, menu);
            return true;
    }


        public int onReceive() {
        wifiManager.startScan();
        wifiScanList = wifiManager.getScanResults();
        int ListSize = wifiScanList.size();
        int allvalues = 0;
        for (int i = 0; i < ListSize; i++) {
                String ssid = wifiScanList.get(i).SSID;
                String bssid = wifiScanList.get(i).BSSID;
                int intFrequency = wifiScanList.get(i).frequency;
                int intLevel = wifiScanList.get(i).level;
                String frequency = String.valueOf(intFrequency);
```

```java
                String leveldBm = String.valueOf(intLevel);

                double doubleDistance = calculateDistance(intLevel, intFrequency);

                String distance = String.format("%.05s m",

                        String.valueOf(doubleDistance));

                // Log.d("bssid", bssid);

                if (bssid.equals(BSSID)) {

                        // allvalues = (leveldBm + " | " + ssid + " | " + bssid);

                        Log.d("BSSID", bssid);

                        Log.d("level okok", leveldBm);

                        allvalues = intLevel;

                }

        }

        return allvalues;

}




public void addValues(int avgRSS) {

        x_axis = x_co.getText().toString();

        y_axis = y_co.getText().toString();

        String data = String.valueOf(avgRSS) + " | " + x_axis + " | " + y_axis

                + " | " + BSSID;

        // File folder = getExternalFilesDir("Scanner");

        // File dbFile = new File(folder, "data.txt");

        writeData(dbFile, data);

}

public void writeData(File dbFile, String data) {

        FileWriter fos = null;

        try {
```

```java
        fos = new FileWriter(dbFile, true);

        BufferedWriter bw = new BufferedWriter(fos);

        Message.message(

                this,

                data + "was written Successfully--"

                        + dbFile.getAbsolutePath());

        bw.write(data);

        bw.newLine();

        bw.close();

    } catch (FileNotFoundException e) {

        e.printStackTrace();

    } catch (IOException e) {

        e.printStackTrace();

    } finally {

        if (fos != null) {

            try {

                fos.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }


}

private Runnable scanTimer = new Runnable() {

    @Override

    public void run() {

        if (run) {

            if (count != 20) {
```

```
                count++;

                counterText.setText(String.valueOf(count));

                int dBm = Math.abs(onReceive());

                Log.d("DBM", String.valueOf(dBm));

                RSS = RSS + dBm;

                Log.d("RSS :", String.valueOf(RSS));

        } else {

                AvgRSS = RSS / 20;

                Log.d("Average", String.valueOf(AvgRSS));

                addValues(AvgRSS);

                run = false;

                count = 0;

                RSS = 0;

        }

        scanHandler.postDelayed(scanTimer, 1000);

    }

}

};

// Here signalLevel is in dBm and frequency is in MHz respectively

public double calculateDistance(double level, double frq) {

        double exp = (27.55 - (20 * Math.log10(frq)) + Math.abs(level)) / 20.0;

        return Math.pow(10.0, exp);

}

}
```

Message.java:

```
package com.example.average;

import android.content.Context;

import android.widget.Toast;

public class Message {
```

```
public static void message(Context context, String message){

        Toast.makeText(context, message, Toast.LENGTH_SHORT).show();

    }

}
```

ReadFile.java:

```java
package com.example.average;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import com.example.average.R;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.TextView;

import android.widget.Toast;

public class ReadFile extends Activity {

        TextView showText;

        Button readBut;

        File folder, dbFile;

        @Override

        protected void onCreate(Bundle savedInstanceState) {

                super.onCreate(savedInstanceState);

                setContentView(R.layout.read_file);

                showText = (TextView) findViewById(R.id.showText);

                readBut = (Button) findViewById(R.id.readBut);
```

```java
                folder = getExternalFilesDir("FingerPrints");

                readBut.setOnClickListener(new OnClickListener() {

                        @Override

                        public void onClick(View arg0) {

                                readFromFile();

                        }

                });

        }

        public void readFromFile() {

                if (dbFile.exists()) {

                        StringBuilder text = new StringBuilder();

                        try {

                                BufferedReader br = new BufferedReader(new
FileReader(dbFile));

                                String line;

                                while ((line = br.readLine()) != null) {

                                        // display line starting with particular string at starting

                                        // if (line.startsWith("-38")){

                                        // display line starting with particular string at end

                                        // if(line.endsWith("4")){

                                        // text.append(line);

                                        // text.append('\n');

                                        // int length = line.length();

                                        // Log.d("Length", String.valueOf(length));

                                        // }

                                        text.append(line);

                                        text.append('\n');

                                }

                        } catch (IOException e) {
```

```
                    }

                    // set the edittext with the text of file

                    showText.setText(text);

            } else {

                    showText.setText("Sorry file does not exists");

            }

    }


    public void chooseFile(View v) {

            boolean checked = ((RadioButton) v).isChecked();

            switch (v.getId()) {

            case R.id.AP1:

                    if (checked) {

                            dbFile = new File(folder, "AP1.txt");

                            Toast.makeText(getApplicationContext(), "AP1 checked",

                                    Toast.LENGTH_SHORT).show();

    }

                    break;

            case R.id.AP2:

                    if (checked) {

                            dbFile = new File(folder, "AP2.txt");

                            Toast.makeText(getApplicationContext(), "AP2 checked",

                                    Toast.LENGTH_SHORT).show();

                    }

                    break;


            case R.id.AP3:

                    if (checked) {

                            dbFile = new File(folder, "AP3.txt");
```

```
                        Toast.makeText(getApplicationContext(), "AP3 checked",

                                Toast.LENGTH_SHORT).show();

                }

                break;

        }

    }

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        // Inflate the menu; this adds items to the action bar if it is present.

        getMenuInflater().inflate(R.menu.activity_main, menu);

        return true;

    }

}
```

7.2.2.2 WifiNavigation:

FindPosition.java:

package com.example.wifinavigation_find;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.util.List;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.net.wifi.ScanResult;

import android.net.wifi.WifiManager;

import android.os.Bundle;

import android.os.Handler;

import android.util.Log;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.RadioGroup.OnCheckedChangeListener;

import android.widget.TextView;

import android.widget.Toast;

public class FindPosition extends Activity {

   Button findOnMap, getAxis;

   RadioGroup APS;

```java
RadioButton AP1, AP2, AP3;

WifiManager wifiManager;

List<ScanResult> wifiScanList;

String BSSID, X_AXIS, Y_AXIS;

Timer scanTimer;

public Handler scanHandler;

TextView scanCounter, ap1Signal, ap2Signal, ap3Signal, x_axis, y_axis;

EditText AverageSignals, AP1Signal, AP2Signal, AP3Signal;

boolean run = true;

int count = 0;

int RSS = 0;

int AvgRSS = 0;

File folder, AP1File, AP2File, AP3File;

@Override
protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.find_position);

        initialize();

        // methods for all button clicks

        clickListener();

}

public void initialize() {

        // This button starts MapActivity.java

        findOnMap = (Button) findViewById(R.id.find_On_Map);

        getAxis = (Button) findViewById(R.id.getAxis);

        APS = (RadioGroup) findViewById(R.id.radioGroup);

        // calling wifi scanning service

        wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
```

```java
        // setting times for scanning

        scanTimer = new Timer();

        scanHandler = new Handler();

        // all text views

        scanCounter = (TextView) findViewById(R.id.scanCounter);

        x_axis = (TextView) findViewById(R.id.x_axis);

        y_axis = (TextView) findViewById(R.id.y_axis);

        // file handling

        folder = getExternalFilesDir("FingerPrints");

        AP1File = new File(folder, "AP1.txt");

        AP2File = new File(folder, "AP2.txt");

        AP3File = new File(folder, "AP3.txt");

    }
    // this is a method for all the buttons clicked
    public void clickListener() {

        findOnMap.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View arg0) {

                Intent mapActivity = new Intent(
    "com.example.wifinavigation_find.MAPACTIVITY");

                startActivity(mapActivity);

            }

        });

        getAxis.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View arg0) {

                findStrongSignal();

            }

        });
```

```
        }
    public void chooseAPs(View v) {

            boolean checked = ((RadioButton) v).isChecked();

            switch (v.getId()) {

            case R.id.AP1:

                    if (!checked) {

                            Toast.makeText(getApplicationContext(),

                                            "Please choose Access Point.",
Toast.LENGTH_SHORT)

                                            .show();

                    } else {

                            BSSID = "90:f6:52:98:c4:a2";

                            Toast.makeText(getApplicationContext(), "AP1 checked",

                                            Toast.LENGTH_SHORT).show();

                            run = true;

                            scanHandler.postDelayed(scanTimer, 1000); // 3000 is for 3

            // seconds

                            Toast.makeText(getApplicationContext(), "Scanning Started",

                                            Toast.LENGTH_SHORT).show();

                            Log.d("Start", "Scan started");

                            AverageSignals = (EditText) findViewById(R.id.AP1Signal);

                    }

                    break;

            case R.id.AP2:

                    if (!checked) {

                            Toast.makeText(getApplicationContext(),

                                            "Please choose Access Point.",
Toast.LENGTH_SHORT)

                                            .show();
```

```java
            } else {

                    BSSID = "90:f6:52:98:c4:a2";

                    Toast.makeText(getApplicationContext(), "AP2 checked",

                            Toast.LENGTH_SHORT).show();

                    run = true;

                    scanHandler.postDelayed(scanTimer, 1000); // 3000 is for 3

        // seconds

                    Toast.makeText(getApplicationContext(), "Scanning Started",

                            Toast.LENGTH_SHORT).show();

                    Log.d("Start", "Scan started");


                    AverageSignals = (EditText) findViewById(R.id.AP2Signal);

            }

            break;

        case R.id.AP3:

            BSSID = "f8:d1:11:79:52:90";

            Toast.makeText(getApplicationContext(), "AP3 checked",

                    Toast.LENGTH_SHORT).show();

            run = true;

            scanHandler.postDelayed(scanTimer, 1000); // 3000 is for 3

    // seconds

            Toast.makeText(getApplicationContext(), "Scanning Started",

                    Toast.LENGTH_SHORT).show();

            Log.d("Start", "Scan started");


            AverageSignals = (EditText) findViewById(R.id.AP3Signal);

            break;
```

```java
        }
}
// This method scans signal from access points returns signal level
public int onReceive() {
        wifiManager.startScan();
        wifiScanList = wifiManager.getScanResults();
        int ListSize = wifiScanList.size();
        int allvalues = 0;
        for (int i = 0; i < ListSize; i++) {
                String ssid = wifiScanList.get(i).SSID;
                String bssid = wifiScanList.get(i).BSSID;
                int intFrequency = wifiScanList.get(i).frequency;
                int intLevel = wifiScanList.get(i).level;
                String frequency = String.valueOf(intFrequency);
                String leveldBm = String.valueOf(intLevel);
                if (bssid.equals(BSSID)) {
                        // allvalues = (leveldBm + " | " + ssid + " | " + bssid);


                        Log.d("BSSID", bssid);
                        Log.d("level okok", leveldBm);
                        allvalues = intLevel;
                }


        }
        return allvalues;
}
private class Timer implements Runnable {
        @Override
        public void run() {
```

```
if (run) {

    if (count != 5) {

        count++;

        scanCounter

            .setText("Scan Counter:" +
String.valueOf(count));


        int dBm = Math.abs(onReceive());

        Log.d("DBM", String.valueOf(dBm));

        RSS = RSS + dBm;

        Log.d("RSS :", String.valueOf(RSS));


    } else {

        AvgRSS = RSS / 5;

        Log.d("Average", String.valueOf(AvgRSS));

        allAverageValues(AvgRSS);

        run = false;

        count = 0;

        RSS = 0;

    }

    scanHandler.postDelayed(scanTimer, 1000);

    }

}

private void allAverageValues(int avgRSS) {

    AverageSignals.setText("-" + String.valueOf(avgRSS));

}

}


public void findStrongSignal() {
```

```
AP1Signal = (EditText) findViewById(R.id.AP1Signal);

AP2Signal = (EditText) findViewById(R.id.AP2Signal);

AP3Signal = (EditText) findViewById(R.id.AP3Signal);

int apsig1 = Math.abs(Integer.parseInt(AP1Signal.getText().toString()));

int apsig2 = Math.abs(Integer.parseInt(AP2Signal.getText().toString()));

int apsig3 = Math.abs(Integer.parseInt(AP3Signal.getText().toString()));

Toast.makeText(getApplicationContext(), "THis is happening",

                Toast.LENGTH_SHORT).show();

if (apsig1 == 0 || apsig2 == 0 || apsig3 == 0) {

        Toast.makeText(getApplicationContext(), "Access Point missing",

                Toast.LENGTH_SHORT).show();

} else {

        Log.d("AP1Sig", String.valueOf(apsig1));

        // scanning from file

        if (AP1File.exists()) {

                String line;

                int AP1Sig = apsig1 - 1;

                for (int i = AP1Sig; i <= AP1Sig + 2; i++) {

                        Log.d("AP1Sigbbb", String.valueOf(i));

                        if (GetterSetter.x == 0 && GetterSetter.y == 0) {

                                try {

                                        BufferedReader br = new
BufferedReader(new FileReader(AP1File));

                                        while ((line = br.readLine()) != null) {

                                                // display line starting with
particular string


                                                if
(line.startsWith(String.valueOf(i))) {

                                                        Log.d("AP1Sig",
String.valueOf(i));
```

```
                    String[] oneFingerprintLine = line.split(" | ");

                    X_AXIS = oneFingerprintLine[2];

                    Y_AXIS = oneFingerprintLine[4];

                    x_axis.setText(X_AXIS);

                    y_axis.setText(Y_AXIS);

            GetterSetter.x = Float.parseFloat(X_AXIS);

            GetterSetter.y = Float.parseFloat(Y_AXIS);


    Toast.makeText(getApplicationContext(),
                          X_AXIS + Y_AXIS, Toast.LENGTH_SHORT)
            .show();


        Log.d(X_AXIS, X_AXIS);

        Log.d(Y_AXIS, Y_AXIS);
break;

                                                    }

                                          }

                            } catch (IOException e) {

                            }

                    }

                }

            }

            else {

                    Toast.makeText(getApplicationContext(), "File not found",

                            Toast.LENGTH_SHORT).show();

            }

            if (AP2File.exists()) {

                    String line;

                    int AP2Sig = apsig2 - 1;

                    for (int i = AP2Sig; i <= AP2Sig + 2; i++) {

                            Log.d("AP2Sigbbb", String.valueOf(i));
```

```java
if (GetterSetter.x == 0 && GetterSetter.y == 0) {

            try {

            BufferedReader br = new BufferedReader(
                        new FileReader(AP2File));

                                    while ((line = br.readLine()) != null) {

                                        // display line starting with
particular string


                                    if
(line.startsWith(String.valueOf(i))) {

                                            Log.d("AP2Sig",
String.valueOf(i));

                                            String[]

                                                .split(" | ");


                                            X_AXIS =
oneFingerprintLine[2];

                                            Y_AXIS =
oneFingerprintLine[4];



                                            x_axis.setText(X_AXIS);
                                            y_axis.setText(Y_AXIS);


                                            GetterSetter.x =
Float.parseFloat(X_AXIS);

                                            GetterSetter.y =
Float.parseFloat(Y_AXIS);



        Toast.makeText(getApplicationContext(),

                                                X_AXIS +
Y_AXIS, Toast.LENGTH_SHORT)

                                                .show();
```

oneFingerprintLine = line

```
                                                    Log.d(X_AXIS,
X_AXIS);

                                                    Log.d(Y_AXIS,
Y_AXIS);


                                                    break;
                                              }
                                        }
                                } catch (IOException e) {


                                }


                        }


                }


                // set the edittext with the text of file}catch (IOException e){


        } else {
                Toast.makeText(getApplicationContext(), "File not found",
                        Toast.LENGTH_SHORT).show();
        }
        if (AP3File.exists()) {
                String line;
                int AP3Sig = apsig3 - 1;


                for (int i = AP3Sig; i <= AP3Sig + 2; i++) {
                        Log.d("AP3Sigbbb", String.valueOf(i));
                        if (GetterSetter.x == 0 && GetterSetter.y == 0) {
```

```java
try {
    BufferedReader br = new BufferedReader(
            new FileReader(AP3File));
    while ((line = br.readLine()) != null) {
        // display line starting with particular string
        if (line.startsWith(String.valueOf(i))) {
            Log.d("AP3Sig", String.valueOf(i));
            String[] oneFingerprintLine = line
                    .split(" | ");

            X_AXIS = oneFingerprintLine[2];
            Y_AXIS = oneFingerprintLine[4];

            x_axis.setText(X_AXIS);
            y_axis.setText(Y_AXIS);

            GetterSetter.x = Float.parseFloat(X_AXIS);
            GetterSetter.y = Float.parseFloat(Y_AXIS);

            Toast.makeText(getApplicationContext(),
                    X_AXIS + Y_AXIS, Toast.LENGTH_SHORT)
```

```
                                                                .show();

                                                     Log.d(X_AXIS,
X_AXIS);

                                                     Log.d(Y_AXIS,
Y_AXIS);


                                         break;


                            }
                        }
                    } catch (IOException e) {


                    }


                }


            }


            // set the edittext with the text of filecatch (IOException e){


        } else {
            Toast.makeText(getApplicationContext(), "File not found",
                Toast.LENGTH_SHORT).show();
        }


    }
}


@Override
```

```java
public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action bar if it is present.

    getMenuInflater().inflate(R.menu.find_position, menu);

    return true;

}

}
```

GetterSetter.java:

```java
package com.example.wifinavigation_find;

public class GetterSetter {

    public static float x = 0, y = 0;

}
```

MainActivity.java:

```java
package com.example.wifinavigation_find;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.view.Menu;

public class MainActivity extends Activity {

  @Override

  protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    Intent findPosition = new Intent(

                        "com.example.wifinavigation_find.FINDPOSITION");

            startActivity(findPosition);

  }

  @Override

  public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action bar if it is present.
```

```
    getMenuInflater().inflate(R.menu.activity_find_position, menu);

    return true;

  }

  @Override

    protected void onPause() {

        // TODO Auto-generated method stub

        super.onPause();

        finish();

    }

}
```

MapActivity.java:

```
package com.example.wifinavigation_find;

import android.app.Activity;

import android.content.Context;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Matrix;

import android.graphics.Paint;

import android.hardware.Sensor;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener;

import android.hardware.SensorManager;

import android.net.wifi.WifiManager;

import android.os.Bundle;

import android.os.Handler;

import android.util.Log;

import android.view.Menu;
```

```java
import android.view.MenuItem;

import android.view.MotionEvent;

import android.view.SurfaceHolder;

import android.view.SurfaceView;

public class MapActivity extends Activity implements SensorEventListener {

        public WifiManager wifiManager;

        private float currentDegree = 0f;

        float degree = 0f;

        private long lastUpdate = 0;

        private float last_x, last_y, last_z;

        private static final float SHAKE_THRESHOLD = 600;

        private float[] gravity = new float[3];

        public Handler scanHandler = new Handler();

        static int step = 0;

        // device sensor manager

        private SensorManager mSensorManager;

        OurView view;

        Canvas canvas;

        Bitmap arrow, miniArrow, backgroundMap, tempMap;

        float x_axis, y_axis, xS, yS;

        @Override

        protected void onCreate(Bundle savedInstanceState) {

                super.onCreate(savedInstanceState);

                view = new OurView(this);

                wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);

                tempMap = BitmapFactory.decodeResource(getResources(),
R.drawable.map);

                arrow = BitmapFactory.decodeResource(getResources(),

                        R.drawable.the_arrow);

                x_axis = GetterSetter.x;
```

```java
        y_axis = GetterSetter.y;

        xS = 20;

        yS = 20;

        setContentView(view);

        // initialize your android device sensor capabilities

        mSensorManager = (SensorManager)
getSystemService(SENSOR_SERVICE);

    }

    @Override

    protected void onPause() {

        // TODO Auto-generated method stub

        super.onPause();

        // to stop the listener and save battery

        mSensorManager.unregisterListener(this);

        view.pause();

    }

    @Override

    protected void onResume() {

        super.onResume();

        // for the system's orientation sensor registered listeners

        mSensorManager.registerListener(this,

mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION),

                SensorManager.SENSOR_DELAY_NORMAL);

        mSensorManager.registerListener(this,

mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),

                SensorManager.SENSOR_DELAY_NORMAL);

        view.resume();

    }
```

```
@Override

public boolean onTouchEvent(MotionEvent e) {

        x_axis = e.getX();

        y_axis = e.getY() - 100f;

        return true;

}

public class OurView extends SurfaceView implements Runnable {

        Thread thread = null;

        SurfaceHolder holder;

        boolean run = false;

        public OurView(Context context) {

                super(context);

                holder = getHolder();

        }

        @Override

        public void run() {

                while (run == true) {

                        // perform canvas drawing

                        if (!holder.getSurface().isValid()) {

                                continue;

                        }

                        canvas = holder.lockCanvas();

                        Paint paint = new Paint();

                        paint.setColor(Color.RED);

                        paint.setTextSize(20);

                        // canvas.drawARGB(0, 0, 0, 0);


                        backgroundMap = Bitmap.createScaledBitmap(tempMap,
```

```
                    canvas.getWidth(), canvas.getHeight(), true);

            canvas.drawBitmap(backgroundMap, 0, 0, null);

            miniArrow = Bitmap.createScaledBitmap(arrow, 25, 25, true);

            Matrix matrix = new Matrix();

            matrix.setTranslate(x_axis - (miniArrow.getWidth() / 2), y_axis
                        - (miniArrow.getHeight() / 2));

            matrix.preRotate(currentDegree, miniArrow.getWidth() / 2,
                        miniArrow.getHeight() / 2);

            // canvas.drawText("x is " + x_axis + ": y is " + y_axis +"D: "
            // + currentDegree,
            // x_axis + 20, y_axis + 20, paint);
            canvas.drawText("x is " + x_axis, x_axis + 20, y_axis + 20,
                        paint);
            canvas.drawText("y is " + y_axis, x_axis + 20, y_axis + 40,
                        paint);
            canvas.drawText("" + currentDegree, x_axis + 20, y_axis + 60,
                        paint);
            canvas.drawText("Steps: " + step, x_axis + 20, y_axis + 80,
                        paint);
            // canvas.drawCircle(x_axis, y_axis, 10f, paint);
            canvas.drawBitmap(miniArrow, matrix, paint);
            holder.unlockCanvasAndPost(canvas);
        }
    }
    public void pause() {
        run = false;
        while (true) {
            try {
                thread.join();
```

```
                        backgroundMap.recycle();

                        miniArrow.recycle();

                } catch (InterruptedException e) {

                        e.printStackTrace();

                }

                break;

        }

        thread = null;

}

public void resume() {

        run = true;

        thread = new Thread(this);

        thread.start();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

        // Inflate the menu; this adds items to the action bar if it is present.

        getMenuInflater().inflate(R.menu.activity_map, menu);

        return true;

}

@Override

public void onAccuracyChanged(Sensor arg0, int arg1) {

        // TODO Auto-generated method stub

}

@Override

public void onSensorChanged(SensorEvent event) {

        Sensor mySensor = event.sensor;

        currentDegree = getDirection(event);
```

```java
// Log.d("Angle", String.valueOf(direction));

if (mySensor.getType() == Sensor.TYPE_ACCELEROMETER) {

        final float alpha = 0.8f;

        gravity[0] = alpha * gravity[0] + (1 - alpha) * event.values[0];

        gravity[1] = alpha * gravity[1] + (1 - alpha) * event.values[1];

        gravity[2] = alpha * gravity[2] + (1 - alpha) * event.values[2];

        float x = event.values[0] - gravity[0];

        float y = event.values[1] - gravity[1];

        float z = event.values[2] - gravity[2];

        long currTime = System.currentTimeMillis();

        if ((currTime - lastUpdate) > 100) {

                long diffTime = (currTime - lastUpdate);

                lastUpdate = currTime;

                float speed = Math.abs(x + y + z - last_x - last_y - last_z)

                            / diffTime * 40000;

                if (speed > SHAKE_THRESHOLD) {


                        if (x_axis > view.getWidth() - miniArrow.getWidth()) {

                                x_axis = view.getWidth() -

miniArrow.getWidth();

                        }

                        if (y_axis > view.getHeight()) {

                                y_axis = view.getHeight();

                        }

                        if (x_axis + xS <= 0) {

                                x_axis = 0;

                                xS = 0;

                        }


                        if (y_axis + yS <= 0) {
```

```
                y_axis = 0;
                yS = 0;
        }


        // for north
        if (currentDegree >= 0 && currentDegree < 90) {


                if (currentDegree >= 0 && currentDegree < 45)
{


                        x_axis -= 2.5;
                        y_axis += 5;


                } else {
                        x_axis -= 5;
                        y_axis += 2.5;
                }
                // for East
        } else if (currentDegree > 90 && currentDegree < 180)
{


                if (currentDegree >= 90 && currentDegree <
135) {


                        x_axis -= 5;
                        y_axis -= 2.5;


                } else {
                        x_axis -= 2.5;
                        y_axis -= 5;
```

```
                }
                // for South
            } else if (currentDegree > 180 && currentDegree < 70)
{

                if (currentDegree >= 180 && currentDegree <
225) {

                    x_axis += 2.5;

                    y_axis -= 5;

                } else {

                    x_axis += 5;

                    y_axis -= 2.5;

                }
                // for West
            } else if (currentDegree > 270 && currentDegree <=
360) {

                if (currentDegree >= 270 && currentDegree <
315) {

                    x_axis += 5;

                    y_axis += 2.5;

                } else {

                    x_axis += 2.5;

                    y_axis += 5;

                }
            } else {
            }step++;
        }
        last_x = x;
        last_y = y;
        last_z = z;
```

```
                }}
        }


        private float getDirection(SensorEvent event) {
                Sensor compassSensor = event.sensor;
                if (compassSensor.getType() == Sensor.TYPE_ORIENTATION) {
                        // get the angle around the z-axis rotated
                        degree = Math.round(event.values[0]);
                }
                return degree;
        }
}
```

7.2.3 Fingerprinting file:

AP1.txt:

78 | 142 | 106 | 00:15:6d:5e:ec:7e

81 | 190 | 106 | 00:15:6d:5e:ec:7e

78 | 142 | 133 | 00:15:6d:5e:ec:7e

58 | 330 | 645 | 00:15:6d:5e:ec:7e

54 | 375 | 645 | 00:15:6d:5e:ec:7e

58 | 375 | 620 | 00:15:6d:5e:ec:7e

65 | 330 | 620 | 00:15:6d:5e:ec:7e

62 | 295 | 620 | 00:15:6d:5e:ec:7e

68 | 295 | 590 | 00:15:6d:5e:ec:7e

61 | 330 | 590 | 00:15:6d:5e:ec:7e

57 | 370 | 590 | 00:15:6d:5e:ec:7e

59 | 370 | 540 | 00:15:6d:5e:ec:7e

65 | 330 | 540 | 00:15:6d:5e:ec:7e

72 | 295 | 540 | 00:15:6d:5e:ec:7e

77 | 295 | 490 | 00:15:6d:5e:ec:7e

67 | 330 | 490 | 00:15:6d:5e:ec:7e

65 | 370 | 490 | 00:15:6d:5e:ec:7e

63 | 370 | 445 | 00:15:6d:5e:ec:7e

65 | 330 | 445 | 00:15:6d:5e:ec:7e

67 | 295 | 445 | 00:15:6d:5e:ec:7e

73 | 295 | 400 | 00:15:6d:5e:ec:7e

67 | 330 | 400 | 00:15:6d:5e:ec:7e

64 | 370 | 400 | 00:15:6d:5e:ec:7e

66 | 370 | 375 | 00:15:6d:5e:ec:7e

65 | 330 | 375 | 00:15:6d:5e:ec:7e

77 | 295 | 375 | 00:15:6d:5e:ec:7e

62 | 375 | 350 | 00:15:6d:5e:ec:7e

66 | 330 | 350 | 00:15:6d:5e:ec:7e

69 | 330 | 300 | 00:15:6d:5e:ec:7e

65 | 370 | 300 | 00:15:6d:5e:ec:7e

80 | 265 | 360 | 00:15:6d:5e:ec:7e

84 | 235 | 360 | 00:15:6d:5e:ec:7e

85 | 190 | 360 | 00:15:6d:5e:ec:7e

87 | 142 | 360 | 00:15:6d:5e:ec:7e

77 | 142 | 290 | 00:15:6d:5e:ec:7e

87 | 190 | 290 | 00:15:6d:5e:ec:7e

86 | 235 | 290 | 00:15:6d:5e:ec:7e

87 | 265 | 290 | 00:15:6d:5e:ec:7e

85 | 265 | 205 | 00:15:6d:5e:ec:7e

86 | 235 | 205 | 00:15:6d:5e:ec:7e

83 | 190 | 205 | 00:15:6d:5e:ec:7e

81 | 142 | 205 | 00:15:6d:5e:ec:7e

88 | 142 | 175 | 00:15:6d:5e:ec:7e

84 | 190 | 175 | 00:15:6d:5e:ec:7e

83 | 190 | 133 | 00:15:6d:5e:ec:7e

85 | 142 | 133 | 00:15:6d:5e:ec:7e

AP2.txt:

79 | 142 | 106 | f8:d1:11:8c:16:42

72 | 190 | 106 | f8:d1:11:8c:16:42

73 | 142 | 133 | f8:d1:11:8c:16:42

73 | 330 | 645 | f8:d1:11:8c:16:42

78 | 375 | 645 | f8:d1:11:8c:16:42

77 | 375 | 620 | f8:d1:11:8c:16:42

75 | 330 | 620 | f8:d1:11:8c:16:42

78 | 295 | 620 | f8:d1:11:8c:16:42

75 | 295 | 590 | f8:d1:11:8c:16:42

79 | 330 | 590 | f8:d1:11:8c:16:42

74 | 370 | 590 | f8:d1:11:8c:16:42

78 | 370 | 540 | f8:d1:11:8c:16:42

76 | 330 | 540 | f8:d1:11:8c:16:42

73 | 295 | 540 | f8:d1:11:8c:16:42

69 | 295 | 490 | f8:d1:11:8c:16:42

74 | 330 | 490 | f8:d1:11:8c:16:42

74 | 370 | 490 | f8:d1:11:8c:16:42

74 | 370 | 445 | f8:d1:11:8c:16:42

75 | 330 | 445 | f8:d1:11:8c:16:42

73 | 295 | 445 | f8:d1:11:8c:16:42

69 | 295 | 400 | f8:d1:11:8c:16:42

65 | 330 | 400 | f8:d1:11:8c:16:42

70 | 370 | 400 | f8:d1:11:8c:16:42

65 | 370 | 375 | f8:d1:11:8c:16:42

66 | 330 | 375 | f8:d1:11:8c:16:42

65 | 295 | 375 | f8:d1:11:8c:16:42

63 | 375 | 350 | f8:d1:11:8c:16:42

65 | 330 | 350 | f8:d1:11:8c:16:42

67 | 330 | 300 | f8:d1:11:8c:16:42

60 | 370 | 300 | f8:d1:11:8c:16:42

74 | 265 | 360 | f8:d1:11:8c:16:42

76 | 235 | 360 | f8:d1:11:8c:16:42

73 | 142 | 360 | f8:d1:11:8c:16:42

72 | 142 | 290 | f8:d1:11:8c:16:42

69 | 190 | 290 | f8:d1:11:8c:16:42

67 | 235 | 290 | f8:d1:11:8c:16:42

68 | 265 | 290 | f8:d1:11:8c:16:42

74 | 265 | 205 | f8:d1:11:8c:16:42

65 | 235 | 205 | f8:d1:11:8c:16:42

76 | 190 | 205 | f8:d1:11:8c:16:42

78 | 142 | 205 | f8:d1:11:8c:16:42

77 | 142 | 175 | f8:d1:11:8c:16:42

73 | 190 | 175 | f8:d1:11:8c:16:42

75 | 190 | 133 | f8:d1:11:8c:16:42

77 | 142 | 133 | f8:d1:11:8c:16:42

AP3.txt:

76 | 142 | 106 | f8:d1:11:79:52:90	59 | 330 | 375 | f8:d1:11:79:52:90
70 | 190 | 106 | f8:d1:11:79:52:90	58 | 295 | 375 | f8:d1:11:79:52:90
70 | 142 | 133 | f8:d1:11:79:52:90	57 | 375 | 350 | f8:d1:11:79:52:90
68 | 330 | 645 | f8:d1:11:79:52:90	62 | 330 | 350 | f8:d1:11:79:52:90
69 | 375 | 645 | f8:d1:11:79:52:90	63 | 330 | 300 | f8:d1:11:79:52:90
70 | 375 | 620 | f8:d1:11:79:52:90	66 | 370 | 300 | f8:d1:11:79:52:90
67 | 330 | 620 | f8:d1:11:79:52:90	70 | 265 | 360 | f8:d1:11:79:52:90
67 | 295 | 620 | f8:d1:11:79:52:90	71 | 235 | 360 | f8:d1:11:79:52:90
65 | 295 | 590 | f8:d1:11:79:52:90	75 | 190 | 360 | f8:d1:11:79:52:90
58 | 330 | 590 | f8:d1:11:79:52:90	74 | 190 | 360 | f8:d1:11:79:52:90
72 | 370 | 590 | f8:d1:11:79:52:90	77 | 142 | 360 | f8:d1:11:79:52:90
65 | 370 | 540 | f8:d1:11:79:52:90	73 | 142 | 290 | f8:d1:11:79:52:90
65 | 330 | 540 | f8:d1:11:79:52:90	72 | 190 | 290 | f8:d1:11:79:52:90
61 | 295 | 540 | f8:d1:11:79:52:90	83 | 235 | 290 | f8:d1:11:79:52:90
62 | 295 | 490 | f8:d1:11:79:52:90	81 | 265 | 290 | f8:d1:11:79:52:90
63 | 330 | 490 | f8:d1:11:79:52:90	81 | 265 | 205 | f8:d1:11:79:52:90
64 | 370 | 490 | f8:d1:11:79:52:90	81 | 235 | 205 | f8:d1:11:79:52:90
65 | 370 | 445 | f8:d1:11:79:52:90	82 | 190 | 205 | f8:d1:11:79:52:90
65 | 330 | 445 | f8:d1:11:79:52:90	82 | 142 | 205 | f8:d1:11:79:52:90
64 | 295 | 445 | f8:d1:11:79:52:90	81 | 142 | 175 | f8:d1:11:79:52:90
59 | 295 | 400 | f8:d1:11:79:52:90	77 | 190 | 175 | f8:d1:11:79:52:90
59 | 330 | 400 | f8:d1:11:79:52:90	81 | 190 | 133 | f8:d1:11:79:52:90
54 | 370 | 400 | f8:d1:11:79:52:90	79 | 142 | 133 | f8:d1:11:79:52:90
52 | 370 | 375 | f8:d1:11:79:52:90	

7.3 Gantt chart:

		Task Name	Duration	Start	End
1		Wi-Fi Based Indoor Navigation System for Android Device	207 days	2014-03-20	2015-01-02
2		⊟ Planning and approval of project	6 days	2014-03-20	2014-03-27
3	📅	Determining the project and its scope	3 days	2014-03-20	2014-03-24
4		Prepare project plan and specification	4 days	2014-03-24	2014-03-27
5		⊟ Initial study of the system	10 days	2014-03-28	2014-04-10
6		Detail study on Android platform architecture and its API	10 days	2014-03-28	2014-04-10
7		⊟ Analysis	36 days	2014-04-11	2014-05-30
8		Preparation of SRS document	3 days	2014-04-11	2014-04-15
9		Preparation of Interim report	7 days	2014-04-16	2014-04-24
10		Submission of interim project report	1 day	2014-04-25	2014-04-25
11		System Data Design	10 days	2014-04-28	2014-05-09
12		System Architecture Design	10 days	2014-05-12	2014-05-23
13		Documentation of design specification	5 days	2014-05-26	2014-05-30
14		⊟ Development	100 days	2014-06-02	2014-10-17
15		Development of wireless fingerprinting module	30 days	2014-06-02	2014-07-11
16		Development of navigation system module	40 days	2014-07-14	2014-09-05
17		Combining results of wireless fingerprinting and navigation s...	20 days	2014-09-08	2014-10-03
18		Implementation of combined system in android based device	10 days	2014-10-06	2014-10-17
19		⊟ Testing	32 days	2014-10-20	2014-12-02
20		Develop test plan	12 days	2014-10-20	2014-11-04
21		Test prototype and record the result	10 days	2014-11-05	2014-11-18
22		Document test result and test log	10 days	2014-11-19	2014-12-02
23		⊟ Evaluation	19 days	2014-12-08	2015-01-01

		Task Name	Duration	Start	End
4		Prepare project plan and specification	4 days	2014-03-24	2014-03-27
5		⊟ Initial study of the system	10 days	2014-03-28	2014-04-10
6		Detail study on Android platform architecture and its API	10 days	2014-03-28	2014-04-10
7		⊟ Analysis	36 days	2014-04-11	2014-05-30
8		Preparation of SRS document	3 days	2014-04-11	2014-04-15
9		Preparation of Interim report	7 days	2014-04-16	2014-04-24
10		Submission of interim project report	1 day	2014-04-25	2014-04-25
11		System Data Design	10 days	2014-04-28	2014-05-09
12		System Architecture Design	10 days	2014-05-12	2014-05-23
13		Documentation of design specification	5 days	2014-05-26	2014-05-30
14		⊟ Development	100 days	2014-06-02	2014-10-17
15		Development of wireless fingerprinting module	30 days	2014-06-02	2014-07-11
16		Development of navigation system module	40 days	2014-07-14	2014-09-05
17		Combining results of wireless fingerprinting and navigation s...	20 days	2014-09-08	2014-10-03
18		Implementation of combined system in android based device	10 days	2014-10-06	2014-10-17
19		⊟ Testing	40 days	2014-10-20	2014-12-12
20	📅	Develop test plan	20 days	2014-10-20	2014-11-14
21		Test prototype and record the result	10 days	2014-11-17	2014-11-28
22	📅	Document test result and test log	10 days	2014-12-01	2014-12-12
23		⊟ Evaluation	14 days	2014-12-15	2015-01-01
24	📅	Evaluation of the final system	7 days	2014-12-15	2014-12-23
25	📅	Evaluation of the final project	5 days	2014-12-26	2015-01-01
26		Submission of the project	1 day	2015-01-02	2015-01-02