An Android App For Indoor Localization

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Information Technology

under the Supervision of

Mr. SHAILENDRA SHUKLA

By

SAGAR KAPOOR (111458)

to

Jaypee University of Information and Technology
Waknaghat, Solan – 173234, Himachal Pradesh

CERTIFICATE

This is to certify that project report entitled “An Android App For Indoor Localization”, submitted by SAGAR KAPOOR in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Mr . SHAILENDRA SHUKLA**

08-05-2015 **Professor**

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to DR. SHAILENDRA SHUKLA for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & my friends for their kind co-operation and encouragement which help me in completion of this project.

08.05.2015 SAGAR KAPOOR

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

There have been many project efforts on location awareness in an indoor environment. Most of them rely on specialized equipment or motion detectors. This project focuses on signal strengths from WiFi transmitters and use these signal strengths to calibrate virtual pinpoints. A pinpoint is a collection of stored signal strengths over time on a predetermined location. These pinpoints can then be used to situate the surroundings of an environment and determine the current measurement its position by providing the necessary information needed to the tracking methods.

The experiments of this project are set up with low-end routers in an actual indoor work environment to get the baseline results with less than perfect circumstances. The tracking methods used are based on different locating techniques. These techniques vary from converting signal strength to distance, or using signal strength as a ratio difference between distances, to using signal strength to get the average distance variation in an area. All these techniques allow us to calculate the distance to the locations of the signals. These distances are then converted into a position by calculating the radical centre of the corresponding circles.

1. Introduction

1.1 Methodology

Positioning finds its applications in locating a person in an area, helping person to navigate or to reach desired location. GPS is the most famous positioning system that we know. To use GPS line of sight must be there between GPS transmitter and receiver. There should not be any shielding effect between transmitter and receiver. However, certain other application demands to locate a person or navigate person in an indoor areas like in shopping malls or college building. GPS fails in such cases because of its inability to work in non-line of sight conditions and signal attenuation due indoor obstacles like walls, roofs, floors etc. Hence, indoor positioning has been challenging task till date.
Various technologies and methodologies has been used in indoor positioning to improve the accuracy. GSM technology is used in indoor positioning. It is based on cellular signals and makes use of existing hardware in mobile phone. GSM fingerprints are used to achieve better accuracy in positioning. In zigbee(IEEE 805.4) [5] is used for indoor localization purpose. Other short range signals like wifi, Bluetooth, Ultra sound are also being used for this purpose. In this report, indoor positioning system using wifi (WLAN IEEE 802.11n) is explained. In wifi, various methods are used for positioning purpose. In [1] Trilateration method is explained for indoor localization which makes use of the point of intersection of three circles of wifi AP which gives the exact position of user.

In trilateration, signal strengths from all the existing wifi AP's is gathered. Relating the received signal strengths from existing AP's is converted to distance from respective AP's using signal propagation models. Trilateration algorithm is used afterwards for obtaining the position of user in indoor environment. Trilateration is simple and easy to implement method for indoor localization. Various methods are further used to reduce the errors and to improve the accuracy in trilateration method. Trilateration is also called as dynamic method of positioning. In fingerprinting method is used for indoor positioning. It has got mainly two phases viz. offline training phase and online phase. In offline training phase, the database of wifi RSS fingerprints with respective location co-ordinates is prepared. This is the most important phase and accuracy of the system depends on this phase. In online phase,fingerprint database in queried for location of user. Various pattern matching algorithms are used to match the database entries with dynamic RSS. The RSS fingerprint database design is the challenging task in this method. We have explained the indoor positioning system based on wifi technology which uses Trilateration method to estimate user position in indoor environment.

1.2 Motivation

Outdoor localization has already been tapped to its best potential and next natural step is moving indoors. The current outdoor localization technology (mainly GPS) does not extend to the indoors due to loss of GPS signal. This has called for a need to develop innovative indoor localization technologies. This need for indoor localization for pervasive mobile computing has led to a lot of interesting research in the past ten years. Some of the initial research included use of specialized emitters and sensors placed inside buildings to localize objects and people. This method though quite accurate is not scalable for commercial deployments and involves a certain overhead cost in installing and maintaining the additional infrastructure. Another popular method makes use of the existing infrastructure using wireless access points to triangulate and localize using mobile devices. This method is quite accurate but it usually requires extensive surveying and training effort to build a radio frequency (RF) map of the building. There are also improvements to this method which reduce the efforts or eliminate them completely but at the cost of accuracy. With mobile technology becoming more powerful over the last few years, it is now embedded with more sensors which can improve the accuracy of the prediction by combining them with these earlier technologies. This is the motivation behind this project in trying to use fairly new methods such as accelerometer readings, acoustic background sound system and barometer readings in addition to wireless RF maps to increase the probability of predicted positions.

1.3 Objectives

Indoor localization as mentioned above is a research area that has attracted quite a lot of attention recently. A lot of work has been done on different innovative approaches to solve this problem. In this project I will analyze these technologies for advantages, disadvantages from a practical deployment point of view, and then implement a system combining the best and most practical technologies to increase the accuracy of location prediction. The goal is to bring together these technologies that currently exist in silos and create a working implementation.

Our project implements these technologies on the Android platform creating an Android application that can localize a user in campus using just the phone itself. It should also be extensible to localize in other places as well.

Campus building will be used as the test bed for testing the app. The results of this app will be compared to using the technologies used in silos to see how combining them affects the localization. I will also look into some possible simulation model to test out the app for different factors.

1.4 SYSTEM DESIGN

Our Indoor Positioning consists of three parts viz. client device (which can be wifi enabled device like mobile, smartphones, PDA etc), existing network of at least 3 wifi access points (AP's) and server. The client device is capable of wifi access and receiving wifi signals from all access points in the vicinity. The key purpose of client is to record the wifi signals strength data i.e. RSSI. The network of wifi access points acts as communication channel between client server. All access points used in the system are similar. D-LINK wifi AP's are used in system. The server is a laptop which runs Windows 7 . The main task of server is of positioning and navigation system. Hence, it is also called as positioning server. The framework for indoor positioning system is depicted in figure 1.

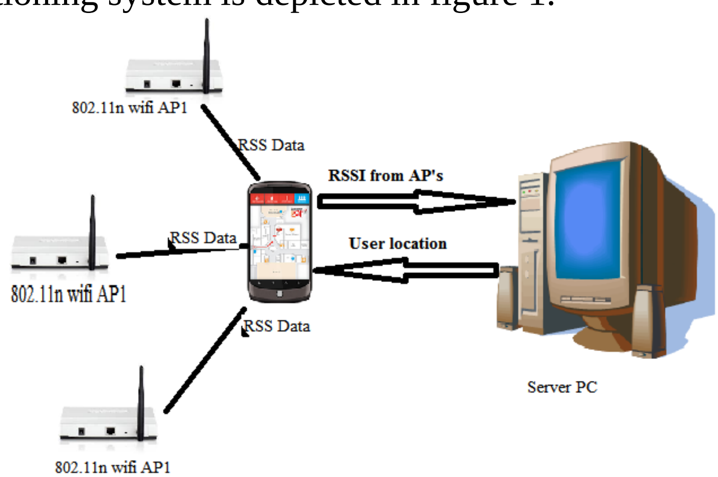

Figure 1 Framework of indoor positioning system [1]

The Android application "WiFiScan" scans the wifi access points in the vicinity of the device. It collects the data like signal strength from the respective access point in dBm, MAC address of AP, channel frequency, SSID etc. This data is the key point in positioning of user in indoor system. Android device then sends this RSS data along with respective MAC address of AP to the positioning server. The server runs the positioning algorithms which calculate the location co-ordinates of the user. Server runs two types of algorithms a) To calculate distance of user from respective AP and b) Tri-lateration algorithm to find exact location of user from distance. Server then sends these location co-ordinates back to the client device. The android device has the front end application showing indoor map of the system. Indoor map in this system is developed using HTML5 Canvas element and invoked in an android application. The location co-ordinates calculated by server are shown in the indoor map as the user position. This is a part of user positioning in indoor environment. As an extension we are also doing the user navigation in indoor environment. Navigation involves the choice of destination in the system. The co-ordinates of destination are known and programmed in indoor map itself. Using the user co-ordinates and destination co-ordinates application helps you to navigate in the system.

1.5 Properties of a good Localization system

1. Should be light enough to run locally on the phone.
2. Should require minimum effort in setting up (fingerprinting).
3. Should be stable with variation of time.
4. Should be robust to changes in the environment.
5. Should work equally well with any measurement device (any mobile phone)
6. Should be accurate enough to predict the room the user is in.

1.6 Limitations

1. Unexpected interference in actual deployment which was not present while fingerprinting.
2. Keeping wifi and GPS on all the time might be draining on the battery.
3. Location of Wifi access points may not be known in places like offices and malls where the deployment is done by external parties.
4. Floor plans might not be publically available.

5. Security in case of privacy breach might be an issue.

2. Literature Survey

2.1 Infrastructure Based methods

As mentioned before the earlier work in localization was mainly based on using pre-installed sensors. Active Badge (Want et al., 1992) is one such system that relies on specialized tags which emitted diffuse infrared pulses detected by ceiling-mounted sensors. Another technology, the Cricket Compass (Priyantha et al., 2001) used specialized ultrasound and radio frequency receivers to detect signals transmitted by fixed beacons. ARIADNE (Bias et al., 2006) deploys wireless sniffers at known locations and makes use of a sophisticated ray-tracing model based on detailed floor maps and uses simulated annealing to estimate radio propagation parameters.

These methods are not scalable so the focus shifted on using off the shelf technologies such as wireless routers, which were usually preinstalled, and measuring radio frequency signal for location discovery. Next we will discuss these methods

2.2 Infrastructure-less based methods

2.2.1 Location Fingerprinting methods using Wifi

<u>Early implimentations</u>

Radar (Bahl and Padmanabhan, 2000[4]) was one of the early implementation of wireless fingerprinting1 to approach the indoor localization problem. They used the RSS (received signal strength) measurements and user orientation as the input parameters for fingerprinting. Since they were recording signal strengths from direct signals received, user orientation was an important factor since the body of the user could dampen the signal too. Without using user orientation the accuracy of the system dropped by over 50%. Therefore for every position in the training phase four readings were taken for the four orientations. During the measurement phase2 multiple readings are taken for each point and averaged. This average signal strength is compared with the fingerprinted data. Radar uses the nearest neighbor in signal space (NNSS) method, computing the minimum distance between the measurements and the observation, to find the most accurate (nearest) location. They also mention that during the training phase taking readings closer than a threshold (two points should not be too near) reduces the accuracy of prediction and usually having three or more samples for a position increases accuracy.

A big limitation for scaling this method is that the training phase is very labor intensive, and recalibration is required if there is any relocation of wifi access points. Thus, they also propose a radio propagation model which tries to estimate the signal strength values over an area using the fact that radio waves weaken with the distance travelled and when passing through objects. They consider three popular radio propagation models and settle on the Floor Attenuation Factor propagation model (Seidel and Rapport, 1992[4]) for its simplicity and accuracy. This model performed worse than the empirical method and also required empirical calculation of the wall attenuation factor (WAF). In practical use the WAF could be different for walls within the building and the real-time movement of people in the indoor environment might affect the calculations, which would affect the accuracy, so we decide not to use a propagation model for this project.

2.2.2 Location determination algorithms

Liu et al., 2007[4] does a survey of the current wireless indoor position systems and compares different techniques for various factors such as accuracy, precision, complexity, scalability, robustness, and cost. Brief overviews of the relevant algorithms from those discussed in the paper are as follows:

1) **Triangulation**: This method estimates location by measuring the distance of the mobile device from multiple reference points.

a. TOA: time of arrival is used to measure the distance of the reference point emitting the signal, as it is directly proportional to the distance between them. All the transmitters and receivers need to be synchronized and timestamps need to be sent with the signals.

b. TDOA: time difference of arrival is used to measure the difference in arrival time of the signal at different positions. The intersection of the hyperbolic arcs to these positions gives the location of the receivers.

c. RSS based: measures RSS (Received Signal Strength) to use as a replacement for distance measures. It is better than TOA and TDOA as it takes into account multipath and shadowing effects. Pre-recording these values increases the accuracy.

2) **Angulation techniques**: uses the intersection of a pair of angle direction lines from the reference points to find the location. It works for 2-d localization with only 2 reference points, but needs extensive hardware for accurate measurements.

3) **Probabilistic methods**: A popular approach is the histogram approach which considers positioning as a classification problem and maximizing the likelihood of the predicted location. Another approach assumes the likelihood of each location candidate is a Gaussian distribution.

2.3 Android Implementations

Martin et al., 2010 [5] show an Android application implementation using the wifi fingerprinting approach and claim to have an accuracy of up to 1.5 meters. The accuracy drops quite a lot with less than three available wifi access points available. They have used the Nearest Neighbor in signal space and Access Point averages approach to calculate the most probable location. Our proposed application will be similar to this implementation, trying to be more accurate and less labor intensive for fingerprinting. The training and experimentation was carried out in very controlled conditions for this application, which may not be very scalable. There is not much detail available about the implementation and testing about this application.

Another Android application called "Locate Me"(Pereira, 2011[5]) also attempts to solve this problem combining three methods: 1)Fingerprint localization using wifi access points 2)Location using geo-referenced access points which stores GPS values with the fingerprint values 3) Localization using geo-referenced mobile base stations, which performs quite badly, but might be the only available method sometimes. This is the only smart phone app available for indoor localization but it doesn't really let the user localize his space. It also does not show GPS and cell values.

2.4 Problems in a Building

The most important problems in respect to this research are the problems created by the building itself. As shown in the section above signal strength degrades fairly linear over distance and will also always do this in a perfect environment. But a building is not a perfect environment because of different kind of line of sight blockers and materials that influence the signal and of course a multi-story building has floors as special line of sight blocker and the with the height difference. These things in a building each have an influence on the strength of the signal in their own way. The problems with the most impact on this research are multicast and line of sight. These two problems are described below in the following two subsections.

2.4.1 Multicast

Multicasting is a very common occurrence in a building. Multicasting is the reflection of a signal of a surface and this reflected signal aids the original signal in its strength or so it appears for the receiver. This reflecting may even have multiple times or on multiple surfaces at the same time. The end result of this is that the receiver has detected the signal with a slight boost. Figure 5 shows the difference between an ideal situation and a real situation. The boost in signal is a real problem for finding the position of the receiver and can lead to incorrect assumptions and will hinder readouts.

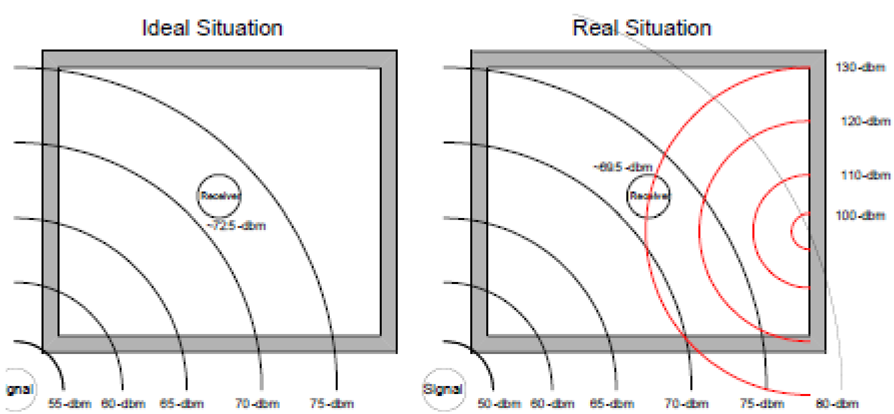

Figure 2 Illustration of the multicast problem. The left image depicts the ideal situation and the right shows a possible realistic scenario.

Actually the problem can be turned into a solution for similar signal strengths at different locations, because even though the signal reflects a surface it will produce a fairly unique measurement at that location. The hypothesis is that if all the signal strengths from the different signal transmitters are measured, that the combined result of these strengths is going to be fairly unique for that position. With this in mind it is believed that the calibration pinpoints will be unique for the positions and will convey into a usable mapping of the environment.

2.4.2 Line of Sight

As mentioned before an indoor environment can have a lot of different “Line of Sight” blockers in between the transmitter and the receiver that will interfere with the signal.
Specifically the interference to the signal will be a decrease in signal strength for every object it needs to pass through. This applies to all objects, but the manner of decrease is dependent on the objects material type and the actual distance the signal has to pass through the object. In an indoor environment the object encountered will be the structure of the building itself like walls, floors, and columns but other things like furniture and even people themselves can also have a negative effect on the signal.
Figure 6 shows the affect walls can have on the signal strength. As can be seen the strength can drop significantly for each wall it passes. This has as effect that it can be difficult to make assumptions and will hinder readouts but can also lead to blind spots where the signal has poor reception or cannot be received at all.

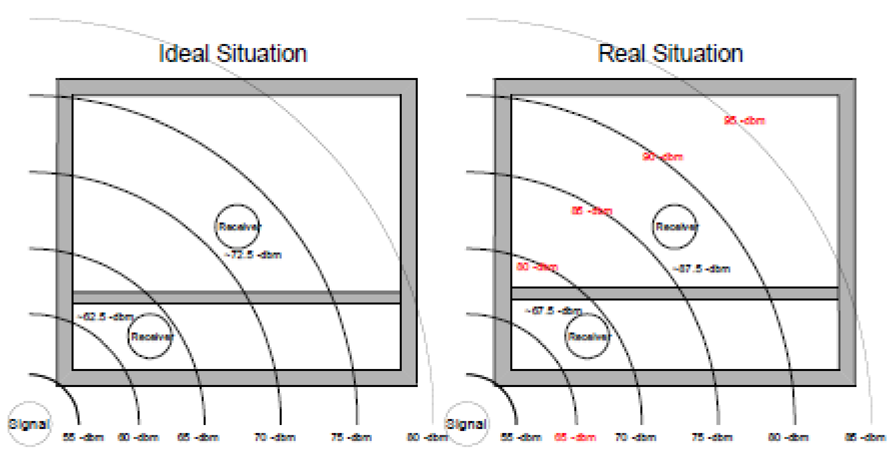

Figure 3 Illustration of the “Line of Sight” problem. The left image depicts the ideal situation and the right shows the affect walls can have on the signal.

This problem really has some negative effects, but just as with the multicast the problem might also be use and thus also has some positive effects. As explained with the multi cast problem the change in signal strength that are caused can be seen more unique values for the calibration pinpoints. The hypothesis is that if all the signal strengths from the different signal transmitters are measured, that the combined result of these strengths is going to be fairly unique for that position. This is because the rooms located in the building will not all be the same and are all in a different angle and position to the transmitters thus will have different combined signal strengths. This will all contribute to the uniqueness of the created pinpoints and will only aid in mapping of the environment. This of course excludes moving objects like people, which only have a negative effect because of the movement factor.

3. Sensor’s Used

3.1 Accelerometer

An accelerometer is an electromechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer.

By measuring the amount of static acceleration due to gravity, you can find out the angle the device is tilted at with respect to the earth. By sensing the amount of dynamic acceleration, you can analyze the way the device is moving. At first, measuring tilt and acceleration doesn't seem all that exciting. However, engineers have come up with many ways to make really useful products with them.

An accelerometer can help your project understand its surroundings better. Is it driving uphill? Is it going to fall over when it takes another step? Is it flying horizontally or is it dive bombing your professor? A good programmer can write code to answer all of these questions using the data provided by an accelerometer. An accelerometer can help analyze problems in a car engine using vibration testing, or you could even use one to make a musical instrument.

In the computing world, IBM and Apple have recently started using accelerometers in their laptops to protect hard drives from damage. If you accidentally drop the laptop, the accelerometer detects the sudden freefall, and switches the hard drive off so the heads don't crash on the platters. In a similar fashion, high g accelerometers are the industry standard way of detecting car crashes and deploying airbags at just the right time.

3.2 WIFI

Wi-Fi (or **WiFi**) is a local area wireless computer networking technology that helps electronic devices to connectand be in a same network without using any wires, mainly using the 2.4 gigahertz (12 cm) UHF and 5 gigahertz (6 cm) SHF ISM radio bands.

The Wi-Fi Alliance defines Wi-Fi as any "wireless local area network" (WLAN) product based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards". Many devices can use Wi-Fi, e.g. personal computers, video-game consoles, smartphones, digital cameras, tablet computers and digital audio players. Wifi routers can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.

Figure 4 wifi router image [10]

Depiction of a device sending information wirelessly to another device, both connected to the local network.

Wi-Fi can be less secure than wired connections, such as Ethernet, precisely because an intruder does not need a physical connection. Web pages that use TLS are secure, but unencrypted internet access can easily be detected by intruders. Because of this Wifi has adopted many encryption standards and security measures .The early encryption WEP proved easy to break. Higher quality protocols (WPA, WPA2) were added later. An optional feature added in 2007, called Wi-Fi Protected Setup (WPS), had a serious flaw that allowed an attacker to recover the router's password. The Wi-Fi Alliance has since updated its test plan and certification program to ensure all newly certified devices resist attacks.

4. Method Used- Trilateration

The method used for improving the accuracy of an indoor position system is **Wi-Fi Trilateration** which uses Wi-Fi signal strength relating with distance formula to determine position of a user. Based on the concept of GPS, minimum of three access points (AP) are needed to determine the position of a user in an indoor location[8]. The Wi-Fi signals are in the form of radio wave where the movements of the signals are highly dependent on the frequency. Signals with different diameters are transmitted by APs in all direction according to the respective signal strength. Since wireless routers provide coverage of about 100 feet (30.5 meters), signal strength is used to find the collision point in order to specify the accurate position of an object.

The standard protocol of Wi-Fi is 802.11 which was introduced by Institute of Electrical and Electronic Engineering (IEEE) and it is used in wireless LAN. The standards come in several flavors which are 802.11a that transmits at 5 GHz and can move up to 54 megabits of data per second. On the other hand, 802.11b is the slowest and slightly less expensive and transmits in the 2.4 GHz frequency and can carry 11 megabits of data per second. Networking standard 802.11g also transmits at 2.4 GHz like 802.11b but it is much faster and theoretically can handle up to 54 megabits of data per second. 802.11n is the newest standard to improve speed and range. These kinds of protocol standards allow communication via internet through channels of communication medium that is available in Wi-Fi. There are 14 channels available in Wi-Fi where the use of each channel can be selected to avoid interferences in the wireless transmissions. This study deploys Wi-Fi technique in conjunction with IEEE 802.11g networking standard. Here, we assume the three APs are known as AP1, AP2, and Ap3.

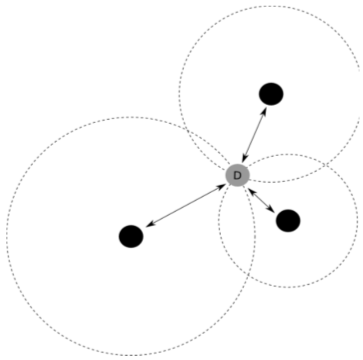

Figure 5: Three access points [8]

Assume that the coordinates of the three APs as Figure 2:

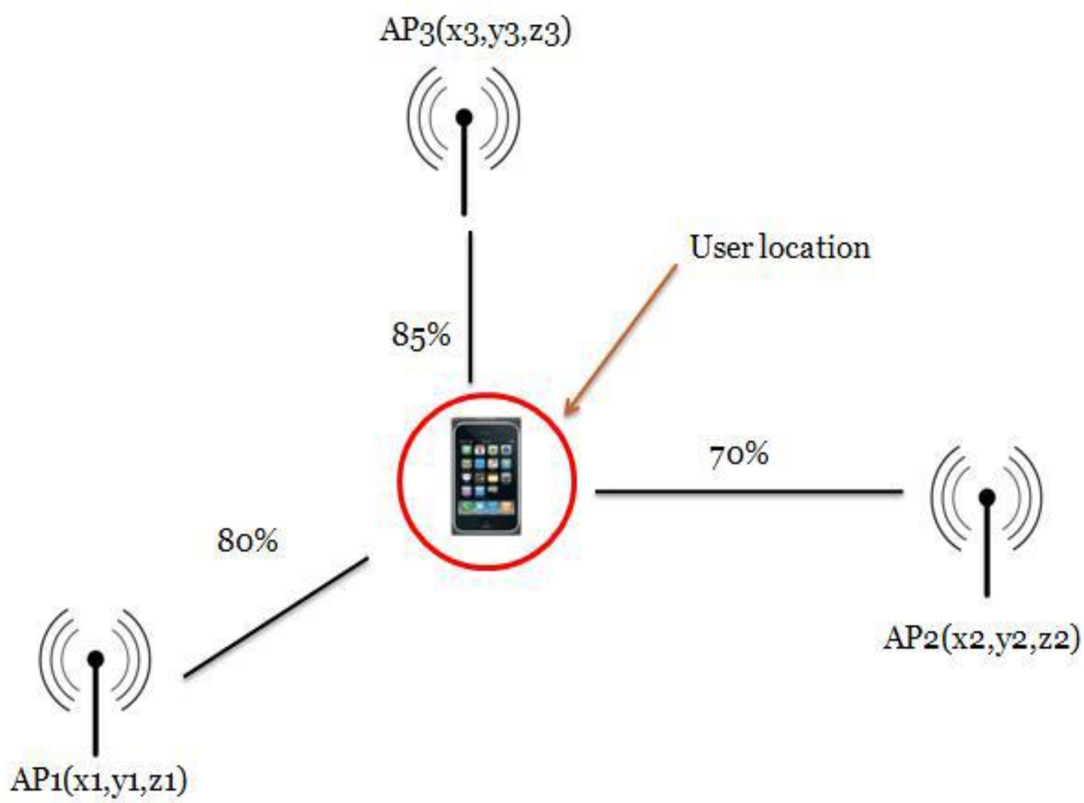

Figure 6: Illustration WiFi signal strenght from three access points[1]

Then, based on three coordinates of the APs, we need to find the coordinates of the user's position that is represented as *Z*.

Let's assume that a user is using a smart phones that serves as a receiver of the signals transmitted from the access points. Application of Wi-Fi analyzer

in the smart phone presents the signal strength in terms of percentage. The highest percentage of signal strength indicates that *Z* is closest to the AP whereas the lowest percentage implies that Z is maximum range of AP.
The percentage of signal strength obtained from the Wi-Fi analyzer can be converted to distance between a user's to each AP using this equation (Equation 1):

Distance, di = p (1– mi)…………….(**3.1**)

Where;

m = is the percentage of signal strength
p = is the maximum coverage of signal strength
i = 1,2,3……n

[1]From Figure 3, let each AP be placed at the center. Assume a scenario where a student who uses a smart phone, is looking for a book in a library. Then, we assuming that signal strength for each AP will spread the signal in wave forms. The signal strength will form 3 circles and intersect each other. The intersection of 3 circles is the position of user and we want to determine the location of user who is labeled by *B* (*x*, *y*). To simplify the calculations, the equations are formulated so intersection of circle is occurred at Cartesian plane (see Figure 2). The equation for any of these circle is as follow (assuming *z* = *0*):

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2$$

………………..(**3.2**)

The intersection of 3 circles is obtained by solving systems of linear equations for 2 variables simultaneously. The linear systems are solved in order to determine the coordinates *x* and *y*.

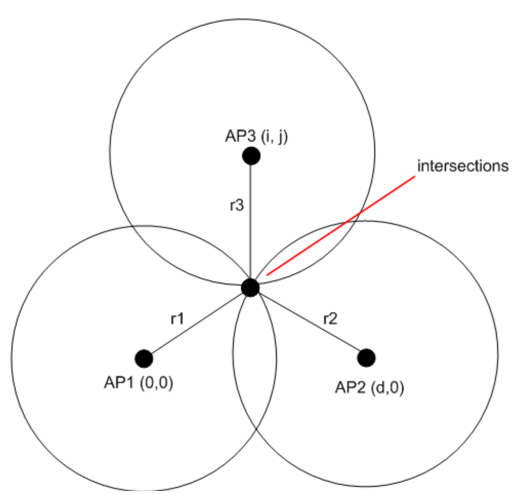

Figure 7: Intersections of 3 circles[1]

Based on Figure 4, we start with the equations for three circles:

$$r_1^2 = x^2 + y^2 + z^2 \qquad \text{……………….}\textbf{(3.3)}$$

$$r_2^2 = (x-d)^2 + y^2 + z^2 \qquad \text{…………….}\textbf{(3.4)}$$

$$r_3^2 = (z-i)^2 + (y-j)^2 + z^2 \qquad \text{……………..}\textbf{(3.5)}$$

To determine the location of B, we have to solve for (*x*, *y*, *z*).

The method to do it is by using systems of linear equations for 2 variables and solve these eaquation of linear system $\tilde{A}\, x = b$. By using this method, the *j*thconstraints is used as a linearizing tool. Adding and subtracting *xj*, *yj*and *zj*in (3.3),(3.4) ,(3.5), gives:

$$(x - x_j + x_j - x_i)^2 + (y - y_j + y_j - y_i) + (z - z_j + z_j - z_i)^2 = r_i^2 \qquad \text{……}$$

………**(3.6)**

With (*i = 1, 2,... , j+1 , ... , n*).

Linear system is easily written in matrix form ***Ãx* = *b*,**

$$A = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ \vdots & \ddots & \vdots \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} b_{21} \\ \vdots \\ b_{n1} \end{bmatrix} \qquad \text{……………}\textbf{(3.7)}$$

Based on the calculation by using (3.7), the position of B is given by (x, y, z).

5. Using Accelerometer

Most of the indoor localization methods focus on single floor implementations. Methods such as Wifi fingerprinting might not always work to determine floors when the Wifi signal is not strong enough to reach other floors. FTrack (Haibo et al., 2012[2]) is an alternative way of determining which floor the user is based on the only accelerometer values. Ideally we can calculate the user location indoors by double integrating the acceleration, but the accelerometer readings are usually very noisy. FTrack proposes using the signatures in human walking patterns (the nature up and down bounce) to obtain the current floor information for the user. Taking stairs and using the elevator have been identified as the two main ways of changing the floor value and the time spent in changing the floors is recorded.

While taking the stairs it can be figured out based on the number of peaks in the acceleration time graph that how many steps were taken and in what direction (up or down). Similarly for taking the elevator time to go to different floors is recorded. The acceleration and deceleration at the start and end of the elevator ride is used to the figure out the time taken and direction of these values is used to figure out the direction of the movement. This information is stored in two states: S<UserID, StartTime, EndTime, Encounters> which is when the user is on a particular floor and M<UserID, Duration, ActionT ype, Direction>, while the user is changing floors.

The fingerprinting is done by several people changing floors over a period of time. It also makes use of Bluetooth technology during fingerprinting to record any encounters (coming in close proximity) between different users. These encounters are later used to verify the predicted floor information. For example, if two users travel the same period of time and in the same direction (up or down), and end up on the same floor and encounter each other, they must come from the same floor. 97% accuracy was achieved after 3 hours of data collection and it was observed that a higher number of encounters positively affected the results.

There is a prime limitation of this method. Taking an escalator (which has become an everyday alternative to stairs and elevators) is not accounted for in this experiment. Readings similar to an elevator can be expected if the person does not move. If the person also climbs the escalator stairs at the same time, it would be difficult to figure out the time taken. The sudden acceleration and deceleration when a person gets on and get off an escalator might still produce a spike in the graph good enough to distinct the time taken, but then again it would differ every time the user takes an escalator as he is free to climb it faster if there is less traffic present.

AAMPL (Accelerometer Augmented Mobile Phone Localization) is another technology which makes use of accelerometers to find the position of a person indoors. It tries to find a correlation between accelerometer readings taken while being in different kind of business stores. It tries to measure the time spent standing or sitting and maps it to the type of business (Ofstad et al., 2008[2]). For example, you are more likely to be standing more at a supermarket than a restaurant. It finds your location, retrieves the businesses around you and then filters them by business type based on your accelerometer readings patterns.

6. Design and Implimentations

6.1 Design

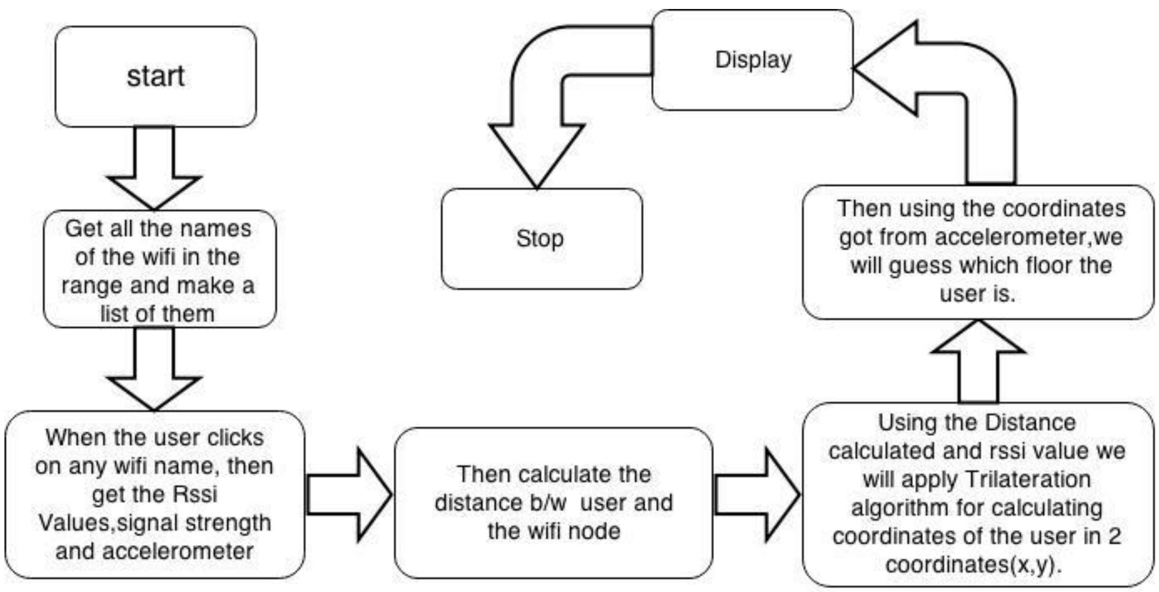

Figure 8 Flow Diagram of the App.

The above diagram explains the working of the app. This also explained the steps with which we will get the desired output which is the coordinates of the user. The next figure tells the relations b/w the classes in the apps. So we have two main classes namely mainactivity and activity2. The mainactivity class just creates the list of all the wifi's in the range and the activity2 class calculates the distance, coordinates and coordinates provided by accelerometer and then guesses the floor on that basis.

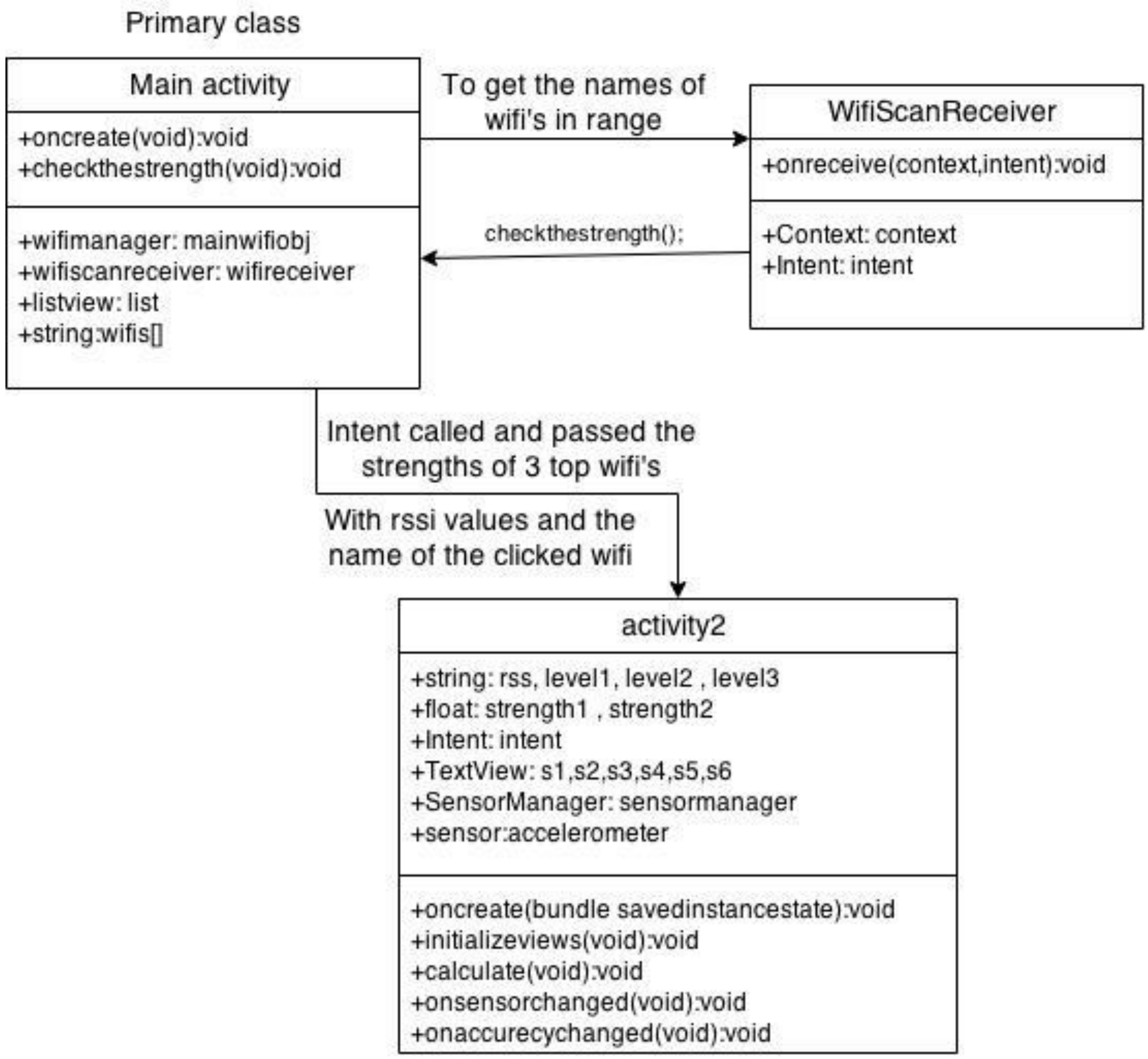

Figure 9 class diagram of the app

6.2 Implimentation

Till date we have implemented the code for calculating the rssi value, distance using the Rssi value, coordinates using the trilateration algorithm and accelerometer values is required in calculating the floor on which the user is for further use of our algorithm. The code we have implemented looks for the available wifi networks in the surroundings and by selecting any of the network it gives us the signal strength of that wifi network. The available networks will vary from device to device depending upon the WiFi analyser present in the device.

```
public class MainActivity extends Activity {

  WifiManager mainWifiObj;
  WifiScanReceiver wifiReciever;
  ListView list;
  String wifis[];
  List<ScanResult> wifiScanList;
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    list = (ListView)findViewById(R.id.list);
    mainWifiObj = (WifiManager) getSystemService(Context.WIFI_SERVICE);
    wifiReciever = new WifiScanReceiver();
    mainWifiObj.startScan();

  }
```

The above module of the code creates a list view for the wifi names and creates an object of wifi manager to get the names and values of the wifi in the range.

```
int pos=0;
          String selectedFromList =(String) (list.getItemAtPosition(position));
          for(int i = 0; i < wifiScanList.size(); i++){
             if(((wifiScanList.get(i)).SSID.toString()).matches(selectedFromList))
             {
                pos=i;
             }

          }
```

```
int strength1 = ((wifiScanList.get(0)).level);
int strength2 = ((wifiScanList.get(1)).level);
int strength3 = ((wifiScanList.get(2)).level);
int value1=WifiManager.calculateSignalLevel(strength1,11);
int value2=WifiManager.calculateSignalLevel(strength2,11);
int value3=WifiManager.calculateSignalLevel(strength3,11);
String val1=Integer.toString(value1);
String val2=Integer.toString(value2);
String val3=Integer.toString(value3);
String rssi=String.valueOf(strength1);
// Toast.makeText(MainActivity.this,val , Toast.LENGTH_SHORT).show();
Intent intent = new Intent(MainActivity.this,Activity2.class);
intent.putExtra("Name", selectedFromList);
intent.putExtra("Strength1", val1);
intent.putExtra("Strength2", val2);
intent.putExtra("Strength3", val3);
intent.putExtra("Rss",rssi);
startActivity(intent);
```

The above module gets the strength and Rssid of the wifi clicked. And then passing it to the next page of the app using the intent function. We are passing the name on which the user click's,strength of the top 3 wifi's and rssi value of the wifi on which the user has cliched.

```
public class WiFiScanReceiver extends BroadcastReceiver {
        private static final String TAG = "WiFiScanReceiver";
        WiFiDemo wifiDemo;

        public WiFiScanReceiver(WiFiDemo wifiDemo) {
          super();
          this.wifiDemo = wifiDemo;
        }

        @Override
        public void onReceive(Context c, Intent intent) {
          List<ScanResult> results = wifiDemo.wifi.getScanResults();
          ScanResult bestSignal = null;
          for (ScanResult result : results) {
            if (bestSignal == null
                || WifiManager.compareSignalLevel(bestSignal.level, result.level) < 0)
              bestSignal = result;
```

```
        }

        String message = String.format("%s networks found. %s is the strongest.",
            results.size(), bestSignal.SSID);
        Toast.makeText(wifiDemo, message, Toast.LENGTH_LONG).show();

        Log.d(TAG, "onReceive() message: " + message);
      }
```

The above module is giving us the wifi scan results and listing them .

```
public boolean onOptionsItemSelected(MenuItem item) {
            // Handle action bar item clicks here. The action bar will
            // automatically handle clicks on the Home/Up button, so long
            // as you specify a parent activity in AndroidManifest.xml.
            int id = item.getItemId();
            if (id == R.id.action_settings) {
                return true;
            }
            return super.onOptionsItemSelected(item);
    }
```

The above module will pass the id of the element selected so that rssid and wifi strength can be shown of the selected item.

```
    intent=getIntent();
  level1  = intent.getStringExtra("Strength1");
  strength1 =Float.parseFloat(level1);
  level2  = intent.getStringExtra("Strength2");
  strength2 =Float.parseFloat(level2);
  level3  = intent.getStringExtra("Strength3");
  strength3 =Float.parseFloat(level3);
```

The above code gets the value’s passed from the previous page using the get string extra function.

```
s1=(TextView)findViewById(R.id.t5);
        s2=(TextView)findViewById(R.id.t2);
        s3=(TextView)findViewById(R.id.t3);
        s4=(TextView)findViewById(R.id.t7);
        s5=(TextView)findViewById(R.id.t6);
        s6=(TextView)findViewById(R.id.t8);
        s7=(TextView)findViewById(R.id.t9);
        String n;
        n=s2.getText().toString();
        n=n+" "+rss;
        s2.setText(n);
        n=s3.getText().toString();
        n=n+" "+level1;
        s3.setText(n);
        d1=10*(1-(strength1/10));
        d2=10*(1-(strength2/10));
        d3=10*(1-(strength3/10));

        n=s4.getText().toString()+d1;
        s4.setText(n);
        n=s6.getText().toString()+d2;
        s6.setText(n);
        n=s7.getText().toString()+d3;
        s7.setText(n);
        initializeViews();
```

In the above line of codes we are making objects of the text views we have created in the app layout (activity_activity2.xml). Then we calculate the distance using the strength and then initializing all the views.

```
   sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        if (sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER) != null) {
```

```
        // success! we have an accelerometer

        accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_NORMAL);
      } else {
        // fai! we dont have an accelerometer!
      }
```

The above line of codes gets an object of the sensor listener which will get the changes in the accelerometer values.

```
void calculate()
  {
      float r1,r2,r3;//radius of the circles. distance caluclated from the strength
      float x1,x2,x3,y2,y3,y1;
      x1=10;
      x2=20;
      x3=30;
      y1=10;
      y2=20;
      y3=30;
      double d21,d31;
      double b2,b3,x,y,z=0;
      x3=(x1+x2)/2;
      y3=5;
      r1=d1;
      r2=d2;
      r3=d3;//distance using strength
              /*double rad1=(double)(x1*x1 + y1*y1 + z1*z1);
              r1=(float)Math.sqrt(rad1);
              double rad2=(double)(Math.pow(x-x2,2)+Math.pow(y-
y2,2)+Math.pow(z-z2,2));
              r2=(float)Math.sqrt(rad2);
              double rad3=(double)(Math.pow(x-x3,2)+Math.pow(z-z3,2)+Math.pow(z-
z3,2));
              r3=(float)Math.sqrt(rad3);*/
      d21=Math.sqrt(Math.pow((x2-x1), 2)+Math.pow((y2-y1), 2));
      d31=Math.sqrt(Math.pow((x3-x1), 2)+Math.pow((y3-y1), 2));
      b2=0.5*(Math.pow(r1, 2)- Math.pow(r2, 2)+ Math.pow(d21, 2));
      b3=0.5*(Math.pow(r1, 2)- Math.pow(r3, 2)+ Math.pow(d31, 2));
```

```
        x= b2/x2;
        y= (b3-(x*x3))/y3;
        //display x,y,z

        String n=s1.getText().toString()+x;
        s1.setText(n);
        n=s5.getText().toString()+y;
        s5.setText(n);


    }
```

The above function will calculate the coordinates using the trilateration algorithm and then changing the values of some text views where we want to show that.

```
public void onSensorChanged(SensorEvent event) {


        displayCurrentValues();
        // get the change of the x,y,z values of the accelerometer
        deltaX = Math.abs(lastX - event.values[0]);
        deltaY = Math.abs(lastY - event.values[1]);
        deltaZ = Math.abs(lastZ - event.values[2]);


    }
```

The above function will call the display current values function when the accelerometer values changes.

```
public void displayCurrentValues() {
        currentX.setText(Float.toString(deltaX));
        currentY.setText(Float.toString(deltaY));
        currentZ.setText(Float.toString(deltaZ));
    }
```

The above function will show the current values of the x, y, z coordinates provided by the accelerometer.

```
Intent intent;
String level1,level2,level3;
```

```
String rss1;String rss2;String rss3;
float  xx1,xx2,xx3;
float d1,d2,d3;
float x11,x12,x13;
float y11,y12,y13;
String name;
Button b,b2,b3;
float x0=0,  y0=0,  z0=0;
float x2_ ;
float y2_  ;
float z2_ ;

float x3_  ;
float y3_  ;
float z3_ ;

TextView s3,s4,s5;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_activity2);
         intent=getIntent();
          level1  = intent.getStringExtra("Strength1");
          xx1 =Integer.parseInt(level1);
          level2  = intent.getStringExtra("Strength2");
          xx2 =Integer.parseInt(level2);
          level3  = intent.getStringExtra("Strength3");
          xx3 =Integer.parseInt(level3);
          b=(Button)findViewById(R.id.button1);
          b2=(Button)findViewById(R.id.button2);

          rss1=intent.getStringExtra("Rss1");
          rss2=intent.getStringExtra("Rss2");
          rss3=intent.getStringExtra("Rss3");
          name=intent.getStringExtra("Name");
         // s1=(TextView)findViewById(R.id.t1);
         // s2=(TextView)findViewById(R.id.t2);
          s3=(TextView)findViewById(R.id.t3);
          s4=(TextView)findViewById(R.id.t4);
          s5=(TextView)findViewById(R.id.s5);
            d1=10*(1-(xx1/1000));
        d2=10*(1-(xx2/1000));
        d3=10*(1-(xx3/1000));
        s4.setText("Distance for first wifi" + d1+"\n"+
                "Distance for second wifi" + d2+"\n"+
                "Distance for third wifi" + d3+"\n"+
```

```
                                "Rss for first wifi" + rss1+"\n"+
                                "Rss for second wifi" + rss2+"\n"+
                                "Rss for third wifi" + rss3+"\n"
                                        );


                b2.setOnClickListener(new OnClickListener(){

                                @Override
                                public void onClick(View arg0) {
                                        int     strength1     =     ((GlobalClass.GetInstance(
).getWifiList().get(0)).level);
                                  int       strength2       =       ((GlobalClass.GetInstance(
).getWifiList().get(1)).level);
                                  int       strength3       =       ((GlobalClass.GetInstance(
).getWifiList().get(2)).level);

                                  int
value1=WifiManager.calculateSignalLevel(strength1,1001);
                                  int
value2=WifiManager.calculateSignalLevel(strength2,1001);
                                  int
value3=WifiManager.calculateSignalLevel(strength3,1001);

                                  y11=10*(1-(value1/1000));
                                  y12=10*(1-(value2/1000));
                                  y13=10*(1-(value3/1000));

                                  calculate();


                                  s4.setText(  "our coordinates with respect to the 1st wifi"
+ x0 +","+ y0 +","+z0 +","+"\n"+
                                               "coordinates of 2nd wifi " + x2_ +","+ y2_
+","+z2_ +","+"\n"+
                                               "coordinates of 3rd wifi " + x3_ +","+ y3_
+","+z3_ +",");
                                }});

                b.setOnClickListener(new OnClickListener(){

                        @Override
                        public void onClick(View arg0) {
                                int       strength1       =       ((GlobalClass.GetInstance(
).getWifiList().get(0)).level);
```

```
            int strength2 = ((GlobalClass.GetInstance(
).getWifiList().get(1)).level);
            int strength3 = ((GlobalClass.GetInstance(
).getWifiList().get(2)).level);

            int value1=WifiManager.calculateSignalLevel(strength1,1001);
            int value2=WifiManager.calculateSignalLevel(strength2,1001);
            int value3=WifiManager.calculateSignalLevel(strength3,1001);

            x11=10*(1-(value1/1000));
            x12=10*(1-(value2/1000));
            x13=10*(1-(value3/1000));


            s4.setText("Distance for first wifi" + x11+"\n"+
                    "Distance for second wifi" + x12+"\n"+
                    "Distance for third wifi" + x13+"\n"+
                    "Rss for first wifi" + strength1+"\n"+
                    "Rss for second wifi" + strength2+"\n"+
                    "Rss for third wifi" + strength3+"\n"
                        );

        }});



    }
    void calculate()
    {

        float d1d = d1;
        float  d2d=d2;
        float d3d=d3;
        float x1 = (float) ((Math.pow(d1d,2)-Math.pow(x11,2))/4 +1);
        float y1=(float) ((Math.pow(d1d,2)-Math.pow(y11,2))/4 +1);
        float z1=(float) Math.sqrt((Math.pow(d1d,2)-Math.pow(x1,2)-
Math.pow(y1,2)));
        float x2=(float) ((Math.pow(d2d,2)-Math.pow(x12,2))/4 +1);
        float y2=(float) ((Math.pow(d2d,2)-Math.pow(y12,2))/4 +1);
        float z2=(float) Math.sqrt(Math.pow(d2d,2)-Math.pow(x2,2)-
Math.pow(y2,2));
        float x3=(float) ((Math.pow(d3d,2)-Math.pow(x13,2))/4 +1);
        float y3=(float) ((Math.pow(d3d,2)-Math.pow(y13,2))/4 +1);
        float z3=(float) Math.sqrt(Math.pow(d3d,2)-Math.pow(x3,2)-
Math.pow(y3,2));
```

```
        //DISPLAY -"SHIFTING WIFI 1 AS ORIGIN
        //          position of user is "

          x0=-x1;
          y0=-y1;
          z0=-z1;

          x2_=x2-x1;
          y2_=y2-y1;
          z2_=z2-z1;

          x3_=x3-x1;
          y3_=y3-y1;
          z3_=z3-z1;
    }
```

The above line of codes help us in getting the coordinates of the 3 wifi in the region with maximum rssi's. In this we have assumed the the coordinates of the user as zero in all three coordinates.

7. Result

We did this experiment to get the accuracy of the distance calculation functioning of the app, which is the back bone of the app. Without the distance we won't be able to get the coordinates. And the accuracy of the coordinates calculation will be almost similar to the accuracy calculated below.

The Below result was calculated in the lab-9 in juit campus for checking purpose and all the above wifi's were present at the time of testing. The above readings shows the accuracy of the app, from the above data we can get that the app is satisfactory.

WIFI NAME	RSSI VALUE	DISTANCE CALCULATED	REAL DISTANCE
JUIT5	-74	5m	3.5m
IDEA-SMARTWIFI-6BAB	-93	9m	6m
JUIT18	-92	10m	5m

Table 1 showing the accuracy in apps distance calculation

The below data was calculated near the faculty cubicals first floor juit campus. At the time of gathering app showed the above information. We can conclude from the above data that the app is giving satisfactory results.

WIFI NAME	RSSI VALUE	DISTANCE CALCULATED	REAL DISTANCE
JUIT18	-75	4.45m	2m
TP-LINK_3B22BA	-86	6.89m	4m
CONNECTIFY-ME	-84	6.45m	5m

Table 2 showing the accuracy in apps distance calculation

The below data was calculated in my room h4-24 shastri bhawan juit campus. At the time of calculating app showed the above information. We can conclude from the above data that the app is giving satisfactory results.

WIFI NAME	RSSI VALUE	DISTANCE CALCULATED	REAL DISTANCE
CONNECTIFY-ME	-81	6m	3m
JUIT420	-74	5m	1m
PETH	-84	5m	2m

Table 3 showing the accuracy in apps distance calculation

The diagrams below show the accuracy of the app wifi checker. In this we have calculated the coordinates theoretically using the same formula and with the assumed distances b/w the user and wifi node. And then applying some mathematical formula such as the distance formula and the trilateration algorithm and then we got this coordinates. The first chart shows the change in the outputs in the x-coordinate theoretically and practically. Then the second one shows the same for y-coordinates. The third Scenario was taken at my room to check how will the app react when the distances are more, whereas the first two scenarios are shown above too as table 1&2.

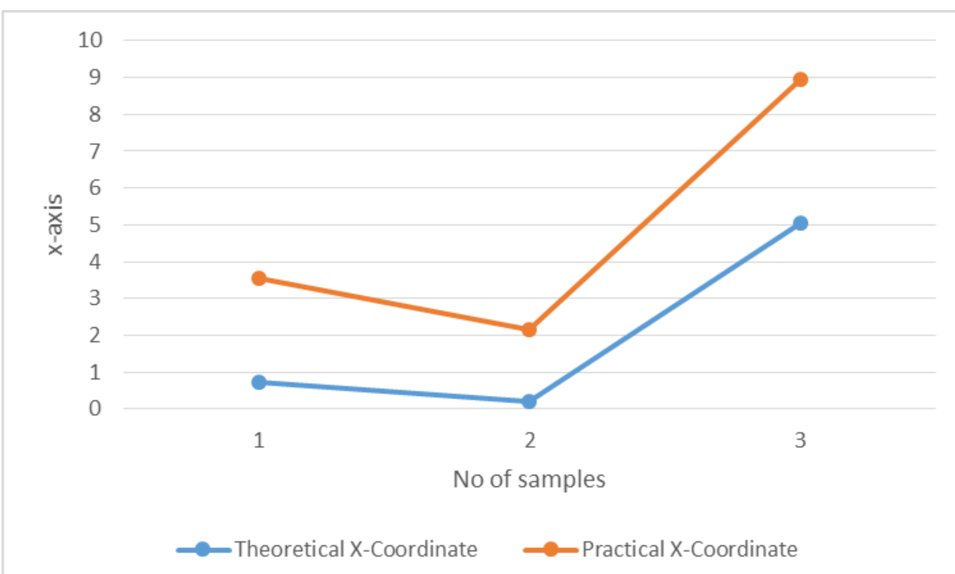

Figure 10 Graph showing the accuracy of x-coordinates practically calculated and theoretically

S.no	Theoretical X-Coordinate	Practical X-Coordinate	Approximate Distance's
1	.74	3.56	3.5, 5 , 6
2	.203	2.17	2, 5, 4
3	5.04	8.95	11, 9 ,15

Table 4 showing the x-coordinates theoretical as well as practical for calculating accurecy.

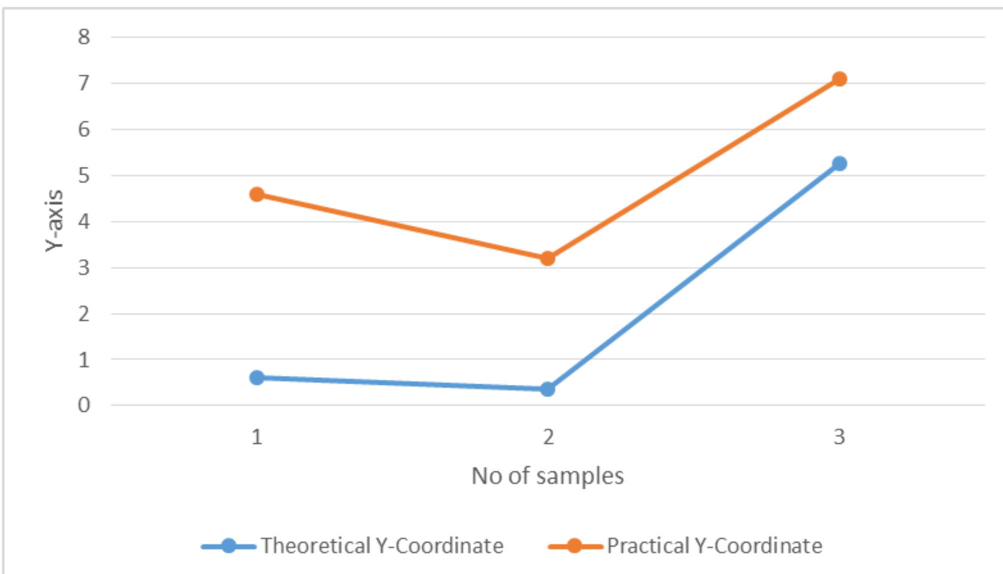

Figure 11 Graph showing the accuracy of y-coordinates practically calculated and theoretically

S.no	Theoretical Y-Coordinate	Practical Y-Coordinate	Approximate Distance's
1	.615	4.6	3.5, 5 , 6
2	.357	3.21	2, 5, 4
3	5.26	7.1	11, 9 ,15

Table 5 showing the y-coordinates theoretical as well as practical for calculating accurecy.

7.1 Screen Shots

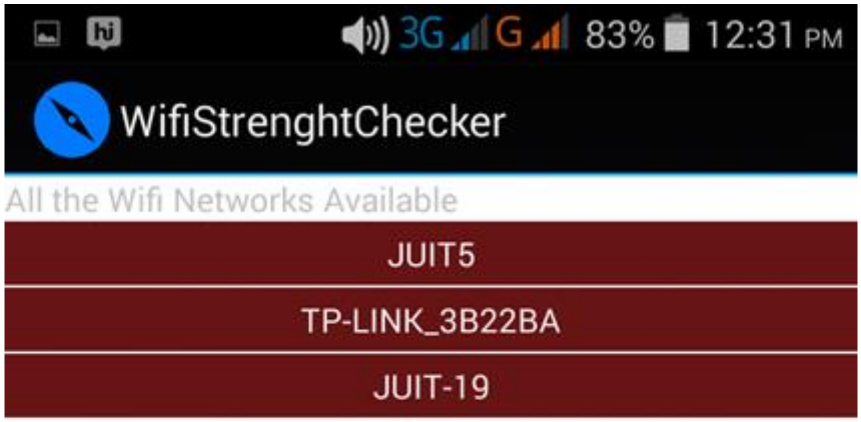

Figure 12: App showing all the wifi's in the region

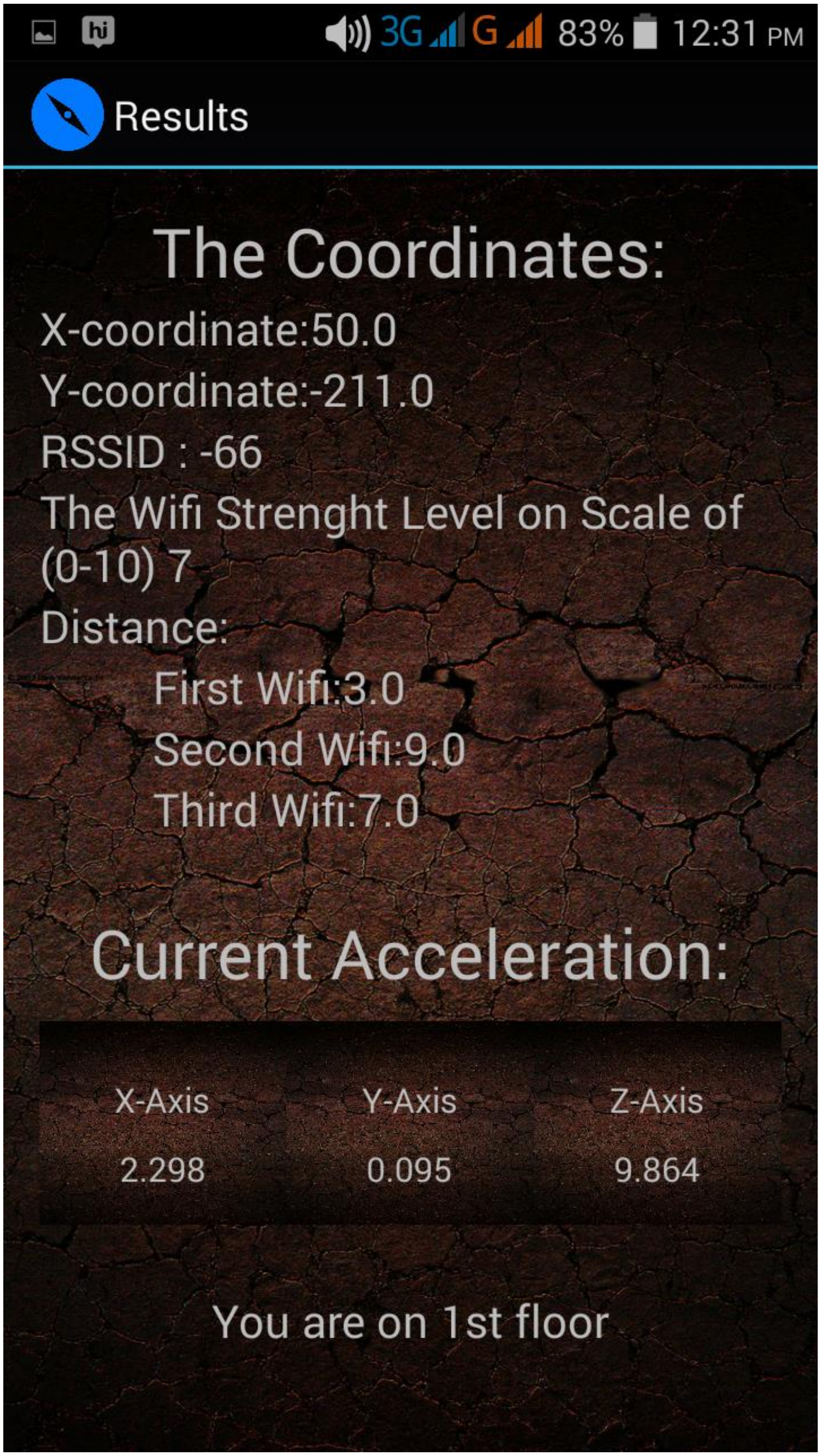

Figure 13: 14 using the trilateration algorithm , wifi Strength , rssi value of a wifi on which the user clicked, current values of the accelerometer and the floor on which the user is.

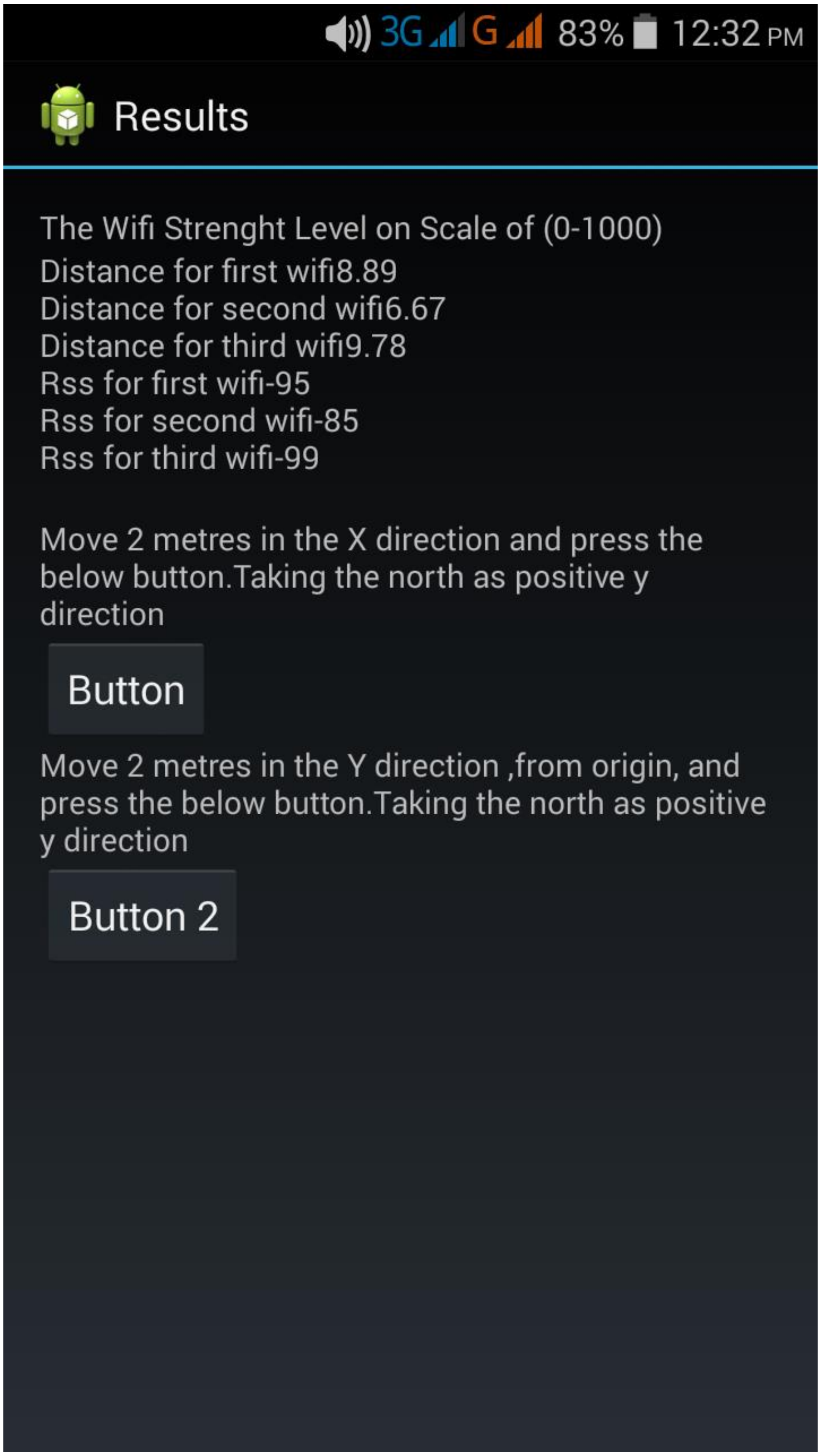

Figure 14. The above app will help to create a room where we can use the indoor localization app by getting us the coordinates of the wifi in the region. Which will help us in calculating the users coordinate using the trilateration algorithm.

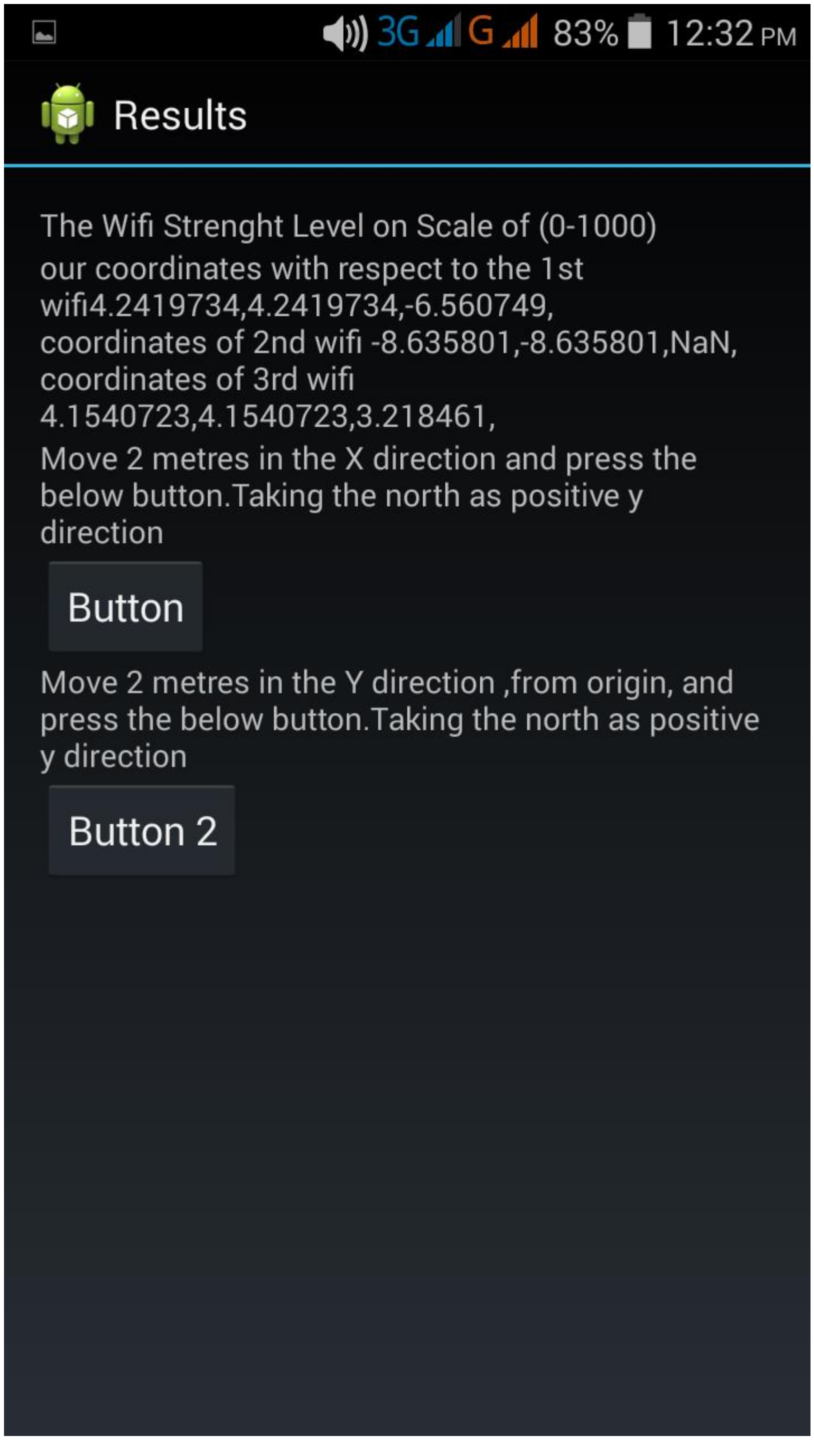

Figure 15. After going 2 meters in the x direction and clicking the first button and then going 2 meters in the y direction will give us 3 equations from which we can calculate 3 coordinates using the basic distance formula. We have assumed the coordinates of the user as zero zero in the beginning.

8. Challenges

8.1 Lack of standardization

With the Android operation system being an open source project, mobile device manufacturers can easily develop and enhance their own devices' operating systems based on Android and produce many variants of the same operating system on many other different phones. Each year, hundreds of Android based mobile devices come into the market. They come in different shapes and sizes, targeting different user groups ranging from budget to high-end. This is good for consumers but not so much for developers.
First of all, there is no standardization. The hardware specification differs from one device to another even though they could be running on similar Android version. They have different processors, rams, antennae, and specifically their Wi-Fi receivers. Even if some of them run on similar frequency bands, their data sampling rate could be different and this would greatly affect the performance and accuracy of the indoor localization application which draws data constantly from the receiver.

8.2 Signal stability

Secondly, the strength of Wi-Fi signal tends to fluctuate easily due to interference and especially so when the receiver is some distance away from the access point (AP). Microwave ovens, cordless phones, Bluetooth devices, wireless video cameras, outdoor microwave links, wireless game controllers, Zigbee devices, fluorescent lights, WiMAX devices, and even bad electrical connections—all these things can cause broad RF spectrum emissions and produce interference with the Wi-Fi signal.
Since many of the interference sources are around us in everyday life, it is difficult to get a constant stabilized Wi-Fi signal without fluctuations. These fluctuations will be one of the main causes in the inaccuracy of the indoor localization application.

8.3 Power consumption

Another factor is the power efficiency of the positioning system, especially on the battery powered mobile devices. Wi-Fi modules have a substantial power consumption (about 300 mW in idle power-saving mode), which drains the battery life of the mobile device quickly.

Possible improvements

8.4 Designing for inaccuracy

One consequence of most indoor positioning systems is a lower degree of accuracy compared to GPS. Precision depends on signal fluctuations, which depend on factors like how many people are in the room, how users are holding their phone, and other vagaries.

An effective mobile app must design for this reality from the very beginning. One technique that will help users greatly is to point out quickly recognizable features of the environment. One of the most efficient and direct way to combat inaccuracy, however, is by making it as easy as possible for users to self-correct. The map of the positioning application should be robust enough for the user to drag around, zoom in and zoom out freely, and simple enough for the user to determine the exact current. If the starting location isn't perfect, the user will instinctively drag the map around and figure it out.

8.5 Making use of other sensors in device

A good way to further improve the accuracy could be to explore the possibilities of using a smart phone's inbuilt sensors to complement and enhance the accuracy of predicting a user's location, in addition to using Wi-Fi signal strength.

9. CONCLUSION & FUTURE WORK

we have surveyed key technologies about Indoor Localization and have described in detail the working of Indoor Localization systems, the most widely-employed of all systems based on Trilateration Algorithm as it is Also been used in GPS .We have also surveyed about other methods other than Trilateration method. We have also studied about accelerometer in indoor localization which will be helpful in decreasing the power consumption and and getting more better location results even in different floors. Accuracy in positioning can be improved a lot with the combination various technologies. And other than surveying We have made a working android app which tells the rssi values , distance of the user from wifi node, coordinates calculated using the trilateration algorithm and not that perfectly working floor guessing system using the accelerometer values. In future we would have made the floor guessing system work better and produce correct outputs and make the app more efficient and accurate.We would also try to find the shortcomings of existing Trilateration method and work on them, and would have made the app more energy efficient using the accelerometer. We wanted to add the coordinates calculated by my other app to a database but due to some errors couldn’t do that which would have have resulted in more accurate results and dynamically creating a room where we could have used the indoor localization app.

10. REFERENCES

1. OnkarPathak, Pratik Palaskar, Rajesh Palkar, Mayur Tawari " Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points –A Tri-lateration Approach " International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014

2. Mikkel Baun Kjærgaard, Mads Vering Krarup, "Indoor Positioning using Wi-Fi – How Well Is the Problem Understood?", Department of Computer Science, Aarhus University, Denmark 28-31st October 2013.

3. Marwan Al- Jemeli & Fawnizu Azmadi Hussain " On location estimation for Method Wireless Sensor Nodes" Maxwell Scientific Organization Research Journal of Applied Sciences, Engineering and Technology8(1):124-130,2014
4. Hung-Huan Liu , Yu-Non Yang ," WiFi-Based Indoor Positioning for Multi-Floor Environment", IEEE 2011.
5. LuisPeneda, AbílioAzenha, AdrianoCarvalho , "Trilateration for Indoors Positioning Within the Framework of Wireless Communications", IEEE 2009.
6. Christos Laoudias, George Larkou, Chin-Lung Li "Accurate Multi-Sensor Localization on Android Devices" KIOS Research Center University of Cyprus ,2014.
7. Martin Klepal, Christian Beder "MapUme – WiFi Based localization system" Cork Institute of Technology Nimbus, Bishopstown Cork, Ireland ,2011.
8. http://www.wikipedia.com

9. Android http://www.source.android.com/.

10. Indoor localization startups https://angel.co/**indoor-positioning**