

# INTRODUCTION

## SIGNAL MAP

# **Chapter 1: Introduction**

## **1.1) Introduction to Signal Map**

As the modern era proceeds, people inevitably tend to keep their smartphones with them all the time, which gives any application to reach out to the masses very easily and impact their lives. Majority of the people who are engaged in some kind of business tend to travel a lot which makes them go in and out of the area in the reach of their corresponding network providers. Because of this, there are constant glitches and cut offs during an ongoing conversation on the phone. The only way to eradicate this constant problem is to help them find a reliable network and that's what we aim at doing. There have been a lot of cases at airports, railway stations, during travels or even at home where people inadvertently go out of the reachable zone and hence the connection goes down. By the help of our application they could know where exactly they got to be when they want a completely uninterrupted conversation.

The similar problem persists with the Wi-Fi signals perhaps because of the inability of the router to reach to a wider area or due to the various obstacles that come in the way. This leads to interruption in various online facilities and most importantly while making important transactions over the internet. In order to deal with such a scenario the user needs to figure out the location where he/she needs to be to carry on with a reliable connection.

The Google Maps API for Android provides developers with the means to create apps with localization functionality. Version 2 of the Maps API was released at the end of 2012 and it introduced a range of new features including 3D, improved caching, and vector tiles. The proposed app will use Google Maps for Android V2 in conjunction with the Google Places API. The app will present a map to the user, mark their current location and nearby network strength depending on which carrier the user is on, and will update when the user checks in.

The processes required to integrate Google Maps and Google Places with Android apps. Android gives your applications access to the location services supported by the device through classes in the `android.location` package. The central component of the location framework is the `LocationManager` system service, which provides APIs to determine location and bearing of the underlying device.

## **1.2) Existing System**

Our application domain is very distinct from the traditional technologies and thus there is hardly any work done to find the solution to the problem that we are trying to address. Through extensive research, we have come to find a couple of systems that appear to be nearly related to our application domain. One of the best is an android application called as OpenSignal. This application, though provides a lot of functionalities including- map cellular coverage, find Wi-Fi hotspots, test and improve reception & get faster data.

The existing system, though provides some valuable information related to cellular strength, but this information is very restricted and frequently rife with errors. The app help the user to search for better signals by showing the direction from where the signal is coming and the user has to navigate in that direction to receive better signals. This approach is not ideal because it is nebulous and vague and doesn't give concrete solution. The existing system has several other shortcomings such as-

- Not all towers and wireless routers are in the database,
- Users will see only one connected tower at a time.
- Moving in the direction of cell tower will not always improve reception due to interference effects.
- User interface is not very intuitive.
- Thus, the existing system is still in its primary stages and thus enough research needs to be done in this domain.

## **1.3) Problem Definition**

In the recent days reliance on telecommunication has surpassed the threshold and it has become inevitable to keep connected to cellular signals. Since it is not possible to achieve ideal cellular strengths at all locations, people face difficulties while travelling to places where cellular strength is very weak. As of now there are no tools to overcome this problem and thus our application domain includes finding solutions to the problem of detecting good and poor signal strengths in the nearby area. Our application domain includes finding solutions to the problem of detecting good and poor signal strengths in the nearby area. Now, since the problem is associated with the

mobility of smartphones, the solution also need to be mobile. One of the best way to implement such a solution would be developing an application for the same smartphones which caused the problem in the first place. With the help of such an application the user will be able to differentiate low signal strength zones to that of the high and medium zones where communication doesn't completely fade away. The concerned metrics such as network provider, signal strength, location, etc would be visible to the user on the interface of the application which itself would be based on Google Maps and extensive use of GPS available on the mobile phone.

#### 1.4) Proposed System

With signal map we are trying to address the problems that are being faced by millions of people in their daily lives of restricted mobile and W-fi network signal strengths. Signal map is an android application that provides users with the information of various network carriers and their signal strengths around them on a Google Map. This representation would depend on which carrier user is on. For Example, if the user has a Vodafone network then the map would show cellular information relating to Vodafone only. The similar concept applies to wi-fi signals also.

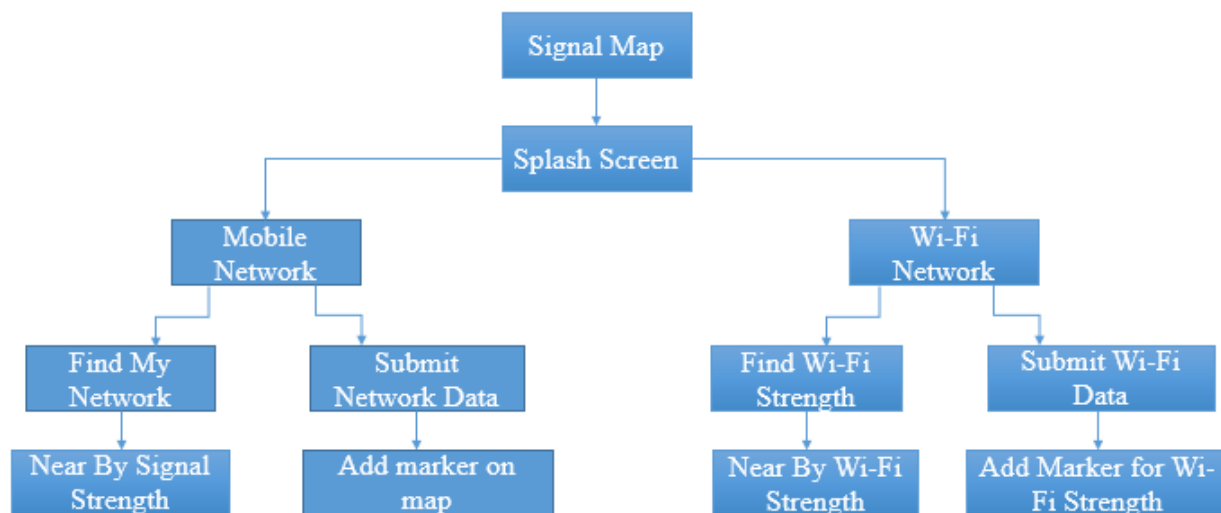


Fig 1.4(a) Modules of the proposed model

## 1.5) Scope of the problem

With Signal Map, we can map cellular coverage, find Wi-Fi hotspots, test and improve your reception & get faster data.

- Wi-Fi map allows you to easily locate nearby public networks.
- Coverage maps allow you see the best places for signal.
- Network Rank allows you to see which carrier is best in your area.
- Keep track of your monthly data, text and voice usage so you don't go over your limit.
- See a map of local cell towers.
- Help improve your coverage and Wi-Fi maps.
- All the users can view other reviews and based on that, they will get to know which are the places on the map with good signal strength as well as good wi-fi strength.
- The user can generate their reviews by adding a marker on their current location and reviewing about the signal status at that particular location.

## **1.6) Motivation**

In today's fast paced life, communication is the key aspect for all the people. We come across various situations when we are attending a very important phone call but due to network issues, we are not able to get a hassle free communication. Also when we are surfing the internet to perform various important tasks such as transactions, video conferencing, many a times it happens that the wi-fi connection is low in certain areas. In order to tackle such a problem, we got the idea to develop something that is at our fingertips and with the help of which we can easily locate the places with good signal and wi-fi strengths on the Google map. ANDROID being the most popular platform of development now a days, is the perfect technology for us to work upon. The inception of mobile computing has given rise to a whole lot of new technologies and opportunities that are on a rise every minute and as being a student of computer engineering we are particularly inclined towards learning this programming paradigm. Also being an upcoming technology, android has a lot of scope for developers to experiment and come up with more innovative ideas and enhancements.

As Android is becoming more popular in the current era and has captured the major smartphone market demanding rapid growth in application development, and due to its open source platform it is easy for developers to develop new applications on this platform to satisfy current user needs. It has all the features that a developer requires to create the best applications. In the current era, most of the mobile companies are manufacturing smartphones and tablets on the Android platform which is used by hundreds of millions of users nowadays motivating the developers to create new applications on the Android platform.

# PLANNING PHASE

## **2.1) Planning Of Project**

### **2.1.1) Project Schedule and Milestones**

#### **WEEK 1**

- Research on Various storage options in Android platform including SQL lite database and external storage.

#### **WEEK 2**

- Populating SQL Lite database with acquired location coordinates- Front End.

#### **WEEK 3**

- Populating SQL Lite database with acquired location coordinates

#### **WEEK 4**

- Modifying the existing user interface to incorporate and present the changes made to the back end.

#### **WEEK 5**

- Adding a new activity for navigation through the application.

#### **WEEK 6**

- Modifying the manifest file and working on the front end of the recent created activity.

#### **WEEK 7**

- Created main activity for WiFi module.

#### **WEEK 8**

- Adding location manager and WiFi manager in the existing WiFi module.

#### **WEEK 9**

- Modifying the manifest file and working on the front end of the recently created activity activitymain\_wifi.

#### **WEEK 10**

- Created database for the WiFi module and maintaining addition and deletion of entry



## WEEK 11

- Working on Report

## WEEK 12

- Final review of report and implemented application.

## 2.2) System Requirement Specification

### 2.2.1) Hardware requirements

The following are the hardware requirements developing an android application:

- Intel Pentium III processor at 750 MHz or faster
- minimum of 512 MB memory RAM
- Internet access via modem - 56.6kbps modem connection
- Screen Resolution - 800 x 600 or greater is preferred

CPU : ATLEAST PENTIUM IV

COPROCESSOR : BUILT IN

TOTAL RAM : MINIMUM 128 MB

DISKETTE A : MINIMUM 1.44 MB FLOPPY 3.5"

HARD DISK : 40 GB OR MORE

DISPLAY : SVGA COLOR

For running our android application, the hardware requirement would be as follows:

- Any device running on operating system ANDROID version 1.6 or higher.

## 2.2.2) Software requirements

FRONT END: XML

BACK END: JAVA

The following are the software requirements in order to develop an android application:

- JAVA DEVELOPMENT KIT (JDK)

Java applications are executed using the Java Development Kit which contains the tools required for compilation, debugging and running. Class libraries can be packaged into single JAR file with the help of programming tools such as javac, jar, archiver, etc. which make up the primary components of JDK. With the help of this tool, various components can be easily managed. These components include JAR files, javadoc which generates document from source code, jdb which performs debugging, jps which is the process status tool and displays process details for the java processes, javap which is the class file disassembler and many others. The JDK also comes with a complete Java Runtime Environment, usually called a private runtime. It consists of a Java Virtual Machine and all of the class libraries present in the production environment, as well as additional libraries only useful to developers.

The JDK has as its primary components a collection of programming tools, some of which are:

- appletviewer – this tool can be used to run and debug Java applets without a web browser
- apt – the annotation-processing tool
- extcheck – a utility which can detect JAR-file conflicts
- idlj – the IDL-to-Java compiler. This utility generates Java bindings from a given Java IDL file.
- java – the loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, no longer comes with Sun JDK, and instead it has been replaced by this new java loader.
- javac – the Java compiler, which converts source code into Java bytecode
- javadoc – the documentation generator, which automatically generates documentation from source code comments

- jar – the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.

- JAVA COMPATIBLE IDE (Eg: ECLIPSE)

Eclipse Classic (32 bit) includes tools made to give developers the freedom of choice in a multi-language, multi-platform, and multi-vendor environment. Eclipse provides a plug-in based framework that makes it easier to create, integrate, and utilize software tools, saving time and money. By collaborating and exploiting core integration technology, tool producers can leverage platform reuse and concentrate on core competencies to create new development technology. The Eclipse Platform is written in the Java language and comes with extensive plug-ins. The Eclipse Process Framework (EPF) aims at producing a customizable software process engineering framework, with exemplary process content and tools, supporting a broad variety of project types and development styles.

Eclipse is a multi-language software development environment comprising a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java, Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Perl, PHP, Python, R, Erlang and many more. The packages for the software Mathematica can also be developed using Eclipse. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Eclipse users can also contribute to its abilities by developing their own plug-in modules for different programming languages. Eclipse supports development for Tomcat, GlassFish and many other servers and is often capable of installing the required server (for development) directly from the IDE. It supports remote debugging, allowing the user to watch variables and step through the code of an application that is running on the attached server.

- ANDROID SDK TOOLS

The Android SDK includes a variety of tools that help you develop mobile applications for the Android platform. The tools are classified into two groups: SDK tools and platform tools. SDK tools are platform independent and are required no matter which Android platform you are developing on. Platform tools are customized to support the features of the latest Android platform<sup>[1]</sup>.

The Android SDK (software development kit) is a set of development tools used to develop applications for Android platform. The Android SDK includes the following:

- The libraries that are needed for the application development
- Tool for debugging
- An Emulator for running the application
- Documentation to deal with various Android APIs
- Sample Codes
- Basic Tutorials for developing Android applications.

A new SDK along with a latest android version is the trend that Google has followed and in order to keep the applications updated with latest developments, the developers have to download and install each and every update as soon as it is released on Google servers.

Although the SDK can be used to write Android programs in the command prompt, the most common method is by using an integrated development environment (IDE). The recommended IDE is Eclipse with the Android Development Tools (ADT) plug-in. However, other IDEs, such as NetBeans or IntelliJ, will also work. Most of these IDEs provide a graphical interface enabling developers to perform development tasks faster. Since Android applications are written in Java code, a user should have the Java Development Kit (JDK) installed<sup>[1]</sup>.

- **ANDROID DEVELOPER TOOLS PLUG-IN FOR ECLIPSE**

Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.

ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute your application.

Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives you an incredible boost in developing Android applications.

ADT (Android Developer Tools) is a plugin for Eclipse that provides a suite of tools that are integrated with the Eclipse IDE. It offers you access to many features that help you develop Android applications quickly. ADT provides GUI access to many of the command line SDK tools as well as a UI design tool for rapid prototyping, designing, and building of your application's user interface<sup>[1]</sup>.

Because ADT is a plugin for Eclipse, you get the functionality of a well-established IDE, along with Android-specific features that are bundled with ADT. The following describes important features of Eclipse and ADT:

### **Integrated Android project creation, building, packaging, installation, and debugging**

ADT integrates many development workflow tasks into Eclipse, making it easy for you to rapidly develop and test your Android applications.

### **SDK Tools integration**

Many of the SDK tools are integrated into Eclipse's menus, perspectives, or as a part of background processes ran by ADT.

### **Java programming language and XML editors**

The Java programming language editor contains common IDE features such as compile time syntax checking, auto-completion, and integrated documentation for the Android

framework APIs. ADT also provides custom XML editors that let you edit Android-specific XML files in a form-based UI. A graphical layout editor lets you design user interfaces with a drag and drop interface.

### **Integrated documentation for Android framework APIs**

You can access documentation by hovering over classes, methods, or variables.

- For running the android application on the computer, the software requirements are as follows:
- EMULATOR:

The Android SDK includes a mobile device emulator — a virtual mobile device that runs on your computer. The emulator lets you develop and test Android applications without using a physical device. The emulator lets you prototype, develop and test Android applications without using a physical device. The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot place actual phone calls. It provides a variety of navigation and control keys, which you can "press" using your mouse or keyboard to generate events for your application. It also provides a screen in which your application is displayed, together with any other active Android applications. To let you model and test your application more easily, the emulator utilizes Android Virtual Device (AVD) configurations. AVDs let you define certain hardware aspects of your emulated phone and allow you to create many configurations to test many Android platforms and hardware permutations. Once your application is running on the emulator, it can use the services of the Android platform to invoke other applications, access the network, play audio and video, store and retrieve data, notify the user, and render graphical transitions and themes<sup>[8]</sup>.

The emulator also includes a variety of debug capabilities, such as a console from which you can log kernel output, simulate application interrupts (such as arriving SMS messages or phone calls), and simulate latency effects and dropouts on the data network.

- **AVD MANAGER**

The AVD Manager is an easy to use user interface to manage your AVD (Android Virtual Device) configurations. An AVD is a device configuration for the Android emulator that allows you to model different configurations of Android-powered devices. When you start the AVD Manager in Eclipse or navigate to your SDK's tools/ directory and execute `android avd`<sup>[5]</sup>.

The Android emulator supports many hardware features likely to be found on mobile devices, including:

- An ARMv5 CPU and the corresponding MMU.
- A 16-bit LCD display
- One or more keyboards (a Qwerty-based keyboard and associated Dpad/Phone buttons)
- A sound chip with output and input capabilities
- Flash memory partitions (emulated through disk image files on the development machine)
- A GSM modem, including a simulated SIM Card
- A camera, using a webcam connected to your development computer.
- Sensors like an accelerometer, using data from a USB-connected Android device.

## **2.3) Users of the system**

- General Users: All the users having an Android smartphone with the Signal Map application installed on their phones can locate the places with the best signal strengths and wi-fi strengths on the Google map.
- Administrator: The admin can make changes in the coding part and can upgrade the application with the desired technologies.

# DESIGN PHASE



## Chapter 3: Design Phase

### 3.1) System Design

#### 3.1.1) Use Case Diagram

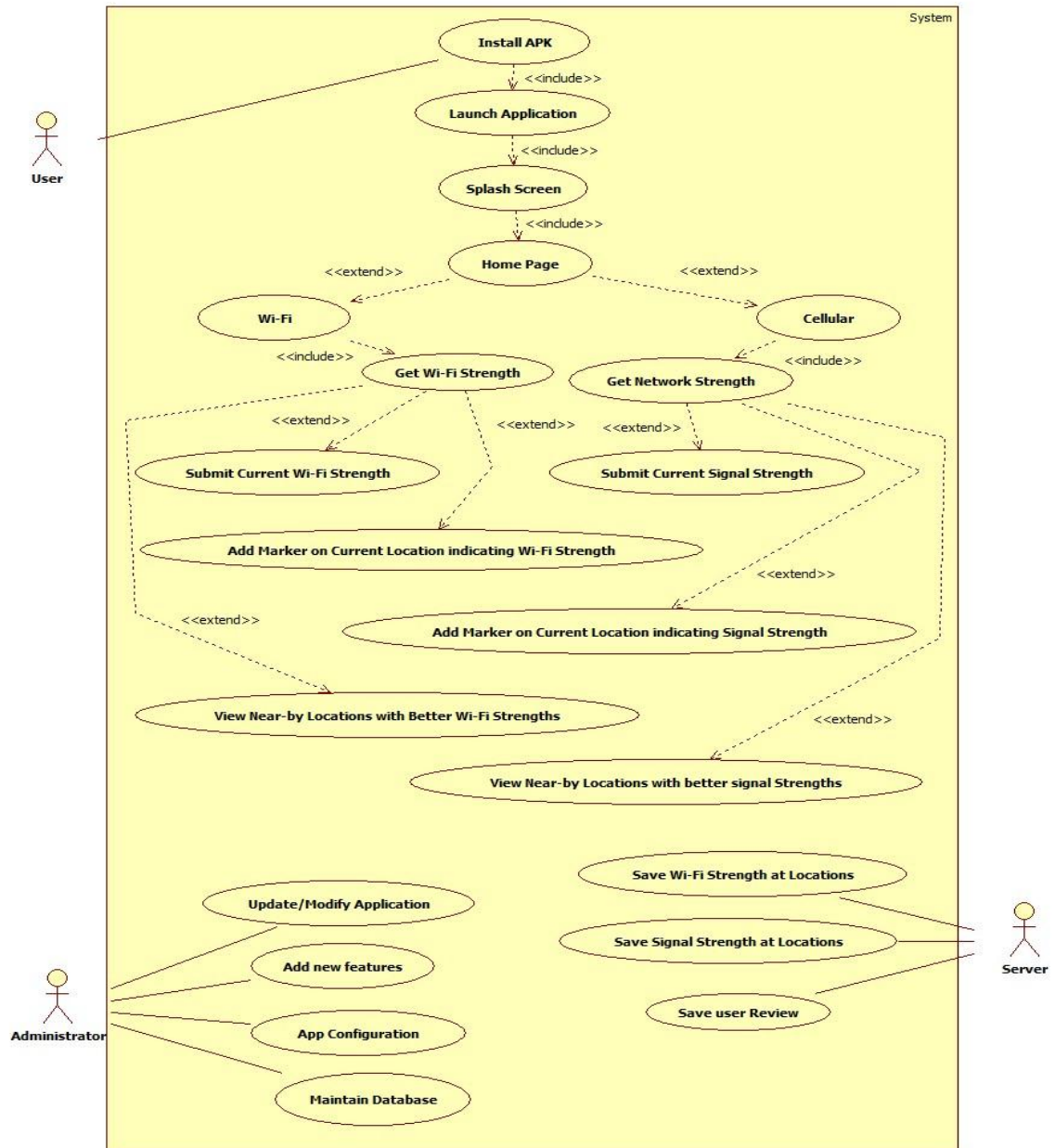


Fig 3.1.1(a) Use Case Diagram

### 3.1.2) Sequence Diagram

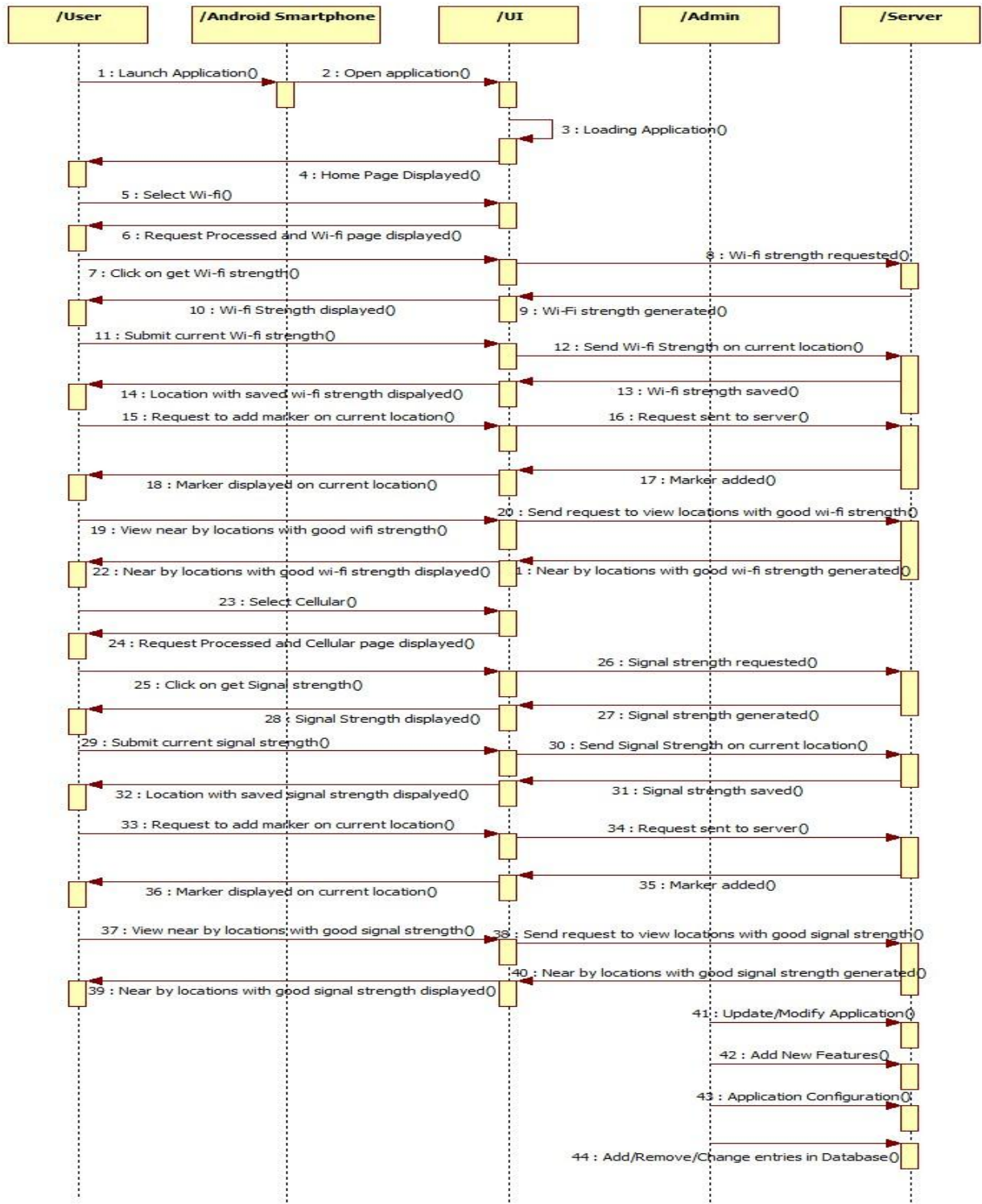


Fig 3.1.2(a) Sequence Diagram

### 3.1.3) Diagram Details

**Use case diagrams** are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

The actors in the use case are:

- User
- Admin
- Server

The major use cases are:

- Submit Current Signal Strength
- Submit current Wifi Strength

**Sequence diagram** is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.

Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines<sup>[18]</sup>.

The objects in sequence diagram are:

- User
- Android Device
- UI
- Admin
- Server

Various messages are sent between the objects in a sequence diagram.

## 3.2) Database Design

### MySQL database

#### 1) sigmap\_cellular



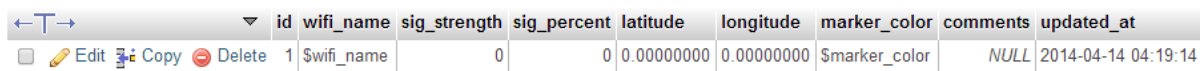
Fig 3.2(a) sigmap\_cellular table

The database table contains the following attributes:

- id - The id field gives a unique id to the entry created in the database. It is a primary key which is not null. The datatype for id is int.
- net\_name - This field provides the name of the carrier which the android device is registered to. The information of the name of the network is obtained from sim state function of telephony manager which is provided in android. The datatype for net\_name is varchar.
- net\_type – It gives the type of the network. The network type can be 2g, 3g or wifi. This information is also obtained with the help of telephony manager. The datatype for net\_type is varchar.
- sig\_strength – This field provides the signal strength of the carrier at a particular location. The signal strength is fetched from phone information of the android device. The strength of the signal can vary from 0 asu to 31 asu. ‘asu’ stands for arbitrary strength unit which is a unit for measuring signal strength. The datatype is int.
- sig\_percent – This field provides the strength of signal in percentage. It is obtained by converting the signal strength from asu to percentage. The datatype is int.
- latitude – This field gives the latitude of the location where the marker is dropped for obtaining the signal strength. The datatype is decimal.
- longitude – This field gives the longitude of the location where the marker is dropped for obtaining the signal strength. The datatype is decimal.

- marker\_color – This gives the color of marker dropped at a particular location. The color can be red, blue or green depending on the varied signal strengths. The datatype is text.
- comments – This is the comment field. The users can write their reviews in the comment field which opens on clicking the marker. The datatype is text.
- updated\_at – The last field is the field which gives the time and date when entry is created in the database. The datatype is timestamp.

## 2) sigmap\_wifi



	id	wifi_name	sig_strength	sig_percent	latitude	longitude	marker_color	comments	updated_at
1	\$wifi_name		0	0	0.00000000	0.00000000	\$marker_color	NULL	2014-04-14 04:19:14

Fig 3.2(b) sigmap\_wifi table

- id - The id field gives a unique id to the entry created in the database. It is a primary key which is not null. The datatype for id is int.
- wifi\_name - This field provides the name of the wifi which the android device is connected to. The datatype for wifi\_name is varchar.
- sig\_strength – This field provides the signal strength of the wifi which is connected. The strength of the signal can vary from 35 dbm to 95 dbm. ‘dBm’ stands for decibel milliwatts which is a unit for measuring wifi strength. The datatype is int.
- sig\_percent – This field provides the strength of wifi in percentage. It is obtained by converting the wifi strength from dBm to percentage. The datatype is int.
- latitude – This field gives the latitude of the location where the marker is dropped for obtaining the wifi strength. The datatype is decimal.
- longitude – This field gives the longitude of the location where the marker is dropped for obtaining the wifi strength. The datatype is decimal.
- marker\_color – This gives the color of marker dropped at a particular location. The color can be red, blue or green depending on the varied wifi strengths. The datatype is text.
- updated\_at – The last field is the field which gives the time and date when entry is created in the database. The datatype is timestamp.

### 3.3) Flowchart of the Application

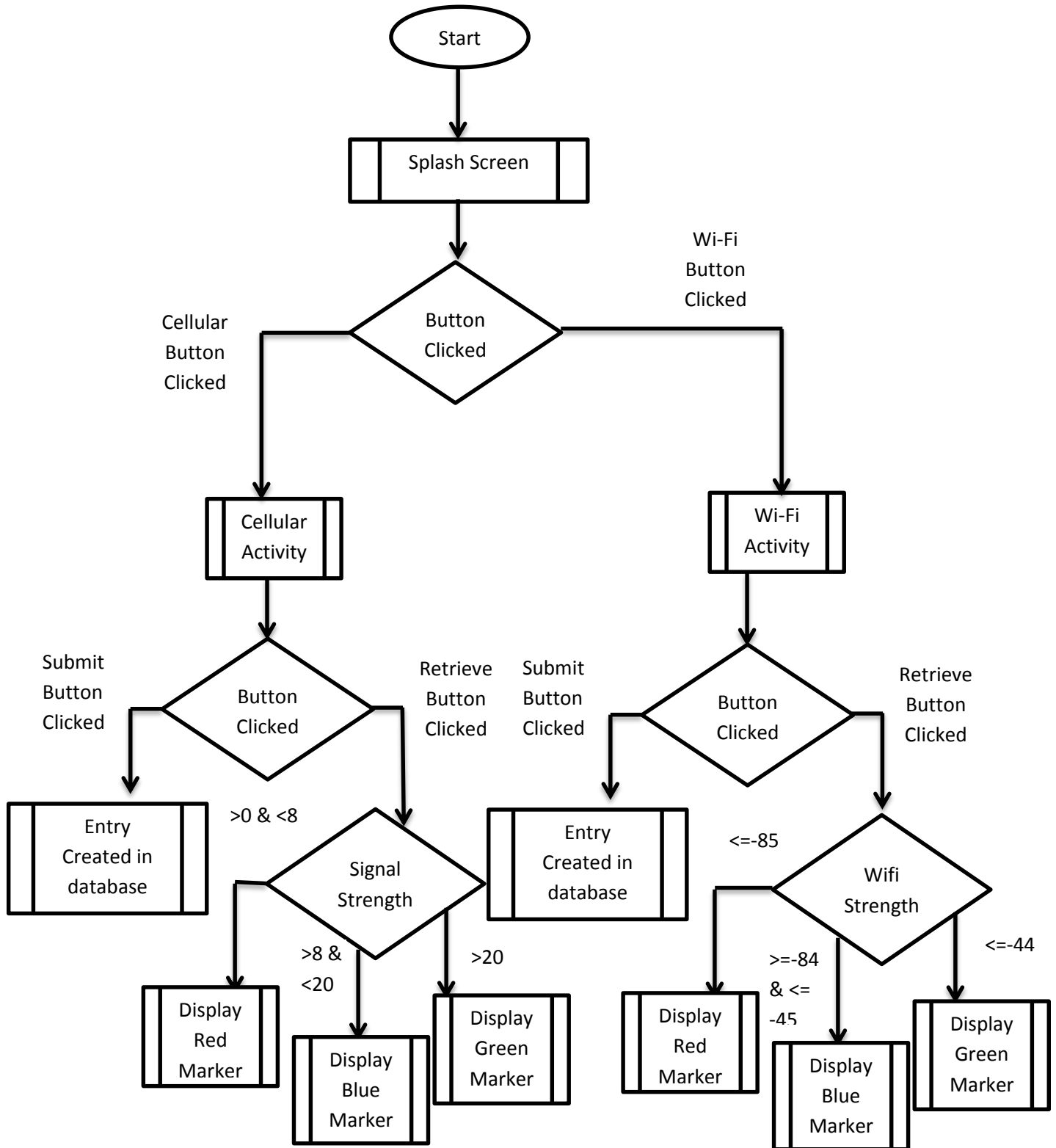


Fig 3.3(a) Flowchart

# DEVELOPMENT PHASE

## Chapter 4: Technology Used

### 4.1) Architecture

The platform we used to develop this project is ‘**Android**’. The major advantage of android is its ability to provide open source platform for developers. The components include the OS, middleware as well as important applications. Google and Open Handset Alliance developed Android OS in the year 2007<sup>[1]</sup>. Android has the following features:

- It uses technologies such as Bluetooth, Wi-Fi, GSM and many more for sending and receiving data.
- It includes all libraries for multimedia components such as audio, video and images.
- DVM – The Dalvik Virtual Machine is the same as optimized Java Virtual Machine for mobile platform.
- SQLite Database for storing data.
- Facility such as IPC for passing messages.
- In-built browser for accessing Internet.
- Library and tools for 2-dimensional and 3-dimensional graphics.
- Various features such as capturing video, touchscreen, navigation and many more.

The major elements which form the basis of android development are:

- Activity – An activity is a screen of the application which the user can view. It is the user interface component corresponding to the screen which gets displayed.
- Broadcast receiver – It reacts to an event which is triggered and gives responses to broadcast intents.
- Services – The processes or tasks which are running in the background which includes audio player, etc.
- Content Provider – This component provides feature of data to be shared with several other apps. Eg: address book.



## Android Architecture

The main components of android are as follows:

- Applications
- Application framework
- Libraries
- Linux kernel

**Applications** – This layer consists of the applications with which the user interaction takes place as well as all third party applications reside in this layer. Programs such as message, contacts, etc. are present. Java language is used for writing android apps.

**Application framework** –It consists of many components which provide a framework with libraries and tools for new applications to be developed. It includes views, resource manager, content providers, activity manager and notification manager helping the application use underlying standard libraries.

- Views – consists of buttons, textboxes, list etc
- Resource manager – provide access to non-code resources like strings, graphics etc.
- Content providers – enables application to access data from other applications.
- Activity manager – manages the life-cycle of an application.
- Notification manager – enables all applications to display custom alerts on status bar.

**Libraries** – These include the C/C++ libraries which are used by the android system and the application framework is responsible for providing these libraries. These libraries include:

- System C library – tuned for embedded Linux based devices.
- Media library – for audio, video and static image files.
- Surface manager – manages access to display subsystem
- LibWebCore – modern web browser engine
- SGL – underlying 2D graphics engine
- 3D libraries – for 3D graphics

- Freetype – for bitmap and vector rendering
- SQLite – powerful relational database engine

**Android Runtime** - Each and every application has its own instance of Dalvik VM and runs in its own process. Dalvik means multiple virtual machines can run on a device efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory foot print. The VM is register-based, and runs classes compiled by java language compiler that have been transformed into the .dex format by the included “dx” tool. Which convert generated byte code from .jar to .dex file, after this byte code becomes much more efficient to run on the small processors. As the result, it is possible to have multiple instances of Dalvik virtual machine running on the single device at the same time. The Core libraries are written in Java language and contains of the collection classes, utilities, IO and other tools. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management<sup>[1]</sup>.

**Linux Kernel** - Android Architecture is based on Linux 2.6 kernel. It helps to manage security, memory management, process management, network stack and other important issues. Android provides the support for the QualcommMSM7K chipset family. For instance, the current kernel tree supports Qualcomm MSM7200A chipsets, but in the second half of 2008 mobile devices with stable version Qualcomm MSM 7200A were introduced, which includes major features:

- WCDMA/HSUPA and EGPRS networks support.
- Bluetooth 1.2 and Wi-Fi support
- Digital audio support for mp3 and other formats
- Support for Linux and other third-party operating system.
- Java hardware acceleration and support for java applications.
- Qcamera up to 6.0 megapixels.

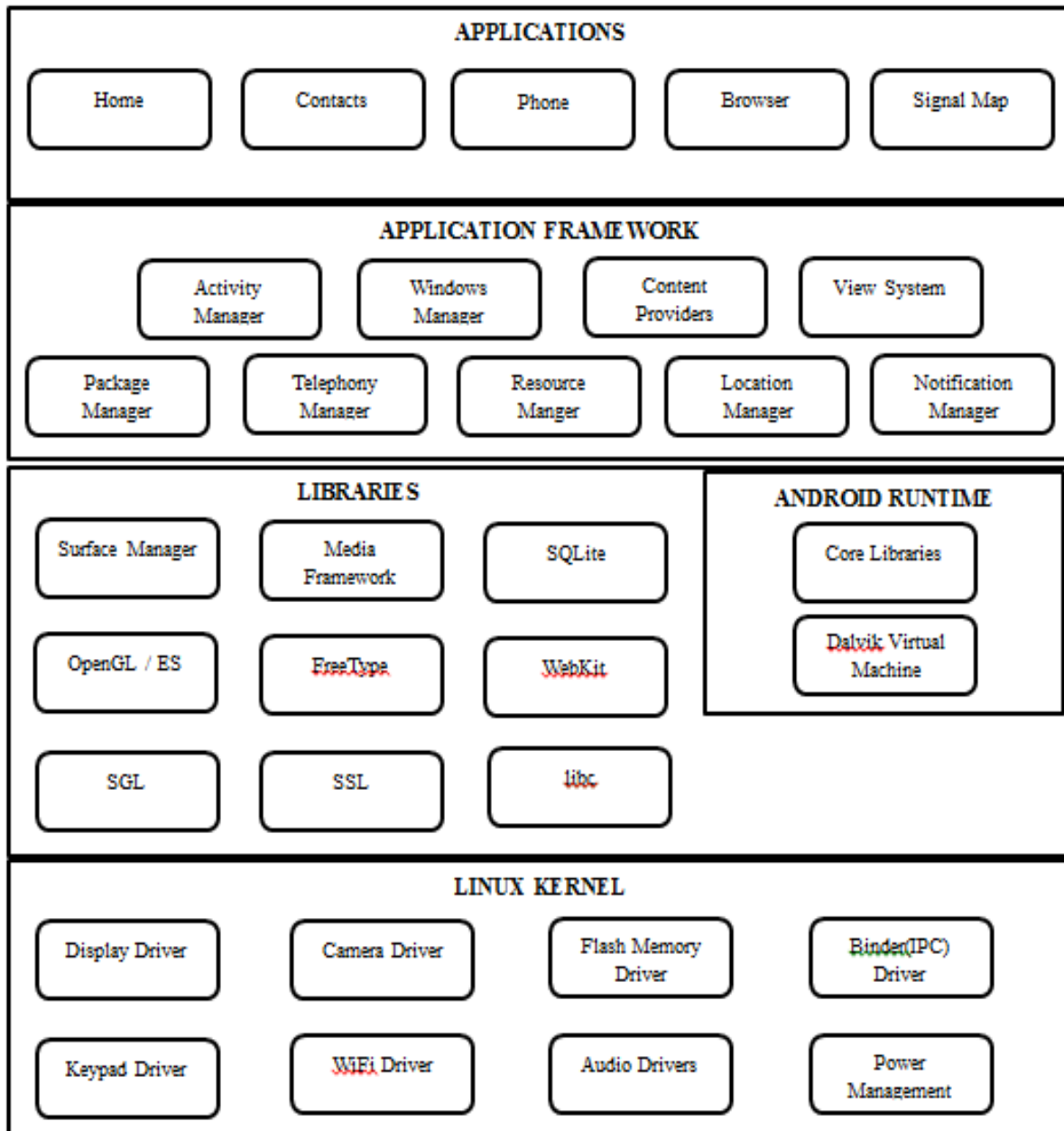


Fig 4.1(a) Architecture of Android

## **Advantages**

- It is an open platform. This means that its code is available for people to look at.
- Without having to make a complete change we can switch from one application to another.
- Android allows third parties to make applications for the phone that can be installed by anyone.
- An Android phone is guaranteed to work with Google products.
- Android platform will work on netbook and computers. This means that you could have device that share the same platform giving you the ability to purchase applications that will work on all devices.
- Multitasking
- Android gives better notification.
- Application freedom is guaranteed.
- Android allows customization of home page and use of widgets.

## **Android Activity**

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with `setContentView(View)`. There are two methods almost all subclasses of Activity will implement:

- `onCreate(Bundle)` is where you initialize your activity. Most importantly, here you will usually call `setContentView(int)` with a layout resource defining your UI, and using `findViewById(int)` to retrieve the widgets in that UI that you need to interact with programmatically.
- `onPause()` is where you deal with the user leaving your activity. Most importantly, any changes made by the user should at this point be committed (usually to the `ContentProvider` holding the data).

The following diagram shows the important state paths of an Activity. The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.

⇒ State Paths of an Android Activity

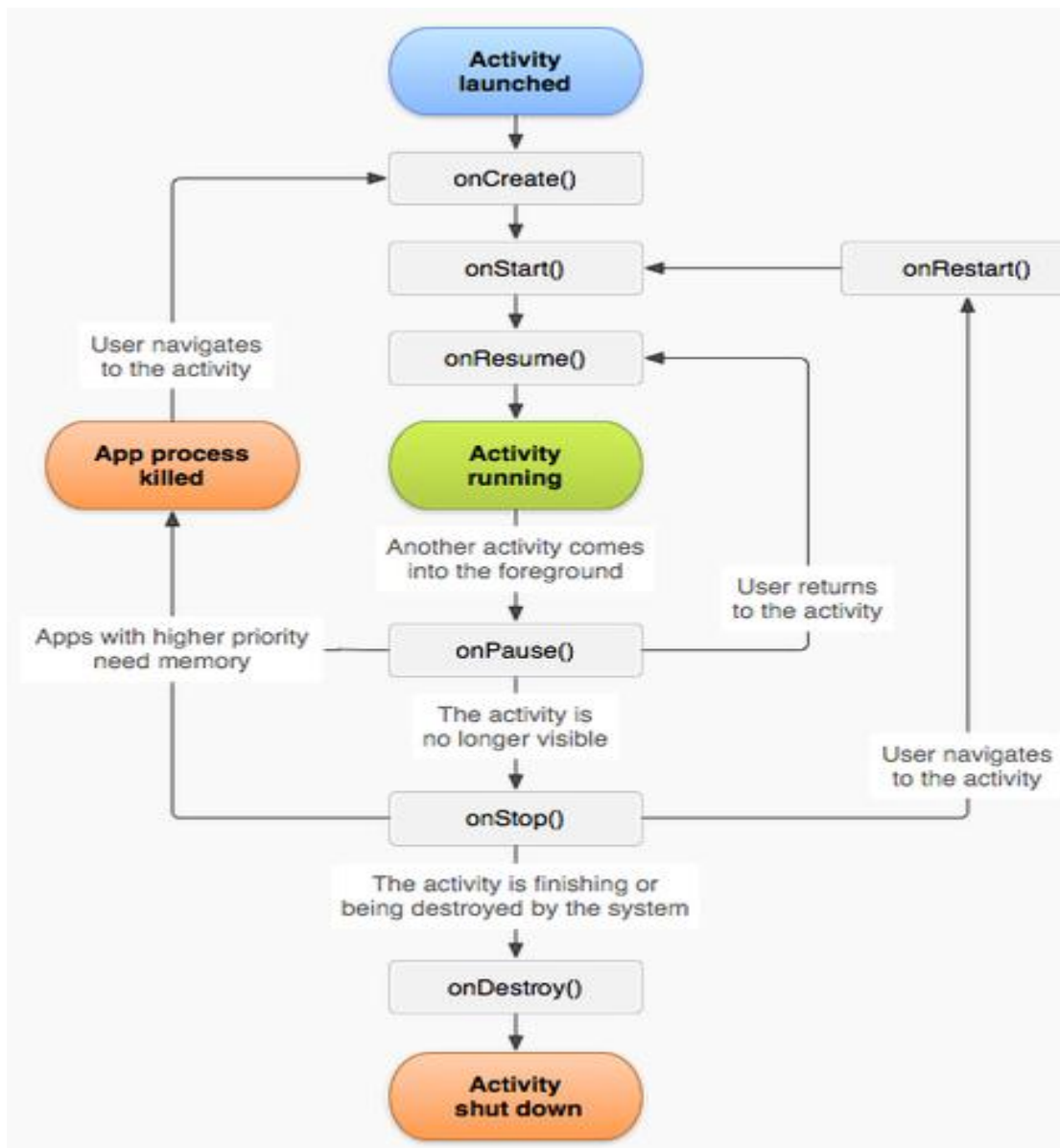


Fig 4.1(b) Android Activity Diagram

## 4.2) Experimental setup and tools as well as techniques used

The experimental setup of the project includes the following:

- A laptop or a desktop
- Eclipse IDE installation
- Java Development Kit
- Android SDK tools
- Android Developer Tools

The different tools and techniques used in the project are as follows:

### ▪ JAVA

Java is a small, simple, safe, object oriented, interpreted or dynamically optimized, byte coded, architectural, garbage collected, multithreaded programming language with a strongly typed exception-handling for writing distributed and dynamically extensible programs. Java is an object oriented programming language. Java is a high-level, third generation language like C, FORTRAN, Small talk, Pearl and many others. You can use java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do. Special programs called applets that can be downloaded from the internet and played safely within web browser. Java supports this application and the following features make it one of the best programming languages<sup>[5]</sup>.

- It is simple and object oriented
- It helps to create user friendly interfaces.
- It is very dynamic.
- It supports multithreading.
- It is platform independent.
- It is highly secure and robust.
- It supports internet programming.

- XML

**Extensible Markup Language (XML)** is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

The design goals of XML emphasize simplicity, generality, and usability over the Internet.<sup>[6]</sup> It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

Many application programming interfaces (APIs) have been developed to aid software developers with processing XML data, and several schema systems exist to aid in the definition of XML-based languages.

XML specifies neither semantics nor a tag set. In fact XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them. Since there's no predefined tag set, there can't be any preconceived semantics. All of the semantics of an XML document will either be defined by the applications that process them or by stylesheets.

Main features of XML are:

- XML is a markup language much like HTML
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive
- XML files are text files, which can be managed by any text editor.
- XML is very simple, because it has less than 10 syntax rules.
- XML is extensible, because it only specifies the structural rules of tags. No specification on tags them self.

## **PHP**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML<sup>[14]</sup>.

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something". The PHP code is enclosed in special start and end processing instructions `<?php` and `?>` that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer<sup>[14]</sup>.

### **PHP in our Application:**

Due to the specifics of databases in our project we ought to use phpmyadmin with My SQL database instead of SQL Lite database. Therefore in order to connect, maintain and perform other CRUD operations on My SQL database from the android application, we made use of several PHP files which interact with the database hosted on the server and return the values as json. This json script is interpreted by the application program and perform required database functionalities.



## **JSON**

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language<sup>[16]</sup>.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

### **Json in our application:**

We have used json as an output from the php files (that interacts with the databases) and this output is fed to the application, which, with the help from JSONParser file, interprets this json input in a format that android architecture supports and understands.

## **Databases Used:**

### **SQL Lite**

SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. The database requires limited memory at runtime (approx. 250 KByte) which makes it a good candidate from being embedded into other runtimes.

SQLite supports the data types TEXT (similar to String in Java), INTEGER (similar to long in Java) and REAL (similar to double in Java). All other types must be converted into one of these fields before getting saved in the database. SQLite itself does not validate if the types written to the columns are actually of the defined type, e.g. you can write an integer into a string column and vice versa<sup>[16]</sup>.

### **SQL Lite in Android**

SQLite is embedded into every Android device. Using SQLite database in Android does not require a setup procedure or administration of the database. You only have to define the SQL statements for creating and updating the database. Afterwards the database is automatically managed for you by the Android platform.

Access to an SQLite database involves accessing the file system. This can be slow. Therefore it is recommended to perform database operations asynchronously.

If your application creates a database, this database is by default saved in the directory `DATA/data/APP_NAME/databases/FILENAME`.

The parts of the above directory are constructed based on the following rules. DATA is the path which the `Environment.getDataDirectory()` method returns. APP\_NAME is your application name. FILENAME is the name you specify in your application code for the database.

## **My SQL**

PhpMyAdmin is one of the most popular applications for MySQL databases management. It is a free tool written in PHP. Through this software you can create, alter, drop, delete, import and export MySQL database tables. You can run MySQL queries, optimize, repair and check tables, change collation and execute other database management commands

## **My SQLite Android**

Android cannot connect directly to the database server. Therefore we need to create a simple web service that will pass the requests to the database and will return the response.

First your android app calls a PHP script in order to perform a data operation, lets say “create”. The PHP script then connects to your MySQL database to perform the operation. So the data flows from your Android app to PHP script then finally is stored in your MySQL database

## **SqlLite vs My SQL**

### **SQLite:**

- easier to setup
- great for temporary (testing databases)
- great for rapid development
- great for embedding in an application
- doesn't have user management
- doesn't have many performance features
- doesn't scale well.

### **MySQL:**

- far more difficult/complex to set up
- better options for performance tuning
- can scale well if tuned properly
- can manage users, permissions, etc

### **Our Decision:**

One of the deciding parameters for database to be used for the application was the fact that we wanted a database that can be hosted on a server and since sqllite database is always maintained locally on the device, we decided to go with My SQL database in our project since our requirements were in sync with My SQL's capabilities.

## **4.3) Modules Developed in our Application**

### **Splash Activity:**

This is the first screen that will show up with just a little animation to inform the user of the app being initialized on the user's device.

### **Option Screen:**

Option screen is the activity class to which the user is redirected automatically as the timer in the 'splash activity' class is timed out. This activity presents users with the options to either select 'Wifi' module or 'Cellular' module.

It just makes use of two buttons and their 'setOnClickListener' class to redirect accordingly.

### **Main Activity:**

It is the activity class that is responsible to handle everything that goes on into the cellular module which ranges from fetching network data from the device to that of integrating and interacting with the Google Maps.

### **MainActivity\_WiFi:**

This activity is quite similar to main activity and performs mostly the same functionalities but with specific focus on WiFi rather than cellular. It is the activity class that is responsible to handle everything that goes on into the WiFi module which ranges from fetching WiFi data from the device to that of integrating and interacting with the Google Maps.

As mentioned earlier, the process of fetching data from My SQL database is being done with the help of JSON (Javascript Object Notation). The json data that is coming from the php files is

interpreted to Android's compiler with the help of a predefined JSON class which in this case is 'JSONParser' activity.

**Manifest.xml:**

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code.

**Connecting to the database:**

Since we are using My SQL database, the process of connecting and implementing database functionalities are performed with the help of PHP.

# TESTING PHASE

## **Chapter 5: Testing Phase**

### **5.1) Types of Testing**

#### **Introduction**

This document is a high-level overview defining our testing strategy for the Signal Map application. Its objective is to communicate project-wide quality standards and procedures. It portrays a snapshot of the project as of the end of the planning phase. This document will address the different standards that will apply to the unit, integration and system testing of the specified application.

#### **Test Objective**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

#### **Unit Testing**

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

In order to perform unit testing, we went through the code to test the logical flow of the various functions that have been implemented in the program.

For example- We tested the flow of the function responding to the add marker button click event. The received program flow was in close proximity to that of the expected.

Several other functions were tested in the same way.

## **Integration Testing**

There are two primary modules that will need to be integrated: the Graphic User Interface module and the database module (back-end). The two components, once integrated, will form the complete Signal Map Application. The following describes these modules as well as the steps that will need to be taken to achieve complete integration. We will be employing an incremental testing strategy to complete the integration.

### **Module 1 - Graphic User Interface (GUI) Module**

This module provides a simple GUI where the user can perform the different actions (functions). This module will be tested separate from the backend to check if each interface (e.g. insert button) is functioning properly, and in general, to test if the click-event actions are working properly. The testing will be performed by writing a stub for each element in the interface.

### **Module 2 – Database Backend Module**

The “database” provides the storage for the data elements and implements the algorithms and associated functionality of the Signal Map. This module will be tested separate from the GUI by printing out the results to the Console. In testing this module we will follow the incremental testing method i.e. testing one function first and then keep adding additional function and test it again until all the required functions are tested.

When the GUI is combined with the backend module, we will have a complete application. To achieve complete integration of these two modules, we will test each element in the GUI by replacing the stubs with the appropriate function from the back end. The results will be displayed within the GUI instead of through the Console. In testing the combined modules, we will follow the incremental testing method. Each stub will be replaced one at a time and tested. This will be done until all stubs have been replaced by the appropriate functions from the backend.



## **System Testing**

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case we will focus only on function validation and performance. And in both cases we will use the black-box method of testing.

## **Function Validation Testing**

The integrated “Signal Map Application” will be tested based on the requirements to ensure that we built the right application. In doing this test, we will try to find the errors in the inputs and outputs, that is, we will test each function to ensure that it properly implements the value fetch and stores, and that the resulting data displays the values in the proper location graphically. The behavior of each function, as well as their respective algorithms, are contained in the Software Program Specification.

In addition, we will test:

- The interfaces to ensure they are functioning as desired (i.e. check if each interface is behaving as expected, specifically verifying the appropriate action is associated with each onclick event).
- The interaction between the GUI and the backend repository. In this case the data will be inserted and check if they are processed in the backend and give the expected output.

### **Performance testing**

This test will be conducted to evaluate the fulfillment of a system with specified performance requirements. It will be done using black-box testing method. And this will be performed by:

- Storing the maximum data in the file and trying to insert, and observe how the application will perform when it is out of boundary.
- Trying to store new data and check if it over writes the existing once.
- Trying to load the data while they are already loaded

## 5.2) Testing Screenshots

### TEST 1 - Disenabled Provider GPS

When the GPS is turned off or the android device is in airplane mode, then a toast appears disenabled provider GPS as the current location cannot be fetched.

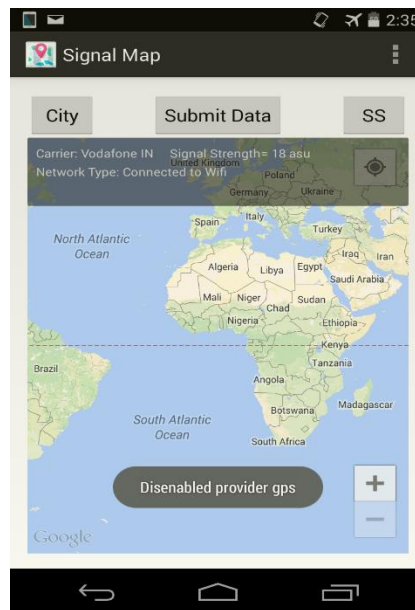


Fig 5.2(a) Disenabled Provider GPS

### Cellular Page without any error

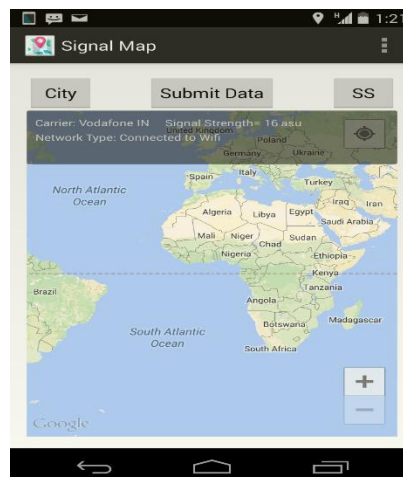


Fig 5.2(b) Cellular Page Working

## TEST 2 – App not responding

The app not responding due to multiple requests from user.

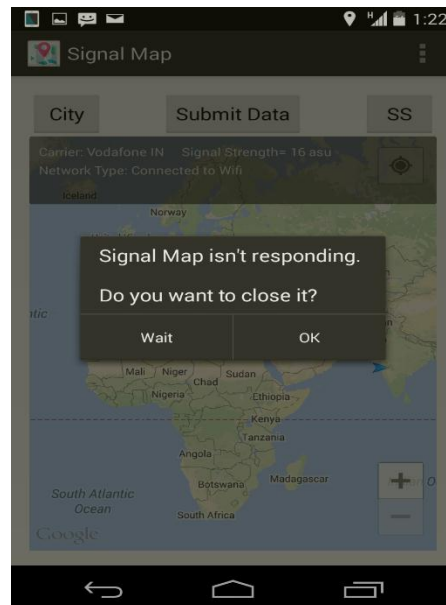


Fig 5.2(c) App not responding

## TEST 3 – Information not getting displayed

The text fields do not display the carrier information and the signal strength. Instead they are displaying the hardcoded text.

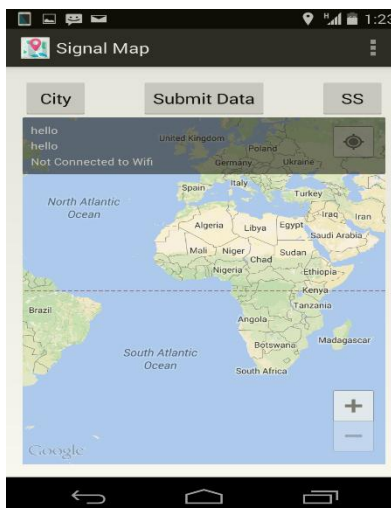


Fig 5.2(d) Incorrect Information

## Wifi Page with proper information

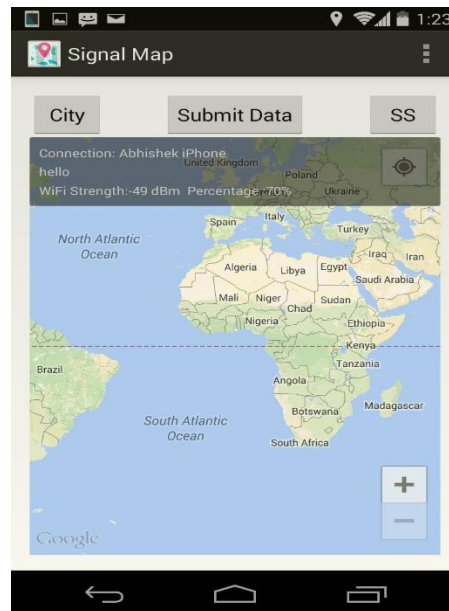


Fig 5.2(e) Wifi Page Working

# USER MANUAL

## Chapter 6: User Manual

### 6.1) User Manual

- **Icon of our application**

You can launch the application by clicking on the app icon.

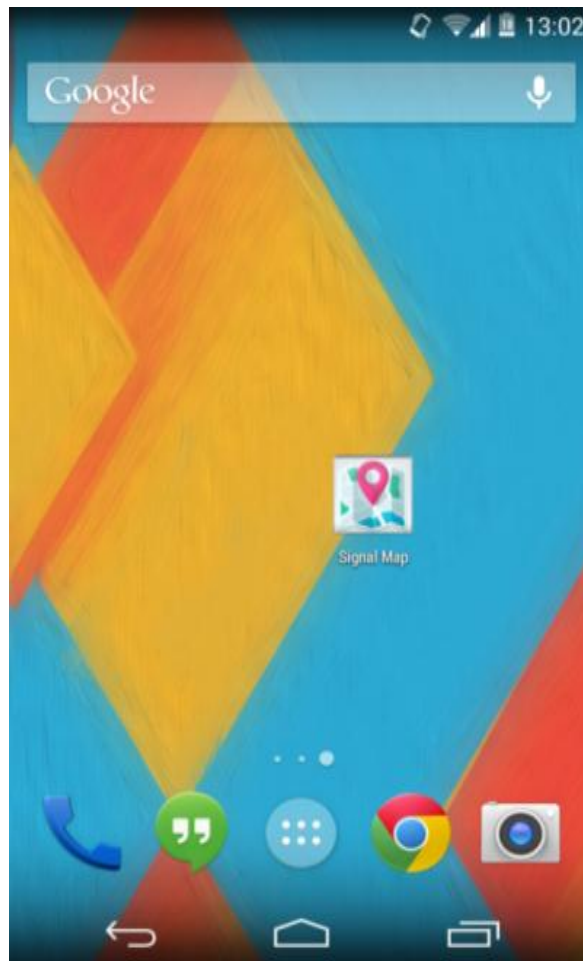


Fig 6.1(a) Application Icon

- **Splash Screen**

Once the app is launched, you will see the splash screen that connects to the internet to fetch the current map configurations.

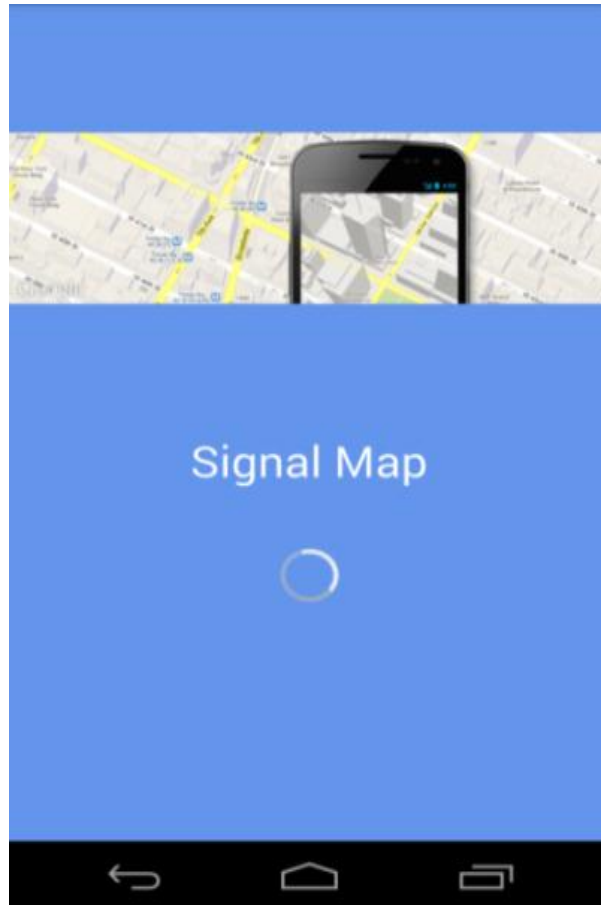


Fig. 6.1(b) Splash Screen



- **Option Page**

When you open our application on your android device, after the splash screen gets loaded, you will see an option page. You can either choose cellular or wifi by clicking either of the buttons.

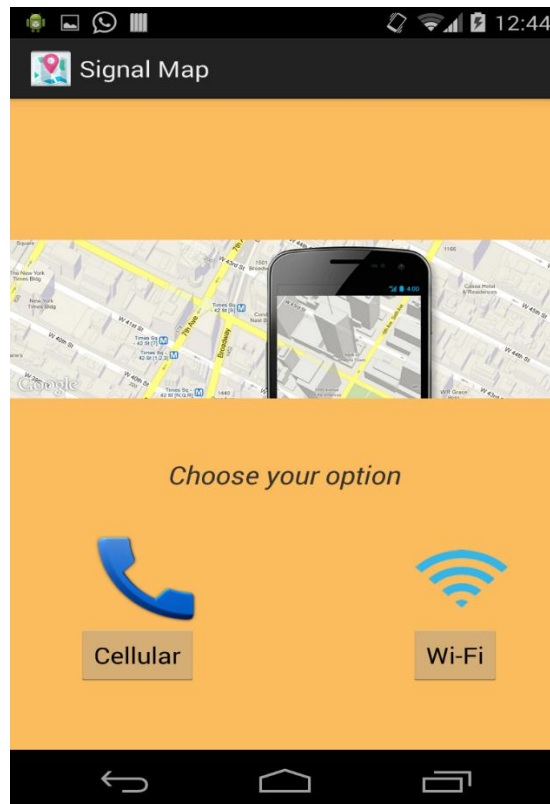


Fig. 6.1(c) Option Page

- **Cellular information Page**

This is the cellular information page that gives the UI of the google map. There are three buttons on this page.

The City button takes you to the current city's location and adds a marker to it.

With the Submit Data button you can drop a pin on your current location indicating the signal strength of your carrier. You can click on the marker to view the signal strength.

SS button is another button on this page that retrieves all the markers existing in the database on the map.

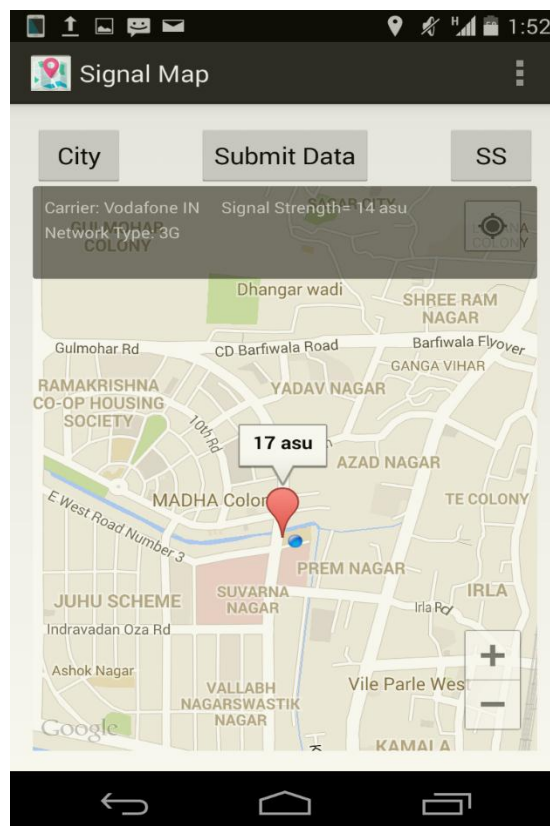


Fig. 6.1(d) Marker with Signal Strength

- **Submit Data Button Click**

When you click on the button submit data, an entry is created in the database and a marker is dropped at your current location displaying the signal strength of your carrier at that location.

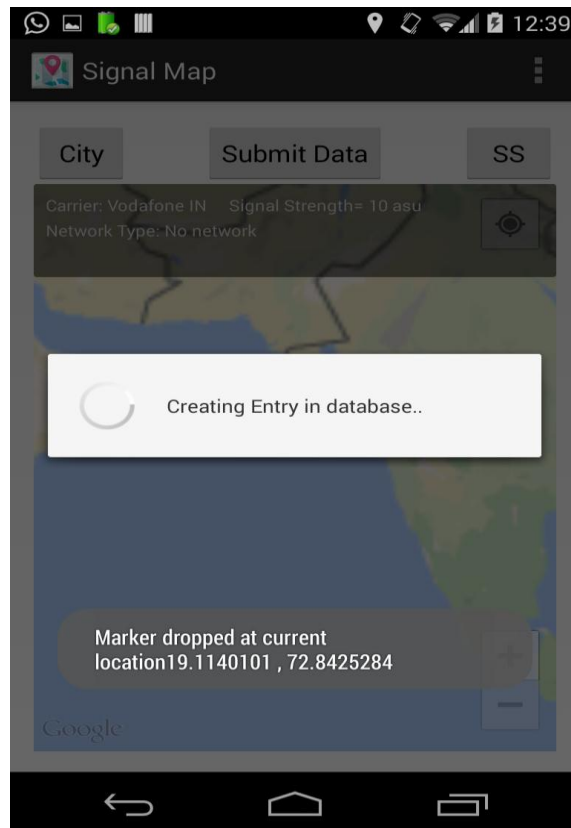


Fig. 6.1(e) Entry created in database

- **SS Button Click**

When you click on this button, all the entries in the database gets displayed on the map in the form of markers with varied signal strengths. The red marker indicates that the signal strength is low, blue marker indicates that the signal strength is average and the green marker indicates that the signal strength at that particular location is good.

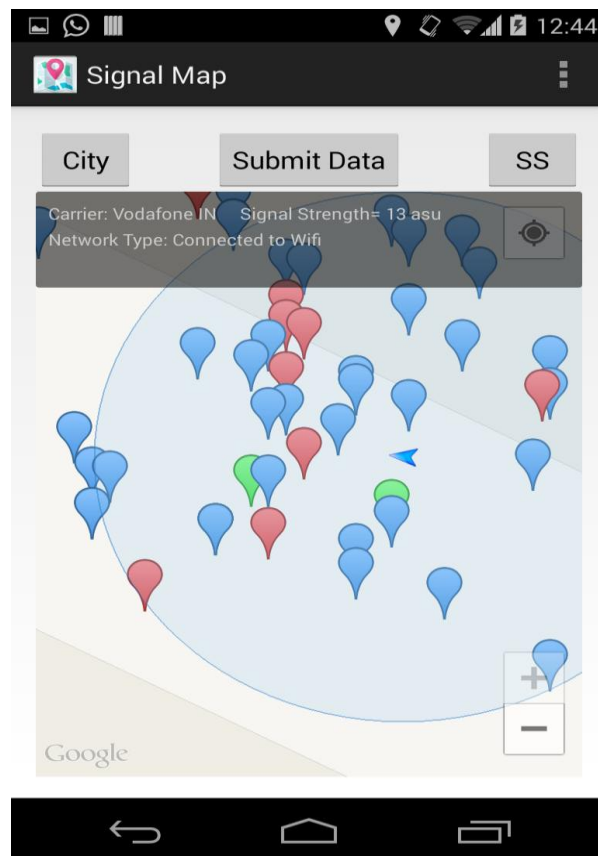


Fig. 6.1(f) All markers with signal strengths

- **Inflater Menu Options**

The inflater menu consists of options such as to select carrier or to view markers within a certain distance.

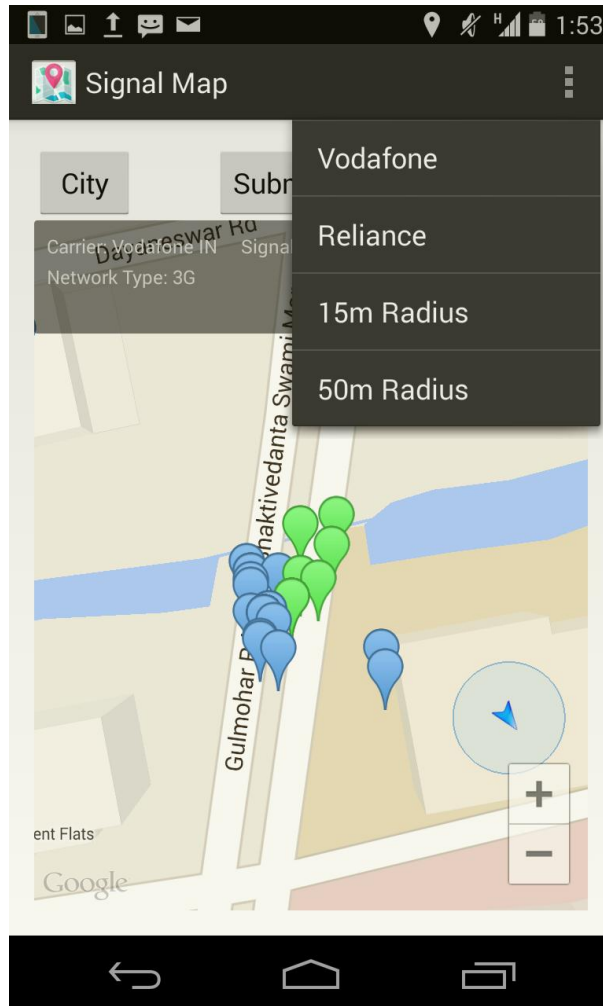


Fig. 6.1(g) Inflater Menu Options

- **Wifi Page**

This is the wifi information page that gives the UI of the google map. There are three buttons on this page.

The City button takes you to the current city's location and adds a marker to it.

With the Submit Data button you can drop a pin on your current location indicating the wifi strength of the wifi to which you are connected. You can click on the marker to view the wifi strength.

SS button is another button on this page that retrieves all the markers existing in the database on the map with varied wifi strengths.

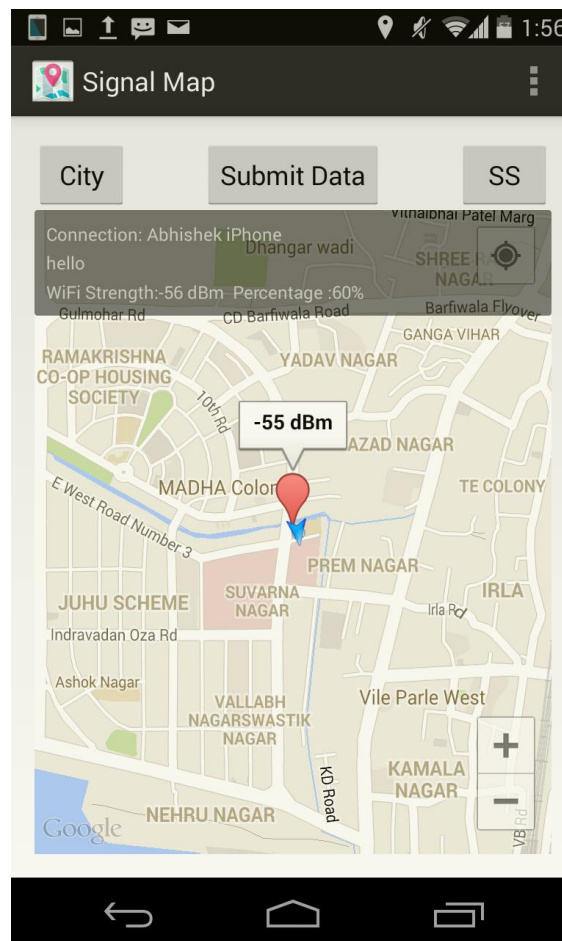


Fig. 6.1(h) Marker with Wifi Strength

- **Wifi Page SS button click**

When you click on this button, all the entries in the database gets displayed on the map in the form of markers with varied wifi strengths. The red marker indicates that the wifi strength is low, blue marker indicates that the wifi strength is average and the green marker indicates that the wifi strength at that particular location is good.

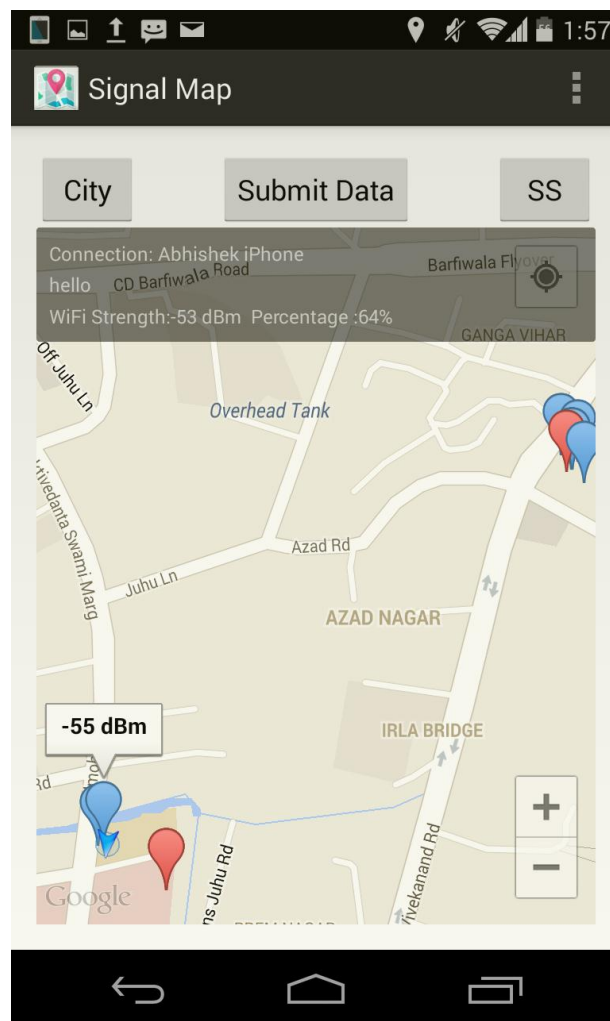


Fig 6.1(i) All markers with wifi strengths

# CONCLUSION AND FUTURE SCOPE



## **Chapter 7: Conclusion and Future Expansion**

### **Conclusion**

Android is the platform for most of the mobile devices in many countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day many people are buying new android phones and tablets and download games, applications for the device. Android provides platform to create games and applications for the Android phones and can distribute it through play store.

As Android is becoming more popular in the current era and has captured the major smartphone market demanding rapid growth in application development, and due to its open source platform it is easy for developers to develop new applications on this platform to satisfy current user needs. It has all the features that a developer requires to create the best applications. In the current era, most of the mobile companies are manufacturing smartphones and tablets on the Android platform which is used by hundreds of millions of users nowadays motivating the developers to create new applications on the Android platform.

Smartphones offer a lot of functionalities but we cannot deny that the primary reason for buying a phone is to communicate. We are working on developing an application named SIGNAL MAP which can be very helpful to the users who want to communicate on their Android smartphones without any network connectivity problems. The application helps in determining the locations with the best signal strengths on the Google map. So users can check the places with the best signal strengths and enjoy communicating without any network issues. We are also providing the functionality to users of locating the places with strong WiFi strengths. So users can browse easily and enjoy all services requiring Internet with great speed and connectivity.

## Future Expansion

As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment. Based on the future security issues, security can be improved using emerging technologies. It is not possible to develop a system that meets all the requirements of the user. User requirements keep changing as the system is being used. Many future enhancements can be done with the changing technologies and up gradations.

Many new features can be added to our application such as:

- Better User Interface
- Decreased memory usage of the app on the system memory.
- Increased customization for the user.
- Fluidic transitions.
- Notifications on the status bar
- Better multitasking features
- HeatMaps and Marker Clustering

## **Bibliography/References**

The different websites used for reference are as follows:

[1]. Android Developers

[www.developer.android.com](http://www.developer.android.com)

[2]. Android Application Development Tutorials (the new Boston)

<http://www.youtube.com/watch?v=SUOWNXGRc6g&list=PL3D7BFF1DDBDAAFE5>

[3]. Google Play

<https://play.google.com/store?hl=en>

[4]. Stack Overflow

[www.stackoverflow.com](http://www.stackoverflow.com)

[5]. Android – XDA Developers 2014

[www.xda-developers.com/](http://www.xda-developers.com/)

[6]. Android Forums

[www.androidforums.com](http://www.androidforums.com)

[7]. Slate Droid

[www.slatedroid.com](http://www.slatedroid.com)

[8]. Android Community Tutorial – Business articles, blogs, videos, apps and reviews

[www.androidpeople.com](http://www.androidpeople.com)

[9]. Android Apps – Android Software downloads and reviews

<http://download.cnet.com/android>

[10]. Android Development Tutorial

<http://www.android-app-market.com>

[11]. Mobile Phone Signal

[http://en.wikipedia.org/wiki/Mobile\\_phone\\_signal](http://en.wikipedia.org/wiki/Mobile_phone_signal)

[12]. Android Application Development

<http://www.kaibits-software.com/>

[13]. Signal Strength

<http://www.tomshardware.com/forum/58497-36-what-meaning-signal-strength-91dbm-6asu>

[14]. What is php.

<http://www.php.net/manual/en/intro-what-is.php>

[15]. Android SQLite Tutorial

<http://www.vogella.com/tutorials/AndroidSQLite/article.html>

[16]. how-to-connect-android-with-php-mysql

<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>

[17]. sqlite-vs-mysql

<http://stackoverflow.com/questions/3630/sqlite-vs-mysql>

[18]. UML Diagrams

<http://www.uml-diagrams.org/>

**The books used for reference are:**

[1]. Rick Rogers, John Lombardo, Zigurd Mednieks, and Blake Meike, Android Application Development, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472

[2]. Mark L. Murphy, The busy coder's guide to Android Development, USA:2008-2010 CommonsWare, LLC.