Chapter 1
INTRODUCTION

1.1General Introduction:

Location based service (LBS) is emerging as a killer application in mobile data services thanks to the rapid development in wireless communication and location positioning technologies. Users with location-aware wireless devices can query about their surroundings (e.g., requesting the information with respect to the location) at any place, anytime. While this ubiquitous computing paradigm brings great convenience for information access, the constraints of mobile environments, the spatial property of location-dependent data, and the mobility of mobile users pose a great challenge for the provision of location-based services to mobile users.

Over-view of location based Services:

A location-based service (LBS) is a mobile application that is dependent on the location of a mobile device, like mobile phone. Location Based Services can be defined as “Information services accessible with mobile devices through the access points and utilizing the ability to make use of the location of the mobile device”. Geospatial Consortium defined LBS service similarly:“A wireless-IP service that uses geographic information to serve a mobile user, any application service that exploits the position of a mobile terminal.”

A Location Based Service (LBS) is an information and entertainment service, accessible with mobile devices through the access points and utilizing the ability to make use of geographical position of the mobile device. A LBS services can be used in a variety of contexts, such as health, work, personal life, etc. LBS include services to identify the location of a person or object, such as discovering the nearest banking cash machine or the where about of a friend or employee.

LBS have two major actions, that is:

1. Obtaining the location of user
2. Utilizing this information to provide a service.

1.2 Statement of the Problem

With the increase in mobile computing devices and wireless LANs, it has become important to determine the location of a device at any point in time. Several applications can be conceived of that can use this information. Detecting the location is one of the first steps towards building context sensitive smart devices. For example, colleges can use such a technology to build location aware hand-held devices. As a visitor/student walks into a campus, holding such a device, information about the campus such as lecture halls, staff rooms flashes on the screen.

Global positioning systemsorpositioning provided by mobile telephony operators are suitable for outside environments where clear line-of-sight respect to the satellites or base stations is available. However, they suffer from multi-path effects within buildings, and therefore, in indoor they show poor performances. The use of Wi-Fi signals as a potential positioning system within buildings has opened doors for many applications is because of the ubiquitous availability of Wi-Fi signals in almost all the buildings. So no additional hardware is required to install a positioning system in the buildings.

1.3Objectives of the project

The main objective of the project is to create a system for the effective Ubiquitous location identification and information dissemination.

- **Capture position:** Getthe current position of the user.
- **Analyze the coordinates:** The system should be able to analyze the proper coordinate of the user and position it on the map.
- **Provides appropriate services:**The system provides appropriate services with respect to the coordinates of the user by accessing the server connected locally.

1.4 Current Scope

It can be used within a room as an indoor positioning system where the information is provided according to the position coordinates. This will be used to provide local services to the users with respect to his/her position which can be in the form of ads or any other information relative to his position.

1.5Future Scope

The system can be extended to be used in shopping malls and big firms as well as integrated with real time positioning to integrate both GPS and Wi-Fi positioning and hence providing services both indoors and outdoors and implemented in the field of health care, security and positioning.

It can be gathered that there is a huge scope of application development in mobile domain. Following the same notion, we can also develop application that can tackle following issues:

1) Location positioning technologies

2) Query processing

3) Cache management

Chapter 2
LITERATURE SURVEYAND ARCHITECTURE

LBS services can be categorized as triggered LBS services (push services) and user-requested LBS services (pull services). In a triggered (push) LBS service, the location of user's mobile device is retrieved when a condition set in advance is fulfilled. For example, a call to emergency centre canautomatically trigger a location request. Advertisement messages can be delivered to users who enter a specific area in a shopping mall, and warning messages can be delivered to users who are in the area where weather conditions will change (e.g. hurricane, rain). In a user-requested (pull) LBS service, the user decides whether and when to retrieve the location ofhis/her mobile device and use it in the service. User-requested LBS service can involve personal location (i.e. finding the current location of the user) or services location (i.e. finding the location of the nearest restaurant or bank). Navigation and direction system is an example of pull LBS services.

2.1 Development Environment

Android

Android is open source platform and provides all information and services to all without any license fees. It has capacity of getting compatible with almost all browsers hence you can make it compatible with your favorite browser. Google has made many features available to android as an open source platform and it is highly scalable. Google's Android mobile platform consists of Android operating system and Android SDK as well as android middleware. SDK is the basic platform which provides developers required tools and technical support for developing apps and APIs. The best thing about Android is, it is based on Linux platform hence it facilitates easy accessibility to environment and the core functionality for building fabulous applications for the smart mobile phones.

2.2 Android Architecture

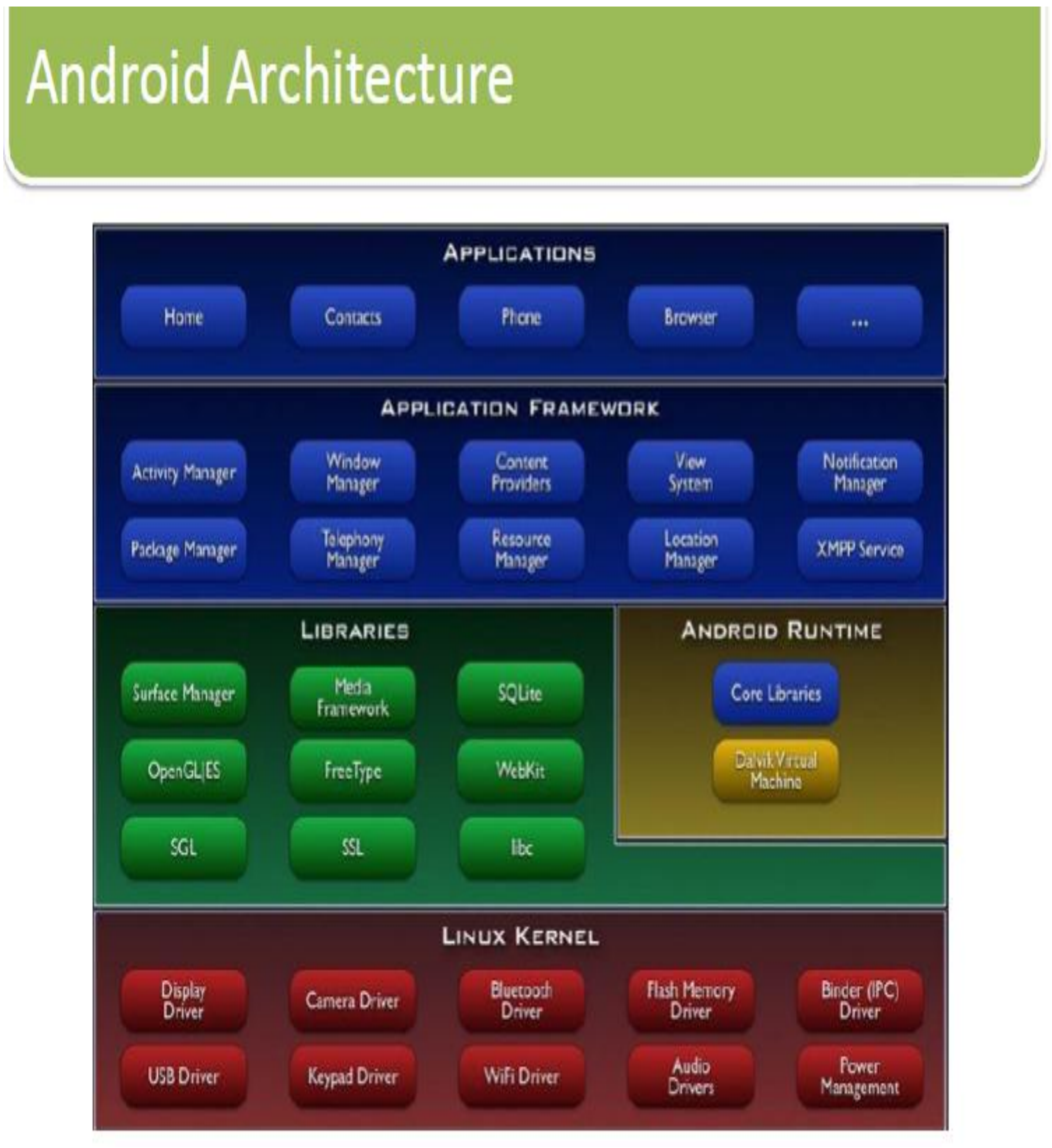
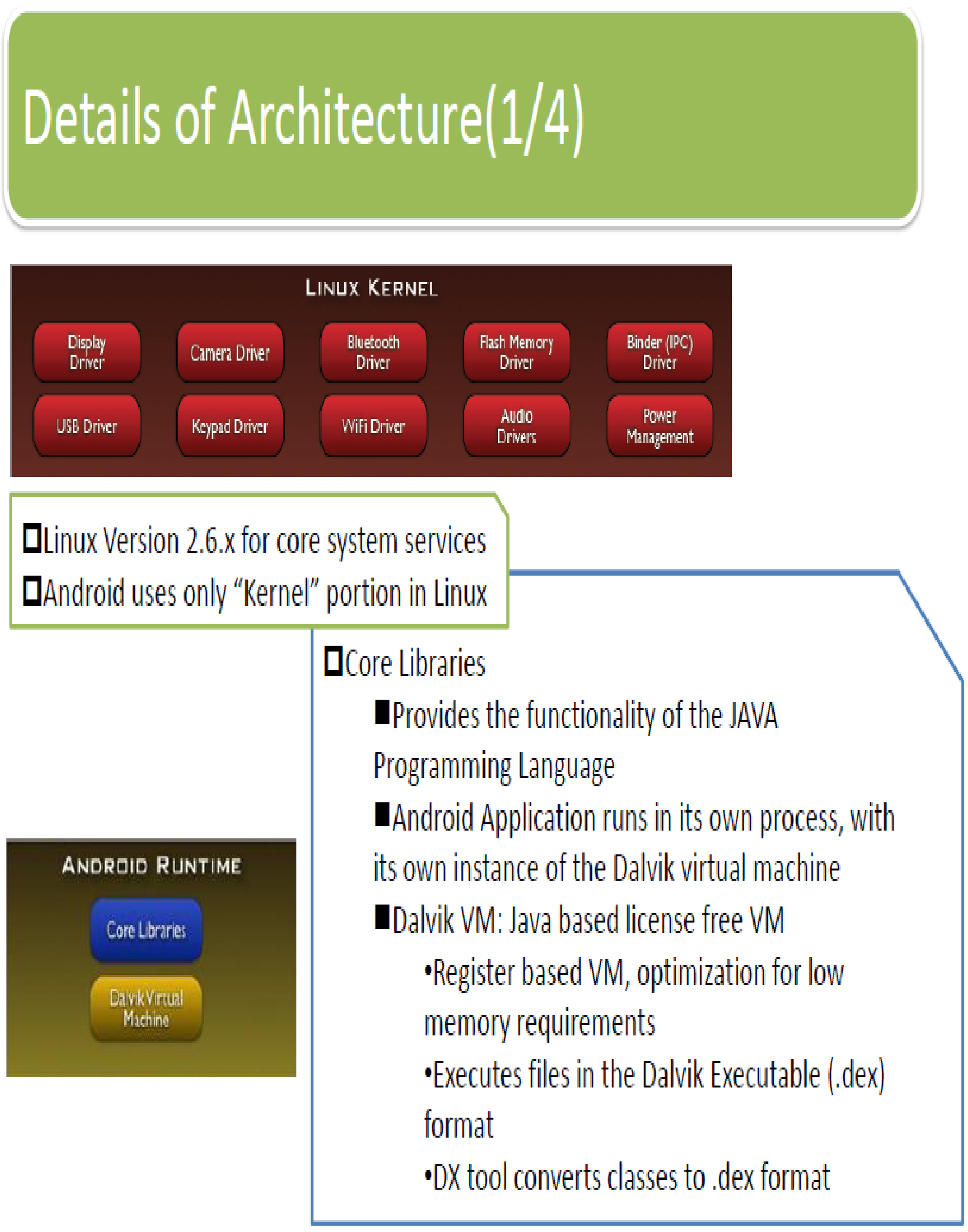


Figure 2.1: Application framework

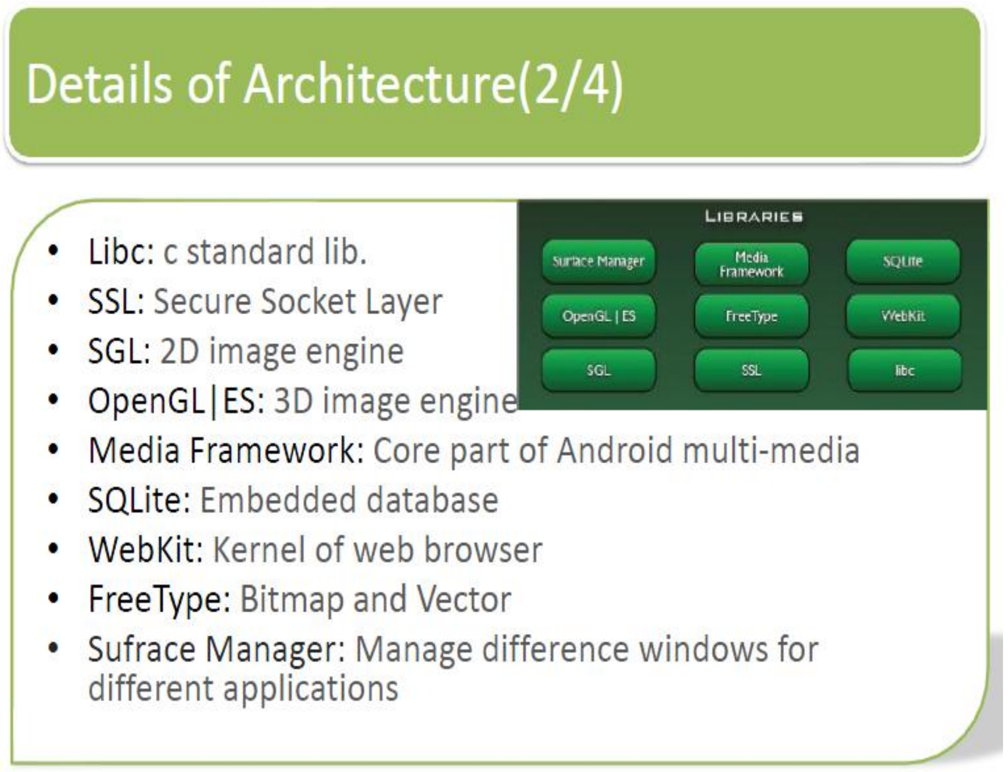

LINUX KERNEL
Display Driver
Camera Driver
Bluetooth Driver
Flash Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management
ANDROID RUNTIME
Core Libraries
Dalvik Virtual Machine

Details of Architecture(1/4)

- Linux Version 2.6.x for core system services
- Android uses only "Kernel" portion in Linux

- Core Libraries
  - Provides the functionality of the JAVA Programming Language
  - Android Application runs in its own process, with its own instance of the Dalvik virtual machine
  - Dalvik VM: Java based license free VM
    - •Register based VM, optimization for low memory requirements
    - •Executes files in the Dalvik Executable (.dex) format
    - •DX tool converts classes to .dex format

Figure 2.2: Linux kernel

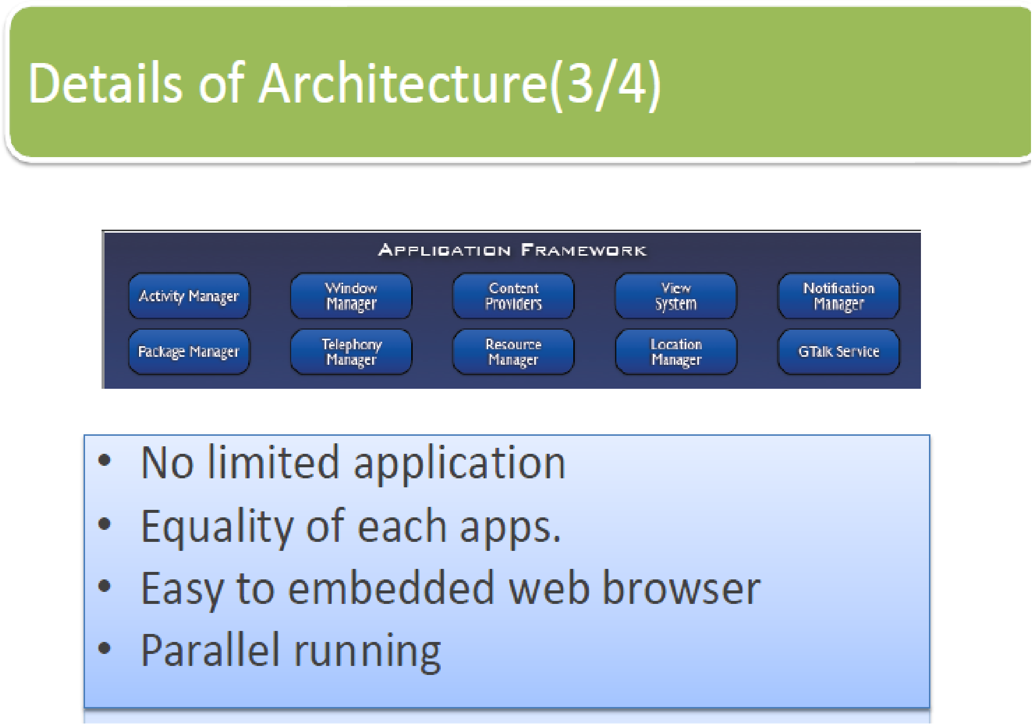


Details of Architecture(2/4)

- Libc: c standard lib.
- SSL: Secure Socket Layer
- SGL: 2D image engine
- OpenGL|ES: 3D image engine
- Media Framework: Core part of Android multi-media
- SQLite: Embedded database
- WebKit: Kernel of web browser
- FreeType: Bitmap and Vector
- Sufrace Manager: Manage difference windows for different applications

LIBRARIES
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc

Figure 2.3: Libraries

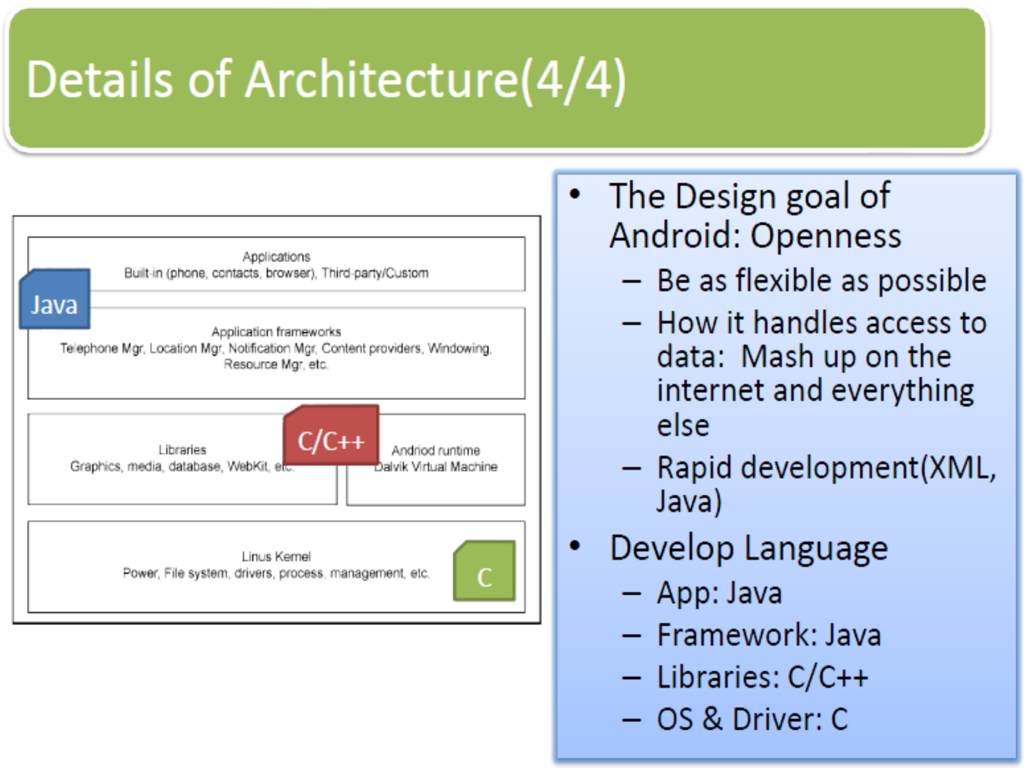


Details of Architecture(3/4)

APPLICATION FRAMEWORK
Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
GTalk Service

- No limited application
- Equality of each apps.
- Easy to embedded web browser
- Parallel running

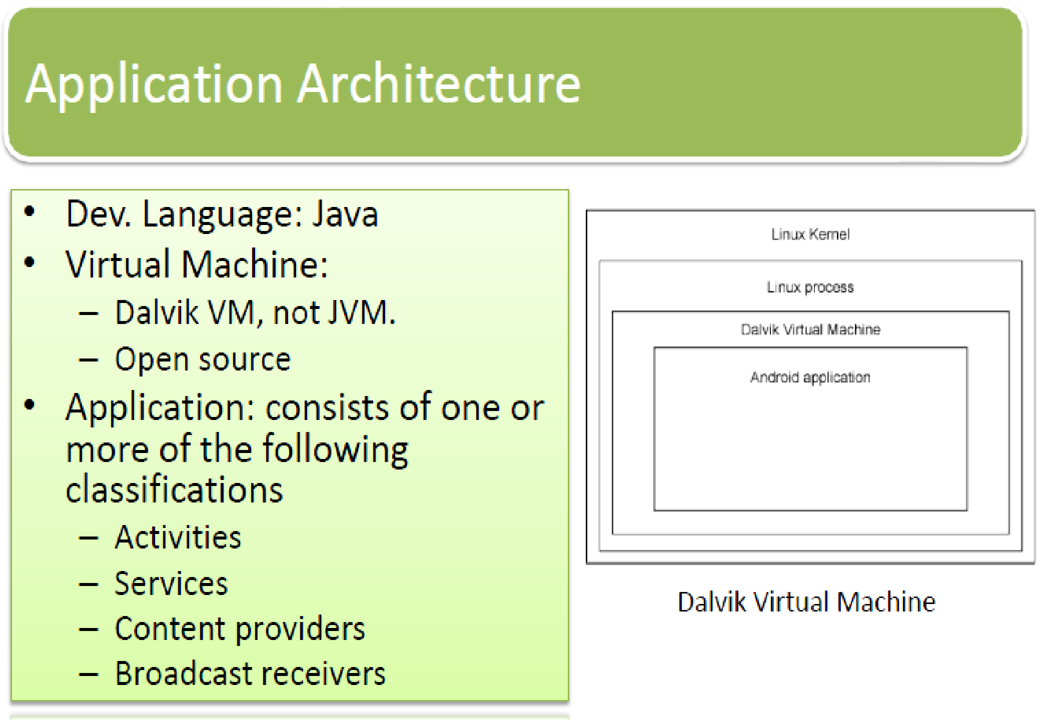
Figure 2.4: Application framework contd.


Applications
Built-in (phone, contacts, browser), Third-party/Custom
Java
Application frameworks
Telephone Mgr, Location Mgr, Notification Mgr, Content providers, Windowing, Resource Mgr, etc.
Libraries
Graphics, media, database, WebKit, etc.
C/C++
Andriod runtime
Dalvik Virtual Machine
Linus Kernel
Power, File system, drivers, process, management, etc.
C

Details of Architecture(4/4)

- The Design goal of Android: Openness
  - Be as flexible as possible
  - How it handles access to data: Mash up on the internet and everything else
  - Rapid development(XML, Java)
- Develop Language
  - App: Java
  - Framework: Java
  - Libraries: C/C++
  - OS & Driver: C

Figure 2.5: Development Platform



Application Architecture

- Dev. Language: Java
- Virtual Machine:
  - Dalvik VM, not JVM.
  - Open source
- Application: consists of one or more of the following classifications
  - Activities
  - Services
  - Content providers
  - Broadcast receivers

Figure 2.6: Application architecture

2.3 Existing Positioning Techniques:

Triangulation:

The triangulation location sensing technique uses the geometric properties of triangles tocompute object location, using distance measurements and angulation, using primarilyangle or bearing measurements

Lateration:

Lateration computes the position of an object by measuring its distance from multiple reference distance measurements from 3 non-collinear points as shown below. But in 3-D, distance measurements from 4 non-coplanar points are required.

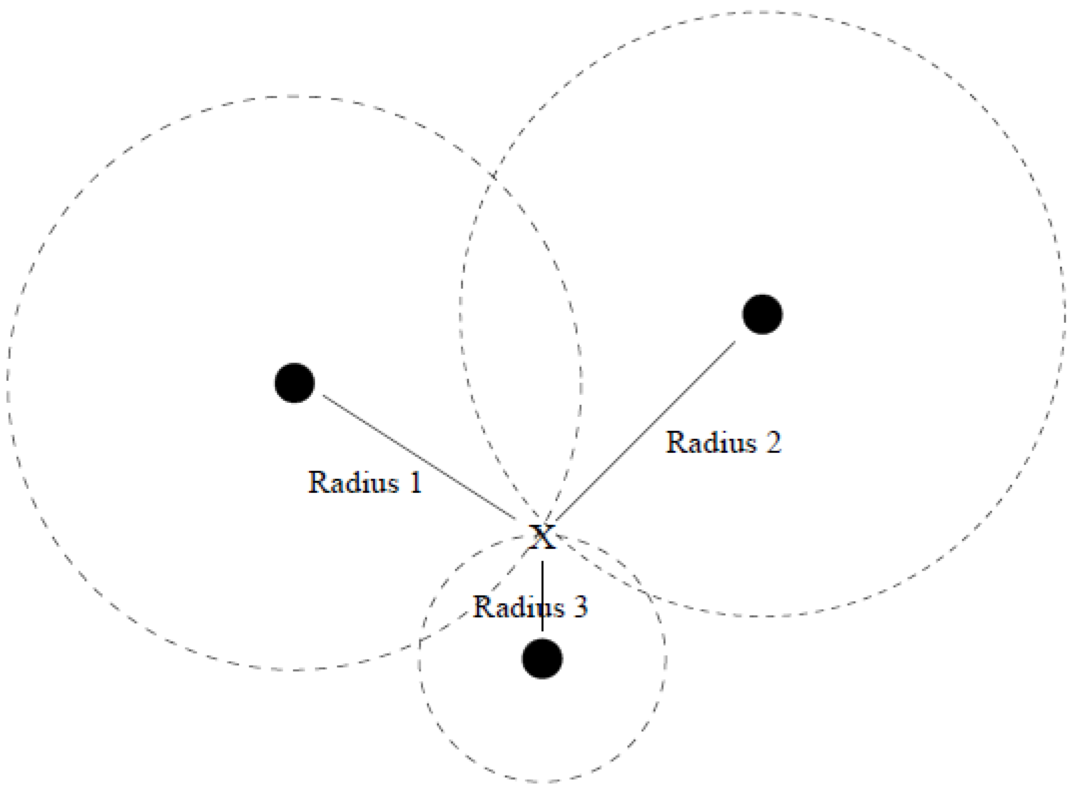


Figure 2.7: Computation of distance using lateration.

There are three general approaches to measuring the distances required by the laterationtechnique.

Direct: Direct measurements of distances use a physical action or movement. Direct distances measurements are simple to understand but difficult to obtain automatically due to the complexities involved in coordinating physical movement.

Time-of-Flight: Measuring distances from an object to some point P using time-of-flight means measuring the time it takes to travel between the object and point P at a known velocity. The object itself may be moving, the object is approximately stationary and we are instead observing the difference in transmission and arrival time of an emitted signal.

Attenuation: The intensity of an emitted signal decreases as the distance from the emission source increase. The decrease relative to the original intensity is the attenuation. Given a functioncorrelating attenuation and distance for a type of estimate and the original strength of the emission, it's possible to estimate the distance from an object to some point P by measuring the strength of the emission when it reaches P.

Angulation:

Angulation is similar to lateration expect, instead of distances, angles are used for determiningthe position of an object. 2-D angulation requires two anglemeasurementsand one lengthmeasurement such as the distances between the reference points as showninfig below.

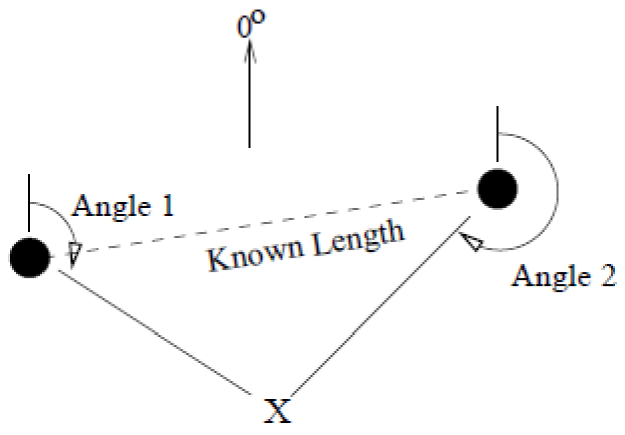


Figure 2.8: Angle measurement technique

Scene Analysis:

The scene analysis location sensing technique uses features of a scene observedfrom particular vantage point to draw conclusion about the location of the observer or of objects in thescene. Usually the observed scenes are simplified to obtain features that is are easy to represent and compared.

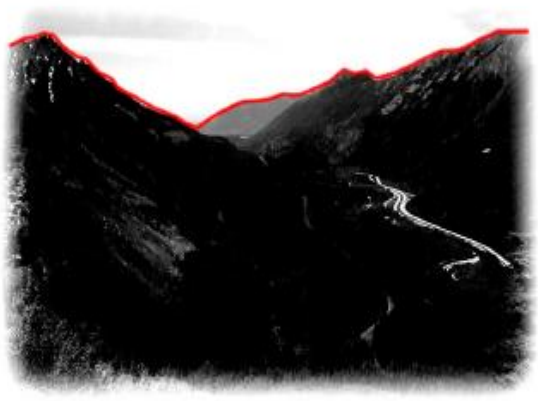


Figure 2.9: Scenery of a hill station

2.4 XAMPP:

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

Components:

- Apache 2.2.21
- MySQL 5.5.16
- PHP 5.3.8
- phpMyAdmin 3.4.5
- FileZilla FTP Server 0.9.39
- Tomcat 7.0.21 (with mod_proxy_ajp as connector)

PHP:

PHP is a general-purpose server-side scripting language originally designed for webdevelopment to produce dynamic web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document, rather than calling an external file to process data. Ultimately, the code is interpreted by a Web server with a PHP processor module which generates the resulting Web page.

Apache Server:

The Apache HTTP Server, commonly referred to as Apache is web server software notable for playing a key role in the initial growth of the World Wide Web. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Popular authentication modules include mod_access, mod_auth, mod_digest, and mod_auth_digest, the successor to mod_digest. A sample of other features include Secure Sockets Layer and Transport Layer Security support (mod_ssl), a proxy module (mod_proxy), a URL rewriter (also known as a rewrite engine, implemented under mod_rewrite), custom log files (mod_log_config), and filtering support (mod_include and mod_ext_filter).

2.5 SYSTEM ARCHITECTURE AND TECHNOLOGY:

Since mobile devices have a considerably inferior capacity of processing that a cluster of servers, Client/Server architecture with light clients and heavy.

Logical Architecture:

The Logical architecture describes the structure of the system in terms of software systems. As it can be observed infigure below, logical architecture consists of three main parts: the server, composed by a database and the logic of application, middleware is the connection between network services and applications and, finally, the clients of the system.

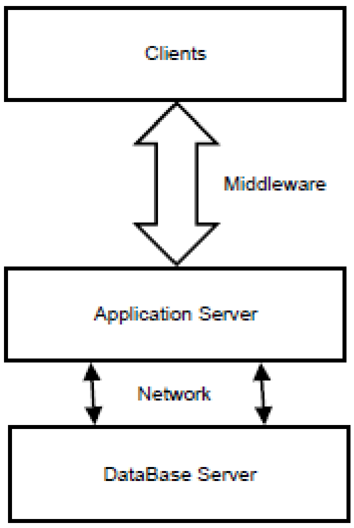


Figure 2.10: Logical architecture

Client: The client is a light weight android application which obtains the RSSI information from the Wi-Fi Access points at periodic intervals of time. The mean of the signal strength information is calculated for each of the access points and then sent to the server. We have developed the client to connect to the strongest access point available at any point of time.

Middleware: It is the part of the system where data from mobile devices (clients) is transferred to the server. In this case we are using a standard Http connection to communicate between the client and server.

.

Server: The server is compound of two main parts: The database storage and the logic of application.

– *DataBasestorage*: Where the server stores the mapping of signal strengths corresponding to the coordinates of the environment. We have used Apache as the operating environment for the server and PHP to connect to the server

– *Application logic*: Application logic is programmed using PHP which is the server side programming language.

2.6 Introduction to Front End Tool

2.6.1 What is android?

Android delivers a complete set of software for mobile devices: an operating system, middleware and key mobile applications.

- <u>android.jar:</u> Java archive file containing all of the Android SDK classes necessary to build your application.
- <u>documention.html</u>and docs directory The SDK documentation is provided locally and on the Web. It's largely in the form of JavaDocs, making it easy to navigate the many packages in the SDK. The documentation also includes a high-level Development Guide and links to the broader Android community.
- <u>Samples directory</u> the samples subdirectory contains full source code for a variety of applications, including ApiDemo, which exercises many APIs. The sample application is a great place to explore when starting Android application development.

- Tools directory contains all of the command-line tools to build Android applications. The most commonly employed and useful tool is the adbutility (Android Debug Bridge).
- UsbdriverDirectory containing the necessary drivers to connect the development environment to an Android-enabled device, such as the G1 or the Android Dev 1 unlocked development phone. These files are only required for developers using the Windows platform.

2.6.2 Tools

Emulator

Android applications may be run on a real device or on the Android Emulator, which ships with the Android SDK.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.

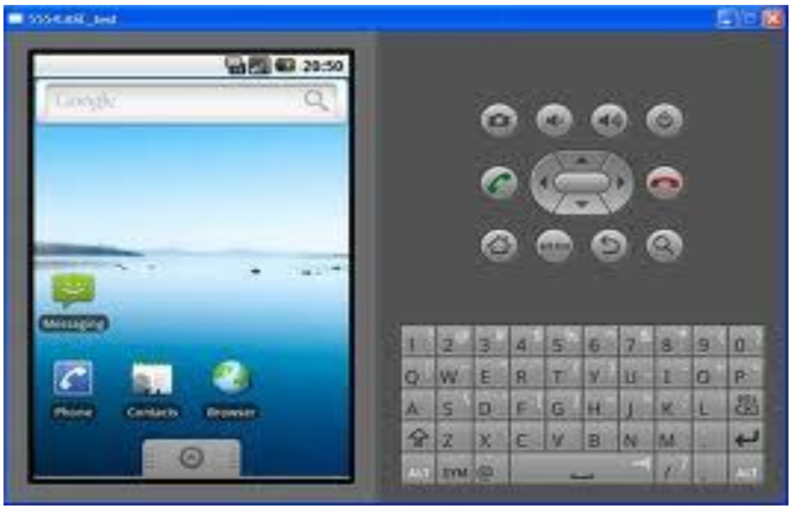
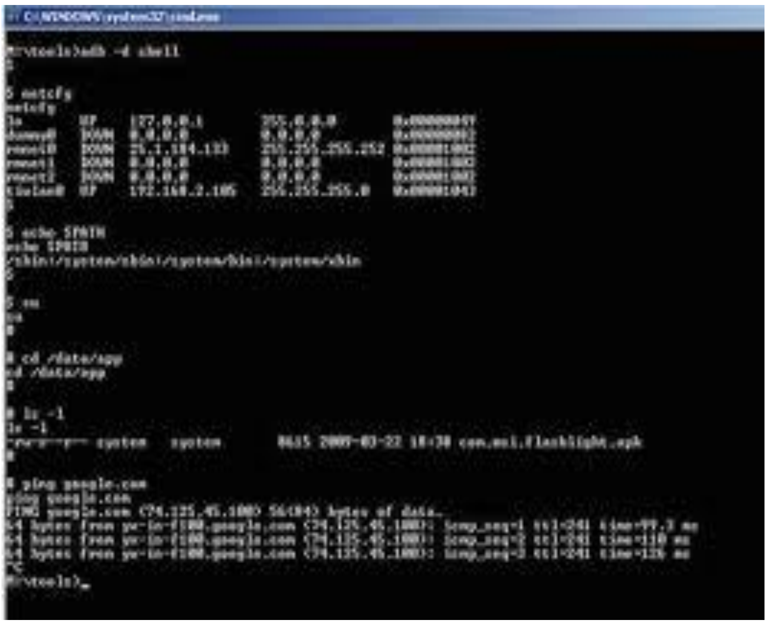


Figure 2.11: Android emulator

ADB (Android Debug Bridge)

The ADB utility lets you connect to the phone itself and issue rudimentary shell commands, such as copying files to and from the device.



```
C:\WINDOWS\system32\cmd.exe
[illegible]:\tools>adb -d shell
$
$ netcfg
netcfg
lo       UP     127.0.0.1       255.0.0.0       0x00000049
dummy0   DOWN   0.0.0.0         0.0.0.0         0x[illegible]
rmnet0   DOWN   25.1.184.133    255.255.255.252 0x[illegible]
rmnet1   DOWN   0.0.0.0         0.0.0.0         0x[illegible]
rmnet2   DOWN   0.0.0.0         0.0.0.0         0x[illegible]
tiwlan0  UP     192.168.2.105   255.255.255.0   0x[illegible]
$
$ echo $PATH
echo $PATH
/sbin:/system/sbin:/system/bin:/system/xbin
$
$ su
su
#
# cd /data/app
cd /data/app
#
# ls -l
ls -l
-rw-r--r-- system   system      8615 2009-03-22 18:38 com.[illegible].flashlight.apk
#
# ping google.com
ping google.com
PING google.com (74.125.45.100) 56(84) bytes of data.
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=1 ttl=241 time=99.3 ms
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=2 ttl=241 time=110 ms
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=3 ttl=241 time=126 ms
^C
[illegible]:\tools>_
```

Figure 2.12: Android Debug Bridge

Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It is a client-server program that includes three components:

- A client, which runs on your development machine. You can invoke a client from a shell by issuing an adb command. Other Android tools such as the ADT plugin and DDMS also create adb clients.
- A server, which runs as a background process on your development machine. The server manages communication between the client and the adb daemon running on an emulator or device.
- A daemon, which runs as a background process on each emulator or device instance.
- You can find the adb tool in <sdk>/platform-tools/.

When you start an adb client, the client first checks whether there is an adb server process already running. If there isn't, it starts the server process. When the server starts, it binds to local TCP port 5037 and listens for commands sent from adb clients—all adb clients use port 5037 to communicate with the adb server.

The server then sets up connections to all running emulator/device instances. It locates emulator/device instances by scanning odd-numbered ports in the range 5555 to 5585, the range used by emulators/devices. Where the server finds an adb daemon, it sets up a connection to that port. Note that each emulator/device instance acquires a pair of sequential ports — an even-numbered port for console connections and an odd-numbered port for adb connections. For example:

Emulator 1, console: 5554
Emulator 1, adb: 5555
Emulator 2, console: 5556
Emulator 2, adb: 5557

As shown, the emulator instance connected to adb on port 5555 is the same as the instance whose console listens on port 5554.

Once the server has set up connections to all emulator instances, you can use adb commands to control and access those instances. Because the server manages connections to emulator/device instances and handles commands from multiple adb clients, you can control any emulator/device instance from any client (or from a script).

The sections below describe the commands that you can use to access adb capabilities and manage the state of an emulator/device. Note that if you are developing Android applications in Eclipse and have installed the ADT plugin, you do not need to access adb from the command line. The ADT plugin provides a transparent integration of adb into the Eclipse IDE. However, you can still use adb directly as necessary, such as for debugging.

2.6.3 Development environment:

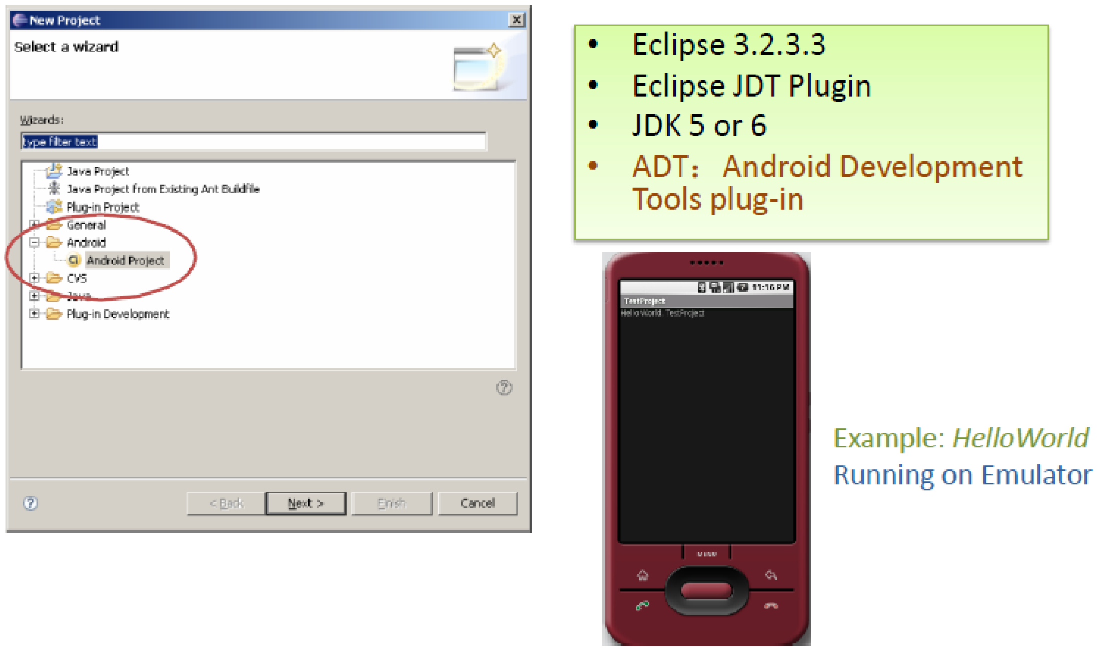


Figure 2.13: Android Development Platform

2.6.4 Summary

•Benefits

–Open Platform/License Free

–Robust OS Kernel, Innovative Library Packages

–Ease App. Development

–Rapid Improvement

•Challenges

–Performance Consideration

–Hard to Integrate for Vendors

–Too Much Google Dependent

•Key Factor: Market Response

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION:

3.1PURPOSE:

The purpose of this project is to provide a system for providing efficient timely location based service

3.2 Definitions, Acronyms:

Apache	A Server Domain
OS	Operating System
API	Application programming Interface
AP	Access Point
HTML	Hypertext Markup Language
PHP	Hypertext Pre Processor
CSS	Cascaded Style Sheets
RSS	Received Signal Strength
NNSS	Nearest Neighbor in Signal Space

3.3SPECIFIC REQUIREMENTS:

3.3.1 Functional Requirements:

Input: automated position of the user with respect to the access point.

Output: services based on the location of the user with respect to the access point.

3.3.2 Software Requirements:

Server Side Operating System	: Linux, Windows XP
Client side OS	: Android platform 2.2 and above
Server side Programming	: PHP,Apache,SQL
Client side Programming	: Android platform

3.3.3 Hardware Requirements:

Client:

- Android Mobile version 2.2 and above
- Wi-Fi and Internet Capability

Server:

- PC with server side program

3.3.4 End User:

- GUEST USERS: This category of users will able to get services with respect to the position of the user.

Chapter 4
SYSTEM DESIGN

4.1 Introduction

Our application is used to find the location of the user and disseminate the information using Android Smartphone. The Smartphone and the personal computer use various access points through which the signal strengths values are recorded. In our scenario we are placing three APs in different corners of the room with at least 15-20 meters distance apart from each AP. In our infrastructure APs are deployed in the network allowing users to access the WLAN. The WLAN-based positioning system uses the signal strength emitted from APs to estimate the location of a user. As the signal attenuation in an indoor environment is very dependent to the physical characteristic of the surrounding, the WLAN-based positioning system often requires the survey of signal strength in the environment, known as the off-line (data collection) phase before the operation. During this off-line phase, the RSS (received-signal-strength) statistics in the area are collected. The collected RSS and the related geographical information form a database to facilitate the tracking phase for the real-time positioning estimation. During the tracking phase, a user measures its RSS and estimates its current location based on its RSS and those collected RSS stored in the database. The WLAN-based positioning system usually employs the NNSS algorithm for the location estimation. In the following subsections, we briefly describe the off-line and tracking phase of a WLAN-based positioning system that uses NNSS algorithm.

4.2 System architecture and design

Although the WLAN-based positioning system has been investigated in the past research papers most of the studies implement the common Nearest Neighbor in Signal Space (NNSS) algorithm to achieve location estimation. While NNSS algorithm yields promising results, it does not account for the variation of the received signal strength

(RSS) from individual access points (AP) of a WLAN which may affect the accuracy of the estimation.

NNSS is the probabilistic approach which makes use of the signal probability distribution to calculate the probability that a user is at a certain location. However, the main disadvantage of the probabilistic approach is the requirement of a large training set, which implies costly data collection. The main motivation of outwork is to improve the accuracy of the NNSS algorithm while keeping the traditional data collection method.

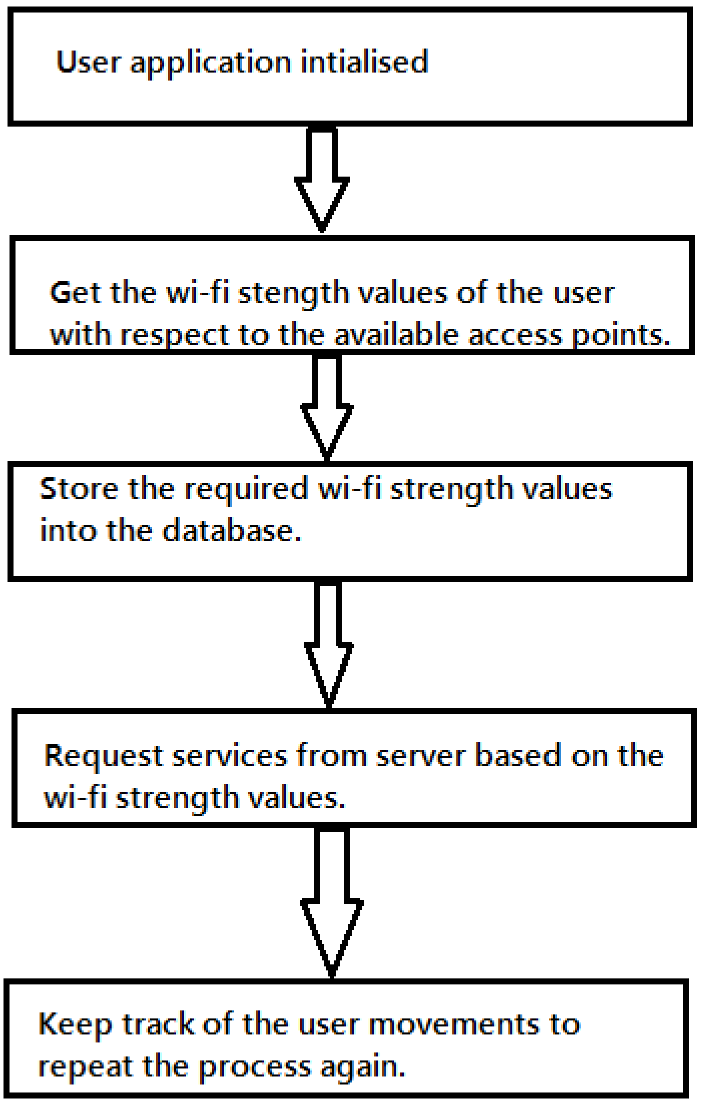

User application intialised
Get the wi-fi stength values of the user with respect to the available access points.
Store the required wi-fi strength values into the database.
Request services from server based on the wi-fi strength values.
Keep track of the user movements to repeat the process again.

Fig 4.1: Functional overview

4.2.1 Off-line Phase

During the off-line phase, a radio map is constructed by measuring the RSS at certain locations on the two dimensional floor plan. Commonly, a virtual rectangular grid is defined and measurements are taken at each of the grid intersections. Assuming that there are totally *N* APs visible throughout the floor plan, at intersection jth, one can obtain the RSS vector for j as Mj =< μj,1, μj,2, . . . , μj,N>, where μj,i is the RSS of APi. Each record in the database includes a mapping of the grid intersection's coordinates (xj, yj) and the RSS vector Mj . Each element in Mj is assumed to be the mean of the RSS from each of the N APs. This assumption can be achieved by collecting a large number of RSS samples for each orientation of the user.

4.2.2 Tracking Phase

During the tracking phase, the NNSS algorithm is used to determine the location of the MS. The RSS of all visible Aps are measured at the MS to produce a vector, denoted as *R* =<*r*1, *r*2, . . . ,*rN*>, where *ri*is the RSS of *APi*. By comparing *R* with *Mj*for each *j*, it is possible to identify a certain *j* such that the difference between *R* and *Mj*is the smallest compared to that of the other *j* values.

4.2.3 Location Determination System Architecture

Figure 4.2 shows our location determination system architecture. The hardware layer covers mobile devices, such as laptops and handhelds, and fixed devices that need location information. All these devices are equipped with wireless cards. The operating system layer includes the operating systems running on the devices. The device driver interacts with wireless cards to collect the signal strength values from the APs in the range. The location determination system layer runs on the location determination algorithm that uses the signal strength values to estimate the user location. A wireless API provides, in a device driver-independent way, the location determination system layer with a method to get the required information from the driver, such as the AP MAC addresses and the RSS. In the same way, a location API provides the user application with the device's current position in way independent of the location determination algorithm.

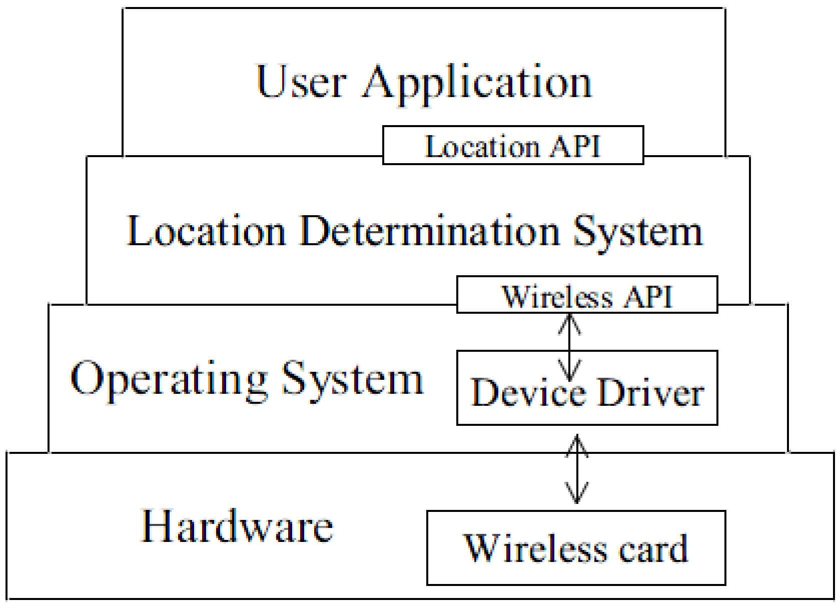


Figure 4.2: Location Determination System Architecture

4.3 Performance Analysis

Suppose that we have a WLAN-based positioning system deployed on a single floor plan inside a building. During the data collection phase, we define a virtual rectangle grid over the floor plan. At certain point, the RSS of visible are measured and their averages are recorded along with the physical position of the point. In the tracking phase, the RSS of each AP is measured at the user and the real-time RSS vector $R =< r1, r2, . . . ,rN>$is used to estimate the location of the user. Each component in R can be considered as a random variable with the following assumptions:

• The random variables ri(in dBm) are mutually independent, where $i= 1, 2, . . .,N$.

• The standard deviation of all the random variables ri is assumed to be identical throughout the area and denoted by σ (in dBm).

The assumption that random variables ri are independent is reasonable since there is no clear relationship between the signals transmitted by different APs. We shall now examine the performance of the enhancement algorithm when the NNSS algorithm return correct and incorrect physically nearest neighbours.

For performance evaluation purposes, we simulate an indoor positioning system as shown in Figure 4.3

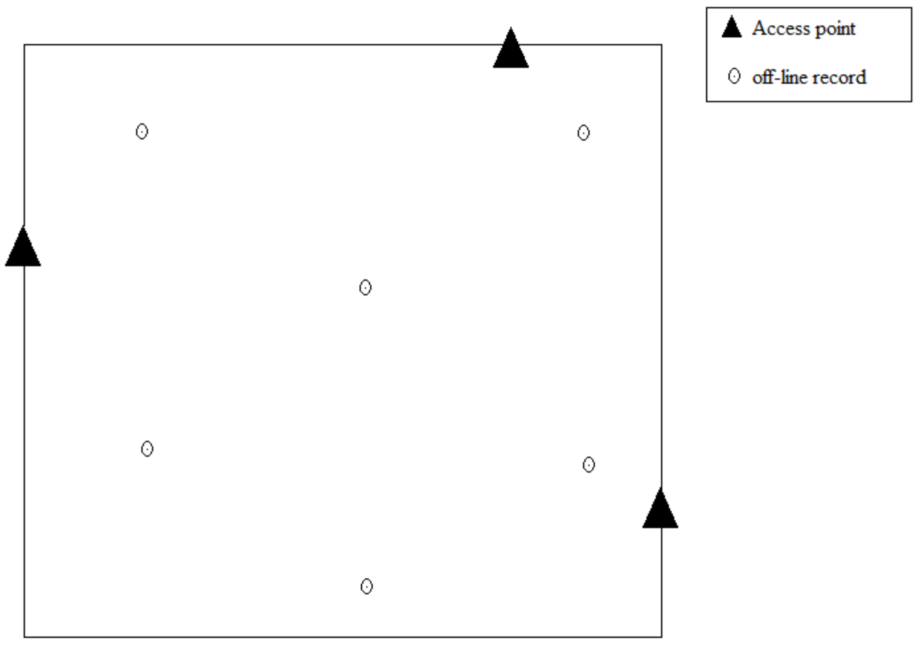


Figure 4.3: A Sample Scenario for Indoor Positioning System

In the above scenario we have placed three APs along the corners of the indoor building or a room. There are six testing points in this building and in addition, during the off-line phase (real-time phase), the user indicates his/her current location by using the API provided to them. During our experiments, we discovered that signal strength at a given location varies quite significantly (by up to 5 dBm) depending on the user’s orientation, i.e., the direction he/she is facing. In one orientation, the mobile host’s antenna may have line-of-sight (LoS) connectivity to abase station’s antenna while in the opposite orientation; the user’s body may form an obstruction. So, in addition to user's location, we also recorded the direction (*d*) (one of north, south, east, or west) that he/she is facing at the time the measurement is made. We discuss the implications of the user's orientation and variations of the signal strengths of different APs in the following figure.

In all, during the off-line phase, we collected signal strength information in each of the 4 directions at 7 distinct physical locations on our floor. For each combination of location and orientation we collected at least 20 signal strength samples.

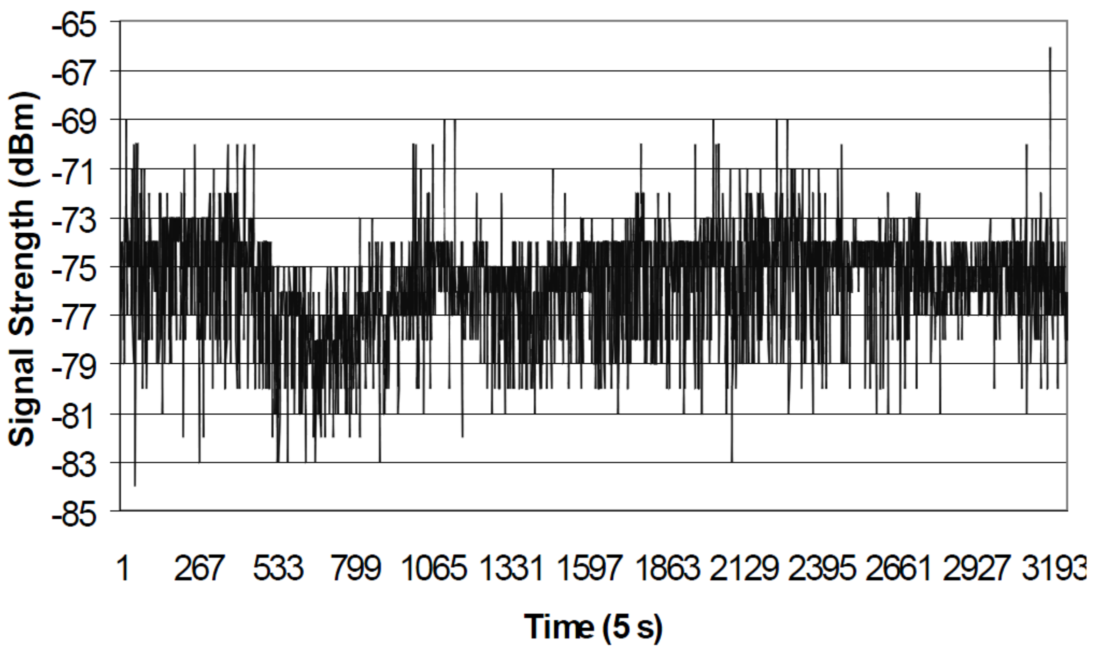


Figure 4.4: Signal strength measurements from a stationary position over time

The first objective was to determine whether signal strength was consistent for a specific location over time. If this was not the case, the reported location measurements would be inconsistent at different time. One way to overcome this obstacle would be to set up system similar to differential GPS in which a stationary station provides correction data to handheld units. Samples were taken over periods of two hours, 3 days a week and for two months at five - second intervals. Figure 4.4show that the data is fairly consistent with a standard deviation of few dBm (decibel mill watt). It was also determined that environmental condition such as opening an office door can create a change of up to 10dBm (this was expected). What was not expected was the effect other handheld devices had on signal strength, having an effect of up to 5 dBm.

Chapter 5
DETAIL DESIGN

The detailed design phase allows us to do some preliminary testing of the components that were initially invested from the preliminary design phase and make adjustments based on the results. In addition, the details about how each component is to operate and how they will function together are more heavily addressed. In this phase, the prototype schematics and parts are listed and presented as a theoretical solution to the project.

5.1 Class Diagram

In software engineering, a **Class Diagram** in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.
Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects.Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. This example is only meant as an introduction to the UML and class diagrams. If you would like to learn more see the Resources page for more detailed resources on UML.
Class diagrams also display relationships such as containment, inheritance, associations and others.

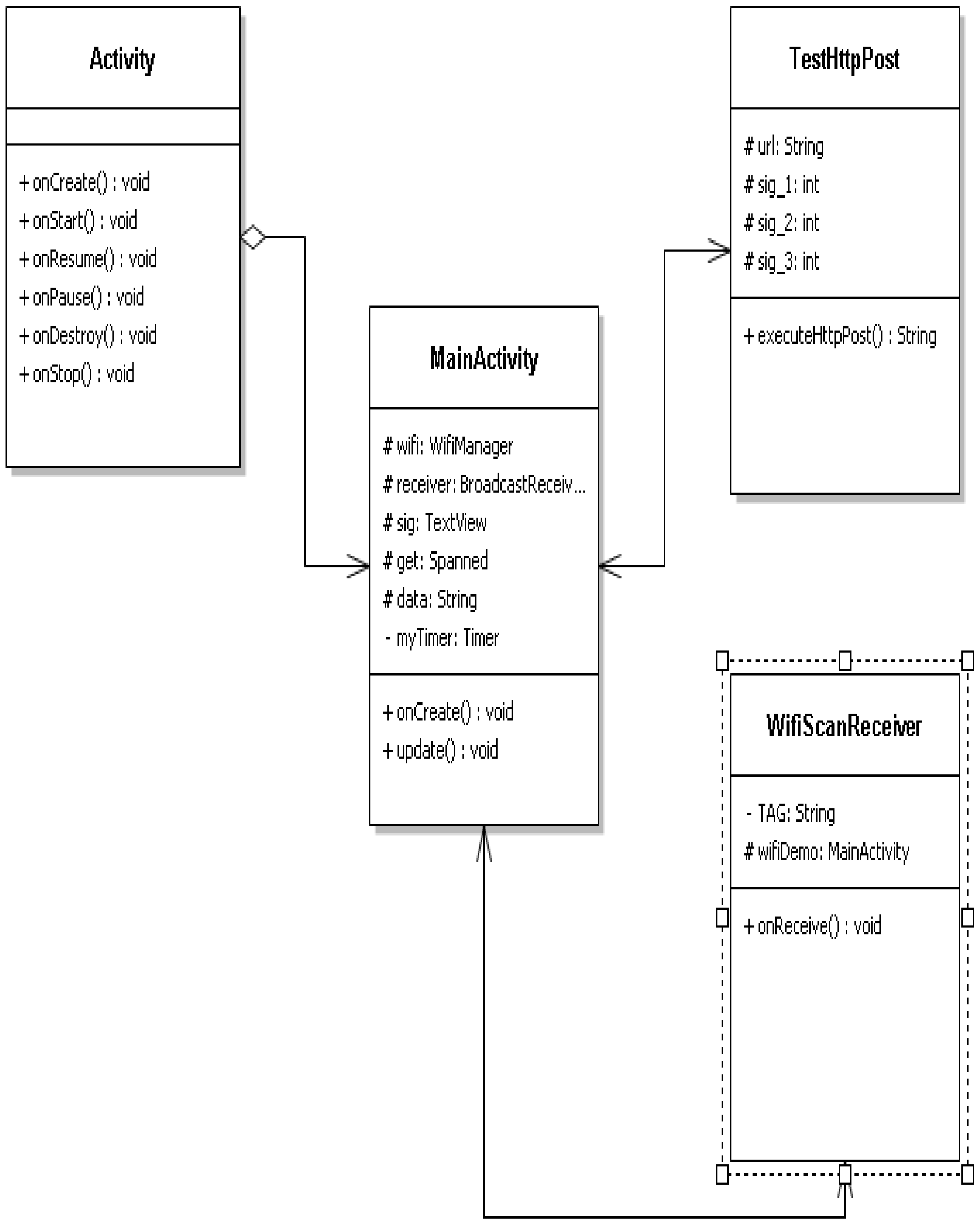


Figure 5.1: Class Diagram

5.2 Overall Block Diagram

The below diagram show the complete model of the application which has two blocks namely client and server. The client consists of three modules which involves Wi-Fi receiver to get the signal strength information, positioning system to request for user location information, and the last module shows the required information about the user with respect to their location.

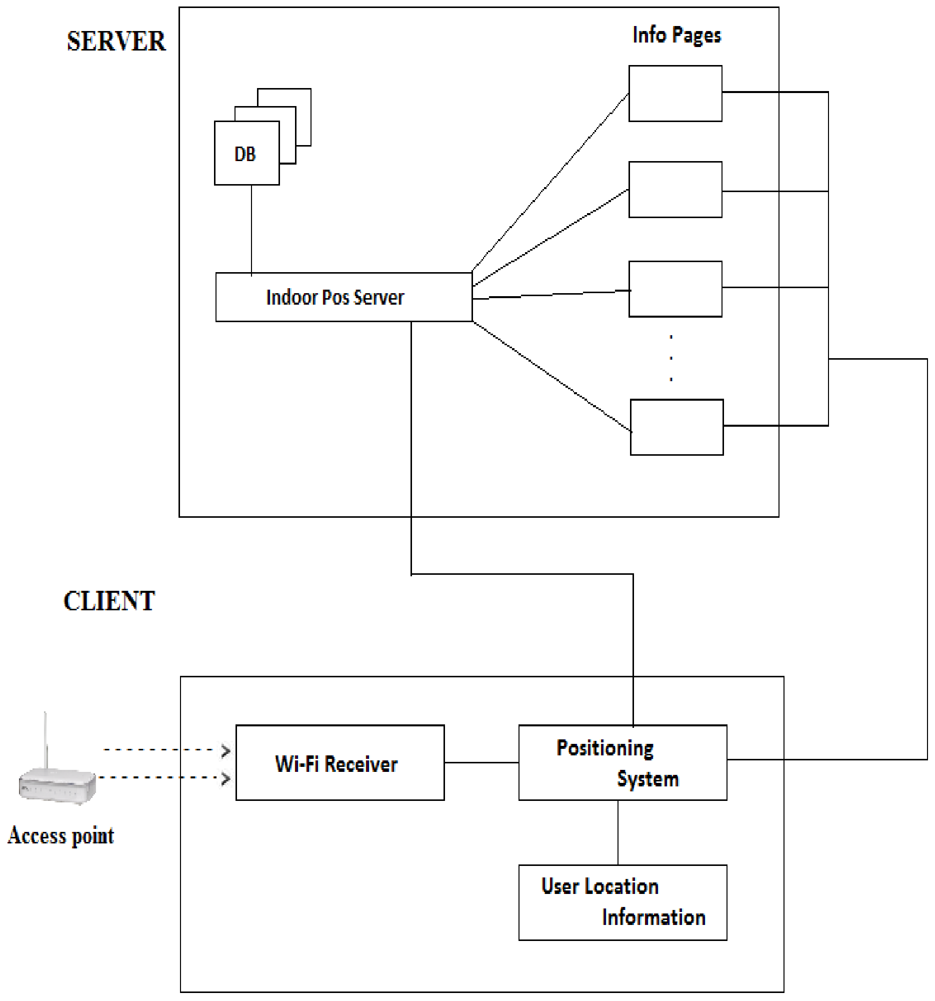


Figure 5.2: Block Diagram

5.3 Sequence diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams.

A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

As from the sequence diagram, the application first requires for the user to authenticate itself and connect to an access point. Connection is established once authentication is done. The application scans for signal strength information from the Access Points and sends the mean of the strength after a considerable repetitions of signal strength readings are taken.

The raw information is sent to the MiddleWare, location based server (LBS). The middleware establishes connection with the Database and maps the signal strength information to that in the servers using neighbour classifier method giving link to the corresponding useful data(services) to be provided to the user at that location.
This process is repeated and an update in position is notified in periodic intervals giving position information.

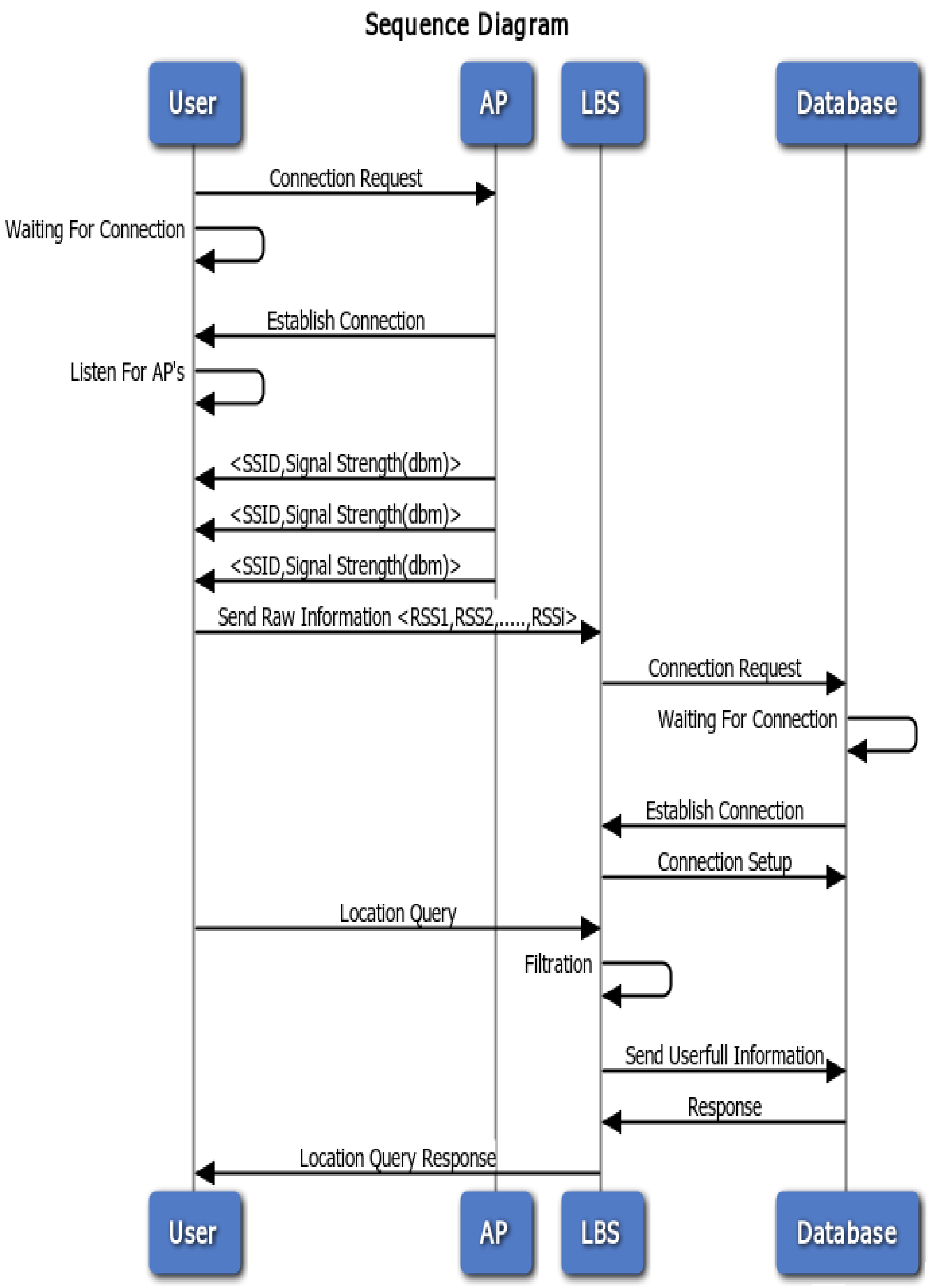


Figure 5.3: Sequence Diagram

5.4 Data Collection and Analysis:

We outline the data processing that we performed for analysis here.

5.4.1 Signal strength information:

As mentioned, the data analysis phase consists of building database of signal strength information. We setup our experimental environment for data analysis of signal strength information in one of the labs with the permission of staff members. Next figure shows the view of the setup of the lab.

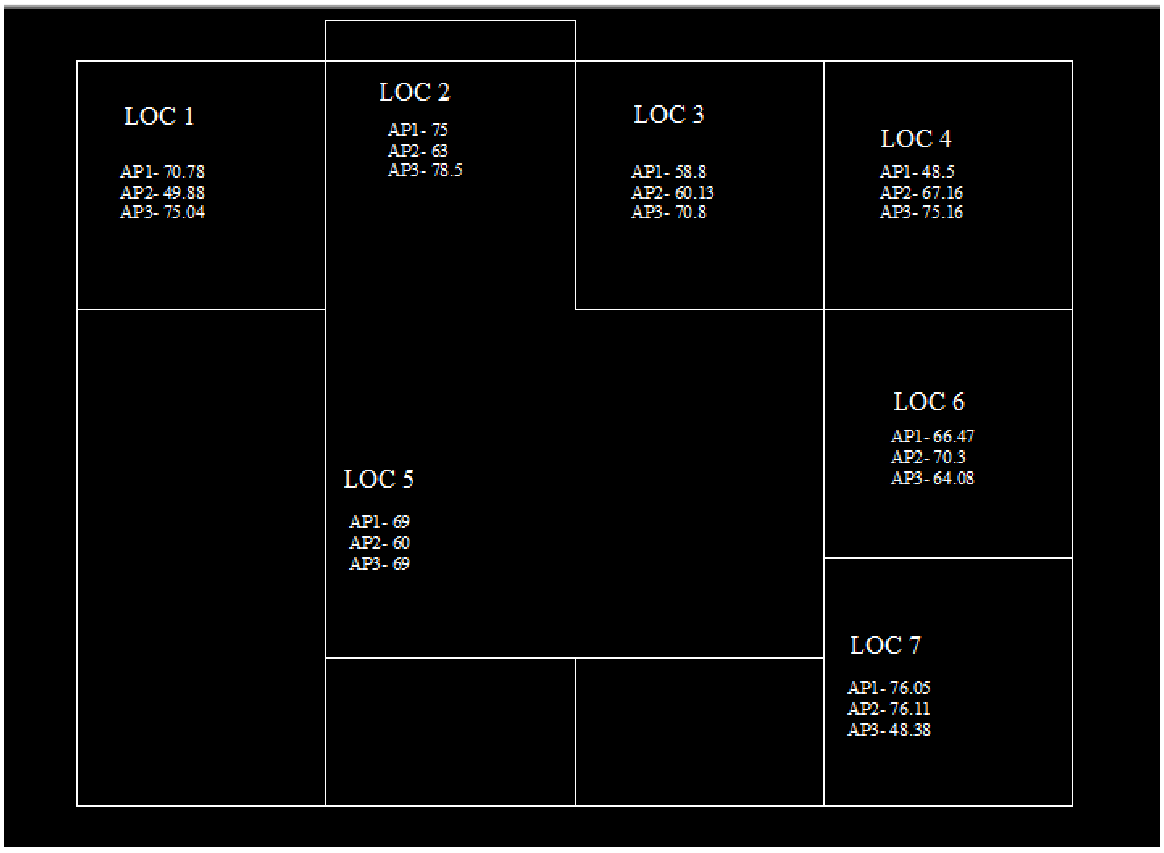


Figure 5.4: 2-D plan of the indoor building

We had placed 3 routers (Access Points) namely AP1, AP2, AP3 in three corners of the lab. We classified the lab with 7 training points as shown in the fig above. Each point defines a class of signal strength information which we define as training points. We made a routine in android client to take the signal strength information from each of these training points. The data collection phase proceeded for a period of 4 days, twice each day. We took the signal strength information at each training point with different orientation of mobile device. The data collection process for each point was repeated twice a day for next 4 days so that we could point the mean signal strength information at any particular point in a day today surroundings and in turn we took the mean of all the means of the signal strength for each of the access points from each training point.

5.4.2 Analysis:

Each of the values of the signal strength at a training point denotes a point in 3d signal space. The data collected information is attached at the end of this report for reference in the appendix. We have in turn calculated the variance and standard deviation for the values of each of the reading phase and calculated the mean of the standard deviation for these training points. The standard deviation value helps to verify whether the point belongs to the class which it is closest to. Thus for a test sample(si1,si2,si3) to belong to a training point it should lie within the parallelepiped formed by the standard deviation values as shown below in the figure.

Graphs:

Below is a Line graph of the of the Data Collected from the training points which we analyzed for implementing our Algorithm.

Reading1:

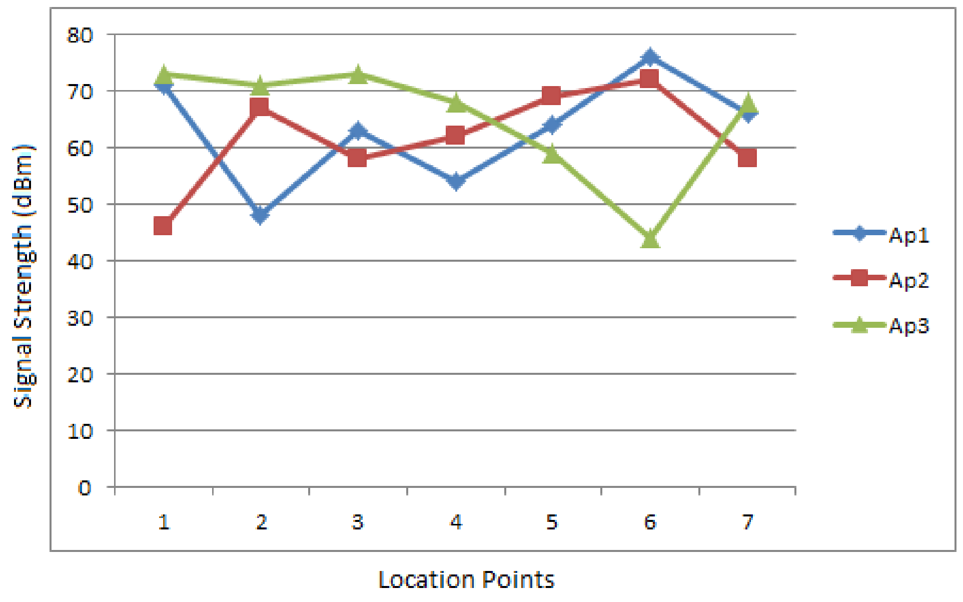


Figure 5.5: Sample Reading 1

Reading2:

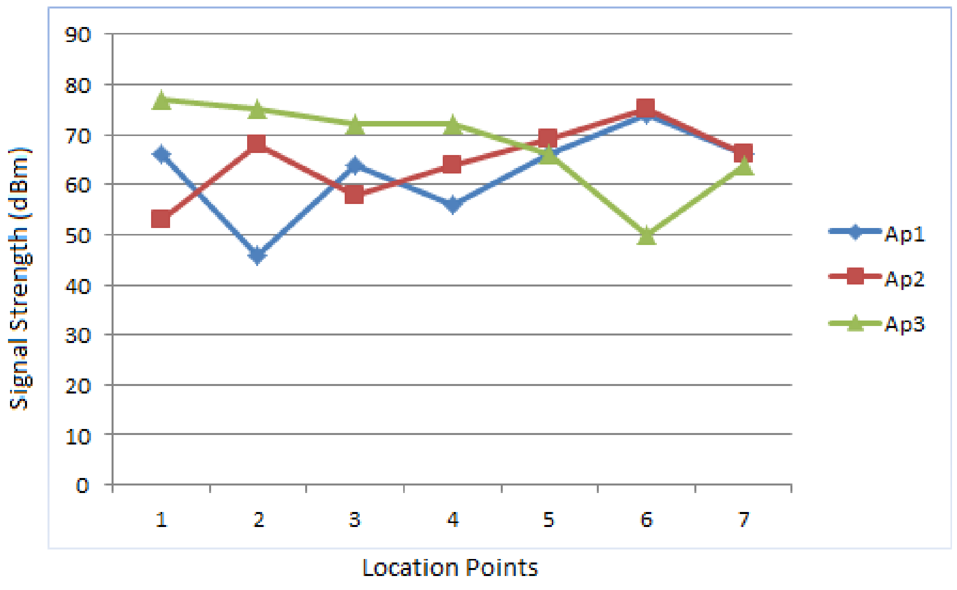


Figure 5.6: Sample Reading 2

Reading3:

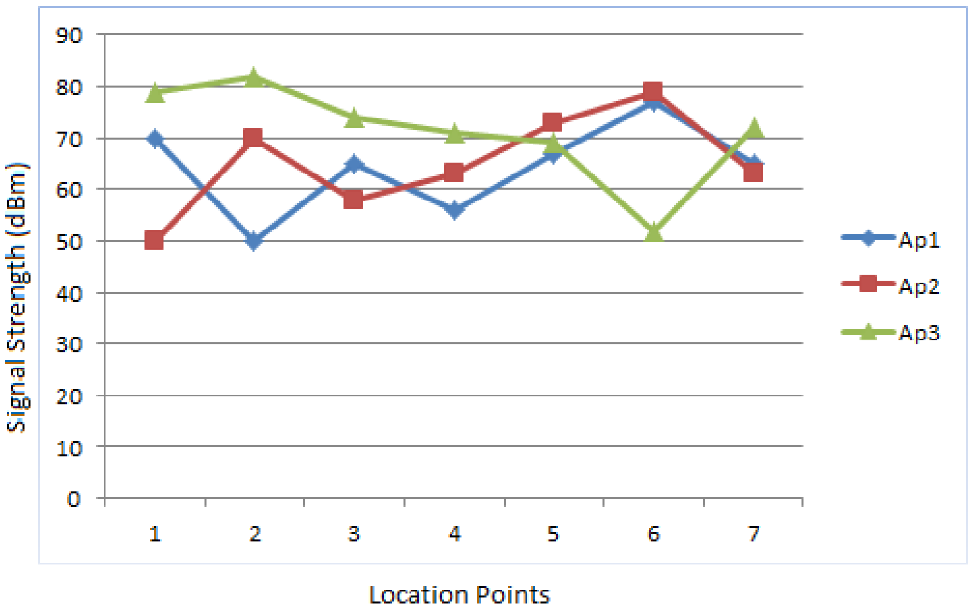


Figure 5.7: Sample Reading 3

Reading4:

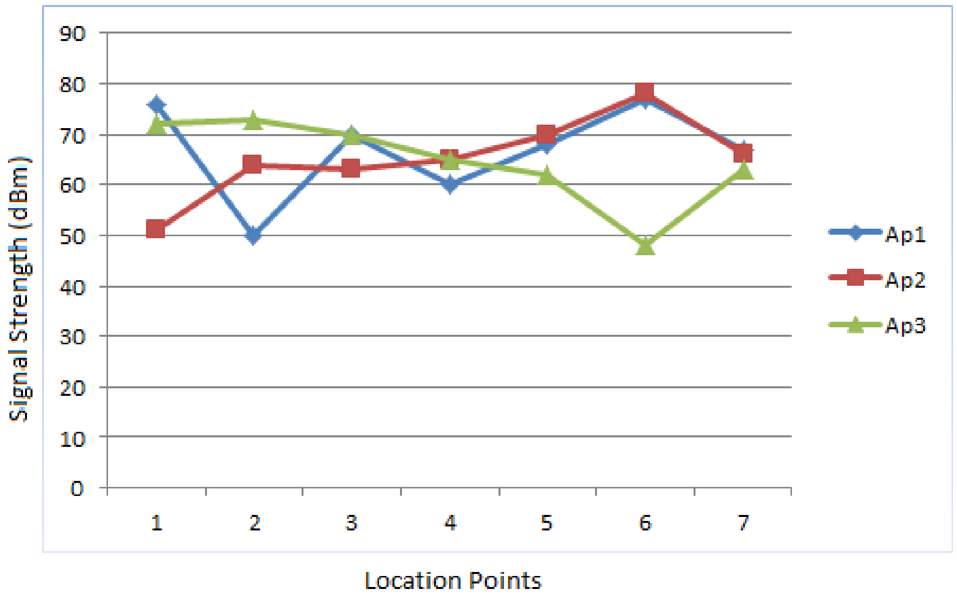


Figure 5.8: Sample Reading 4

As we can observe from the graph analysis above. For a particular training point the mean of the signal strength values for each of the Access Points Ap1, Ap2 and Ap3 during the 1st, 2nd, 3rd and 4th reading respectively were very close by. Thus we could implement this information to detect the position of the user.

5.4 Algorithm:

We convert the physical space to signal space by taking the signal strength value at that point. To detect whether a test sample belongs to a class we find the Nearest Neighbor in Signal Space (NNSS) using Euclidean distance.

$$\text{Euclidian Distance} = \sqrt{(ss1 - ss'1)^2 + (ss2 - ss'2)^2 + (ss3 - ss'3)^2}$$

Where ss1=Signal Strength at AP1
ss2=Signal Strength at AP2
ss3=Signal Strength at AP3
ss’1=Mean Signal Strength of AP1 for a test sample t
ss’2=Mean Signal Strength of AP2 for a test sample t
ss’3=Mean Signal Strength of AP3 for a test sample t

By applying the distance formula we find the closest training point we then apply the standard deviation value over it to check if the point is close enough to the point since sometimes the mobile user may be out of bounds from the region but still closer to one of the access points.
Thus a test sample t belongs to training point S if
(ss1-var)<ss’1<(ss1+var) where var=variance for AP1
(ss1-var)<ss’2<(ss1+var) where var=variance for AP2
(ss1-var)<ss’3<(ss1+var) where var=variance for AP3

Chapter 6

IMPLEMENTATION

CLIENT SIDE PROGRAMING

WifiScanReciever:

WifiScanReciever class extends BroadcastReceiver which initialize the application and receives the signal strength information.

BroadcastReceiver:

A broadcast receiver is a class which extends BroadcastReceiver and which is registered as a receiver in an Android Application via the AndroidManifest.xml file(or via code). Alternatively to this static registration, you can also register a BroadcastReceiver dynamically via theContext.registerReceiver() method.This class will be able to receive intents. Intents can be generated via the Context.sendBroadcast()method.The class BroadcastReceiver defines the onReceive() method. Only during this method yourBroadcastReceiver object will be valid, afterwards the Android system can recycle theBroadcastReceiver. Therefore you cannot perform any asynchronous operation in the onReceive()method.

```
publicclassWiFiScanReceiverextendsBroadcastReceiver
{
        publicWiFiScanReceiver(Main2Activity wifiDemo)
        {
                super();
        }
        .
        .
        .
        publicvoidonReceive(Context c, Intent intent)
        {
                //getting the signal strength
                ……………………………
        }
}
```

TestHttpPost

In this class the received signal strength is sent to the server and required information is requested.

HttpClient:

Android contains the Apache HttpClient library. We can either use the DefaultHttpClient orAndroidHttpClient to setup the HTTP client.DefaultHttpClient is the standard HttpClient and uses the SingleClientConnManager class to handle HTTP connections. SingleClientConnManager is not thread-safe; this means that access to it via several threads will create problems.

```
publicclassTestHttpPost
{
        publicTestHttpPost(int sig1,int sig2,int sig3)
        {
                …………….
        }
        publicStringexecuteHttpPost() throws Exception
        {
                //sending http request
                • ……..
        }
}
```

Main2Activity

Activity

Activity represents the presentation layer of an Android application. A simplified description is that an Activity represents a screen in your Android application. This is slightly incorrect as Activities can be displayed as Dialogs or can be transparent. An Android application can have several Activities.

Intents

Intents are asynchronous messages which allow the application to request functionality from other components of the Android system, e.g. from Services or Activities. An application can call a component directly (explicit Intent) or ask the Android system to evaluate registered components for a certain Intent (implicit Intents).

```
public class Main2Activity extends Activity
{
        /** Called when the activity is first created. */
        .
        .
        .
                private Runnable mUpdateTimeTask = new Runnable()
        {
                public void run()
                {
                        // scans signal strengths of available access points in background
                        //at periodic intervals
                        . . . . . . . . . . .
                }
        };
        /** Called when the activity is first created. */

        public void onCreate(Bundle savedInstanceState)
        {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);

                •

                •

                •

                wifi.startScan();
        }

        public void update(List<ScanResult> results)
        {
                //receives user position update

                •

                •

                •

                wifi.startScan();

        }

}
```

SERVER SIDED PROGRAMING

Process.php

```
<?php
 $c1 = $_REQUEST['signal1'];
 $c2 = $_REQUEST['signal2'];
 $c3 = $_REQUEST['signal3'];
require ("config/db_confg.php");
$connection = mysql_connect($db_host, $db_user, $db_password) or die("err in
connection");
echo "connection made";
mysql_select_db($db_name);
$mdis=1000;
$result=mysql_query("select * from SIG_PROCESS ",$connection);
for($i=0;$i<mysql_num_rows($result);$i++)
{
        //algorithm which traces the user location
        •
        •
        •
        •
        •
        •
        •
}
header ("location:".$mainlink.".php");
mysql_close();
 ?>
```

Chapter 7
TESTING

Testing Detail:

The software test plan is designed to prescribe the scope, approach, resources and schedule of all testing activities. The plan identifies the items to be tested, the type of testing to be performed, the personnel responsibility for testing, the resources and schedule required to complete testing and the risk associated with the plan.

7.1 Purpose

Mission for testing the implementation is to

- Check out the robustness
- Identify the defects and track it to closure
- Check that the program demonstrates all functionalities as defined in the requirements.

7.1.1 Test objectives

The primary goal of testing is to check whether thoughts, actions and products are desirable. The goal of testing is to uncover and fix as many errors as possible and to remove non quality characteristics. The main objective is to verify that the system being tested should effectively satisfy all the objectives on which it has been built.

7.1.2 Test Types

1. Unit Testing
2. Integration Testing
3. System Testing

7.2 Conducting Tests

All the modules will be tested for quality of project. In this phase unit, integration and some level of system testing was conducted. Details of various test phases are as follows:

7.2.1 Unit Testing

Unit testing is the first level of testing. This is conducted to ensure that proper functionalities and code coverage have been achieved during code. The testing will focus on verification efforts on the code developed for each module. Each function is tested by giving all possible inputs. In the table below, test cases for the project are presented.

CLIENT:

Method Responsible	Test condition	Expected result	Actual result	Pass/fail
onCreate()	Wi-Fi not connected	Open Wi-Fi setting to connect Wi-Fi	As per the expected result	Pass
executeHttp()	Data from server not available	Terminate normally	Terminated normally	Pass
WifiReceiver.update()	Wi-Fi connection lost	Wait for connection and start scan	As per the expected result	Pass
wifiHandler()	Application terminated	Terminate handler and stop scan	As per the expected result	Pass

Table 7.1: Unit Testing of Client Module

SERVER

Test condition	Expected result	Actual result	Pass/fail
Positioning not available	Route to webpage not available	As per the expected result	Pass
Database connection	Connected to database	Connection established	Pass

Table 7.2: Unit Testing of Server Module

USER INTERFACE

Test condition	Expected result	Actual result	Pass/fail
Wi-Fi failed	Show notification	As per the expected result	Pass
User out of range	Show notification	As per the expected result	Pass

Table 7.3: Unit Testing of User Interface Module

7.2.2 Integration Testing

This is the second phase of testing. In its simplest form, two units that have already been unit tested are combined into a component and the interface between them is tested. This will help in assuring the seamless integration and consistency of an integrated sub system.

There are two parts in integration:

- Interaction between middle ware(positioning server) and the Database

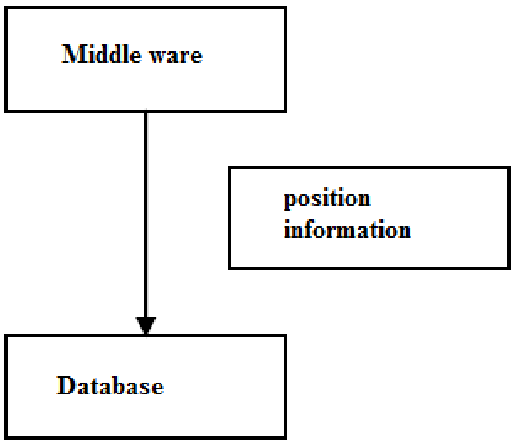


Figure 7.1: Interaction between middle ware (positioning server) and the Database

- Interaction between the UI and the Database

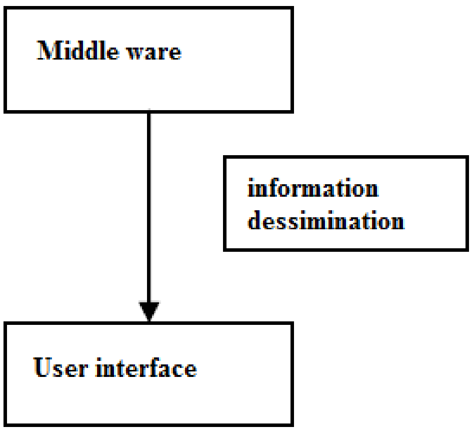


Figure 7.2: Interaction between the UI and the Database

7.2.3 System Testing

System testing focuses on the behavior of complete integrated systems to evaluate compliances with specified requirements. The testing was performed when the entire system is integrated with yielded positive results.

Chapter 8
CONCLUSION AND FUTURE ENHANCEMENT:

From the results, it is evident that position can be determined in an indoor environment using Wi-Fi access point's signal strengths. The accuracy of location determination depends on how close the two points to be distinguished are and signals from how many access points are being used. Using more number of access points surely makes it easy for a location detection system, simply because the amount of information that can be obtained increases with the addition of each extra access point, and the probability of two distant points having the same signal strength profile decreases. But it introduces problems related to network performance and hardware cost. There has to a trade-off between performance, cost and accuracy of the location detection system. Our system works quite well with a minimal number of access points and excels in performance if given an infrastructure with large number of access points giving an accuracy of almost 2-3 metres.

Our results indicate that it is possible to build an interesting class of location-aware services, such as printing to the nearest printer, navigating through a building, etc., on Using Wi-Fi Access Points, thereby adding value to such a network.

Chapter 9
BIBLOGROPHY AND REFERENCES:

[1] R. Want and B. Schilit, "Expanding the horizons of location-awarecomputing," IEEE Computer, vol. 34, no. 8, Aug. 2001.

[2] H. Gellersen, A. Schmidt, and M. Beigl, "Adding some smartness todevices and everyday things," in Proc. of IEEE Workshop on MobileComputing Systems and Applications, Dec. 2000.

[3] Jeffrey Hightower and Gaetano Borriello, "Location systems forubiquitous computing," IEEE Computer, vol. 34, no. 8, pp. 57–66,Aug. 2001.

[4] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricketlocation-support system," in Proc. of 6th ACM MOBICOM, Boston,MA, Aug. 2000, pp. 32–43.

[5] A. Ward, A. Jones, and A. Hopper, "A new location technique forthe active office," IEEE Personal Commun., vol. 4, no. 5, pp. 42–47,Oct. 1997.

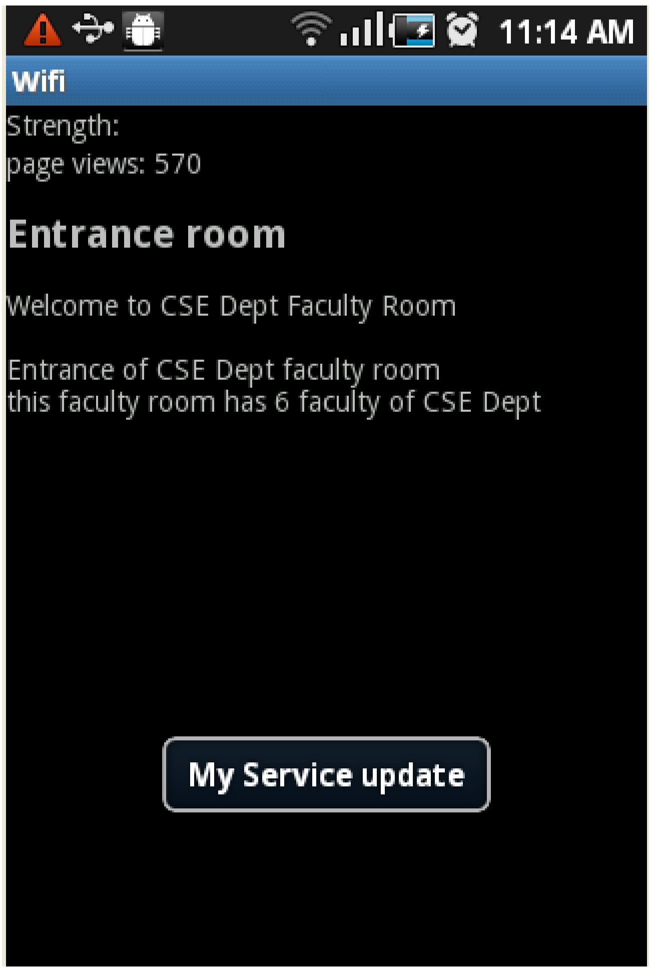
[6] ParamvirBahl and Venkata N Padmanabhan, "Radar: An in-buildingrf based user location and tracking system," in Proc. of IEEEINFOCOM 2000, Mar. 2000, pp. 775–784.

[7] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao, "A wirelessLAN based indoor positioning technology," IBM Journal of Research andDevelopment, vol. 48, no. 5/6, 2004.

[8] M. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh, "A probabilistic clustering-based indoor location determination system," University of Maryland, College Park, Tech. Rep. CS-TR 4350, 2002.

Chapter 10
SCREENSNAPSHOT

LOC 1:


11:14 AM

Wifi

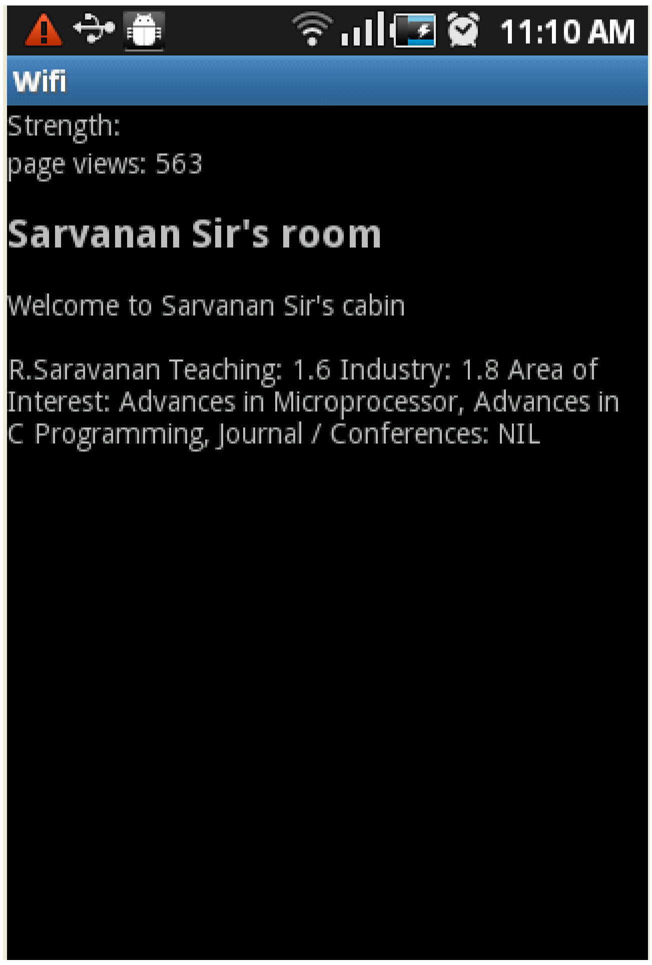
Strength:
page views: 570

Entrance room

Welcome to CSE Dept Faculty Room

Entrance of CSE Dept faculty room
this faculty room has 6 faculty of CSE Dept

LOC 2:



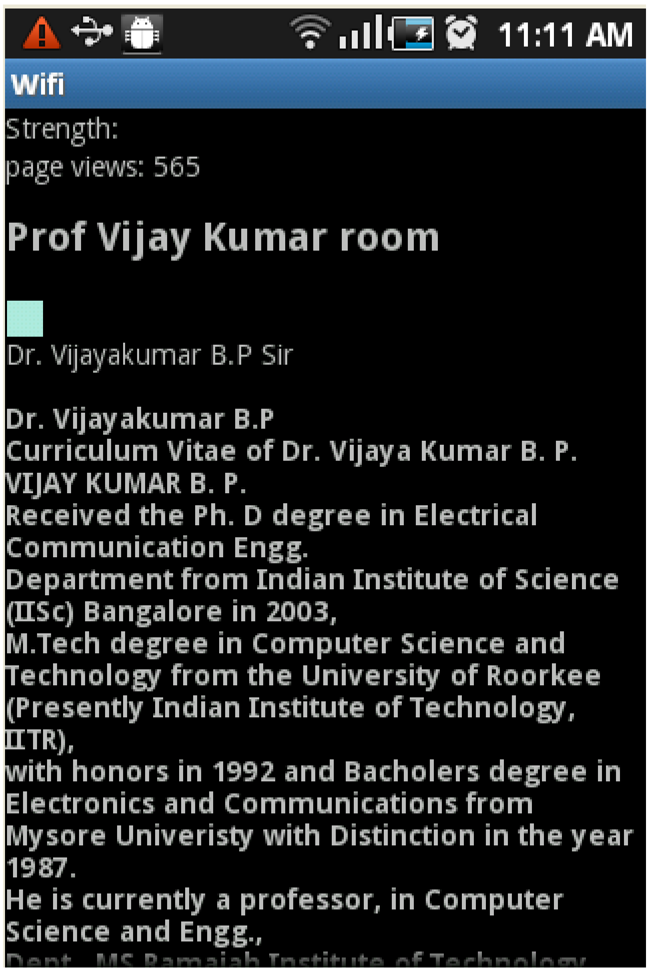
Strength:
page views: 563

Sarvanan Sir's room

Welcome to Sarvanan Sir's cabin

R.Saravanan Teaching: 1.6 Industry: 1.8 Area of Interest: Advances in Microprocessor, Advances in C Programming, Journal / Conferences: NIL

LOC 3:



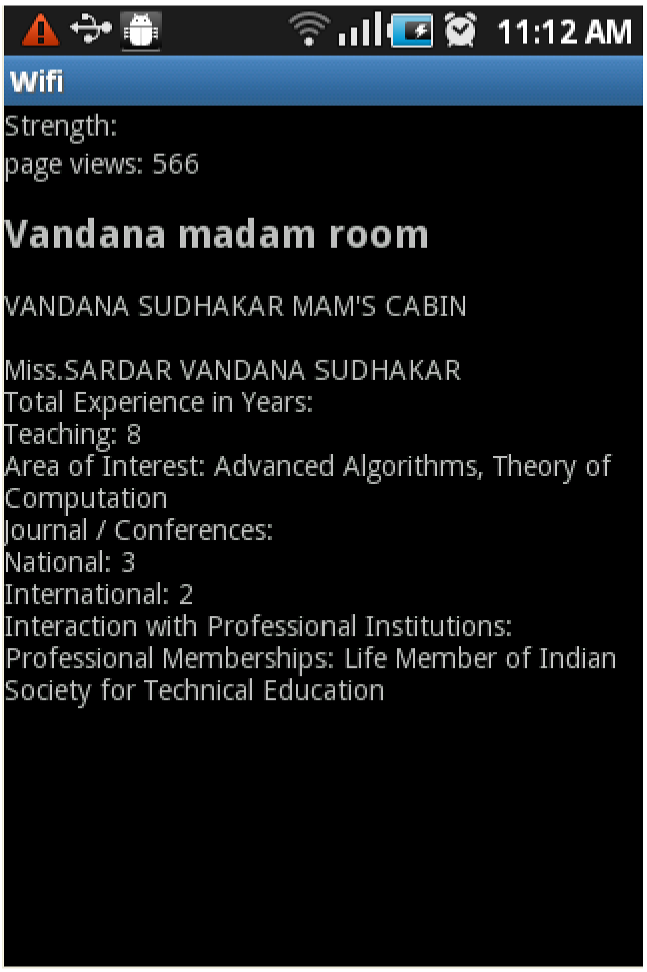
Strength:
page views: 565

Prof Vijay Kumar room

Dr. Vijayakumar B.P Sir

Dr. Vijayakumar B.P
Curriculum Vitae of Dr. Vijaya Kumar B. P.
VIJAY KUMAR B. P.
Received the Ph. D degree in Electrical Communication Engg.
Department from Indian Institute of Science (IISc) Bangalore in 2003,
M.Tech degree in Computer Science and Technology from the University of Roorkee (Presently Indian Institute of Technology, IITR),
with honors in 1992 and Bacholers degree in Electronics and Communications from Mysore Univeristy with Distinction in the year 1987.
He is currently a professor, in Computer Science and Engg.,

LOC 4:



11:12 AM

Wifi

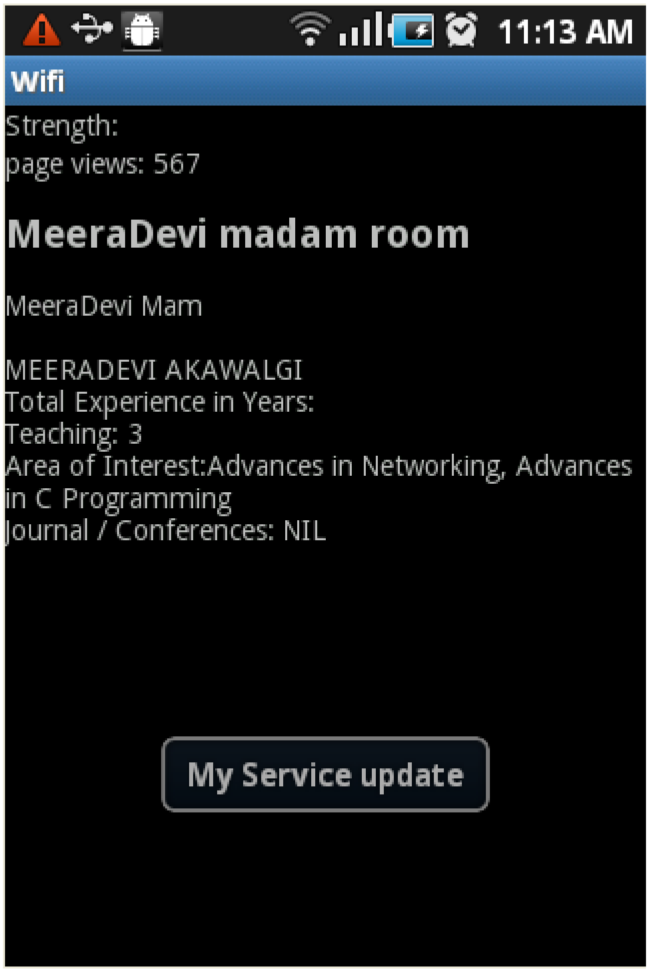
Strength:
page views: 566

Vandana madam room

VANDANA SUDHAKAR MAM'S CABIN

Miss.SARDAR VANDANA SUDHAKAR
Total Experience in Years:
Teaching: 8
Area of Interest: Advanced Algorithms, Theory of Computation
Journal / Conferences:
National: 3
International: 2
Interaction with Professional Institutions:
Professional Memberships: Life Member of Indian Society for Technical Education

LOC 5:



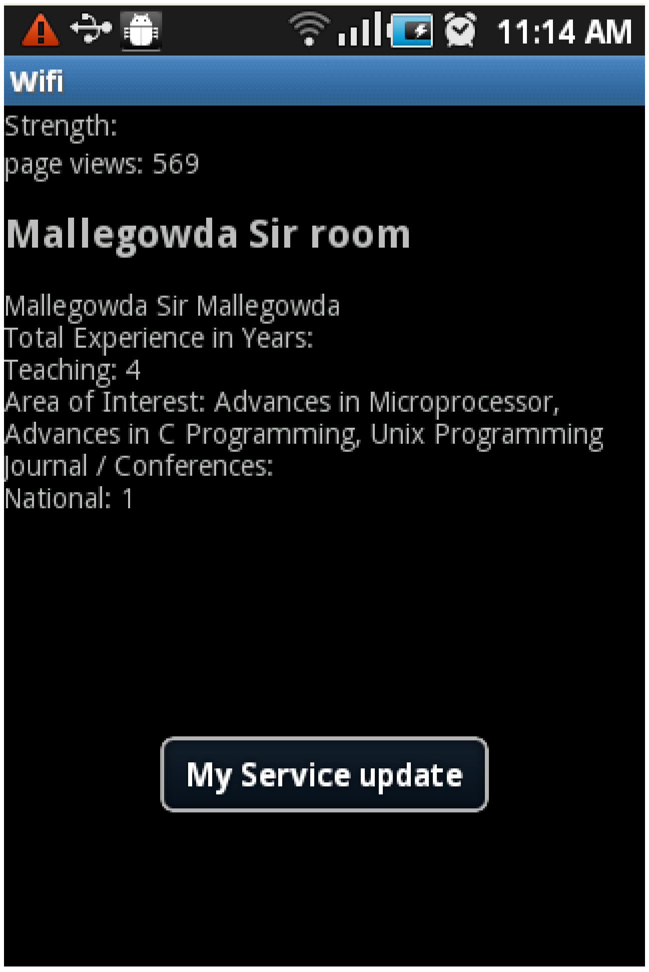
Strength:
page views: 567

MeeraDevi madam room

MeeraDevi Mam

MEERADEVI AKAWALGI
Total Experience in Years:
Teaching: 3
Area of Interest:Advances in Networking, Advances in C Programming
Journal / Conferences: NIL

LOC 6:

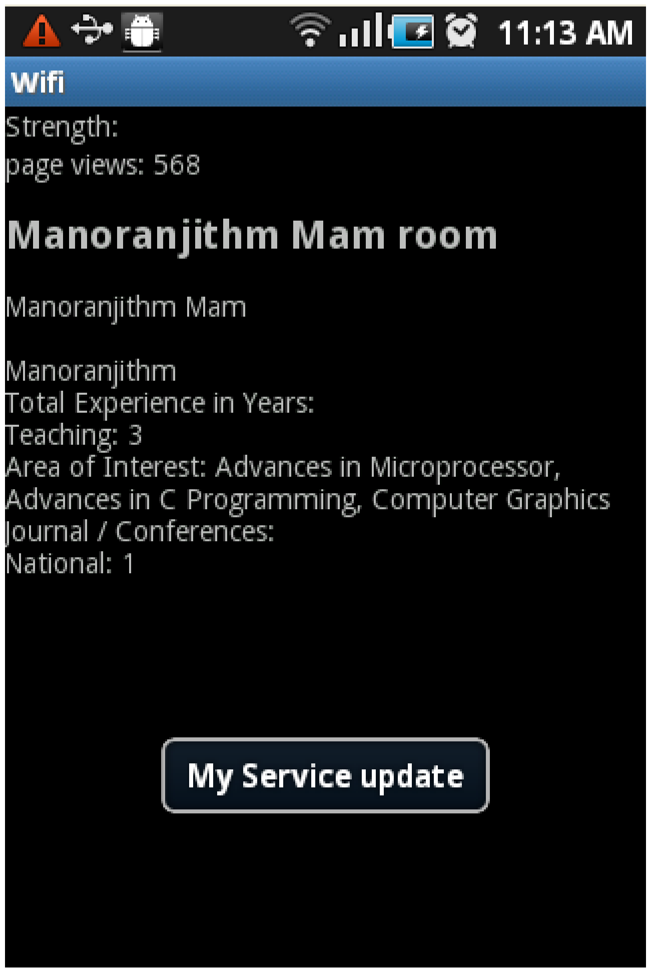


Strength:
page views: 569

Mallegowda Sir room

Mallegowda Sir Mallegowda
Total Experience in Years:
Teaching: 4
Area of Interest: Advances in Microprocessor, Advances in C Programming, Unix Programming
Journal / Conferences:
National: 1

LOC 7:



Strength:
page views: 568

Manoranjithm Mam room

Manoranjithm Mam

Manoranjithm
Total Experience in Years:
Teaching: 3
Area of Interest: Advances in Microprocessor, Advances in C Programming, Computer Graphics
Journal / Conferences:
National: 1