ANDROID BASED VEHICLE TRACKING SYSTEM

Thesis Report

submitted in partial fulfillment of the requirements

for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Abhinav Bhargav

(801432001)

Under the supervision of:

Dr. Parteek Bhatia

Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2016

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Android Based Vehicle Tracking System"**, in partial fulfilment of the requirements for the award of degree of Master of Engineering in **Computer Science and Engineering** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Parteek Bhatia** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Abhinav Bhargav

801432001

ME(CSE)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Parteek Bhatia
Associate Professor
Computer Science and Engineering Department
Thapar University

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

i

Acknowledgement

First of all, I would like to thank the Almighty, who has always guided me to work on the right path of the life. It is a great privilege to express my gratitude and admiration towards my respected supervisor Dr. Parteek Bhatia. He has been an esteemed guide and great support behind achieving this task. This work would not have been possible without the encouragement and valuable guidance of him. I am thankful for his time, patience, discussions and valuable comments. His enthusiasm and optimism made this experience both rewarding and enjoyable. I am truly grateful to him for extending his total cooperation and understanding whenever I needed help and guidance from him. I am also heartily thankful to Dr. Deepak Garg and Dr. Maninder Singh for motivation and providing useful guidance and support throughout the preparation of the thesis report.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable. Last but not least, I would like to thank my family for their wonderful love and encouragement, without their blessings none of this would have been possible.

Abhinav Bhargav

ii

Abstract

The current location and estimated arrival time and distance of the vehicle is provided to the passenger by the vehicle tracking system. In relation to the estimated arrival time the passengers can decide whether to wait for the vehicle or not. This helps in making better travelling decisions by the passengers. From this system it can also be determined whether the vehicle is yet to come or has been missed. To make life more efficient the system provides necessary basic information.

Android based vehicle tracking system is required which tracks the real time location of the vehicle and this information is used effectively by the commuters to make better travelling decision. GPS is growing its popularity in day to day life applications. In this era of 4^{th} Generation Smartphone and palmtops have become a valuable part of the human beings. We often listen to the words Android and Maps. Android has made life easier and comfortable

This thesis describes the process of designing a client server android based application for the efficient vehicle tracking. Quality of life of people in India is demeaning with the growing traffic congestion. Congestion leads to air pollution, decrease in accessibility and increased travel time. System and technology is growing even in microseconds. GPS is mounting its reputation in day to day life applications. With increasing use of smart phones and growing popularity of the GPS based applications, it has become feasible for the commuter to track the real time position of the vehicle and make better travelling decision.

Table of Contents

List of Figures

Chapter 1
Introduction

1.1 Introduction to GPS Based Vehicle Tracking System

Various uncertain conditions in the daily operation of public transportation systems, affects the movement of vehicles as the day progresses. Uncertainty in passenger demand, traffic congestion, uneven vehicle dispatching times, unexpected delays and incidents are the major reasons for the passengers to reach office late. Many commuters are late to their respective errands because they are unable to choose whether to wait for the vehicle or take an alternative. LED (Light Emitting Diode) screens at bus stops showing the bus arrival time help reduce the apprehension of passengers waiting for the bus. Interfaces such as smart phone could make public conveyance system more user-friendly by disseminating arrival time information of the bus and thus increases its effectiveness among various transportation modes [7].

The current position and estimated arrival time of the vehicle is provided to the passenger by the Vehicle Tracking System. In relation to the estimated arrival time the passengers can decide whether to wait for the vehicle or not. This helps in making better travelling decisions by the passengers. From this system it can also be determined whether the vehicle is yet to come or has been missed. To make life more efficient the system provides necessary basic information.

For enhancing public transport, it has to be made more reliable for which various concepts have been proposed. Intelligent transport systems are developed to make transportation easier and comfortable. Intelligent Transport System is route that is tested to mitigate traffic congestion models. There are various concepts that are included in it. Some of them are providing the commuters with the reliable information like real time passenger information system, pre-trip information systems, vehicle arrival notifications, automatic vehicle location system, timed transfers and determining priority of road to vehicle at intersections. Other notions are about providing comfort, improving stops and number of passenger information systems.

By providing the information of the arrival time of the transit vehicle, GPS (Global Positioning System), Wireless communication systems and other systems have made public transport in developing countries way more reliable. Travel time information system is another most important information system. However, directly this information cannot be extracted. The satisfaction of transit users and the ridership increases with the reliable travel time information system. As the number of vehicle reduces this decreases the congestion. Successful real time vehicle arrival system on highways have been developed by many metropolitan areas of developed countries but due to stochastic nature of urban traffic still there are difficulties in the provision of real time vehicle arrival time information on urban streets. Considering undisciplined traffic and the lack of collected data, application of such a system in developing countries is even worse. Hence, to come with better vehicle arrival systems a good algorithm that can predict the arrival time of the vehicle with reasonable accuracy is required.

1.2 Need for GPS Based Vehicle Tracking System

Following are the needs for Vehicle Tracking System.

- Quality of life of people in India is demeaning with the growing traffic congestion. Congestion leads to air pollution, decline in accessibility and amplified travel time. In developed nations still many people use their private vehicles. In developing nations also, the degree of vehicle ownership is increasing at a faster rate. Many concepts have been applied for the mitigation of congestion. One of them is to expand and improve the public transportation system. A good public transport system is very important for the financial growth of the country. With good and reliable public transport satisfaction among the travellers increases which decreases the number of private vehicles. A good public transport system advances the quality of life providing better availability, mobility, social cohesion and secures the environment.

- Public transport can be impractical for people who need to follow strict schedules. The variations from the official vehicle schedule are reasonable and inevitable. But along with this if there is lack of communication regarding delay of the vehicle then traveller might be wasting time. If the vehicle is late it will make the travellers late and if the vehicle is early than its time will even, make travellers late because they might miss the vehicle. Even if the vehicle is on time, travellers have no way to know the information about the vehicle and ends up adopting other modes of transportation.

- With the increasing use of smart phones, it has become very easy for individuals to stay in contact. Initially business was conducted only during business hours and pre-planned sites. With the advances in recent years, meetings have become spontaneous. Because of these advances more precision is required in arrangement of activities with more accuracy.

- These technologies have improved making business run more efficiently by making commuters who use public transport reach on time. The lack of certainty that vehicle travellers face applies not only to business world but also to everyone. Students need to reach on time for their classes, commuters traveling to go social get together. This technology helps the transit users know the potential delay in the vehicle schedules so that they can plan accordingly and increase the efficiency. Employees can inform the managers or the clients about the delay and postpone the meetings. Students can also inform about their delay to the respective group fellows.

1.3 Techniques for Vehicle Tracking System

Vehicle Tracking System can be broadly classified into Real Time Location Based System and the prediction system as shown in Figure 1.1. Real Time Location Based System uses the current location and speed of the vehicle to calculate its arrival time and prediction system takes into consideration the traffic pattern on the route to calculate the arrival time of the vehicle. However, here the emphasis is on real time location based system.

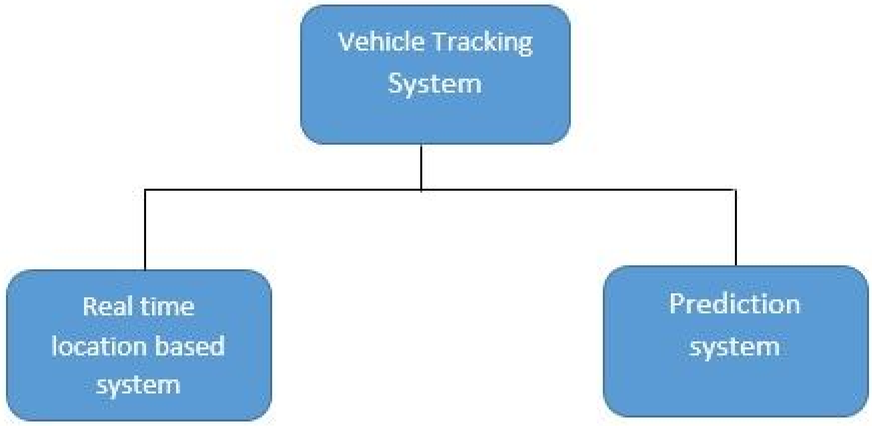

Figure 1.1 Techniques for Vehicle Tracking System

Real Time Location Based System can be further classified as follows.

i. Real time monitoring using LCD (Liquid Crystal Display)
ii. SMS (Short Message Service) Based Vehicle Tracking System
iii. Web based application
iv. Mobile application

Prediction system can be further classified as follows.

i. Historical data based model
ii. Time series model
iii. Regression model
iv. Kalman Filtering model
v. Machine learning model

Real Time Location Based Systems can also be standalone systems that displays the arrival time of the transit vehicle on the LCD screens that are placed on every stop, through SMS facility, web based application or by using android application using GPRS (General Packet Radio Service). Real time location based models are illustrated below.

i. Real Time Location Based Systems uses LCD screens. These screens are placed on every stop, displaying the arrival time of the vehicle using RF (Radio Frequency) transceivers. The location is displayed on the LCD screen along with vehicle number as the vehicle enters the range of reception.

ii. SMS is used over the GSM (Global System for Mobile communication) networks to transfer the vehicle location coordinates. The location information is sent to the central server over the GSM networks using SMS and is stored in the database. The commuter sends the request and receives the information through SMS.

iii. Through the web based application users can track the vehicle graphically. Also the web based systems permits users with different operating systems platforms to easily reach the details with the help of internet access.

iv. Vehicle Tracking System using Android application has inbuilt GPS service provided by the Smartphone to get its GPS coordinates. These GPS coordinates are transferred to the central server. The users can retrieve information through android application.

Another type of vehicle arrival time system is based on prediction system. It is based on the large amount of collected data.

1.4 Technologies

GPS based tracking system is required which tracks the real time position of the vehicle and this information is used effectively by the commuters to make better travelling decision. GPS is growing its popularity in day to day life applications. In this era of 4^{th} Generation Smartphone and palmtops have become a valuable part of the human beings. Android has made life easier and comfortable. The technologies like GPS and Android are illustrated below.

1.4.1 Global Positioning System

Now-a-days various agencies employ GPS to track the position of the transit's vehicle. The major concern is reliability of the results obtained from the GPS. GPS is considered to be one of the most widely used and reliable techniques for tracking the location. It is operated by United States of Department of Defence. The idea behind the working of GPS is that time is calculated by the receiver that a pseudo code takes to get from the GPS Satellite to that receiver on the surface of earth. This time is approximately 0.1 seconds in practice. From this distance, the distance(X) between the receiver on earth's surface and the GPS Satellite can be calculated, as speed of sending code is known and the time can be measured. The Satellite imaginarily places the receiver somewhere on the surface of a sphere with radius (X). The setup of the arrangement is shown in Figure 1.2a. Then the second satellite carries the same process. This narrows the possibilities of existence of the receiver as it can be present only at intersection of the two virtual spheres. The elliptical region is the intersection of the two spheres where there is a possibility of the receiver to be present. Figure 1.2b shows the arrangement. By further carrying out the same process for the third satellite, the possible locations can be further reduced to just two points as shown in Figure 1.2c.

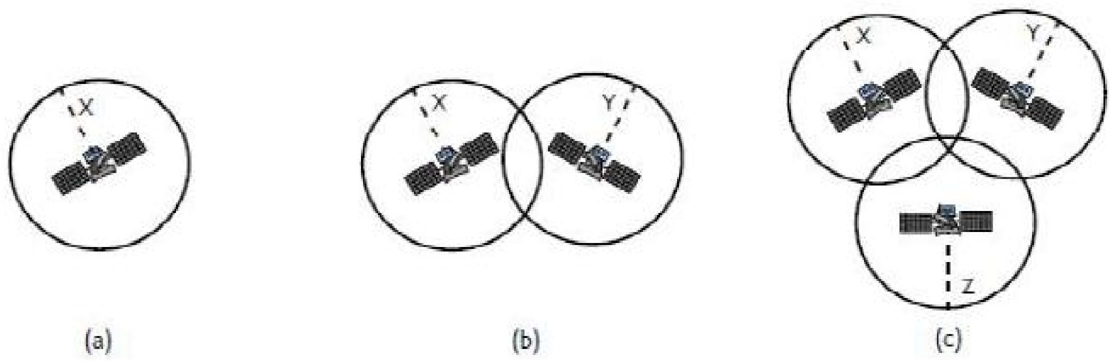

Figure 1.2 Computation of GPS position using (a) one satellite (b) two satellites (c) three satellites.

One of the two points is ignored because either it is moving with unrealistic speed or it is much far from the surface of the earth. Measurements from the fourth satellite are counted if three dimensional positions are required, *i.e.*, longitude, latitude and altitude. Figure 1.2 explains the 2- dimensional system.

The accuracy of the standard GPS facility is satisfactory, but there are few applications where much higher accuracy is required. For standing up to such higher accuracy differential GPS was developed. Errors caused due to delays when signal passes through the troposphere and the ionosphere are corrected in the differential GPS for higher accuracy. GPS gives longitude and latitude in following 3 conventions.

- Degrees minutes' seconds: 40° 27′ 46″ N 79° 48′ 56″ W
- Degrees decimal minutes: 40° 16.767′ N 79° 158.933′ W
- Decimal degrees: 40.436° N 79.782° W

The system implementation uses Decimal degrees' representation. Following are the sources of errors that might arise.

- Disturbances in the atmosphere slow down the speed with which the radio waves travel.
- In the absence of clear Line of Sight (LOS) like underground subways, clouds correct results might not be collected.
- Skyscrapers bounce off the radio waves which gives the wrong results.
- Satellite might also send corrupt location data, misreporting the position.

Following are the methods of correction.

- By using Differential GPS (DGPS) hardware receiver's inaccuracy can be easily calculated. The signal correction information is broadcasted by the station for that area.
- Advanced systems which have higher accuracy are deployed for the area where GPS enabled devices is present and estimates the distance to the base station.
- Signals between the adjacent antenna service providers are interpolated with which device is always connected.

India has developed IRNSS (Indian Regional Navigation Satellite System) which is an autonomous regional navigation satellite system. The primary goal is to provide accurate location information facility to Indian users and also to the areas ranging 1500 km from the boundary of India. IRNSS is likely to provide location accuracy of better than 20m [19].

1.4.1.1 Architecture of IRNSS

The architecture of IRNSS mainly consists of following segments.

i. Space Segment
ii. Ground Segment
iii. User Segment

Figure 1.3 depicts the architecture of IRNSS

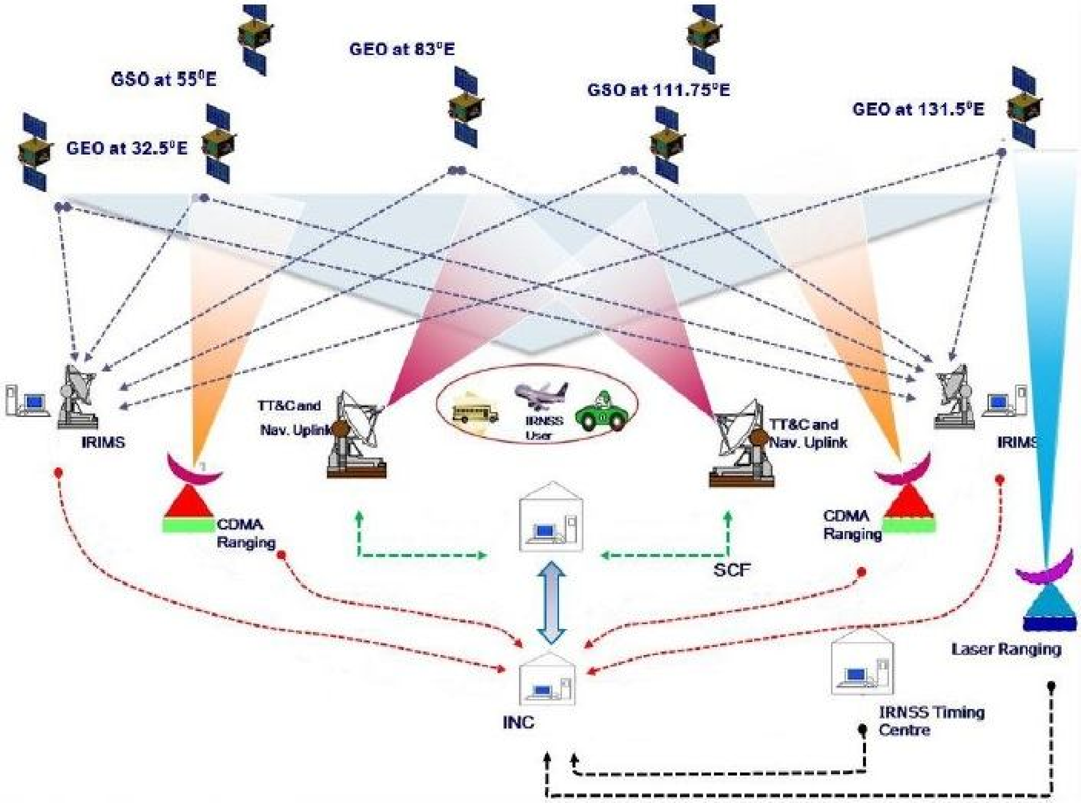

Figure 1.3 Architecture of IRNSS [19]

i. IRNSS Space Segment

Based on various considerations the minimum number of satellites required for IRNSS constellation is worked out to be 7 (4 GSO and 3 GEO). The 4 inclined GSOs (Geosynchronous Orbit) have their longitude crossings 55° E and 111.75° E (two in each plane) and the 3 GEOs (Geostationary Orbit) are located at 32.5° E, 83° E, 131.5°E.

ii. IRNSS Ground Segment

Ground Segment is responsible for the operation and maintenance of the IRNSS constellation. The Ground Segment includes.

- ISRO Navigation Centre
- IRNSS Spacecraft Control Facility
- IRNSS Range and Integrity Monitoring Stations
- IRNSS Network Timing Centre
- IRNSS CDMA (Code Division Multiple Access) Ranging Stations
- Laser ranging Stations
- Data Communication Network

iii. IRNSS User Segment

The user segment primarily consists of.

- Signal frequency IRNSS receiver capable of receiving SPS (Standard Position Services) signal at L5 (1164.45 – 1188.45 MHz) or S (2483.5 – 2500 MHz) band frequency.
- A dual frequency IRNSS receiver capable of receiving both L5 and S band frequencies.
- A receiver compatible to IRNSS and other GNSS signals.

Figure 1.4 specifies the radio frequency interface between space and user segment. Each IRNSS satellite provides SPS signals in L5 and S band.

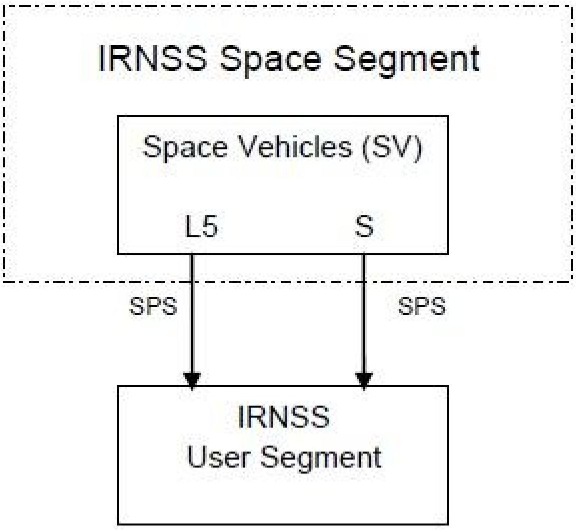

Figure 1.4 IRNSS Space Segment Interface with User Segment [19]

Following are some applications of IRNSS.

- Disaster Management
- Voice and visual navigation for drivers
- Marine, Aerial and Terrestrial Navigation
- Integration with mobile phones
- Terrestrial navigation aid for hikers and travellers
- Vehicle tracking and fleet management
- Precise Timing

1.4.2 Android

Android Smartphone consists of GPS system, with the use of which the current position of the transit vehicle and the commuter can be easily tracked. GPS system in the Smart phones can be easily used with appropriate security permissions. Android is developed by Google and is an open source operating system for mobile devices. One of the key feature provided by Android is the set of Android applications (apps) provided through Android market that enables the users to extend the functionality. Android applications have made life much easier and faster. Applications are made very user friendly which makes it very popular. These apps are developed by professionals and a group of hobbyist programmers using enhances form of java. Android provides a lot of documented help for new developers.

The Android SDK (Software Development Kit) is a set of tools and APIs (Application Program Interface) that facilitates the development of Android application. The Android SDK Manager provides an excellent feature to install API components according to the different versions of the Android OS (Operating System) with ease. Another key feature of SDK is that it provides the developers with Android emulator which ensures the developers to deploy their code and test its working on different virtual phones with different specifications. Figure 1.5 shows the architecture of android. Android Software stack or Android architecture is roughly divided into following categories.

i. **Linux kernel**

It exists at the root of android architecture and is the heart of android architecture. Linux kernel is responsible for memory management, device management, power management, resource access and device drivers.

ii. **Native Libraries**

On the top of Linux kernel, there are libraries such as SQLite, FreeType, Webkit, Media, OpenGL, C runtime library *etc*. The SQLite is responsible for database, WebKit for browser support, Media for playing and recording audio and video formats, FreeType for font support.

iii. **Android Runtime**

In android runtime, there is DVM (Dalvik Virtual Machine) whose responsibility is to run android application and core libraries. DVM is like JVM but it is enhanced for mobile devices. It consumes less memory and provides fast performance. The Dalvik VM makes usage of Linux core features like memory management and multi-threading which is inherent in the Java language. The Dalvik VM permits all Android application to run in its own process, with its own instance of the Dalvik virtual machine.

iv. **Application Framework**

On the top of android runtime and libraries, there is android framework. It includes Android API's such as telephony, UI (User Interface), resources, package managers, Content Providers (data), locations. For android application development, application framework provides a lot of interfaces and classes.

v. **Applications**

At the top layer, there are Android applications. You will carve your application to be fixed on this layer only. Home, contact, settings, games, browser *etc*. are the examples of such applications.

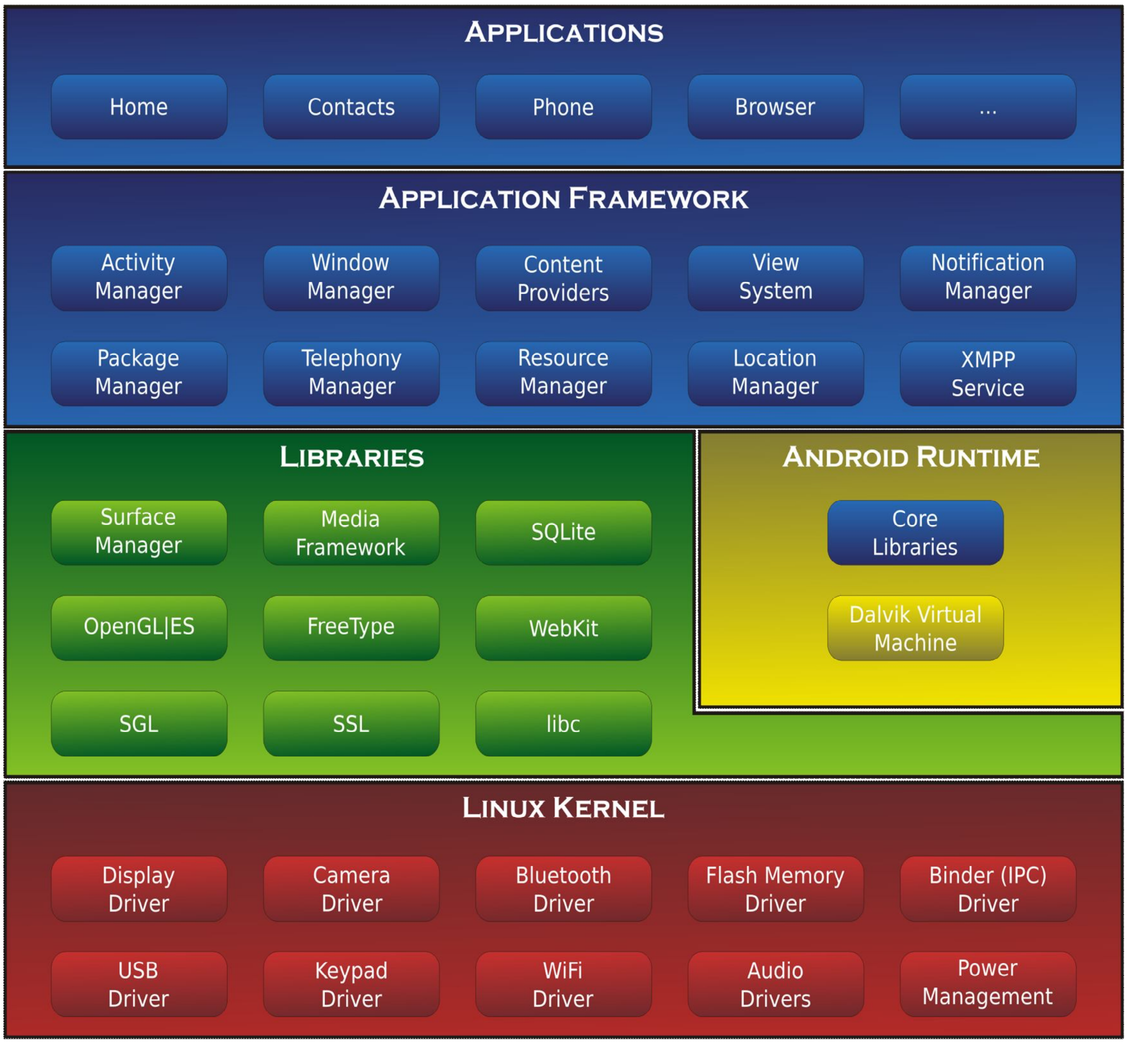

Figure 1.5 Android architecture [15]

1.5 Various Classes and Interfaces

Following are the classes and interfaces discussed.

1.5.1 Activity

As in C, C++ or Java programming language the program starts from main() function. Very similar way, Android structure initiates its program with in an Activity opening with a call on onCreate() callback method. There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity as shown in Figure 1.6 Activity life cycle diagram. Figure 1.7 shows various events of Activity class.

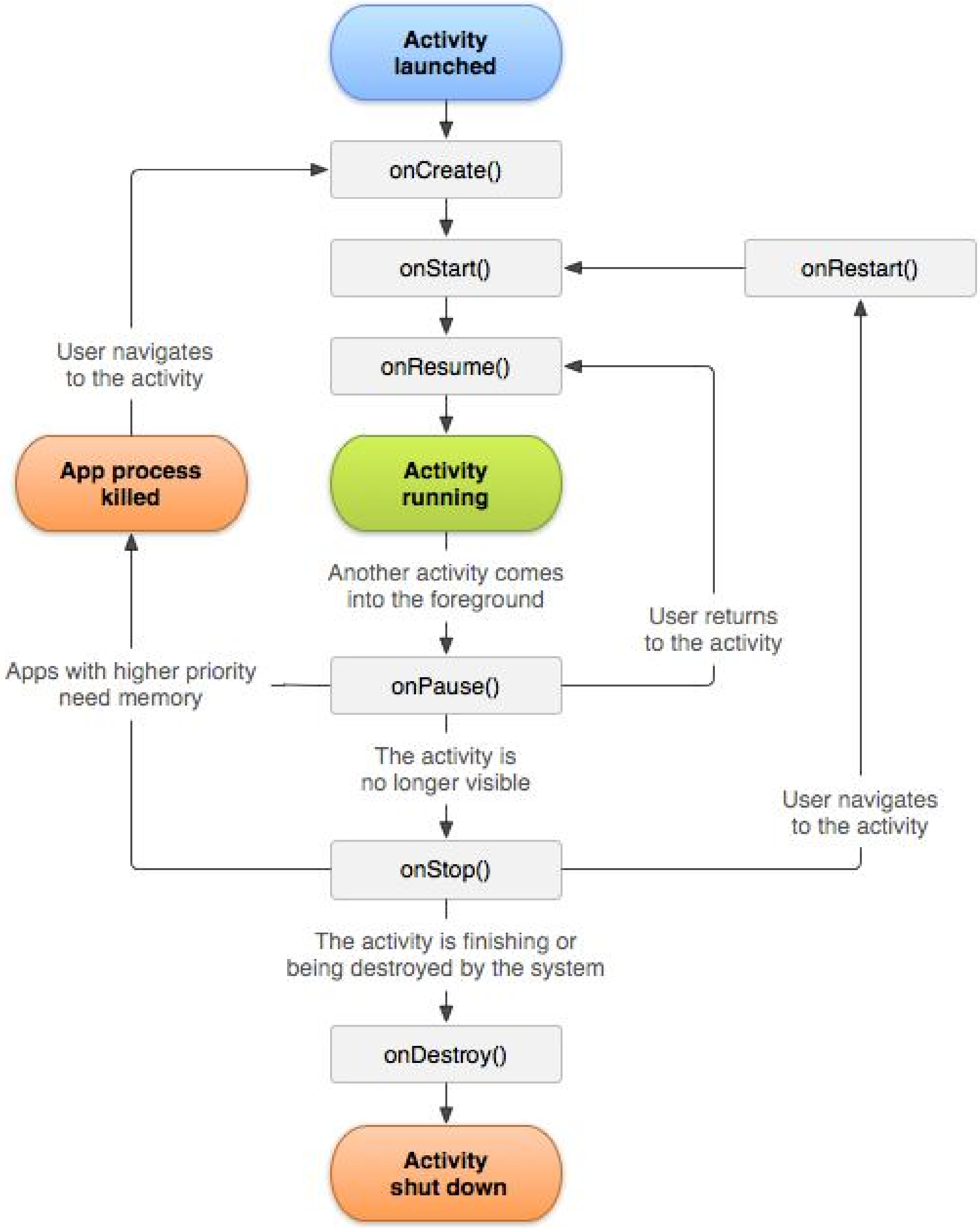

Figure 1.6 Activity life cycle diagram [22]

Callback	Description
onCreate()	This is the first callback and called when the activity is first created.
onStart()	This callback is called when the activity becomes visible to the user.
onResume()	This is called when the user starts interacting with the application.
onPause()	The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.
onStop()	This callback is called when the activity is no longer visible.
onDestroy()	This callback is called before the activity is destroyed by the system.
onRestart()	This callback is called when the activity restarts after stopping it.

Figure 1.7 Events of Activity class [22]

1.5.2 Location

The Location object signifies a terrestrial location which can consist of a time stamp, longitude, latitude, and other information such as altitude, velocity and bearing. To get location specific information, methods can be used with Location object as shown in Figure 1.8.

Method & Description
double getLatitude() Get the latitude, in degrees.
double getLongitude() Get the longitude, in degrees.
float getSpeed() Get the speed if it is available, in meters/second over ground.
void setLatitude(double latitude) Set the latitude, in degrees.
void setLongitude(double longitude) Set the longitude, in degrees.
void setSpeed(float speed) Set the speed, in meters/second over ground.

Figure 1.8 Methods of Location class [22]

1.5.3 LocationManager

Access to the system location services is provided by LocationManager class. These services license applications to get intervallic updates of the device's physical position, or to fire an application-specified Intent when the device enters the closeness of a given physical position. Figure 1.9 shows various constants of LocationManager class. All Location API methods require the ACCESS_FINE_LOCATION or ACCESS_COARSE_LOCATION permissions.

Constants	
String	GPS_PROVIDER Name of the GPS location provider.
String	NETWORK_PROVIDER Name of the network location provider.

Figure 1.9 Constants of LocationManager class [15]

1.5.4 Geocoder

Geocoder is a class that handles reverse geocoding and geocoding. Geocoding is the process of altering a street address into a (longitude, latitude) coordinate. Reverse geocoding is the process of altering a (longitude, latitude) coordinate into address. The amount of detail in a reverse geocoded location description may differ, for example one might contain the full street address of the nearby building, while another might contain only a postal code and city name. A backend service is required by the Geocoder class that is not contained within the core android framework. Geocoder query methods will return an empty list if there is no backend service in the platform. Use the isPresent() method to determine whether a Geocoder implementation exists as shown in Figure 1.10.

Public constructors
Geocoder(Context context, Locale locale) Constructs a Geocoder whose responses will be localized for the given Locale.
Geocoder(Context context) Constructs a Geocoder whose responses will be localized for the default system Locale.

Public methods	
List<Address>	getFromLocation(double latitude, double longitude, int maxResults) Returns an array of Addresses that are known to describe the area immediately surrounding the given latitude and longitude.
List<Address>	getFromLocationName(String locationName, int maxResults, double lowerLeftLatitude, double lowerLeftLongitude, double upperRightLatitude, double upperRightLongitude) Returns an array of Addresses that are known to describe the named location, which may be a place name such as "Dalvik, Iceland", an address such as "1600 Amphitheatre Parkway, Mountain View, CA", an airport code such as "SFO", etc..
List<Address>	getFromLocationName(String locationName, int maxResults) Returns an array of Addresses that are known to describe the named location, which may be a place name such as "Dalvik, Iceland", an address such as "1600 Amphitheatre Parkway, Mountain View, CA", an airport code such as "SFO", etc..
static boolean	isPresent() Returns true if the Geocoder methods getFromLocation and getFromLocationName are implemented.

Figure 1.10 Constructors and methods of Geocoder class [15]

1.5.5 View

The basic building block for user interface components is represented by this class. A view is responsible for event handling and drawing and on the screen it occupies a rectangular area. For widgets, view is the base class, which are used to make interactive UI components (text fields, buttons *etc*.).

IDs

An integer id may be associated with views. These ids are used to find specific views within the view tree and are assigned in the layout XML files. A common pattern is to:

- Define a Button in the layout file and assign it a unique ID.

```
<Button
    android:id=”@id/my_button”
    android:layout_height=”wrap_content”
    android:layout_height=”wrap_content”
```

android:text=”@string/my_button_text” />

- From the onCreate method of an Activity, find the Button

  Button myButton=(Button) findViewById(R.id.my_button);

1.5.6 Toast

A toast is a view containing a quick little message for the user. The toast class helps you create and show those. Figure 1.11 shows various constants and methods of Toast class.

Constants	
int	LENGTH_LONG Show the view or text notification for a long period of time.
int	LENGTH_SHORT Show the view or text notification for a short period of time.

Public methods	
static Toast	makeText(Context context, int resId, int duration) Make a standard toast that just contains a text view with the text from a resource.
static Toast	makeText(Context context, CharSequence text, int duration) Make a standard toast that just contains a text view.
void	show() Show the view for the specified duration.
void	setMargin(float horizontalMargin, float verticalMargin) Set the margins of the view.

Figure 1.11 Constants and methods of Toast class [15]

1.5.7 OutputStream

The OutputStream abstract class is the superclass of all classes representative of an output stream of bytes. An output stream accepts output bytes and directs them to some sink. Applications that want to define a subclass of OutputStream must always provide at least a method that writes one byte of output. Figure 1.12 shows constructor and various methods of OutputStream class.

Public constructors
OutputStream()

Public methods	
void	close() Closes this output stream and releases any system resources associated with this stream.
void	flush() Flushes this output stream and forces any buffered output bytes to be written out.
void	write(byte[] b) Writes b.length bytes from the specified byte array to this output stream.
void	write(byte[] b, int off, int len) Writes len bytes from the specified byte array starting at offset off to this output stream.
abstract void	write(int b) Writes the specified byte to this output stream.

Figure 1.12 Constructor and methods of OutputStream class [15]

1.5.8 ByteArrayOutputStream

This class implements an output stream in which the data is written into a byte array. As the data is written to it, the buffer automatically grows. The data can be regained using toString() and toByteArray(). There is no effect on closing a ByteArrayOutputStream. After the stream has been closed, the methods in this class can be called without generating an IOException. Figure 1.13 shows various constructors and methods of ByteArrayOutputStream class.

Public constructors
ByteArrayOutputStream() Creates a new byte array output stream.
ByteArrayOutputStream(int size) Creates a new byte array output stream, with a buffer capacity of the specified size, in bytes.

Public methods	
void	close() Closing a ByteArrayOutputStream has no effect.
byte[]	toByteArray() Creates a newly allocated byte array.
String	toString(String charsetName) Converts the buffer's contents into a string by decoding the bytes using the specified charsetName.
String	toString() Converts the buffer's contents into a string decoding bytes using the platform's default character set.
void	write(byte[] b, int off, int len) Writes len bytes from the specified byte array starting at offset off to this byte array output stream.
void	write(int b) Writes the specified byte to this byte array output stream.

Figure 1.13 Constructors and methods of ByteArrayOutputStream class [15]

1.5.9 InputStream

This abstract class is the superclass of all classes representative of an input stream of bytes. Applications that want to define a subclass of InputStream must always provide a method that returns the succeeding byte of input. Figure 1.14 shows constructor and various methods of InputStream class.

Public constructors
InputStream()

Public methods	
void	close() Closes this input stream and releases any system resources associated with the stream.
abstract int	read() Reads the next byte of data from the input stream.
int	read(byte[] b, int off, int len) Reads up to len bytes of data from the input stream into an array of bytes.
int	read(byte[] b) Reads some number of bytes from the input stream and stores them into the buffer array b.

Figure 1.14 Constructor and methods of InputStream class [15]

1.5.10 PrintStream

A PrintStream enhances functionality to another output stream, namely the ability to print representations of numerous data values conveniently. Two other features are provided as well. A PrintStream never throws an IOException; instead, exceptional situations merely set an internal flag that can be tested via the checkError method. A PrintStream can be created so as to flush automatically, this means that the flush method is automatically invoked after a byte array is written, a newline character or byte ('\n') is written, or one of the println methods is invoked. Figure 1.15 shows various constructors and methods of PrintStream class.

Public constructors
`PrintStream(OutputStream out)` Creates a new print stream.
`PrintStream(OutputStream out, boolean autoFlush)` Creates a new print stream.
`PrintStream(OutputStream out, boolean autoFlush, String encoding)` Creates a new print stream.

Public methods	
`void`	`close()` Closes the stream.
`void`	`print(int i)` Prints an integer.

Figure 1.15 Constructors and methods of PrintStream class [15]

The parameter inside print() function can be of double, Boolean, char, long, float string object type.

1.5.11 InetAddress

InetAddress class represents an Internet Protocol (IP) address. An IP address is either a 128-bit unsigned number or a 32-bit used by IP, a lower level protocol on which protocols like TCP and UDP are built. Figure 1.16 shows methods of InetAddress class.

Public methods	
String	getHostAddress() Returns the IP address string in textual presentation.
boolean	isSiteLocalAddress() Utility routine to check if the InetAddress is a site local address.

Figure 1.16 Methods of InetAddress class [15]

1.5.12 NetworkInterface

It denotes a Network Interface made up of list of IP addresses allotted to this interface and a name. It is used to recognize the local interface on which a multicast group is joined. Figure 1.17 shows methods of NetworkInterface class.

Public methods	
Enumeration<InetAddress>	getInetAddresses() Convenience method to return an Enumeration with all or a subset of the InetAddresses bound to this network interface.
static Enumeration<NetworkInterface>	getNetworkInterfaces() Returns all the interfaces on this machine.

Figure 1.17 Methods of NetwrokInterface class [15]

1.5.13 Serversocket

A server socket waits over the network for requests to come in. It accomplishes some action based on that request, and then probably yields a result to the requester. Figure 1.18 shows constructors and methods of ServerSocket Class.

Public constructors
ServerSocket() Creates an unbound server socket.
ServerSocket(int port) Creates a server socket, bound to the specified port.

Public methods	
Socket	accept() Listens for a connection to be made to this socket and accepts it.
void	close() Closes this socket.
int	getLocalPort() Returns the port number on which this socket is listening.

Figure 1.18 Constructors and methods of ServerSocket Class [15]

1.5.14 Intent

An intent is an abstract explanation of an action to be completed. It can be used with broadcastIntent to send it to any interested BroadcastReceiver components, startActivity to launch an Activity, and bindService(Intent, ServiceConnection, int) or startService(Intent) to communicate with a background Service. The following Figure 1.19 shows methods of Intent class.

Sr.No	Method & Description
1	**Context.startActivity()** The Intent object is passed to this method to launch a new activity or get an existing activity to do something new.
2	**Context.startService()** The Intent object is passed to this method to initiate a service or deliver new instructions to an ongoing service.
3	**Context.sendBroadcast()** The Intent object is passed to this method to deliver the message to all interested broadcast receivers.

Figure 1.19 Methods of Intent class [22]

1.5.15 LocationListener

When the location has changed, the LocationListener interface is used for receiving notifications from LocationManger. If the LocationListener has been registered with the location manager service using the requestLocationUpdates(String, long, float, LocationListener) method, then these methods are called. Figure 1.20 shows methods of LocationListener interface.

Public methods	
abstract void	onLocationChanged(Location location) Called when the location has changed.
abstract void	onProviderDisabled(String provider) Called when the provider is disabled by the user.
abstract void	onProviderEnabled(String provider) Called when the provider is enabled by the user.
abstract void	onStatusChanged(String provider, int status, Bundle extras) Called when the provider status changes.

Figure 1.20 Methods of LocationListener interface [15]

1.6 Thesis Outline

This thesis has been divided into six chapters.

- Chapter 1 includes the introduction to GPS Based Vehicle Tracking System. It also covers its applications and challenges. This chapter also include various classes of android.
- Chapter 2 describes the different approaches for Vehicle Tracking System. It describes the various ways to get interface like LEDs, SMS, web or Android application. Also it explains hardware based systems using transmitters and VT & PIS.
- Chapter 3 presents the problem statement, objectives and methodology for developing android based Vehicle Tracking System.

- In chapter 4, architecture and working of android based Vehicle Tracking System has been proposed.
- In chapter 5 results of the system have been discussed.
- Chapter 6 gives the conclusion and the future scope of the work done in the thesis.

Chapter 2
Literature Review

2.1 Overview of Vehicle Tracking System

Vehicle Tracking System has been a part of Intelligent Transport System since 1970s but they were hardware based and uses sensors. The data mining techniques were used by software based solution which came in late 1990s. The reliability is increased with addition of traffic pattern. In 2000s GPS was started being used to trace the real time location of the vehicle. Figure 2.1 show various techniques followed for Vehicle Tracking System.

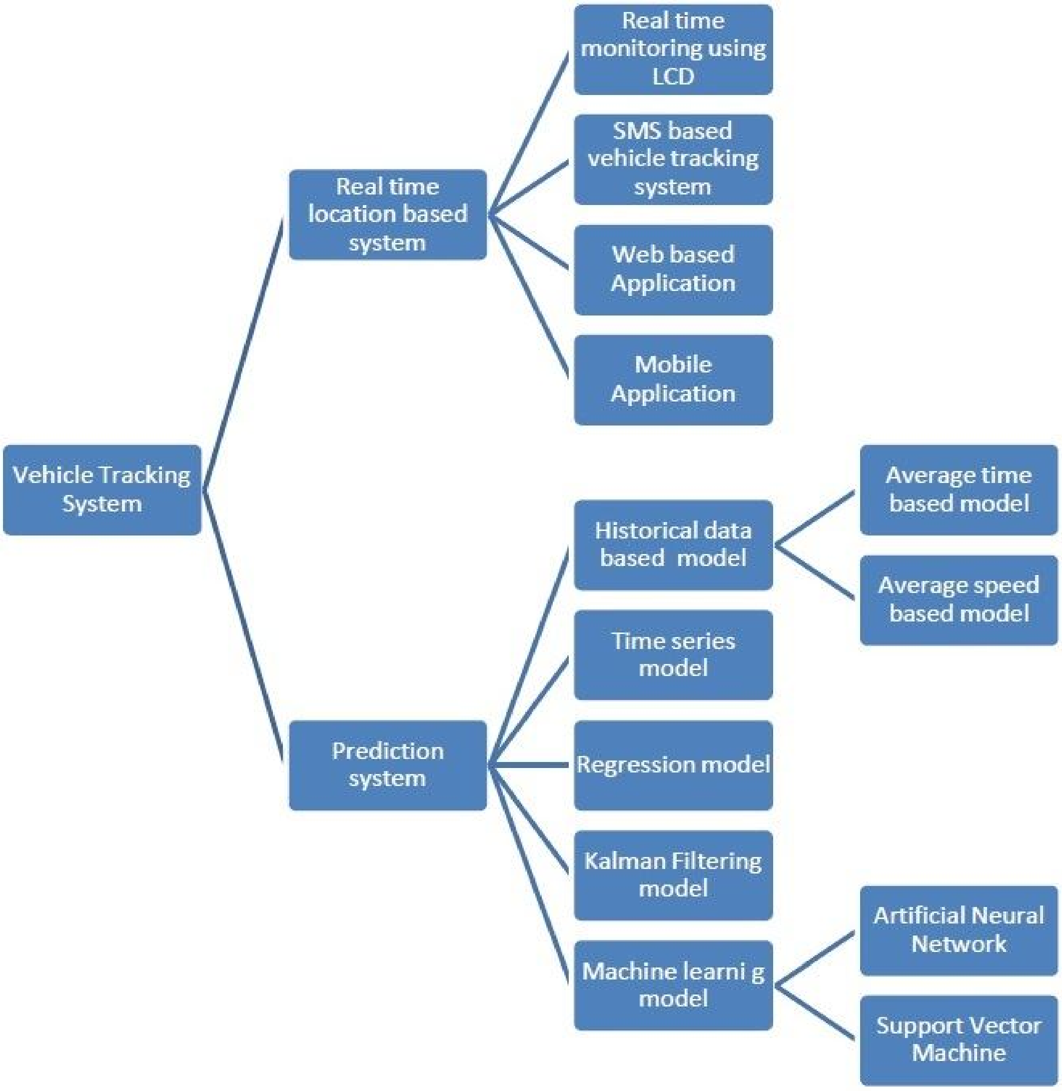

Figure 2.1 Techniques for Vehicle Tracking System

2.2 Real Time Location Based Systems

Real Time Location Based Systems are used to track the current position and speed of the vehicle. Figure 2.2 shows various real time location based systems.

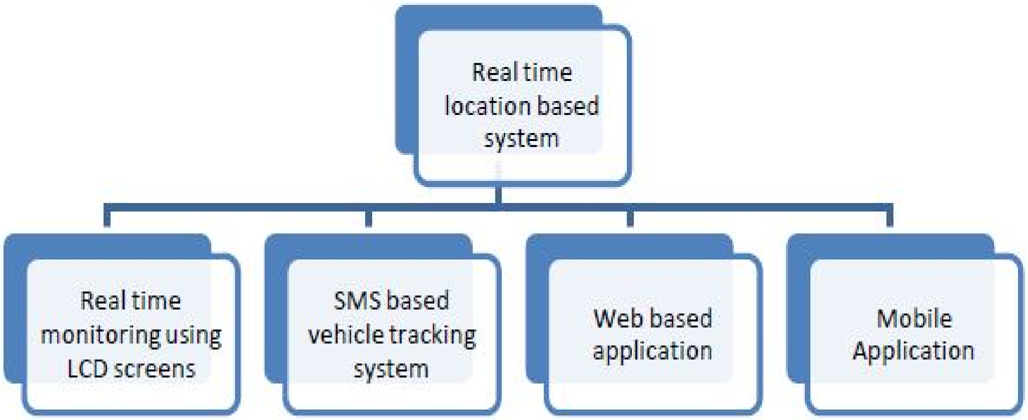

Figure 2.2 Various Real Time Location Based Systems

2.2.1 Real time vehicle monitoring system using LCD screens

The system has been proposed for the Mumbai city, India. The goal of the GPS based vehicle monitoring system is to give a product to the passengers of Mumbai city so that it can help them choose whether to wait for the vehicle or walk or take alternate transport. The system consists of microprocessor, LED's, RF transceiver, power source, battery. RF transceiver gets the signal that contains its GPS coordinates and it is installed over every bus. Then microprocessor processes the data. At every bus stop RF transceivers are installed to receive information regarding bus coordinates. When the transmitter enters the range of reception they will get active otherwise these will be passive circuits [1]. LCD screens are placed on the vehicle stops of the city, that displays the vehicle number and the routes of the vehicle that are near the stop along with the time it will take to reach the stop. Vehicle routes are illustrated on the map imposed on Plexiglas cover. LEDs are used to depict approximate geographic positions on this map where all stops are also marked. Following is the general framework of the system.

i. **Transmitter module**

GPS is mounted on each vehicle and is able identify its position when any three GPS satellites, measure distance to the receiver and triangulate. Data from all the satellites that triangulate is gathered to determine the position of the vehicle. Microcontroller gets the data from the GPS module using some serial link where processing is done to get the longitude and latitude. Transmitter polls a signal through a RF transmitter.

ii. **Receiver module**

Receiver unit at each vehicle stop contains RF receiver and microcontroller which receives the data through the RF receiver. To interpret the data, programmable logic is applied to process the data and then it is sent to the LCD screens.

iii. **Control module**

This is a centralized system where GPS data from all the vehicles is collected to plot the locations of the vehicles on the map. After processing the data from the receivers, LEDs are illuminated corresponding to the coordinates received. Figure 2.3 illustrates the functional block diagram of real time vehicle monitoring systems using LCD's.

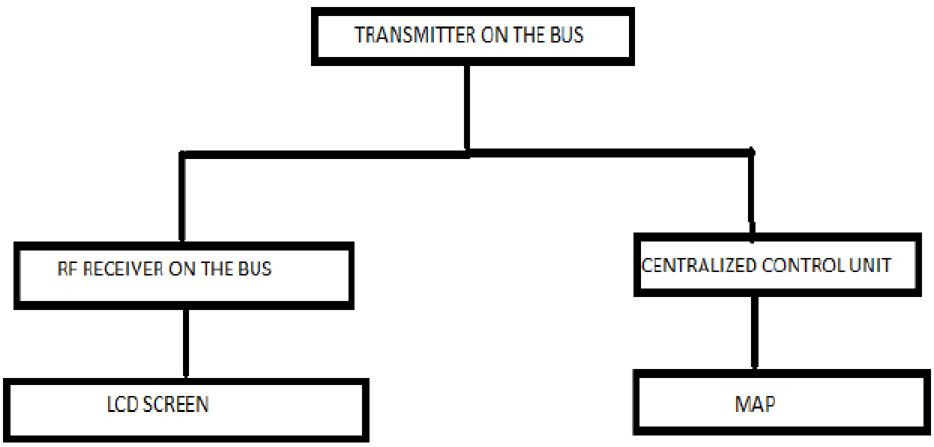

Figure 2.3 Functional block diagram of LCD based system [1]

2.2.2 SMS Based Vehicle Tracking System

SMS is used to transfer the vehicle location coordinates over the GSM networks. The GPS receiver in the vehicle calculates the latitude and longitude of the vehicle coordinates. Using SMS this information is directed to the central server over the GSM networks and this information is stowed in the database. Users can retrieve the information by sending the vehicle number and the route number. SMS is sent to user that contains the arrival time of the vehicle [2]. Following is the general framework of the system.

i. **Vehicle module**

GPS device is installed over the vehicle which is to be tracked. To make sure GPS antenna receives the signal, it is connected to the right jack. GPS device is turned on and tuned to receive signals from satellite. Device is capable of receiving longitude and latitude coordinates of the location. GPS data is regularly sent to the server using SMS service over the GSM networks. For this purpose, SIM (Subscriber Identity Module) card is inserted to connect to the GSM networks. Server analyses the data.

ii. **Server module**

It manages and analysis the GPS data and manage the queries from the users. Initially a setup message is sent to the vehicle module from the server to set the time interval at which GPS data is sent. SMS enabler acts as the interface between the vehicle module and the server. At server side PC suit is installed. From the vehicle module, SMS enabler receives the SMS and extracts the longitude and latitude values from the entire message and stores in the database.

iii. **User module**

Users send SMS to the server to retrieve real time location of the vehicle. Server side application processes the user's request and draws the location name from the database for the corresponding vehicle number and sends a response to the user. On server side initially user message is validated if it's in a valid format with valid vehicle number. After this, request is processed and receives a reply from the server. There are two ways to send SMS to a phone from a computer. Firstly, by connecting a GSM modem to a computer, by applying AT commands GSM modem sends the SMS. Secondly, by connecting computer to the SMS centre (SMSC) to send SMS using a protocol. SIM card is inserted in the GSM modem that is connected to the computer. Figure 2.4 illustrates the architecture of the SMS based real time vehicle location system in which the following are the sequence of steps.

i. Server module sends requests to vehicle module to set up the time interval.
ii. Vehicle module sets the time interval and sends the GPS information at regular intervals and stores in the database.
iii. User requests the current location from the server module.
iv. After verification, server module sends the location of the vehicle to the user.

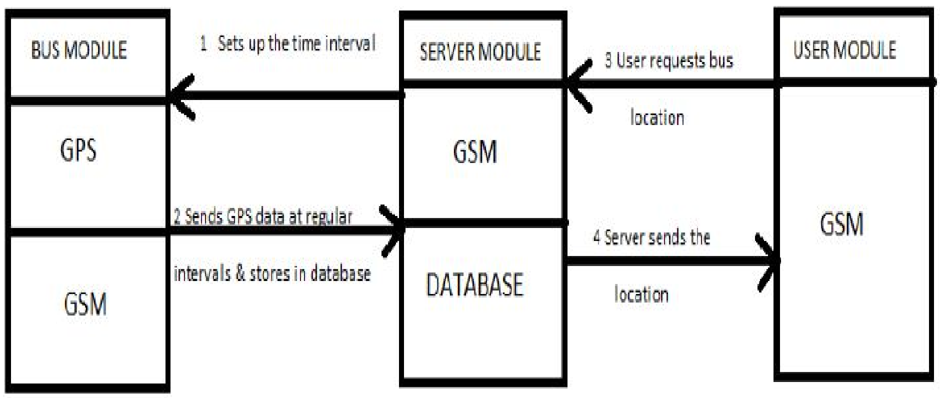

Figure 2.4 Architecture of SMS Based Real Time Vehicle Location System

2.2.3 Web based application

The system has been proposed for the Baghdad city, Iraq. Web based vehicle tracking system comprises of the central server system, vehicle- mounted tracking devices and the web based application. Web based application not only queries about the vehicles real time location but also can track the route on the map through the web application with the embedded Google map and interacts with the database server for the track details. Web based application also permits users with diverse operating system platforms to easily query the real time location of vehicle using the internet access [3]. The coordinates of the vehicle to be tracked is acquisitioned from the satellite using GPS receiver. Coordinates are sent to the server using GPRS service over GSM networks as a HTTP packet. Figure 2.5 shows the overview of web based vehicle tracking system. Following is the general framework of the system.

i. Web design

Overall functionality of the system is eased using various web application languages. Objectives of the web application are to define and manage all users' accounts and vehicles account. Receiving the GPS coordinates and storing them in database corresponding to the vehicle number. Replying to the client's request to get real time location of a particular vehicle number and displaying the route on the electronic map.

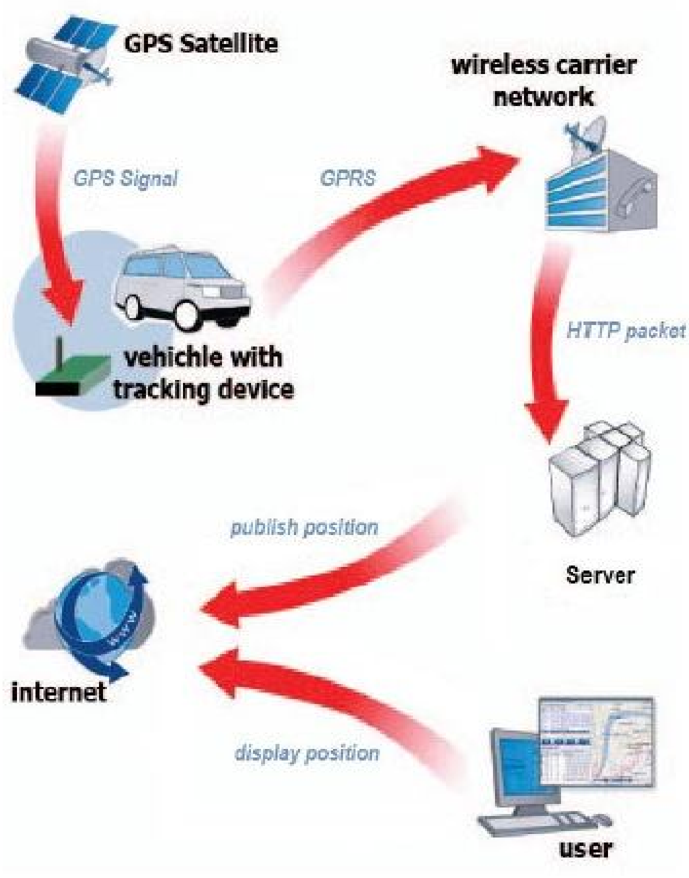

Figure 2.5 Overview of Web Based Vehicle Tracking System [3]

ii. Communication – vehicle and server

GPS device on tracked vehicle queries the coordinates, satellite responds with string of data with latitude and longitude coordinates, time and speed. Details are separated from the received data and the set of data along with the vehicle number are sent through the GPRS to the path defined in the URL using HTTP get method.

iii. Track Browsing

Map page consists of two drop down lists for selection of vehicle number, embedded Google map, route type. Through this page of application users and vehicle drivers can view track on map. By selecting route type, date and vehicle number, location will be presented. For static tracking all locations in the track table corresponding to the vehicle number, route and date will be displayed on the map using coloured markers and the lines connecting the markers from the start position to the last position. Whereas for the simulated tracking markers are added one by one at a time gap, say after 2 seconds of duration.

2.2.4 Mobile application

The system has been proposed for the Pune city, India. Bus tracking system using Android application uses the inbuilt GPS service provided by the Smartphone. Smartphone is mounted on each bus and to get its GPS coordinates. These coordinates are transferred to the central server. Users can retrieve information through android application where users select the bus number, route number and, receive the arrival time of the bus with respect to the user's current location instead of the bus stop. Maps are also used to graphically plot the bus and the user on the Google maps [4]. Getting real time location of the vehicle has become very easy with the rapid increase in the use of programmable smart phones with built in GPS facility. The massive influx in the use of Android has made the system very user friendly. It has been observed that by using Android phones the acceptance to the system has increased enormously. LBS are used by Android to get the real time location of the vehicle. Following is the general framework of the Vehicle tracking system using Android application.

i. Vehicle tracker system

This is the android application that is installed on the smartphone that is placed on the vehicle that is to be tracked. This application provides the driver with the GUI where driver starts the tracking by selecting the vehicle number, Route number and clicks on "BEGIN TRACKING". With this web service gets to know that the vehicle with that vehicle number and route number has begun its journey. Once the initiation is completed Vehicle tracking system sends the location data (GPS coordinates) of the vehicle to the web service periodically after 6 seconds. With these frequent submissions web service is able to plot the vehicle on the maps correctly on the mobile application or the web application.

ii. Web service

This service collects the information from all other systems like Vehicle tracker system, Web Application and Mobile Application. It keeps all the data and uses it in the most efficient way. It provides ability to for multiple platforms to work together within or around the system.

iii. Web application

This system provides the users with an interface where all the routes are present, with all the vehicles on those routes. The system allows the users to select the route and the vehicle on that route. With this Web service provides the users with the current location of the vehicle on the Google maps. The site is continuously updated by making AJAX calls to the web service to update the web map accordingly.

iv. Mobile application

This system is installed by the users on their programmable Smartphone from the Android market. The application provides with user friendly interface where user selects the route number and the vehicle number from the drop down lists. With this request at web service is generated along with GPS location of the passenger. Location of passenger is tracked using Location Based Services. In response to this web service sends the detailed information in which the passenger might be interested in. The detailed information includes the vehicle's current speed, Distance between the current locations of the vehicle and the passenger and the time it will take for the vehicle to reach the passenger's location, the estimated time to reach the final destination.

Reference points are plotted along the routes and these points are put in correspondence to the coordinates on the Google map. Sometimes the current GPS coordinates of the vehicle does not match with the coordinates on the same route of the Google map. In result of this when the vehicle's current location is mapped on Google map it might be shown on the road but near it. To overcome this problem reference points are plotted along the route in corresponds to the Google map. With this GPS location generated by the mobile devices are associated with its nearest reference point.

Some other tracking systems are as follows.

2.2.5 VT & PIS (Vehicle Tracking & Passenger Information System) by APSRTC

To encourage the usage of public transport instead of private vehicles, APSRTC (Andhra Pradesh State Road Transport Corporation) has implemented VT & PIS (Vehicle Tracking & Passenger Information System). It has been implemented to address the crucial problem of road congestion by making it comfortable, convenient, attractive and introducing value added services [12].

Functionality of VT & PIS

Following is the workflow of VT & PIS.

The buses are equipped with the GPS based vehicle tracking equipment. The vehicle tracking equipment captures the geographical location coordinates (longitude and latitude) and are transmitted to the Central Data Centre at regular interludes, through GPRS. In vehicle tracking unit SIM card will be installed. In the Data Centre an application is installed that calculates the vehicles current location and the estimated time of travel. Further the information is the transmitted to the LCD/LED concerned, through GPRS. Display Unit displays this information. Figure 2.6 shows the functionality of VT & PIS.

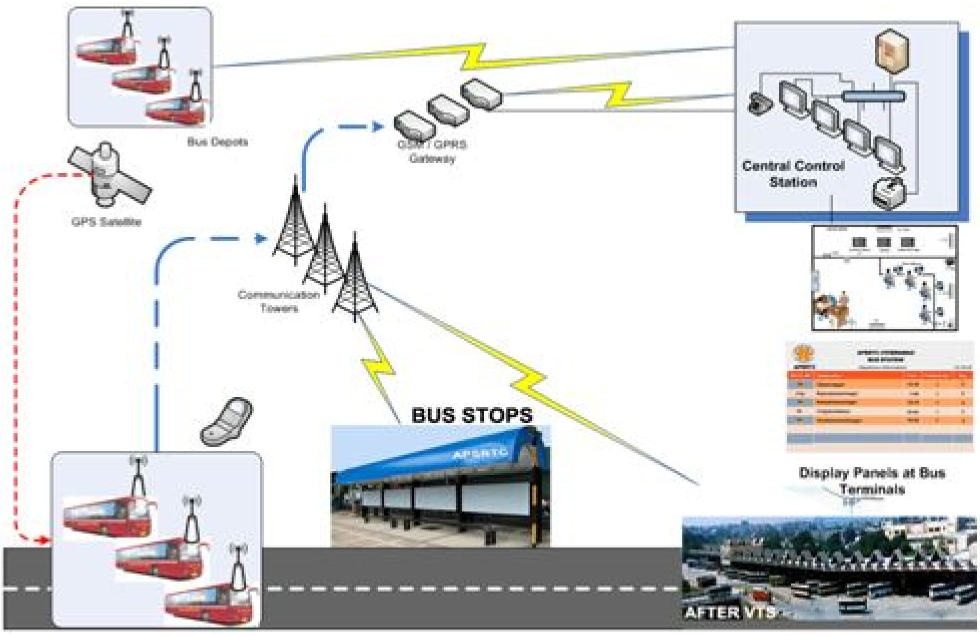

Figure 2.6 Functionality of VT & PIS [12]

Components of VT & PIS

Following are the major components of VT & PIS.

i. **Vehicle Tracking System (VTS) with real time connectivity**

GPS based VMU (vehicle Mounted Unit) form the core of the VTS. The vehicle location coordinates are recorded and then transferred to the central server through GSM/GPRS network by the VMU.

ii. PIS (Passenger Information System)

The real time information of the departure and arrival of the buses at the bus stops and bus stands are displayed by the LED/LCD display boards.

iii. Mobile Application and SMS facility

To check when the bus will reach the particular stop where the commuter is waiting, the commuters can check the APSRTC website or he can download the mobile application if he is having a smart phone. The commuters who doesn’t have smart phone, can send an SMS to the central call centre and a return SMS having details like route number, bus number and ETA (Estimated Time of Arrival) will be sent.

iv. In-bus display, stage announcements and recording

For displaying the information about the next stop, the current stop and voice announcement of the same, the destination boards and voice announcements are provided in the city buses. For recording the events in the buses, on-board cameras with recording facility is installed for later viewing and analysis purpose.

v. Data centre, Control centres

This section holds the key in the operation of VTS. In is the heart of the system. Every operation is controlled through data centre and control centre.

Chapter 3
Problem Statement

GPS based vehicle tracking system provides an opportunity to the traveller to track the current position of the vehicle and get the time it will take to reach the transit traveller. This system helps the passengers to make better travelling decision as passengers can decide whether to wait for the vehicle or not in accordance to the estimated arrival time. From this system passenger can also determine if the vehicle has been missed or yet to come. The system provides the basic information which is necessary to make day to day life more efficient. The system can be used for tracking the school/office vehicle.

Android Application that meets the user's need has become a key idea with the rapid growth and huge advances in Android. Android device has become a powerful device which provides much more than basic facilities. Therefore, aim is to provide an android application which locates the real time location of the vehicle on the map and gives the arrival time and distance of the vehicle to the passenger.

3.1 Objectives

For the proposed research work following objectives have been framed.

i. To analyse the working of existing android based Vehicle Tracking Systems.

ii. To propose the framework for android based Vehicle Tracking System to track the current position of the vehicle and the estimated arrival time.

iii. To develop the server side module for the vehicle to send the location coordinates (latitude and longitude) to the client (commuter).

iv. To develop the client side module for the commuters to receive the location coordinates from the vehicle and map the vehicle location onto Google map.

v. To test and validate the proposed GPS based vehicle tracking system.

3.2 Methodologies

To obtain the objectives as mentioned in section 3.1 following methodologies have been adopted.

i. Literature survey has been carried on the basis of existing approaches like systems based on real time Location Based Vehicle Tracking System.
ii. For GPS based system in android architecture entire developing environment with Android SDK and ADT tools are downloaded and installed.
iii. Vehicle module sends the location data to the commuter. The sent location data can be GPS coordinates or GSM coordinates. If the GPS coordinates are available, then GPS coordinates are sent by the server to the client otherwise the nearest GSM network location coordinates are sent.
iv. The commuter receives the location coordinates sent by vehicle and can see the location, estimated distance and arrival time of the vehicle on Google map.
v. Testing and validation is done at Thapar University which has been carried out as a part of the case study of the thesis.

Chapter 4

Implementation

4.1 Architecture and Working of the Proposed System

The idea of the proposed system is to track the vehicle and get the estimated arrival time and its estimated distance from the commuter. The architecture of the proposed plan is shown below in Figure 4.1.

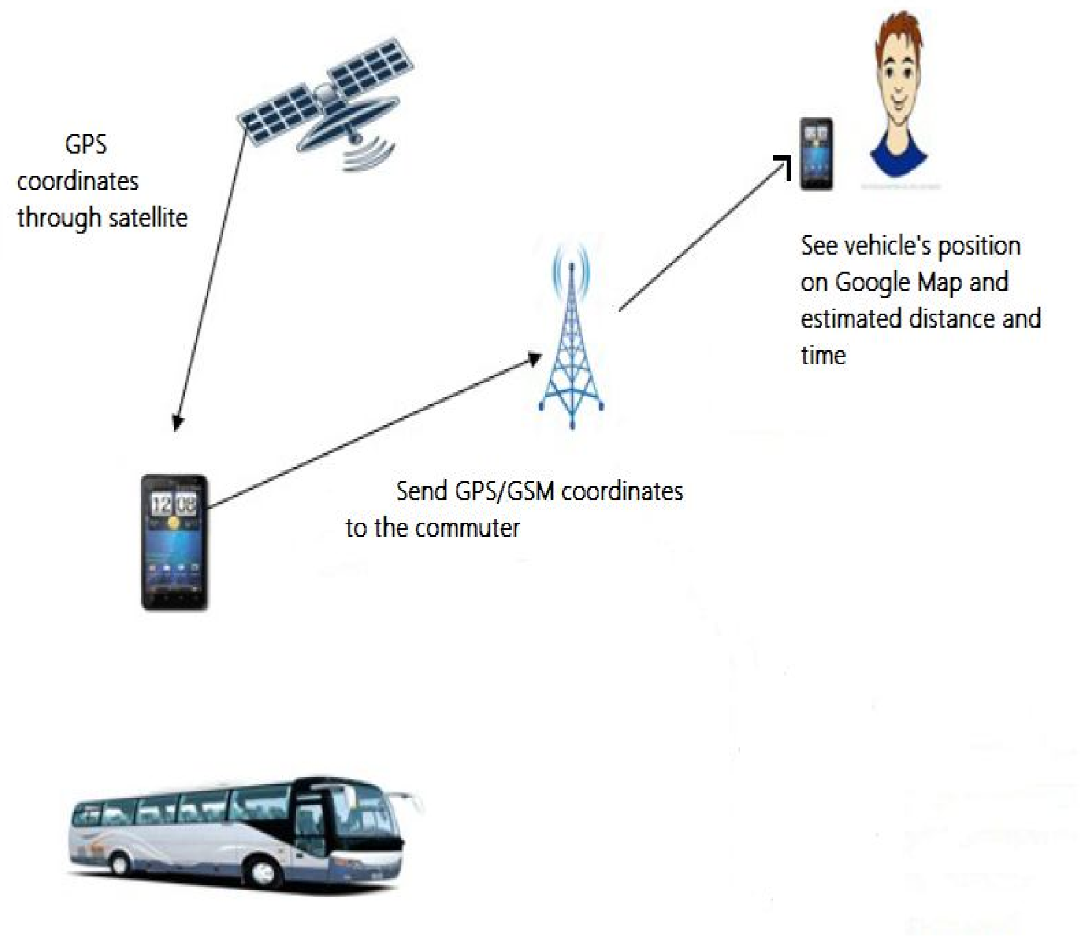

Figure 4.1 Architecture of the proposed system

The proposed system which consists of two types of modules client side module (commuters) and the server side module (vehicle) is built on the client server technology. Android application is to be developed for client side as well as server side.

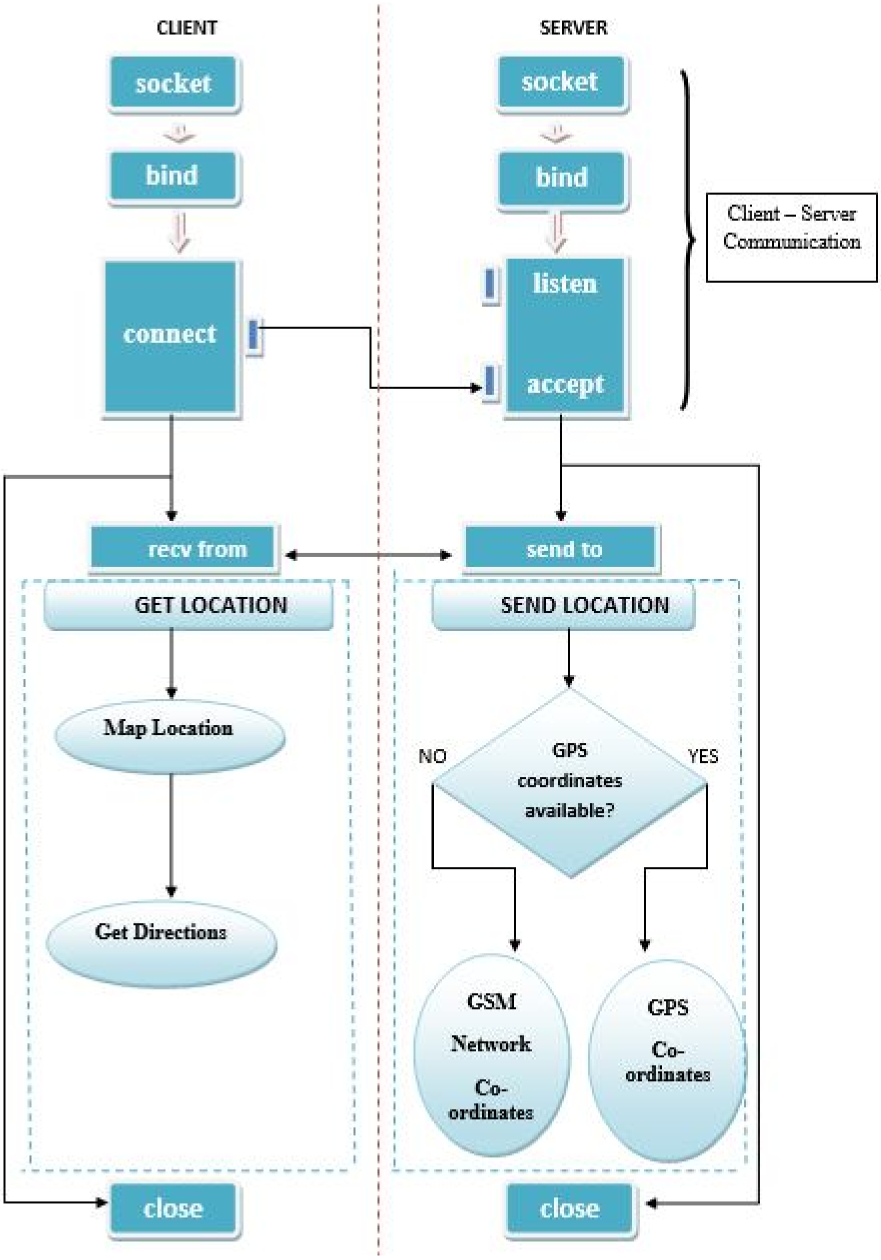

Figure 4.2 Working of the proposed system

The workflow for proposed system is illustrated in Figure 4.2. It is based on the client-server application in which the commuter acts as the client and the vehicle in which GPS is mounted acts as the server. Two applications are developed in which one acts as client side application and other acts as server side application. The commuter uses the client side application on his smart phone whereas the server side application is used by the vehicle in which GPS system is installed.

The proposed tracking system is categorized into two major modules as follows.

- **Module 1(Commuter module)**

This module is the client side of our android application which is installed on android based smart phone used by the commuter. It is used by the commuter to track the location of the vehicle it wants to travel in and also to get an idea of its estimated arrival time and distance. The basic requirement of this application is presence of internet, GPS and GPRS.

- **Module 2(Vehicle module)**

This module is the server side of our android application. On establishing successful connection with the client, it sends current real time location coordinates of the vehicle to the commuter from which the commuter can map the location on the Google map.

The working of the client server application developed is explained below.

i. Before connection establishment

The following Figure 4.3 shows how the port number is seen on the server side as shown in Figure 5.1.

```
private class SocketServerThread extends Thread {
    static final int SocketServerPORT = 8080;
    int count = 0;
    @Override
    public void run() {
        try {
            serverSocket = new ServerSocket(SocketServerPORT);
            MainActivity.this.runOnUiThread(() → {
                    info.setText("\n \n I'm waiting here : "
                            + serverSocket.getLocalPort());
            });

            while (true) {
                Socket socket = serverSocket.accept();
                count++;
                if(count==1)
                    message +="\n\n\n\n\n\n\n\n\n Client Connected...";
                else{}
                MainActivity.this.runOnUiThread(() → {
                        msg.setText(message);
                });
```

Figure 4.3 Port number code snippet

In Figure 4.3 getLocalPort() is the public method of type Integer of ServerSocket class. It returns the port number on which this socket is listening, *i.e.*, 8080. accept() is the public method of type Socket of ServerSocket class. It listens for a connection to be made to this socket and accepts it. The following Figure 4.4 illustrates how the IP address is seen on the server side as shown in Figure 5.1.

```
if (inetAddress.isSiteLocalAddress()) {
    ip += "\n \n\n \n SiteLocalAddress : "
            + inetAddress.getHostAddress() + "\n";
}
```

Figure 4.4 IP address code snippet

In Figure 4.4 inetAddress is an object of InetAddress class. isSiteLocalAddress() is the public method of type Boolean of InetAddress class. It checks if the InetAddress is a site local address. getHostAddress() is a public method of type String of InetAddress class. It returns the IP address string in textual appearance.

In order to establish the connection with the server, the client initiates the communication by creating the socket (IP + Port number) and then it sends the connect request to the server and waits for server's response. The server responds with the message "Client Connected..." as shown in Figure 5.3. This is how the connection is established between the server and the client. After successful connection establishment, the server and client are now ready to exchange the data.

ii. After connection establishment

The server (vehicle) sends its location co-ordinates to the client (commuter). If the GPS coordinates are available, then GPS co-ordinates are sent by the server to the client otherwise the nearest GSM network location co-ordinates are sent as shown in Figure 4.5. For the location co-ordinates to be available, the GPS of the smart phone must be turned on.

```
btnSendLoc.setOnClickListener((arg0) → {
        Location gpsLocation = appLocationService
                .getLocation(LocationManager.GPS_PROVIDER);
        Location nwLocation = appLocationService
                .getLocation(LocationManager.NETWORK_PROVIDER);

        if (gpsLocation != null) {
            latitude = gpsLocation.getLatitude();
            longitude = gpsLocation.getLongitude();
            Toast.makeText(getApplicationContext(),
                    "Your GPS Location Sent : \nLatitude: " + latitude
                            + "\nLongitude: " + longitude,
                    Toast.LENGTH_LONG).show();
            getMyLocationAddress();
        } else {
            if (nwLocation != null) {
                latitude = nwLocation.getLatitude();
                longitude = nwLocation.getLongitude();
                Toast.makeText(getApplicationContext(),
                        "Your NW Location Sent : \nLatitude: " + latitude
                                + "\nLongitude: " + longitude,
                        Toast.LENGTH_LONG).show();
                getMyLocationAddress();
            } else {
                Toast.makeText(getApplicationContext(),
                        "No Network available, Please check your GPS",
                        Toast.LENGTH_LONG).show();
            }
```

Figure 4.5 Fetching location coordinates code snippet

In Figure 4.5 *GPS_PROVIDER* is a constant of type String of LocationManager class. It shows the name of GPS location provider,

NETWORK_PROVIDER is also a constant of type String of LocationListener class. It shows the name of network location provider,

getLatitude() is the method of type Double of Location class. It gets the latitude in degrees,

getLongitude() is the method of type Double of Location class. It gets the longitude in degrees,

LENGTH_LONG is a constant of type Integer of Toast class. It shows the text notification or the view for an elongated period of time.

On receiving the location coordinates by the server, the client can map it on the Google map to get the estimated time and distance for the vehicle to reach the commuter. Client can also get the directions, *i.e.*, the shortest path between the vehicle and the commuter. This is how the commuter can track the vehicle it wishes to board on.

Chapter 5

Results and Discussion

5.1 Case Study - I

Case study has been carried on the proposed system to test its working. A case study has been carried in the campus of Thapar University, Patiala, India. The campus has an area of more than 250 acres. The university has a policy of go—green where the pollution free campus is maintained. No vehicles are allowed inside the campus. As the campus is distributed over large area it makes it difficult for the students to travel inside the campus. To overcome this problem and for the testing of the proposed plan auto rickshaws were installed in the campus. Auto rickshaws will act as the vehicle/server and students will act as the client/commuter.

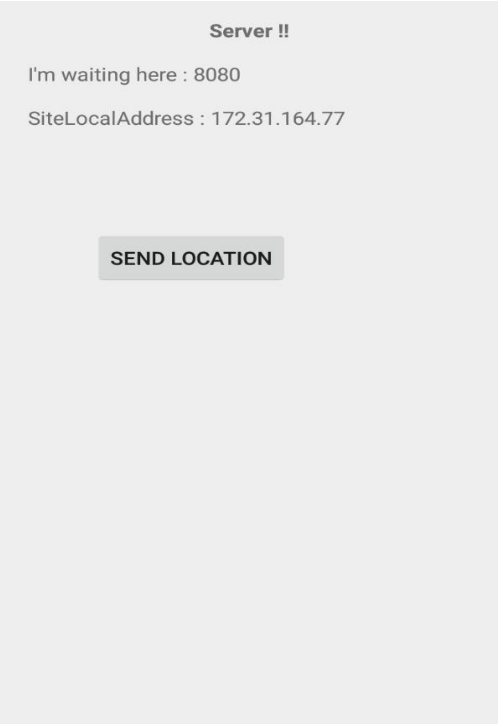

Figure 5.1 Server Module

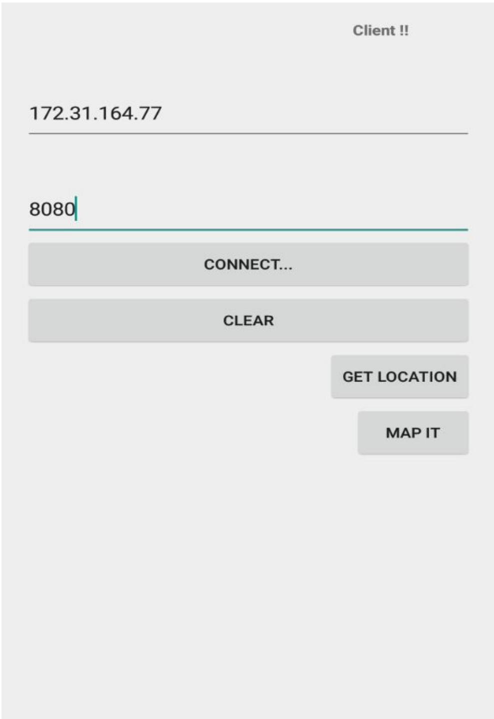

Figure 5.2 Client Module

Figure 5.1 is the server side application (auto rickshaw) which has IP (172.31.164.77) and port number (8080) for communication. Server/Vehicle is waiting for the client to communicate. Figure 5.2 is the client side application which tries to make connection to the server side by using the port number and IP address of the server.

In order to establish the connection with the server, the client initiates the communication by creating the socket (IP + Port number) and then it sends the connect() request to the server and waits for server's response.

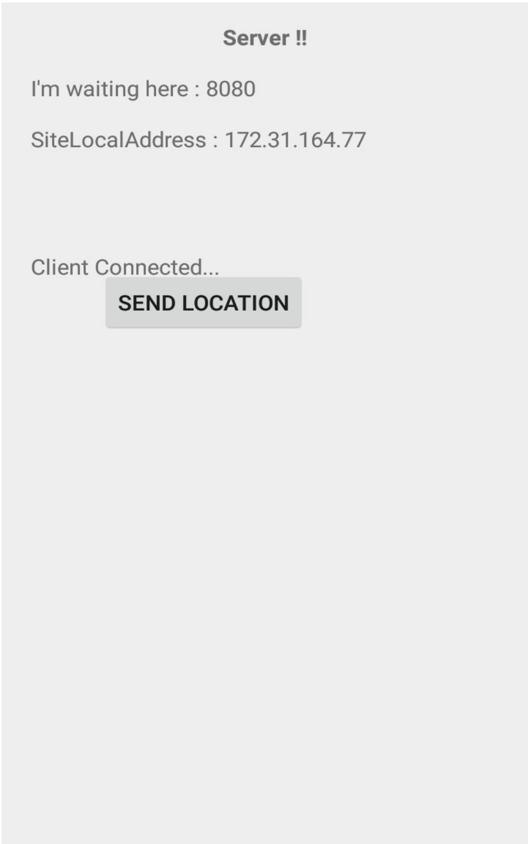

Figure 5.3 Client Server Connection established

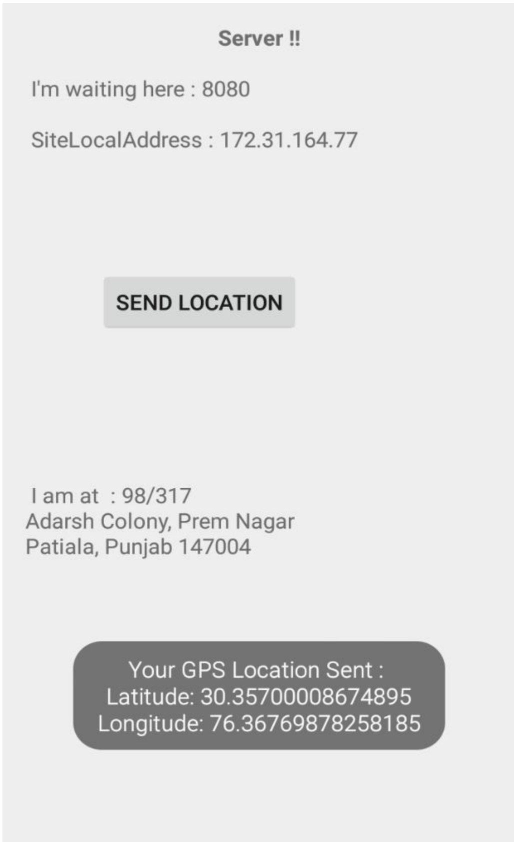

Figure 5.4 GPS coordinates of the vehicle sent to commuter

Figure 5.3 shows the connection establishment between the client and the server. In Figure 5.4 vehicle is sending its GPS coordinates to the commuter so that the commuter can trace its location using Google map.

The server responds with the message "Client Connected..." as shown in Figure 5.3. This is how the connection is established between the server and the client. After successful connection establishment, the server and client are now ready to exchange the data.

On click event of Send Location button as shown in Figure 5.4, the server (vehicle) sends its location coordinates to the client (commuter). If the GPS coordinates are available, then GPS coordinates are sent to the client by the server otherwise the nearest GSM network location coordinates are sent. For the location coordinates to be available, the GPS of the smart phone must be turned on.

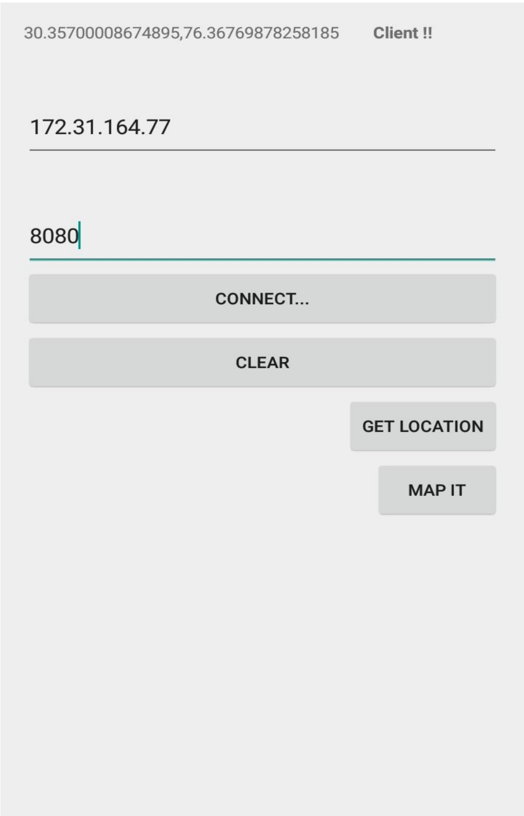

Figure 5.5 Client receiving GPS coordinates

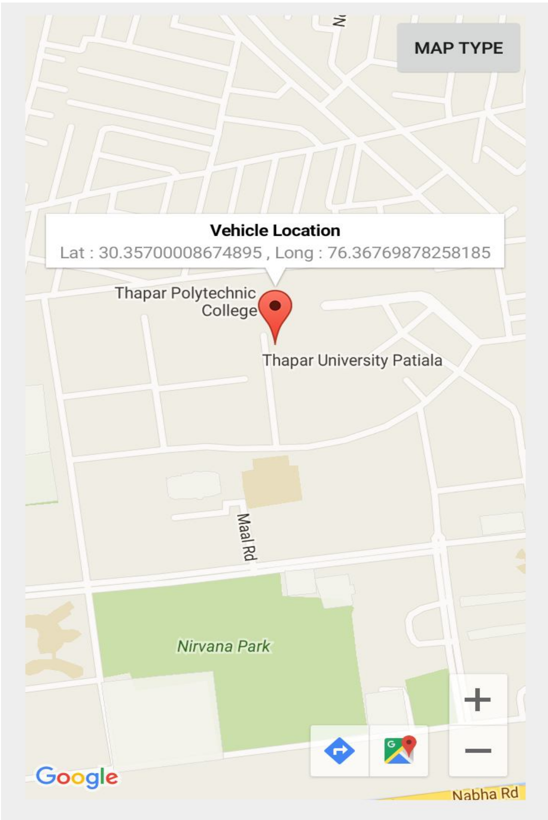

Figure 5.6 Vehicle's location on Google map

In Figure 5.5 the GPS co-ordinates sent by the vehicle are received by the commuter on the top left of its screen. In Figure 5.6 The commuter can map the vehicle's location in Google map corresponding to the co-ordinates it had received.

On click event of Get Location button the client receives location coordinates from server and then the client can map it on the Google map. Figure 5.6 shows that the vehicle is near Hostel I of Thapar University.

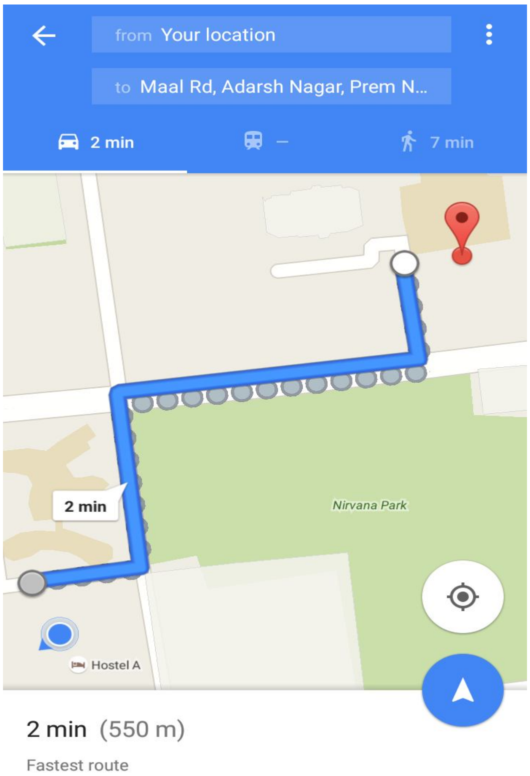

Figure 5.7 Estimated time and distance between vehicle and commuter

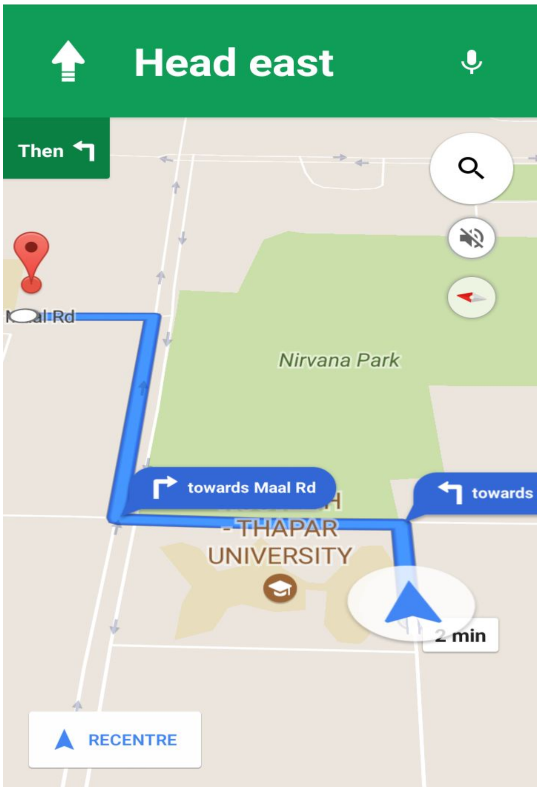

Figure 5.8 Directions

In Figure 5.7 the estimated time and distance is calculated between the vehicle and the commuter. In Figure 5.8 direction is shown.

While having seen the location of vehicle on Google map as shown in Figure 5.6, now the commuter can see the estimated time and distance that the vehicle will take to reach the commuter as shown in Figure 5.7. Client can also get the directions, *i.e.*, the shortest path between the vehicle and the commuter. This is how the commuter can track the vehicle it wishes to board on.

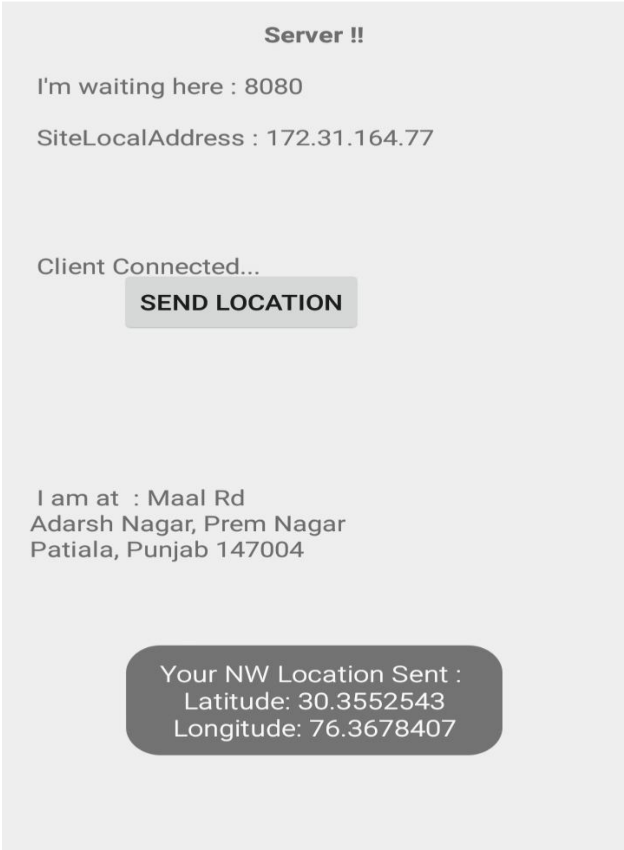

Figure 5.9 GSM Network coordinates of the vehicle sent to the commuter

In Figure 5.9 the vehicle sends its nearest GSM network co-ordinates to the commuter if the GPS coordinates are not available.

On click event of Send Location button as shown in Figure 5.9, the server (vehicle) sends its location coordinates to the client (commuter). If the GPS coordinates are available, then GPS coordinates are sent to the client by the server otherwise the nearest GSM network location co-ordinates are sent. For the location coordinates to be available, the GPS of the smart phone must be turned on.

5.2 Case Study – II

Another case study has been carried in the campus of Thapar University and also outside the University campus. This case study helps in monitoring the movement of the vehicle. The route followed by the vehicle on a particular day or during the day can be seen.

It has always been a challenge to manage employees in the field, especially for small businesses offering delivery or in-home services. With this system one can see the particular path followed by the vehicle on a particular day as shown in Figure 5.10.

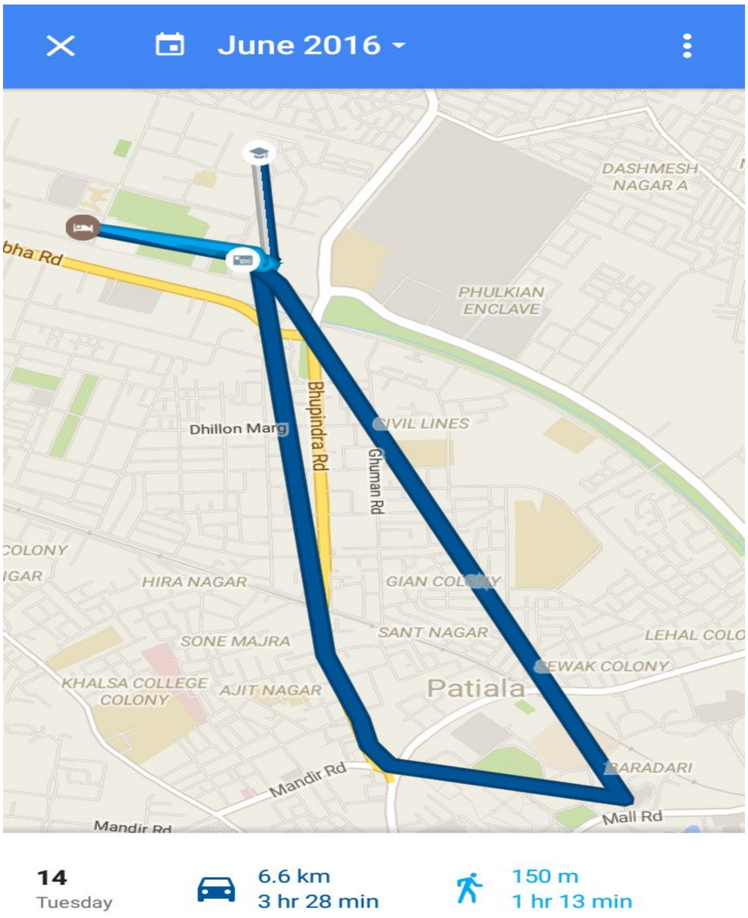

Figure 5.10 Route followed by vehicle outside the university

The route followed by the vehicle on 14th June, 2016 as shown in Figure 5.10 is as follows.

Hostel A of Thapar University → University academic area →22 number road → Mall → Thapar University

If someone runs a transportation business, this system can be handy for such users. If the owner has fixed a predetermined route for a particular vehicle and if the vehicle pilot is not following the desired route. This can easily be noticed, as the owner can monitor the route followed by the vehicle. The owner can also analyse the distance covered by the vehicle on daily basis. With this the owner can have an idea of average distance covered by the vehicle per day. Figure 5.11 shows the path followed by a vehicle in Thapar University campus.

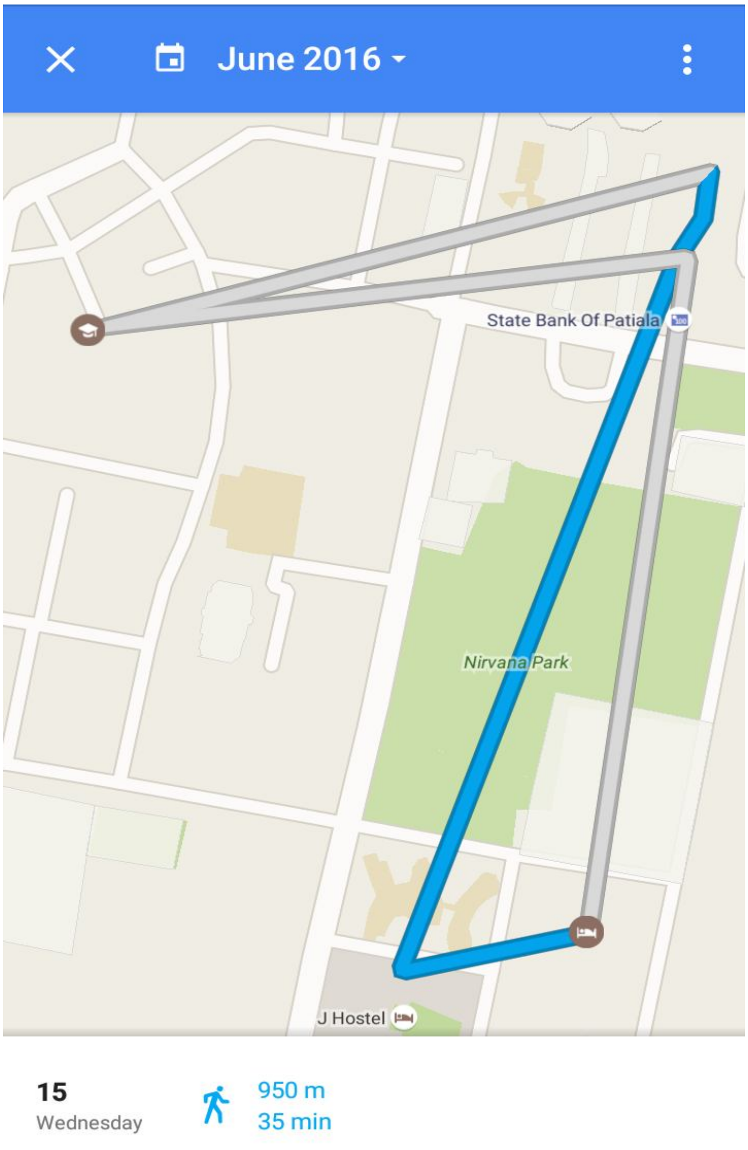

Figure 5.11 Route followed by a vehicle inside the university

The path followed by the vehicle on 15th June, 2016 as shown in Figure 5.11 in university campus is as follows.

Hostel A → University Academic Area→ Hostel J →Hostel A

Chapter 6
Conclusion and Future Scope

6.1 Conclusion

The system aims at reliable transportation in the city by providing passengers with the real time location of the vehicle. The system also gives the details of the estimated arrival time and distance of the vehicle to the passenger. In relation to the estimated arrival time the passengers can decide whether to wait for the vehicle or not. This helps in making better travelling decisions by the passengers. The basic idea behind the proposed system is to track the vehicle and get the arrival time and distance of the vehicle from the commuter.

After successful connection establishment between commuter and the server, the server (vehicle) sends its location coordinates to the client (commuter). If the GPS coordinates are available, then GPS coordinates are sent to the client by the server otherwise the nearest GSM network location coordinates are sent. For the location coordinates to be available, the GPS of the smart phone must be turned on. On receiving the location coordinates by the server, the client can map it on the Google map to get the estimated time and distance for the vehicle to reach the commuter. Client can also get the directions, *i.e.*, the shortest path between the vehicle and the commuter. A case study has been carried in which the system is deployed in the campus of Thapar University.

6.2 Limitations and Future Scope

Following are the limitations of the proposed system and the future scope to resolve these limitations.

- The working of the system is highly dependent on the coordinates provided by the GPS. Many android applications and many other important systems are working on the grounds of coordinates provided by the GPS. To overcome this problem Transmitters could also be deployed over the vehicle and the stops but this is very less efficient way.

- The working of the system entirely depends upon the availability of the internet. The passengers should have smartphones with the availability of the internet in it. To overcome this problem LCDs or LEDs could be deployed at each vehicle stop which displays the required information.

- The speed module of the vehicle has been developed. With speed the accuracy of estimated arrival time of the vehicle will increase. This module has to be attached to the system.

References

[1] G. Chheda, N. Gajra, M. Chhaya, J. Deshpande and S. Gharge. "Real time vehicle monitoring and passenger information system." International Journal of Soft Computing and Engineering (IJSCE), vol. 1, no. 6, pp. 34-58, 2012.

[2] R. Maruthi and C. Jayakumari. "SMS based bus tracking system using open source technologies." International Journal of Computer Applications, vol. 86, no. 9, pp. 44-46, 2014.

[3] A.K. Salim, and I.M. Idrees. "Design and implementation of web-based GPS-GPRS vehicle tracking system." International Journal of Science, Engineering and Computer Technology (IJCSET), vol. 3, no. 12, pp. 443-448, 2013.

[4] Y. Sardey, P. Deshmukh, P. Mandlik, S. Shelar and M. Nerkar. "A mobile application for bus information system and location tracking using client-server technology." International Journal of Emerging Technology and Advanced Engineering, vol. 4, no. 4, pp. 86-91, 2014.

[5] P. Verma and J.S. Bhatia. "Design and development of GPS-GSM based tracking system with Google map based monitoring." International Journal of Computer Science, Engineering and Applications, vol. 3, no. 3, pp. 33-40, 2013.

[6] R.C. Gadri, A. Chavan, R. Sonawane, and S. Kamble. "Land vehicle tracking application on android platform." International Journal of Engineering Research and Applications (IJERA), vol. 2, no. 3, pp. 1978-1982, 2012.

[7] S. Chandurkar, S. Mugade, S. Sinha and P. Borkar. "Implementation of real time bus monitoring and passenger information system." International Journal of Scientific and Research Publications, vol. 3, no. 5, pp. 1-5, 2013.

[8] Y. Kasture, S. Gandhi, S. Gundawar and A. Kulkarni. "Multi-tracking system for vehicle using GPS and GSM." International Journal of Research in Engineering and Technology, vol. 3, no. 3, pp. 127-130, 2014.

[9] D.K. Jain, and R. Goel. "GPS based low cost intelligent vehicle tracking system (IVTS)." International Conference on Traffic and Transportation Engineering, vol. 26, pp. 93-97, 2012.

[10] S.J. Lee, G. Tewolde and J. Kwon. "Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application." In Internet of Things (WF-IoT), IEEE World Forum on, pp. 353-358, 2014.

[11] G. Jemilda1, R. B. Krishnan, B. Johnson and G. L. Sangeeth. "Mobile application for college bus tracking." International Journal of Computer Science and Mobile Computing, vol. 4, no. 3, pp. 500-507, 2015.

[12] "Functionality of VT & PIS in APSRTC". Available at: http://telematicswire.net/vehicle-tracking-passenger-information-system-vt-pis-in-apsrtc-a-case-study/ [As accessed on 10th March, 2016]

[13] M.R.A. Fuad, and M. Drieberg. "Remote vehicle tracking system using GSM modem and Google map." In IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (CSUDET), pp. 15-19, 2013.

[14] I. M. Almomani, N. Y. Alkhalil, E. M. Ahmad and R. M. Jodeh. "Ubiquitous GPS vehicle tracking and management system." In Applied Electrical Engineering and Computing Technologies (AEECT), IEEE Jordan Conference on, pp. 1-6, 2011.

[15] "Android Tutorial" Available at: https://developer.android.com/ [As accessed on 6th September, 2015]

[16] S. Darekar, A. Chikane, R. Diwate, A. Deshmukh, and A. Shinde. "Tracking system using GPS and GSM: Practical Approach." International Journal of Scientific & Engineering Research, vol. 3, no. 5, pp. 1-4, 2012.

[17] Fleischer, Paul Benjamin, et al. "Design and development of GPS/GSM based vehicle tracking and alert system for commercial inter-city buses." In IEEE 4th International Conference on Adaptive Science & Technology (ICAST), pp. 1-6, 2012.

[18] M.A. Al Rashed, O. A. Oumar, and D. Singh. "A real time GSM/GPS based tracking system based on GSM mobile phone." In IEEE 2nd International Conference on Future Generation Communication Technologies (FGCT), pp. 65-68, 2013.

[19] "IRNSS Architecture" Available at: http://irnss.isro.gov.in/ [As accessed on 17th April, 2016]

[20] K. Sujatha, P.V. Nageswara Rao, K. J. Sruthi, and A. Arjuna Rao. "Design and development of android mobile based bus tracking system." In IEEE First International Conference on Networks & Soft Computing (ICNSC), pp. 231-235, 2014.

[21] S. Kotte and H. B. Yanamadala. "Advanced vehicle tracking system on Google earth using GPS and GSM." International Journal of Computer Trends and Technology (IJCTT), vol. 6, no. 3, pp. 130-133, 2013.

[22] "Android Tutorial" Available at: http://www.tutorialspoint.com/android/ [As accessed on 3rd October, 2015]

[23] D. S. Bhadane, P. B. Bharati, S. A. Shukla, M. D. Wani and K. K. Ambekar. "A Review on GSM and GPS Based Vehicle Tracking System." International Journal of Engineering Research and General Science, vol. 3, no. 2, pp. 351-353, 2015.

[24] A. Dhumal, A. Naikoji, Y. Patwa, M. Shilimkar and M. K. Nighot. "Survey Paper on Vehicle Tracking System using GPS and Android." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 3, no. 11, pp. 3762-3765, 2014.

[25] A. Agrawal. Indian Innovators. Mumbai: Jaico, 2015, pp. 95-105.

[26] "Android Studio" Available at: https://developer.android.com/studio/index.html [As accessed on 21st August, 2015]

List of Publications

Abhinav Bhargav and Parteek Bhatia, "GPS/GSM BASED VEHICLE TRACKING SYSTEM", in IEEE International Conference on Microelectronics, Circuits & Systems, Kolkata, India. July 9-10, 2016. [Accepted]

Video Link

A video describing the "Android Based Vehicle Tracking System" has been uploaded at www.youtube.com. The URL of the same is as follows:

https://www.youtube.com/watch?v=AZbFTHu6wgE&feature=youtu.be