PROJECT REPORT

ON

Vehicle Tracking System Using GPS Sensor and GPRS Modem

Submitted for the partial fulfillment of the degree of

Bachelor of Technology

in

Computer Science Engineering

By

DEVANSHIKA SINGH PARIHAR
Roll No. – 101317

Under the Supervision of

Mr. Suman Saha
Mr. Kapil Saini

May 2014

Jaypee University of Information Technology

Waknaghat, Solan – 173234, Himachal pradesh

CERTIFICATE

This is to certify that the work titled **"Vehicle Tracking System using GPS Sensor and GPRS modem"** submitted by **DEVANSHIKA SINGH PARIHAR** for the partial fulfillment of the award of degree of Bachelor of Technology in Computer Science & Engineering, Jaypee University of Information Technology, Waknaghat has been Carried out under my **supervision.** This work has not been submitted partially or wholly to any other University or institute for the award of this or any other degree or diploma.

Signature of Guide: …………………………………………

Name of Guide: …………………………………………

Designation: …………………………………………

Date: ………………………………..............

ACKNOWLEDGEMENTS

I would like to express my deep gratitude and heartfelt thanks to my supervisor Mr. Suman Saha, for his guidance, encouragement and gracious support throughout my project work. His valuable suggestion effusive co-operation and encouraging interactions were a great driving force for me to carry out this thesis work.

I would like to express my sincere to all my fellow B.Tech friends who extended their helping hands towards me in various ways during my short tenure at JUIT, Waknaghat for their friendly co-operation during this project work.

I would also like to thank Noopur Singh Sisodiya, MITS for his co-operation during this project work.

I am thankful to all faculty members who were approachable all the time for any help and suggestion and for their affection and career guidance. I am thankful to Department of CSE and IT, JUIT for providing me all the facilities needed for completion of my project.

Finally, I would like to thanks my parents for their moral support and constant encouragement towards the fulfillment of degree.

Signature of the students: ……………………………

Name of Students: ……………………………

Date: ……………………………

ABSTRACT

Now a days the normal view of the communication is extending, it is not only the people who use the telecommunication and internet technologies to communicate but the machines around us also have started to communicate with each other, this lead to machine to machine communication.

In this thesis the Vehicle tracking system is a total security protection to find out the location of vehicle. By using the latest GSM & GPS technology to protect and monitor our car, truck, boat (moveable asset) virtually anywhere and then locate it to within a few meter. In this thesis the coding is developed in Java & Android software's and this coding can be used for vehicle tracking instruments. In the thesis the software is developed to track the vehicle. Vehicle data has been stored in the database table by which we can search the vehicle. SQLLite database is used here to store the vehicle data and SQLLite SQL queries used to access and display the data in different User Interfaces of this project.

TABLE OF CONTENTS

Chapter 1

1 INTRODUCTION

1.1 GENERAL

Even though the various technologies that have been introduced in recent years to discourage car thefts and tracking it, It was reported that as many as cars were stolen yearly in the world. As per as National Crime Information Center (NCIC), in 2006, 1,192,809 motor vehicles were reported stolen, the losses were 7.9$ billions.

There are several security and tracking systems designed to assist corporations with huge number of vehicles and several usage purposes. Fleet management system can diminish the cost and effort of employees to finish road assignments within a minimal time. Besides, assignments can be scheduled in advanced based on current vehicles location. Therefore, central fleet management is essential to large enterprises to meet the varying requirements of customers and to improve the productivity.

However, there are still some security gaps where these technologies don’t prevent a vehicle from theft, don’t assist to recover it and don’t allow the users to know the status of their vehicles. They can’t permit the owner to connect with the vehicle online, even if the owner is confident that his vehicle was stolen.

The planned security system in this paper is designed to monitor and track vehicles that are used by certain party for specific purposes, also to stop the vehicle if stolen and to track it online for recovery, this system is an integration for several modern embedded and communication technologies. This can provide position and time information of any place on Earth, in the space-based global navigation satellite system the GPS is commonly used. The information of location conveyed by GPS systems can be imagined using Google Earth. GSM and SMS technology is a common feature in wireless data transporting with all mobile network service providers. Utilization of SMS technology has become popular because it is an inexpensive, convenient and accessible way of transferring and receiving data with high reliability.

Fig. 1 shows the proposed system which consists of: GPS receiver, GSM modem, and embedded controller. Users of this application can monitor the location graphically on Google Earth and they can stop any vehicle of the fleet if it was stolen & they also can view other relevant information of each vehicle in the fleet.

As shown in Fig. 1, when the vehicle is in motion, the client receives a confirmation SMS of running information at that time. If this is criminal operation or any intruders try to run the vehicle, the owner can send SMS to switch off the vehicle. Later, the system will check the mobile number for received message, to confirm that the contact number could access the security system & if the contact number is legal the system will turn off the vehicle. If the owner needs to track the vehicle, he/she have to send SMS contains special code, after that he/she will receive a SMS containing the GPS coordinates of the car, the SMS updating its content every predetermined period. Also the car owner can connect another GSM modem with laptop to track the vehicle immediately using Google Earth. The implemented tracking and security system can be used to monitor various parameters related to safety; anti-theft, emergency services and engine stall. The paper shows an implementation of several modern technologies to achieve a desirable goal of fleet monitoring and management.

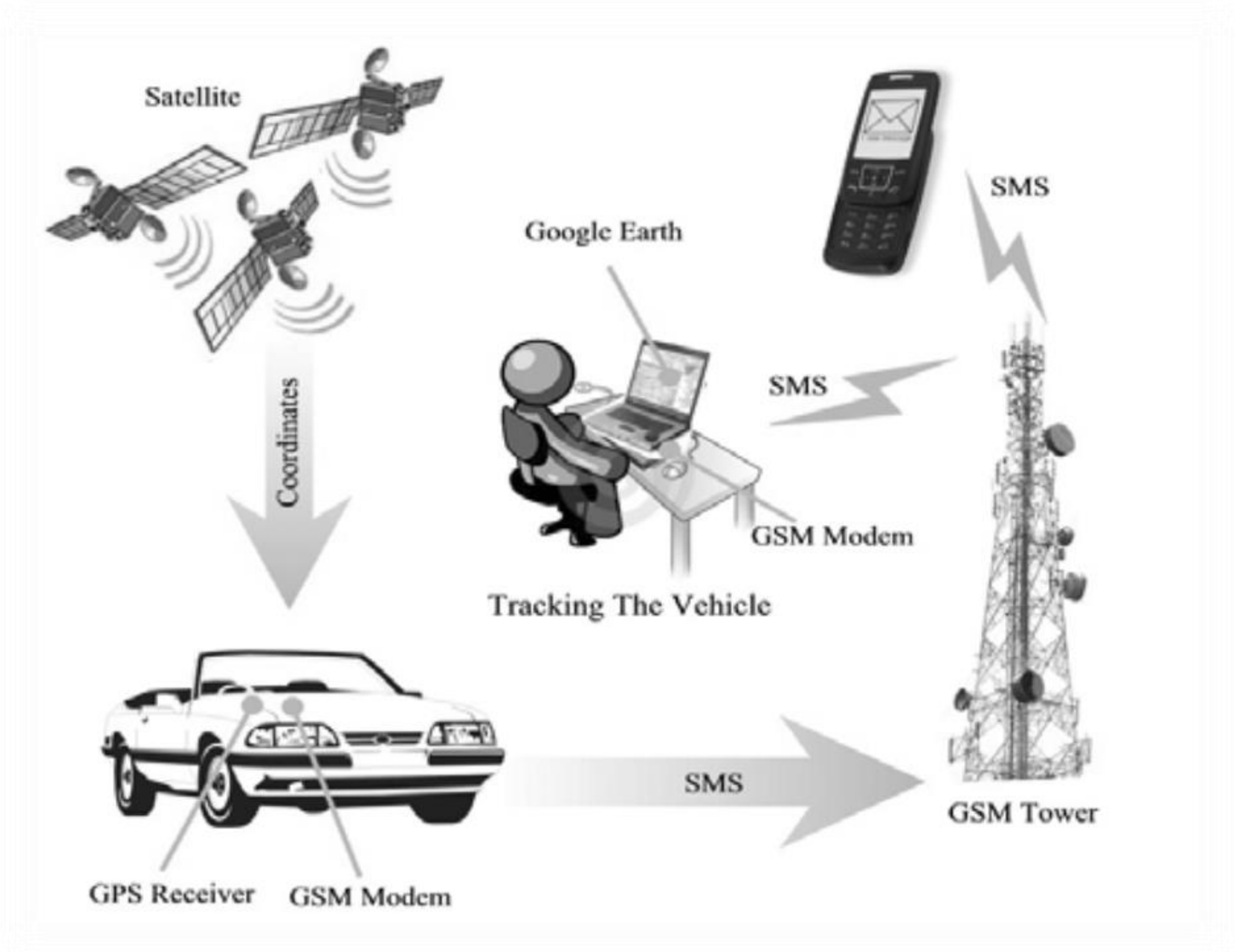


Figure 1: The Block Diagram of Security system

1.2 PURPOSE OF STUDY

The purpose of this study is to know about perimeters which are used in vehicle tracking device and how to use them in real situation. Generally this study needs GPS, Internet and google earth to track the devices.

1.3 APPLICATIONS OF STUDY

- Categorization of vehicle type.
- To find out current location of vehicle.
- To find out freight assignment position.

1.4 ADVANTAGES OF STUDY

- It provides more security than other system.
- From the remote place we can access the System.
- By this we can position the vehicle in exact place.

1.5 PROBLEM STATEMENT

With the increase in number of the vehicles plying in the street, the problems of traffic management, theft of the vehicles and trouble in navigation have increased in enormous fashion. These problems have been rocketing with time. To get rid of these problems, cost effective technologies such as GPS, GPRS and Internet for the tracking of vehicles or for other M2M communication system is quite justifiable.

To resolve such problems, a system is developed using GPS and GSM technologies and an application is introduced in this research work.

Various problems that we face:

1. In critical condition (when vehicle is stolen), one is confused what to do

2. If one has something expensive and he wants to check it regularly

3. To find the shortest path available

All these problems are overcome by the system.

This system has Global Positioning System (GPS) which will receive the coordinates from the satellites among other critical information. Tracking system is very important in modern world. This can be useful in soldier monitoring, tracking of the theft vehicle and various other applications. The system is microcontroller based that consists of a global positioning system (GPS) and global system for mobile communication (GSM). This project uses only one GPS device and a two way communication process is achieved using a GSM modem. GSM modem, provided with a SIM card uses the same communication process as we are using in regular phone.

For Example

- There are dumpers which need to be tracked.
- Tracking includes-
    - Checking whether the dumper started its operation or not.
    - No of trips made by each employee in a particular shift.
    - Total no of trips in particular shift of each dumper.
    - Total no of trips of each dumper in whole day, moth or year.
    - Total no of trips by all drivers in a given day, month, year etc.

All of the above requirement should be automated to decrease the manual errors and implement a good fleet management system.

1.6 MOTIVATION

The speed limitation of software approaches for vehicle tracking system implementation motivates us to look for alternative methods. For vehicle tracking, there are two ways to improve real-time operations. One is to develop a new tracking algorithm that is simpler and faster, thus requiring less computational time.

To minimize and eliminate congestion based traffic path during transit by mapping the vehicle to traffic in multiple paths. To provide a safe and intelligent solution of identifying optimal paths to the vehicle driver. To provide traffic video information this is useful for driver to know particular junction during traffic congestion. To identify the landmark to the vehicle driver or to a new entrant in the city. To enable continuous communication (conference) with other vehicle through wire-free networks in non-line of-sight.

Motivated by the limitations of the software approaches for vehicle tracking, there have been recent research interests in hardware implementation of algorithms used in different areas of ITS applications. The main initiative of hardware implementation of tracking algorithm is to minimize the computational time and thus to improve real-time operation of vehicle tracking, but directly proceeding to hardware implementation of a tracking algorithm may make the overall design process impractical considering other factors.

1.7 THESIS ORGANIZATION

In this thesis first chapter include introduction about vehicle tracking, purpose, application & advantage of study, it also include problem statement and motivation. Second chapter include literature review related to vehicle tracking. Third chapter is about implementation which includes theory and coding (Java, Android). Fourth chapter include flow chart of the process. Fifth chapter include Project Plan. Sixth chapter include Conclusion of the thesis. Last chapter include bibliography.

Chapter2

2 LITERATURE REVIEW

***Kamal Jain et. al. presented a paper about GPS Based Low Cost Intelligent Vehicle Tracking System*:** The paper defines the application of Global positioning systems in IVTS systems. After that a critical GPS based low cost IVTS architecture has been defined. The first part of the paper is about the necessity and the basic architecture of a general GPS based IVTS systems. The three IVTS units (i.e. In-Vehicle unit, Communication link and Base station) are described individually. Further the paper defines how and why cost plays a major role in spreading an IVTS systems. The modifications that should be carried out in the distinct units to gain a low cost GPS which is based on IVTS system which suits the present dynamic urban environment are described. The changes carried out in IVTS units also help in finding a reliable and accurate planimetric solution in case of poor visibility of the GPS satellites, which usually is common in urban environment. The GPS based low cost intelligent vehicle tracking system which can be successfully designed and applied in the urban situation of a developing country i.e. India. If this is implemented in a well-planned way, it will bring significant revolutionary improvement in the Indian transportation industry. The Information Superhighway in India is dynamically heading towards a new dimension in Geometrics Industry. Geometrics is the interaction of geo related sciences that is mainly including Surveying (Land/Geodetic), GPS, GIS, Photogrammetric, Remote Sensing. With advancement in Information

Technology these sciences have emerged as real growth engine of the nation Economy. Implementation of GPS in vehicles can certainly bring a revolutionary impact in transportation science in a developing country like India where there is an extremely high urban as well as rural vehicular transition every day. The low cost IVTS combined with available high performance processors can provide a highly accurate yet inexpensive vehicle tracking and navigation solution which is the need of the hour in fast moving urban cities of India. Arial ion wide integrated business plan for including automobile companies and GPS system providers is desired to bring this revolution.

Montaser N. Ramadan et. al. presented a paper about Intelligent Anti-Theft and Tracking System for Automobiles: As per as to this paper an efficient automotive security system is applied for anti-theft using an embedded system engaged with a Global Positioning System and a Global System of Mobile. The customer interacts through this system with vehicles and governs their current locations and position using Google Earth very quickly. The operator can track the location of targeted vehicles on Google Earth by this software. By the use of GPS locator, the target present position is determined and sent, along with numerous parameters received by vehicle's data port which is given via Short Message Service through Global System of Mobile networks to a GSM modem that is attached to PC or laptop. The Global Positioning System coordinates are modified using a discrete Kalman filter. To protect the vehicle, the user of a group of operators can turn off any vehicle of the fleet if any burglars try to run it by hindering the gas feeding line. It is very safe and efficient to report emergency situations as crash reporting or engine failure.

This paper explains a low-cost vehicle tracking and monitoring system. The application included a transmitting module which comprises an embedded system to combine GPS and GSM devices to recover location and vehicle status information and send it to the other stationary module and the second portion is the delivery module which collects the transmitted information by SMS and process it to a companionable format to Google Earth to sight the position and vehicle status online.

Rajiv Kumar et. al. presented a paper about Efficiency Related Parameters in GPS based Vehicle Tracking System Working for an Organization: Tracking vehicles can be the outcome of the obligation to keep track of a fleet or even simply to keep way of personal vehicles. Now a days where vehicle thefts are very common, the GPS vehicle tracking system can come in helpful almost all over. It is a fast growing service which enables field where role is done from the side of numerous national and international service providers. This paper involves the detail of parameters by which service suppliers can deliver the efficient services to spread the overall benefits of GPS vehicle tracking system. Performance of system is based on constraints related with system or environment. Some constraints are known and intricate in the distribution of facilities of implemented systems. Working of GPS based vehicle tracking systems is nonstop improving by considering the known constraints. Best performance of the systems is still awaited and some unproductive situations happened during working with the systems. The substantial additional factors which should be investigated and kept in mind when GPS based vehicle tracking systems are being installed. The adeptness of GPS based vehicle tracking systems can be upgraded by considering anticipated parameters.

This paper is about proposing some additional constraints which are crucial with respect in effectiveness of GPS based vehicle tracking systems.

1. If the organization is going to organize any GPS based system than user should be taught so that all the features of organized system are identified in such a manner that they can consume the system to generate fruitful scenario.

2. The deployment of GPS based vehicle tracking system must be according the structure of working building. GPS does not work in basement. The floor of implementation must be according guidelines.

3. The proximity of electric wires must be analyzed during deployment of the GPS based vehicle tracking systems so that the best performance got.

4. In vehicle tracking, coverage of speed for vehicle is also a measurable factor in fetching the information through GPS based vehicle tracking systems, therefore maximum satellite support should inherent for fetching the efficient information.

In this research related effort we have analyzed the existing support parameters for GPS based vehicle tracking systems. In our survey the questions were asked related with existence of new parameters and the views of technical persons about existing parameters crosschecked. The limitation and advantages of these parameters also discussed with technical persons. The authors brought out required changes in the set of existing parameters for improving the working of GPS based vehicle tracking systems. This paper is survey based research attempt extending the range of efficiency related parameters for service providers. The best performance can be achieved with the help of crucial parameters proposed by the authors for improving the efficiency of GPS based vehicle tracking systems.

<u>Baburao Kodavati et. al. presented a paper about GSM AND GPS Based Vehicle Location & Tracking System:</u> According to this paper, a vehicle tracking system combines the installation of an electronic device in a fleet of vehicle or vehicles, with purpose-designed computer software to allow the owner or a third person/party to track the vehicle's position, gathering data in the process. Modern vehicle tracking systems frequently use Global Positioning System technology for localizing the vehicle, but other types of automatic vehicle positioning technology can also be used. The vehicle information can be viewed on electronic maps via Internet or specialized software. Mainly they are easy to steal, and the ordinary motorist has very little knowledge of what it is all about. For avoiding this kind of steal we are going to implement a system that provides more security to the vehicle. In the preceding system security lock and alarm is executed in a car. If a robber can break & open the lock then it becomes easy for the robber to rob the car, and in old security system if the car is robbed then it is out of the owner control. Owner/user doesn’t have any wakefulness about the current position of the vehicle. In the Modern Proposed Systems the attachment of RF transmitter with the vehicles which has its own ID. The data from this will be continuously transmitted to the RF receiver linked to the microcontroller. This GPS will be locating the location of vehicle and convey that data to the microcontroller. If the RF receiver not getting signal from the transmitting unit, receiver unit direct the signal to the microcontroller, from that we can find the theft. If the vehicle is robbed, it automatically sends position of the vehicle to its owner/third party as a SMS over GSM modem. It will be a much simpler and low cost method linked to others. If a password massage like SMS is sent by the owner/third party, it automatically turned off the vehicle engine. Vehicle tracking system is becoming progressively important in large cities and it is more protected than other systems. At these days vehicle robbing is continuously increasing and with this we can have a good control in it. Vehicle can be stopped by simply with a simple password SMS. Since, in present days the cost of the vehicles are growing they will not step back to pay for it. This setup can be made more communicating by adding a display to show some basic information about the vehicles and also added emergency numbers which can be used in case of emergency. Advancement of this setup is very easy which makes it open to future desires without the requirement of renewal everything from scratch, which also makes it more capable.

Chapter 3

3 IMPLEMENTATION

3.1 ANDROID

Automated Numeration of Data Realized by Optimized Image Detection (ANDROID) is an operating system for devices such as Smartphone and tablet computers. And this is established by the Open Handset Alliance run by Google. Android comprises of a kernel built on the Linux kernel, with middleware, libraries and APIs inscribed in C and application software running on an application basis which comprises Java-compatible libraries on the basis of Apache Harmony. Android uses the Dalvik virtual mechanism with just-in-time compilation to run Dalvik dex-code (Dalvik Executable), which is usually interpreted from Java byte code.

3.1.1 ADT Plug-in for Eclipse

Android Development Tools is a plug-in for the Eclipse IDE that is designed to give you an influential, integrated environment in which to form Android applications. ADT spreads the capabilities of Eclipse to let you quickly set up modern Android projects, create an application UI, add constituents based on the Android Framework API, debug your requests using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute your applications. Rising in Eclipse with ADT is extremely recommended and is the fastest way to get started.

3.1.2 Android Open Source Project

The Android Open Source Project is controlled by Google, and is tasked with the repairs and development of Android. The aim of the Android Open Source Project is to form a successful real-world product that grows the mobile information for end operators. It also maintains the Program of Android Compatibility, defining an Android compatible device as one that can run any application written by third-party creators using the Android NDK and SDK, to avoid incompatible Android applications.

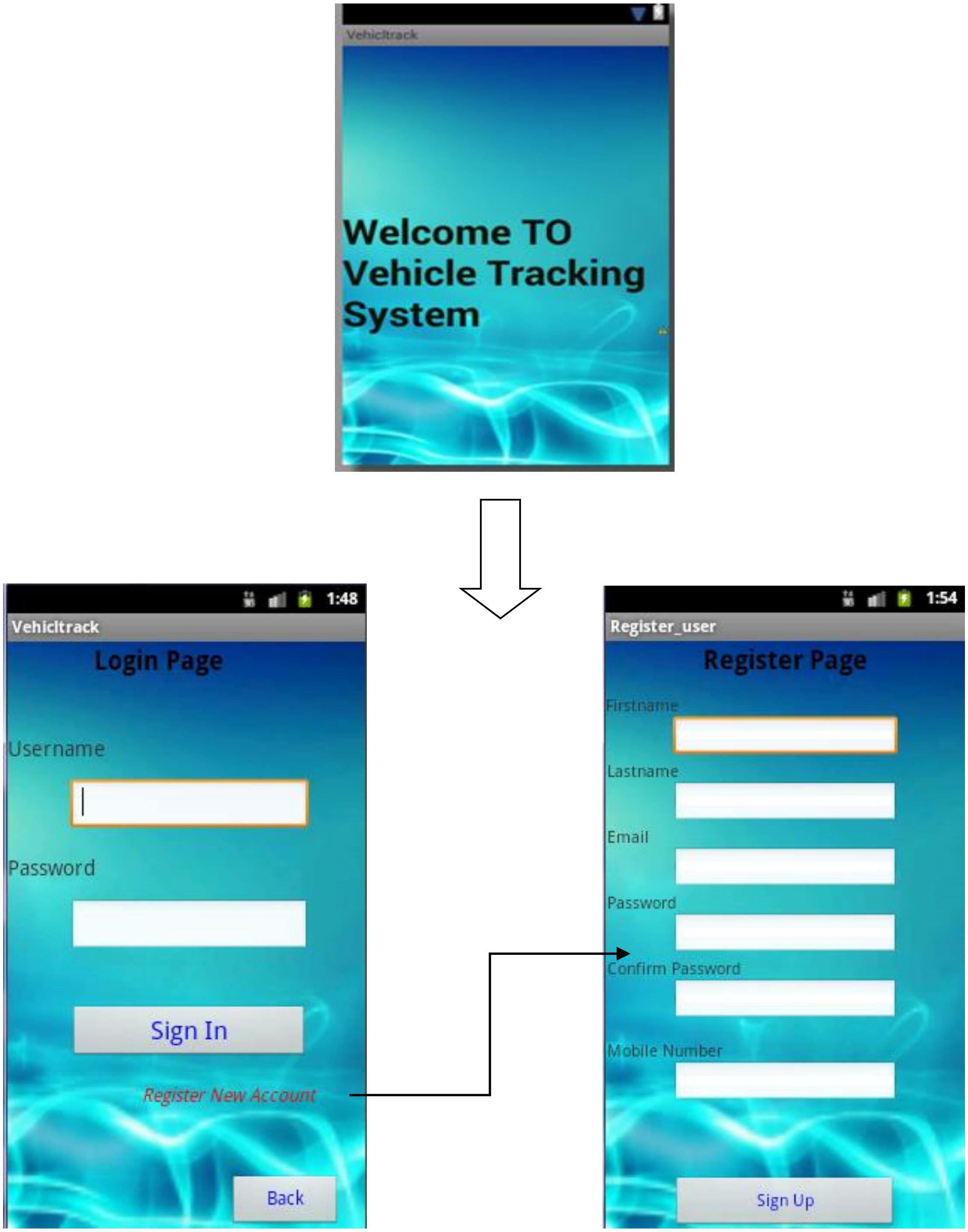
3.1.3 Linux Kernel

Android's kernel is based on the Linux kernel and has additional architecture deviations by Google outside the classic Linux kernel development cycle. Android does not have a natural X Window Scheme nor does it support the full set of usual GNU libraries, and this makes it tough to port current Linux applications or libraries to Android. Certain features that Google subsidized back to the Linux kernel, particularly a power management feature called wake locks, were rejected by mainline kernel designers, partially due to kernel maintainers sensed that Google did not show any intent to maintain their own code.

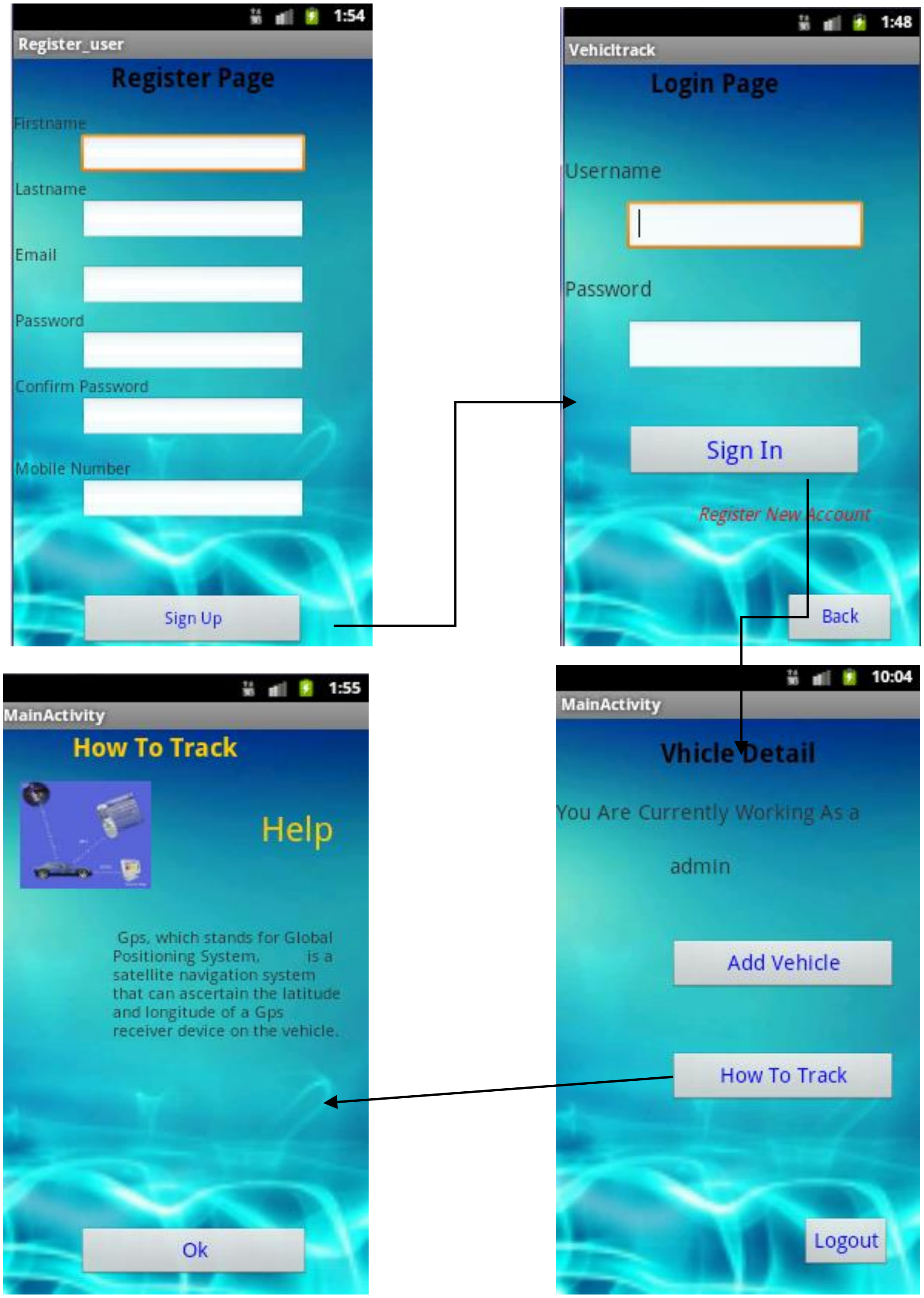
3.2 STEPWISE PROCESS IN VEHICLE TRACKING SYSTEM

The following are the pages which are included in the whole vehicle tracking system are: 1. Welcome Page 2. Login Page 3. User vehicle detail
4. How to track a vehicle 5. Register Page 6. Add Vehicle
7. List Item 8. Vehicle Option 9. List of vehicle 10. Location page

3.3 Step wise User Interface



Welcome TO Vehicle Tracking System



1:54

Register_user

Register Page

Firstname

Lastname

Email

Password

Confirm Password

Mobile Number

Sign Up

1:48

Vehicltrack

Login Page

Username

Password

Sign In

Register New Account

Back

1:55

MainActivity

How To Track

Help

Gps, which stands for Global Positioning System, is a satellite navigation system that can ascertain the latitude and longitude of a Gps receiver device on the vehicle.

Ok

10:04

MainActivity

Vhicle Detail

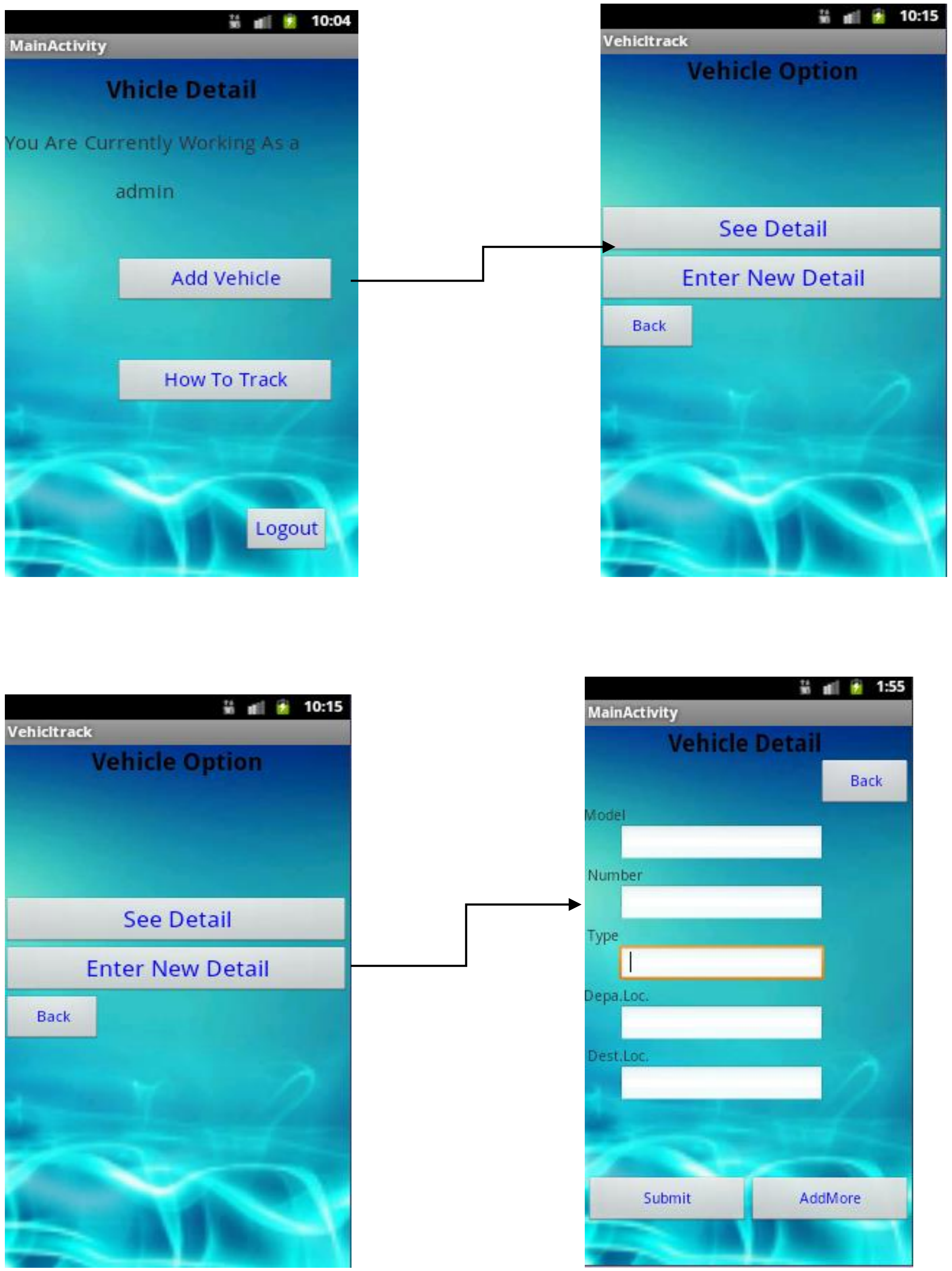
You Are Currently Working As a

admin

Add Vehicle

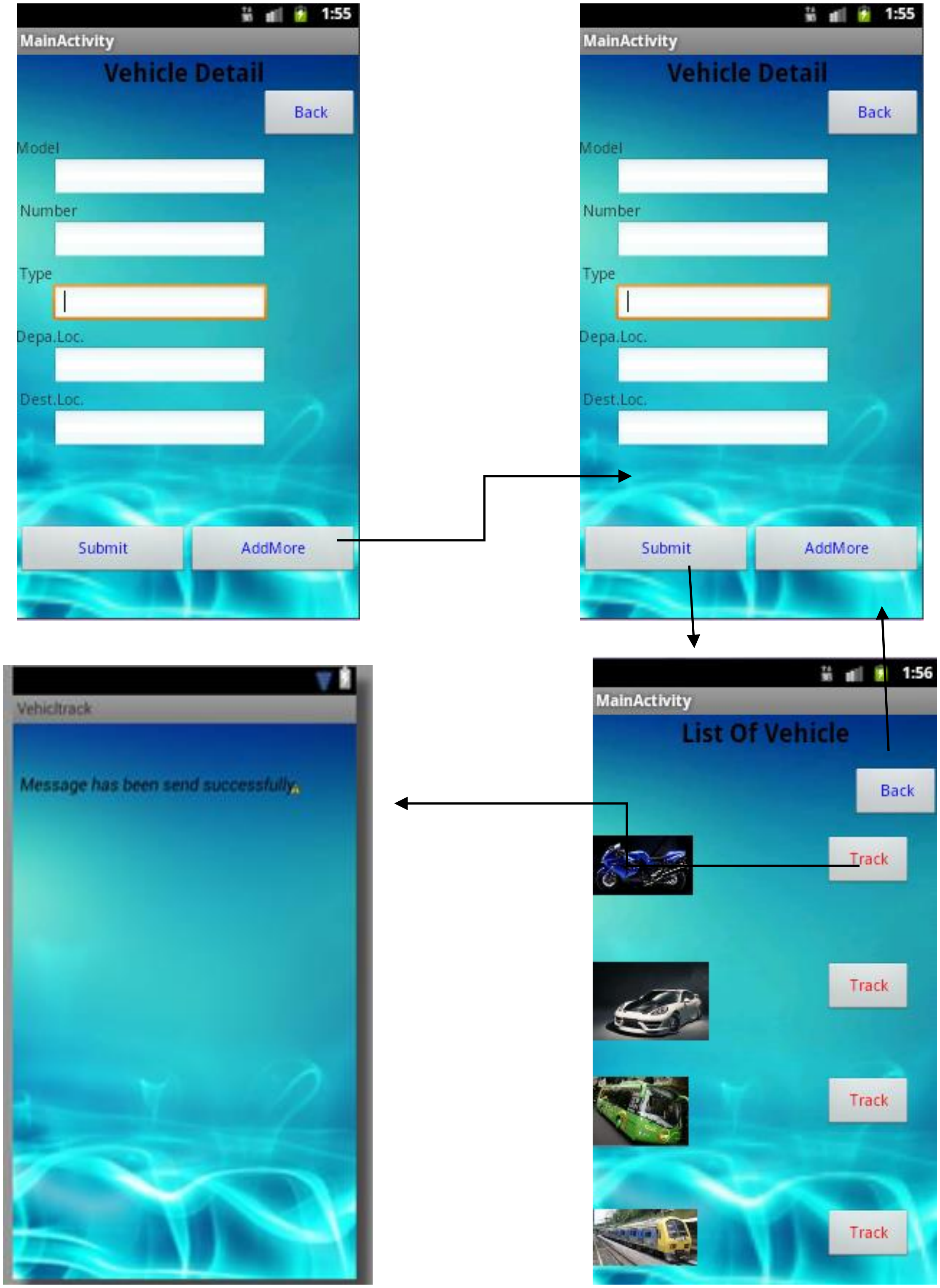
How To Track

Logout



You Are Currently Working As a

admin



Message has been send successfully.

3.4 JAVA CODING INVOLVED IN EACH STEPS

3.4.1 WELCOM PAGE

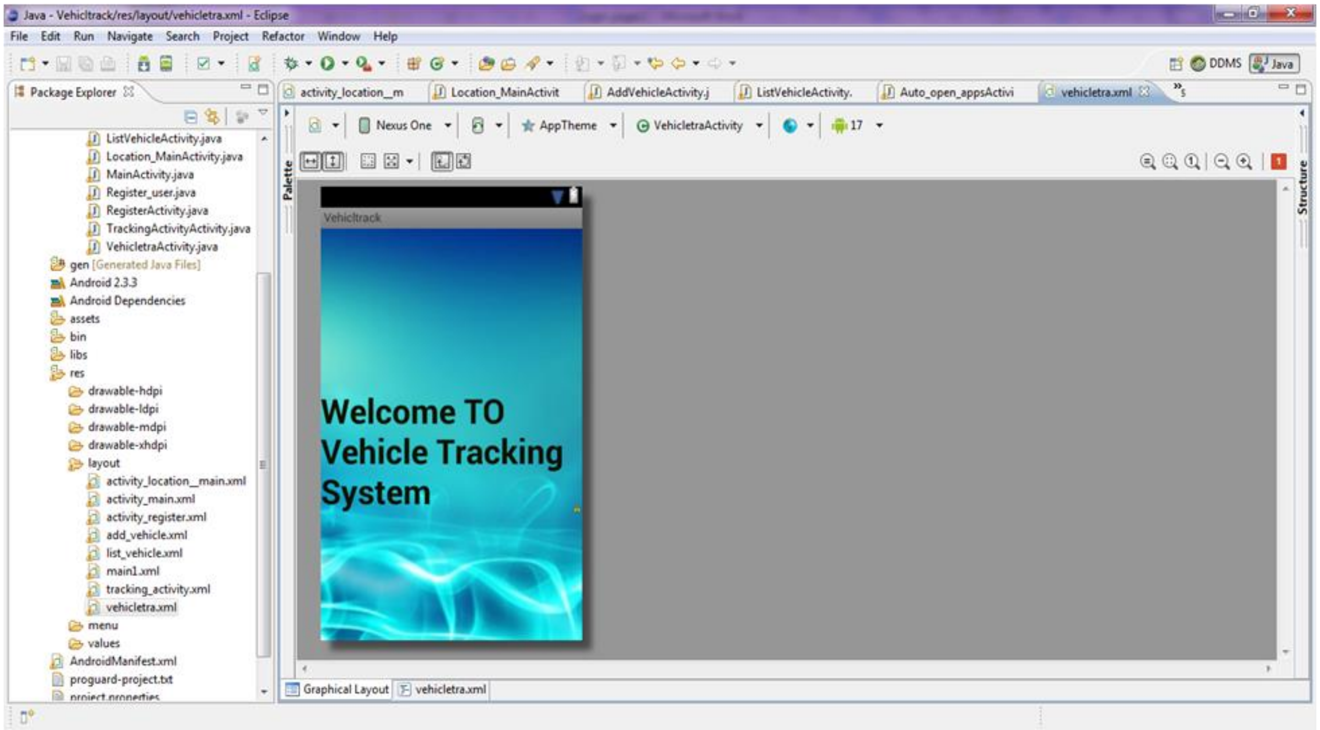


Figure 2: Welcome Page

3.4.1.1 Java Coding:

```
package example.vehicltrack;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.view.View;
public class Auto_open_appsActivity extends BroadcastReceiver
```

```
{
@Override
public void onReceive(Context context, Intent intent)
{
System.out.println("autocalled");
Intent service_intent = new Intent(context, Location_MainActivity.class);
service_intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
context.startActivity(service_intent);
}
}
```

3.4.2 LOGIN PAGE

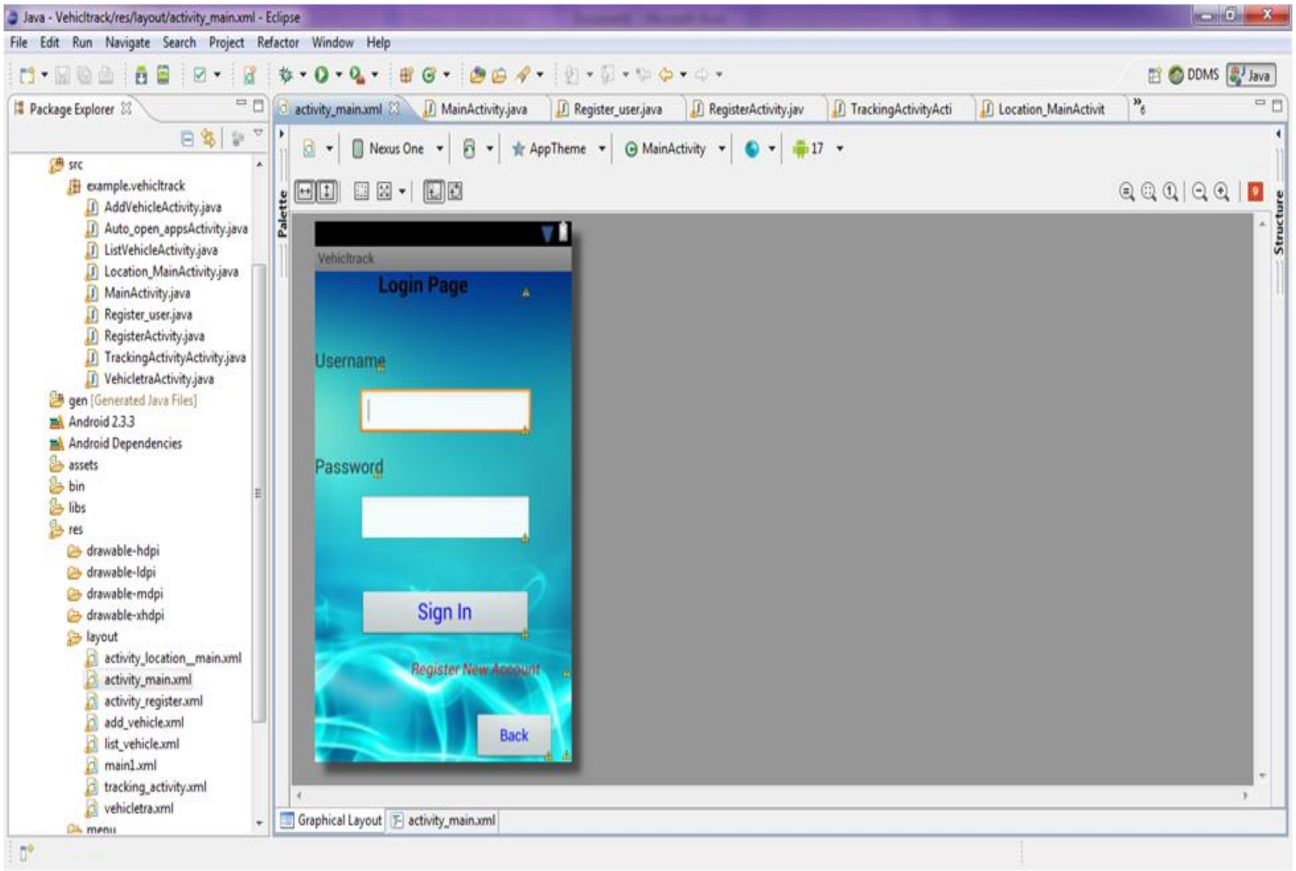


Figure 3:Login Page

3.4.2.1 Java Coding:

```
package example.vehicltrack;
import android.os.Bundle;
import android.app.Activity;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnTouchListener;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity {
Context context;
Intent intent=null;
EditText name;
EditText pass;
static String p,n;
SQLiteDatabase dv,rv;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
name=(EditText)findViewById(R.id.editText1);
pass=(EditText)findViewById(R.id.editText2);
Intent yahoo=getIntent();
```

```
/*String value1=yahoo.getStringExtra("name");

String value2=yahoo.getStringExtra("pass");

Toast.makeText(this, "Name  from is:=="+value1+"\tpassword
from="+value2,Toast.LENGTH_LONG).show();   */

}

public void call(View v)

{

Intent intent=new Intent(this,Register_user.class);

startActivity(intent);

}

public void call2(View v)

{

n=name.getText().toString();

p=pass.getText().toString();

loginconn er=new loginconn(this);

dv=er.getWritableDatabase();

rv=er.getWritableDatabase();

ContentValues value=new ContentValues();

value.put(loginconn.COL_NAME, n);

value.put(loginconn.COL_PASS,p);

//long row=dv.insert(loginconn.TABLE_NAME, null, value);

//       if(row>0)

{

Toast.makeText(getApplicationContext(), "data inserted",
Toast.LENGTH_LONG).show();
}
System.out.println(n.toUpperCase());
Toast.makeText(this, "Name
is:=="+n+"\tpassword="+p,Toast.LENGTH_LONG).show();
if((n.equals("admin"))&&(p.equals("pass")))
{
Toast.makeText(this,"first",Toast.LENGTH_LONG).show();
Intent intent=new Intent(this,RegisterActivity.class);
startActivity(intent);
name.setText("");
pass.setText("");
}
else
{
Toast.makeText(this,"second",Toast.LENGTH_LONG).show();
Intent intent=new Intent(this,MainActivity.class);
startActivity(intent);
}
}
public void call12(View v)
{
Intent intent=new Intent(this,VehicletraActivity.class);
startActivity(intent);
}}
```

3.4.3 REGISTER NEW ACCOUNT

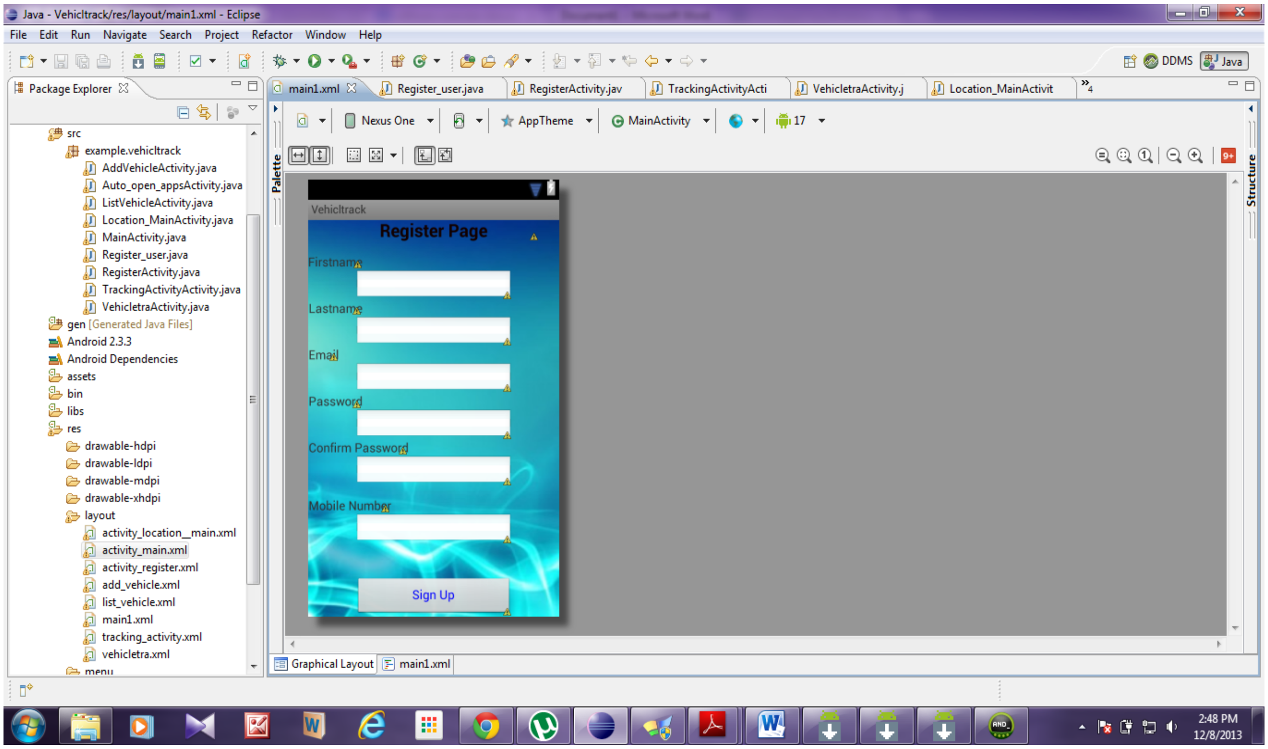


Figure 4: Register Page

3.4.3.1 Java Coding

```
package example.vehicltrack;
import android.os.Bundle;
import android.app.Activity;
import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
public class Register_user extends Activity
{
EditText fname;
EditText pass;
EditText lname;
EditText cpass;
EditText email;
EditText mobno;
String p,n,r,q,s,m;
SQLiteDatabase dv,rv;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main1);
fname=(EditText)findViewById(R.id.editText1);
lname=(EditText)findViewById(R.id.editText2);
pass=(EditText)findViewById(R.id.editText4);
cpass=(EditText)findViewById(R.id.editText5);
email=(EditText)findViewById(R.id.editText3);
mobno=(EditText)findViewById(R.id.editText6);
}
```

```
public void call1(View v)

{

n=fname.getText().toString();

p=pass.getText().toString();

q=lname.getText().toString();

r=cpass.getText().toString();

s=email.getText().toString();

m=mobno.getText().toString();

registerconn er=new registerconn(this);

dv=er.getWritableDatabase();

//rv=er.getWritableDatabase();

ContentValues value=new ContentValues();

value.put(registerconn.COL_fNAME, n);

value.put(registerconn.COL_PASS,p);

value.put(registerconn.COL_lNAME, q);

value.put(registerconn.COL_cPASS,r);

value.put(registerconn.COL_email, s);

value.put(registerconn.COL_mobilno,m);

//long row=dv.insert(registerconn.TABLE_NAME, null, value);

//if(row>0)

{

Toast.makeText(getApplicationContext(), "data inserted",
Toast.LENGTH_LONG).show();

}
System.out.println(n.toUpperCase());

Toast.makeText(this, "Name
is:=="+n+"\tpassword="+p,Toast.LENGTH_LONG).show();

Toast.makeText(this,"Data
is:=="+q+"\t"+r+"\t"+"\t"+s+"\t"+m+"\t"+m,Toast.LENGTH_LONG).show();

Intent intent=new Intent(this,MainActivity.class);

intent.putExtra("name",n);// value)

intent.putExtra("pass",p);

startActivity(intent);

fname.setText("");

lname.setText("");

pass.setText("");

cpass.setText("");

}

}
```

3.4.4 USER VEHICLE DETAIL

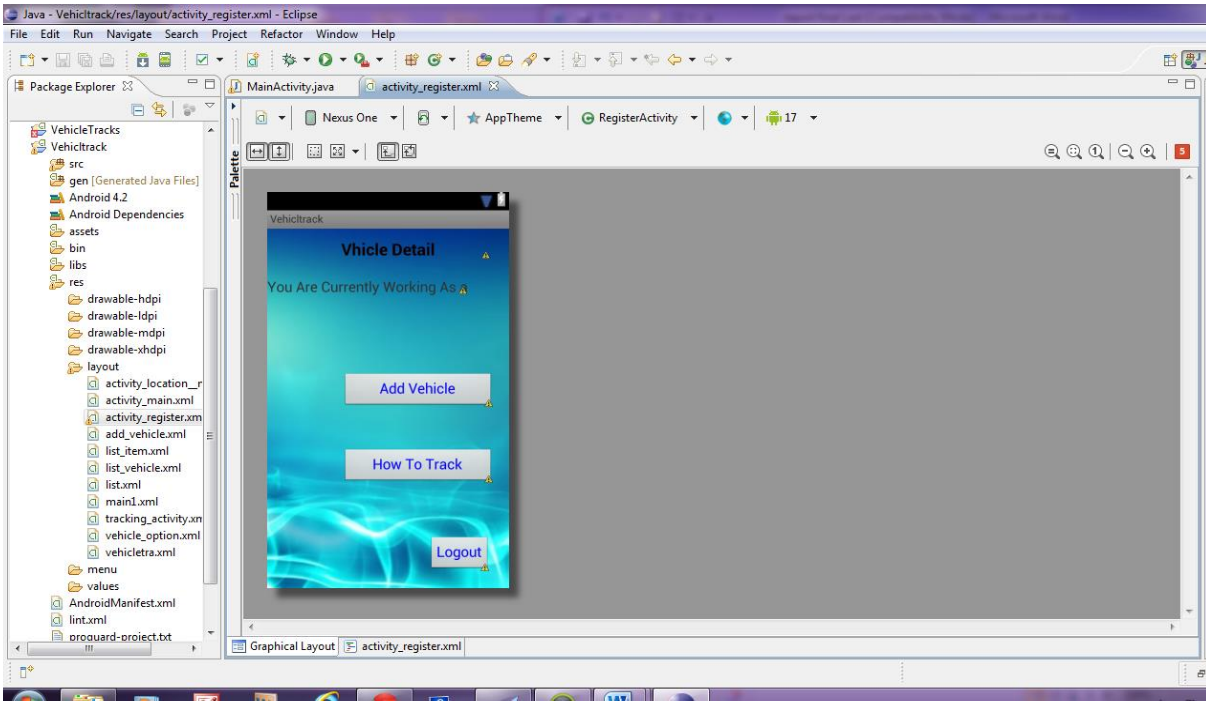


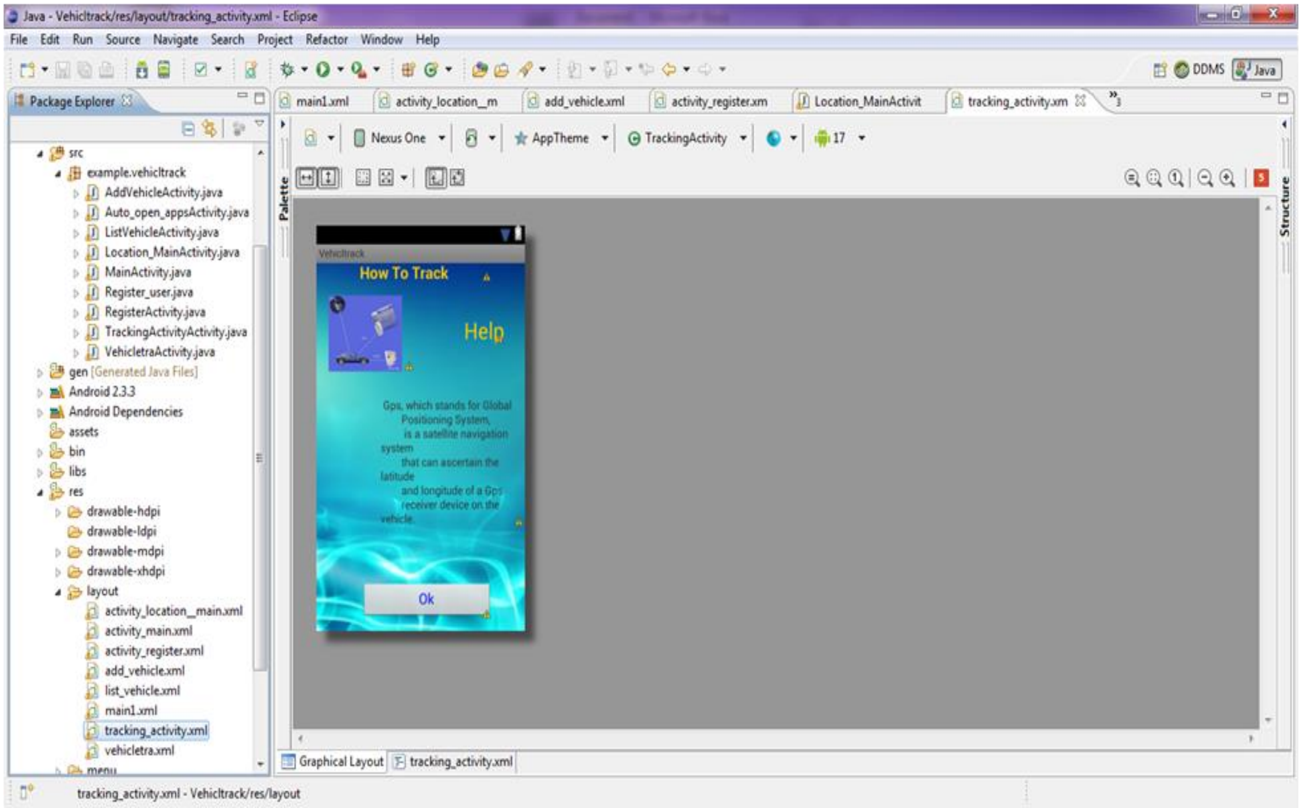
Figure 5: User Vehicle Detail

3.4.4.1 Java Coding

```
package example.vehicltrack;

import android.os.Bundle;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.view.Menu;

import android.view.View;

public class RegisterActivity extends Activity {
```

```
Context context;
Intent intent=null;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_register);
}
public void call3(View v)
{
Intent intent=new Intent(this,MainActivity.class);
startActivity(intent);
}
public void call4(View v)
{
Intent intent=new Intent(this,AddVehicleActivity.class);
startActivity(intent);
}
public void call5(View v)
{
Intent intent=new Intent(this,TrackingActivityActivity.class);
startActivity(intent);
}
}
```

3.4.5 HOW TO TRACK



Gps, which stands for Global Positioning System, is a satellite navigation system that can ascertain the latitude and longitude of a Gps receiver device on the vehicle.

Figure 6: How to track page

3.4.5.1 Java Coding

```
package example.vehicltrack;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
public class TrackingActivityActivity extends Activity
{
Context context;
Intent intent=null;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.tracking_activity);
}
public void call9(View v)
{
Intent intent=new Intent(this,RegisterActivity.class);
startActivity(intent);
}
}
```

3.4.6 ENTER NEW DETAIL

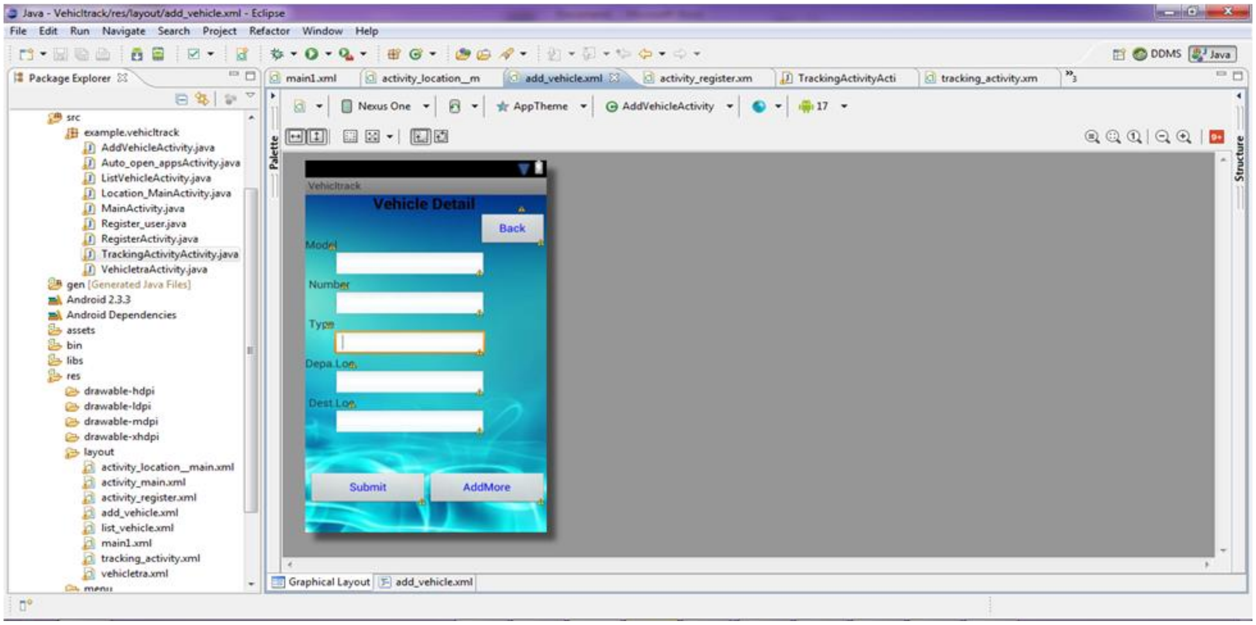


Figure 7: Vehicle Detail

3.4.6.1 Java Coding

```
package example.vehicltrack;
import android.os.Bundle;
import android.app.Activity;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class AddVehicleActivity extends Activity
{
Context context;
Intent intent=null;
EditText model;
EditText number;
EditText type;
EditText deploc;
EditText destloc;
String p,n,r,q,s;
SQLiteDatabase dv,rv;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.add_vehicle);
System.out.println("AddVehicleActivity.onCreate()");
model=(EditText)findViewById(R.id.editText1);
number=(EditText)findViewById(R.id.editText2);
type=(EditText)findViewById(R.id.editText3);
deploc=(EditText)findViewById(R.id.editText4);
destloc=(EditText)findViewById(R.id.editText5);
}
```

```
public void call6(View v)
{
n=model.getText().toString();
p=number.getText().toString();
q=type.getText().toString();
r=deploc.getText().toString();
s=destloc.getText().toString();
addvehicle er=new addvehicle(this);
dv=er.getWritableDatabase();
//rv=er.getWritableDatabase();
ContentValues value=new ContentValues();
value.put(addvehicle.COL_Model, n);
value.put(addvehicle.COL_Number,p);
value.put(addvehicle.COL_Type, q);
value.put(addvehicle.COL_Deploc,r);
value.put(addvehicle.COL_Destloc, s);
long row=er.insert_vehicle(n,p,q,r,s);
if(row>0)
{
Toast.makeText(getApplicationContext(), row+"data inserted",
Toast.LENGTH_LONG).show();
}
System.out.println(n.toUpperCase());
Toast.makeText(this, "Name
is:=="+n+"\tpassword="+p,Toast.LENGTH_LONG).show();
Toast.makeText(this,"Data
is:=="+q+"\t"+r+"\t"+"\t"+s,Toast.LENGTH_LONG).show();
Intent intent=new Intent(this,ListVehicleActivity.class);
startActivity(intent);
}
public void call7(View v)
{
Intent intent=new Intent(this,AddVehicleActivity.class);
startActivity(intent);
}
public void call26(View v)
{
Intent intent=new Intent(this,VehicleOption.class);
startActivity(intent);
}
public void startActivityForResult (Intent intent)
{
}
}
```

3.4.7 LIST OF VEHICLE

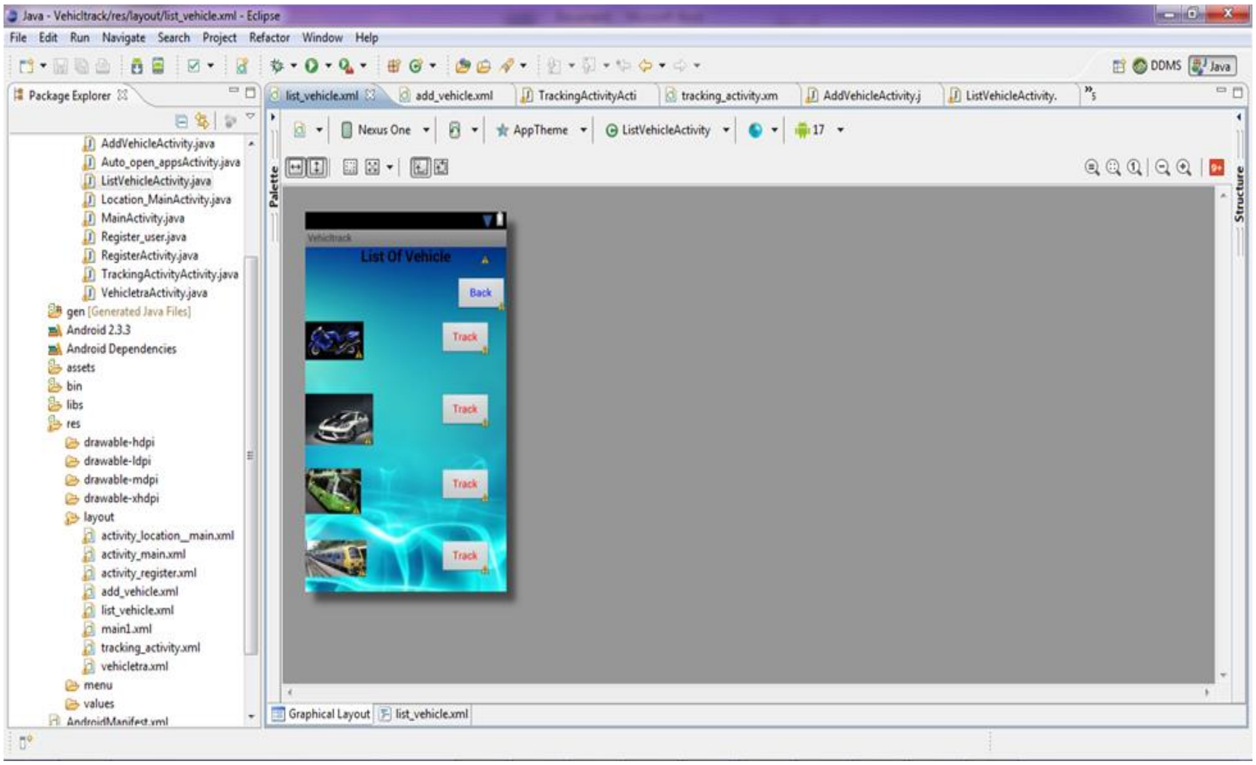


Figure 8: List of vehicle

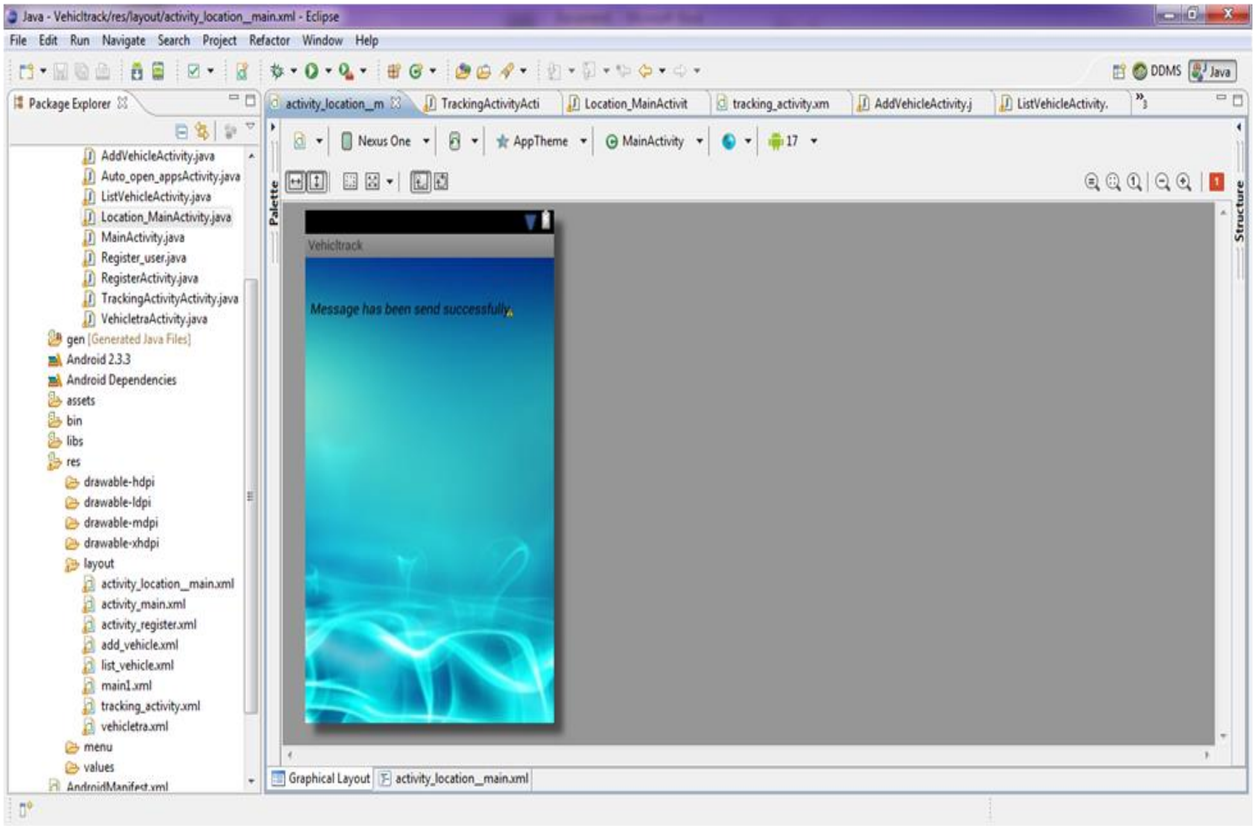
3.4.7.1 Java Coding

```
package example.vehicltrack;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
public class ListVehicleActivity extends Activity {
```

```
Context context;
Intent intent=null;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.list_vehicle);
}
public void call10(View v)
{
Intent intent=new Intent(this,AddVehicleActivity.class);
startActivity(intent);
}
public void call13(View v)
{
Intent intent=new Intent(this,Location_MainActivity.class);
startActivity(intent);
}
public void call14(View v)
{
Intent intent=new Intent(this,Location_MainActivity.class);
startActivity(intent);
}
public void call15(View v)
{
```

```
Intent intent=new Intent(this,Location_MainActivity.class);
startActivity(intent);
}
public void call16(View v)
{
Intent intent=new Intent(this,Location_MainActivity.class);
startActivity(intent);
}
}
```

3.4.8 LOCATION PAGE



Message has been send successfully

Figure 9: Location Page

3.4.8.1 Java Coding

package example.vehicltrack;

import android.location.Criteria;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.app.Activity;

import android.content.Context;

```
import android.telephony.SmsManager;
import android.util.Log;
import android.view.Menu;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
public class Location_MainActivity extends Activity implements
LocationListener{
LocationManager locationManager ;
String provider;
double lat=0.0,log=0.0;
SmsManager sms=null;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_location__main);
// Getting LocationManager object
locationManager =
(LocationManager)getSystemService(Context.LOCATION_SERVICE);
// Creating an empty criteria object
Criteria criteria = new Criteria();
// Getting the name of the provider that meets the criteria
provider = locationManager.getBestProvider(criteria, false);
if(provider!=null && !provider.equals("")){
// Get the location from the given provider
```

```
Location location = locationManager.getLastKnownLocation(provider);
locationManager.requestLocationUpdates(provider, 20000, 1, this);
if(location!=null)
onLocationChanged(location);
else
Toast.makeText(getBaseContext(), "Location can't be retrieved",
Toast.LENGTH_SHORT).show();
}else{
Toast.makeText(getBaseContext(), "No Provider Found",
Toast.LENGTH_SHORT).show();
}
}
@Override
public void onLocationChanged(Location location) {
// Getting reference to TextView tv_longitude
//Toast.makeText(getApplicationContext(), "location is +"+location,
10000).show();
// Setting Current Longitude
log=location.getLongitude();
lat=location.getLatitude();
// tvLongitude.setText("Longitude:" + location.getLongitude());
// Setting Current Latitude
// tvLatitude.setText("Latitude:" + location.getLatitude() );
try{
sms=SmsManager.getDefault();
```

```
sms.sendTextMessage("",null," Current Location Longitude is:-" +log +
"Current
Location Latitude is:-" +lat, null, null);
//  Toast.makeText(getApplicationContext(), "massage   send +..........."+sms,
10000).show();
}
catch(Exception
e){e.getStackTrace();//Toast.makeText(getApplicationContext(), "massage
not
send +..........."+sms, 10000).show();
}
// Toast.makeText(getApplicationContext(), "massage   send +..........."+sms,
10000).show();
}
@Override
public void onProviderDisabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onProviderEnabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
// TODO Auto-generated method stub
```

```
}

}
```

3.5 Sample of Android Coding Involved in Different Steps

3.5.1 Login Page

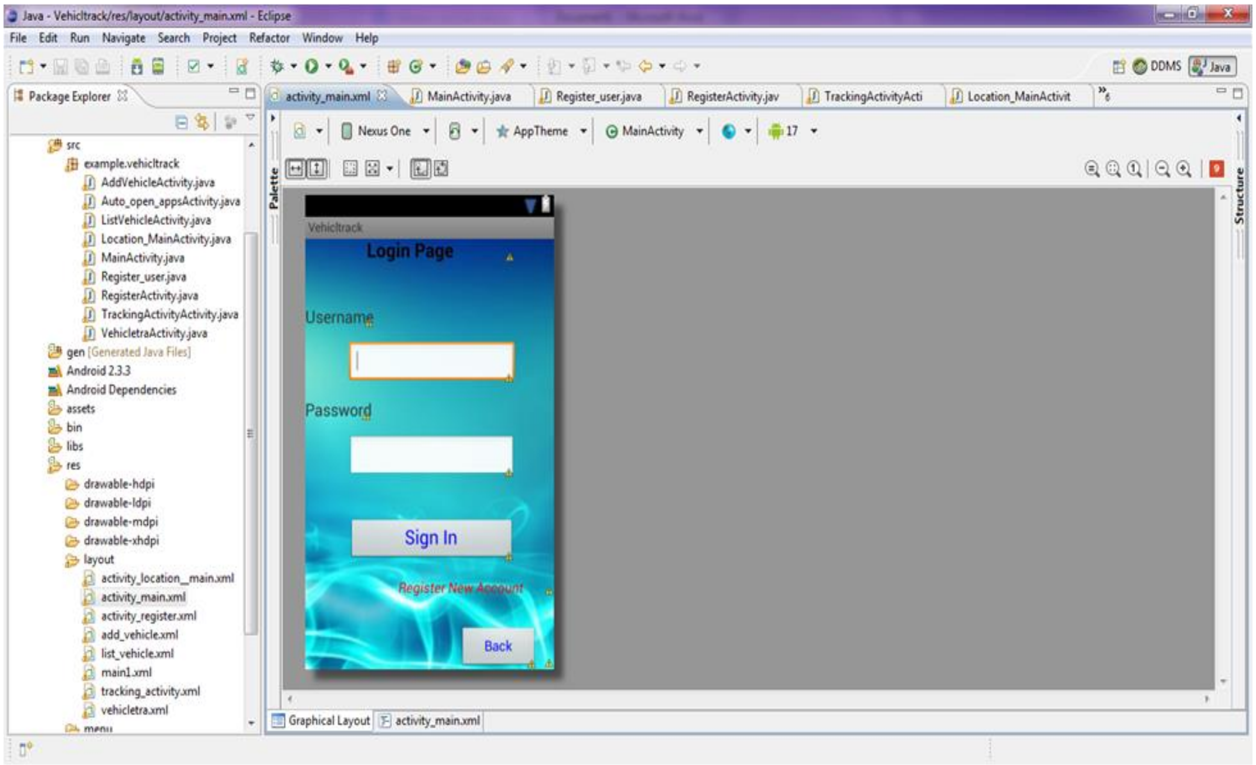


Figure 4: Login Page

3.5.1.1 Android Code Of GUI

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/re
s/android"
```

```
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
android:background="@drawable/backg">

<TextView
android:id="@+id/textView2"
android:layout_width="269dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:gravity="center"
android:text="Login Page"
android:textColor="#100"
android:textSize="22sp"
android:textStyle="bold" />

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/textView2"
android:layout_marginTop="50dp"
android:text="Username"
android:textSize="19sp"
android:textStyle="normal"/>

<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignRight="@+id/textView2"
android:layout_below="@+id/textView3"
```

```
android:layout_marginTop="16dp"
android:ems="10" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/editText1"
android:layout_marginTop="18dp"
android:text="Password"
android:textSize="19sp"
android:textStyle="normal"
/>

<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText2"
android:layout_below="@+id/textView1"
android:layout_marginTop="15dp"
android:ems="10" >

<requestFocus />

</EditText>

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText2"
```

```
android:layout_alignRight="@+id/editText2"
android:layout_below="@+id/editText2"
android:layout_marginTop="46dp"
android:text="Sign In"
android:textColor="#00f"
android:onClick="call2"
android:textSize="22sp"
android:textStyle="normal" />

<TextView
android:id="@+id/textView4"
android:layout_width="200dp"
android:layout_height="wrap_content"
android:layout_alignParentRight="true"
android:layout_below="@+id/button1"
android:layout_marginTop="20dp"
android:clickable="true"
android:onClick="call"
android:text="Register New Account"
android:textColor="#f00"
android:textSize="16sp"
android:textStyle="italic" />

</RelativeLayout>
```

3.6 SAMPLE OF SQLITE CODING INVOLVED IN DIFFERENT STEPS

3.6.1 SQLite Connectivity in Login Page

```
package example.vehicltrack;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class loginconn extends SQLiteOpenHelper
{
public static final String DatabaseName="login.db";
public static final int DATABSE_VERSION=2;
public static final String TABLE_NAME="LOGIN";
public static final String COL_ID="id";
public static final String COL_NAME="name";
public static final String COL_PASS="PASSWORD";
public loginconn (Context c)
{
super(c,DatabaseName, null, DATABSE_VERSION);
System.out.println("Login constructor");
}
String g="Create table "+TABLE_NAME+
"( "+COL_ID+" INTEGER PRIMARY KEY  AUTOINCREMENT, "+
COL_NAME +" TEXT NOT NULL, " +
COL_PASS +" TEXT NOT NULL )";
@Override
public void onCreate(SQLiteDatabase db)
{
db.execSQL(g);
}
@Override
public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
// TODO Auto-generated method stub
arg0.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
System.out.println("loginconn.onUpgrade()");
onCreate(arg0);
}
public void insert_login(String name,String PASSWORD){
SQLiteDatabase arg0=getWritableDatabase();
ContentValues content=new ContentValues();
content.put(COL_NAME,name);
content.put(COL_PASS,PASSWORD);
arg0.insert(TABLE_NAME, null,content);
}
public Cursor signin(){
SQLiteDatabase arg0=getWritableDatabase();
```

```
return  arg0.query(TABLE_NAME,new
String[]{COL_NAME,COL_PASS},null,null,null,null,null);

}

}
```

3.6.2 _*SQLite Connectivity in Register Page*

```
package example.vehicltrack;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class registerconn extends SQLiteOpenHelper

{

public static final String DatabaseName="register.db";

public static final int DATABSE_VERSION=2;

public static final String TABLE_NAME="REGISTER";

public static final String COL_ID="id";

public static final String COL_fNAME="firstname";

public static final String COL_PASS="PASSWORD";

public static final String COL_lNAME="lastname";

public static final String COL_cPASS="CONFPASSWORD";

public static final String COL_email="email";

public static final String COL_mobilno="mobno";
public registerconn (Context c)
{
super(c,DatabaseName, null, DATABSE_VERSION);
System.out.println("Register constructor");
}
String g="Create table "+TABLE_NAME+
"( "+COL_ID+" INTEGER PRIMARY KEY AUTOINCREMENT,"+
COL_mobilno +" TEXT NOT NULL,"+
COL_email +" TEXT NOT NULL,"+
COL_cPASS +" TEXT NOT NULL, "+
COL_lNAME +" TEXT NOT NULL, "+
COL_fNAME +" TEXT NOT NULL, " +
COL_PASS +" TEXT NOT NULL )";
@Override
public void onCreate(SQLiteDatabase db)
{
db.execSQL(g);
}
@Override
public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
// TODO Auto-generated method stub
arg0.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
System.out.println("registerconn.onUpgrade()");
onCreate(arg0);
}
public void insert_register(String firstname,String PASSWORD,String
lastnane,String email,String mobno,String CONFPASSWORD){
SQLiteDatabase arg0=getWritableDatabase();
ContentValues content=new ContentValues();
content.put(COL_fNAME,firstname);
content.put(COL_PASS,PASSWORD);
content.put(COL_lNAME,lastnane);
content.put(COL_cPASS,CONFPASSWORD);
content.put(COL_email,email);
content.put(COL_mobilno,mobno);
arg0.insert(TABLE_NAME, null,content);
}
public Cursor signup()
{
SQLiteDatabase arg0=getWritableDatabase();
return  arg0.query(TABLE_NAME,new
String[]{COL_fNAME,COL_PASS,COL_lNAME,COL_cPASS,COL_email,
COL
_mobilno},null,null,null,null,null);
}
}
```

3.6.3 *SQLite Connectivity in Vehicle Detail*

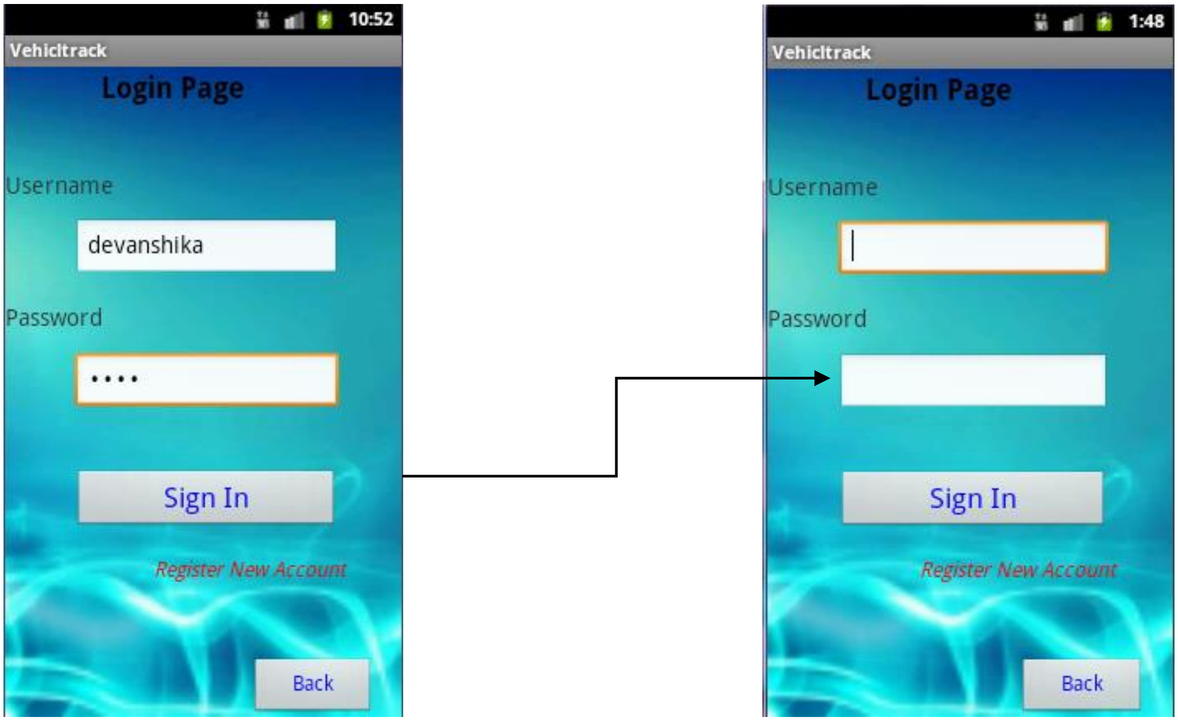
```
package example.vehicltrack;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class addvehicle extends SQLiteOpenHelper
{
public static final String DatabaseName="addvehicle.db";
public static final int DATABSE_VERSION=4;
public static final String TABLE_NAME="ADDVEHICLE";
public static final String COL_ID="_id";
public static final String COL_Model="model";
public static final String COL_Number="number";
public static final String COL_Type="type";
public static final String COL_Deploc="deploc";
public static final String COL_Destloc="destloc";
public addvehicle (Context c)
{
super(c,DatabaseName, null, DATABSE_VERSION);
System.out.println("Addvehicle constructor");
}
```

```
String g="Create table "+TABLE_NAME+
"( "+COL_ID+" INTEGER PRIMARY KEY AUTOINCREMENT,"+
COL_Model +" TEXT NOT NULL,"+
COL_Number +" TEXT NOT NULL,"+
COL_Type +" TEXT NOT NULL, "+
COL_Deploc +" TEXT NOT NULL, "+
COL_Destloc +" TEXT NOT NULL)";
@Override
public void onCreate(SQLiteDatabase db)
{
db.execSQL(g);
}
@Override
public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
// TODO Auto-generated method stub
arg0.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
System.out.println("addvehicle.onUpgrade()");
onCreate(arg0);
}
public long insert_vehicle(String model,String type,String number,String
deploc,String destloc){
SQLiteDatabase arg0=getWritableDatabase();
ContentValues content=new ContentValues();
content.put(COL_Model,model);
content.put(COL_Type,type);

content.put(COL_Number,number);

content.put(COL_Deploc,deploc);

content.put(COL_Destloc,destloc);

return arg0.insert(TABLE_NAME, null,content);

}

public Cursor submit(){

SQLiteDatabase arg0=getWritableDatabase();

return  arg0.query(TABLE_NAME,new

String[]{COL_ID,COL_Model,COL_Type,COL_Number,COL_Deploc,COL_Des

tloc},null,null,null,null,null);

}

}
```

3.7 OUTPUT

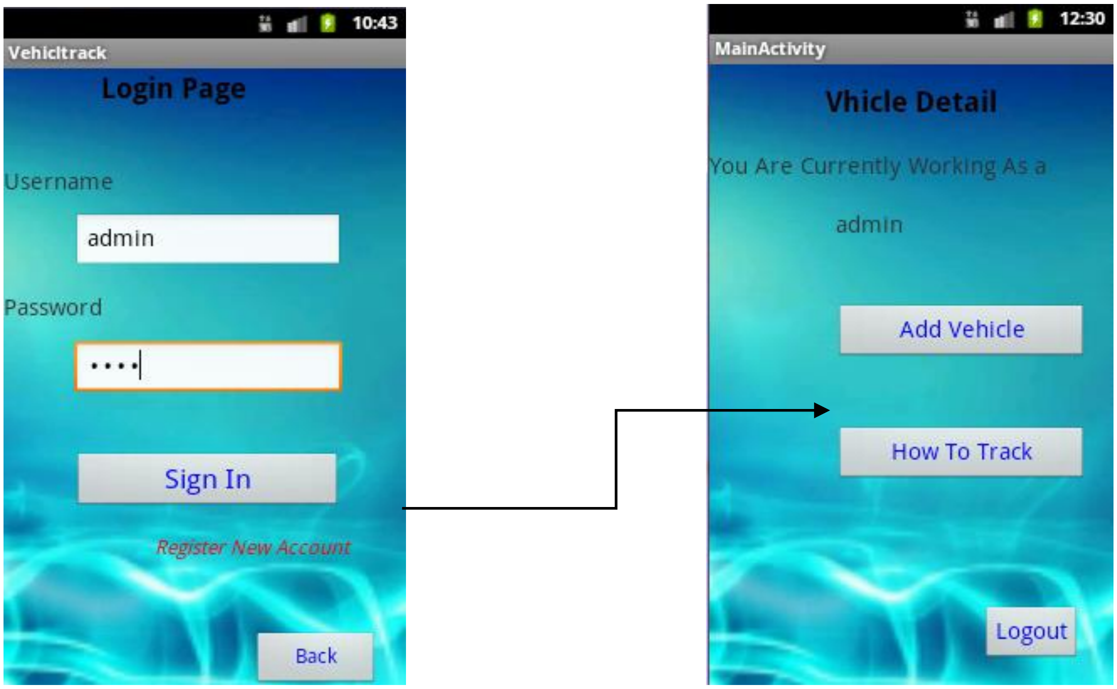
3.7.1 User interface for creating new user or signing up for existing user:

3.7.1.1 LOGIN UNSUCCESSFUL



LOGIN FAIL…………………………………………………………………

3.7.1.2 LOGIN SUCCESSFUL



Username

admin

Password

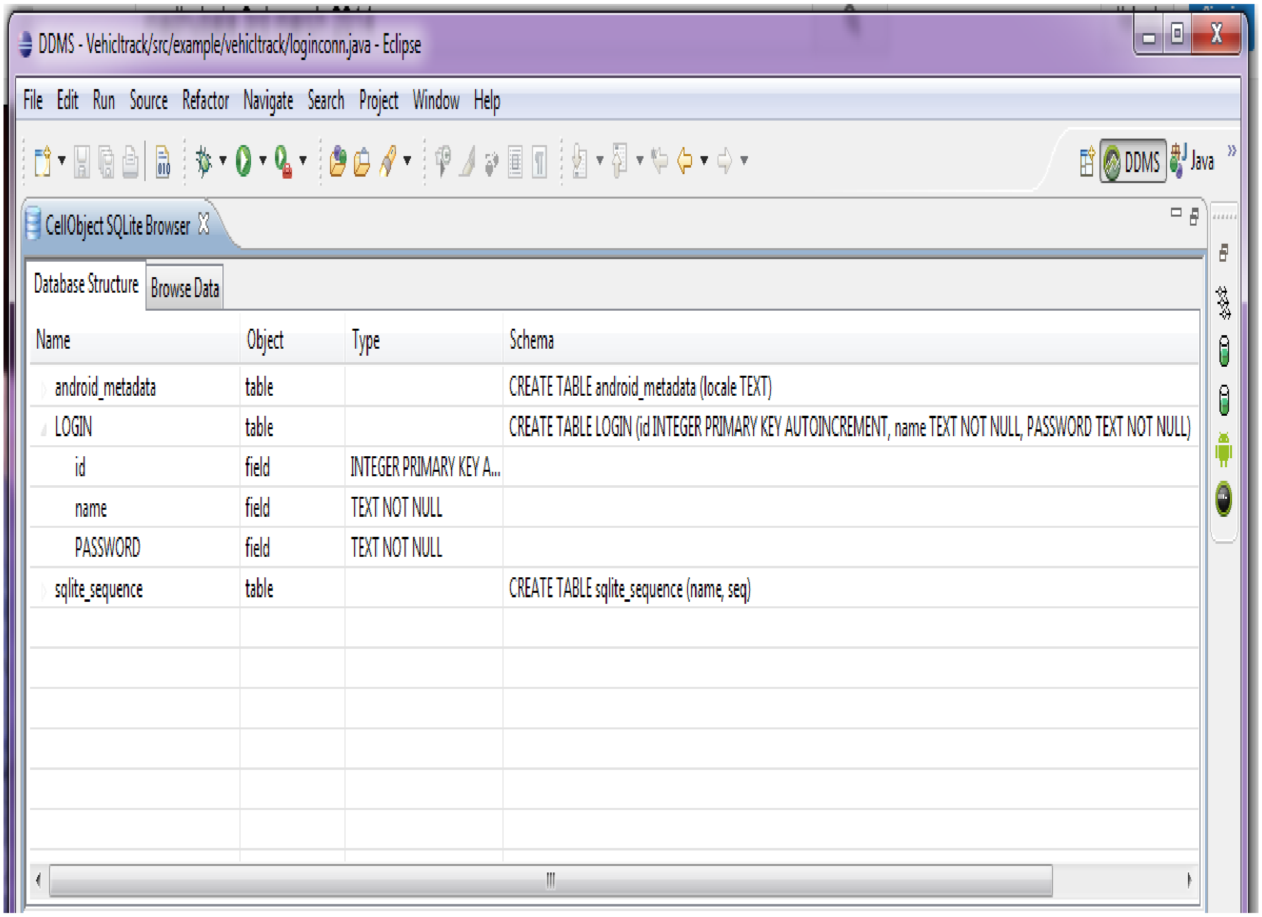
••••

You Are Currently Working As a

admin

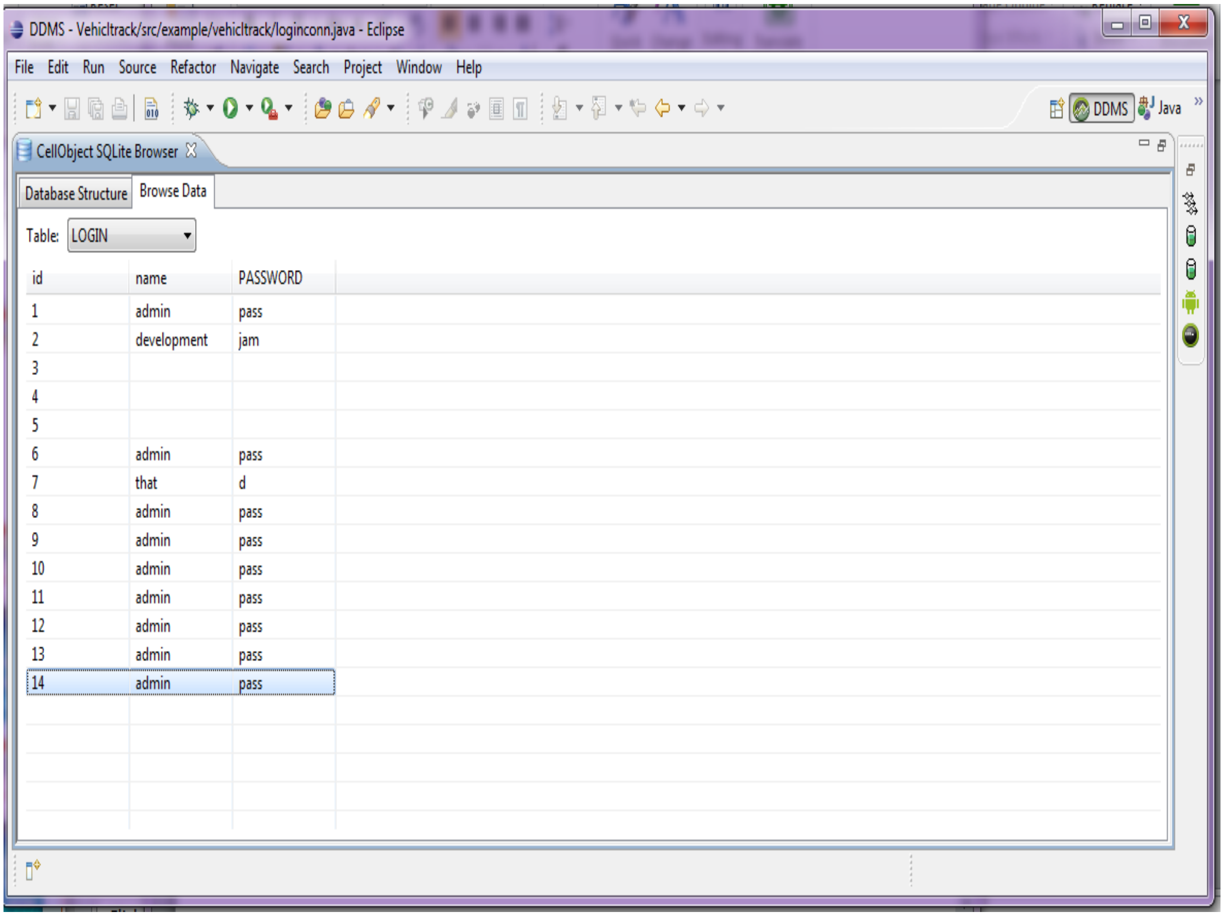
LOGIN SUCCESSFUL…………………………………………………

3.7.1.3 TABLE CREATED IN SQLITE DATABASE



DDMS - Vehicltrack/src/example/vehicltrack/loginconn.java - Eclipse
File Edit Run Source Refactor Navigate Search Project Window Help
DDMS
Java
CellObject SQLite Browser
Database Structure
Browse Data

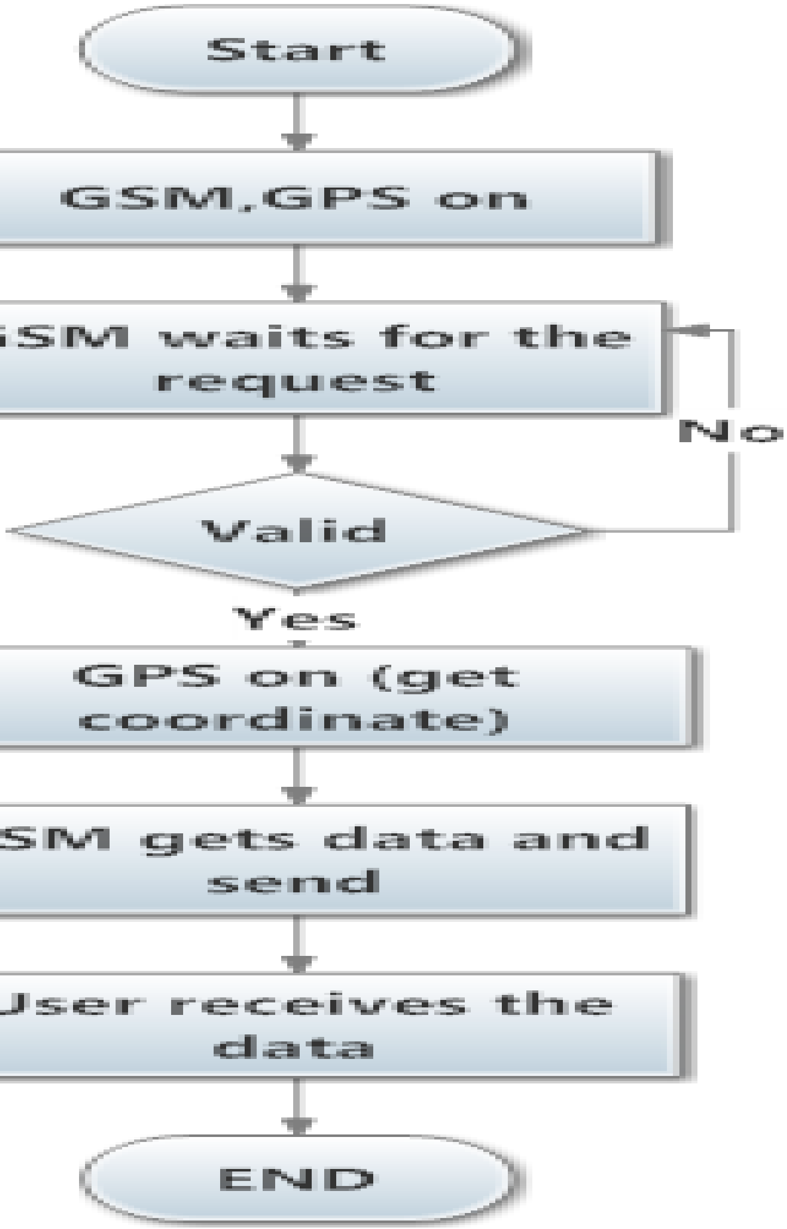
Name	Object	Type	Schema
android_metadata	table		CREATE TABLE android_metadata (locale TEXT)
LOGIN	table		CREATE TABLE LOGIN (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, PASSWORD TEXT NOT NULL)
id	field	INTEGER PRIMARY KEY A...	
name	field	TEXT NOT NULL	
PASSWORD	field	TEXT NOT NULL	
sqlite_sequence	table		CREATE TABLE sqlite_sequence (name, seq)


DDMS - Vehicltrack/src/example/vehicltrack/loginconn.java - Eclipse
File Edit Run Source Refactor Navigate Search Project Window Help
DDMS
Java
CellObject SQLite Browser
Database Structure
Browse Data
Table: LOGIN

id	name	PASSWORD
1	admin	pass
2	development	jam
3		
4		
5		
6	admin	pass
7	that	d
8	admin	pass
9	admin	pass
10	admin	pass
11	admin	pass
12	admin	pass
13	admin	pass
14	admin	pass

Chapter 4

4 FLOW CHART



Chapter 5

5 CONCLUSION

This project proposes a new vehicle tracking system, which makes use of social network as a value added service for traditional tracking systems. For vehicle tracking in a real time, in-vehicle component and a tracking server is used. The information is conveyed to tracking server using GPRS/GSM module on GSM network by direct TCP/IP connections with tracking server over GPRS. Vehicles informations is documented in tracking server database. This information like vehicle position on google maps, and vehicle status i.e. door, and ignition is only obtainable to authorized users of the system via web interface over the internet. User can send different commands to in-vehicle unit (restart, shut down) to remotely control his vehicle, which can be used as vehicle security and tracking system.

BIBLIOGRAPHY

1. M. A. Al-Taee, O. B. Khader, and N. A. Al-Saber,“ Remote monitoring of vehicle diagnostics and location using a smart box with Global Positioning System and General Packet Radio Service,” in Proc. IEEE/ACS AICCSA, Amman, 2007, pp. 385–388.
2. J. E.Marca, C. R. Rindt,M.Mcnally, and S. T. Doherty, “A GPS enhanced in-vehicle extensible data collection unit,” Inst. Transp. Studies, Univ.California, Irvine, CA, Uci-Its- As-Wp-00-9, 2000.
3. C. E. Lin, C.-W. Hsu, Y.-S. Lee, and C.C.Li, “Verification of unmanned air vehicle flight control and surveillance using mobile communication,”J. Aerosp. Comput. Inf. Commun., vol. 1, no. 4, pp. 189–197, Apr. 2004.
4. T. Hapsari, E.Y. Syamsudin, and I. Pramana, “Design of Vehicle Position Tracking System Using Short Message Services And Its Implementation on FPGA”, in Proc. Conference on Asia South Pacific Design Automation, Shanghai, China, 2005.
5. X. Fan, W. Xu, H. Chen, and L. Liu, “CCSMOMS: A Composite Communication Scheme for Mobile Object Management System”, in Proc. 20th International Conference on Advanced Information Networking and Applications, Vienna, 2006, pp. 235–239.
6. W. C. M. Hsiao, and S. K. J. Chang, “The Optimal Location Update Strategy of Cellular Network Based Traffic Information System”, in Proc. Intelligent Transportation Systems Conference, Toronto, 2006, pp. 248-253.
7. E. M. Tamil, D. B. Saleh, and M. Y. I. Idris, “A Mobile Vehicle Tracking System with GPS/GSM Technology”, in Proc. 5th Student Conference on Research and Development (SCORED), Permala Bangi, Malaysia, 2007, pp. 398-402.
8. Lita, I. B. Cioc and D. A. Visan, “A New Approach of Automobile Localization System Using GPS and GSM/GPRS Transmission,” in Proc. Int. Spring Seminar on Electronics Technology, 2006, pp. 115- 119.

9. T. K. Kishore, T. S. Vardhan, and N. L. Narayana, “Vehicle Tracking Using a Reliable Embedded Data Acquisition Sysytem With GPS and GSM”, Int. Journal of Computer Science and Network Security, vol. 10, no. 2, pp. 286-291, 2010.
10. Dr. Kamal Jain & Rahul Goel, GPS Based Low Cost Intelligent Vehicle Tracking System (IVTS), 2012 International Conference on Traffic and Transportation Engineering (ICTTE) 2012.
11. Montaser N. Ramadan et. al., Intelligent anti-theft and tracking system for Automobiles, International Journal of Machine Learning and Computing (IJMLC) 2012.
12. Rajeev Kumar et. al., Efficiency related parameters in GPS based vehicle tracking system working for an organization, International Journal of Computer Applications(IJCA) 2013.
13. Baburao Kodavati et. al, Gsm and gps based vehicle location and tracking System, International Journal of Engineering Research and Applications (IJERA).
14. http://en.wikipedia.org/wiki/Vehicle_tracking_system
15. http://www.cdactvm.in/vehicletracking.pdf