## CHAPTER – 1

## INTRODUCTION

Roadways are the most important mode of transport and mobility in the 21st century. But because of the excessive population and ever-increasing number of vehicles on the road, it has become difficult even for civic authorities to stop and control accidents. Accident-Hotspot warning application will alert and guide the travelers to drive safely and avoid accidents. This app is mainly developed in Android which will focus on searching, alert, notification about the accident-prone area (and the available emergency services nearby). Also, the accessibility of the users is not restricted, so it will be increase the scope of project.

This project is intended for making use of today’s technology of Android Development, Database management, Map extraction. Currently, there are many open-source and commercial tools available for Android Development, Database Management and Maps Extraction. So, evaluation of tools is an important job in order to find more efficient tool. Android Studio is a tool which basically used for preparing software emulation to the apps.

* 1. **Black spots**

Black spot was originally defined as “a road location of limited are with a high concentration of accidents”. The term “Black spots” is said to derive from the method that was originally used to identify hazardous sites. Accidents were pinpointed on a map, using coloured pins to represent the trauma severity of each of these events. Black was reserved for accidents having caused property damage only and the significant proportion of these accidents created black dots at concentration points. There is no universally accepted definition of a Black spot to the best knowledge of the author. The terms “hazardous location” and “high accident locations” often used as synonym. This definition has progressively evolved as several researchers now recommend including the concept of “Potential for Improvement (PI)”.

* 1. **Accident**

Accident is defined as an error in driver-vehicle-roadway system and it must be recognized that different types of accidents are caused due to different at any given location namely, rear end, side-swipes, head-on, night-time, bad-weather, etc. for instance, predominance of rear-end accidents will indicate slippery pavement whereby the drivers have difficulty in stopping in time. Side-swipe accidents will indicate ambiguous traffic control devices, causing confusion among the drivers regarding right-of-way. Head-on collisions signify lack of adequate sight distances at the location. Predominance of night-time accidents at the location will indicate serious problems with night time visibility. Bad weather accidents can results due to a road pavement which becomes dangerously slippery when wet or it may be due to inadequate signs for inclement weather. Similarly, there are numerous other types of accidents which occur due to a variety of reasons. The job of the accidents analyst is, therefore, to relate the accidents experience at any location to one or more causes. The state of the art permits such treatments.

* 1. GPS

Turning ON the GPS module on the phone would not cost us anything but getting a location usually involves transaction with cell phone service provider so as to extract the location fast and with as little network connectivity as possible plus non visibility of satellites. In short: no cell phone service implies any GPS location, as far as handheld devices are considered.

* + 1. Normal GPS

The method is called trilateration. The receiver listens to a particular frequency and gets data packets in the form of time coded messages from satellites. The receiver figures which satellites it can hear from. It starts gathering those messages containing time information from atomic clocks, current satellite positions etc. Nominal time to get a location is around 30-60 seconds. The same information needs to be confirmed by at least two other satellites.

* + 1. Only Using Mobile Services

The user location in an area is calculated with the help of signal measurements with the information received from cell towers. Information analyzed are angle to approach towers, multipath fading characteristics with signal strength comparisons. No GPS module used.

* + 1. Assisted GPS

This is what a cell phone normally uses for mapping and GPS use purposes. User location information is retrieved within 5-10 seconds. The GPS components are shared with other mobile components and hence simultaneous use of GPS and normal voice/video usage is done. First, gross positioning information from service provider based on what cell tower is being accessed and the same is fed to the GPS receiver. Next, the phone switches from phone to GPS mode for around 0.1 seconds and collects raw GPS data from satellites. It then switches back to phone mode and sends the data to the service provider to be analyzed.

The service provider uses its servers to process the data and send the most accurate location back to the phone to be displayed on a map overlay.

**1.3.4 Full Chip GPS Receiver**

The module still gets data from the service provider such as tower positioning and satellites to hear from. Switching is for 1 second but after that the receiver keeps track of information with very low power drawn from circuit.

* 1. Android as an Operating System

Being a mobile operating system, android OS is a modified version of Linux, originally developed by a start-up, Android, Inc. As Google entered mobile market, it purchased Android and in a bid to encourage independent development works, it released the developer tools under the open source Apache License. The permissive licensing allows the OS and related software to be modified and distributed by enthusiastic developers, network operators & device manufacturers.

## CHAPTER – 2

**LITERATURE SURVEY**

We went through a couple of Research papers by international University students and found the following methods as useful for a low budget country like India. Because road safety and Identification is the sole prerogative of the Ministry of National Highways and Road Transport, Government of India, citizens and civilians can only suggest ways to curtail road accidents. They are:

1) Safety Auditing: A basic premise in accident remedial work is that engineering safety principles are adhered to at all times and it is a necessary exercise to ensure that no adverse features are unwittingly introduced at a treatment site. By making extensive use of check lists, safety auditing aims at providing guidance on how to cater for the safety needs of the road user. Objectives of safety auditing include ensuring that all highway schemes operate as safely as possible and that accident producing elements are not present in any completed scheme. It is essential that safety auditing is carried out independently of the remedial measure design team.

2) Analysis and Determination using WSI Method: Yearly accident data (secondary data) for the Pune district were collected from Ministry of Highways, Government of India . The top ranked six accident black spots in Pune were identified using Weighted Severity Index Method (WSI) by assigning scores based on the number and severity of accidents in that particular location during the last year. Weighted Severity Index, (WSI) = (41 x K) + (4 x GI) + (1 x MI) Where, K is the number of persons killed; GI is the number of grievous injuries; and MI is the number of minor injuries.

Ajinkya Patankar. et.al. [1]. If the accident percentage is more on upcoming spot then one could bypass the road by taking left or right turn which is before the hot-spot and again connect to previous road. If user required some services like nearby hospital, petrol pump, hotels etc. is provided on map. For accessing current accidental data the database will connect to a particular newspaper. So that the driver will get an idea about recent accidents. So updated information will help a lot and a driver will aware of it. For analyzing a data, we are using clustering. We are using clustering for data analysis. Important points of project are Google Maps API and Database Management. Start from Android App Development with the help of Android Studio, then extract Google Maps API and prepare their database with respect to location of Hotspot, time of alert notification, provision for the user to input the information, etc. Users of this product environment are developers, end-users, customers, public authorities. So, it is reasonable to assume that an average developer has knowledge about functionalities and usage of cloud tools. For other users detail documentation and tutorials will be provided. Apps are portable means running on most number of different platforms without additional efforts. For tools operations, we can develop an app using various tools like android studio, eclipse, etc. So, our app uses a combination of database development and management, android app development, and Google Maps API.

ABDALWHAB BAKHEET. et. al. [2] Mobile phones and their applications have become an essential part of our lives. They are not only connecting us with friends and families, but also they can now tell us where we are, where to go, what to do, and how to do it. The Internet is chock full of applications that can entertain us and make our lives easier. Location-aware and mapping applications have gone from a desirable feature to an essential part of any smart phone. Companies, universities, airports and organizations of today are now providing maps either as part of their websites or as stand-alone applications to give direction services. Users provided with information about how to reach the specific organization, and how to navigate inside that organization. Due to this massive need, scores of companies and developers have developed customized mapping applications and mapping APIs (application programming interface).

## CHAPTER – 3

**PROBLEM IDENTIFICATION**

Because of the Excessive increase in the population there is also increase in number of vehicles. As there is increase in vehicles the traffic is increasing in excessive amount. And this leads to the accidents. So to avoid accidents we need to prepare an application which alerts about the black spots. The black spots are Accidents zones which are identified by the Highway authority of India. And there are Sign boards at the roadside which indicates accident zones, speed limit in certain areas and Humps.

## CHAPTER – 4

1. **OBJECTIVES**

* Main Objective is to prevent accidents.
* To alert the user about Black spots during the night with Voice alert.
* To alert the user if the user is over speeding.
* Track the current user location.
* Alert user about speed limit on particular road.
* Alert if any Hospitals are ahead.
* To display the markers in 30m radius of users current location.

## CHAPTER – 5

1. **REQUIREMENTS ANALYSIS**

**5.1 OVERVIEW**

The requirement analysis includes functional and nonfunctional requirements, Hardware and Software requirements (defined in the respective sections). The functional requirements are the activities that perform with the admin dashboard and database interactions. The nonfunctional requirements include feasibility, reliability, and scalability. Inclusive of hardware and software requirements.

Functional requirements are supported by non-functional requirements, which impose constraints on the design or implementation. Generally, functional requirements are expressed in the form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>". The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

**5.2 FUNCTIONAL REQUIREMENTS**

Functional requirements for a system describe the functionality or services that the system is expected to provide a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.

Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Therefore, for this project the functional requirements are as follows:

* The android app able to update the signals which are available in database for the user current location.
* The app should be able to display the message if no location found.
* The locations updated by admin must be stored in the server.
* The app should able to perform requesting and fetching of stored data from server.
* If any locations found for the current location of user, app should able display the signals and should be alert through the voice.

**5.3 NON FUNCTIONAL REQUIREMENTS**

In system engineering and requirement engineering a nonfunctional requirement is a requirement that specifies criteria that can be used to judge the operation for system, rather than specific behavior they are contrasted with functional requirements that define specific behavior of function**.**

**User Interface:** User Interface must be user friendly, showing all kind of results and non-technical person also must be able to operate.

**Error handling:** It is essential that system must generate appropriate messages, when the requirements and not following.

**5.4 Software** **Requirements:**

* Operating System: Windows 7 or higher.
* Notepad++
* Android Studio.
* MySQL Database.

**5.5 Hardware Requirements:**

* Processor: i3 or higher
* Processor speed: 2.5Ghz or higher
* RAM: 2GB or higher
* VRAM: 512MB or higher
* I/O: Keyboard, Mouse and Monitor
* Android device: Android Kitkat or higher.

**CHAPTER-6**

**SYSTEM DESIGN**

**6.1 DATAFLOW DIAGRAM(DFD)**

DFD-0

DATABASE

SERVER

APP

USER

ADMIN

Fig 6.1 data flow diagram-0

In the fig 6.1, the user is directly uses a android app which is connected to the server for requesting and fetching of the data. The server is connected to the database where manually stored signals are present. These signals are stored by the admin, where database is connected to the server and admin.

DFD-1

USER

Google map interaction

Android to server integration

Marker display module

Voice interpretation module

SERVER

ADMIN

Add module

Authentication module

DATABASE

Signal interaction module

Google map integration and database integration

Fig 6.2 data flow diagram-1

In fig 6.2 the android app is divided into 4 modules those are Google map integration, Android to server integration, marker display and voice interpretation module which are connected each other respectively. The whole modules are connected to the server.

The admin is divided into four modules Authentication module, Add module, create, read update and delete signals module and Google map integration .The last two modules are interconnected to each other further those are connected to the database which stores signals with its type and name along with signal latitude and longitude are stored.

**6.2 ER-DIAGRAM**

Black Spots

Fig 6.3 ER diagram for admin

Location type

Types

Points

Fig 6.4 ER diagrams

**6.3 USE CASE DIAGRAM**

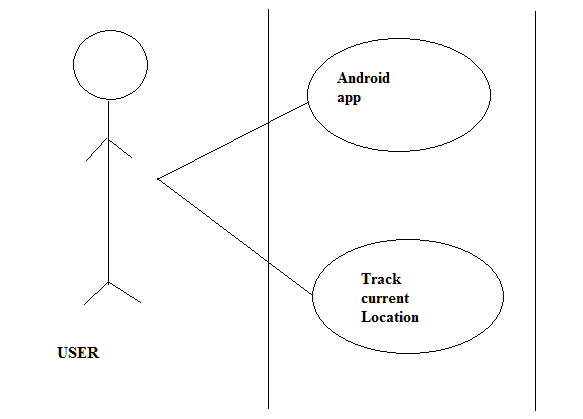
****

Fig 6.5 use case diagram

The fig 6.5 is a use case diagram ,in which user is directly access the android app and can access the his/her current location also

**6.4 SEQUENCE DIAGRAM**

Admin Section Database Server Android app

Update Signals

Storing a signals Load current

signal

to the server

Requests a

for signal through a

Delete, View Operations current location

Fetching a signal

from the server Di Displaying

marker signal Alert

**CHAPTER-7**

**METHODOLOGY**

**7.1 TECHONLOGIES USED**

**Android App Development:** It uses grade based build support to determine the order in which tasks can be run. It provides the facility to execute the codes in any language suitable to the user be it JAVA, C++, Python, etc. It uses Android-specific refactoring and quick fixes. It is based on JetBeans’s IntelliJ IDEA software. Android Studio is designed specifically for Android development. It allows the programmer to develop a android code in its IDE and then boot it on simple mobile phone for trial run.

**HTML5**: HTML5 is a software solution stack that defines the properties and behaviors of web page content by implementing a markup based pattern to it. HTML is the structure of the website. It is also being called the World Wide Web's markup language. HTML is used for structure and presenting content of the World Wide Web. An HTML document can provide information for browsers such as what style to use and where to get it. CSS is a style sheet language that describes the presentation of an HTML document. The first HTML4 emendation took place in 1997. However, after that the world’s internet traffic has increased by several hundred-folds, and constant HTML updates have become essential. Furthermore, websites like YouTube have gone mainstream and watching TV or listening to music online has become the norm and most web browsers have started to rely on plug-in and other resource hogging elements. As a result, the upgrading of HTML4 to HTML5 took place which established a proper handling for new contents. The last version of HTML5 was released in 2008. In mid-2012, a new editing team was introduced at the World Wide Web Consortium (W3C) to take care of creating an HTML5 recommendation and prepare a Working Draft for the next HTML version.

**CSS:** CSS is the language which controls the presentation of HTML and its latest version is CSS3, which is completely backward-compatible with the earlier versions of the CSS. Moreover, currently the CSS4 module is under development .It helps to give a different design to the HTML presentation and to change the colours, layout and fonts of the application. In addition, CSS is the style of the website and it allows using a specific syntax and rules to change how the element looks like in the page, the size of the font, colour, background, border, positions and so forth.

**JAVA**: Java is general-purpose programming language that is class-based, object-oriented and designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.Java applications are typically compiled to "byte code" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2018, Java was one of the most popular programming languages in use according to particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses.

**7.2 Tools used**

**Android Studio**: is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development). It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.2.1, which was released in October 2018.

**Notepad++:** Notepad++ is a [text editor](https://en.wikipedia.org/wiki/Text_editor) and [source code editor](https://en.wikipedia.org/wiki/Source_code_editor) for use with [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). It supports [tabbed](https://en.wikipedia.org/wiki/Tab_(GUI)) editing, which allows working with multiple open files in a single window. The project's name comes from the [C](https://en.wikipedia.org/wiki/C_(programming_language)) [increment operator](https://en.wikipedia.org/wiki/Increment_operator). Notepad++ is distributed as [free software](https://en.wikipedia.org/wiki/Free_software). At first the project was hosted on [SourceForge.net](https://en.wikipedia.org/wiki/SourceForge.net), from where it has been downloaded over 28 million times, and twice won the Source Forge Community Choice Award for Best Developer Tool. The project was hosted on [Tux Family](https://en.wikipedia.org/w/index.php?title=TuxFamily&action=edit&redlink=1) [[fr](https://fr.wikipedia.org/wiki/TuxFamily)] from 2010 to 2015; since 2015 Notepad++ has been hosted on [GitHub](https://en.wikipedia.org/wiki/GitHub). Notepad++ uses the [Scintilla editor component](https://en.wikipedia.org/wiki/Scintilla_(software)).

**7.3 Haversine formula**

The haversine formula determines the [great-circle distance](https://en.wikipedia.org/wiki/Great-circle_distance) between two points on a [sphere](https://en.wikipedia.org/wiki/Sphere) given their [longitudes](https://en.wikipedia.org/wiki/Longitude) and [latitudes](https://en.wikipedia.org/wiki/Latitude). Important in [navigation](https://en.wikipedia.org/wiki/Navigation), it is a special case of a more general formula in [spherical trigonometry](https://en.wikipedia.org/wiki/Spherical_trigonometry), the law of haversines, that relates the sides and angles of spherical triangles

range=0.03;

lat\_range = range/69.172;

lon\_range = range/(cos(latitude) \* 69.172));

min\_lat = number\_format(latitude - lat\_range);

max\_lat = number\_format(latitude + lat\_range);

min\_lon = number\_format(longitude - $lon\_range);

max\_lon = number\_format(longitude + $lon\_range);

**7.4 FLOW DIAGRAM OF PROJECT**

START STRART

Connect to server fetch nearest signal/points within 100m range ahead.

Fetch user GPI location

Is location available?

NO

YES

Wait for 20 sec and reconnect

Apply Haversine distance and find the points within 100m. range=======

If present

Connect to database to fetch

Find and place the signal on the map.

Wait for user location on change event

Call speech API to notify user with a voice notification.

YES

STOP

**CHAPTER-8**

**IMPLEMENTATION**

**8.1 Android Manifest**

The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code.

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="in.com.android.black\_spotapp">  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>  
 <uses-permission android:name="android.permission.INTERNET"/>  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/map\_icon"  
 android:label="@string/app\_name"  
 android:usesCleartextTraffic="true"  
 android:roundIcon="@mipmap/map\_icon"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="@string/google\_maps\_key" />  
  
 <activity  
 android:name=".MapsActivity"  
 android:label="@string/title\_activity\_maps">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>

**8.2 Maps Activity**

**8.2.1 Getting User Location**

In the Maps activity initially we check for the User GPS is enabled or not. If the GPS is not enabled its displays an error message. The current latitude and Longitude are displayed on the application. And that location’s Latitude and Longitude are sent to the server and check for the stored locations available if the stored locations are not available in the current location the message is displayed on the application "no stored location available".

LocationListener listener=new UserLocationListener();  
manager.requestLocationUpdates(Provider\_name, 60000, 1f, listener);  
if(current\_location!=null)  
{  
  
 latitude=current\_location.getLatitude();  
 longitude=current\_location.getLongitude();  
 Log.d("SMART","THE CURRENT LOC IS LAT LON: "+current\_location.getLatitude()+" "+current\_location.getLongitude());  
 Toast.makeText(MapsActivity.this,"LATITUDE IS : "+latitude+"\n LONGITUDE IS :"+longitude+"",Toast.LENGTH\_LONG).show();  
 //new track\_user().execute();  
}  
else  
{  
 Toast.makeText(MapsActivity.this,"NO STORED LOCATION AVAILABLE",Toast.LENGTH\_LONG).show();  
}  
  
  
manager = (LocationManager) this.getSystemService(Context.LOCATION\_SERVICE);  
isGpsEnabled = manager.isProviderEnabled(LocationManager.GPS\_PROVIDER);  
isNetworkEnabled = manager.isProviderEnabled(LocationManager.NETWORK\_PROVIDER);  
if (isGpsEnabled || isNetworkEnabled) {  
 if (isGpsEnabled) {  
 Provider\_name = LocationManager.GPS\_PROVIDER;  
 }  
 if (isNetworkEnabled) {  
 Provider\_name = LocationManager.NETWORK\_PROVIDER;  
 }  
} else {  
 Toast.makeText(MapsActivity.this, "NO LOCATION DATA AVAILABLE", Toast.LENGTH\_LONG).show();  
}  
  
Log.d("SMART", "THE PROVIDER NAME IS : " + Provider\_name);  
if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  
 return;  
}  
current\_location = manager.getLastKnownLocation(Provider\_name);  
  
Log.d("SMART","THE CURRENT LOC IS : "+current\_location);

**8.2.2 Fetching Data to and from server**

The current location is sent to the server “onPreExecute” method and the stored locations are fetched back from the server. All this things are done in the background by using the “doInBackground” method. The “doInBackground” method is completely done in background it won’t affect the UI.

private class track\_user extends AsyncTask<Void, Void, Void> {  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
 pDialog = new ProgressDialog(MapsActivity.this);  
 pDialog.setMessage("Tracking Location..");  
 pDialog.setCancelable(false);  
 pDialog.show();  
 }  
 @Override  
 protected Void doInBackground(Void... arg0) {  
 HttpClient httpclient = new DefaultHttpClient();  
 HttpPost httppost = new HttpPost(URL\_TRACK);  
 try {  
 // Add your data  
 List <NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(1);  
 nameValuePairs.add(new BasicNameValuePair("lat",Double.toString(latitude)));  
 nameValuePairs.add(new BasicNameValuePair("lon",Double.toString(longitude)));  
 httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
 ResponseHandler<String> responseHandler = new BasicResponseHandler();  
 String response = httpclient.execute(httppost, responseHandler);  
 Log.d("SMART","URL RES IS : "+response);  
 if(response.equals("NONE"))  
 URL\_Response="NONE";  
 else  
 {  
 URL\_Response=response;  
 }  
 }  
 catch (ClientProtocolException e)  
 {  
 Log.d("SMART","CPE response " + e.toString());  
 // TODO Auto-generated catch block  
 URL\_Response="INVALID";  
 }  
 catch (IOException e)  
 {  
 Log.d("SMART","IOE response " + e.toString());  
 // TODO Aut return 0  
 URL\_Response="INVALID";  
 }  
 return null;  
 }

**8.2.3 Displaying Marker**

Once the Stored location are found near to the current location of the user the nearby stored on location’s Latitude and Longitude the marker will be displayed according to its type. The markers are Signal, Hospital and Speed Limit different markers are displayed.

protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
 if (pDialog.isShowing())  
 pDialog.dismiss();  
 Log.d("SMART","the URL VALU IS : "+URL\_Response);  
 Toast.makeText(MapsActivity.this, "the URL VALU IS : "+URL\_Response, Toast.LENGTH\_LONG).show();  
 if(URL\_Response.equals("NONE"))  
 {  
 Toast.makeText(MapsActivity.this, "NO LOCATION FOUND", Toast.LENGTH\_LONG).show();  
 }  
 else  
 {  
 Toast.makeText(MapsActivity.this, "LOCATION FOUND : "+URL\_Response, Toast.LENGTH\_LONG).show();  
 String [] users\_data=URL\_Response.split(";");  
 map.clear();  
 v\_data="";  
 v\_data=Integer.toString(users\_data.length)+" Points Have Been Detected";  
 for(int i=0;i<users\_data.length;i++)  
 {  
 final String [] places=users\_data[i].split("#");  
 v\_data+=Integer.toString(i+1)+" "+places[1]+" Ahead Name is "+places[0]+" ";  
 Log.d("TRACK","V DATA"+v\_data);  
 Log.d("TRACK","message is :"+places[2]+" "+places[3]);  
 final LatLng f\_pos=new LatLng(parseFloat(places[2]), parseFloat(places[3]));  
 switch(places[1])  
 {  
 case "Hospitals" :  
 Thread thread1 = new Thread(new Runnable(){  
 @Override  
 public void run() {  
 try {  
 if(places[1].indexOf("Hospitals")>=0)  
 {  
 runOnUiThread(new Runnable(){  
 public void run(){  
 map.addMarker(new MarkerOptions().position(f\_pos)  
 .title(places[0]+"("+places[1]+")").icon(bitmapDescriptorFromVector(MapsActivity.this,R.drawable.hospital\_24dp)));  
 }  
 });  
 }  
 }  
 catch(Exception e)  
 {  
 Log.d("LBA","the data "+e);  
 }  
 }  
 });  
 thread1.start();  
 break;  
 case "Signals" :  
 Thread thread2 = new Thread(new Runnable(){  
 @Override  
 public void run() {  
 try {  
 if(places[1].indexOf("Signals")>=0)  
 {  
 runOnUiThread(new Runnable(){  
 public void run(){  
 map.addMarker(new MarkerOptions().position(f\_pos).title(places[0]+"("+places[1]+")").icon(bitmapDescriptorFromVector(MapsActivity.this,R.drawable.signal\_24dp)));  
 }  
 });  
 }  
 }  
 catch(Exception e)  
 {  
 Log.d("LBA","the data "+e);  
 }  
 }  
 });  
 thread2.start();  
 break;  
 case "SpeedLimit" :  
 Thread thread3 = new Thread(new Runnable(){  
 @Override  
 public void run() {  
 try {  
 if(places[1].indexOf("SpeedLimit")>=0)  
 {  
 runOnUiThread(new Runnable(){  
 public void run(){  
 map.addMarker(new MarkerOptions().position(f\_pos)  
 .title(places[0]+"("+places[1]+")").icon(bitmapDescriptorFromVector(MapsActivity.this,R.drawable.alpha\_s\_circle)));  
 }  
 });  
 }  
 }  
 catch(Exception e)  
 {  
 Log.d("LBA","the data "+e);  
 }  
 }  
 });  
 thread3.start();  
 break;  
 case "Tolls" :  
 Thread thread4 = new Thread(new Runnable(){  
 @Override  
 public void run() {  
 try {  
 if(places[1].indexOf("Tolls")>=0)  
 {  
 runOnUiThread(new Runnable(){  
 public void run(){  
 map.addMarker(new MarkerOptions().position(f\_pos)  
 .title(places[0]+"("+places[1]+")").icon(bitmapDescriptorFromVector(MapsActivity.this,R.drawable.barn)));  
 }  
 });  
 }  
 }  
 catch(Exception e)  
 {  
 Log.d("LBA","the data "+e);  
 }  
 }  
 });  
 thread4.start();  
 break;  
 }  
 }  
 }

**8.2.4 Voice alert**

After displaying the markers on the stored location the voice alert will be executed. The voice alert will be done by using “ttospeech” function.

int is\_speaking=0;  
public void start\_speech(final String VoiceData)  
{  
 stop\_speech\_working();  
 ttospeech=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {  
 @Override  
 public void onInit(int status) {  
 if(status != TextToSpeech.ERROR) {  
 ttospeech.setLanguage(Locale.UK);  
 }  
 }  
 });  
 final Thread logoTimer = new Thread() {  
 public void run() {  
  
 try {  
 sleep(5000);  
 String toSpeak = VoiceData;  
 ttospeech.speak(toSpeak, TextToSpeech.QUEUE\_FLUSH, null);  
 sleep(12000);  
 is\_speaking=0;  
  
 } catch (InterruptedException e) {  
 // TODO Auto-generated catch block  
 e.printStackTrace();  
 }  
 }  
 };  
 logoTimer.start();  
}  
  
public void stop\_speech\_working()  
{  
 if(logoTimer!=null) {  
 logoTimer.interrupt();  
 }  
 if(ttospeech!=null)  
 ttospeech.stop();  
}

**8.3 RESULTS**

**8.3.1 SNAPSHOTS OF ADMIN UI**

The figure shows the homepage of the admin user interface.

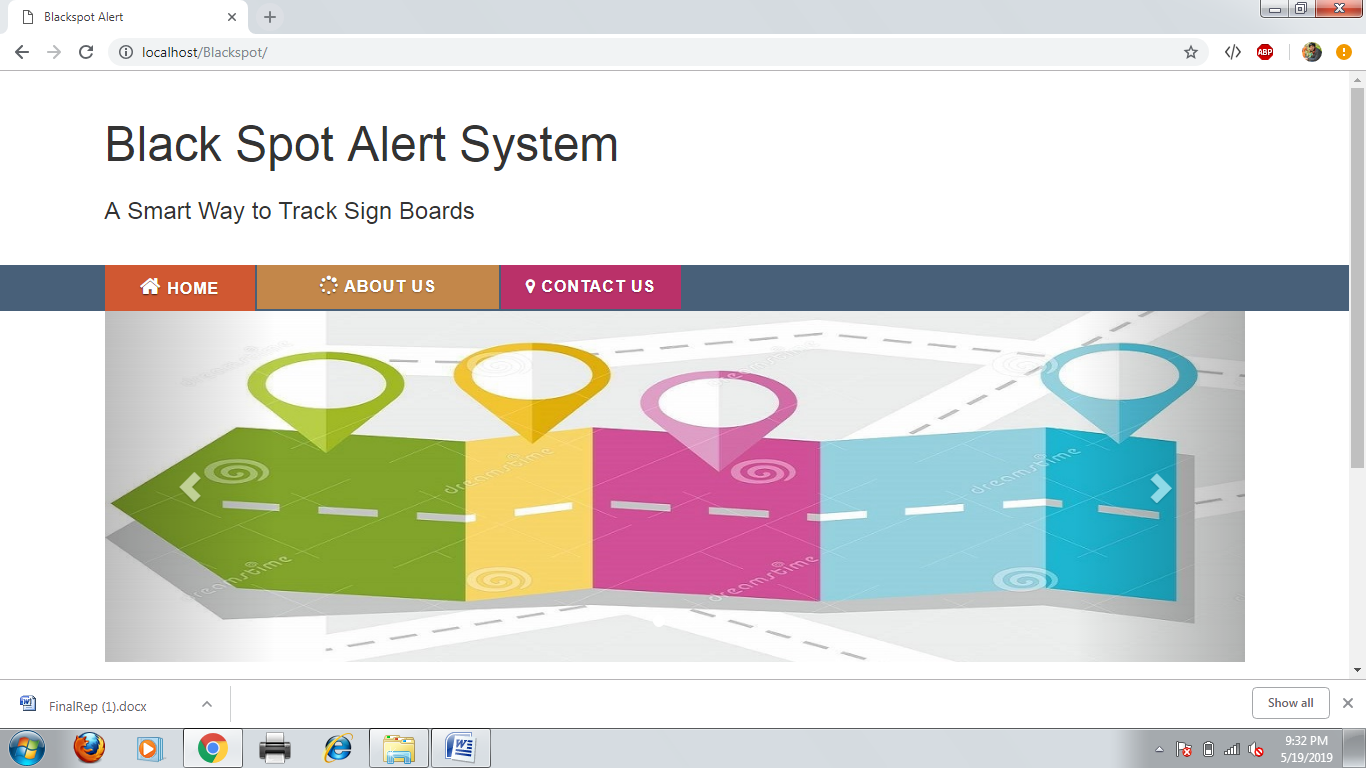
****

Fig 8.1 Homepage

The figure shows the login form for the admin to login using admin email and password.

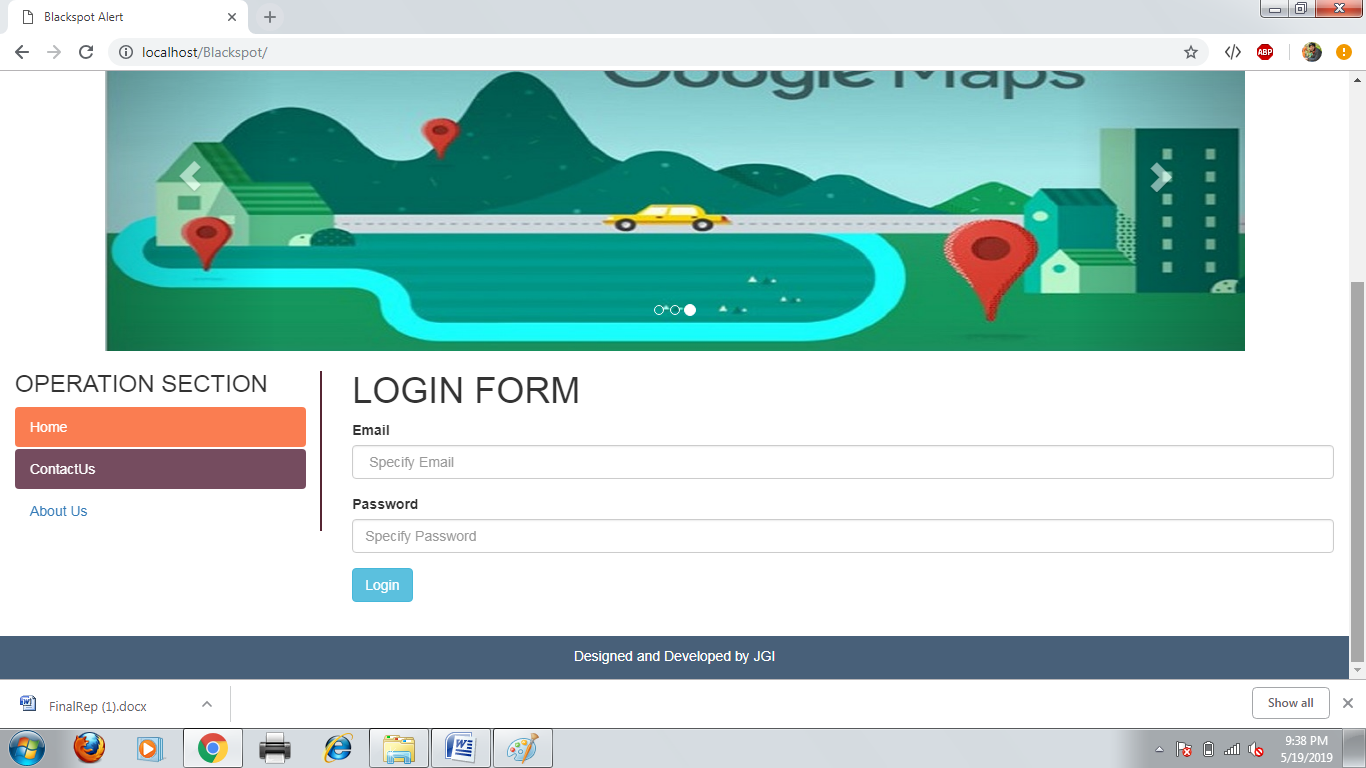
****

Fig 8.2 Login page for admin

The figure shows the adding a new points using a point name ,specific type and its latitude and longitude.

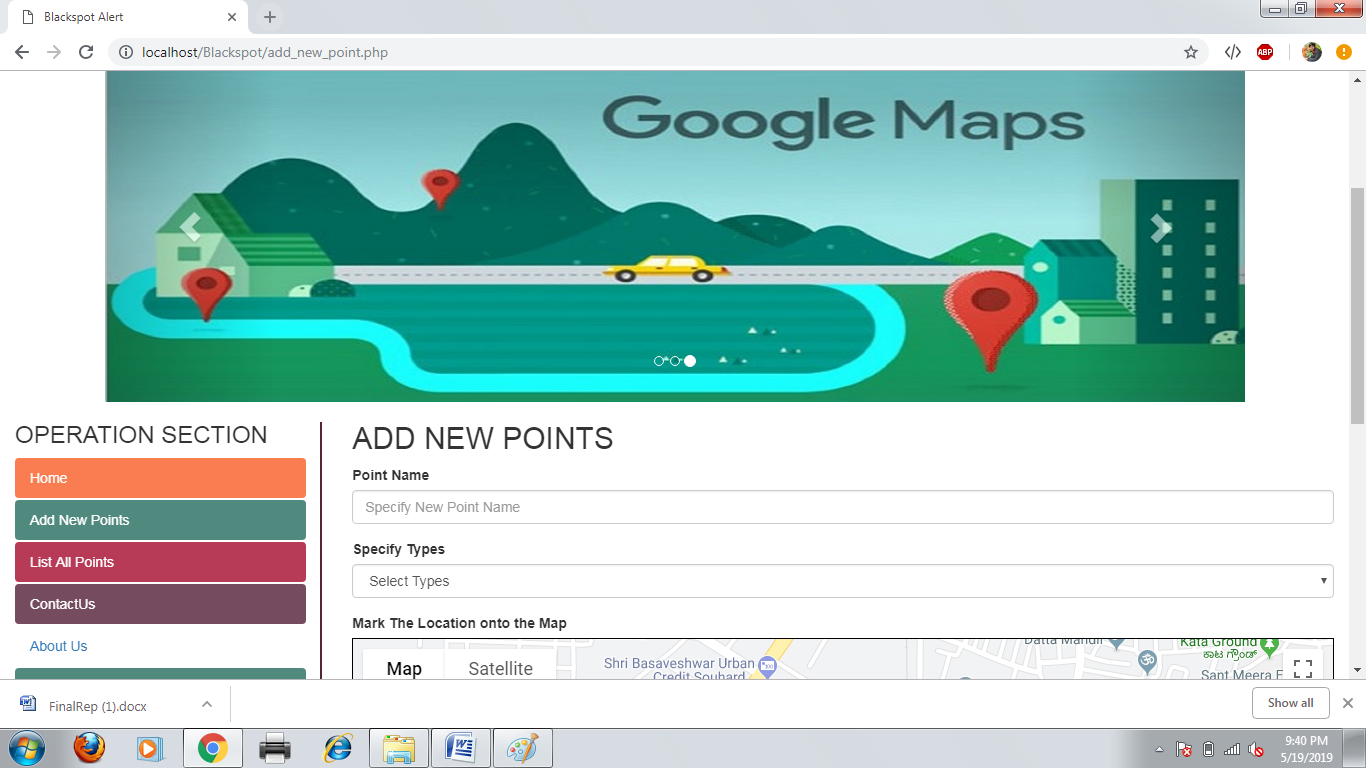


Fig 8.3 Add new points page

The figure shows the register of new points using a Google map and point latitude and longitude.

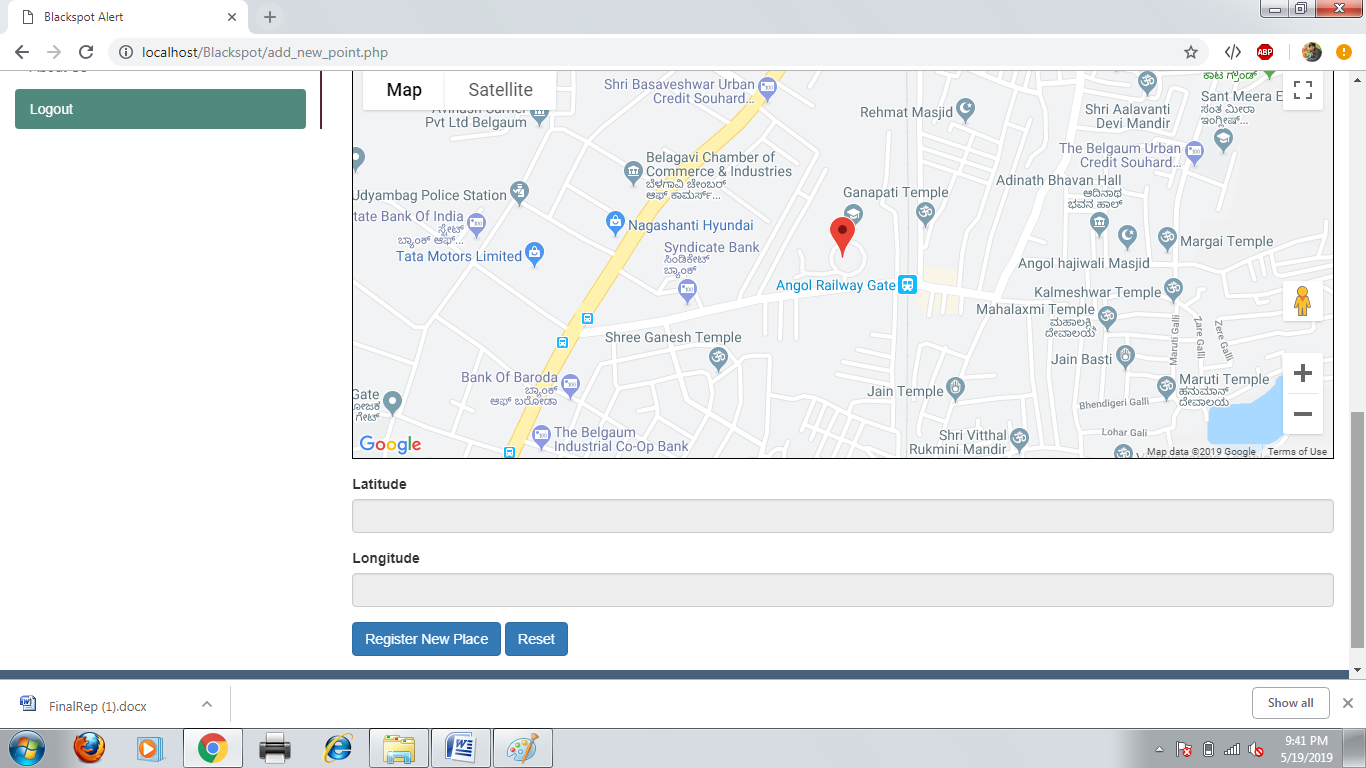


Fig 8.4 Register of new sign board

The figure shows the list of all registered points which can be viewed ,edit and delete of points.

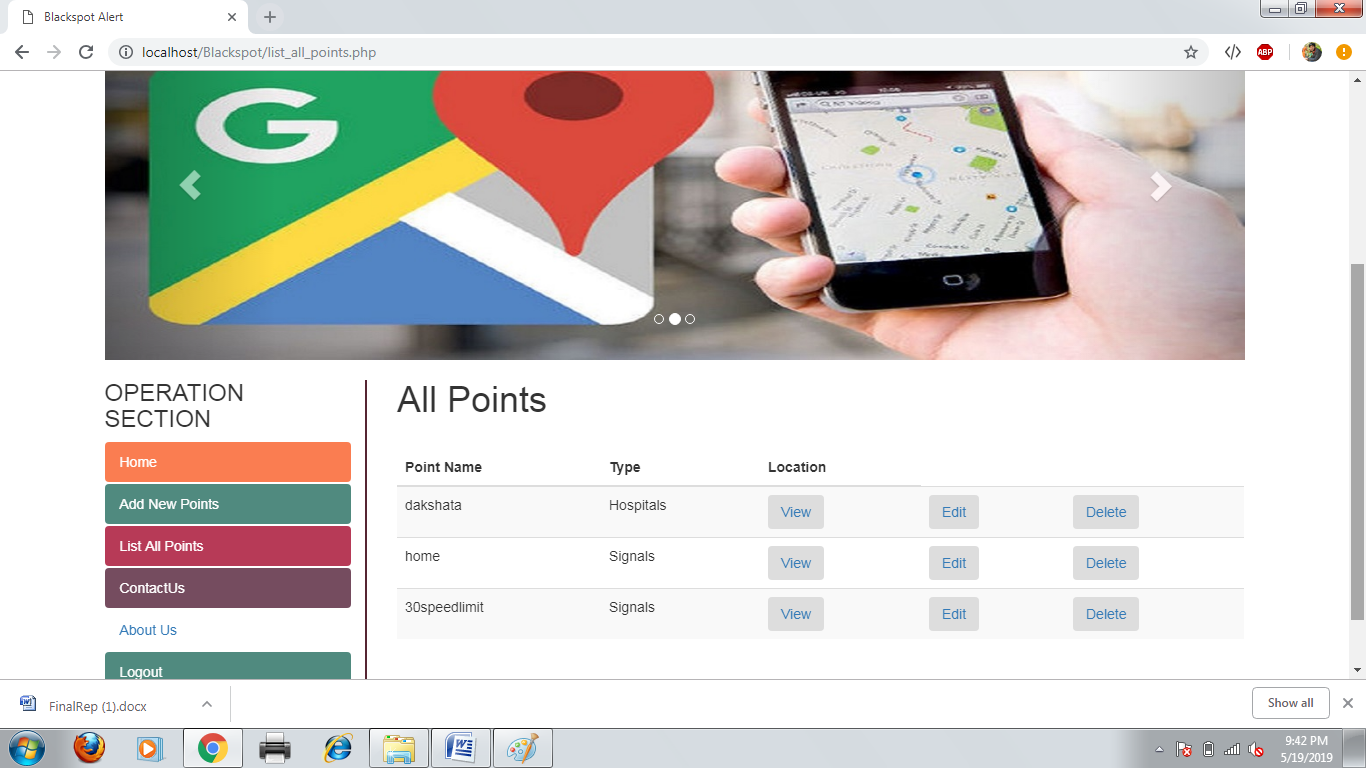


Fig 8.5 List of all points page

The figure shows the contact of jain college of engineering address and its website.

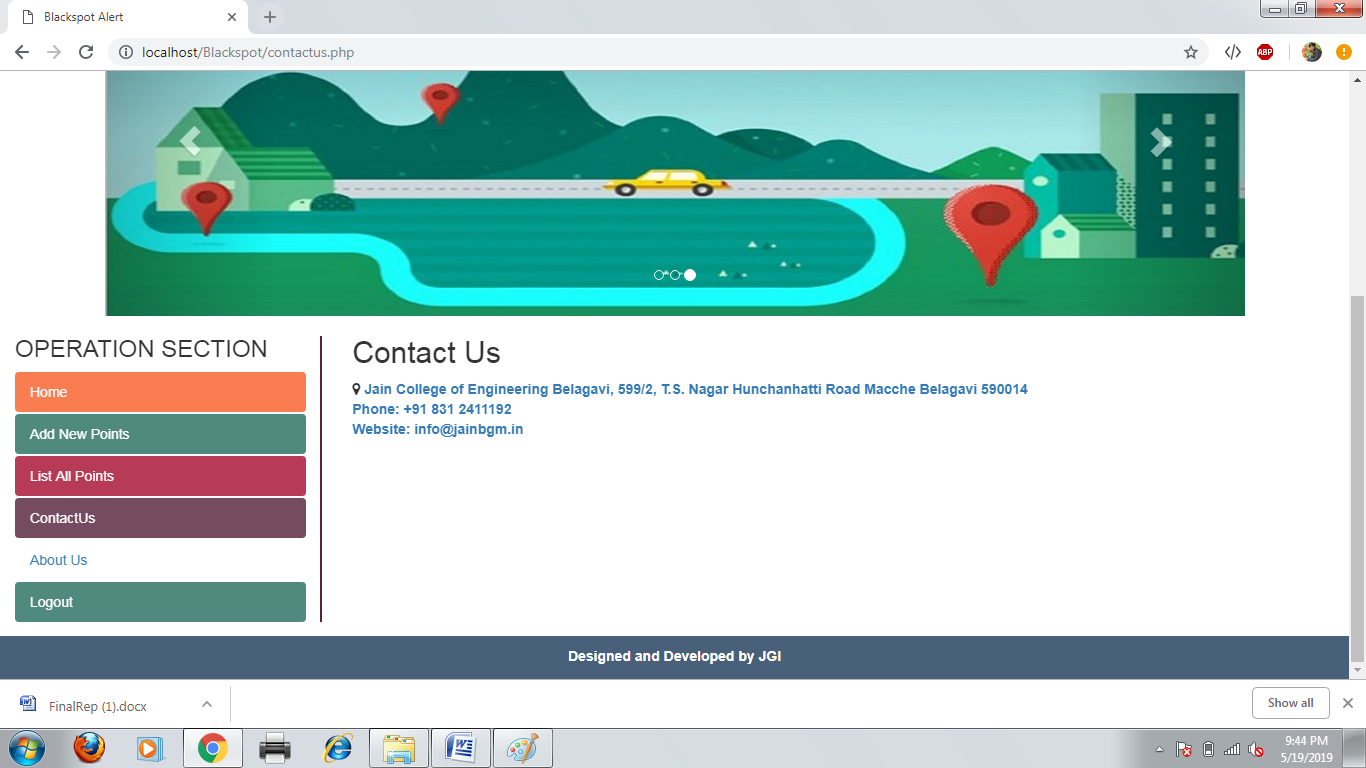
****

Fig 8.6 contact us page

**8.3.2 SNAPSHOTS OF USER APP**

The figure shows the tracking of user current location on the Google map with a marker.

****

Fig 8.7 Tracking current location

The figure shows the marker of different signals with a specific name on the signal marker.

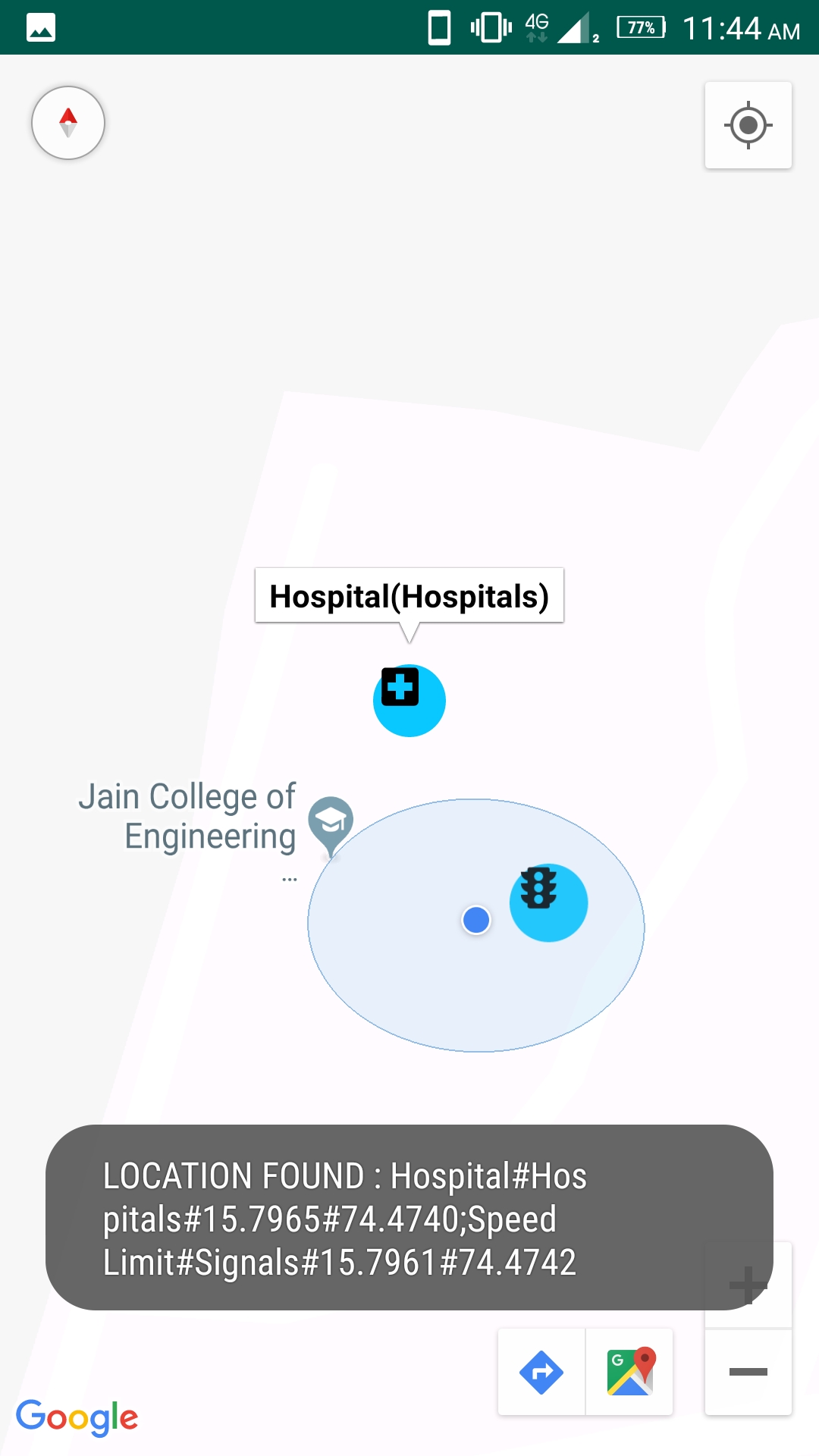
****

Fig 8.8 Displaying the marker

The figure shows the speed limit marker along with a name of that signal.

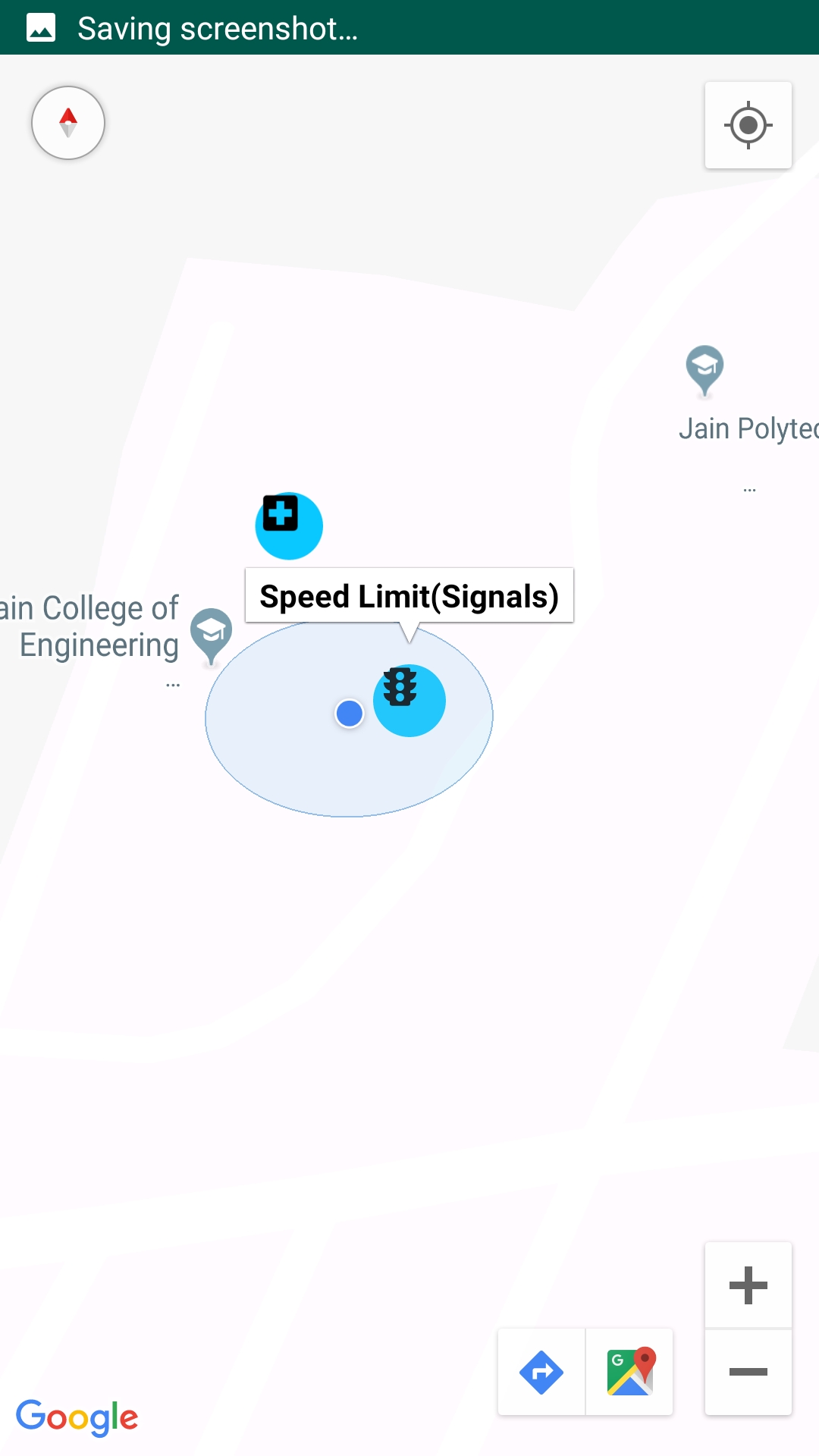


Fig 8.9 displaying a speed limit signal

**CHAPTER-9**

**TESTING**

The development of software system involves a series of production activities where opportunities for injection human facilities are enormous. Error may begin to occur at very inception of the process where the objectives, may be erroneously or imperfectly specified, as well as in later design and development stages. Because of human inability to perform and communicate in perform software development is accompanied by quality assurance activity.

Software testing is a critical element of software assurance and represents the ultimate review of specification, design and coding.

**9.1 Test Reports**

Here we specify all the test cases that are used for system testing. The different conditions that need to be tested along with the test cases used for testing those conditions and the expected outputs are given. The goal is to test the different functional requirements, as specified in the requirements document. Test cases have been selected for both valid and invalid inputs. In this project I have used three phases testing unit testing, integration testing and system testing respectively.

**9.1.2 Test cases for Admin Login Page**

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | **Expected output** | **Output** | **Test Result** |
| For logging in:  Username: invalid Password: invalid | Error Message | Error Message | PASS |
| For logging in:  Username: invalid Password: valid | Error Message | Error Message | PASS |
| For logging in: Username: valid  Password: invalid | Logged in | Logged in | FAIL |
| For logging in: Username: invalid  Password: invalid | Logged in | Logged in | FAIL |

**Table 9.1:** Test Cases for Admin Login page

**9.1.2 Test Cases for Add new point form**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case No | Input | Expected Output | Output | Test Result |
| 1 | Point Name: valid  Type: valid  Latitude: valid  Longitude: valid | Point Added Successfully | Point Added Successfully | PASS |
| 2 | Point Name: Invalid  Type: Invalid  Latitude: Invalid  Longitude: valid | Error Message | Error Message | PASS |
| 3 | Point Name: Invalid  Type: Invalid  Latitude: valid  Longitude: valid | Error Message | Error Message | PASS |
| 4 | Point Name: Invalid  Type: valid  Latitude: valid  Longitude: valid | Error Message | Error Message | PASS |
| 5 | Point Name: Invalid  Type: Invalid  Latitude: valid  Longitude: Invalid | Error Message | Error Message | PASS |
| 6 | Point Name: Invalid  Type: valid  Latitude: Invalid  Longitude: valid | Error Message | Error Message | PASS |
| 7 | Point Name: valid  Type: valid  Latitude: valid  Longitude: Invalid | Error Message | Error Message | PASS |
| 8 | Point Name: Invalid  Type: Invalid  Latitude: Invalid  Longitude: Invalid | Error Message | Point Added Successfully | FAIL |
| 9 | Point Name: Invalid  Type: Invalid  Latitude: Invalid  Longitude: valid | Error Message | Point Added Successfully | FAIL |
| 10 | Point Name: Invalid  Type: Invalid  Latitude: valid  Longitude: valid | Error Message | Point Added Successfully | FAIL |
| 11 | Point Name: Invalid  Type: valid  Latitude: valid  Longitude: valid | Error Message | Point Added Successfully | FAIL |
| 12 | Point Name: Empty  Type: valid  Latitude: valid  Longitude: valid | Display message “Please fill out the empty field” | Displayed message “Please fill out the empty field” | PASS |
| 13 | Point Name: Empty  Type: valid  Latitude: valid  Longitude: valid | Display message “Please fill out the empty field” | Displayed message “Please fill out the empty field” | PASS |
| 14 | Point Name: Empty  Type: valid  Latitude: valid  Longitude: valid | Display message “Please fill out the empty field” | Displayed message “Please fill out the empty field” | PASS |
| 15 | Point Name: Empty  Type: Empty  Latitude: valid  Longitude: valid | Display message “Please fill out the empty field” | Displayed message “Please fill out the empty field” | PASS |
| 16 | Point Name: Empty  Type: valid  Latitude: valid  Longitude: valid | Display message “Please fill out the empty field” | Display message “Please fill out the empty field” | PASS |

**Table 9.2:** Test Cases for Add new point page

**9.1.3 Test Cases for Android app**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Expected output** | **Output** | **Test Result** |
| The Mobile Data is  Turned off  GPS is turned Off | APP closes automatically | APP closes automatically | PASS |
| The Mobile Data is  Turned ON  GPS is turned ON | App opens smoothly | App opens smoothly | PASS |
| The Mobile Data is  Turned off  GPS is turned Off | APP closes automatically | App opens smoothly | FAIL |
| The Mobile Data is  Turned ON  GPS is turned ON | App opens smoothly | APP closes automatically | FAIL |
| The Mobile Data is  Turned off  GPS is turned ON | APP closes automatically | APP closes automatically | PASS |
| The Mobile Data is  Turned ON  GPS is turned OFF | APP closes automatically | APP closes automatically | PASS |
| The Mobile Data is  Turned ON  GPS is turned ON.  App is opened | Map Opens and Displays “Tracking Location” | Map Opens and Displays “Tracking Location” | PASS |
| After Tracking the Location | Display The Current Latitude and Longitude | Display The Current Latitude and Longitude | PASS |
| There is no Stored locations in the Current Location | Displays “NO LOCATIONS FOUND” | Displays “NO LOCATIONS FOUND” | PASS |
| There is no Stored locations in the Current Location | Displays “NO LOCATIONS FOUND” | Displays “LOCATION FOUND” | FAIL |
| Stored location found in current location. | Displays marker and Alert through voice | Displays marker and Alert through voice | PASS |
| Stored location found in current location. | Displays marker and Alert through voice | Only Displays marker. No voice alert. | FAIL |

**Table 9.3:** Test Cases for Android app

**Conclusion**

Roadways are the most important mode of the transportation currently. And there are some Accidental zones and speed limits are identified by the government, these are known as the black spots which need to take care of. So we developed an Android app which incorporates Google map and the identified Black spots are displayed on the map with unique marker and also alert through the Voice notes. The voice alerts are even helpful for the user even in night.

So our App will be helpful in preventing accidents and even for night travelers it will help with voice alert.

**FUTURE SCOPE**

**GETTING DATA THROUGH GOOGLE PAID API:**

Currently we are using free version of Google API. And inserting the location data that is latitude and longitude manually to the database and then we are extracting that to the android app. With Google paid version of API, we can have the data of the accidental zones and the sign boards latitude and longitude and we can alert on that specific location about the black spot.

**NAVIGATION**

Currently In our app. As the user moving on the stored locations around the current locations are displaying. So we can do the same with navigation while user navigating through a particular route all the black spot in that road will be displayed and even the number of black spots also we can get. so in the navigation also we can use this technique.

**References**

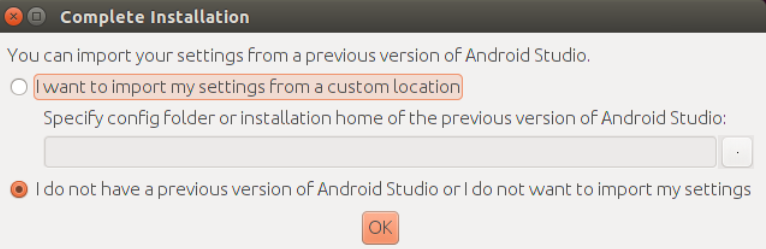
1. Ajinkya Patankar, Prof.S.G Walunj, Anand Pimple, Darshan Kadhane,” Blackspot Warning System Identification and Analysis”, VJER-Vishwakarma Journal of Engineering Research Volume 1 Issue 4, December 2017.
2. Abdalwhab Bakheet, Ahmed Abd Almahmoud and Wigdan Ahmed.” ANDROID MAPPING APPLICATION”, Dhinaharan Nagamalai et al. (Eds) : CSE, DBDM, CCNET, AIFL, SCOM, CICS, CSIP – 2014 pp. 11–22, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.4402.
3. Accidents black spots on highways and their low cost remedial measures.By Hafeez & M. A. Kamal Department of Civil Engineering, University of Engineering and Technology, Taxila, Pakistan.
4. Maen Ghadia , Árpád Törökb , “Comparison Different Black Spot Identification Methods” , Maen Ghadi et al. / Transportation Research Procedia 27 (2017) 1105–1112.
5. Identification and analysis of accident black spots on nh147 using gis Harsh Naik Nirma University
6. <https://en.wikipedia.org/wiki/Android_software_development>
7. <https://en.wikipedia.org/wiki/Android_Studio>
8. <https://developers.google.com/maps/documentation/android-sdk/map-with-marker>
9. <https://www.sciencedirect.com/science/article/pii/S2352146517310013>

**APPENDIX**

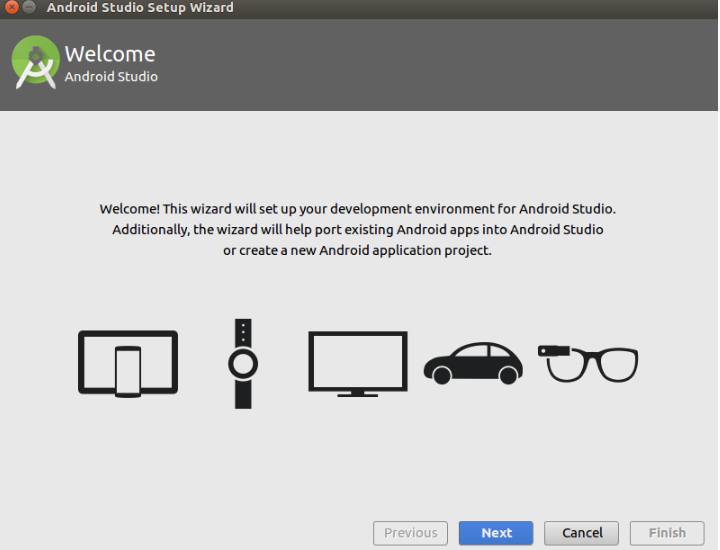
**Download and Installation of Android Studio:**

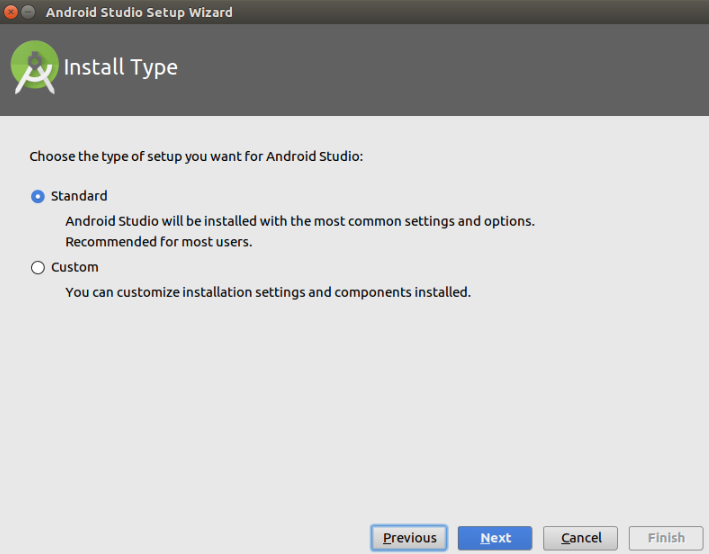
Installation for Windows is simple, just launch the .exe you downloaded. On Max OSX drag and drop Android Studio into the Applications folder.

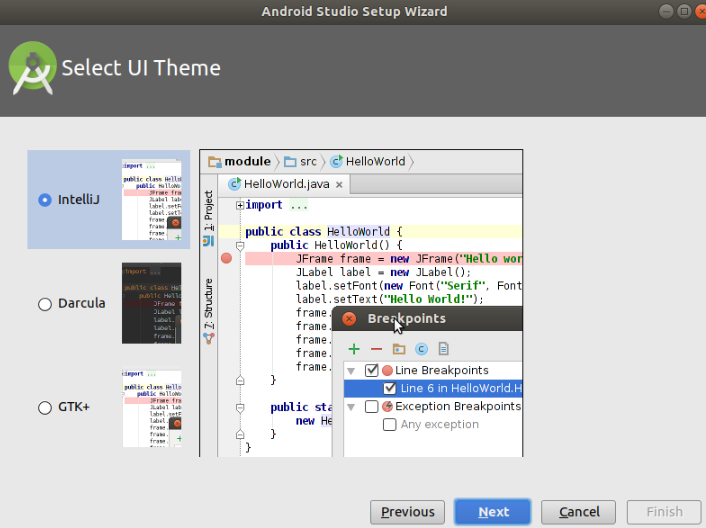
The first time you start a new Android Studio installation, you have the option to import your existing settings.

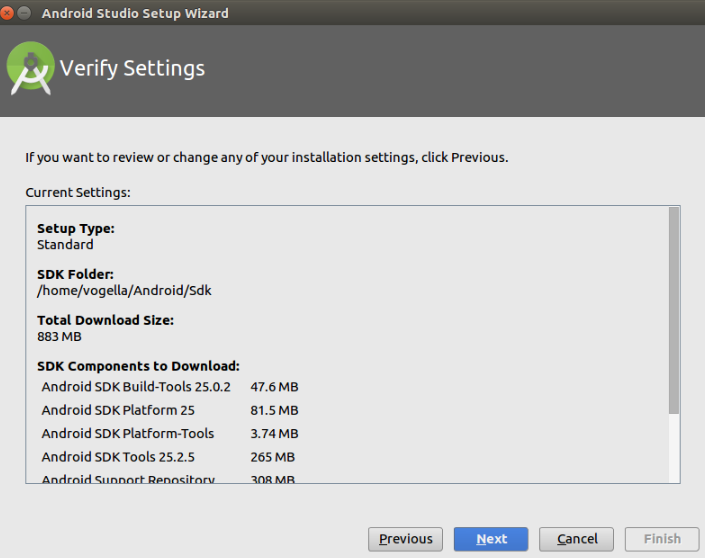


Afterwards click through the setup guide.

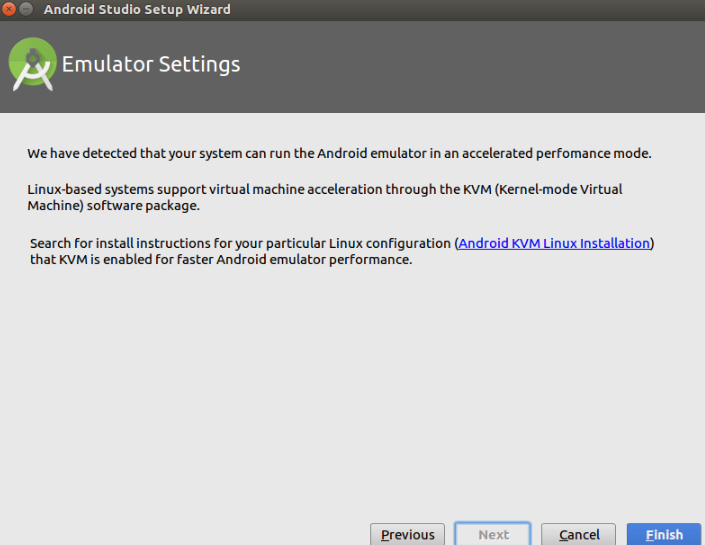








Once you reach the last page, press the Finish button.

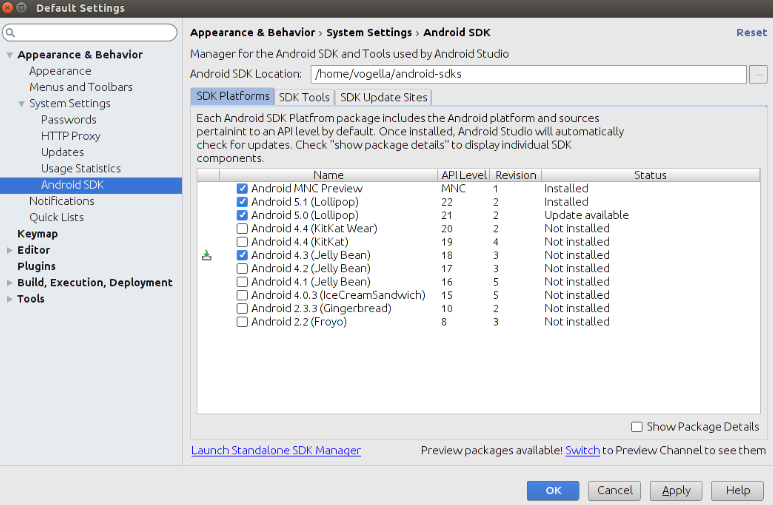


**Installation of Android SDKs:**

The Android SDK manager allows you to install and delete specific Android OS versions.

Select Tools> Android>  SDK Manager or the SDK Manager icon in the toolbar of Android Studio to open the Android SDK manager.

In the Android SDK manager select the version of desired Android version the tree and press the Install button. The following screenshot shows the selection for the API 18 version of Android.



Press OK button to start the installation.

The SDK Platforms tab is used to install API versions, which the SDK Tools is used to install the development tools.

**Creating New Android Projects:**

Press the Start a new Android Studio project link to get started. Alternatively you can select the File >New Project entry from the menu, if you already created a project earlier.

