TOURIST GUIDE

Project report submitted

In partial fulfilment of the requirements for the degree of

Bachelor of Computer Applications

by

Amul Bakshish Singh (1601060005)

Sandipta Subir Khare (1601060002)

Under the Supervision Of

Ms. Anjali Singh

Assistant Professor, CSE

School of Engineering and Technology

K. R. Mangalam University, Gurugram, Haryana, India

May 2019

CERTIFICATE

It is certified that the work contained in the project report titled "Tourist Guide" by the following students:

Name of the Student	**Roll Number**
AMUL BAKSHISH SINGH	1601060005
SANDIPTA SUBIR KHARE	1601060002

has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Ms. Anjali Singh,

Assistant Professor,

Computer Science and Engineering,

K R Mangalam University, Gurugram, Haryana, India

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of the Student	**Roll Number**	**Signature**
Amul Bakshish Singh	1601060005	
Sandipta Subir Khare	1601060002	

Date: _____

APPROVAL SHEET

This project report entitled **Tourist Guide** by **Amul Bakshish Singh and Sandipta Subir Khare** is approved for the degree of **Bachelor of Computer Applications, School of Engineering and Technology**

Dean, SOET	Supervisor
Prof. (Dr.) Ranjit Varma	Ms. Anjali Singh, Assistant Prof., CSE, SOET

Date: _____

Place: _____

ACKNOWLEDGEMENT

It gives me immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide **Ms. Anjali Singh, Assistant Prof., Computer Science and Engineering, SOET**, for her valuable guidance, encouragement and help for completing this work. Her useful suggestions for this whole work and cooperation are sincerely acknowledged.

We would like to express our sincere thanks to **Prof. (Dr.) Ranjit Varma, Dean, SOET** for giving us the opportunity to undertake this project. We are also grateful to all our teachers and the Computer Science and Engineering department, SOET, for the constant support and guidance.

We also wish to express our indebtedness to our parents as well as our family member whose blessings and support always helped us to face the challenges ahead.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

Place: Gurugram Amul Bakshish Singh (1601060005)

Date Sandipta Subir Khare (1601060002)

ABSTRACT

Tourist Guide is the software aimed at providing a wide range of access to the clients to get know the locality of their nearby places in just seconds by just selecting the type of place they want to visit.

The clients can even maintain their favourite place that they frequently visit and can save them for further usage.

The app will use a Google account of a client to maintain their favourite places in a database.

All the Data is collected in the app is from Google so the client can be satisfied about the place they are visiting will be genuine. The app also provide the navigation to the place the client want to visit using Google maps client just need to click the button and the app will take you to the Google map.

The app also provides the location and number of the place so the client can directly reach the place to clear his/her query.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Overview

Tourist Guide application is a Place to Visit application useful to find the Popular places to visit in India. Following things can be done with the application:

- User will also be able view a geographical map of the area.
- The data fed into the application will be used to search for the Popular Places in India and same as place to visit in that location.
- The user will only have to provide a radius of area to be searched.

The project is on the Tourist Guide Android Application. In this Application we would have few things and that are:

- **Places to Visit in India:** When you click on Place to Visit in India, with the help of Google map you would get many places to visit such as Amritsar, Himachal Pradesh, etc.
- **Popular area to Visit in the Area which you had chosen in Places to Visit in India.** In other words, it means when you choose the option in Places to visit in India, there are many places to visit in that location. For example, if you choose the option Amritsar in Places to Visit in India, the Popular Area to Visit would show many option such as Golden Temple, Wagah Border, etc.

1.2 Different Phases of The Development Process:

1.1.2 Requirement Analysis:

Requirement analysis is done in order to understand the problem the software system is to solve. The problem could be automating an existing manual process, developing a new automated system, or a combination of the two. The emphasis in requirements analysis is on identifying what is needed from the system, not how the system will achieve its goals. There are atleast two parties involved in the software development-a client and a developer. The developer has to develop the system to satisfy the client's needs. The developer does not understand the client's problem domain, and the client does not understand the issues involved

in the software systems. This causes a communication gap, which has to be adequately bridged during requirements analysis.

Software Requirement Specification

If a document that completely describes what the proposed software should do without describing how the software will do it, The basic goal of requirements specific to produce the requirements, which describes the complete external behaviour of the proposed software. However, producing is said to be done.

The basic limitation is that the user's need keep on changing as the environment in which system was to function changes with time. This happens more in complex application where all the needs may not be known to any set of people during the requirement phase. Types of error in this specification can clearly be avoided or at least considerably, by properly performing the requirement phase.

As many times a developer or administrator does not know what a user and the supplier on what the software product will do. This is the only document which tells the time agreement and procedure of complete project as well as what a user wants.

There are some steps which are considered in the Software Requirement Specification Such As

- Information Description
- Function Description
- Behavioural Description
- Validation Checks and Criteria
- Reference

1.2.2 Software Design

The purpose of the design phase is to plan a solution of the problem specified by the requirements documents. This phase is the first step in moving from the problem domain to the solution domain. Starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the later phases, particularly testing and maintenance.

The design activity is divided into two phases: System Design and Detailed Design. In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules.

1.2.3 Coding

The goal of the coding phase is to translate the design of the system into code in a given programming language. Hence during coding, the focus should be on developing programs that are easy to read and understand, and not simply on developing programs that are easy to write.

1.2.4 Testing

Testing is the major quality control measure used during software development. Its basic function is to detect errors in the software. Testing not only uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of the testing is to uncover requirement, design and coding errors in the programs. Therefore, different levels of testing are used. Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. The output of the testing phase is the test report and the error report. Test report contains the set of test cases and the result of executing the code with these test cases. The error report describes the errors encountered and the action taken to remove the errors.

1.2.5 Operation & maintenance phase

Software maintenance is a task that every development group has to face, when the software is delivered to the customer's site, installed and is operational. Software maintenance is a very broad activity that includes error correction, enhancement of capabilities, deletion of obsolete capabilities and optimization.

1.3 Android Software Development

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit, but other development tools are available. As of October 2012, more than 700,000 applications have been developed for Android, with over 25 billion downloads. A June 2011 research indicated that over 67% of mobile developers used the platform, at the time of publication. In Q2 2012; around 105 million

units of Android smart phones were shipped which acquires a total share of 68% in overall smart phones sale till Q2 2012.

Fig. No. 1.1

1.3.1 Software development tools

1.3.1.1 Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools.[6] These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later; for the moment one cannot develop Android software on Android itself. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

1.3.1.2 Android Debug Bridge

The Android Debug Bridge (ADB) is a toolkit included in the Android SDK package. It consists of both client and server-side programs that communicate with one another. The ADB is typically accessed through the command-line interface.

The format for issuing commands through the ADB is typically:

adb [-d|-e|-s <serialNumber>] <command>

In a security issue reported in March 2011, ADB was targeted as a vector to attempt to install a rootkit on connected phones using a "resource exhaustion attack".[13]

1.3.1.3 Fast boot

Fast boot is a diagnostic protocol included with the SDK package used primarily to modify the flash file system via a USB connection from host computer. It requires that the device be started in a boot loader or Second Program Loader mode in which only the most basic hardware initialization is performed. After enabling the protocol on the device itself, it will accept a specific set of commands sent to it via USB using a command line. Some of most commonly used fast boot commands include:

- Flash - Rewrites a partition with a binary image stored on the host computer.
- Erase - Erases a specific partition.
- Reboot - Reboots the device into either the main operating system, the system recovery partition or back into its boot loader.
- Devices - Displays a list of all devices (with the serial number) connected to the host computer.
- Format - Format a specific partition. The file system of the partition must be recognized by the device.

1.3.1.4 Native development kit

Libraries written in C and other languages can be compiled to ARM, MIPS or x86 native code and installed using the Android Native Development Kit. Native classes can be called from Java code running under the Dalvik VM using the System. loadLibrary call, which is part of the standard Android Java classes.

Complete applications can be compiled and installed using traditional development tools.[16] The ADB debugger gives a root shell under the Android Emulator which allows native ARM, MIPS or x86 code to be uploaded and executed. Native code can be compiled using GCC on a standard PC. Running native code is complicated by Android's use of a non-standard C library (libc, known as Bionic). The graphics library that Android uses to arbitrate and control access to this device is called the Skia Graphics Library (SGL), and it has been released under an open source license. Skia has back ends for both Win32 and Unix, allowing the development of cross-platform applications, and it is the graphics engine underlying the Google Chrome web browser.

Unlike Java application development based on the Eclipse IDE, the NDK is based on command-line tools and requires invoking them manually to build, deploy and debug the apps. Several third-party tools allow integrating the NDK into Eclipse and Visual Studio.

1.3.1.5 Android Open Accessory Development Kit

The Android 3.1 platform (also backported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode. When an Android powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple protocol (Android accessory protocol) that allows them to detect Android powered devices that support accessory mode.

1.3.1.6 App Inventor for Android

On 12 July 2010, Google announced the availability of App Inventor for Android, a Web-based visual development environment for novice programmers, based on MIT's Open Blocks Java library and providing access to Android devices' GPS, accelerometer and orientation data, phone functions, text messaging, speech-to-text conversion, contact data, persistent storage, and Web services, initially including Amazon and Twitter. "We could only have done this because Android’s architecture is so open," said the project director, MIT's Hal Abelson. Under development for over a year, the block-editing tool has been taught to non-majors in computer science at Harvard, MIT, Wellesley, Trinity College (Hartford,) and the University of San Francisco, where Professor David Wolber developed an introductory computer science course and tutorial book for non-computer science students based on App Inventor for Android.

1.3.1.7 Hyper net Android Creator

Hyper net Android Creator (HAC) is a software development system aimed at beginner programmers that can help them create their own Android apps without knowing Java and the Android SDK. It is based on HyperCard that treated software as a stack of cards with only one card being visible at any one time and so is well suited to mobile phone applications that have only one window visible at a time. Hyper net Android Creator's main programming language is simply called hyper net and is loosely based on HyperCard's HyperTalk language. Hyper net is an interpreted English-like language and has many features that allow creation of Android applications. It supports a growing subset of the Android SDK including its own versions of the GUI control types and automatically runs its own background service, so apps can continue to run and process information while in the background.

1.3.1.8 SDL

The SDL library offers also a development possibility beside Java, allowing the development with C and the simple porting of existing SDL and native C applications. By injection of a small Java shim and JNI the usage of native SDL code is possible, allowing Android ports like e.g. the Jagged Alliance 2 video game.

CHAPTER 2: LITERATURE REVIEW

For this application we basically used help of YouTube and Google to know the

- Structure [2]
- Code [2]
- GUI [2]
- Internal Programs [2]
- Working of an application [2]

Some Quotes about Tourist Guide:

- [1] “Travel makes one modest. You see what a tiny place you occupy in the world.” - Gustave Flaubert

- [1] “See the world. It's more fantastic than any dream made or paid for in factories. Ask for no guarantees, ask for no security.” -Ray Bradbury, Fahrenheit 451

- [1] “It is always sad to leave a place to which one knows one will never return. Such are the melancholies du voyage: perhaps they are one of the most rewarding things about traveling.” -Gustave Flaubert, Flaubert in Egypt: A Sensibility on Tour

CHAPTER 3: OBJECTIVES AND METHODOLOGY

3.1 Objective •

To facilitate the tour operators by providing them with qualified tourist guides for guiding assignments.

- Applications provide you all helps that your travel needs, such as location details, Restaurant details and explore local experience

- The use of mobile apps for traveling will increase and continue to evolve in the coming days.

- Travel apps are also used as powerful marketing tools. You can consider this idea to promote your business.

3.2 Methodology

- The application is made by a software ANDROID STUDIO

- We used YouTube to understand and how to make the application

CHAPTER 4: IMPLEMENTATION / CODING

4.1 The Classes

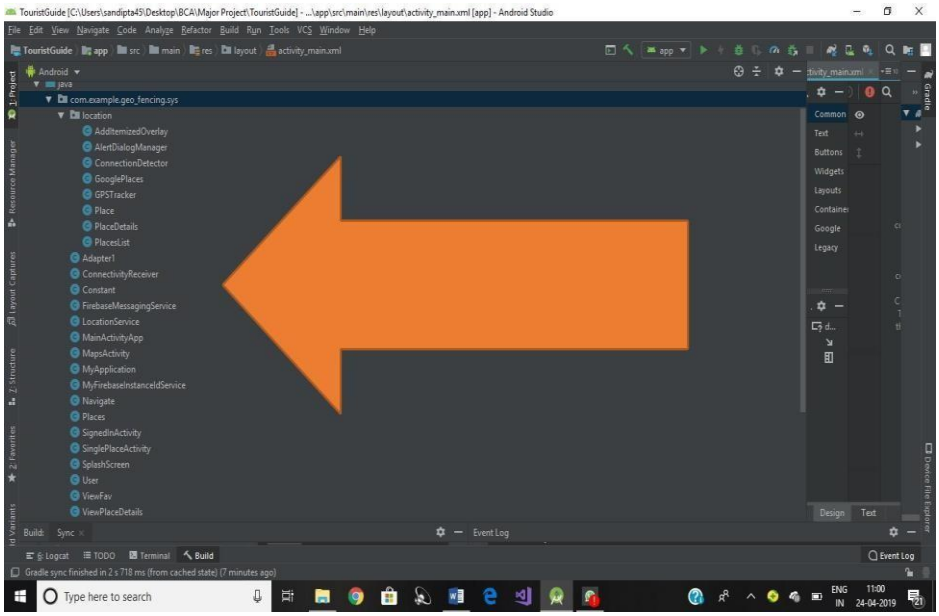

Fig 4.1: The Classes

4.2 Main Activity

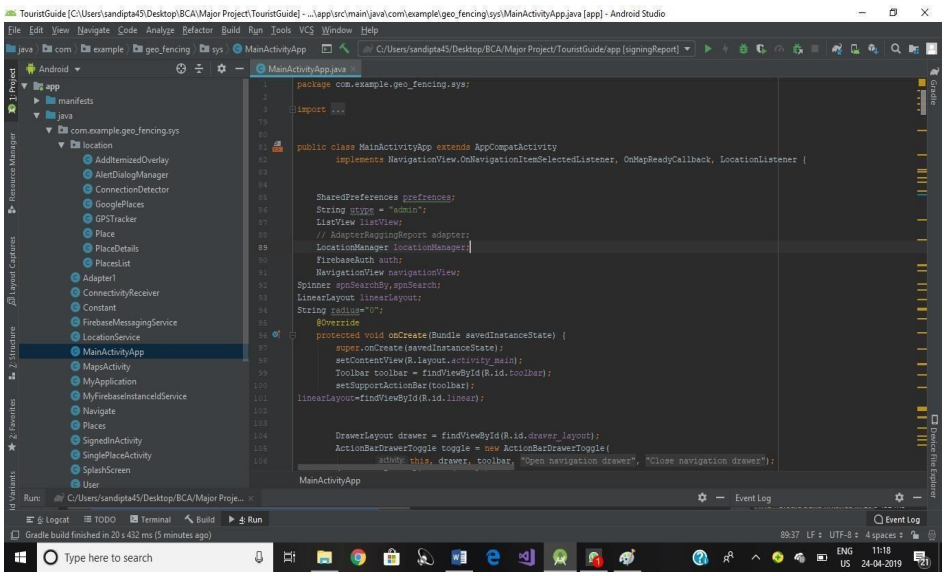

Fig 4.2

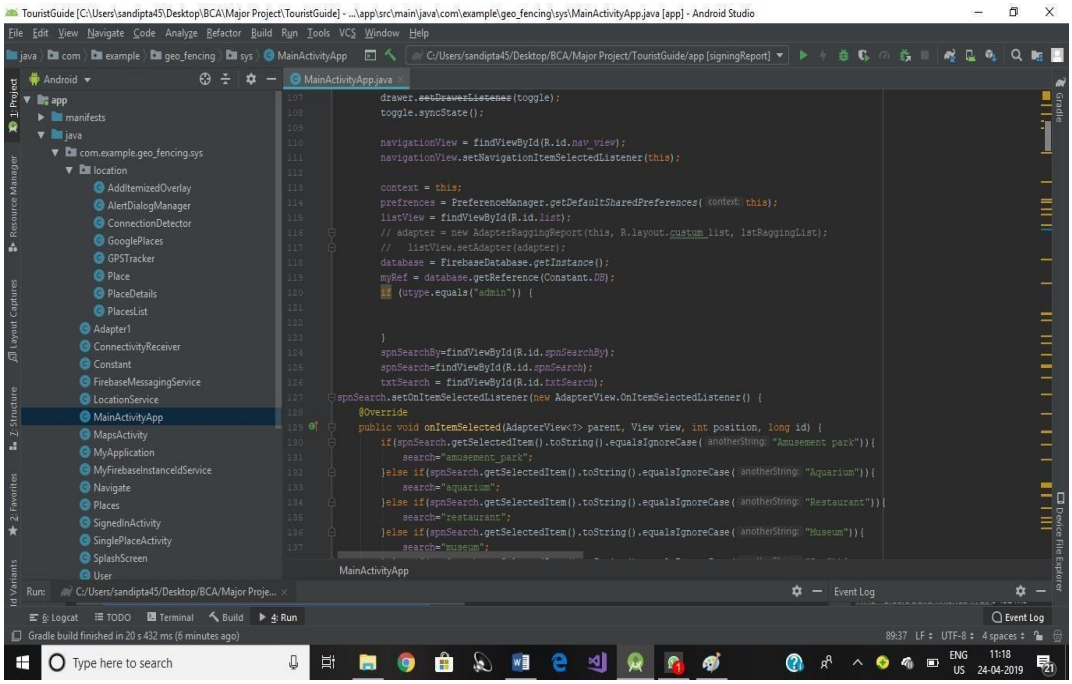
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
MainActivityApp.java
MainActivityApp
Type here to search

Fig 4.3

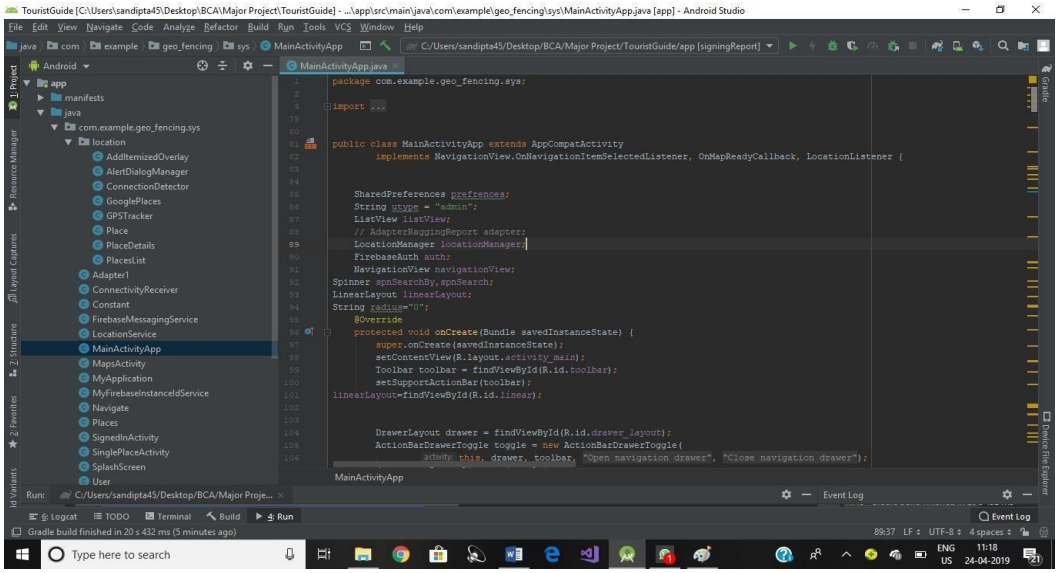
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
MainActivityApp.java
MainActivityApp
Type here to search

Fig 4.4

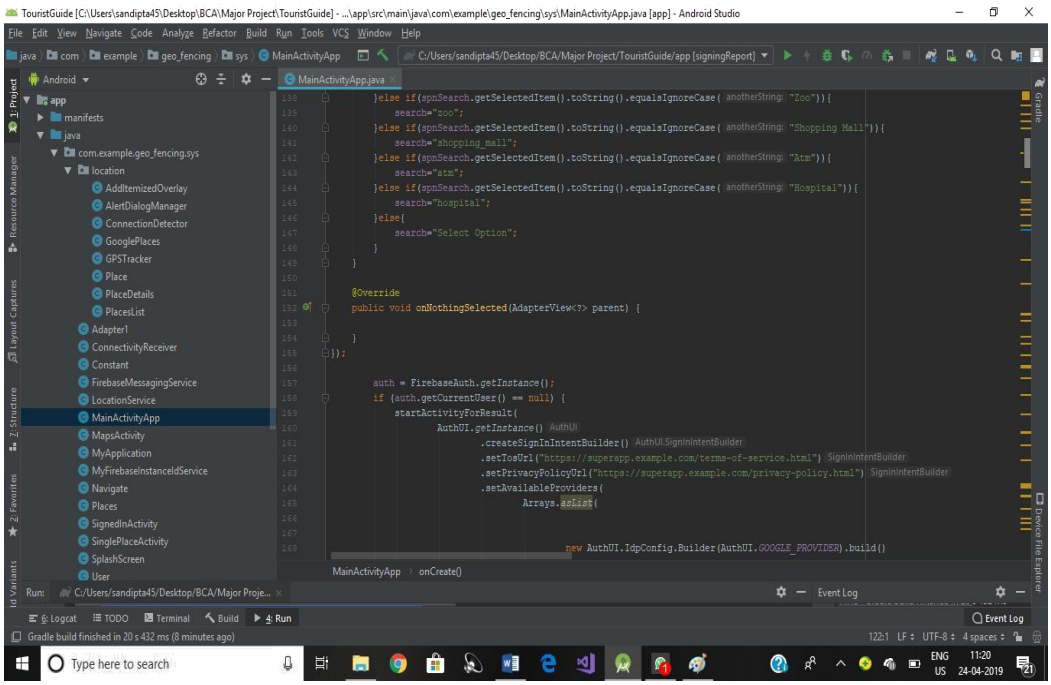
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MainActivityApp.java
app
manifests
java
com.example.geo_fencing.sys
location
AddItemizedOverlay
AlertDialogManager
ConnectionDetector
GooglePlaces
GPSTracker
Place
PlaceDetails
PlacesList
Adapter1
ConnectivityReceiver
Constant
FirebaseMessagingService
LocationService
MainActivityApp
MapsActivity
MyApplication
MyFirebaseInstanceIdService
Navigate
Places
SignedInActivity
SinglePlaceActivity
SplashScreen
User
}else if(spnSearch.getSelectedItem().toString().equalsIgnoreCase(anotherString: "Zoo")){
search="zoo";
}else if(spnSearch.getSelectedItem().toString().equalsIgnoreCase(anotherString: "Shopping Mall")){
search="shopping_mall";
}else if(spnSearch.getSelectedItem().toString().equalsIgnoreCase(anotherString: "Atm")){
search="atm";
}else if(spnSearch.getSelectedItem().toString().equalsIgnoreCase(anotherString: "Hospital")){
search="hospital";
}else{
search="Select Option";
}
}
@Override
public void onNothingSelected(AdapterView<?> parent) {
}
});
auth = FirebaseAuth.getInstance();
if (auth.getCurrentUser() == null) {
startActivityForResult(
AuthUI.getInstance() AuthUI
.createSignInIntentBuilder() AuthUI.SignInIntentBuilder
.setTosUrl("https://superapp.example.com/terms-of-service.html") SignInIntentBuilder
.setPrivacyPolicyUrl("https://superapp.example.com/privacy-policy.html") SignInIntentBuilder
.setAvailableProviders(
Arrays.asList(
new AuthUI.IdpConfig.Builder(AuthUI.GOOGLE_PROVIDER).build()
MainActivityApp › onCreate()
Run: C:/Users/sandipta45/Desktop/BCA/Major Proje...
Event Log
6: Logcat TODO Terminal Build 4: Run
Gradle build finished in 20 s 432 ms (8 minutes ago)
122:1 LF UTF-8 4 spaces
Type here to search
ENG US 11:20 24-04-2019

Fig 4.4

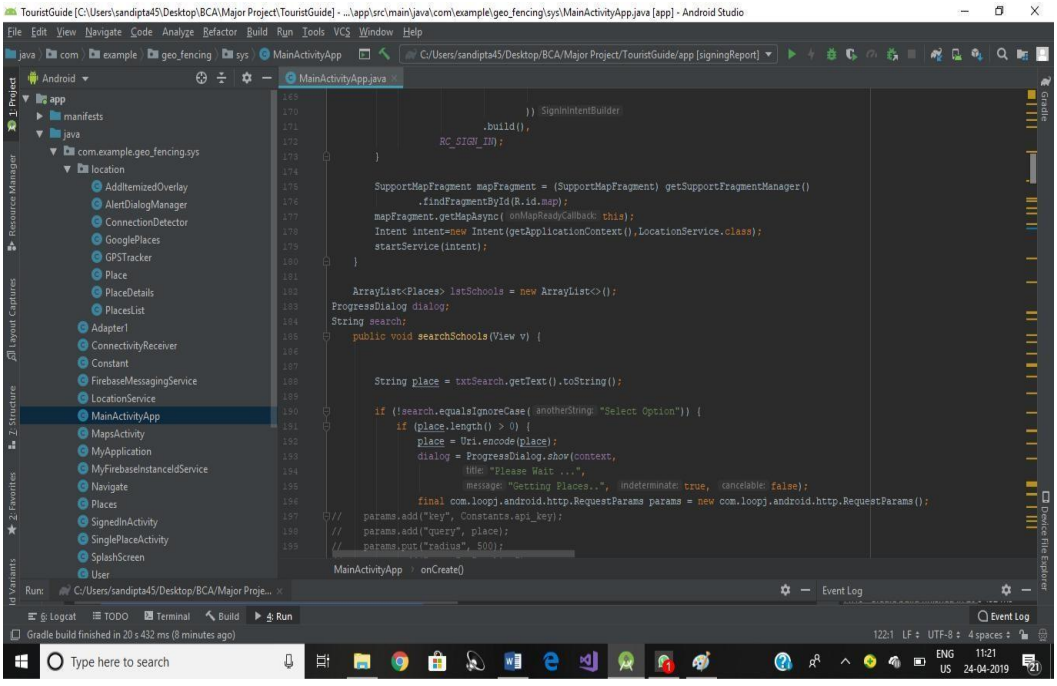
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MainActivityApp.java
app
manifests
java
com.example.geo_fencing.sys
location
AddItemizedOverlay
AlertDialogManager
ConnectionDetector
GooglePlaces
GPSTracker
Place
PlaceDetails
PlacesList
Adapter1
ConnectivityReceiver
Constant
FirebaseMessagingService
LocationService
MainActivityApp
MapsActivity
MyApplication
MyFirebaseInstanceIdService
Navigate
Places
SignedInActivity
SinglePlaceActivity
SplashScreen
User
)) SignInIntentBuilder
.build(),
RC_SIGN_IN);
}
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
.findFragmentById(R.id.map);
mapFragment.getMapAsync(onMapReadyCallback: this);
Intent intent=new Intent(getApplicationContext(),LocationService.class);
startService(intent);
}
ArrayList<Places> lstSchools = new ArrayList<>();
ProgressDialog dialog;
String search;
public void searchSchools(View v) {
String place = txtSearch.getText().toString();
if (!search.equalsIgnoreCase(anotherString: "Select Option")) {
if (place.length() > 0) {
place = Uri.encode(place);
dialog = ProgressDialog.show(context,
title: "Please Wait ...",
message: "Getting Places..", indeterminate: true, cancelable: false);
final com.loopj.android.http.RequestParams params = new com.loopj.android.http.RequestParams();
// params.add("key", Constants.api_key);
// params.add("query", place);
// params.put("radius", 500);
MainActivityApp › onCreate()
Run: C:/Users/sandipta45/Desktop/BCA/Major Proje...
Event Log
6: Logcat TODO Terminal Build 4: Run
Gradle build finished in 20 s 432 ms (8 minutes ago)
122:1 LF UTF-8 4 spaces
Type here to search
ENG US 11:21 24-04-2019

Fig 4.5

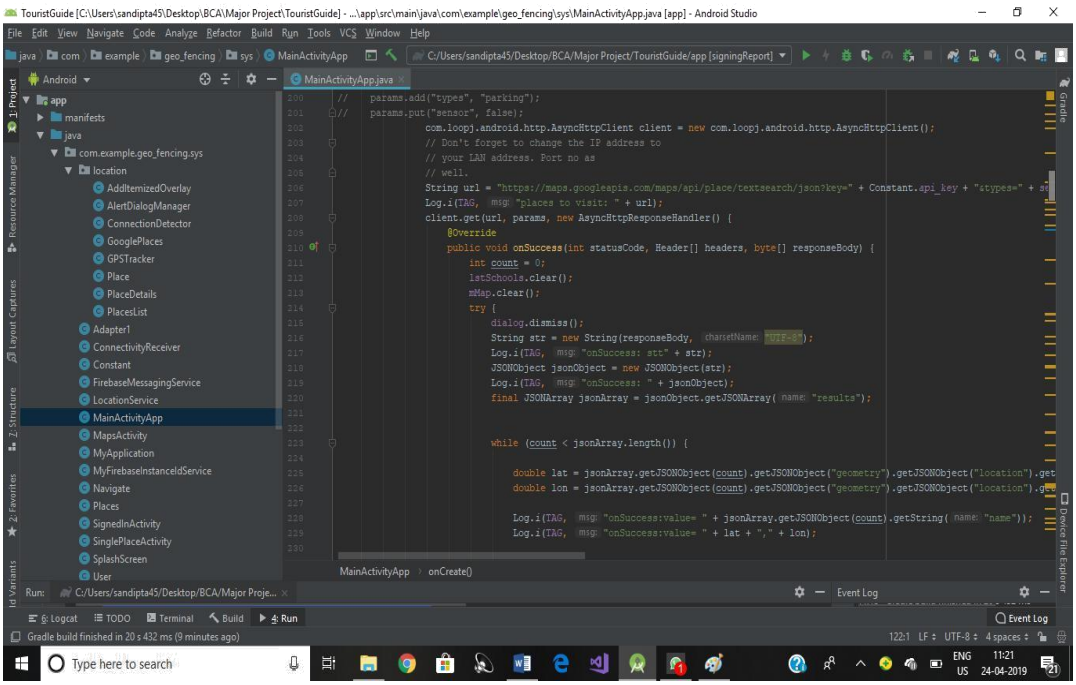
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MainActivityApp.java
// params.add("types", "parking");
// params.put("sensor", false);
com.loopj.android.http.AsyncHttpClient client = new com.loopj.android.http.AsyncHttpClient();
// Don't forget to change the IP address to
// your LAN address. Port no as
// well.
String url = "https://maps.googleapis.com/maps/api/place/textsearch/json?key=" + Constant.api_key + "&types=" + s
Log.i(TAG, msg: "places to visit: " + url);
client.get(url, params, new AsyncHttpResponseHandler() {
@Override
public void onSuccess(int statusCode, Header[] headers, byte[] responseBody) {
int count = 0;
lstSchools.clear();
mMap.clear();
try {
dialog.dismiss();
String str = new String(responseBody, charsetName: "UTF-8");
Log.i(TAG, msg: "onSuccess: str" + str);
JSONObject jsonObject = new JSONObject(str);
Log.i(TAG, msg: "onSuccess: " + jsonObject);
final JSONArray jsonArray = jsonObject.getJSONArray(name: "results");
while (count < jsonArray.length()) {
double lat = jsonArray.getJSONObject(count).getJSONObject("geometry").getJSONObject("location").get
double lon = jsonArray.getJSONObject(count).getJSONObject("geometry").getJSONObject("location").g
Log.i(TAG, msg: "onSuccess:value= " + jsonArray.getJSONObject(count).getString(name: "name"));
Log.i(TAG, msg: "onSuccess:value= " + lat + "," + lon);
MainActivityApp › onCreate()
Run: C:/Users/sandipta45/Desktop/BCA/Major Proje...
Event Log
6: Logcat TODO Terminal Build 4: Run
Gradle build finished in 20 s 432 ms (9 minutes ago)
122:1 LF UTF-8 4 spaces
Type here to search
ENG US 11:21 24-04-2019

Fig 4.6

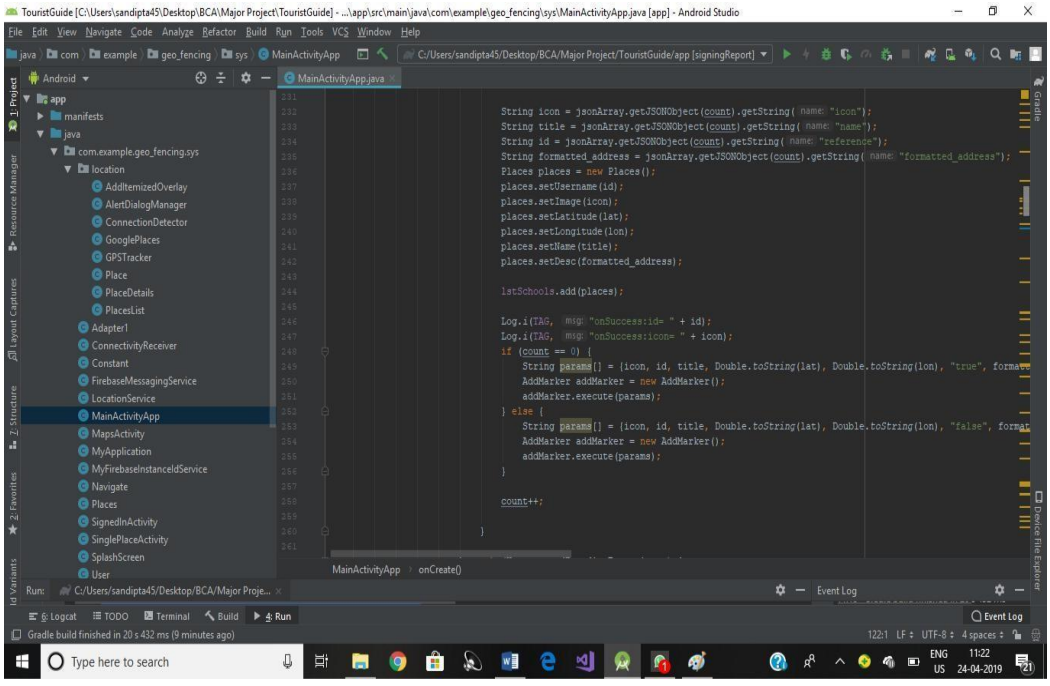
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
MainActivityApp.java
String icon = jsonArray.getJSONObject(count).getString(name: "icon");
String title = jsonArray.getJSONObject(count).getString(name: "name");
String id = jsonArray.getJSONObject(count).getString(name: "reference");
String formatted_address = jsonArray.getJSONObject(count).getString(name: "formatted_address");
Places places = new Places();
places.setUsername(id);
places.setImage(icon);
places.setLatitude(lat);
places.setLongitude(lon);
places.setName(title);
places.setDesc(formatted_address);
lstSchools.add(places);
Log.i(TAG, msg: "onSuccess:id= " + id);
Log.i(TAG, msg: "onSuccess:icon= " + icon);
if (count == 0) {
String params[] = {icon, id, title, Double.toString(lat), Double.toString(lon), "true", formatte
AddMarker addMarker = new AddMarker();
addMarker.execute(params);
} else {
String params[] = {icon, id, title, Double.toString(lat), Double.toString(lon), "false", format
AddMarker addMarker = new AddMarker();
addMarker.execute(params);
}
count++;
}
MainActivityApp › onCreate()
Run: C:/Users/sandipta45/Desktop/BCA/Major Proje...
Event Log
6: Logcat TODO Terminal Build 4: Run
Gradle build finished in 20 s 432 ms (9 minutes ago)
122:1 LF UTF-8 4 spaces
Type here to search
ENG US 11:22 24-04-2019

Fig 4.7

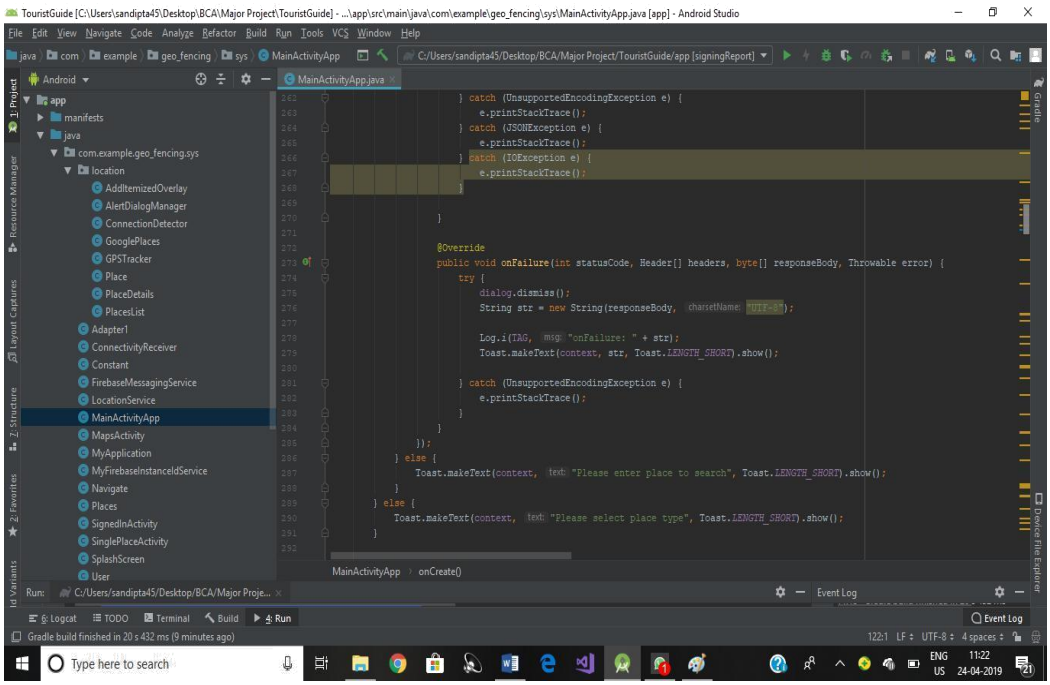
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MainActivityApp.java
MainActivityApp › onCreate()
Run: C:/Users/sandipta45/Desktop/BCA/Major Proje...
Event Log
Logcat TODO Terminal Build Run
Gradle build finished in 20 s 432 ms (9 minutes ago)
Type here to search
ENG US 11:22 24-04-2019

Fig 4.8

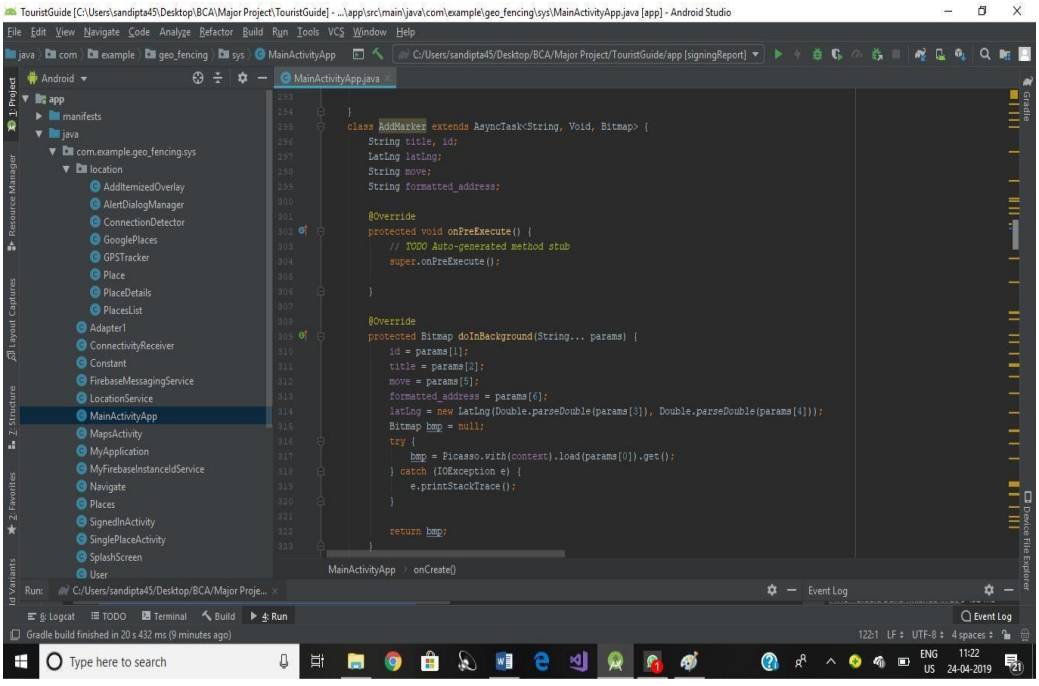
TouristGuide [C:\Users\sandipta45\Desktop\BCA\Major Project\TouristGuide] - ...\app\src\main\java\com\example\geo_fencing\sys\MainActivityApp.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MainActivityApp.java
MainActivityApp › onCreate()
Run: C:/Users/sandipta45/Desktop/BCA/Major Proje...
Event Log
Logcat TODO Terminal Build Run
Gradle build finished in 20 s 432 ms (9 minutes ago)
Type here to search
ENG US 11:22 24-04-2019

Fig 4.9

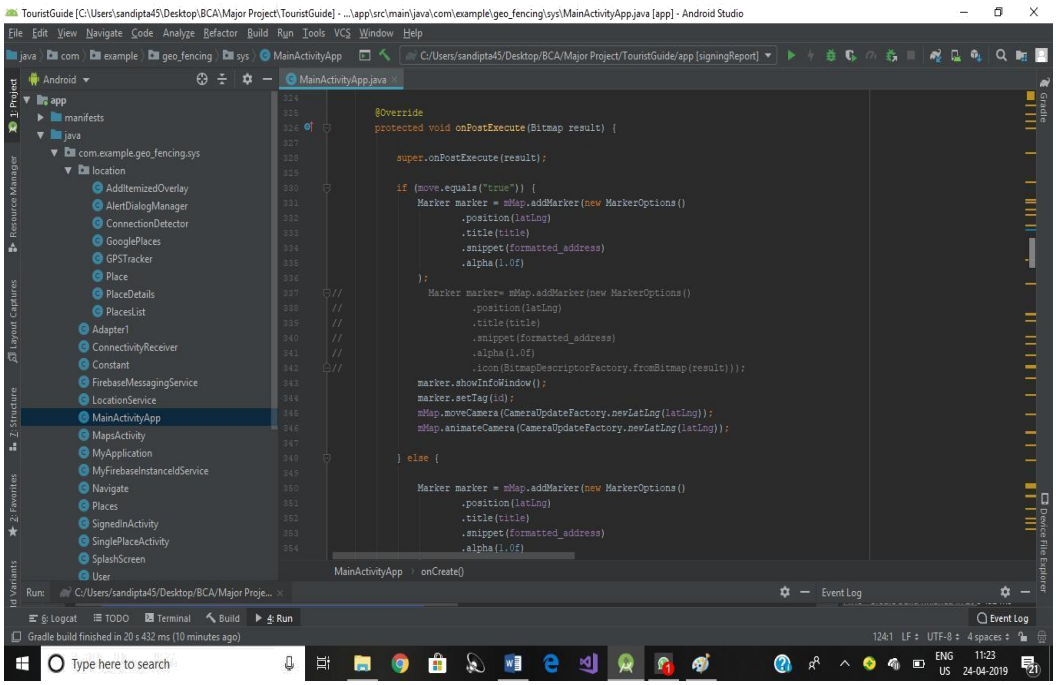

Fig 4.10

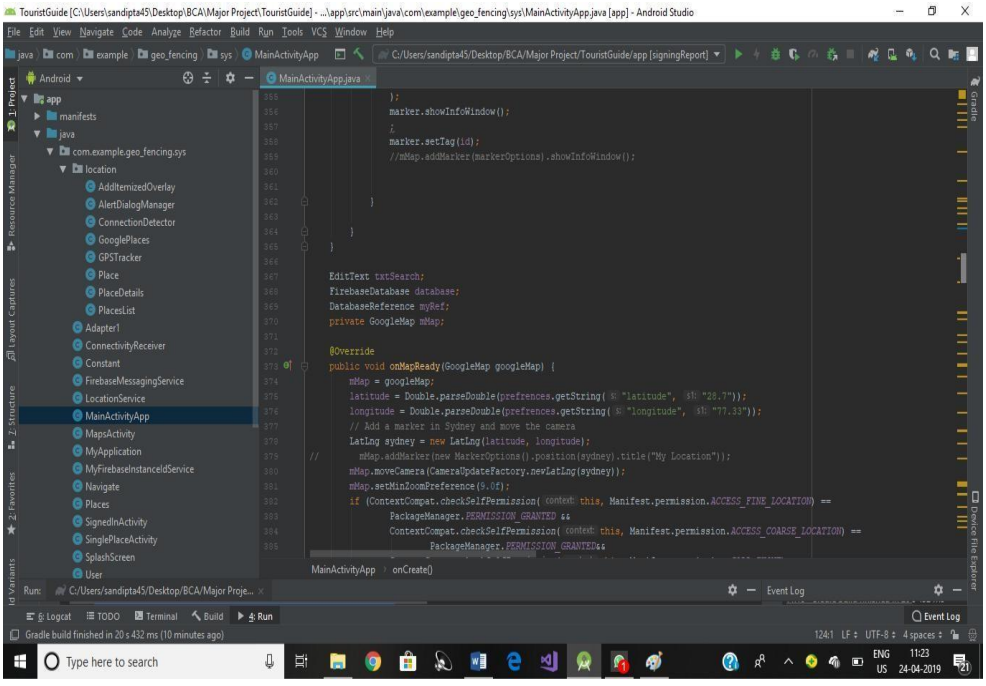

Fig 4.11

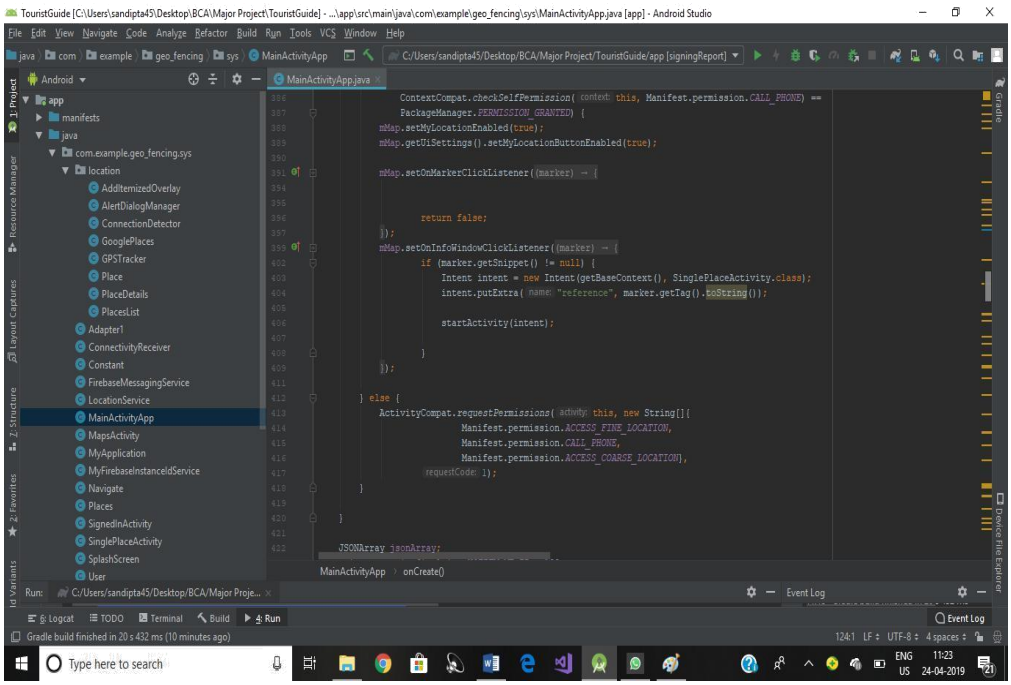

Fig 4.11

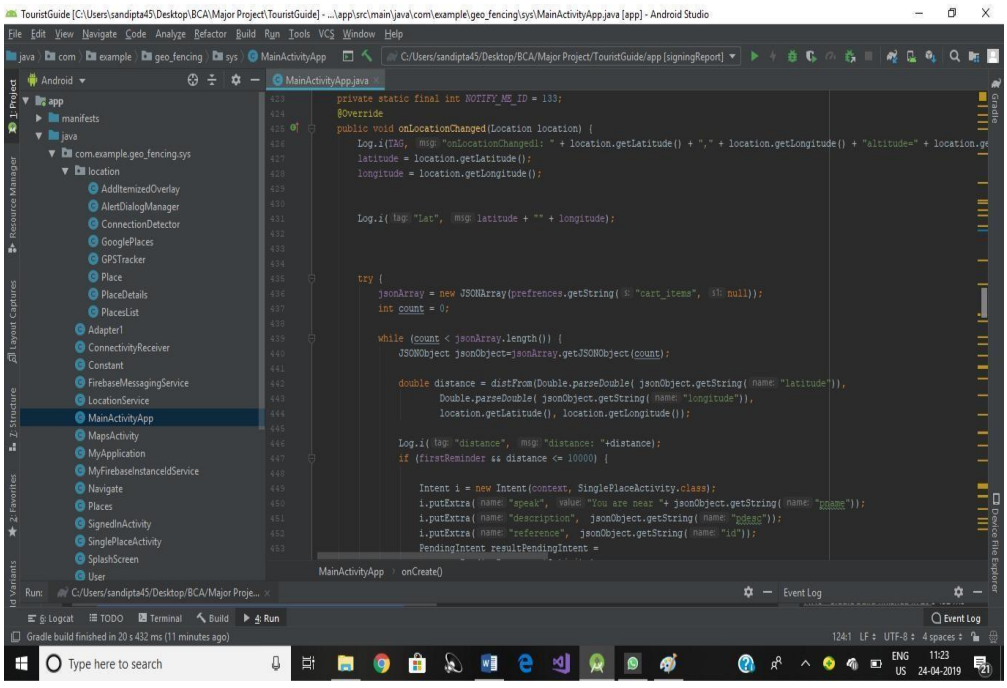

Fig 4.12

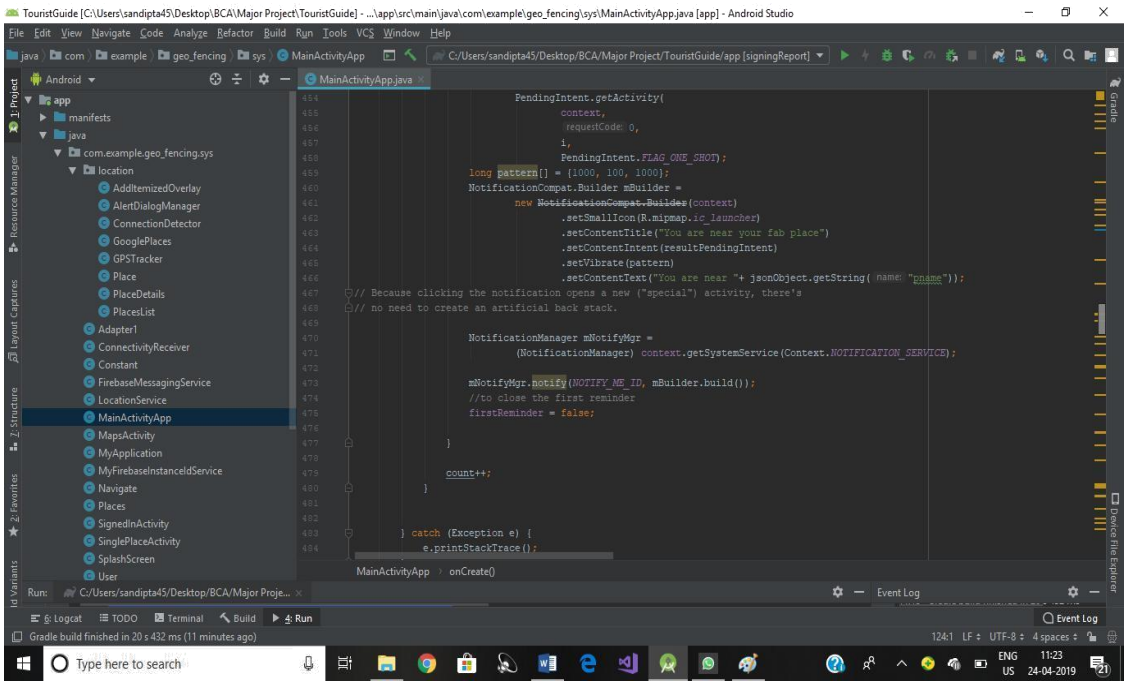

Fig 4.13

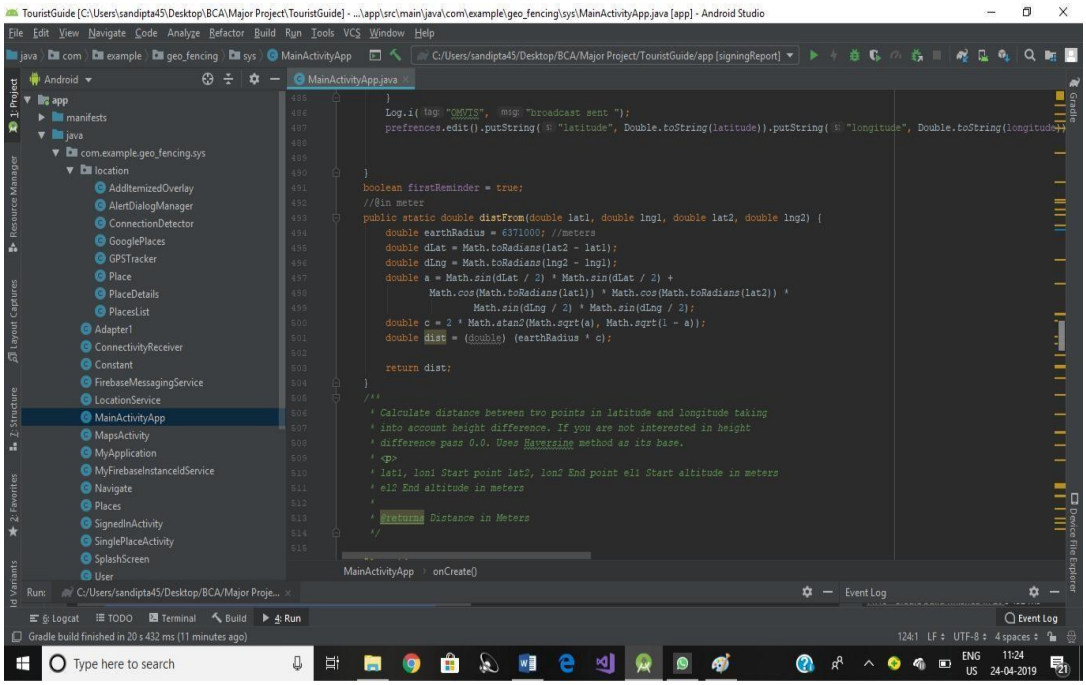

Fig 4.14

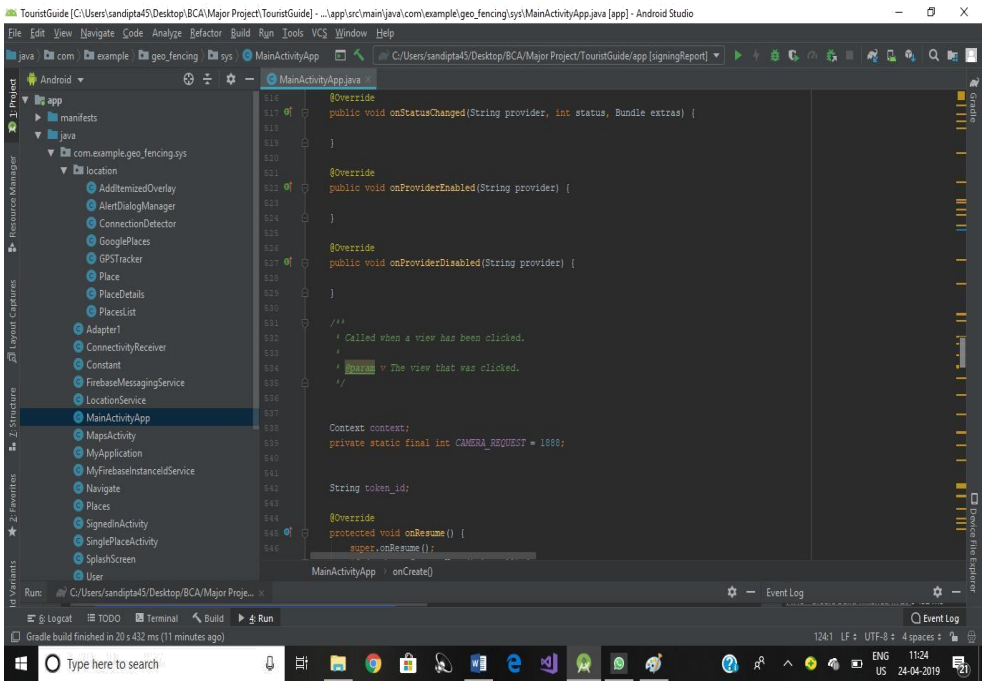

Fig 4.15

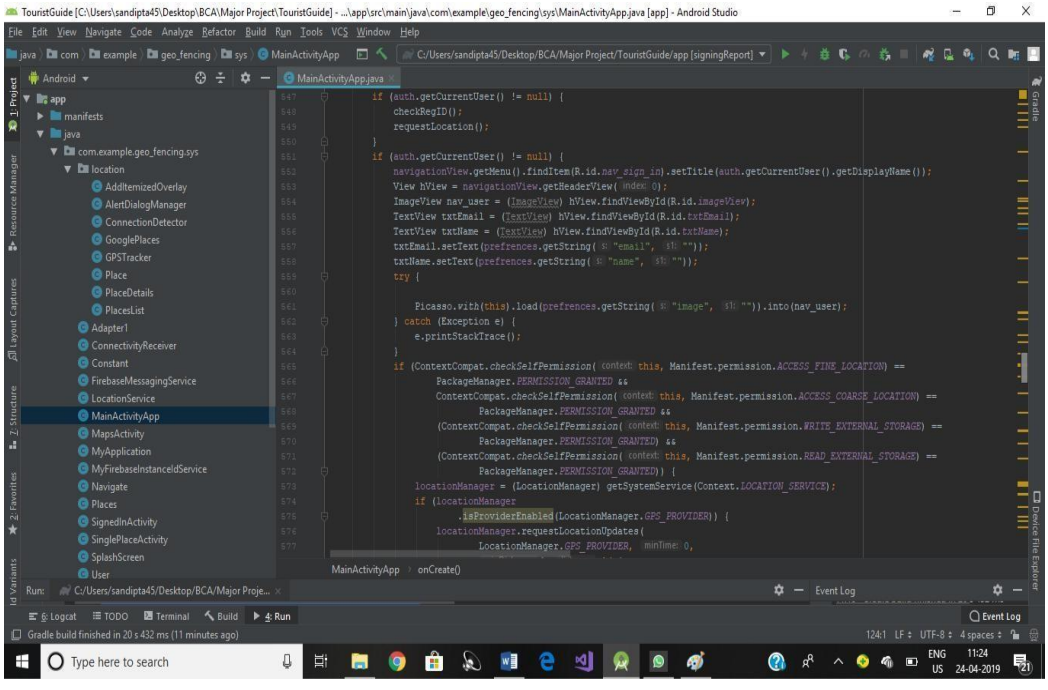

Fig 4.16

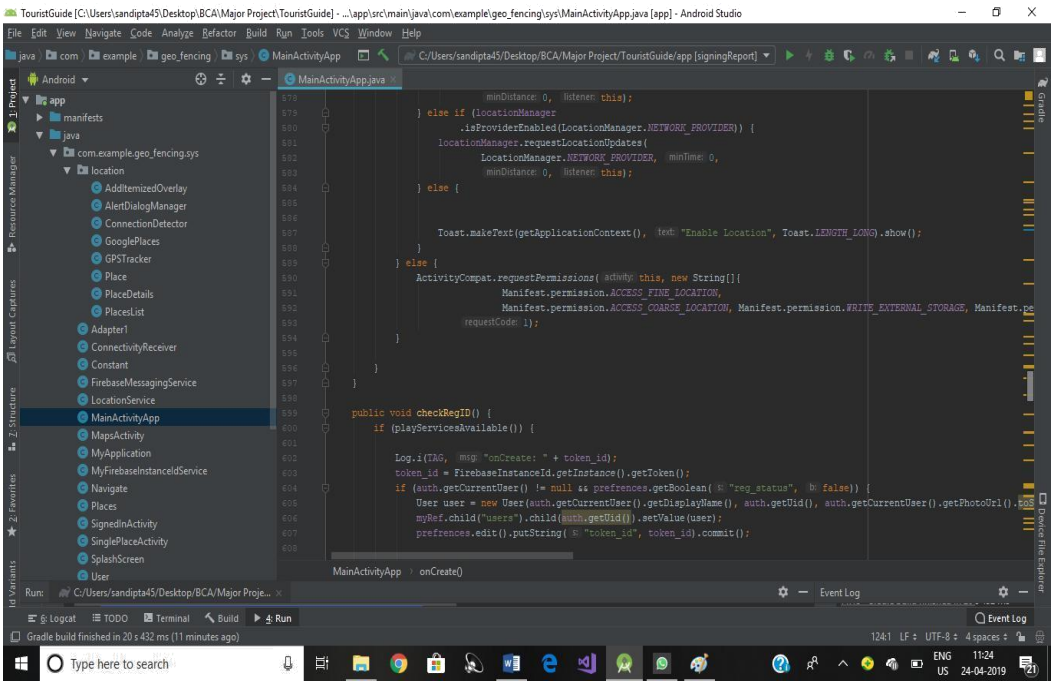

Fig 4.17

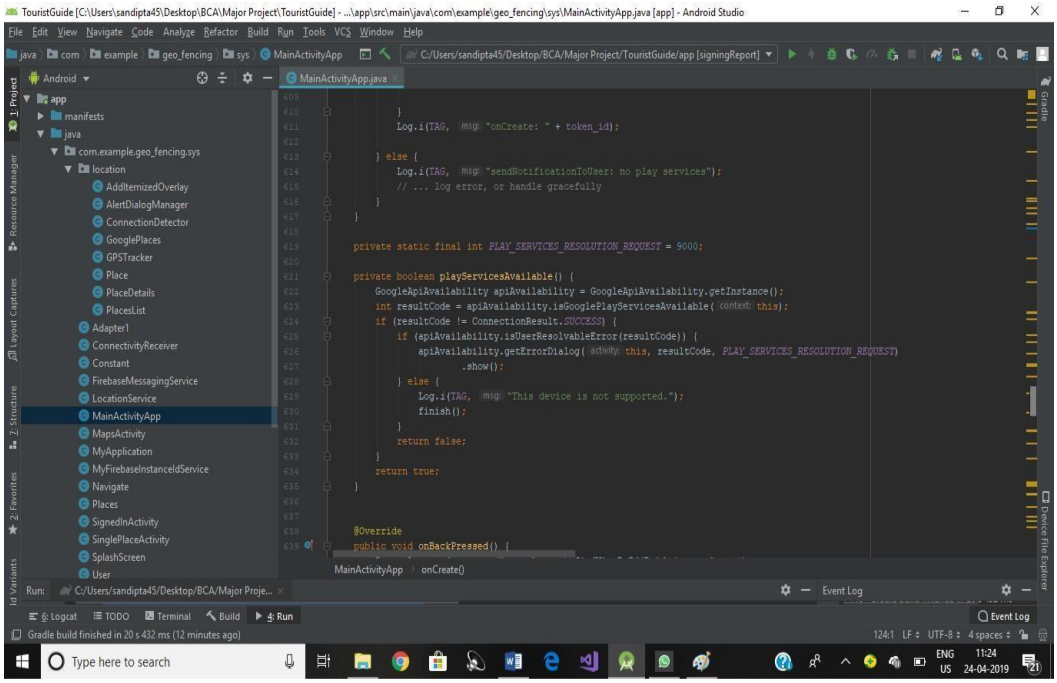

Fig 4.18

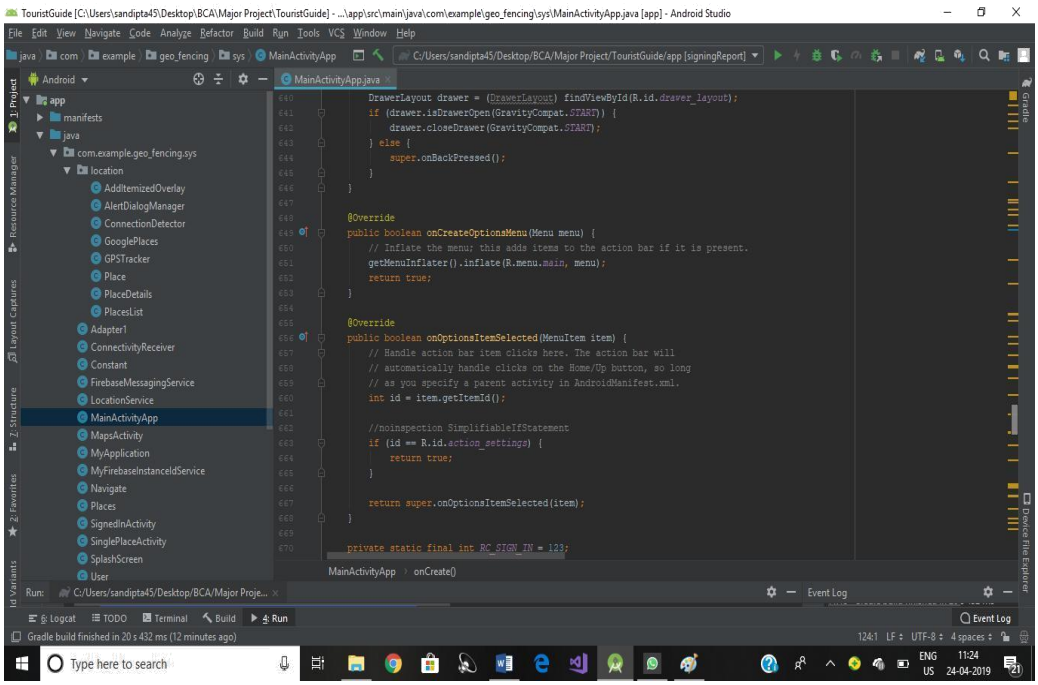

Fig 4.19

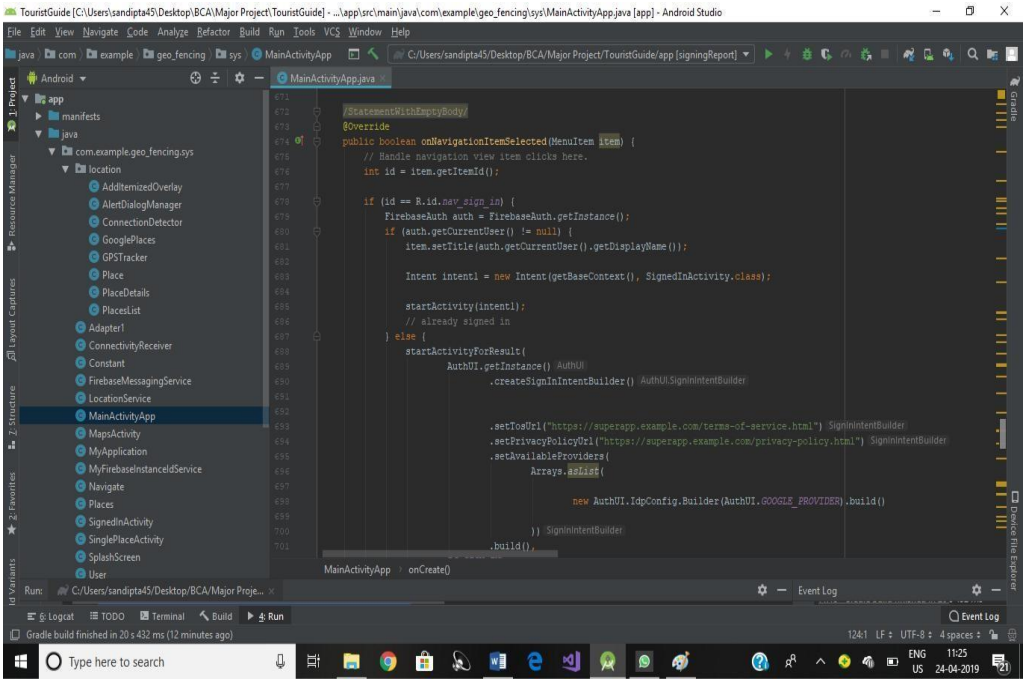

Fig 4.20

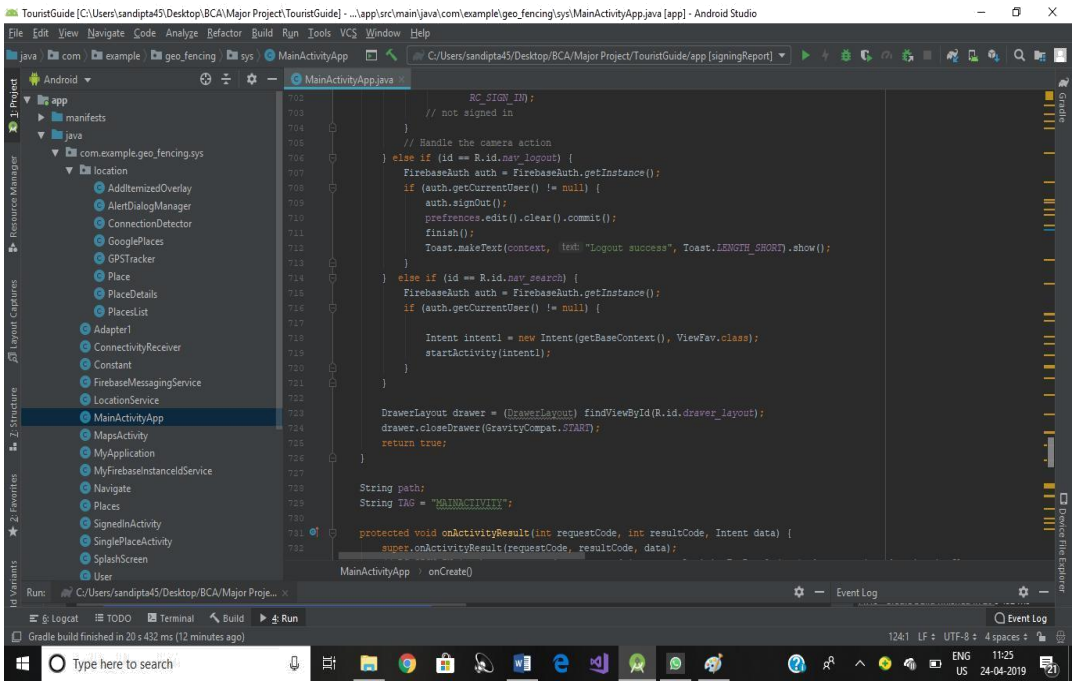

Fig 4.21

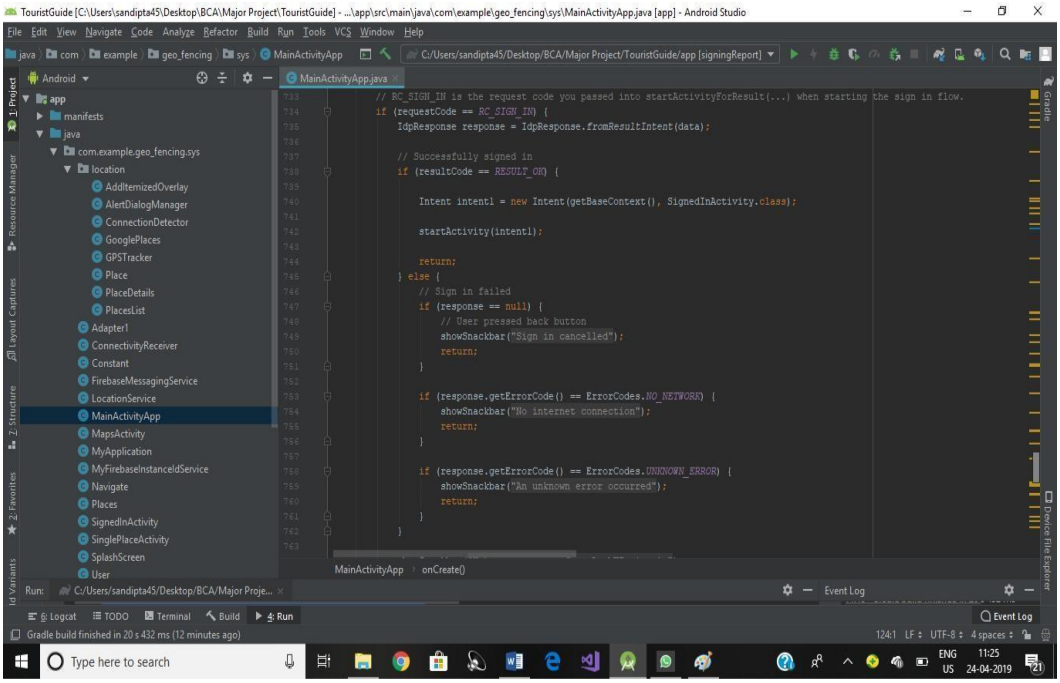

Fig 4.22

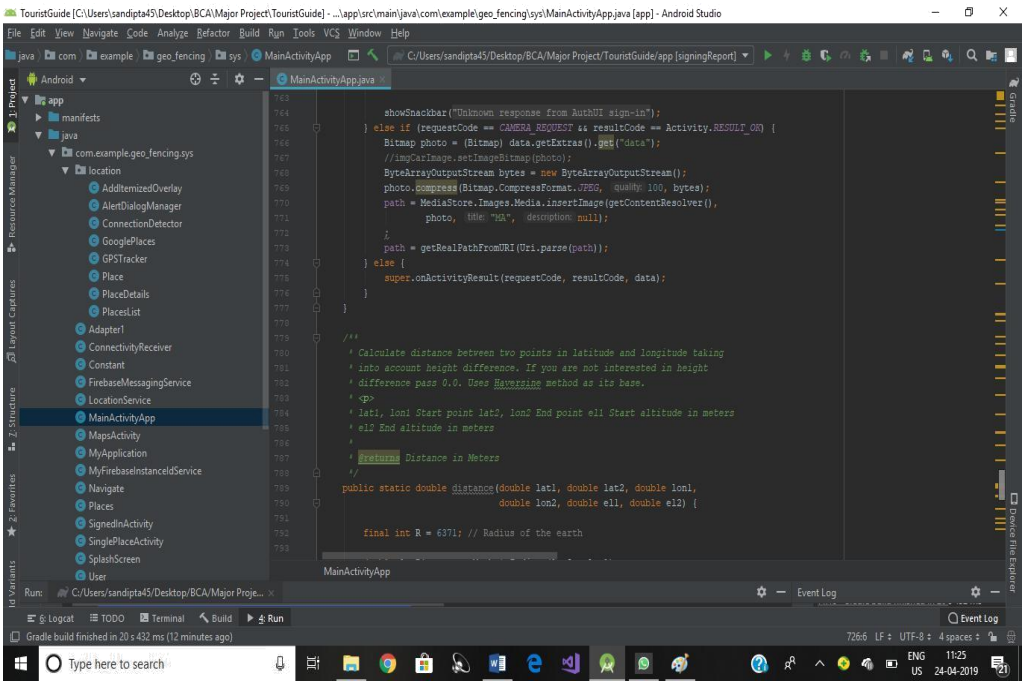

Fig 4.23

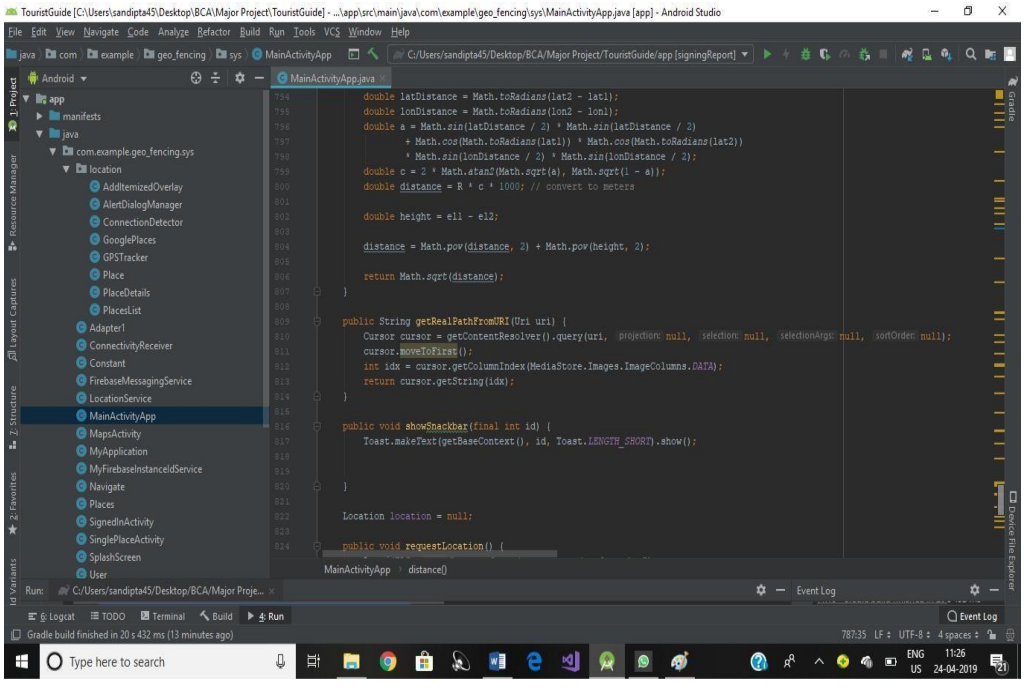

Fig 4.24

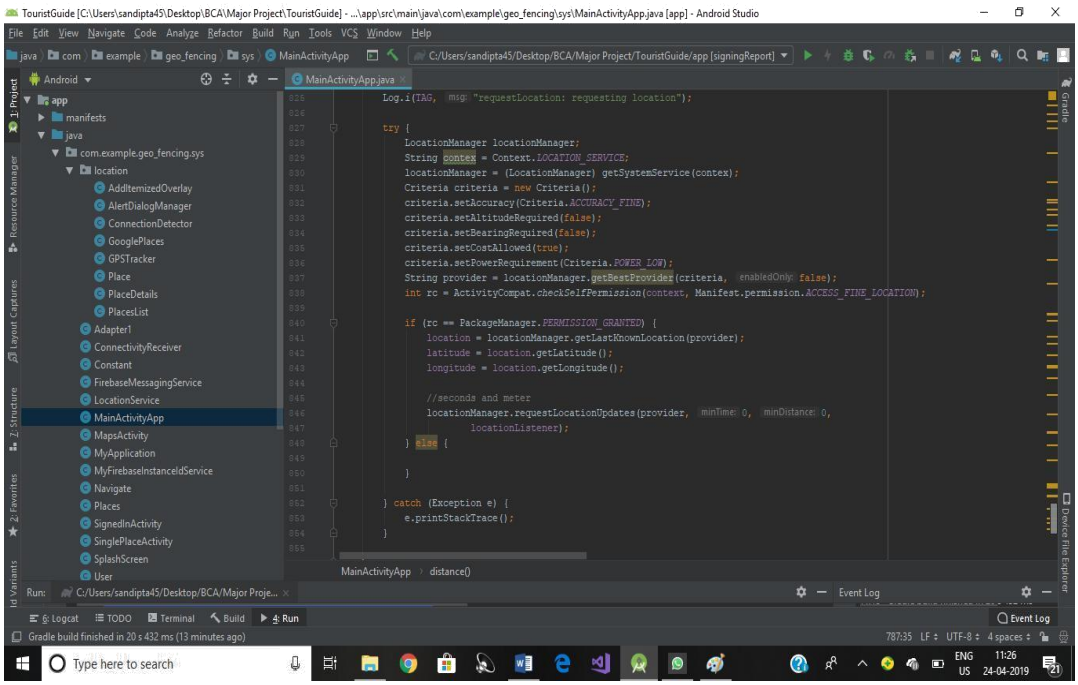

Fig 4.25

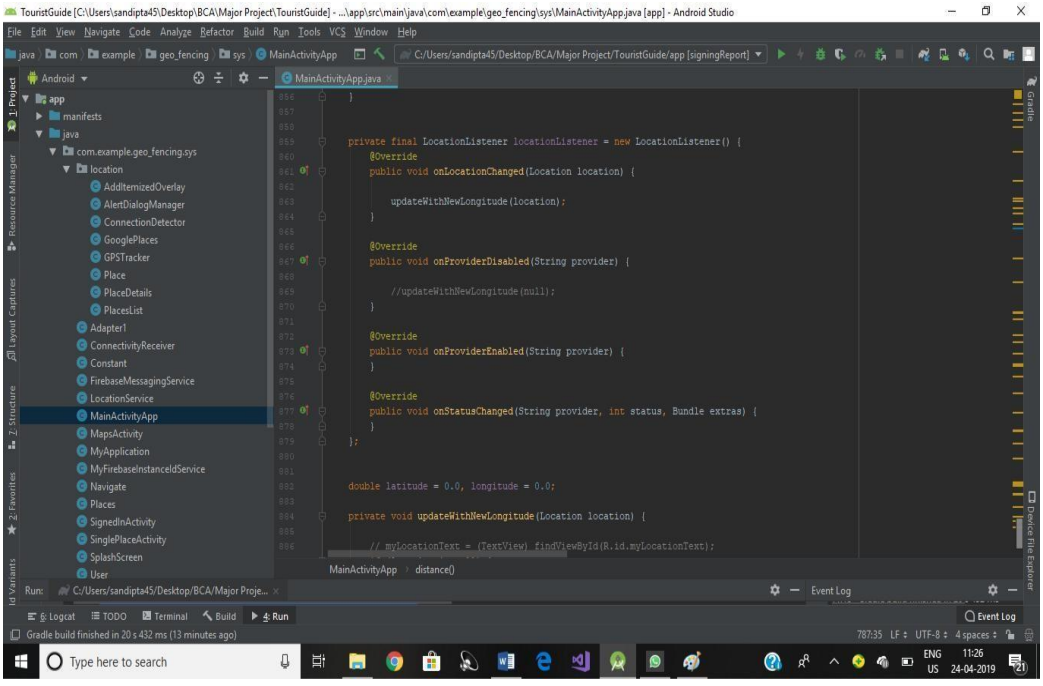

Fig 4.26

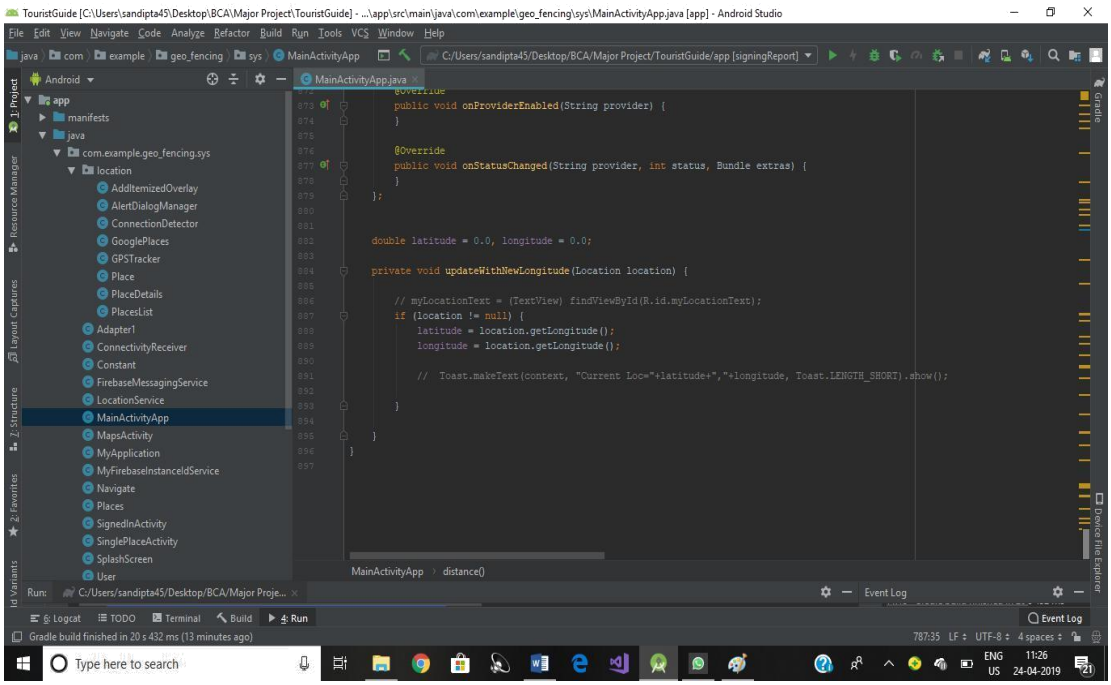

Fig 4.27

4.3 **The Database Connection**

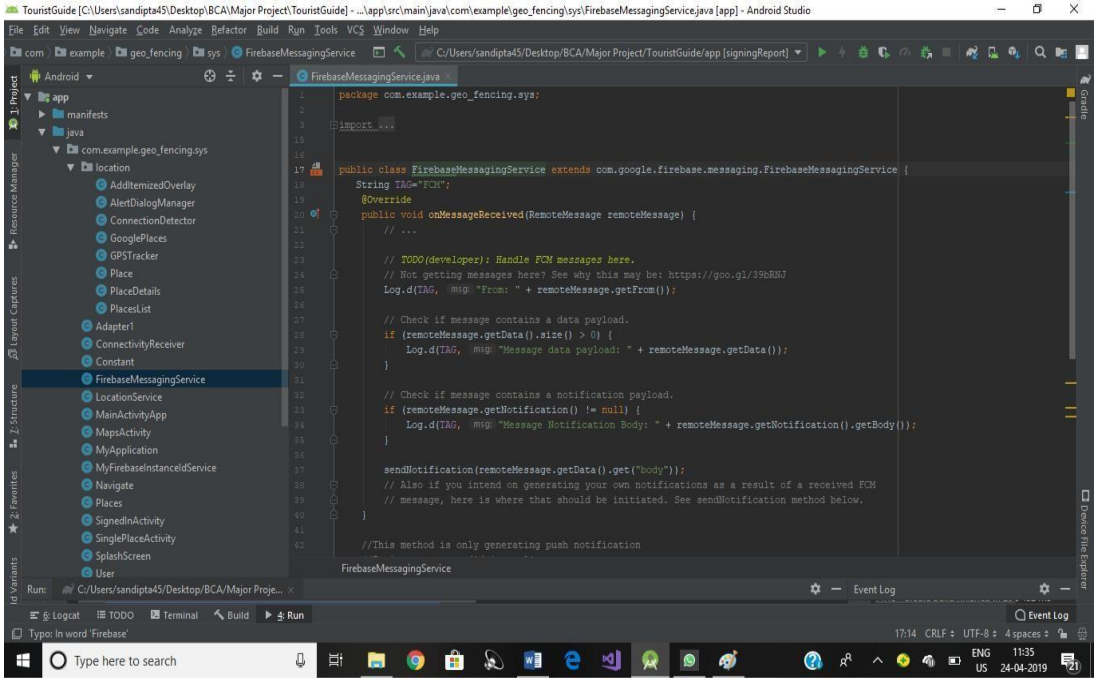

Fig 4.28

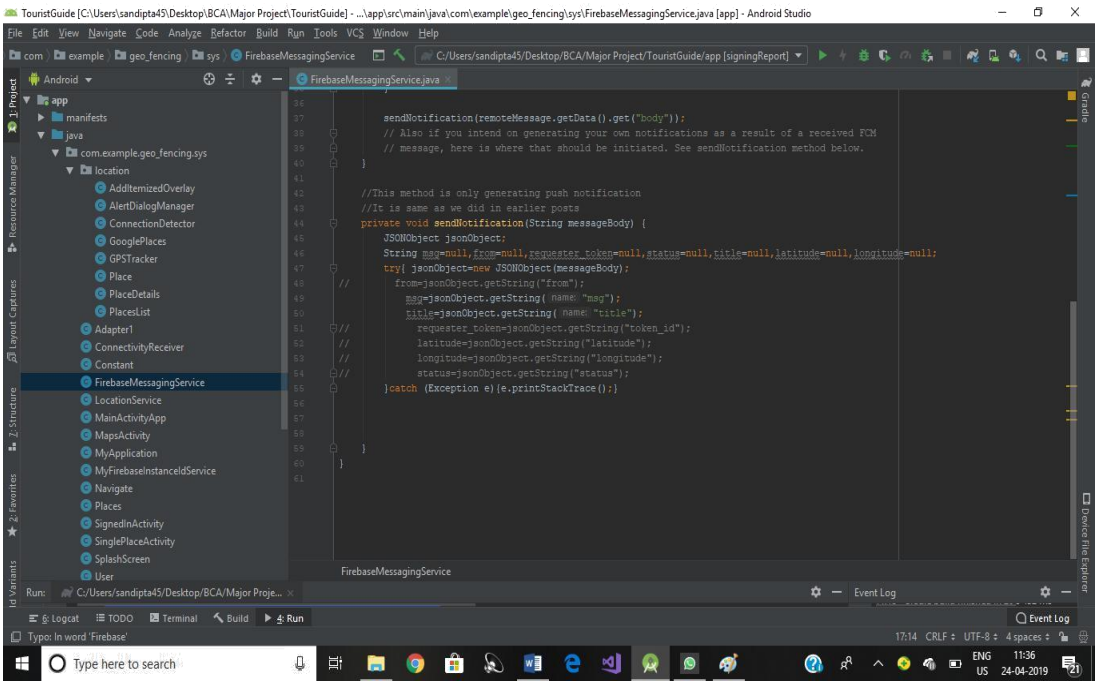

Fig 4.29

4.4 **Location Service**

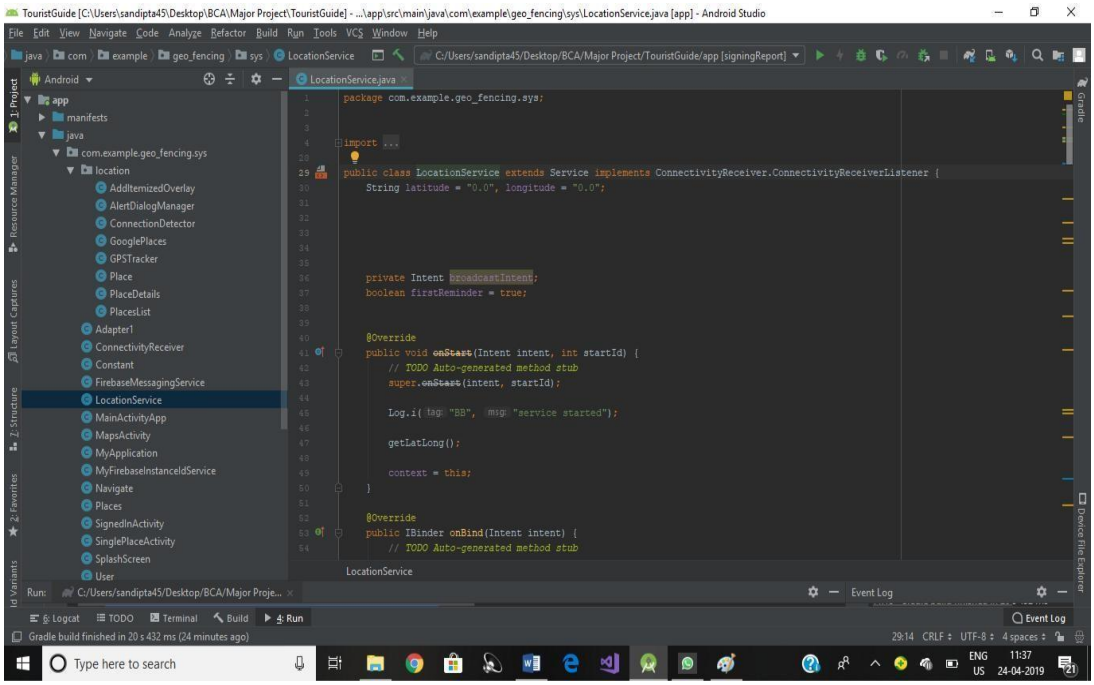

Fig 4.30

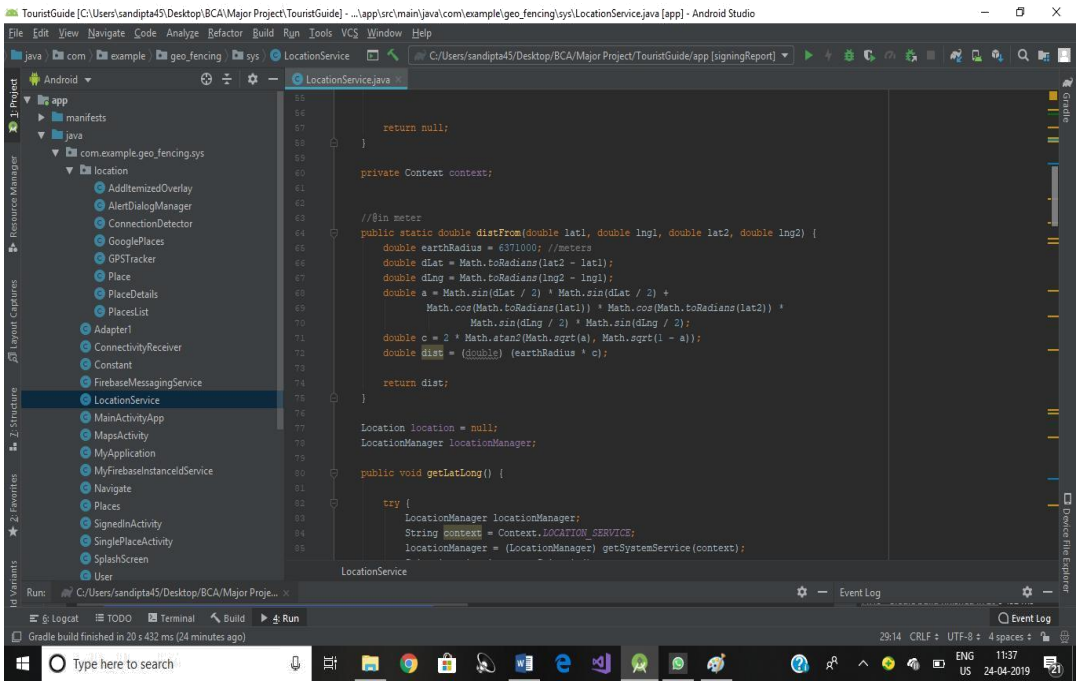

Fig 4.31

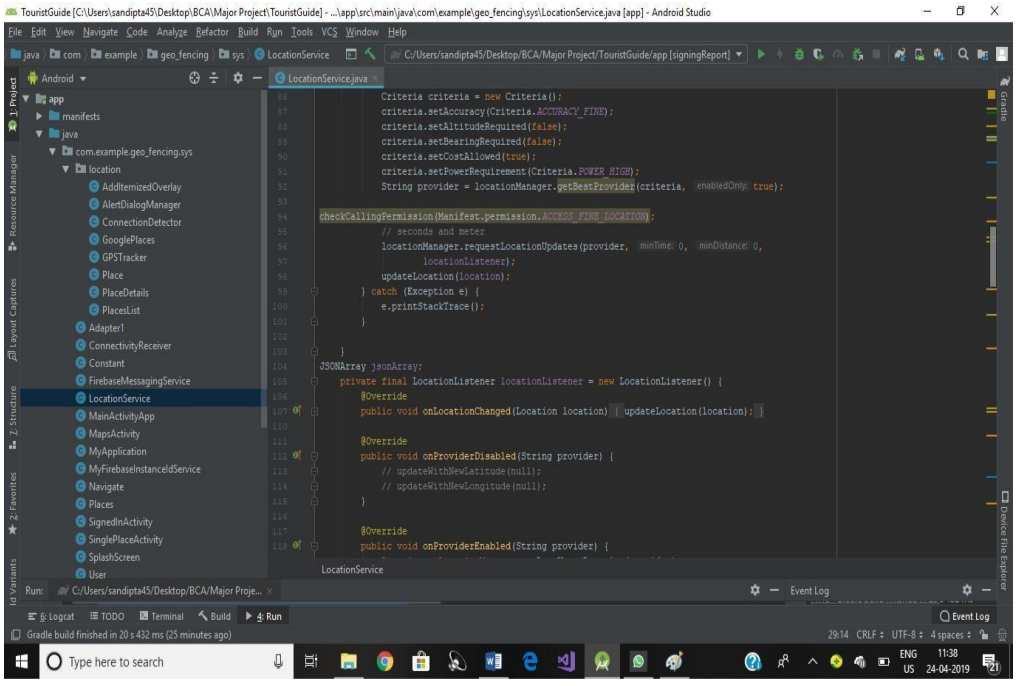

Fig 4.32

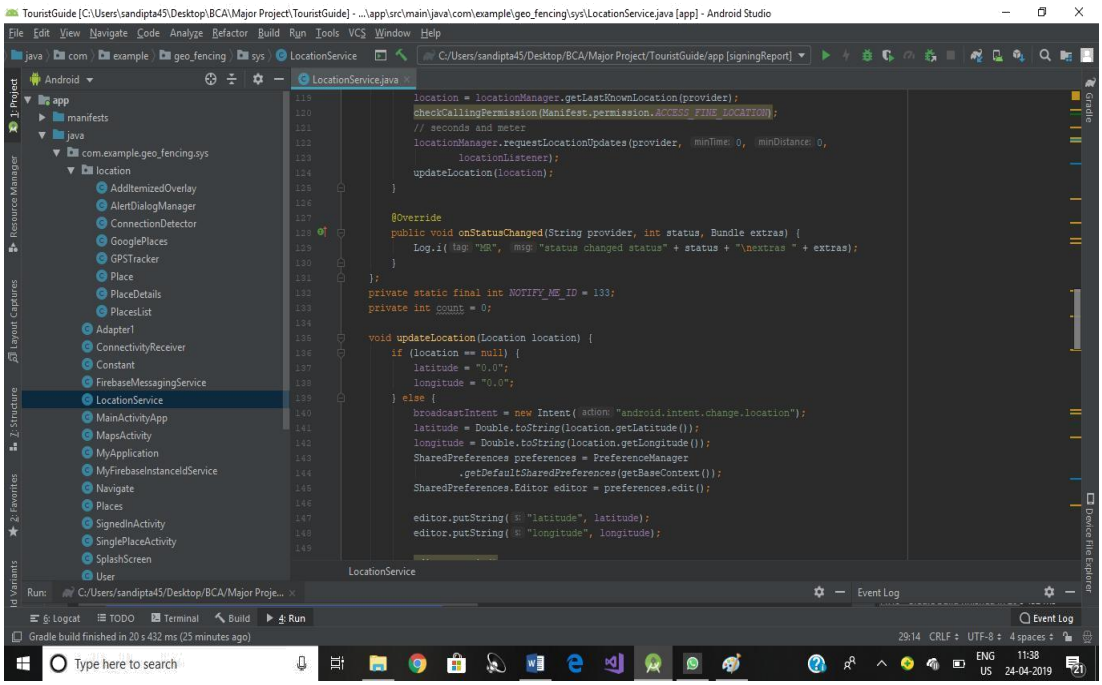

Fig 4.33

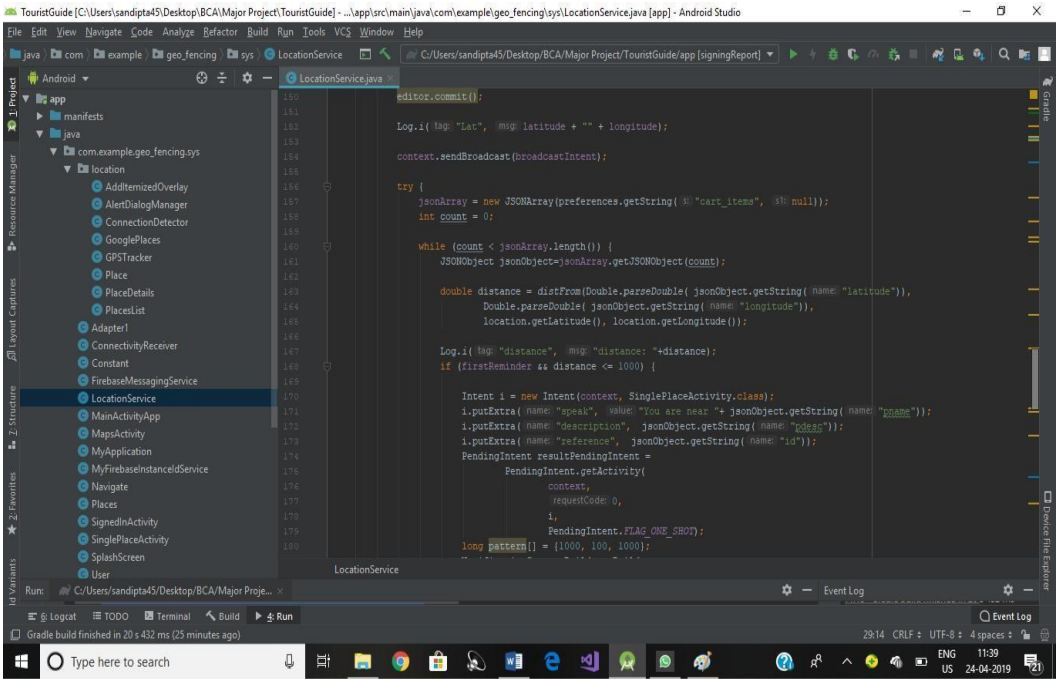

Fig 4.34

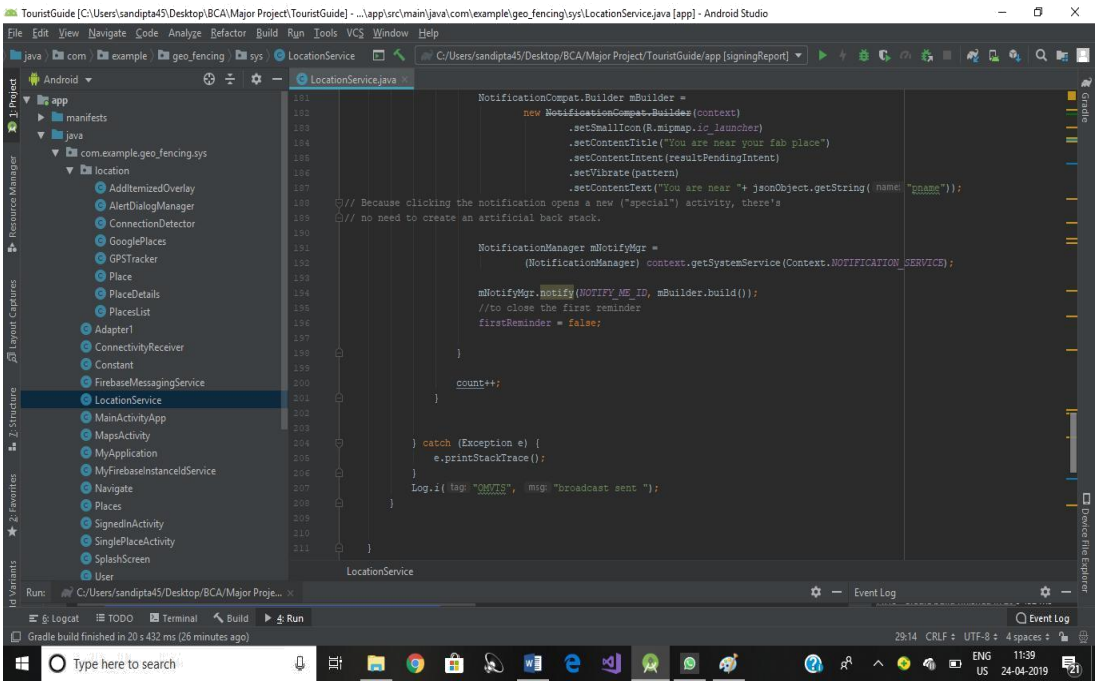

Fig 4.35

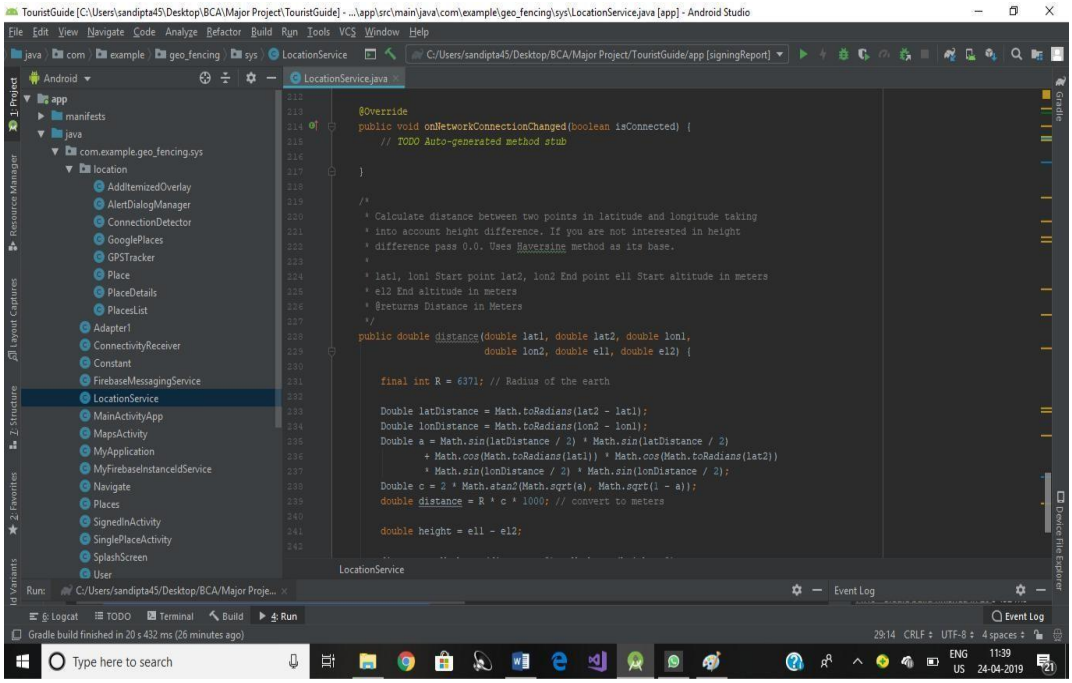

Fig 4.36

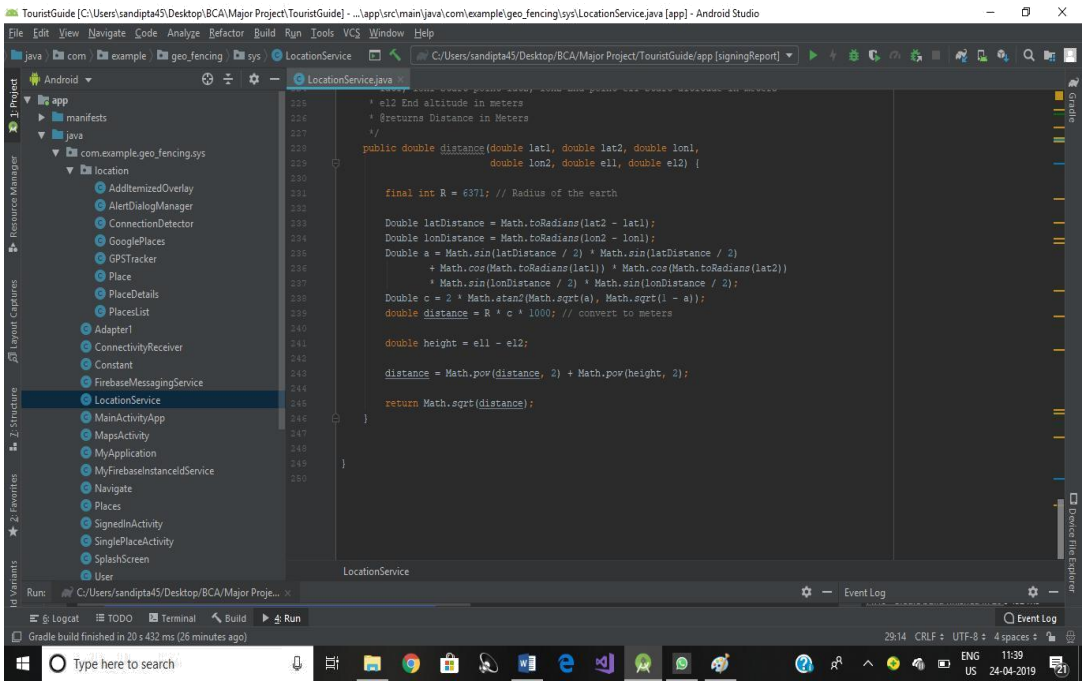

Fig 4.37

4.5 **Navigate Service**

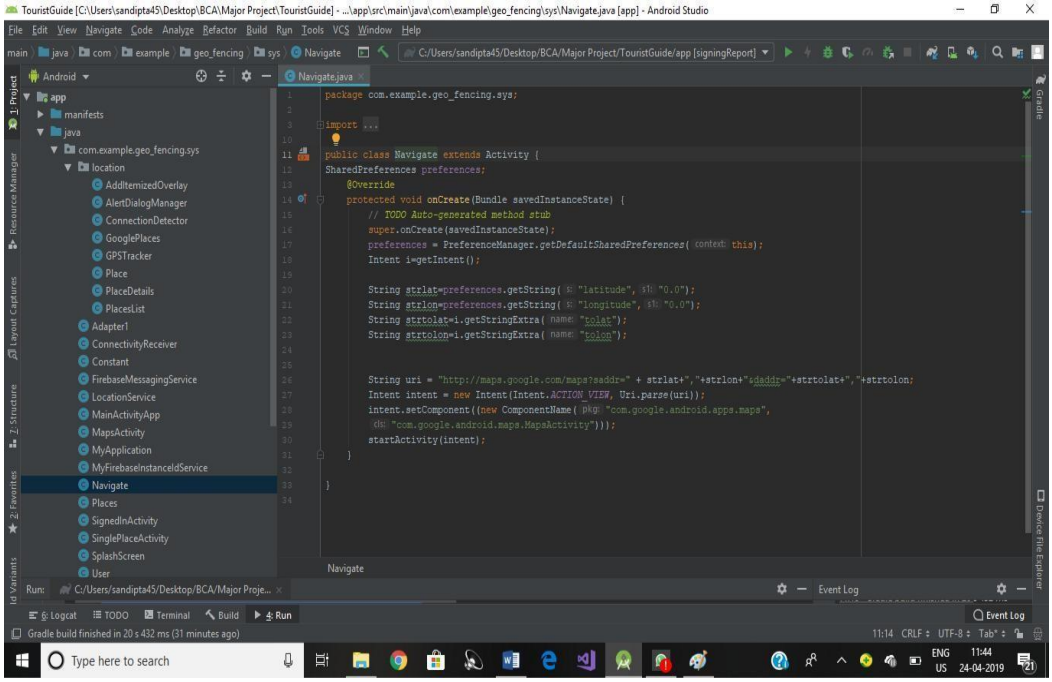

Fig 4.38

4.6 **The Google Services**

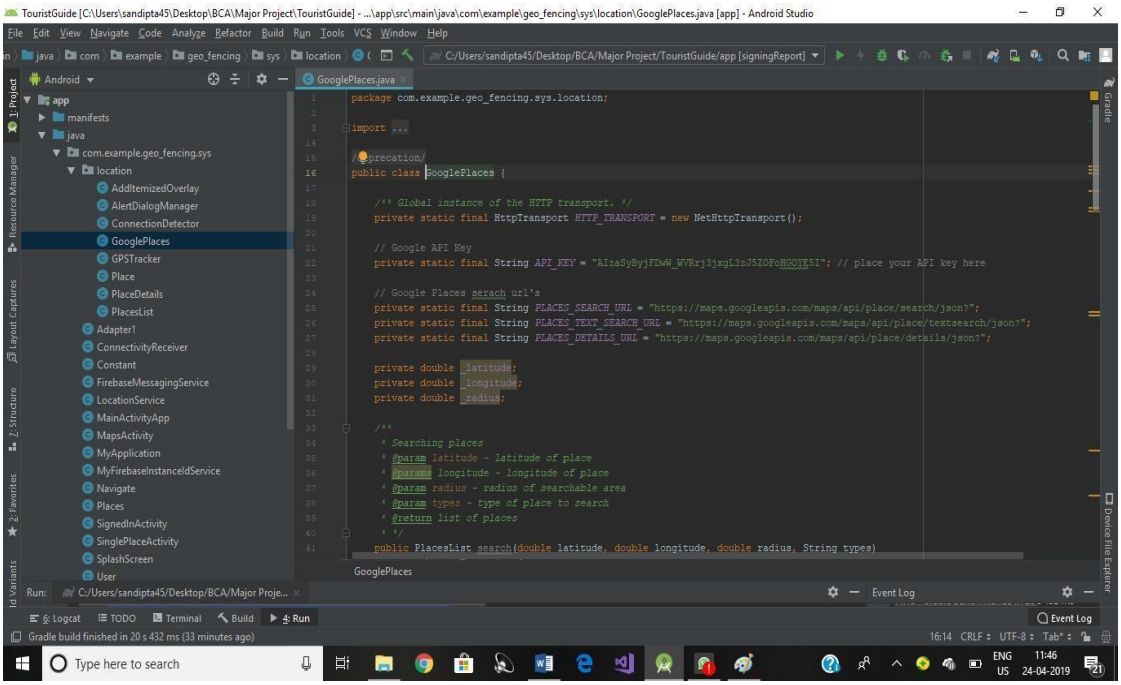

Fig 4.39

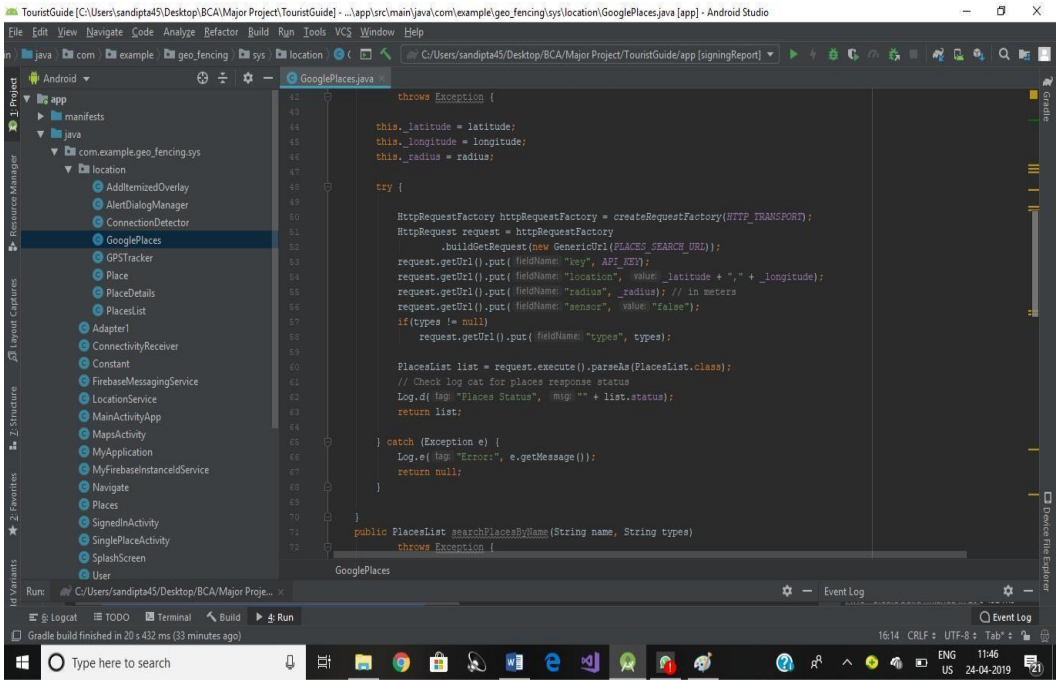

Fig 4.40

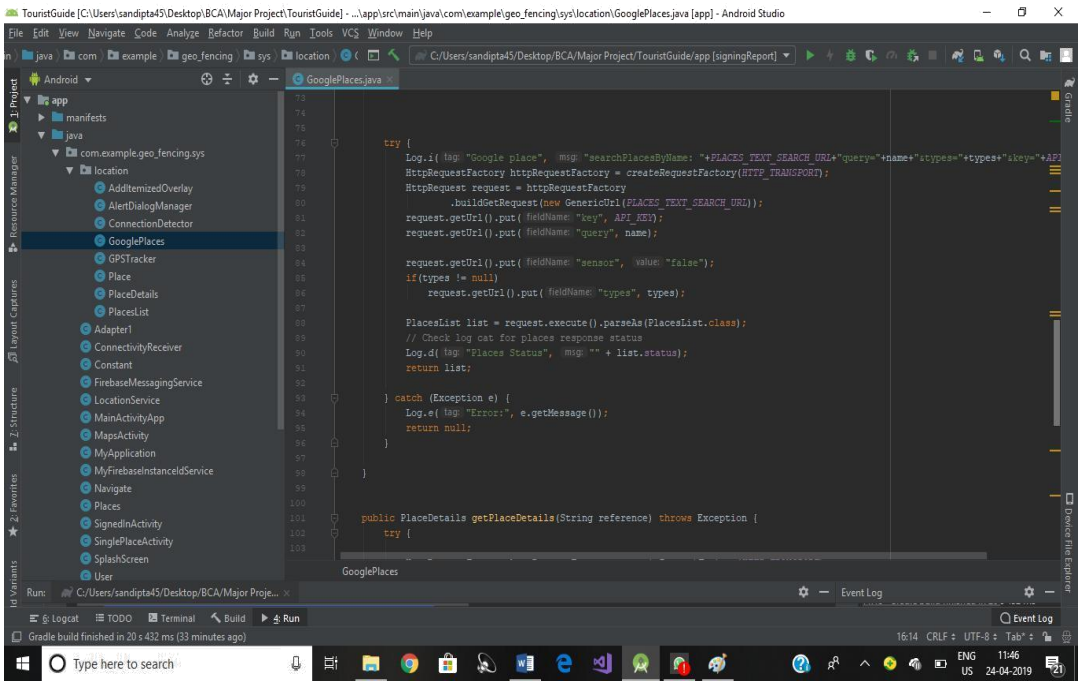

Fig 4.41

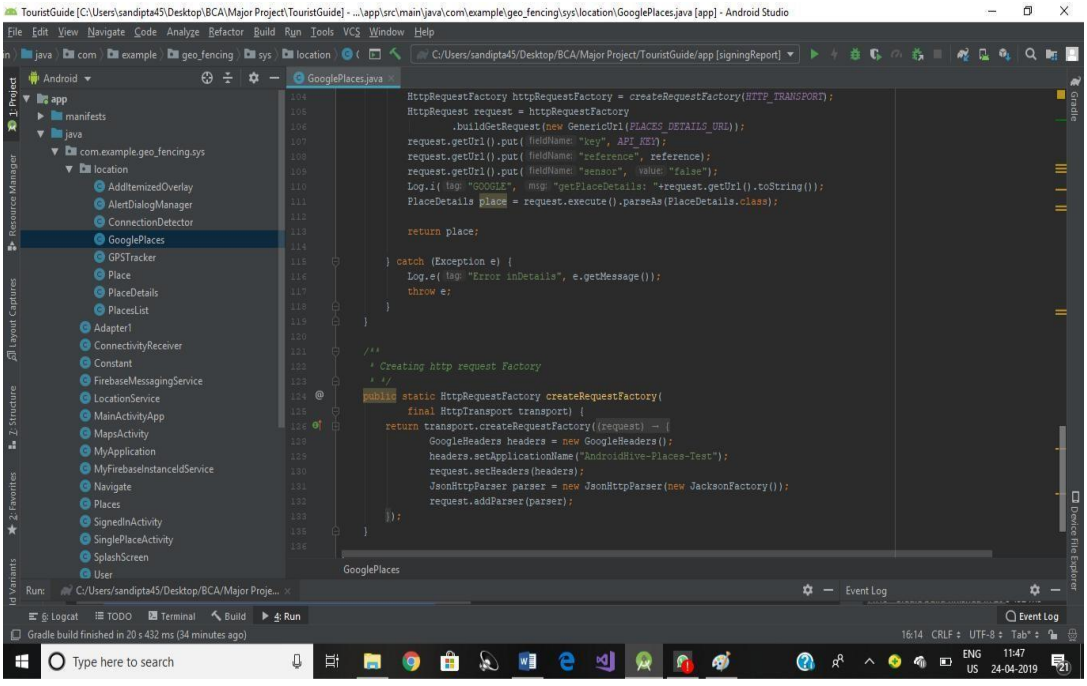

Fig 4.42

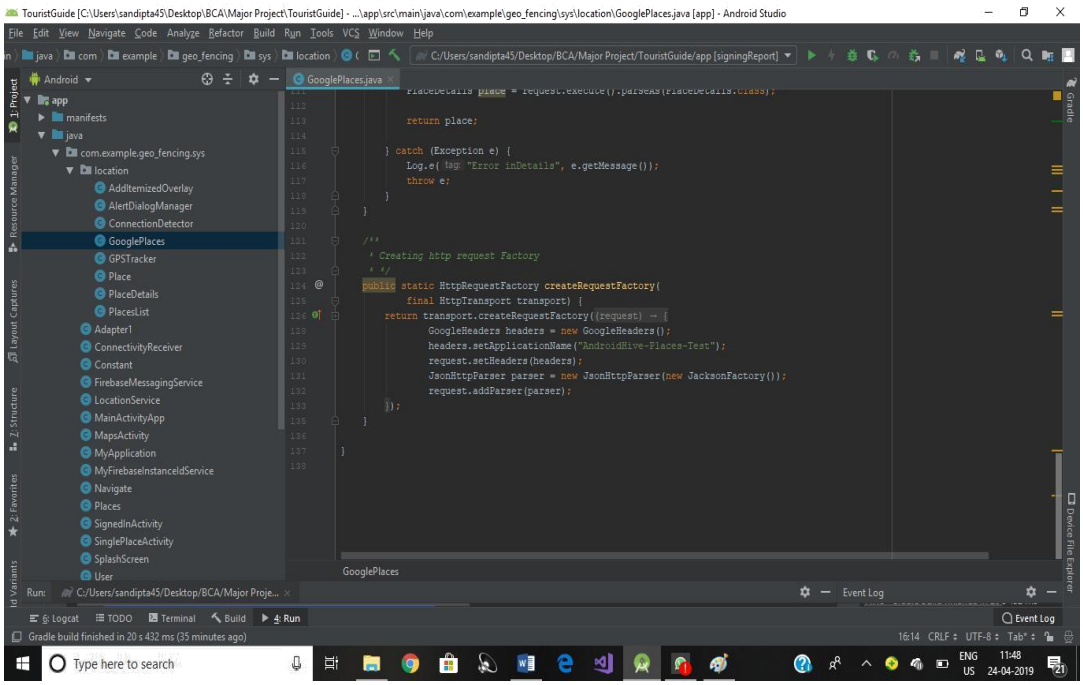

Fig 4.43

4.7 **Connection Detector**

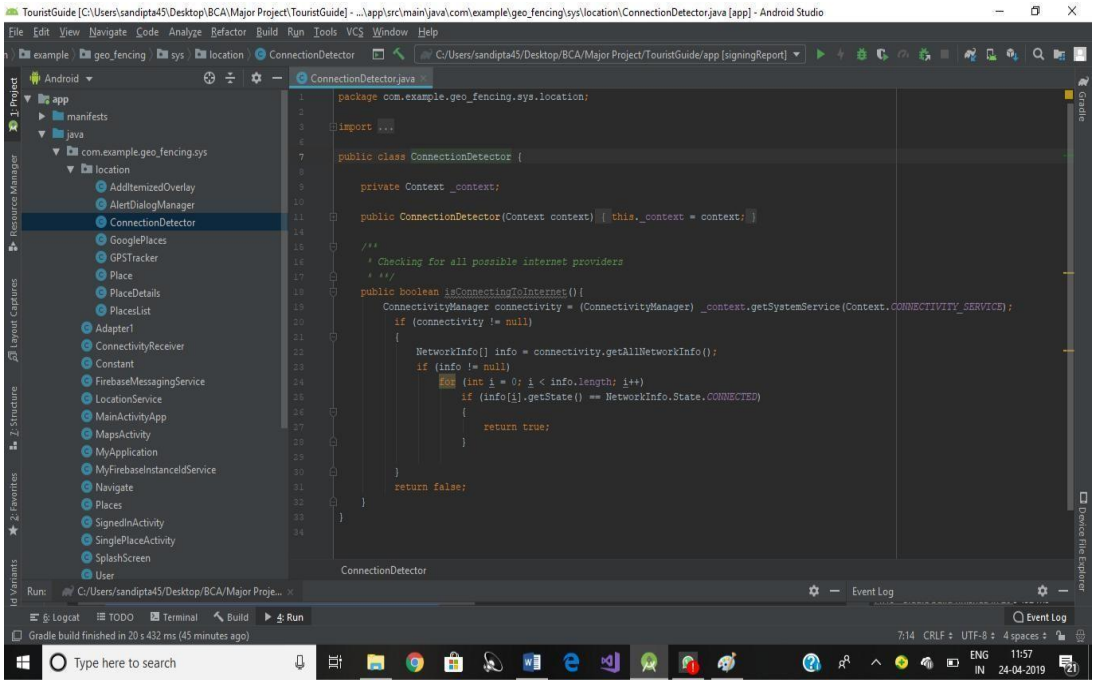

Fig 4.44

CHAPTER 5: RESULT AND DISCUSSION

5.1 Snapshots/Results

Fig 5.1 : The Main Screen

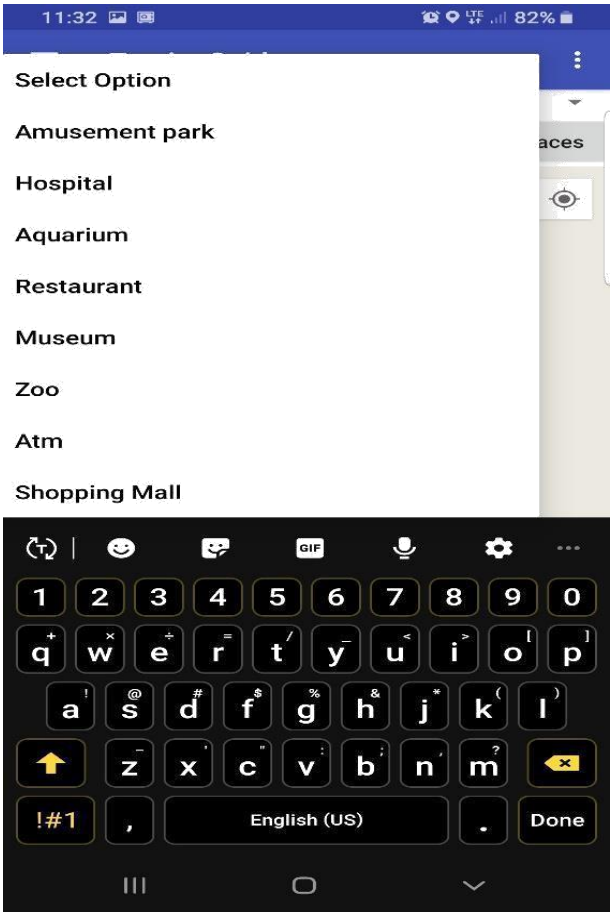

Fig 5.2: Selection of Place

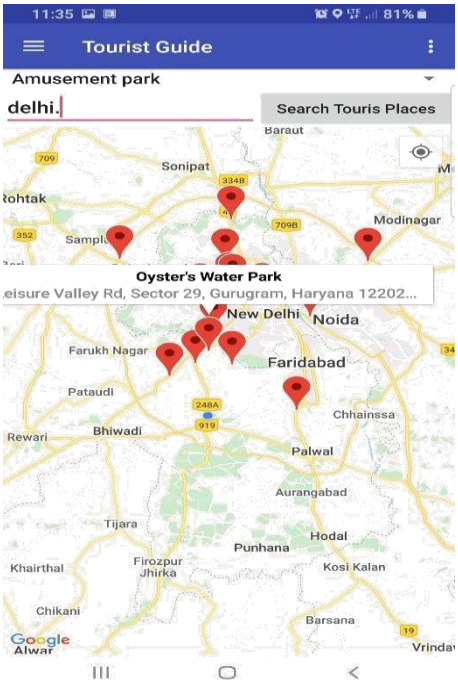

Fig. 5.3: Search Tourist Place

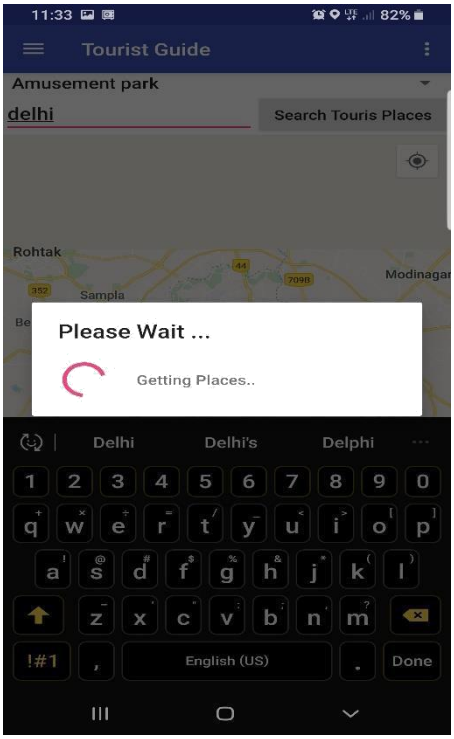

Fig 5.4: Processing

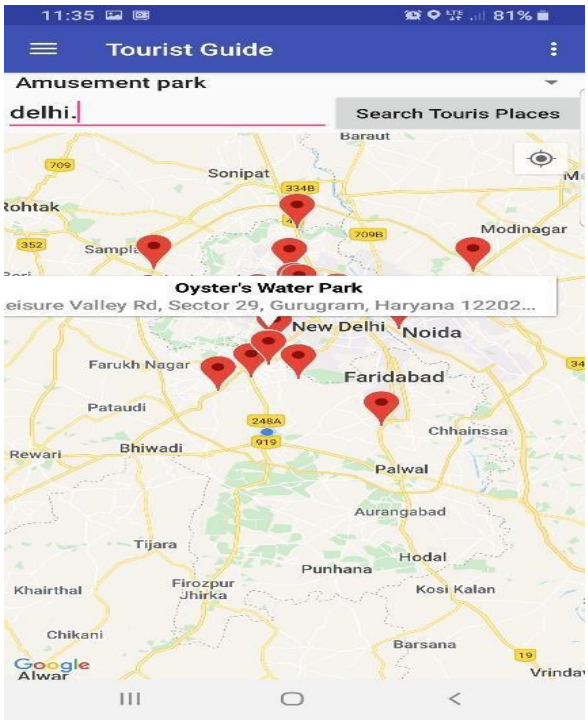

Fig 5.5 Select A Place

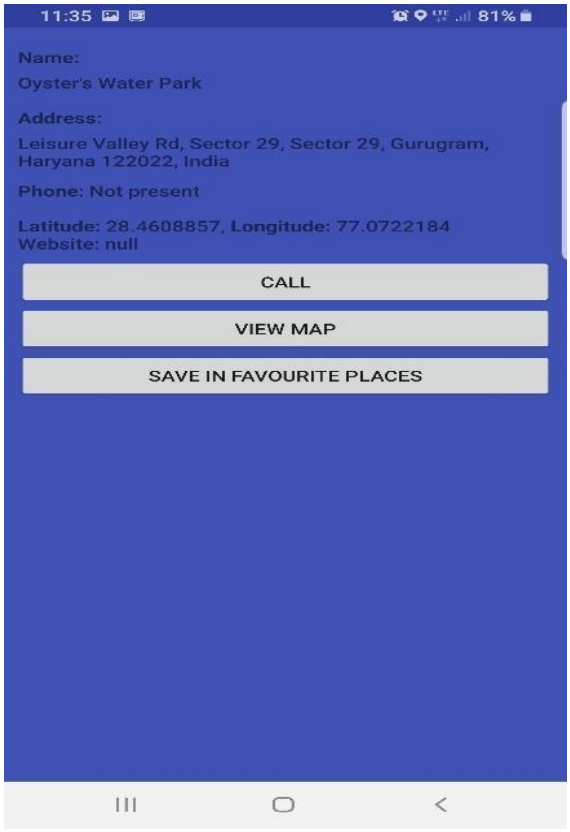

Fig 5.6: Detail of Place

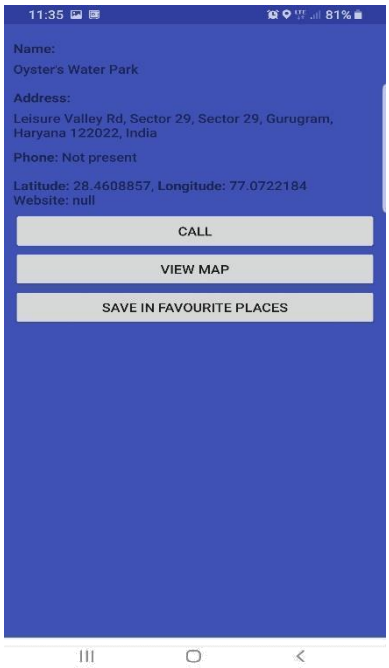

Fig 5.7: Navigation of The Place

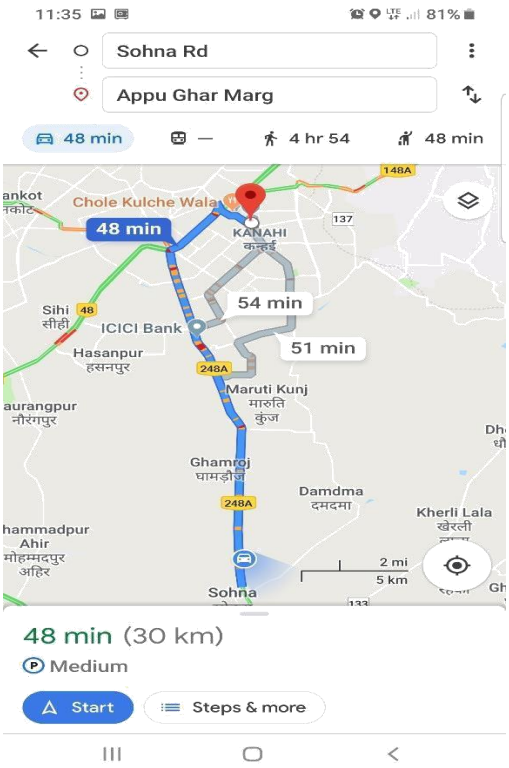

Fig 5.8: Map Google Location

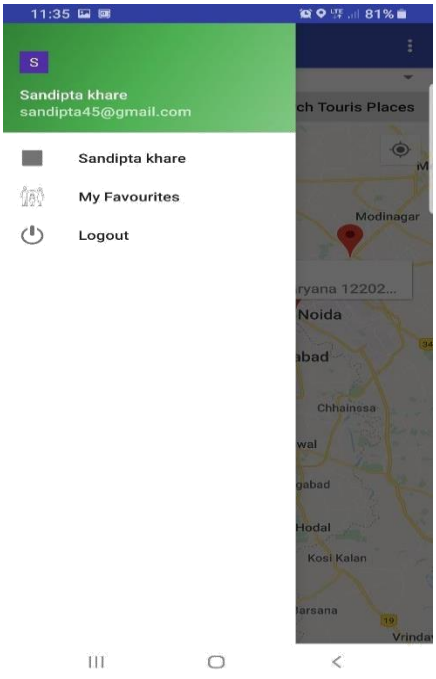

Fig 5.8: Account

Fig 5.9: Database

5.2 Discussion

Our project is on “Tourist Guide” Application which is built on the Android Studio. The above Screen Shots are the output of the Tourist Guide Application. The first Screen Shot, **Fig 5.1**, shows the home page of the Tourist Guide Application. The Home Page contains few things and that are Select Option, Location, Search Tourist Places and Maps. Select Option contains options such as Aquarium, Zoo, Amusement Park, Restaurant, etc. Location is to select place where you must visit. Search Tourist Place means to search place.

In **Fig 5.2,** you would see that we are selecting the Select Option and that is Amusement park.

In **Fig 5.3,** you would see that we are selection the location and that is Delhi. After selecting the location, we would click on Search Tourist Place.

In **Fig 5.4,** you would see that Application is searching for the Location.

In **Fig 5.5,** you would see that there would be various Amusement Park at Delhi. We had selected the Oyster’s Water Park.

In **Fig 5.6,** you would see that the Screen Shot would be showing the details of Oyster’s Water Park.

In **Fig 5.7,** you would see google map which would be showing the route for Oyster’s Water Park from your location. The google map would be open only when a user wants to find the location of the Oyster’s Water Park.

In **Fig 5.8,** you would see that there are three things and that are first is login id, second is Favorite and third is logout.

In **Fig 5.9,** you would see some places name which are stored as favourite. It only happens when the users like place and they click on favourite Button.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

It is a great way for people to spend their money and for cities to attract other people. In doing so, cities across the world attract millions of dollars just so foreigners get the chance to experience life in a new place. I hope to do much travelling myself because I believe it to be a wonderful way of living life to the fullest. Experiencing culture, religion, foods, sports, entertainment shows, and general ways of life in a new place is a great way to live life to the fullest. I one day hope I could open a hotel chain myself and attract many visitors to new cities. I want to travel the world and once I am free of the responsibility of raising children I want to spend two consecutive years just travelling the world. The biggest plus point for Tourist Guide is it would help anyone anywhere and at any time.

6.2 Future Scope

Application also causes income growth that has its own secondary impact on jobs. These indirect and induced effects could result in an increase in total employment by up to 8 times during the period 2014 to 2016. At its best, the Application economy could generate over “6, 00,000 jobs,” ICRIER Director & Chief Executive Rajat Kathuria said.

Function we can add in future to our application

- SOS function by just triple click on lock button that will send current location to the choices of three number a client want
- Enhanced GUI experience
- More user friendly

REFERENCES

[1] https://www.goodreads.com/quotes/tag/tourism

[2] https://www.youtube.com/watch?v=ZLNO2c7nqjw

[3] https://www.tutorialspoint.com/android/