

A PROJECT REPORT
ON
“LOCATION BASED PLACES OF INTEREST”

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER ENGINEERING

BY

ASHFAQUE AHMAD HAMEEDA KHATOON 15CO14
KHAN MOHAMMED YUSUF 16CO31
KHAN MOHAMMED HASMUDDIN 16CO30

UNDER THE GUIDANCE OF
PROF. SAYED AMER HASHMI



DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam’s Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206
2019-2020

AFFILIATED TO
UNIVERSITY OF MUMBAI

**A PROJECT II REPORT
ON**

“LOCATION BASED PLACES OF INTEREST”

**Submitted to
UNIVERSITY OF MUMBAI**

In Partial Fulfilment of the Requirement for the Award of

**BACHELOR’S DEGREE IN
COMPUTER ENGINEERING**

BY

**ASHFAQUE AHMAD HAMEEDA KHATOON 15CO14
KHAN MOHAMMMED YUSUF 16CO31
KHAN MOHAMMED HASMUDDIN 16CO30**

**UNDER THE GUIDANCE OF
PROF. SAYED AMER HASHMI**



**DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam’s Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206**

**2019-2020
AFFILIATED TO**



UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,

Khandagaon, New Panvel - 410206



CERTIFICATE

This is certify that the project entitled

“LOCATION BASED PLACES OF INTEREST“

submitted by

ASHFAQUE AMAD HAMEEDA KHATOON	15CO14
KHAN MOHAMMED YUSUF	16CO31
KHAN MOHAMMED HASMUDDIN	16CO30

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at *Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai* under the University of MUMBAI. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. SAYED AMER HASHMI)
Project Supervisor

(Prof. KALPANA BHODKE)
Project Coordinator

(Prof. TABREZ KHAN)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express my sincere thanks to my guide **Prof.SAYED AMER HASHMI**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

We also grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. TABREZ KHAN**, Head of Department of Computer Engineering and **Prof. KALPANANA BHODKE**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

ASHFAQE AHMAD HAMEEDA KHATOON
KHAN MOHAMMED YUSUF
KHAN MOHAMMED HASMUDDIN

Project I Approval for Bachelor of Engineering

This project entitled *LOCATION BASED PLACES OF INTEREST* by *ASH-FAQUE AHMAD HAMEEDA KHATOON, KHAN MOHAMMED YUSUF, KHAN MOHAMMED HASMUDDIN* is approved for the degree of *Bachelor of Engineering in Department of Computer Engineering*.

Examiners

1.
2.

Supervisors

1.
2.

Chairman

.....

Declaration

We declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I 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.

ASHFAQUE AHMAD HAMEEDA KHATOON

15CO14

KHAN MOHAMMED YUSUF

16CO31

KHAN MOHAMMED HASMUDDIN

16CO30

ABSTRACT

Title:Location based places of Interest

Location based Services offer many advantages to the mobile users to retrieve the information about their current location and process that data to get more useful information near to their location. With the help of this app in phones and through Web Services using GPRS, Location based Services can be implemented on Android based smart phones to provide these value-added services: providing routing information, helping them find nearby hotels, Malls, Park, Hospitals . We propose the implementation of Location based services through Google Web Services and Google map APIs on Android Phones to give multiple services to the user based on their Location.

Keywords and Glossary

Keywords:

Live Tracking Automation, Destination Detection, Automated Suggestion.

Glossary:

A:

Altitude - distance of a point from a chosen reference surface along a line perpendicular to that surface.

API - a set of functions and procedures allowing the creation of applications that

access the features or data of an operating system, application, or other service.

Application - a software program that runs on your computer. Analysis - detailed examination of the elements or structure of something.

Automation - the use or introduction of automatic equipment in a manufacturing or other process or facility.

D:

Database - a structured set of data held in a computer, especially one that is accessible in various ways.

E:

Event - a thing that happens or takes place, especially one of importance.

G:

Geocoding- translation of one form of location into another.

Geofence- A virtual perimeter for a real world geographic entity. Defined as the extent of an imaginary fence around a location or place, a geofence can have polygonal vector boundaries or be constrained by a radius around a centroid.

Geolocation-The identification of the geographic location of an object.

Geospatial-The combination of spatial software and analysis method with geographic data sets.

I:

Integrate - combine (one thing) with another to form a whole.

Instant message - send (someone) an instant message.

L:

Location - identifiable geographic place [ISO 19112]. Explicitly identified by name or geocode.

Location Tracking - technologies that physically locate and electronically record and track the movement of people or objects

M:

Maintenance - the process of preserving a condition or situation or the state of being preserved.

Mobile Application - application software designed to run on a mobile device, such as a smartphone or tablet computer.

Monitoring - observe and check the progress or quality of (something) over a period of time; keep under systematic review.

N:

Notification - the action of notifying someone or something.

P:

Place - An aggregate term that can be used to refer to a point, a location, a POI or a meaningful combination thereof.

Point-0-dimensional geometric primitive, representing a position. [ISO 19107].

Point Of Interest Often abbreviated to POI. A specific point location that is of interest. Frequently used to refer to business locations and tourist or well know sites and locations. A Point-of-Interest (POI) is the common expression for a place on maps,

navigations systems or route planners. These places are considered to have an exceptional meaning for people - they embody a special interest regarding an activity on that place such as a restaurant, a hotel, a petrol station.

POI - The properties of a physical position (location) in the world that may correspond to the past, present, or future location of a person, event, or device.

Position-data type that describes a point or geometry potentially occupied by an object or person.

R:

Reminder - a thing that causes someone to remember something.

S:

Schedule - a plan for carrying out a process or procedure, giving lists of intended events and times.

Semantic Web - A group of technologies that allow computer interpretation of the semantics of information available on the World.

T:

Tracking system - observing of persons or objects on the move and supplying a ordered sequence of data for further processing.

V:

Validation - the action of checking or proving the validity or accuracy of something.

W:

Web application - an application program that is stored on a remote server and de-

livered over the Internet through a browser interface.

Web services - a software service used to communicate between two devices on a network.

O:

Opportunities - a time or set of circumstances that makes it possible to do something.



Contents

Acknowledgement	iii
Project I Approval for Bachelor of Engineering	iv
Declaration	v
Abstract	vi
Table of Contents	xiv
1 Introduction	2
1.1 Purpose	2
1.2 Project Goals and Objectives	3
2 Literature Review	4
2.1 Personalized Location Based Recommendations For Location Based Social Network	4
2.1.1 Weaknesses	4
2.1.2 How to Overcome	5
2.2 Enhanced Sentiment Classification Using Geo Location Tweets	5
2.2.1 Weaknesses	6

2.2.2	How to Overcome	6
2.3	Location Based Reminder Android App Using Google Maps API . .	7
2.3.1	Weaknesses	7
2.3.2	How to Overcome	7
3	Project Planning	8
3.1	Members and Capabilities	8
3.2	Roles and Responsibilities	9
3.3	Assumptions and Constraints	9
3.4	Project Management Approach	9
3.5	Ground Rules for the Project	9
3.6	Project Budget	10
3.7	Project Timeline	11
4	Software Requirements Specification	12
4.1	Project Requirements	12
4.1.1	Software Requirements	12
4.1.2	Hardware Requirements	13
5	System Design	14
5.0.1	Project Architecture	14
5.1	Methodology	15
5.1.1	Class Diagram	15

5.1.2	DFD Diagram	16
5.1.3	Use Case Diagram	18
5.1.4	Activity Diagram	19
5.1.5	Sequence Diagram	20
6	Implementation	21
6.1	Module 1	21
6.2	Module 2	25
6.3	Module 3	29
6.4	Module 4	34
6.5	Module 5	41
7	System Testing	43
7.1	Test Cases and Test Results	43
7.2	Sample of a Test Case	44
8	Screenshots of Project	46
9	Conclusion and Future Scope	67
9.1	Conclusion	67
9.2	Future Scope	68
	References	68



List of Figures

3.1	Gantt Chart For The Current Semester Created In Smartsheet	11
5.1	System Architecture	14
5.2	Class Diagram For Location Based Interest App	15
5.3	DFD Level 0 and level 1 For Location Based Interest App	16
5.4	DFD level 2 For Location Based Interest App	17
5.5	Use Case For Location Based Interest App	18
5.6	Activity Diagram For Location Based Interest App	19
5.7	Sequence Diagram For Location Based Interest App	20
6.1	REGISTER	22
6.2	LIST OF INTEREST	25
6.3	UPDATE POI	29
6.4	FETCH DETAILS	34
6.5	CAPTURE POI LIST	41

List of Tables

3.1	Table of Capabilities	8
3.2	Table of Responsibilities	9



Chapter 1

Introduction

When people navigate through cities to get work or visit new areas they pass by many potential point of interest i.e. Supermarkets, Hospital, Gym, Cinemas, Motel, Mall, Parks etc. Yet they still have to query Google and/or check Google Map each Time they need visit this places. The Location Based Places of Interest app would be an application that operates on your phone and automatically detects bookmarks, and store details of these places of interests determined by you whenever you navigate an area or go on a trip.

1.1 Purpose

Location Based Places Of Interest provides a lot of facility to their user. The scope of Location Based Places Of Interest to record the details various activities of user. It will simplify the task of user and paper work. During implementation every user will be given appropriate training to suit their specific needs. Specific support will also

be provided at key points within the academic calendar. Training will be provided on a timely basis, and you will be trained as the new is Seating Arrangement System rolled out to your area of responsibility

1.2 Project Goals and Objectives

When people navigate through cities to get work or visit new areas they pass by many potential point of interest i.e. Supermarkets, Hospital, Gym, Cinemas, Motel, Mall, Parks etc. Yet they still have to query Google and/or check Google Map each Time they need visit this places. The Location Based Places of Interest app would be an application that operates on your phone and automatically detects bookmarks, and store details of these places of interests determined by you whenever you navigate an area or go on a trip.

Chapter 2

Literature Review

2.1 Personalized Location Based Recommendations For Location Based Social Network

With the development of social networks and wire-less communication technology, location-based social networks(LBSNs) are developing rapidly. Personalized location service in location-based social networks can provide users with a new point-of-interest (POI). Compared to traditional recommendation, point-of-interest recommendation integrates the social network with the location that connects online users and physical. places.

2.1.1 Weaknesses

[noitemsep]Data sparsity happens, when the check-in data in the recommendation system are insufficient for identifying similar users and POI and it is one of the major issues limiting the quality of recommendations. The imbalance

case is attributed to the fact that high density areas are usually workplaces or crowded areas among which most human activities take place. As a result, the user's personalized distribution will depend on dense areas or sparse areas. It requires many location suggestion hence an overhead of managing on their basis. It does not suggest any offers about their point of interest.

2.1.2 How to Overcome

[noitemsep] Taking into account the characteristics that people tend to explore new events and evolution of user preference, one is more likely to explore new point-of-interests. This Location Recommendation provides help to find users new places of interest according to their suggestion. By modeling spatial properties using a location density estimation approach to generate a unique distribution for each user. By Analyzing many social networks app to show suggestion of interest by their user need.

2.2 Enhanced Sentiment Classification Using Geo Location Tweets

Twitter is an important source of knowledge. It is useful to perform a moral assessment of the service. The classification of emotions on tweets has given a financial and effective form to determine public sentiment. In this research, more than 30,000 location-based tweets are collected from the Twitter page also emotions (emoji's) and the abbreviated form of words are used for the sentiment classification

by replacing in top form, then perform the opinion classification in 5 categories, extreme positive, positive, neutral, negative and extreme negative classes. Suppliers and shareholders help to take into account the opinions of modes, services, transactions, etc from a particular geographical location.

2.2.1 Weaknesses

[noitemsep]We can get the maximum information through social media because character limit. It can be wrong information because of every information on social media is not wrong. It can take more time if network is not present at that location. Sentiment Analysis can be different at Multiple Location Like in Mumbai it can be different from pune.

2.2.2 How to Overcome

[noitemsep]In sentiment analysis, it takes the input from user's choice. In our application we will provide the offers on the product when users visit that particular location. Sentiment analysis is used to determine the opinion of user's online text data towards services or product which they wrote.

2.3 Location Based Reminder Android App Using Google Maps

API

Many times it is not confirmed that we will be present at the specific location for the work for which we have set the reminder. Instead it is beneficial if the notification alarm triggers when we are actually present near or at that specific location. To remind modern people of something at a specific time and location, Smart Location Reminder is a boon.

2.3.1 Weaknesses

[noitemsep]SQLite Database is used which should not store data permanently in it. Issue in Showing Correct Path when user is nearby location. Cannot store user favorite location

2.3.2 How to Overcome

[noitemsep]Shows Alert Notification when user is near by the location to not be wrong path. Time can be Modify to according to their changes of plans.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Mohd. Yusuf	Database, Firebase connection
2	Ashfaque Ahmad	UI design
3	Mohammed Khan	UI design

Work Breakdown Structure

1. All of the members are equally important in developing the project.
2. We work on a different part of the project based on one's capability.
3. Firstly we came up with documentation, And based on the documentation we set our goal and created a blueprint.
4. We then started going hands-on with the project to develop it according to the flow as decided earlier.

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Mohd. Yusuf	Team Leader	database, Firebase connection
2	Ashfaque Ahmad	Team member	UI design
3	Mohammed Khan	Team member	UI design

3.3 Assumptions and Constraints

- a. To reduce the stress and work of user.
- b. Users can access information easily.
- c. Teachers can send notifications at any time to all students.

3.4 Project Management Approach

- a. Planning of project.
- b. Defining the scope of the project.
- c. Estimation of time and It's management.
- d. Creating Gantt Charts and properly assigning tasks to members.
- e. Reporting the progress of project with the guide.

3.5 Ground Rules for the Project

- a. Properly planning and gathering relevant information is very important.

- b. Developing a Blueprint of the project and work accordingly.
- c. All the members should report to the guide whenever required
- d. Setting up small goals every week.
- e. Achieving the small goal within that span of time.
- f. Keeping tracks of the progress towards project.

3.6 Project Budget

- a. It is a light project.
- b. Cost of the project is very low and efficient.

3.7 Project Timeline

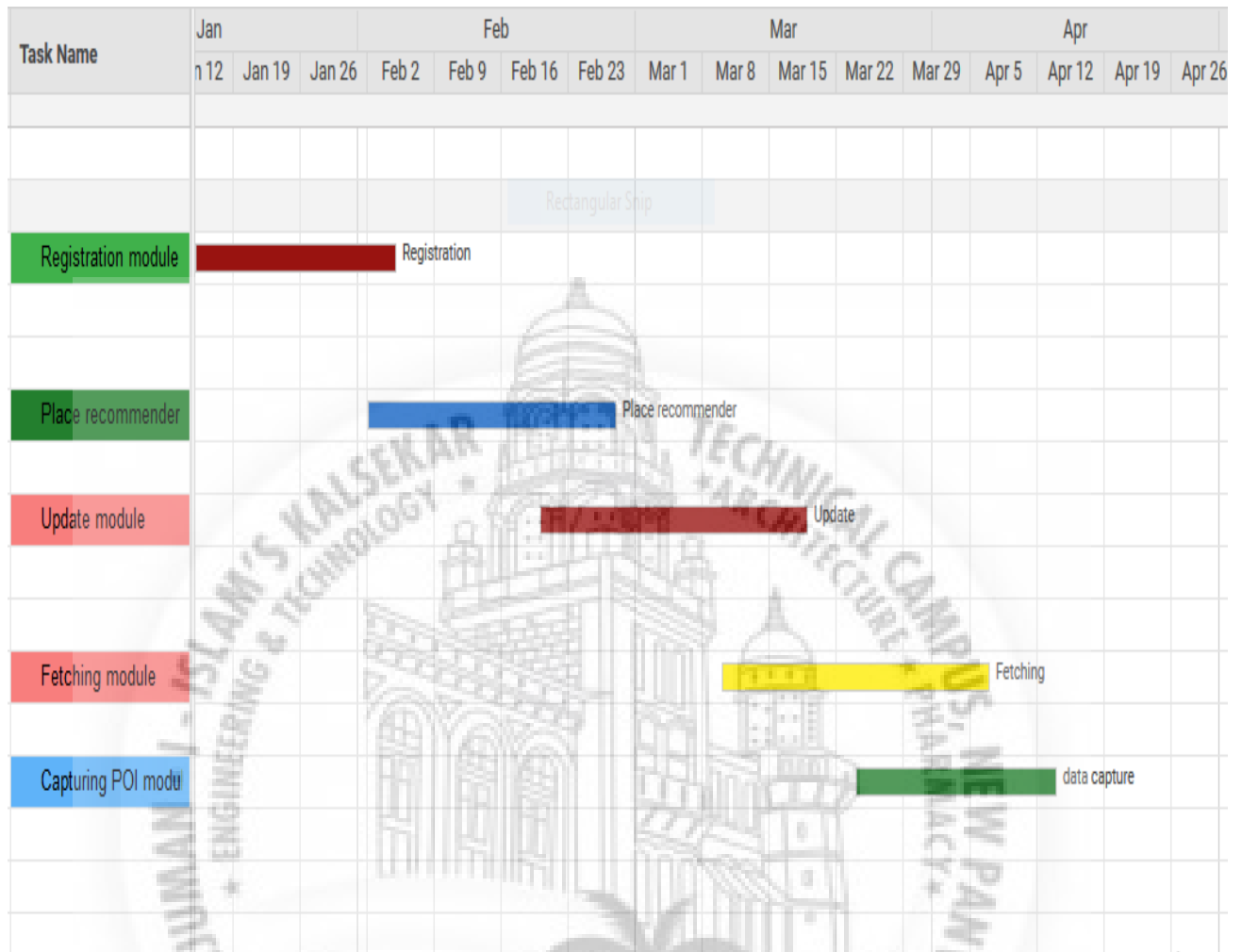


Figure 3.1: Gantt Chart For The Current Semester Created In Smartsheet

Chapter 4

Software Requirements Specification

4.1 Project Requirements

4.1.1 Software Requirements

- Android Studio Tool.
- Google API.
- Windows 10 64-Bit/Linux.
- FireBase.
- Adobe XD.

4.1.2 Hardware Requirements

- Processor intel i5.
- Ram 6GB DDR4.
- Android Device-OS-Marshmallow(6.0).



Chapter 5

System Design

5.0.1 Project Architecture

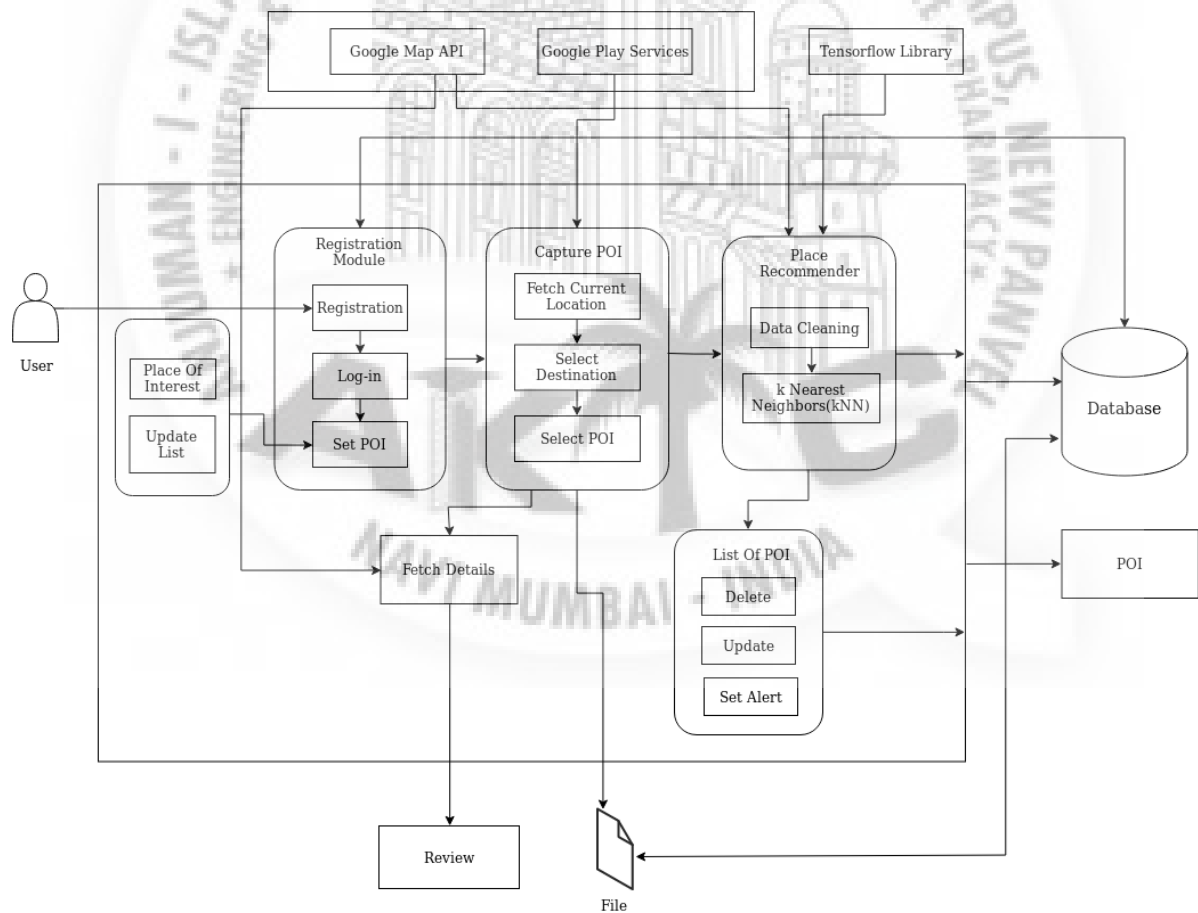


Figure 5.1: System Architecture

5.1 Methodology

5.1.1 Class Diagram

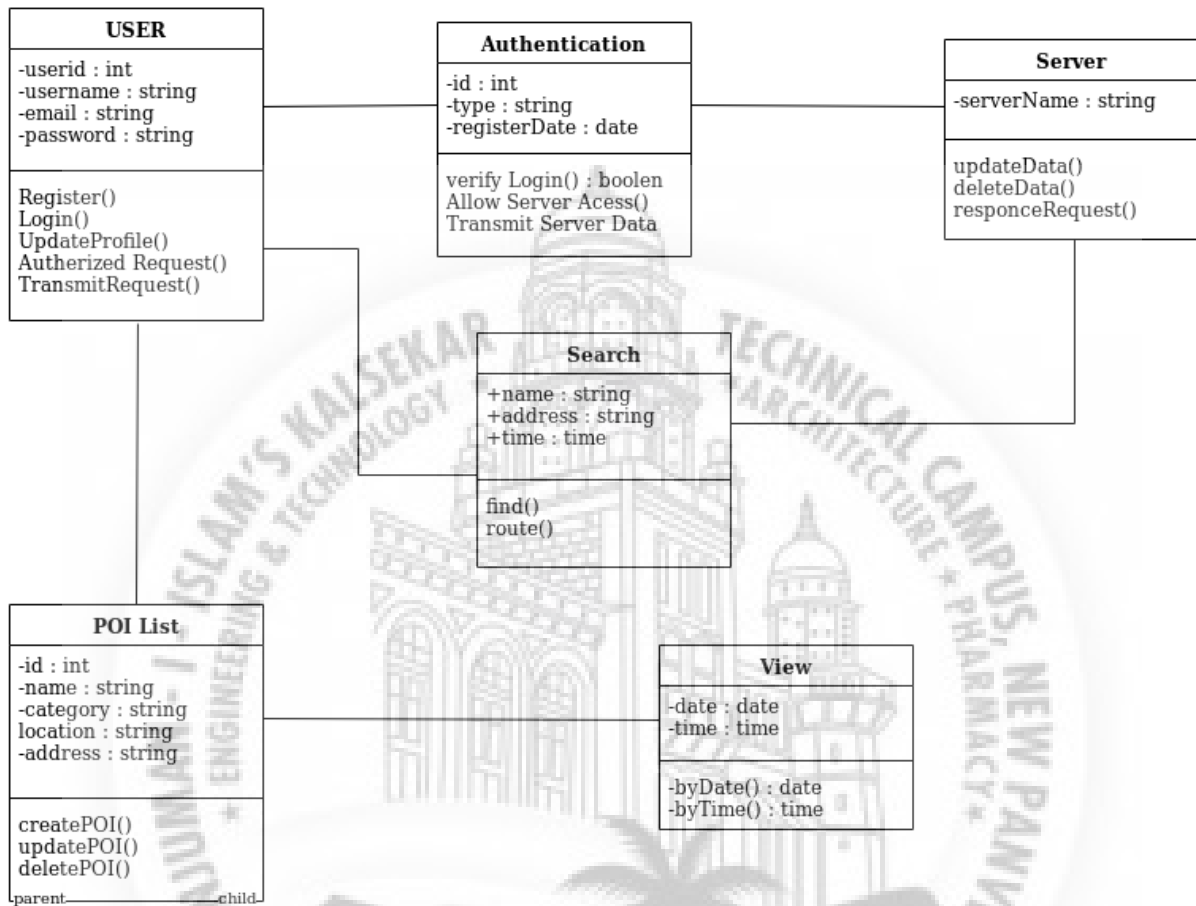


Figure 5.2: Class Diagram For Location Based Interest App

5.1.2 DFD Diagram

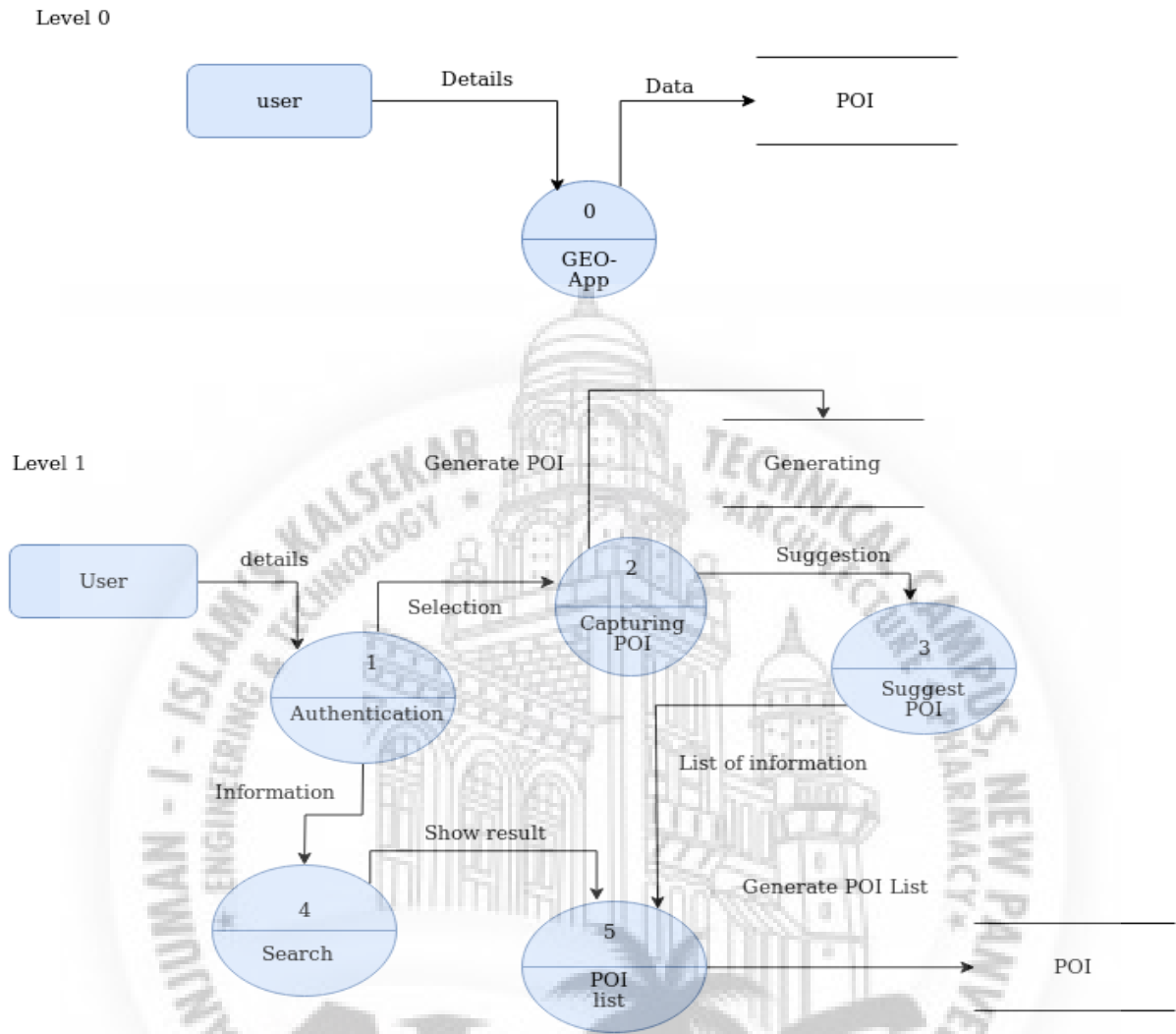


Figure 5.3: DFD Level 0 and level 1 For Location Based Interest App

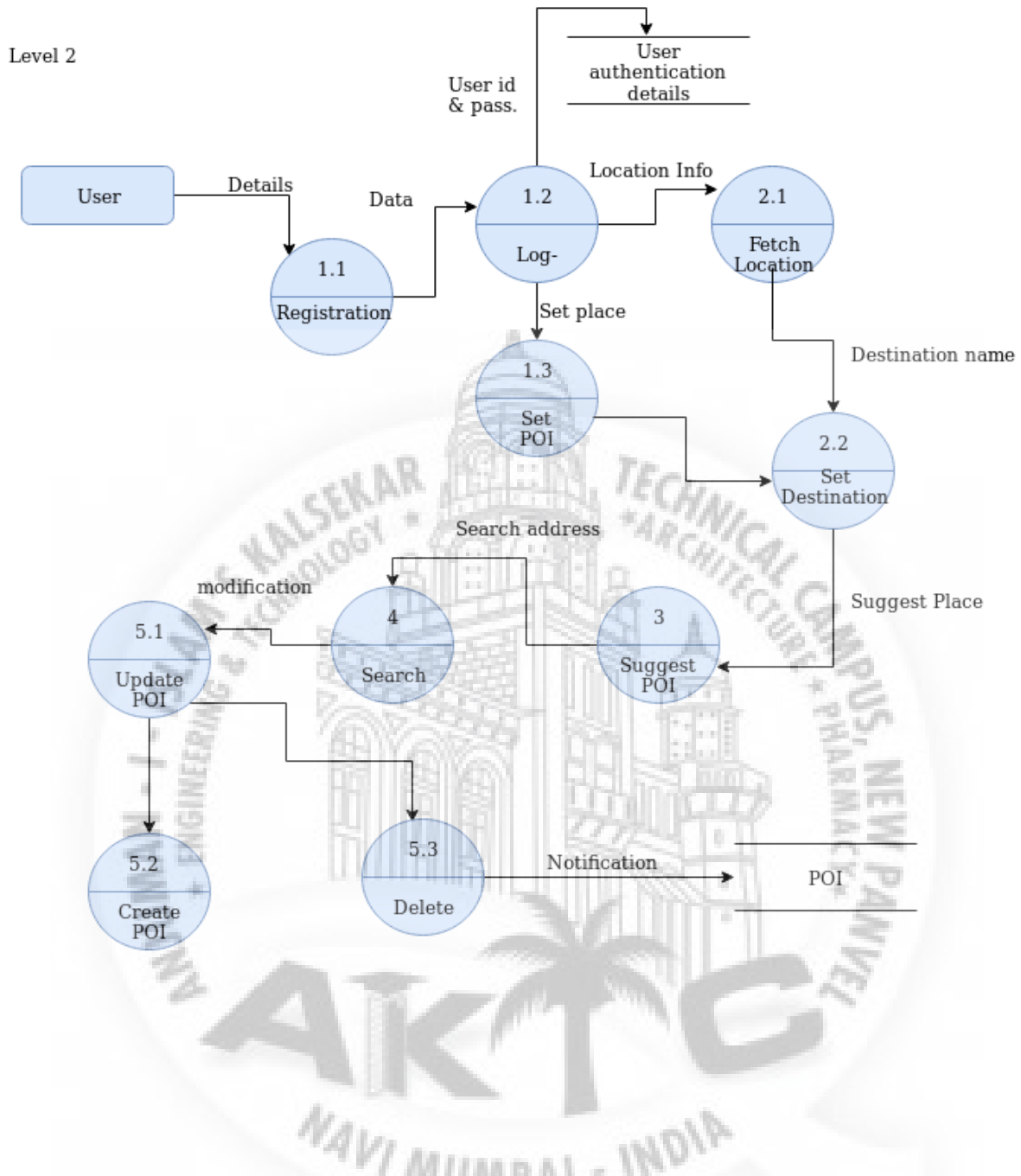


Fig: DFD diagram for Place recommender App "GEO_INTEREST"

Figure 5.4: DFD level 2 For Location Based Interest App

5.1.3 Use Case Diagram

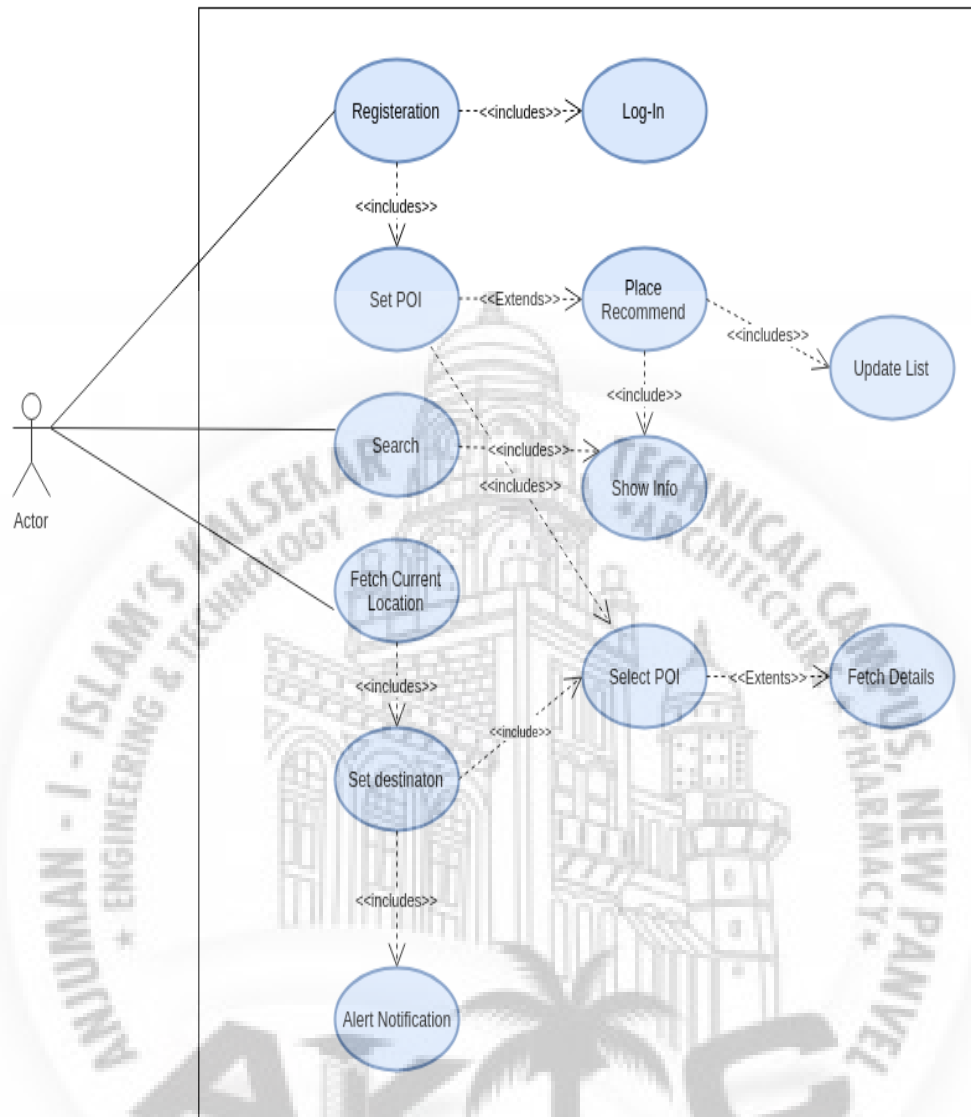


Figure 5.5: Use Case For Location Based Interest App

5.1.4 Activity Diagram

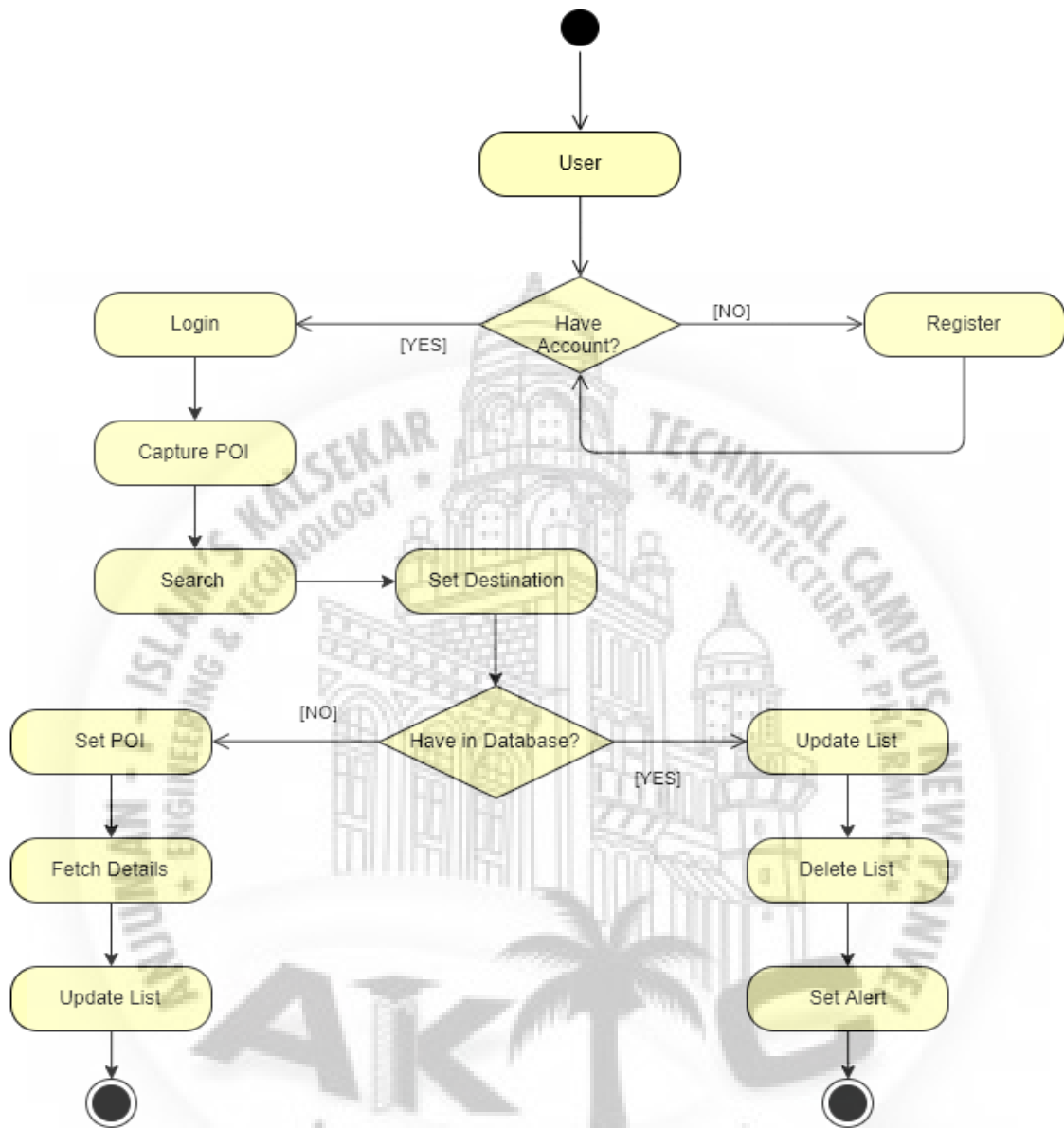


Figure 5.6: Activity Diagram For Location Based Interest App

5.1.5 Sequence Diagram

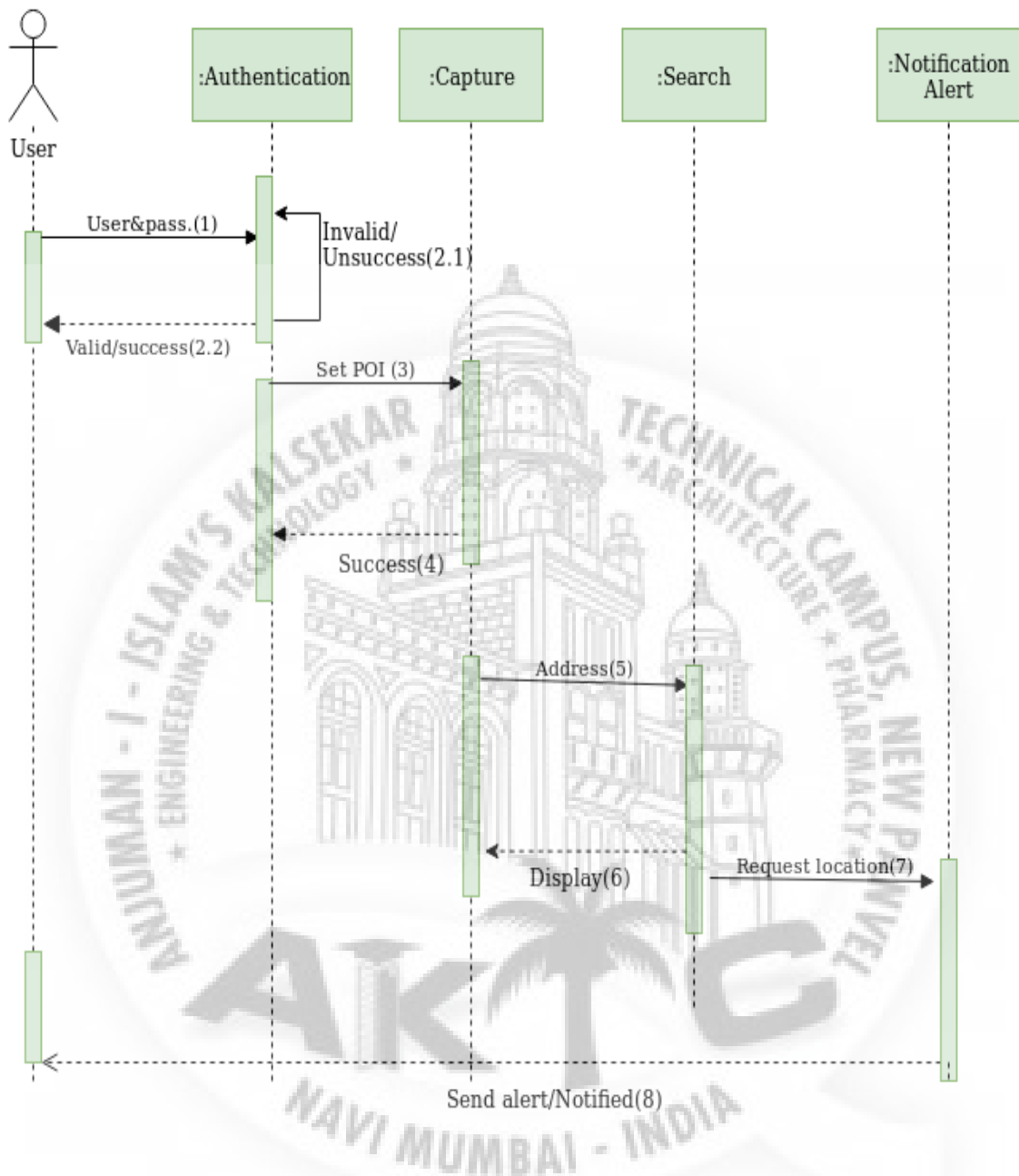


Fig:Sequence diagram for Place recommender app GEO-INTEREST

Figure 5.7: Sequence Diagram For Location Based Interest App

Chapter 6

Implementation

6.1 Module 1

Registration Module

A user enters the details of username & password. The passwords are stored in encrypted hash ensuring the security aspects. With the help of username / password a user will be able to gain access to the Application. After Registration of User It Automatically Login the User to App Login to the System And new Page Appear of Search Bar In That User has to set POI Of their Needs.

Figure 6.1: REGISTER

```

1 public class Register_Page extends AppCompatActivity {
2
3
4     private TextInputLayout FullName, Email, Password, Repassword, PhoneNumber;
5     private RadioButton GenderM;
6     private Button Register, Backbtn;
7     private FirebaseAuth mAuth;
8     private FirebaseDatabase mDatabase;
9     private DatabaseReference mRef;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_register_page);
15
16         initView();
17         mAuth = FirebaseAuth.getInstance();
18         mDatabase = FirebaseDatabase.getInstance();
19         mRef = mDatabase.getReference("User");

```

```
20     Register.setOnClickListener(this::createUser);
21     Backbtn.setOnClickListener(this::backButton);
22
23 }
24
25 mAuth.createUserWithEmailAndPassword(email, password)
26 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
27     @Override
28     public void onComplete(@NonNull Task<AuthResult> task) {
29         if (task.isSuccessful()) {
30             Objects.requireNonNull(mAuth.getCurrentUser()).sendEmailVerification()
31             .addOnCompleteListener(new OnCompleteListener<Void>() {
32                 @Override
33                 public void onComplete(@NonNull Task<Void> task) {
34                     if (task.isSuccessful()) {
35
36                         getValues(fullName, email, password, phoneNumber, gender, isNew);
37
38                         User user = new User(fullName, email, password, phoneNumber, gender, isNew);
39                         mRef.child(mAuth.getCurrentUser().getUid()).setValue(user);
40                         double currentLocation = 0.0;
41                         mRef.child(mAuth.getCurrentUser().getUid()).child("Location").child("
42                             lastLocation").child("Latitude").setValue(currentLocation);
43
44                         mRef.child(mAuth.getCurrentUser().getUid()).child("Location").child("
45                             lastLocation").child("Longitude").setValue(currentLocation);
46
47                         Toast.makeText(Register_Page.this, "User Created Successfully\n Waiting For
48                             Authentication ", Toast.LENGTH_LONG).show();
49                         startActivity(new Intent(Register_Page.this, MainActivity.class));
50                     }
51                 }
52             });
53         }
54     }
55 }
```

```
48  overridePendingTransition(R.anim.slide_in_left , R.anim.slide_out_right);
49  } else {
50
51  Toast.makeText(Register_Page.this , task.getException().getMessage() , Toast.
    LENGTHLONG).show();
52  }
53  }
54  });
55
56
57  } else {
58  Toast.makeText(Register_Page.this , task.getException().getMessage() , Toast.
    LENGTHLONG).show();
59  }
60  }
61  }
62  )
63  .addOnFailureListener(new OnFailureListener() {
64  @Override
65  public void onFailure(@NonNull Exception e) {
66  if (e instanceof FirebaseAuthUserCollisionException) {
67  Toast.makeText(Register_Page.this , "Email Already in Database", Toast.
    LENGTHLONG).show();
68  }
69  }
70  });
```


6.2 Module 2

Place Recommender Module

By Fetching The Current location the API Services Set of user And the User details of Destination set by API by their Live Tracking of given suggestion of places through Interest Where he Moved And Bookmarks Saved And Giving Suggestion Of Places Of Interest.

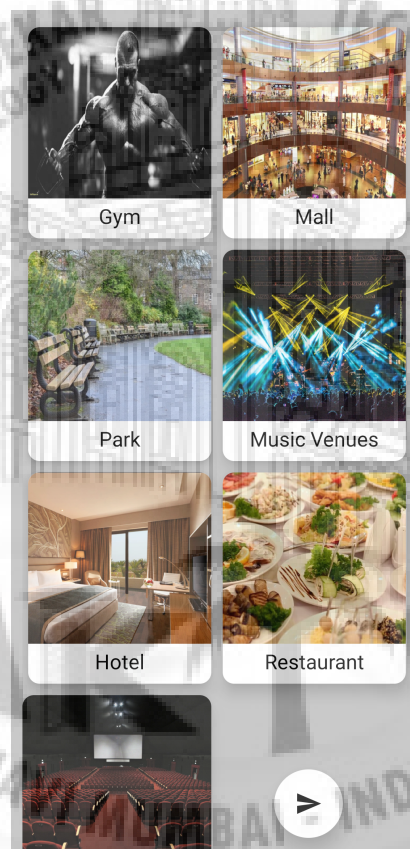


Figure 6.2: LIST OF INTEREST

```

1      private void showNearbyPlaces(List<HashMap<String , String>>
2          nearbyPlaceList)
3      {
4          for(int i = 0; i < nearbyPlaceList.size(); i++)
5          {
6              HashMap<String , String> googlePlace = nearbyPlaceList.get(i);
  
```

```

6      String rating = googlePlace.get("rating");
7      float ret= Float.parseFloat(rating);
8      String user_ratings_total = googlePlace.get("user_ratings_total");
9      if (ret >=3.0){
10         String placeName = googlePlace.get("place_name");
11         String vicinity = googlePlace.get("vicinity");
12         double lat = Double.parseDouble(Objects.requireNonNull(
13             googlePlace.get("lat")));
14         double lng = Double.parseDouble(Objects.requireNonNull(
15             googlePlace.get("lng")));
16         LatLng latLng = new LatLng(lat, lng);
17         mMap.addMarker(new MarkerOptions().position(latLng).title(
18             placeName)
19             .snippet("ADDRESS : " + vicinity + "\nRATING : " +
20                 rating + " (" + user_ratings_total + ")"));
21         Log.i(TAG, "showNearbyPlaces: " + i + nearbyPlaceList);
22     }
23 }
24
25 for(int i = 0; i < nearbyPlaceList.size(); i++)
26 Log.i(TAG, "showNearbyPlaces: List Print"+list_Lat+" "+list_Long);
27 //     home.addList(list_Lat , list_Long);
28 }
29
30 class DownloadURL {
31     String readUrl(String myUrl) throws IOException
32     {
33         String data = "";
34         InputStream inputStream = null;
35         HttpURLConnection urlConnection = null;
36         try {
37             URL url = new URL(myUrl);

```

```
33     urlConnection=(HttpURLConnection) url.openConnection();
34     urlConnection.connect();
35     inputStream = urlConnection.getInputStream();
36     BufferedReader br = new BufferedReader(new InputStreamReader(
37         inputStream));
38     StringBuffer sb = new StringBuffer();
39     String line = "";
40     while((line = br.readLine()) != null)
41     {
42         sb.append(line);
43     }
44     data = sb.toString();
45     br.close();
46 } catch (MalformedURLException e) {
47     e.printStackTrace();
48 } catch (IOException e) {
49     e.printStackTrace();
50 }
51 finally {
52     inputStream.close();
53     urlConnection.disconnect();
54 }
55 Log.d("DownloadURL", "Returning data= "+data);
56 return data;
57 }
58 }
59
60 private void Poi_Status(CardView cardView, ImageView imageView, String
61     viewName) {
62     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
```

```
62     if (cardView . getCardBackgroundColor () == getColorStateList (R. color .
63         colorWhite)) {
64         status . put (viewName , true) ;
65         status_Name . put (viewName , viewName) ;
66         imageView . setAlpha (0.95 f) ;
67         cardView . setCardBackgroundColor (getResources () . getColor (R. color .
68             Yellow)) ;
69         Log . i (TAG, "GymClick: IF RUN" + status . get (viewName)) ;
70     }
71     else {
72         status . put (viewName , false) ;
73         status_Name . remove (viewName) ;
74         imageView . setAlpha (1 f) ;
75         cardView . setCardBackgroundColor (getResources () . getColor (R. color .
76             colorWhite)) ;
77         Log . i (TAG, "GymClick: ELSE RUN") ;
78     }
79 }
```

6.3 Module 3

Updation Of POI Module

After User Updating the Details Of POI the User Can Also **Update,Delete,or Modify** it According The User Wants And First User Pass The Range of User Its Passes It Get Saved In Bookmarks and When User Passes Second Pass Time It Alert Notify That User Pass and it Also Suggests the POI Of New Areas According the Details of User.

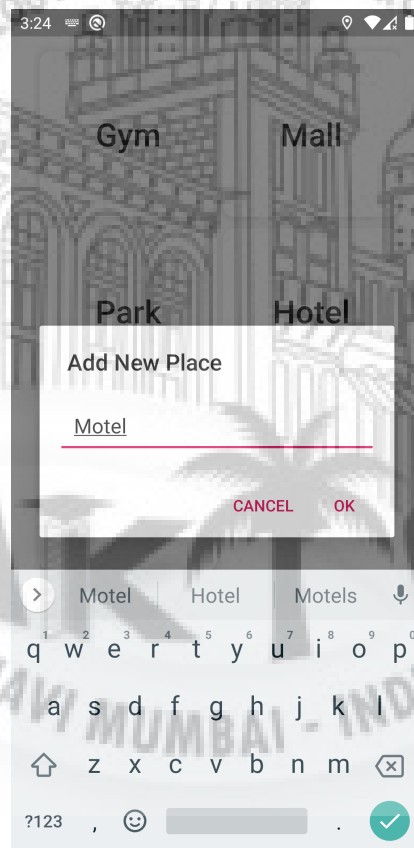


Figure 6.3: UPDATE POI

```

1 private void getPOIList(FirebaseUser user) {
2     if (user != null) {
3
4         mDataRef.child(user.getId()).child("POI List").
  
```

```
addValueEventListener(new ValueEventListener() {
5
    @Override
6
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
7
        long Count = dataSnapshot.getChildrenCount();
8
        User_POI_List = new String[(int) Count];
9
        Log.i(TAG, "onDataChange: Count" + Count);
10
        for (i = 0; i < Count; i++) {
11
            User_POI_List[i] = (String) dataSnapshot.child(String.
                valueOf(i + 1)).getValue();
12
            titleList.add(User_POI_List[i]);
13
            Log.i(TAG, "onDataChange: POI LIST" + User_POI_List.
                length + User_POI_List[i]);
14
        }
15
        poiAdapter = new POIAdapter(titleList);
16
        recyclerView.setAdapter(poiAdapter);
17
    }
18
    @Override
19
    public void onCancelled(@NonNull DatabaseError databaseError) {
20
    }
21
    });
22
    }
23
    }
24
    }
25
    }
26
    }
27
    private void add(View view) {
28
        openDialog();
29
    }
30
    }
31
    public void returnValue(String string) {
32
```

```
33     Log.i(TAG, "returnValue: titleList Size Before " + titleList.size() +
34           Count);
35     int position = titleList.size() + 1;
36     Count = titleList.size();
37     titleList.add(string);
38
39     Log.i(TAG, "returnValue: titleList Size After " + titleList.size() +
40           Count);
41     poiAdapter.notifyItemInserted(position);
42     poiAdapter.notifyDataSetChanged();
43
44 }
45
46 public void deleteItemInfo(HashMap data, int size) {
47     Log.i(TAG, "deleteItemInfo: " + data + " Size " + data.size() + data.get
48         (0) + "\n Fixed Size " + size);
49     int j = 0;
50     SIZE = data.size();
51     positionFinal = new Integer[SIZE];
52
53     for (int i = 0; i < size; i++) {
54         Object value = data.get(i);
55         Object value1 = data.get(size + 1);
56         if (value != value1) {
57             Log.i(TAG, "deleteItemInfo: " + value);
58             positionFinal[j] = value;
59             Log.i(TAG, "deleteItemInfo: " + positionFinal.length);
60             j++;
61         }
62     }
63
64     for (int i = 0; i < positionFinal.length; i++) {
```

```

61         Log.i(TAG, "deleteItemInfo: Final List" + positionFinal[i]);
62     }
63     saveInFirebase(positionFinal);
64 }
65
66 private void saveInFirebase(Object[] positionFinal) {
67     DatabaseReference reference;
68     FirebaseUser firebaseUser;
69     firebaseUser = FirebaseAuth.getInstance().getCurrentUser();
70     reference = FirebaseDatabase.getInstance().getReference("User");
71     List list = new ArrayList();
72     for (int i = 0; i < positionFinal.length; i++) {
73         list.add(positionFinal[i]);
74         Log.i(TAG, "deleteItemInfo: Final List" + positionFinal[i] + " List
75         " + list.get(i));
76     }
77     reference.child(firebaseUser.getId()).child("Delete POI List").child("
78     index").setValue(list);
79 }
80
81 private void delete(View view) {
82     if (mDataRef.child(user.getId()).child("Delete POI List") != null) {
83         mDataRef.child(user.getId()).child("Delete POI List").child("index"
84         ).addListenerForSingleValueEvent(new ValueEventListener() {
85             @Override
86             public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
87                 List<Integer> list = new ArrayList();
88                 long Count = dataSnapshot.getChildrenCount();
89                 Log.i(TAG, "onDataChange: Count" + Count + "TITLE LIST SIZE

```



```

    " + titleList.size());
89     for (int i = 0; i < Count; i++) {
90         int temp = dataSnapshot.child(String.valueOf(i)).
            getValue(Integer.class);
91         Log.i(TAG, "onDataChange: Selected Item" + i + temp);
92         list.add(temp);
93         Log.i(TAG, "onDataChange: Count" + i + Count + "TITLE
            LIST SIZE " + titleList.size());
94         titleList.remove(temp);
95         poiAdapter.notifyItemRemoved(temp);
96     }
97     poiAdapter.notifyDataSetChanged();
98 }
99 @Override
100 public void onCancelled(@NonNull DatabaseError databaseError) {
101 }
102 });
103 }
104 }
105 startActivity(new Intent(this, Home.class));
106 finish();
107 }
```

6.4 Module 4

Fetching Of Details Module

Fetch provides a generic definition of **Request and Response objects**(This will allow them to be used wherever they are needed in the future, whether it's for Any User, Cache API and other similar things that handle or modify requests and responses, or any kind of use case that might require or Review to generate your own responses programmatically. Once a Response is retrieved, there are a number of methods available to user in files.



Figure 6.4: FETCH DETAILS

```

1 Object[] dataTransfer = new Object[2];
2 GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData(

```

```
    getApplicationContext());
3    GetNearbyPlacesDataog getNearbyPlacesDataog = new GetNearbyPlacesDataog(
        getApplicationContext());
4
5    private String getUrl(double latitude, double longitude, String
        nearbyPlace, String type) {
6        StringBuilder googlePlaceUrl = new StringBuilder("https://maps.
            googleapis.com/maps/api/place/nearbysearch/json?");
7        googlePlaceUrl.append("location=" + latitude + "," + longitude);
8        googlePlaceUrl.append("&radius=" + 500);
9        googlePlaceUrl.append("&key=" + apiKey_nearByPlace);
10       Log.i(TAG, "url = " + googlePlaceUrl.toString());
11       return googlePlaceUrl.toString();
12   }
13   dataTransfer[0] = mMap;
14   dataTransfer[1] = url;
15   getNearbyPlacesData.execute(dataTransfer);
16   public GetNearbyPlacesData(Context context) {
17       this.context = context;
18   }
19
20   class DataParser {
21       private HashMap<String, String> getPlace(JSONObject googlePlaceJson)
22       {
23           HashMap<String, String> googlePlaceMap = new HashMap<>();
24           String placeName = "";
25           String vicinity = "";
26           String latitude = "";
27           String longitude = "";
28           String reference = "";
29           String rating = "";
```

```
30     String user_ratings_total="";
31     Log.d("DataParser", "jsonobject =" + googlePlaceJson.toString());
32     try {
33         if (!googlePlaceJson.isNull("name")) {
34             placeName = googlePlaceJson.getString("name");
35             rating = googlePlaceJson.getString("rating");
36             user_ratings_total = googlePlaceJson.getString("
37                 user_ratings_total");
38         }
39         if (!googlePlaceJson.isNull("vicinity")) {
40             vicinity = googlePlaceJson.getString("vicinity");
41         }
42         latitude = googlePlaceJson.getJSONObject("geometry").getJSONObject("
43             location").getString("lat");
44         longitude = googlePlaceJson.getJSONObject("geometry").getJSONObject(
45             "location").getString("lng");
46         reference = googlePlaceJson.getString("reference");
47         googlePlaceMap.put("place_name", placeName);
48         googlePlaceMap.put("vicinity", vicinity);
49         googlePlaceMap.put("lat", latitude);
50         googlePlaceMap.put("lng", longitude);
51         googlePlaceMap.put("reference", reference);
52         googlePlaceMap.put("rating", rating);
53         googlePlaceMap.put("user_ratings_total", user_ratings_total);
54     }
55     catch (JSONException e) {
56         e.printStackTrace();
57     }
58     return googlePlaceMap;
59 }
60 private List<HashMap<String, String>>getPlaces(JSONArray jsonArray)
```

```
58     {
59         int count = jsonArray.length();
60         List<HashMap<String , String>> placelist = new ArrayList<>();
61         HashMap<String , String> placeMap = null;
62
63         for(int i = 0; i<count;i++)
64         {
65             try {
66                 placeMap = getPlace((JSONObject) jsonArray.get(i));
67                 placelist.add(placeMap);
68             } catch (JSONException e) {
69                 e.printStackTrace();
70             }
71         }
72         return placelist;
73     }
74     List<HashMap<String , String>> parse(String jsonData)
75     {
76         JSONArray jsonArray = null;
77         JSONObject jsonObject;
78         Log.d("json data", jsonData);
79         try {
80             jsonObject = new JSONObject(jsonData);
81             jsonArray = jsonObject.getJSONArray("results");
82         } catch (JSONException e) {
83             e.printStackTrace();
84         }
85         return getPlaces (jsonArray);
86     }
87 }
88 @Override
```

```
89     protected String doInBackground(Object... objects){
90         mMap = (GoogleMap) objects [0];
91         url = (String) objects [1];
92         DownloadURL downloadURL = new DownloadURL ();
93         try {
94             googlePlacesData = downloadURL.readUrl(url);
95         } catch (IOException e) {
96             e.printStackTrace();
97         }
98         return googlePlacesData;
99     }
100
101     @Override
102     protected void onPostExecute(String s){
103         List<HashMap<String, String>> nearbyPlaceList;
104         DataParser parser = new DataParser();
105         nearbyPlaceList = parser.parse(s);
106         Log.d("nearbyplacesdata", "called parse method");
107         showNearbyPlaces(nearbyPlaceList);
108     }
109
110     private void showNearbyPlaces(List<HashMap<String, String>> nearbyPlaceList)
111     {
112         for(int i = 0; i < nearbyPlaceList.size(); i++)
113         {
114             HashMap<String, String> googlePlace = nearbyPlaceList.get(i);
115
116             String rating = googlePlace.get("rating");
117             float ret= Float.parseFloat(rating);
118             String user_ratings_total = googlePlace.get("user_ratings_total");
119             if (ret >=3.0){
```

```
120     String placeName = googlePlace.get("place_name");
121     String vicinity = googlePlace.get("vicinity");
122     double lat = Double.parseDouble(Objects.requireNonNull(googlePlace.
123         get("lat")));
124     double lng = Double.parseDouble(Objects.requireNonNull(googlePlace.
125         get("lng")));
126     latLng = new LatLng(lat, lng);
127     mMap.addMarker(new MarkerOptions().position(latLng).title(placeName)
128         .snippet("ADDRESS : " + vicinity + "\nRATING : " + rating + " ("
129             + user_ratings_total + ")"));
130     Log.i(TAG, "showNearbyPlaces: " + i + nearbyPlaceList);
131 }
132 }
133
134 for(int i = 0; i < nearbyPlaceList.size(); i++)
135     Log.i(TAG, "showNearbyPlaces: List Print"+list_Lat+" "+list_Long);
136 }
137
138 class DownloadURL {
139     String readUrl(String myUrl) throws IOException
140     {
141         String data = "";
142         InputStream inputStream = null;
143         HttpURLConnection urlConnection = null;
144         try {
145             URL url = new URL(myUrl);
146             urlConnection=(HttpURLConnection) url.openConnection();
147             urlConnection.connect();
148             inputStream = urlConnection.getInputStream();
149             BufferedReader br = new BufferedReader(new InputStreamReader(
150                 inputStream));
151             StringBuffer sb = new StringBuffer();
```

```
147     String line = "";  
148     while((line = br.readLine()) != null)  
149     {  
150         sb.append(line);  
151     }  
152     data = sb.toString();  
153     br.close();  
154 } catch (MalformedURLException e) {  
155     e.printStackTrace();  
156 } catch (IOException e) {  
157     e.printStackTrace();  
158 }  
159 finally {  
160     inputStream.close();  
161     urlConnection.disconnect();  
162 }  
163 Log.d("DownloadURL", "Returning data= "+data);  
164 return data;  
165 }
```


6.5 Module 5

Capturing POI Module

After Entering The Details Of POI the **Google Play Services** Start Capturing the Details of User By Fetching The Current location the API Services Set of user And the User details of Destination set by API by Live Tracking Where he Moved And Bookmarks Saved And Giving Suggestion Of Places Of Interest According The Details The User Given .

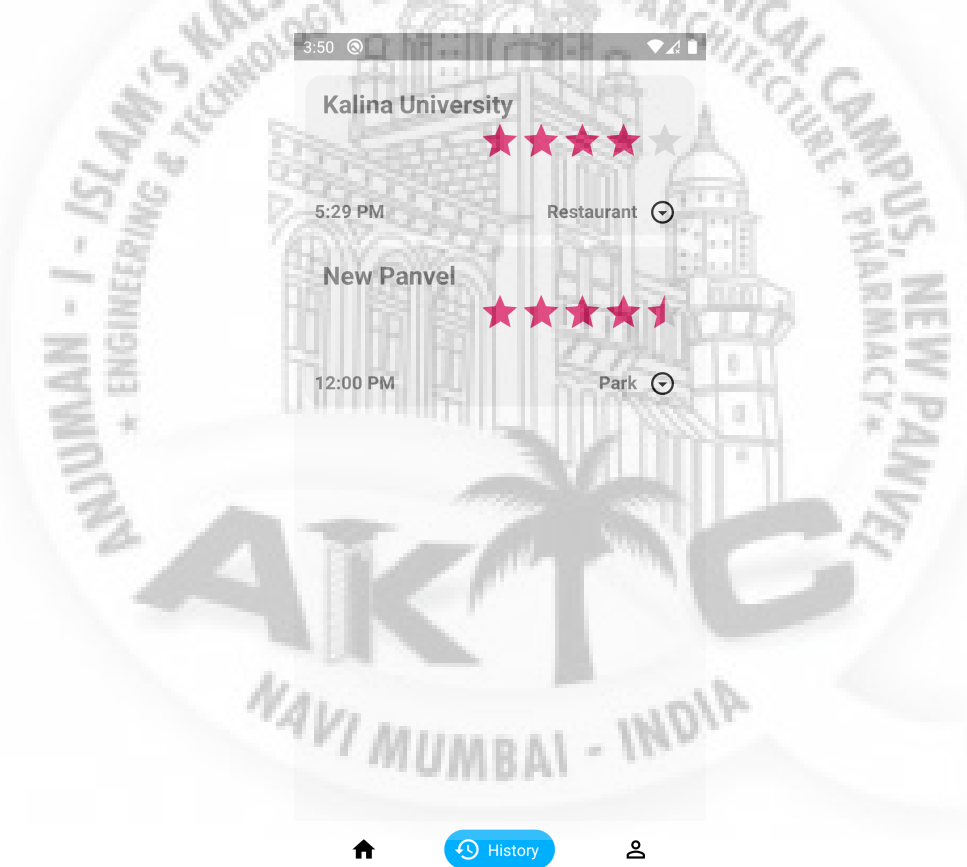


Figure 6.5: CAPTURE POI LIST

1
2

```
AutocompleteSupportFragment autocompleteSupportFragment = (
    AutocompleteSupportFragment) getSupportFragmentManager().
    findFragmentById(R.id.place_autocomplete_fragment1);
```

```
3      autocompleteSupportFragment . setActivityMode ( AutocompleteActivityMode .
4          OVERLAY );
5      autocompleteSupportFragment . setPlaceFields ( Arrays . asList ( Place . Field . ID ,
6          Place . Field . NAME , Place . Field . LAT_LNG , Place . Field . ADDRESS ,
7          Place . Field . PHONE_NUMBER , Place . Field . WEBSITE_URI , Place . Field .
8          RATING , Place . Field . USER_RATINGS . TOTAL ) );
9      autocompleteSupportFragment . setHint ( " Search Here ... " );
10     autocompleteSupportFragment . setOnPlaceSelectedListener ( new
11         PlaceSelectionListener () {
12             @ SuppressLint ( " SetTextI18n " )
13             @ Override
14             public void onPlaceSelected ( @NonNull Place place ) {
15                 String text = autocompleteSupportFragment . getText (
16                     autocompleteSupportFragment . getId () ) . toString () ;
17                 Toast . makeText ( New_Place . this , text , LENGTH_LONG ) . show () ;
18                 Log . i ( TAG , " PlaceText : " + text ) ;
19                 Toast . makeText ( New_Place . this , " Place : " + place . getName () + " , " +
20                     place . getId () , LENGTH_LONG ) . show () ;
21                 if ( place . getLatLng () != null ) {
22                     Log . i ( TAG , " Place : " + place . getName () ) ;
23                     placeName = place . getName () ;
24                     placeID = place . getId () ;
25                     Destination . setText ( " Destination : " + placeName ) ;
26                     Destination . setVisibility ( View . VISIBLE ) ;
27                     Source . setVisibility ( View . VISIBLE ) ;
28                 }
29                 Log . i ( TAG , " PlaceText : " + text ) ;
30             }
31         }
```

Chapter 7

System Testing

We,as testers are aware of the various types of software testing such as functional testing,non-functional testing ,automation testing,agile testing and their sub-type,etc. Each of us would have come across several types of testing in our testing journey.Each type of testing has its own features,advantages,and disadvantages as well.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Registration	Register login	Log in successfully	Login to app
T02	Updation	Update	update successfully	show update
T03	Details	add details	successfully add	change details

7.2 Sample of a Test Case

Title: Login Page – Authenticate Successfully on gmail.com

Description: A registered user should be able to successfully login at gmail.com.

Precondition: the user must already be registered with an email address and password.

Assumption: a supported browser is being used.

Test Steps:

1. Navigate to gmail.com
2. In the 'email' field, enter the email of the registered user.
3. Click the 'Next' button.
4. Enter the password of the registered user
5. Click 'Sign In'

Expected Result: A page displaying the gmail user's inbox should load, showing any new message at the top of the page.

Actual Result:

When we enter the email id and password in sign in column, it gives the result that what we expected.



Chapter 8

Screenshots of Project

← BACK

Welcome To Register

Enter Full Name

Enter Email

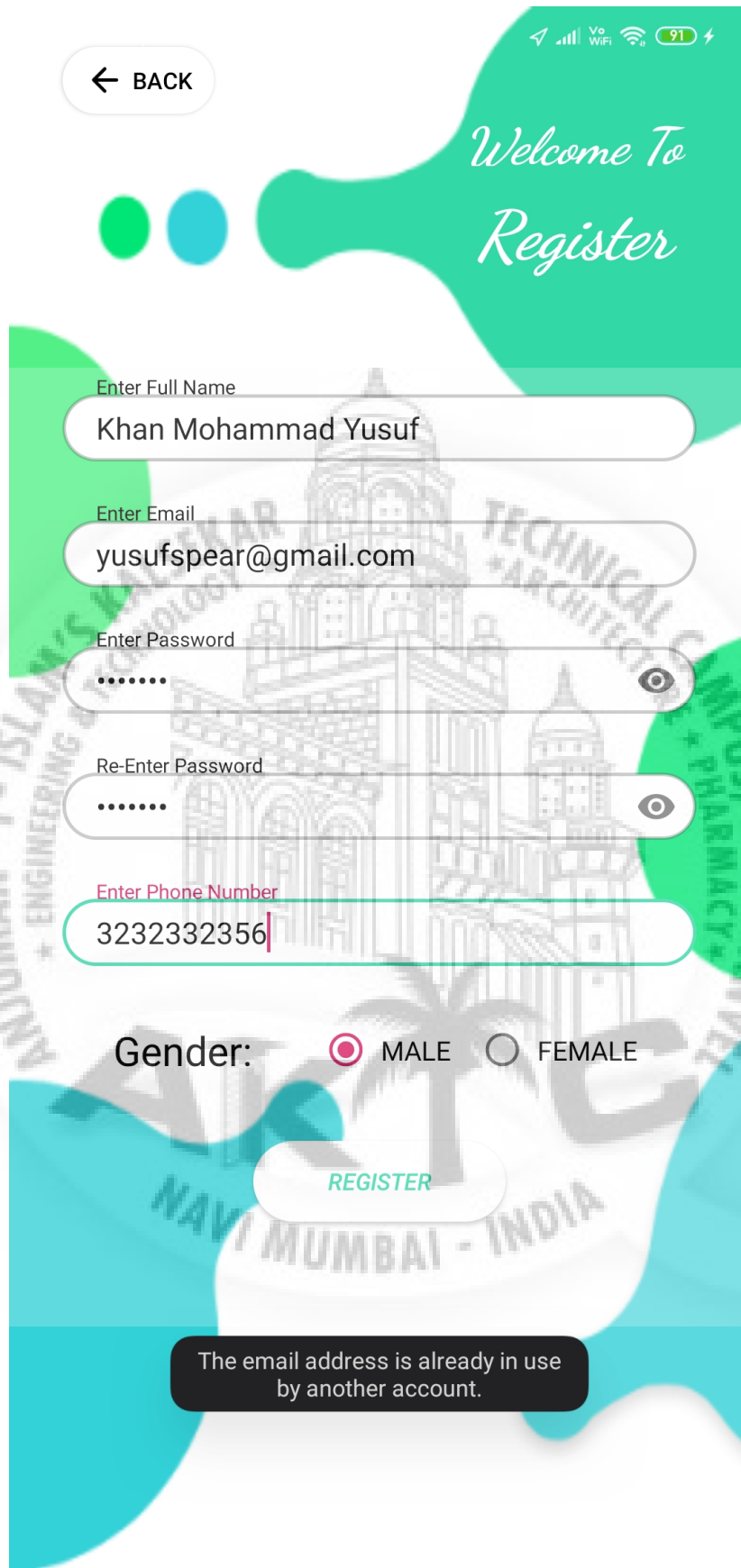
Enter Password

Re-Enter Password

Enter Phone Number

Gender: MALE FEMALE

REGISTER



← BACK

VoWiFi 91

Welcome To Register

Enter Full Name
Khan Mohammad Yusuf

Invalid Email. Enter valid email address.
yusufspear@gmail

Enter Password
.....

Re-Enter Password
.....

Enter Phone Number
3232332356

Gender: MALE FEMALE

REGISTER

← BACK

91

VoWiFi

Welcome To Register

Enter Full Name
Mohammed Yusuf Khan

Enter Email
uzairspear@gmail.com

Enter Password
.....

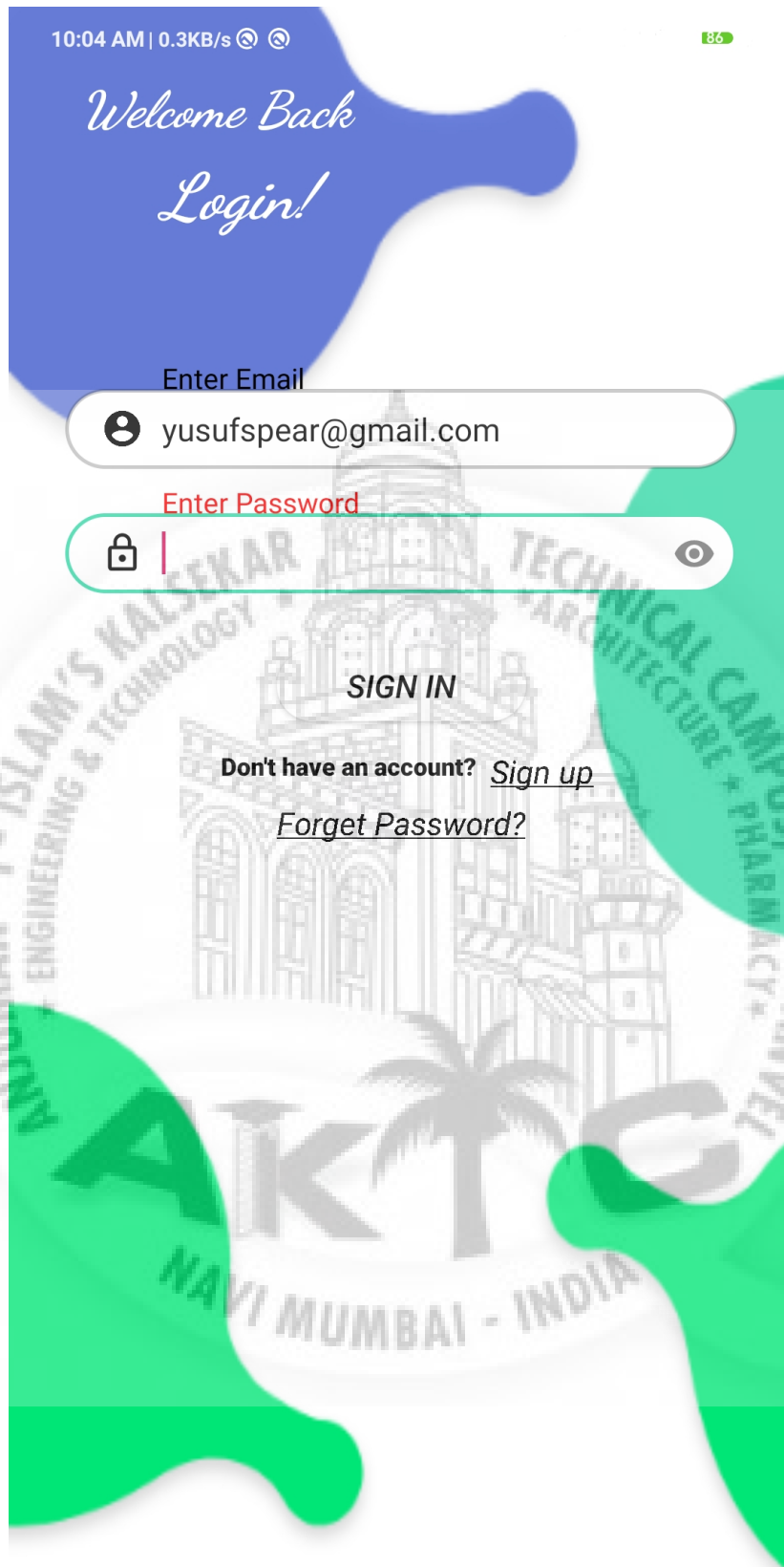
Re-Enter Password
.....

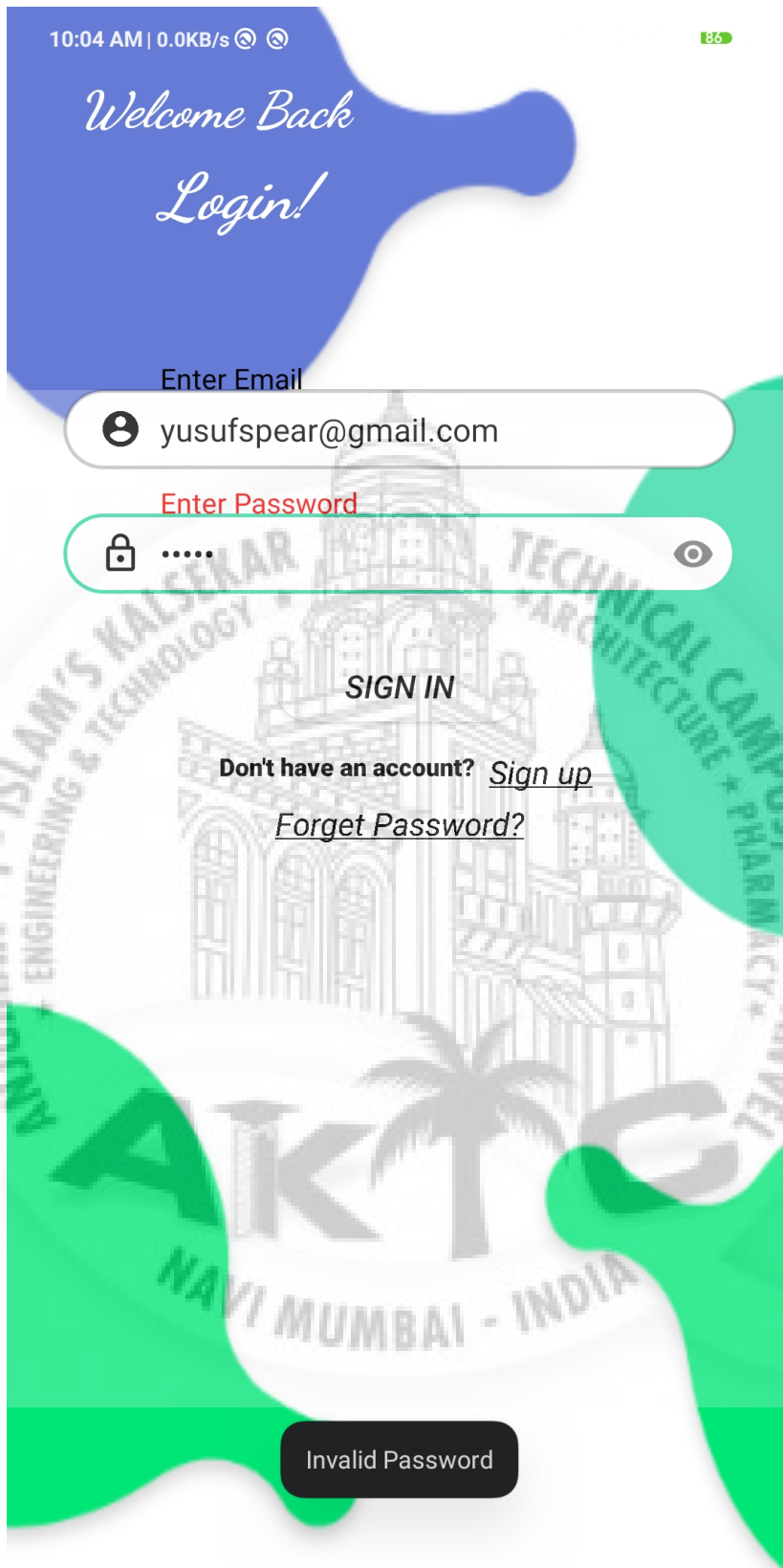
Enter Phone Number
6985253780

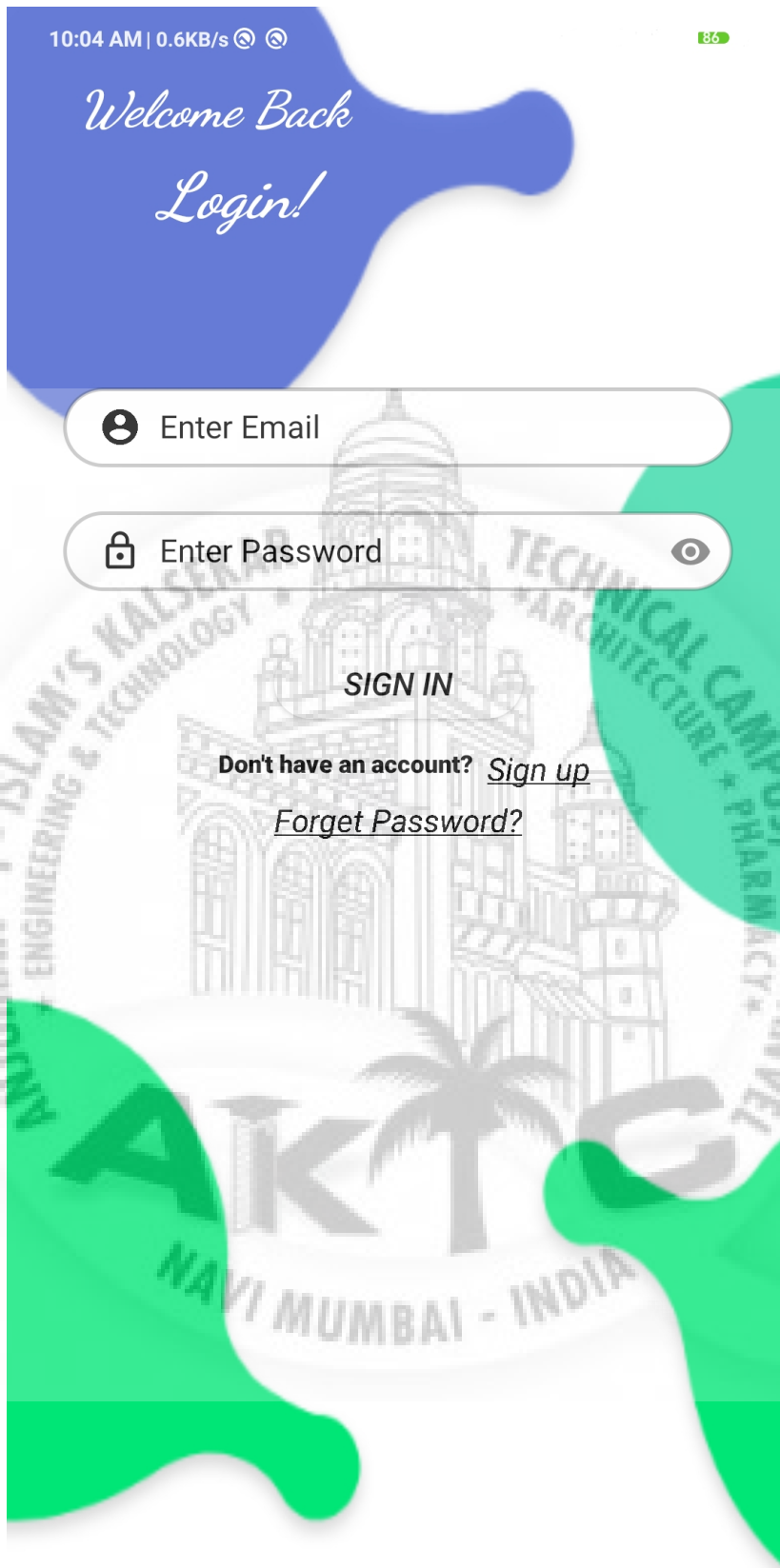
Gender: MALE FEMALE

REGISTER

Password NOT Match





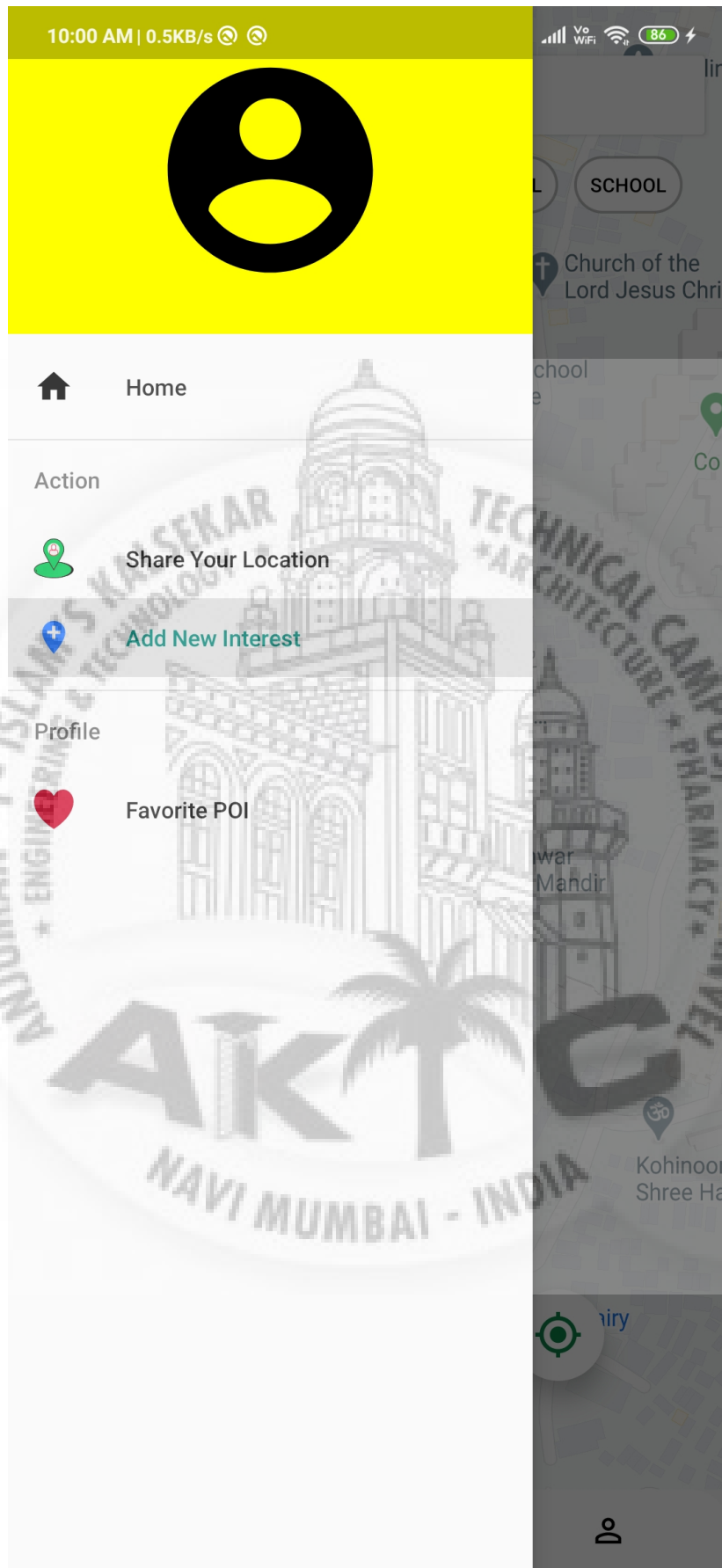


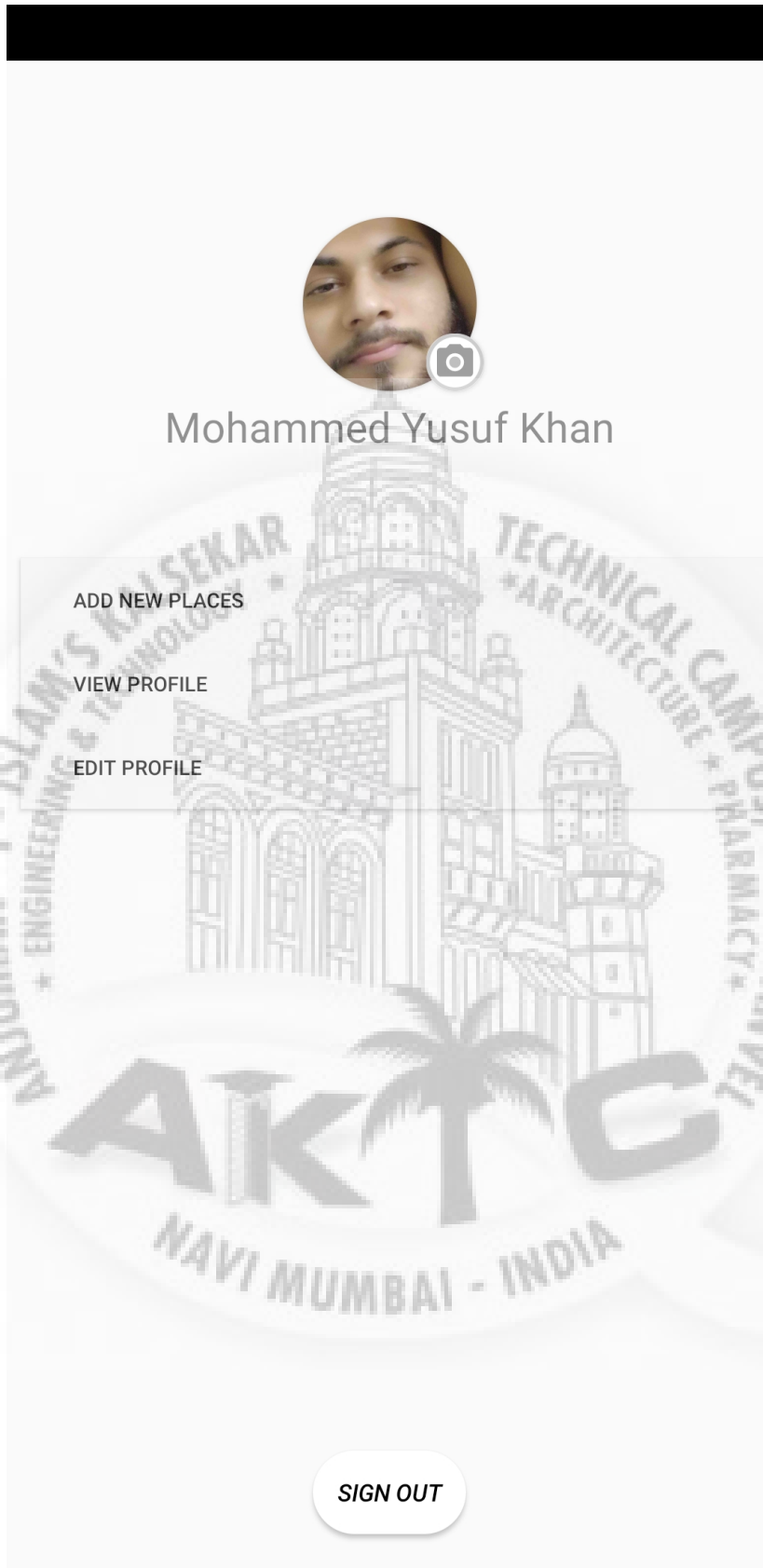
Forget Password

 Enter Email

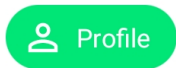
RESET PASSWORD

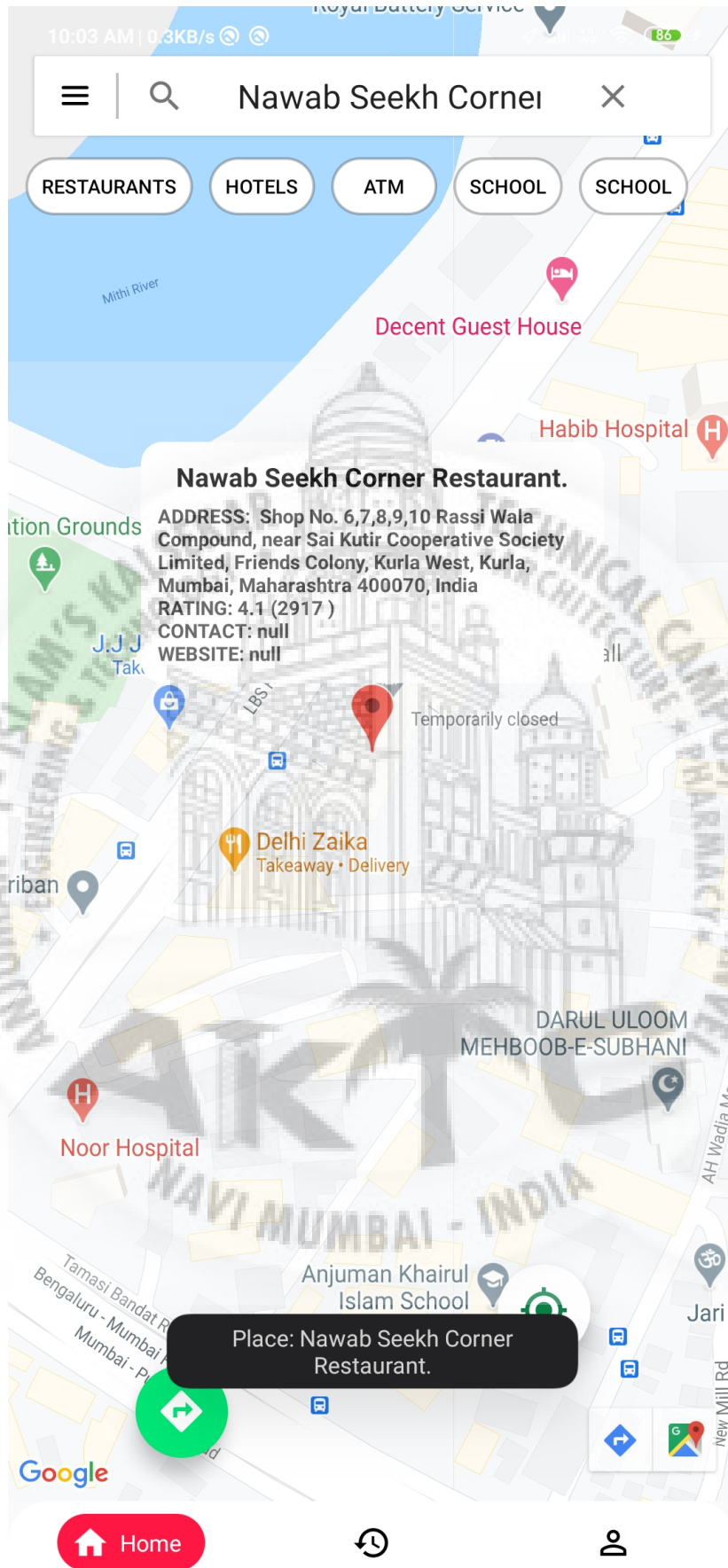


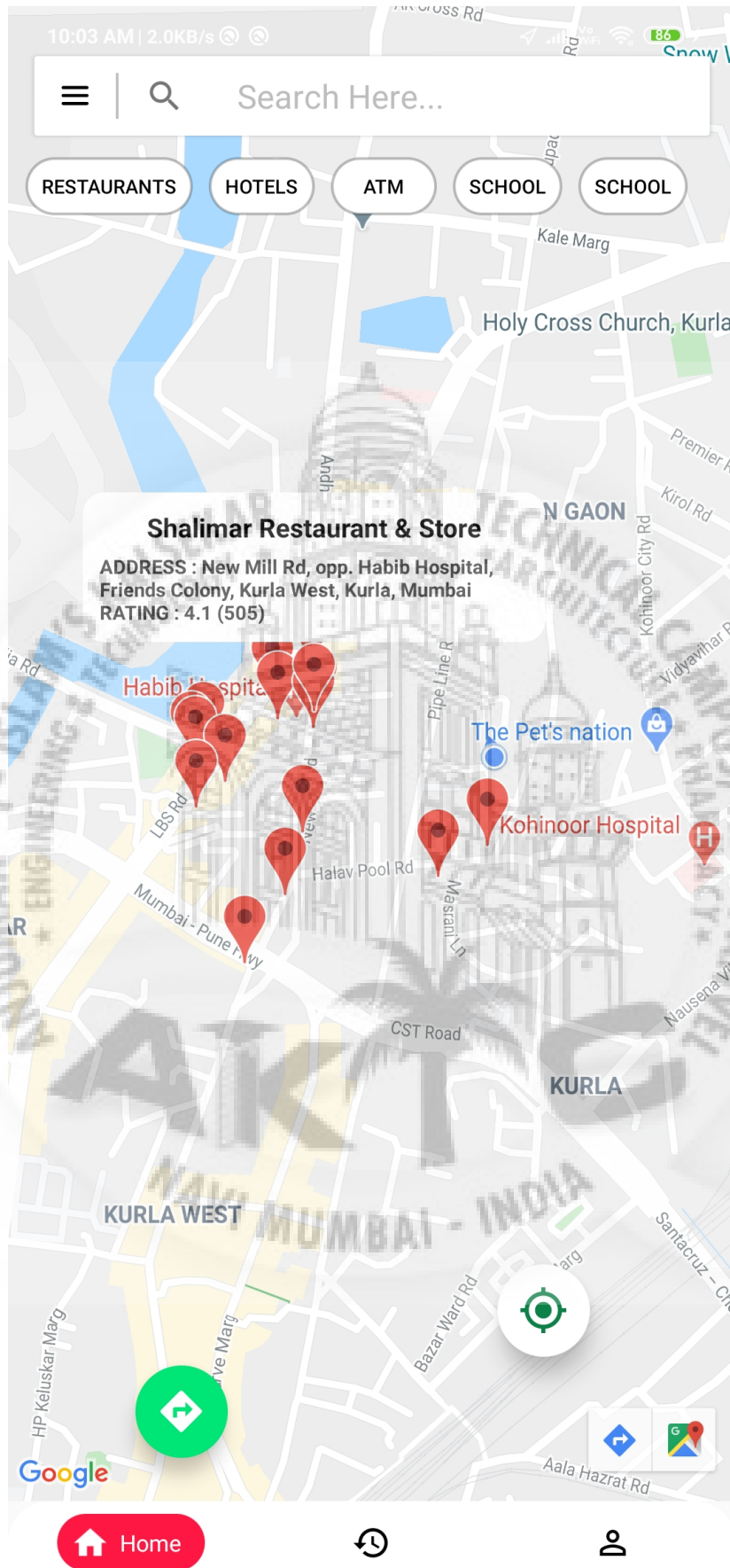




The image shows a user profile interface. At the top, there is a circular profile picture of a man with a beard and a camera icon next to it. Below the picture is the name "Mohammed Yusuf Khan". A semi-transparent menu is overlaid on the profile, containing three options: "ADD NEW PLACES", "VIEW PROFILE", and "EDIT PROFILE". At the bottom of the profile area is a "SIGN OUT" button. The background of the profile area features a large, faded watermark of the AIKTC logo, which includes a building illustration and the text "ANJUMAN - I - ISLAM'S KATSEKAR + ENGINEERING & TECHNOLOGY", "TECHNICAL CAMPUS + ARCHITECTURE + PHARMACY + NEW PANVEL", "AIKTC", and "NAVI MUMBAI - INDIA".







10:00 AM | 0.0KB/s

VoWiFi 86

Enter Your Destination

Source: Current Location

Destination: Kalina Mumbai University

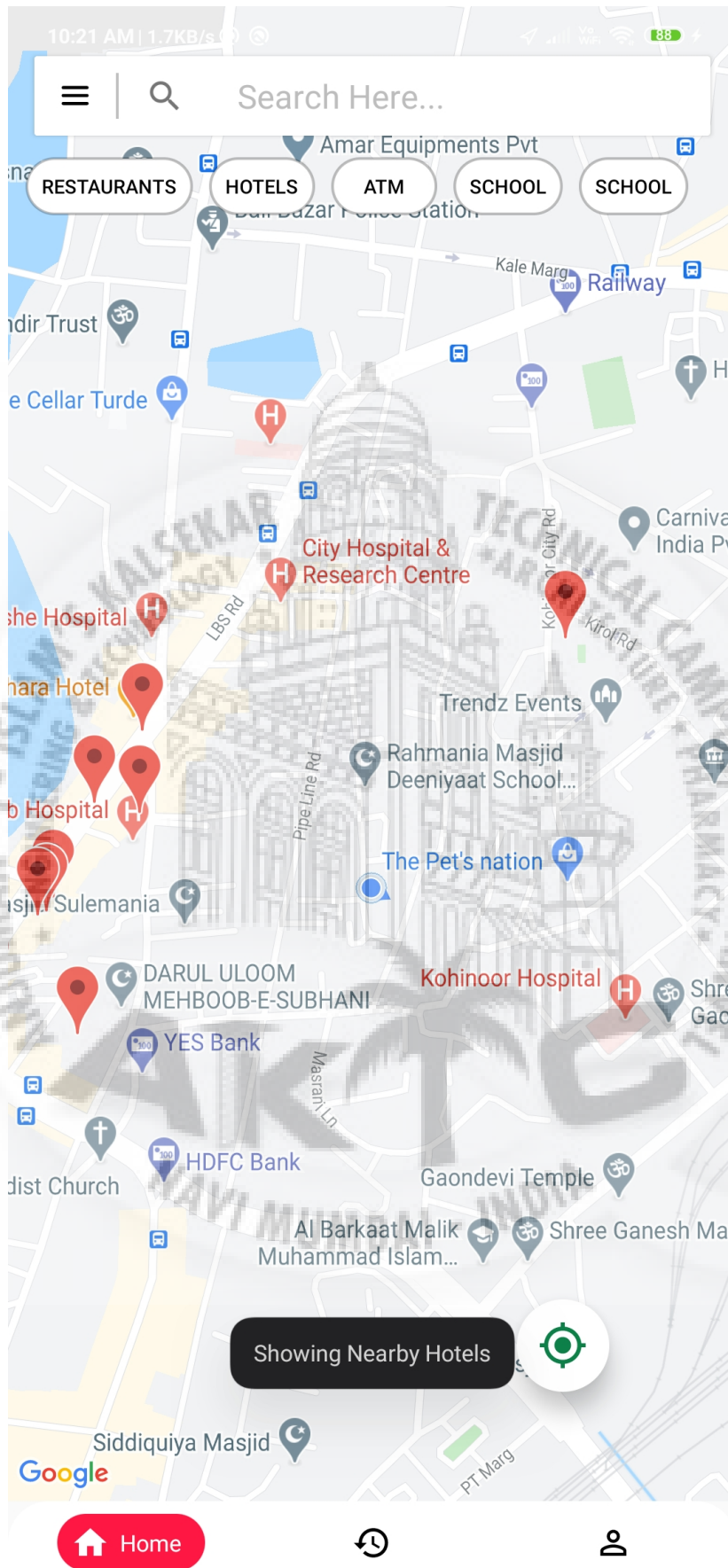
Select Place Of Interest

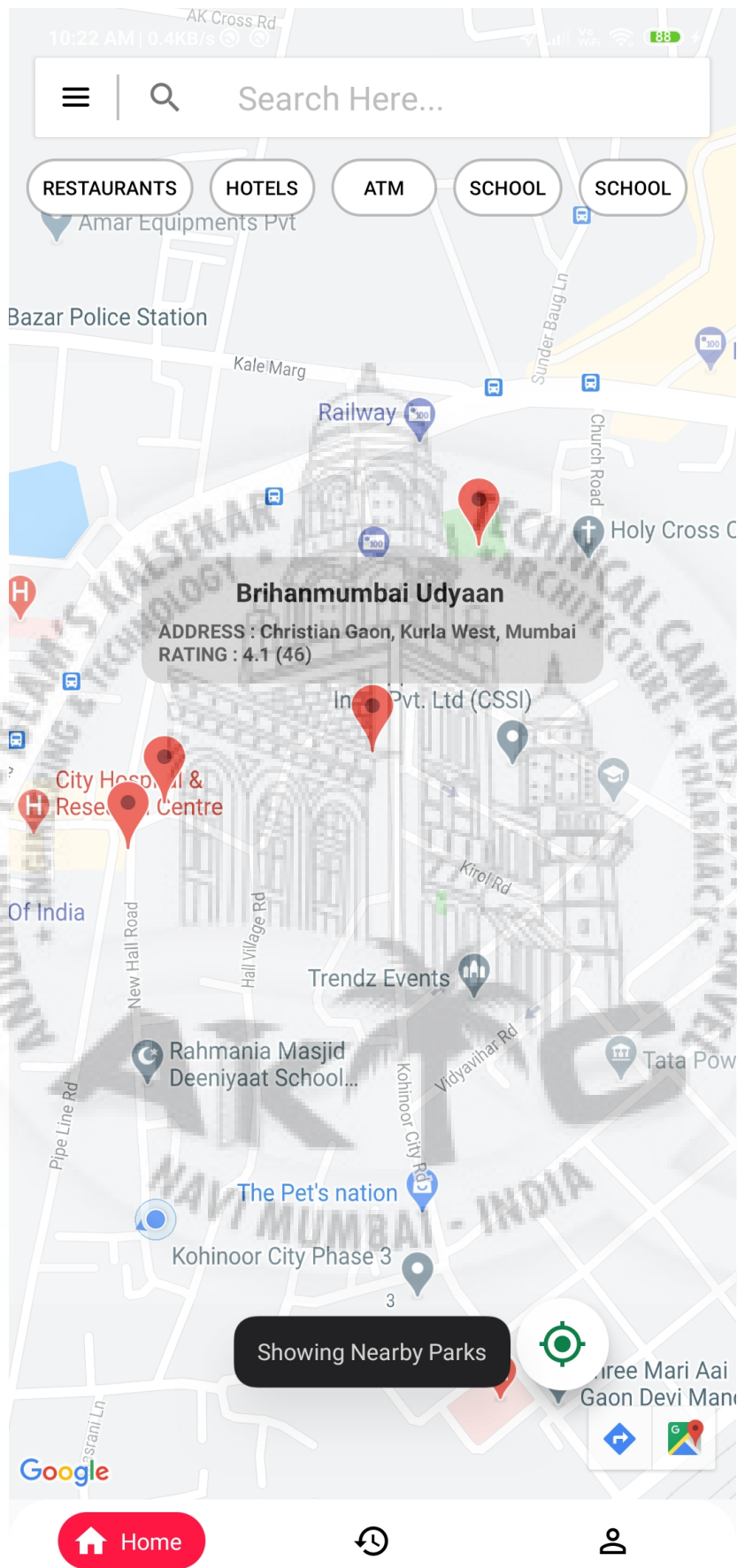
- Gym
- Park
- Restaurant
- Hotel
- Mall
- [Unlabeled]

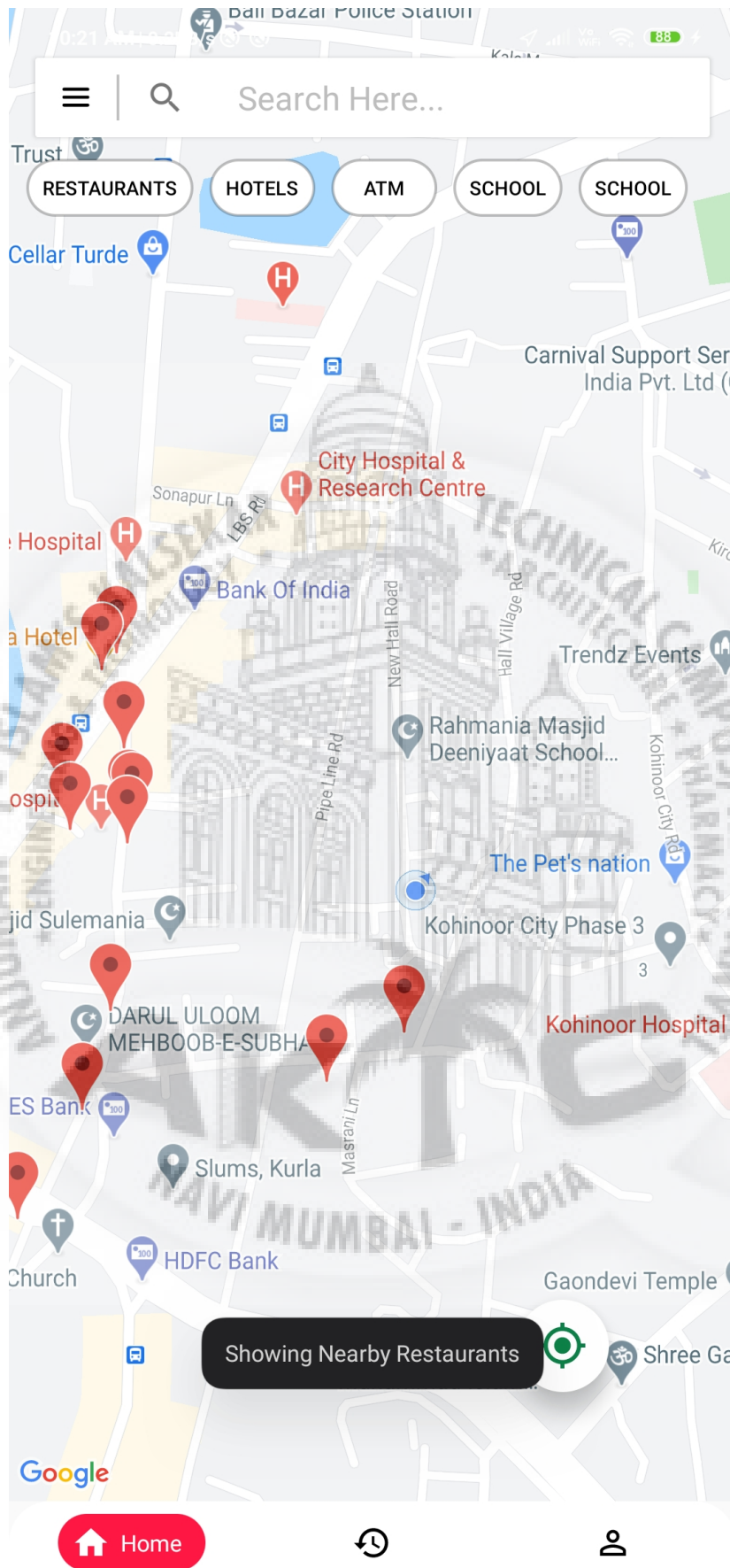
Place: Kalina Mumbai University
ChIJ5x8g7_DI5zsRBWSyvHsCvls

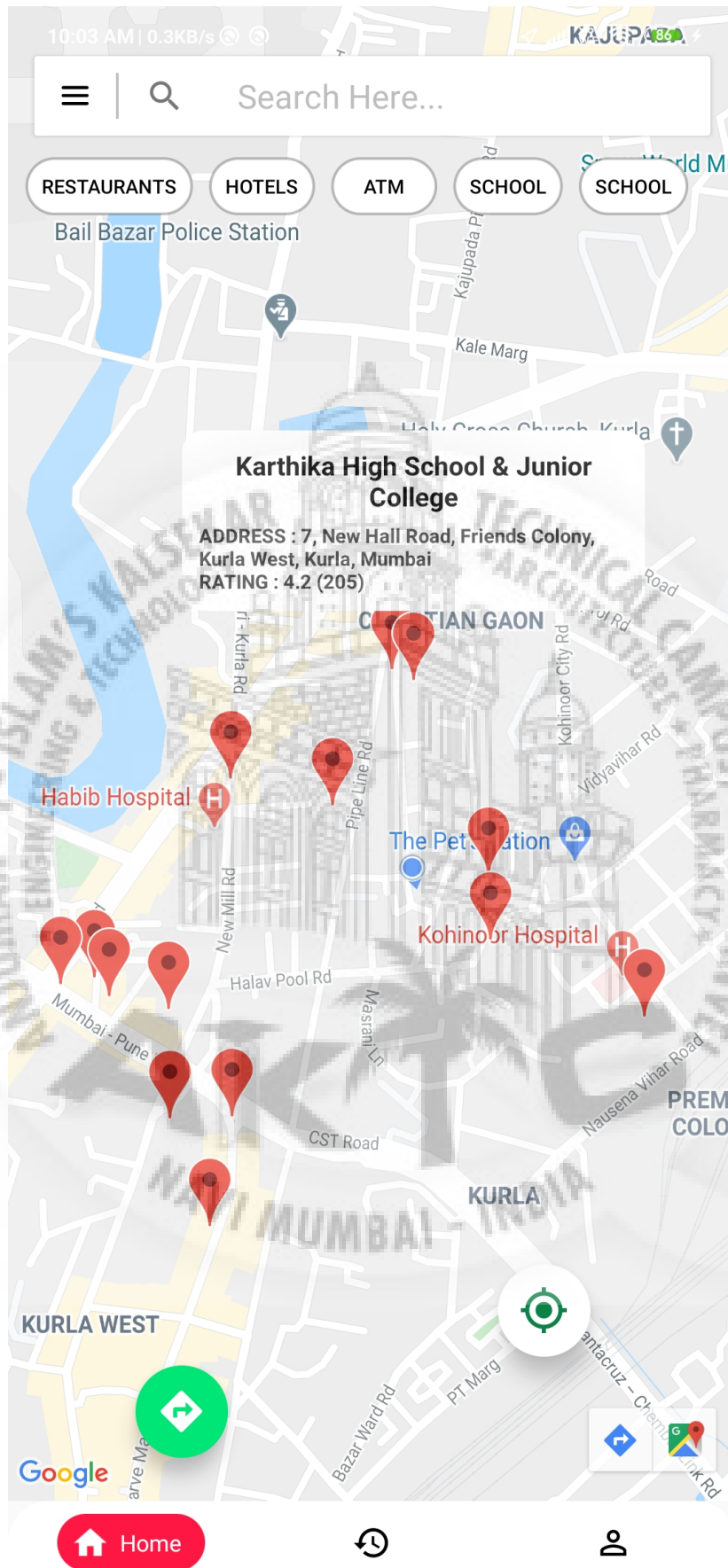


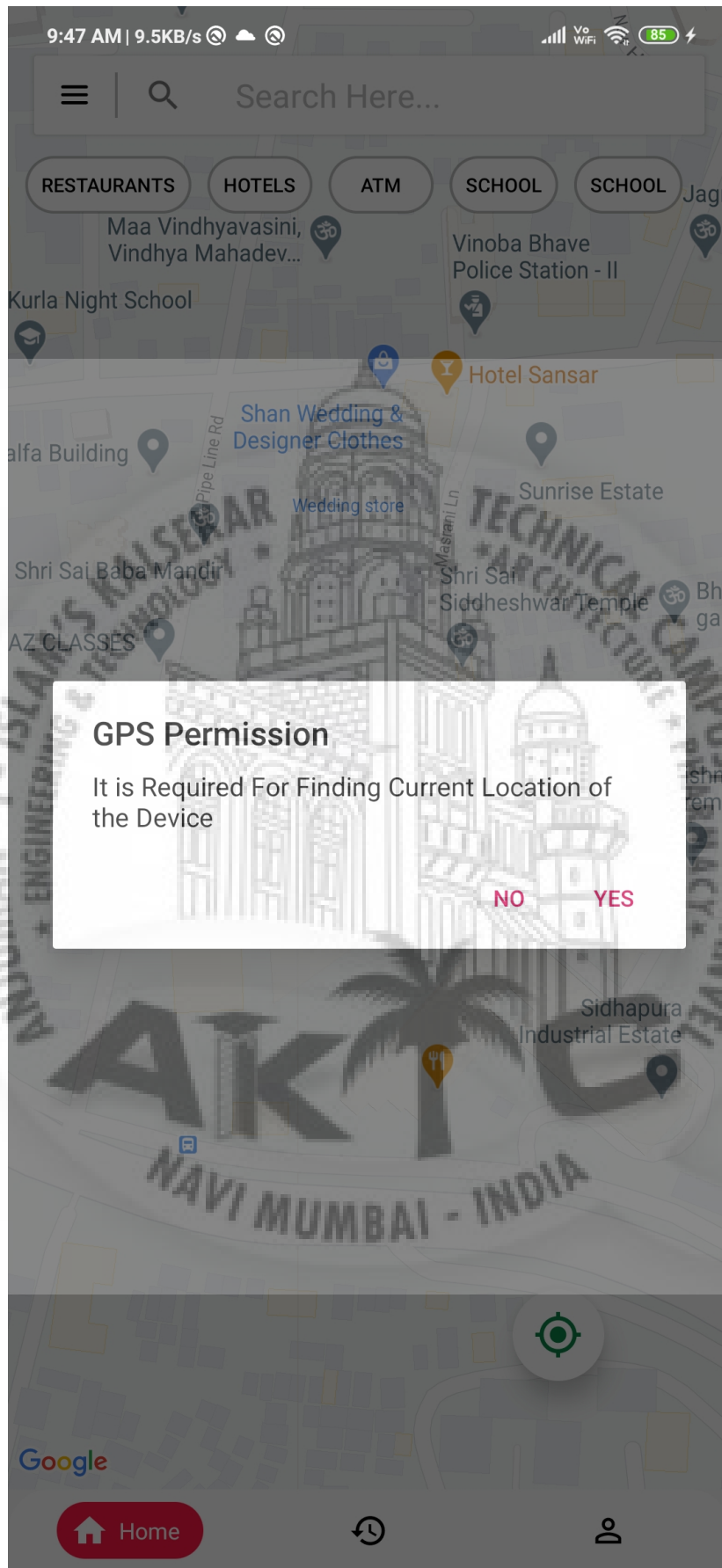


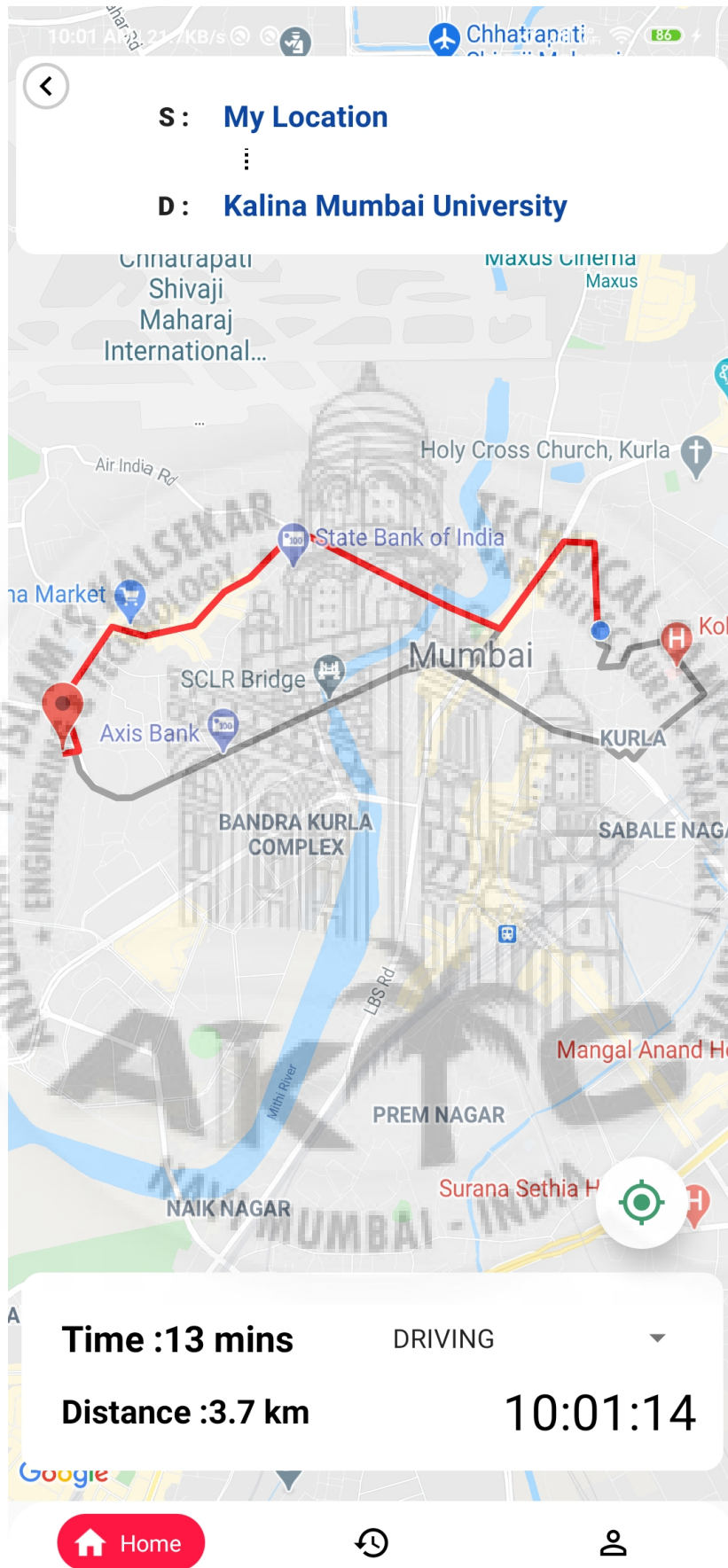














Chapter 9

Conclusion and Future Scope

9.1 Conclusion

The Location Based Places Of Interest is to record the details various activities of user. It will simplify the task of user and paper work.

The causes of POI App are huge, it will increase the user t heir need of the hour is to have such platforms that track & automate their daily routine activity.

Location based Services can also be implemented on Android based smart phones to provide these value-added services: providing routing

information, helping them find nearby hotels, Malls, Park, Hospitals.

9.2 Future Scope

Location Based Places Of Interest provides a lot of facility to their user. The scope of Location Based Places Of Interest to record the details various activities of user. It will simplify the task of user and paper work. During implementation every user will be given appropriate training to suit their specific needs. Specific support will also be provided at key points within the academic calendar. Training will be provided on a timely basis, and you will be trained as the new is Seating Arrangement System rolled out to your area of responsibility.

References

Location Based Reminder Android Application Using Google Maps API, “Pradnya Battin Dr. S.D.Markande” *International Conference on Automatic Control and Dynamic Optimization Techniques (ICAC-DOT 2016)*]

Personalized Location Recommendation for Location-Based Social Networks, “Qianfang Xu, Jiachun Wang and Bo Xiao”/CIC *International Conference on Communications in China*][ICCC 2017 IEEE]

Enhanced Sentiment Classification Using Geo Location Tweets, “Miss. Shital Anil Phand Mrs. V. A. Chakkarwar” [*Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018)*]

The Tripify Apps for POI) *tripify.com*; Tyler C. Dalton [https://www.](https://www.tripify.com/contributor/christina-tarantola/2017/12/the-tripify)

[tripify.com/contributor/christina-tarantola/2017/12/the-tripify](https://www.tripify.com/contributor/christina-tarantola/2017/12/the-tripify)

October, 2019.

The Sygenic Traveler App *sygictrip.com*; [https://www.sygenic.](https://www.sygenic.com/search?q=SygicTravelTriplanner&safe=active&sxsrf)

[com/search?q=SygicTravelTriplanner&safe=active&sxsrf](https://www.sygenic.com/search?q=SygicTravelTriplanner&safe=active&sxsrf), October,

2019

FireBase Extension *android police.com*; [https://www.androidpolice.](https://www.androidpolice.com/2019/09/27/google-firebase-extensions-app-distribution-ios)

[com/2019/09/27/google-firebase-extensions-app-distribution-ios-](https://www.androidpolice.com/2019/09/27/google-firebase-extensions-app-distribution-ios)

2019



Achievements

1. Publications

- (a) *LOCATION BASED PLACES OF INTEREST*; Prof.SAYED AMER HASHMI, ASHFAQUE AHMAD, KHAN MOHD.YUSUF, KHAN MOHAMMED, "International Journal of Innovative Science and Research Technology(IJISRT)" ,March-2020(<https://ijisrt.com/location-based-places-of-interest>)



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Location Based Places of Interest

AUTHORED BY
Ashfaque Ahmed

HAS BEEN PUBLISHED IN
Volume 5 | Issue 3 | March - 2020

ARTICLE DIGITAL NO.
IJISRT20MAR234



EDITOR IN CHIEF IJISRT

A DIGITAL LIBRARY

WWW.IJISRT.COM

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Location Based Places of Interest

AUTHORED BY
Khan Mohammed

HAS BEEN PUBLISHED IN
Volume 5 | Issue 3 | March - 2020

ARTICLE DIGITAL NO.
IJISRT20MAR234



EDITOR IN CHIEF IJISRT

A DIGITAL LIBRARY

WWW.IJISRT.COM

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Location Based Places of Interest

AUTHORED BY
Khan Mohd Yusuf

HAS BEEN PUBLISHED IN
Volume 5 | Issue 3 | March - 2020

ARTICLE DIGITAL NO.
IJISRT20MAR234



EDITOR IN CHIEF IJISRT

A DIGITAL LIBRARY

WWW.IJISRT.COM

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com