A PROJECT REPORT

ON

"PRIORITY BASED CAB SEARCH ENGINE"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

MORBIWALA QUID ZOHAR NAEEEM YASMIN	14CO40
SHAIKH SHANAWAZ SALIM FARZANA	14CO49
PIPEWALA TAHA MOHAMMED TASNEEM	14CO38

UNDER THE GUIDANCE OF
PROF. JAVED KHAN SHEIKH

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
2017-2018

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"PRIORITY BASED CAB SEARCH ENGINE"

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER ENGINEERING

BY

MORBIWALA QUID ZOHAR NAEEEM YASMIN	14CO40
SHAIKH SHANAWAZ SALIM FARZANA	14CO49
PIPEWALA TAHA MOHAMMED TASNEEM	14CO38

UNDER THE GUIDANCE OF
PROF. JAVED KHAN SHEIKH

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

2017-2018
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

"PRIORITY BASED CAB SEARCH ENGINE"

submitted by

MORBIWALA QUID ZOHAR NAEEM YASMIN	**14CO40**
SHAIKH SHANAWAZ SALIM FARZANA	**14CO49**
PIPEWALA TAHA MOHAMMED TASNEEM	**14CO38**

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 2017-2018, under our guidance.

Date: / /

Prof. JAVED KHAN SHEIKH
Project Supervisor

Prof. KALPANA BODKE
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 our sincere thanks to our guide **JAVED KHAN SHEIKH**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout our project research work. Without his kind guidance & support this was not possible.

We are grateful to him/her for his timely feedback which helped us track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help us to complete our 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. KALPANA BODKE**, 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 us directly or indirectly during this course of work.

MORBIWALA QUID ZOHAR NAEEM YASMIN

SHAIKH SHANAWAZ SALIM FARZANA

PIPEWALA TAHA MOHAMMED TASNEEM

Project II Approval for Bachelor of Engineering

This project entitled *P̈riority based cab search engine¨* by ***Quid Zohar Morbiwala,Shanawaz Shaikh,Taha Pipewala*** 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 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.

Morbiwala Quid Zohar Naeem Yasmin
14CO40

Shaikh Shanawaz Salim Farzana
14CO49

Pipewala Taha Mohammed Tasneem
14CO38

ABSTRACT

In last two years, the rapid development of Internet based ride-sharing has brought great changes to travel pattern of residents. By comparing the trip records of OLA and UBER there is a big market in this industry of internet ride sharing.The technologies that are used for the implementation are firebase database,REST API and Android Studio.The security login authentication to the system is firebase Google authentication This paper shows the implementation of comparison of the price, distance and time by car selection of OLA and UBER .

Keywords: Firebase-Database,REST-API, Comparison,GPS

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the rapid popularization and development of the Internet/Web technology,all kinds of information of resources can be easily obtained from the Internet. People are enjoying the benefits of the Internet, and the Internet is gradually becoming a part of work and life.Opening multiple cab apps to compare ETA and pricing, and then booking a cab consumes time, especially when you are at a place with weak internet GPS signals. With our Cab Bookings, we allow users to book their nearest cab in just several seconds by tapping and holding a button on their home screen. Sometimes people want to travel in cheap rates and sometimes in less amount of time so by considering this two factor as a priority we are designing this project.In this busy world, time and money are two important factor by booking OLA or UBER individually we don't know which cab is nearer to our source location and provide us cheap rate of the journey.With our Cab Bookings, we allow users to book their nearest cab in just several seconds . So by integrating these two app, we can provide this factor in a single app and user can decide its priority and book the cab.

1.1 Purpose

The establishment and improvement of interaction system is a very important requirement, especially now when the mobile communication technology is developing rapidly. The advantages of mobile web can be made full use of to make up the time and distance gap between cabs and customer and to provide fast and adequate customer services. Through the connection between mobile terminals and specific service, both cabs and customer are able to obtain required data to achieve a better interaction. Android is a Linux based open source operating system which is mainly used in portal devices with excellent performance thus making its market share growing. The platform, Web services and database technology are all gradually maturing, so that we can develop a cabs- customer interaction system on Android platform to meet the needs of the customer and provide cabs more efficient and convenient means of communication with customer.

1.2 Project Scope

This project has a wide scope.This would prove a Major breakthrough in reducing stress of user for booking the cab .The app will provide a great experience for user and also for individual driver or employee of OLA and uber .In future we will provide services of MERU,KAALI-PEELI and ETC, to this app thus creating a multi-Optional App.

1.3 Project Goals and Objectives

1.3.1 Goals

This project has a wide scope.This would prove a Major breakthrough in reducing stress of user for booking the cab .The app will provide a great experience for user and also for individual driver or employee of OLA and uber .In future we will provide services of MERU,KAALI-PEELI and ETC, to this app thus creating a multi-Optional App.

1.3.2 Objectives

There are mainly five objective in this project. The first objective is to fetch the data from the ola server about the cabs that are present near to the location of the source using ola API. The second objective is to fetch the data from UBER server about the cabs that are near to the source destination same as done in ola using API of UBER.The third objective is to store these data about the cabs which are in json format in a NOSQL database. The fourth objective is to compare the data using various greedy and comparing algorithm. The fifth objective is to display thses data yto the user in GUI of the application.

1.4 Organization of Report

In Chapter 1: we have considered Project overview under which we have explained various important terminologies like Introduction of the project.Motivation (what exactly motivated us to create priority based cab search),problem definition, About current system ,Advantages over current system,Goals and Objectives,Scope and Application.

In Chapter 2: we have discussed about various paper that we have referred for our project. We have mentioned the description , pros and cons and how the overcome the problem under every paper. a total of three paper have been referred.

In Chapter 3: we have discussed about the requirement analysis under it we have consider about the requirement the platform requirement supporting the os of the software and hardware requirement along with the feasible study.

In Chapter 4: we can see the system design and architecture various diagram can be seen in this chapter which represent the software , diagram including our system architecture usecase diagram dfd diagram class diagram and component diagram.

In Chapter 5: we have seen the methodology here we have explain the project in detail by dividing into module.various module of priority based cab search are explained with the help of few diagram.

In Chapter 6: we have discussed about the implementation details the assumption and dependencies this part contains details of the implementation of methodology that we discuss earlier.

In Chapter 7: we have shown the test cases and result along with analytic discussion this part contained the result of the output of the project.

In Chapter 8: we have concluded the whole project and future scope along with the limitation followed up by reference and chapter 9 with Appendix.

Chapter 2

Literature Survey

2.1 Study of Rest API

Representational state transfer (REST), which is a style of software architecture with simplicity and ’leaky, has been widely adopted in web services. To solve challenges like multidemensional API validation requirements, can sequencing and organization of test cases, data dependency between test cases, this thesis proposes a REST API testing model with an expressive description language based on JSON.

2.1.1 Advantages of Paper

a. It is used to get the information from the web server.

b. Easy to handle and efficient to use

2.1.2 Disadvantages of Paper

a. It compares taxi and private car but does not compare within taxi’s.

b. The problem was it only compares the price and not all the other parameter like time and distance .

2.1.3 How to overcome the problems mentioned in Paper

a. We will use a algorithm which is going to compare the factors such as time,distance, price in single attempt.We are also going to compare this factors within one systems that is online cabs.We are using Rest Api which is provided from the account of system which is scrapping data for single purpose in json format.

2.2 The Comparison of Travel Patterns between Taxi and Private Car at Beijing Capital International Airport Area

In last two years, the rapid development of Internet based ride-sharing has brought great changes to travel pattern of residents. However, few studies has been made to find out the unique travel patterns of Internet based ride-sharing. In this paper, License plate data at the Beijing Capital International Airport area are used to study the travel patterns of taxis and private cars. By comparing the trip records of taxis and private cars, two indicators, the distribution characteristics of vehicle volume and the balanced patterns of time, are selected to identify the ride-sharing cars from private cars. The results showed that there are more taxis than private cars arrival the airport in the evening and the arriving time of private cars is more concentrated. Finally, we used these indicators identify the ride-sharing vehicles from private cars. All these findings will be useful for further research and policy making.

2.2.1 Advantages of Paper

a. It gave the idea about the usage of cabs and private car in metro cities.

b. Algorithms used for comparison.

2.2.2 Disadvantages of Paper

a. It compares taxi and private car but does not compare within taxi's.

b. The problem was it only compares the price and not all the other parameter like time and distance .

2.2.3 How to overcome the problems mentioned in Paper

a. We will use a algorithm which is going to compare the factors such as time,distance, price in single attempt.We are also going to compare this factors within one systems that is online cabs.We are using Rest Api which is provided from the account of system which is scrapping data for single purpose in json format.

2.3 Best Price Algorithm in Finding routes based On Unconventional Public Transportation

2.3.1 Advantages of Paper

a. Algorithm used to calculate the optimal price.

b. Algorithm of distance vector to calculate the distance of the routes.

2.3.2 Disadvantages of Paper

a. It compares taxi and private car but does not compare within taxi's.

b. The problem was it only compares the price and not all the other parameter like time and distance .

2.3.3 How to overcome the problems mentioned in Paper

a. We will use a algorithm which is going to compare the factors such as time,distance, price in single attempt.We are also going to compare this factors within one systems that is online cabs.We are using Rest Api which is provided from the account of system which is scrapping data for single purpose in json format.

2.4 Technical Review

The technologies that we are using in our project are as follows:-
1.Android, 2.REST API, 3.Firebase, 4.XML

2.4.1 Advantages of Technology

a. Access apps from anywhere without installing in the device.Compatible with all the Android versions from Jelly Beans in ANDROID.

b. Due to its scalability. This protocol stands out due to its scalability. Thanks to the separation between client and server, the product may be scaled by a development team without much difficulty. Due to its flexibility and portability in REST API.

c. Built-in analytics High-quality app development in Firebase

2.4.2 Reasons to use this Technology

a. Android creates an App in Efficient way which is easy to use.

b. Firebase provides dynamic database for analysis,authentication and storage.

c. REST API is used to get the data from different server in simple format.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Shanawaz	Google login,Maps
2	Taha	UI Design,Logo design
3	Zohar	Firebase signUp,Data fetching display

Work Breakdown Structure

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Zohar	Team Leader	Firebase signUp,Data fetching display
2	Shanawaz	Member	Google login,Maps
3	Taha	Member	UI Design,Logo design

3.3 Assumptions and Constraints

Assumption is that the data that is coming from the ola and uber server is in correct from and validated.

Constraints:-
Variuos Company available but comparing only ola and uber.
Application is not independent.
if ola, uber and firebase server goes down system will fail.

3.4 Project Management Approach

We have use Agile methadology for the development of this project.The Agile Project Management Process is a value-centered methods of project management that allows projects to get processed in small phases or cycles. The methodology is one that is extremely flexible and projects that exhibit dynamic traits would benefit from this process as you would find that project managers working in this environment treat milestones the goal being to continuously adapt to abrupt changes from our project guide feedback.

3.5 Ground Rules for the Project

1. We treat each other with respect.

2. We intend to develop personal relationships to enhance trust and open communication.

3. We value constructive feedback. We will avoid being defensive and give feedback in a constructive manner.

4. As team members, we will pitch in to help where necessary to help solve problems and catch-up on behind schedule work.

5. Additional meetings can be scheduled to discuss critical issues or tabled items upon discussion and agreement with the team leader.

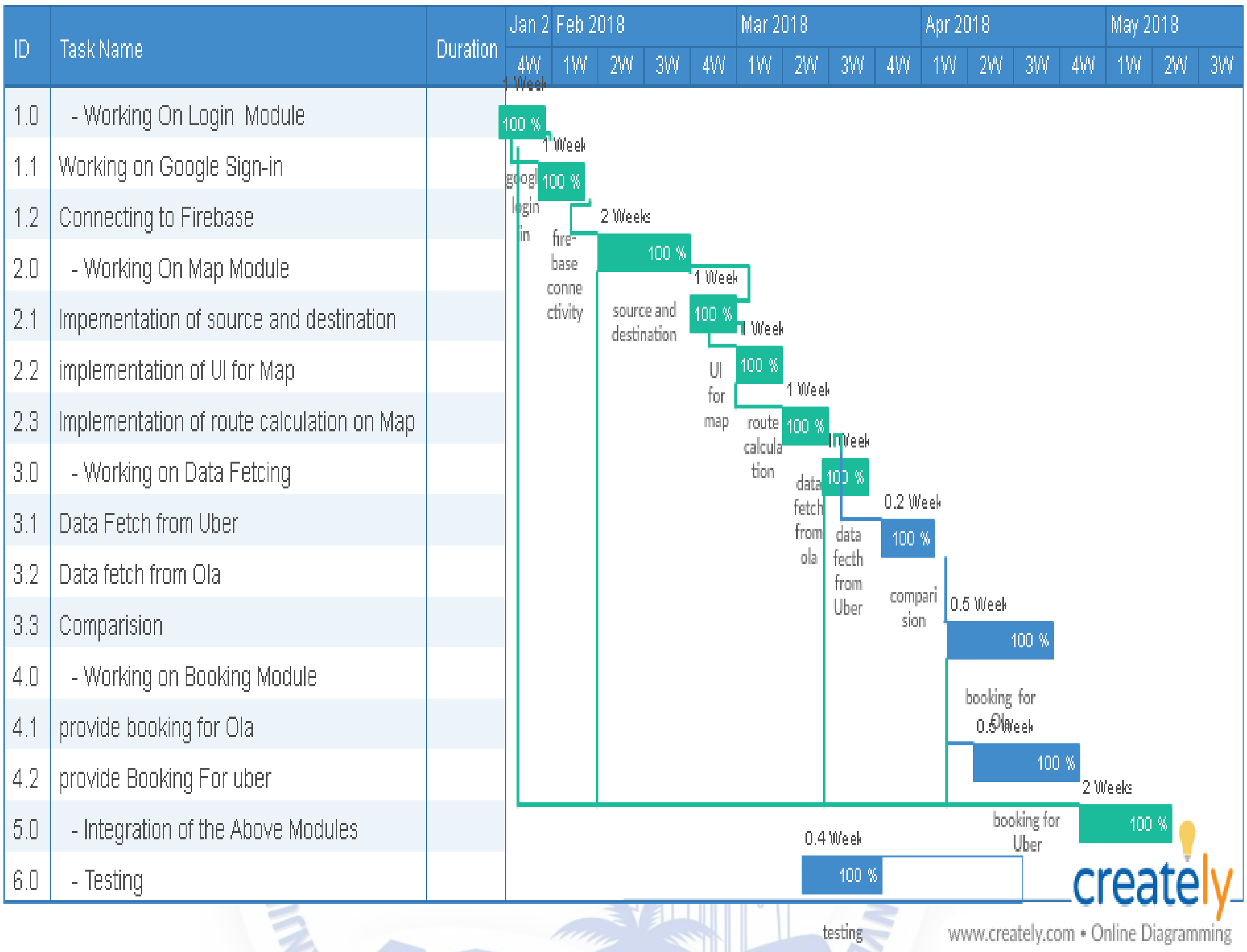
6. One person talks at a time; there are no side discussions.

7. When we pose an issue or a problem, we will also try to present a solution.

3.6 Project Budget

The budget for this project is very low as most of the tools we have use are open source.Following are the budget for the project
1. Operating System:linux mint (Open Source).
2. IDE:Andriod Studio (Open Source).
3. API:Google API,REST API(Open Source).

3.7 Project Timeline



ID	Task Name	Duration
1.0	- Working On Login Module	
1.1	Working on Google Sign-in	
1.2	Connecting to Firebase	
2.0	- Working On Map Module	
2.1	Impementation of source and destination	
2.2	implementation of UI for Map	
2.3	Implementation of route calculation on Map	
3.0	- Working on Data Fetcing	
3.1	Data Fetch from Uber	
3.2	Data fetch from Ola	
3.3	Comparision	
4.0	- Working on Booking Module	
4.1	provide booking for Ola	
4.2	provide Booking For uber	
5.0	- Integration of the Above Modules	
6.0	- Testing	

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

The origin of the product that is our App is totally based on the two big cab industry App that is Ola and Uber ,it provide a similar UI and features same as that of Ola and Uber. but it is an extention to both the App that is it will Compare both the App based on Price,time And Distance.it will be useful for those type of user who like to travel Affordably and will help also help the two biggest cab industry to boom in the existting Environment

4.1.2 Product Features

The major features of the product is it provides a quick glance on price of both Ola and Uber and provide User to select one of them to book a cab based on their choice and also provides a Google authentication to ensures that user information is secure and safe.

4.1.3 User Classes and Characteristics

Different User will Use the Product Differently and the user class related to the app will change according to the need of user.but the pertinent characteristics of the classes will remain the same and user will primary interact will three main class of product that is authentication,maps and booking class,the rest are less important according to this three class. User interaction with this Classes will Also enhance The User experience with the Product.

4.1.4 Operating Environment

The environment in which the software will operate is Android And the hardware platform On which The software will run will be Any android based Smart Phone with GPS enabled.The android Version should be Greater that 5.1 that is Android

Lolly pop or Higher. and the software will run peacefully Co-exist in the smart phone until it is uninstalled.

4.1.5 Design and Implementation Constraints

The major challenge that will hurdle the development of the system is the failure of the server of ola and uber,if the server does not connect properly and responds quickly than whole system is of no use.Another constraint would be the internet connectivity,if there is no internet connection available than all the services will not be provided to the user.Also of the firebase data connectivity is lost with the application than the system will fail to authenticate the user.

4.2 System Features

The major features of our system is to provide the estimated comparison of the rates,ETA,distance and availability of the cabs from the source location of the user.The major comparison is between the cabs of the OLA and UBER cab industries.As the online cab booking is booming in toady's world.

4.2.1 System Feature

1.Optimal comparison.
2.Google authentication.
3.Easy booking.
4.Optimal data usage.
5.Optimal memory usage.

Description and Priority

1.Optimal comparison
This feature will compare the cabs rates and between the cabs of ola and uber and will show to the user on the UI.
priority:-High.
2.Google authentication
This feature will authenticate the user than only will provide the login into the system.
priority:-High
3.Easy booking
This feature will allow the user to book the desired cab from our system.
priority:-Medium.

Stimulus/Response Sequences

1.The user need to login in to the system.
2.The user will enter the source and destination
3.The optimized cabs rate will be available to the user on the system
4.The user will select the cab and will book the ride

Functional Requirements

1.The user should login in to the system.
2.The user should enter correct source and destination location.
3.The correct details of the compared cabs should be available.
4.The servers should response quickly.

4.3 External Interface Requirements

4.3.1 User Interfaces

1. User shall be able to login in the system.
2. After the login session shall be maintain.
3. The user can enter the source and destination.
4. The user can view the optimized cabs suggested by the system.
5. The user can book the cab.

4.3.2 Hardware Interfaces

Andriod enabled device:The andriod enabled device should have andriod version above 5.0.Inorder for the smooth functioning of the application the andriod device must have atleast 512mb of ram and atleast 200mb of free storage on device. The application can also function on a tablet device.

4.3.3 Software Interfaces

Operating System:Andriod above 4.4.
Databaes:google Firebase.
tools:andriod studio IDE.
Apis:google maps,REST API.

4.3.4 Communications Interfaces

1.The major communication for location purposes will be done by google api,the data is accessed by the google by using the google apis.

2.The interface between the firebaseDB and the system will be done by using http protocol

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

1.Optimize comparison
The performance for this feature is mainly dependent on the cabs details that will be provided by the ola and uber.If the cabs details are not proper than the comparison will not be optimal.
2.Authentication
The performance for this feature is mainly dependent on the google id of the user.If the google id and account is not there than login in to the system wont be done .

4.4.2 Safety Requirements

If there is any damage to the servers pf ola and uber as well as to the server of firebase than the whole system will go down.
The database should be periodically maintained and have to keep upon it.
The data which is updated by the user should be commited in the database.

4.4.3 Security Requirements

The major security requirements for the system will be the safeguarding of the user data from any kind of exploit. Inorder to protect the user data the data is not stored in local databases we will be storing in the cloud for better security.

Chapter 5

System Design

5.1 System Requirements Definition

Our system is an andriod application on a mobile device,the system will function overall on the basis of the location which is captured by the gps location.we have survey various application related to our project.we have decided the system specification for our project.we have studied the end-user requirements and based on that we have decided the functional and non-functional requirement.

5.1.1 Functional requirements

1.The customers must register for create the account and login using username and password to use function in the application.
2.Fetch the distance and route from Google maps using source and destination.
3.Optimize cab details is shown on the GUI.
4.User can book the desired cab.

Use-case Diagram

This is the Usecase diagram of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

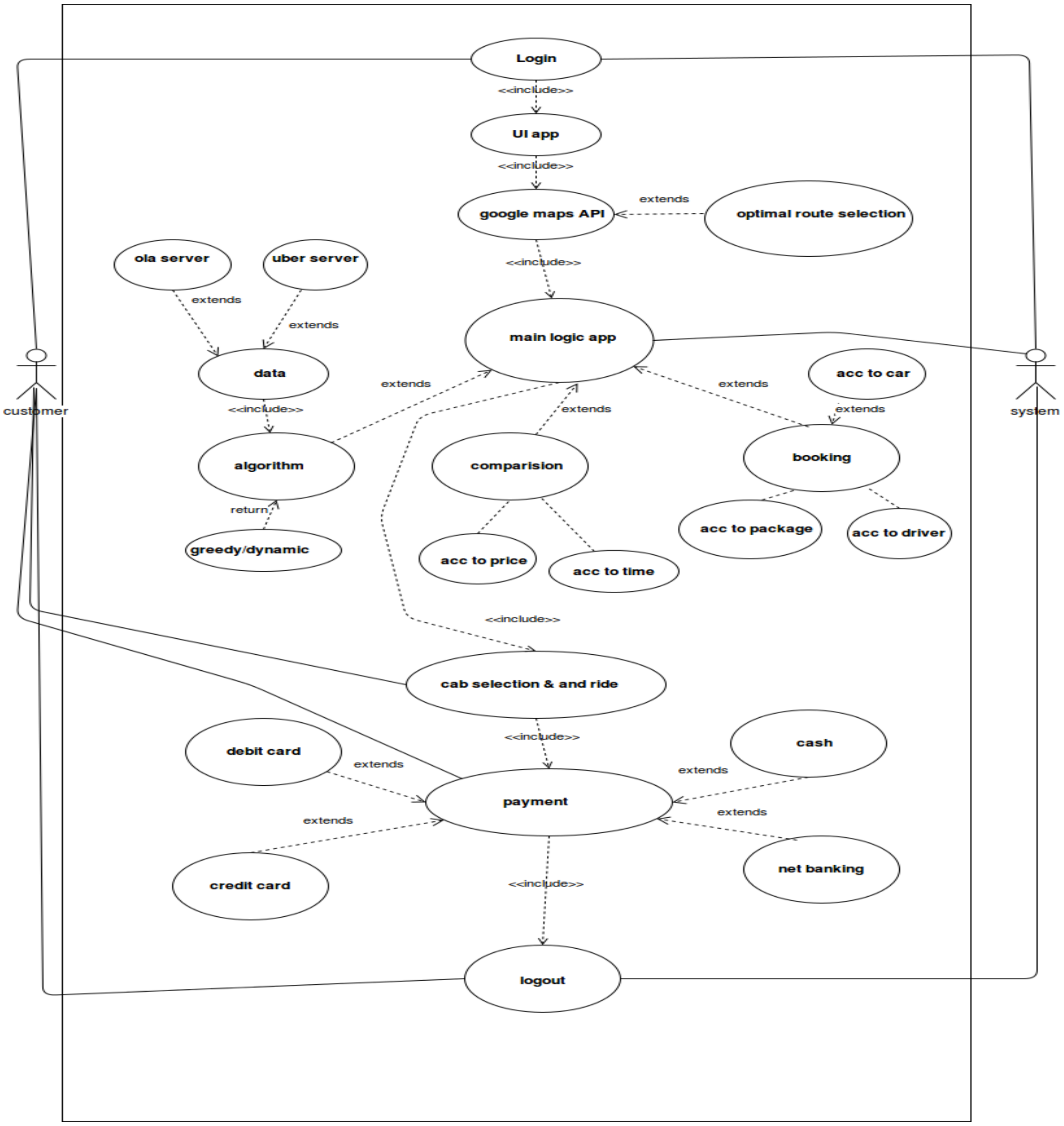


Figure 5.1: Usecase

Data-flow Diagram

This is the Data flow diagram level 0 of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

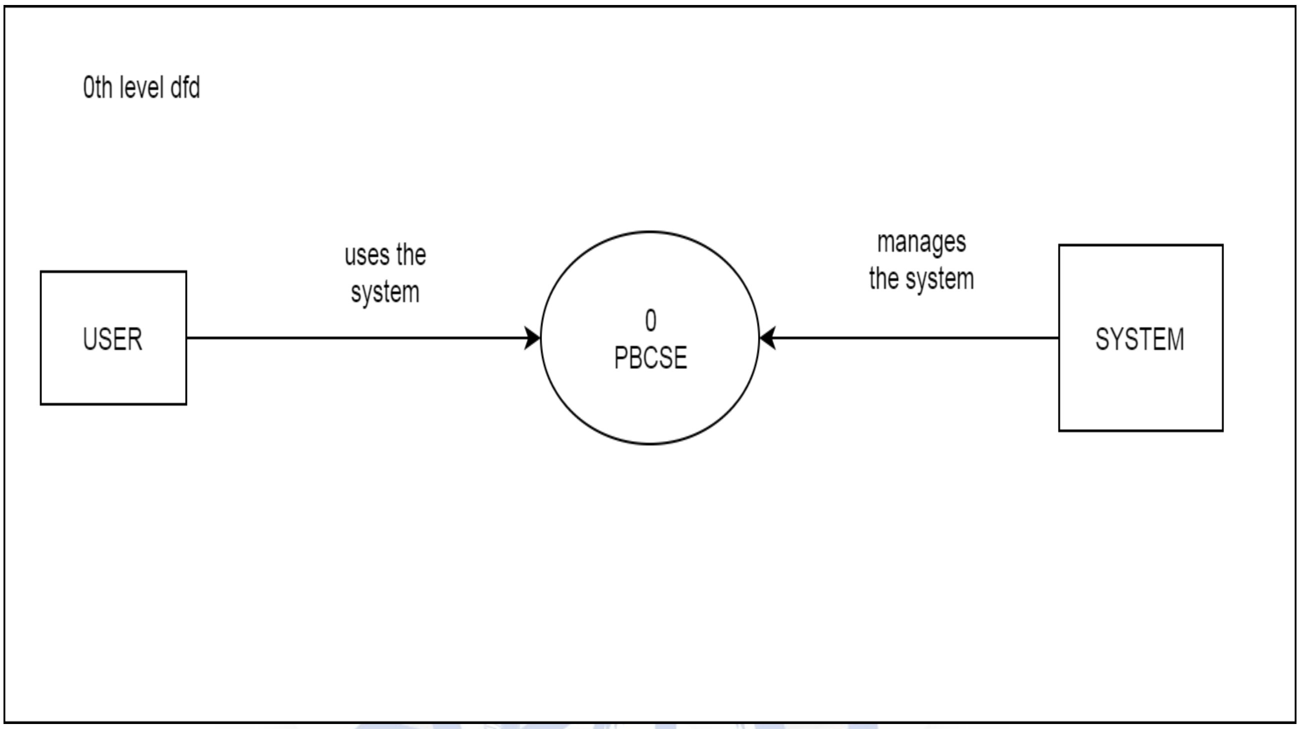


Figure 5.2: Dfd level 0

This is the Data flow diagram level 1 of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

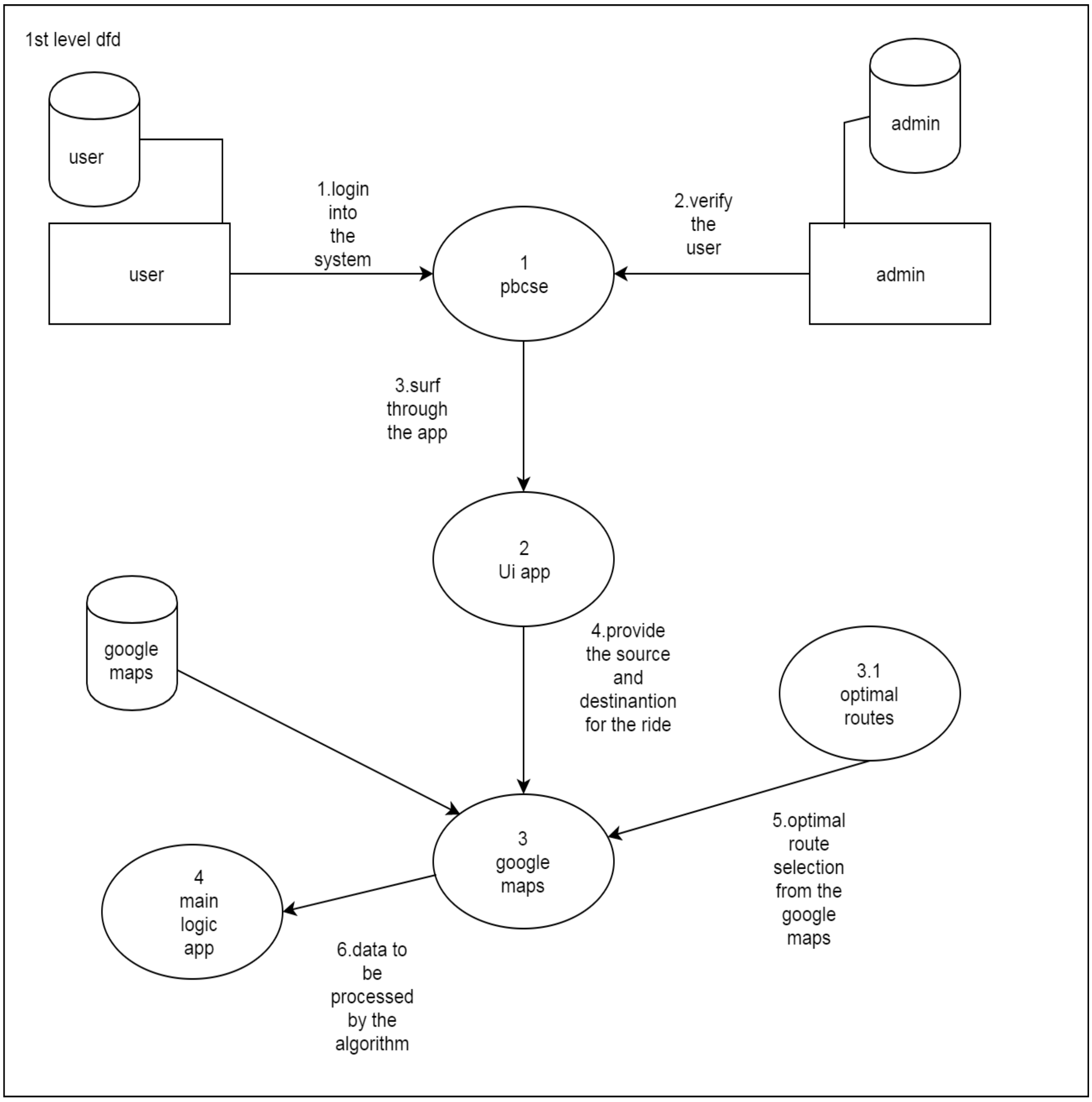


Figure 5.3: Dfd level 1

This is the Data flow diagram level 2 of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

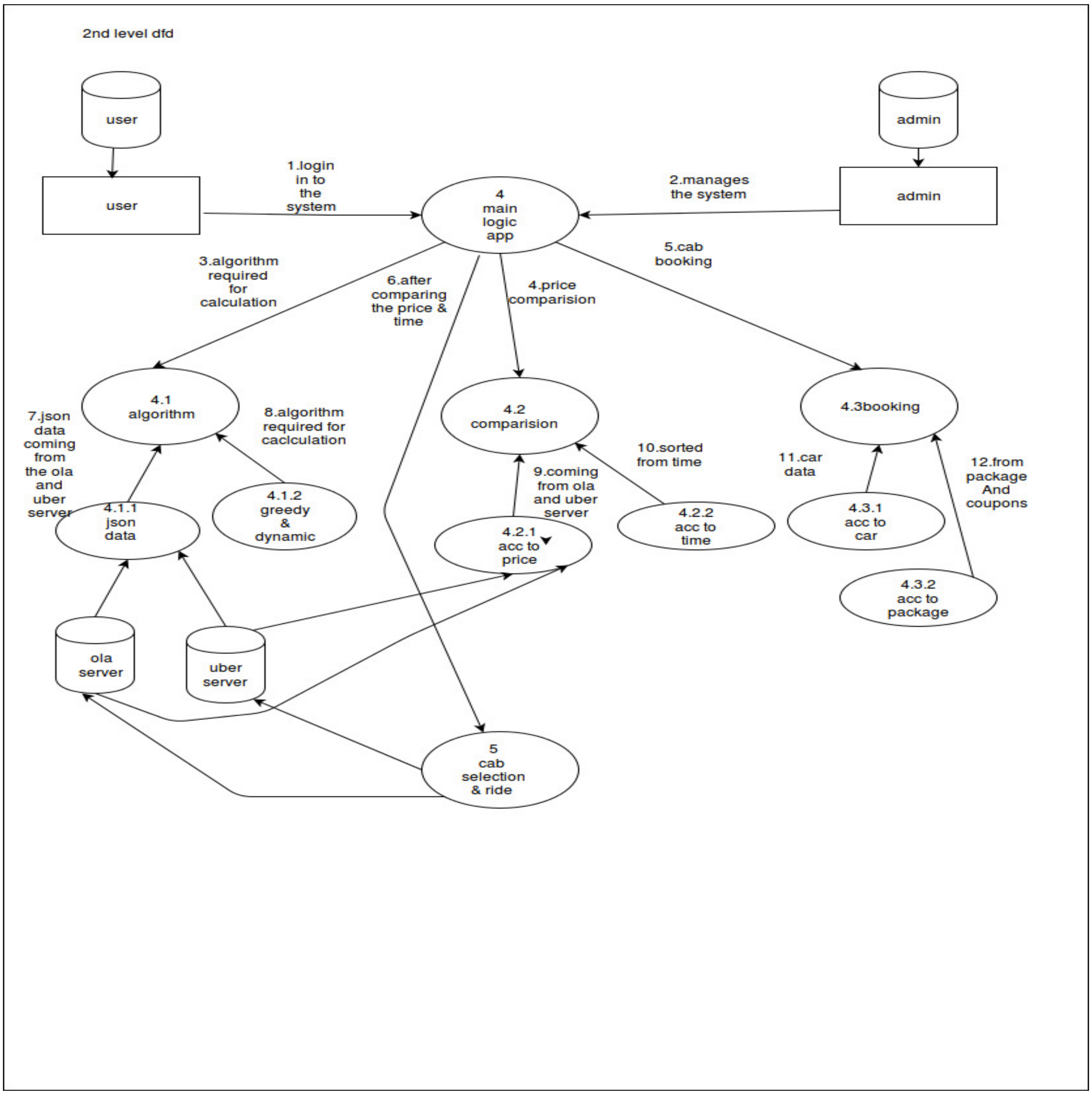


Figure 5.4: Dfd level 2

5.1.2 System requirements (non-functional requirements)

Performance Requirements

Optimal memory usage:Since we are comparing two apps the installation of two different application is avoided which intern favours as a memory saving factor.

Security:To provide a safe usage of the system authentication id a user is done through Google id

Safety Requirements

To provide the system from unauthorized user we are using Google id as a basic requirement to log in into the system which will provide a better safety feature .the database of the system is done using firebase which a feature of the Google which is also safe to implement.

Security Requirements

The major security requirements for the system will be the safeguarding of the user data from any kind of exploit.
Inorder to protect the user data the data is not stored in local databases we will be storing in the cloud for better security..

Database Schema/ E-R Diagram

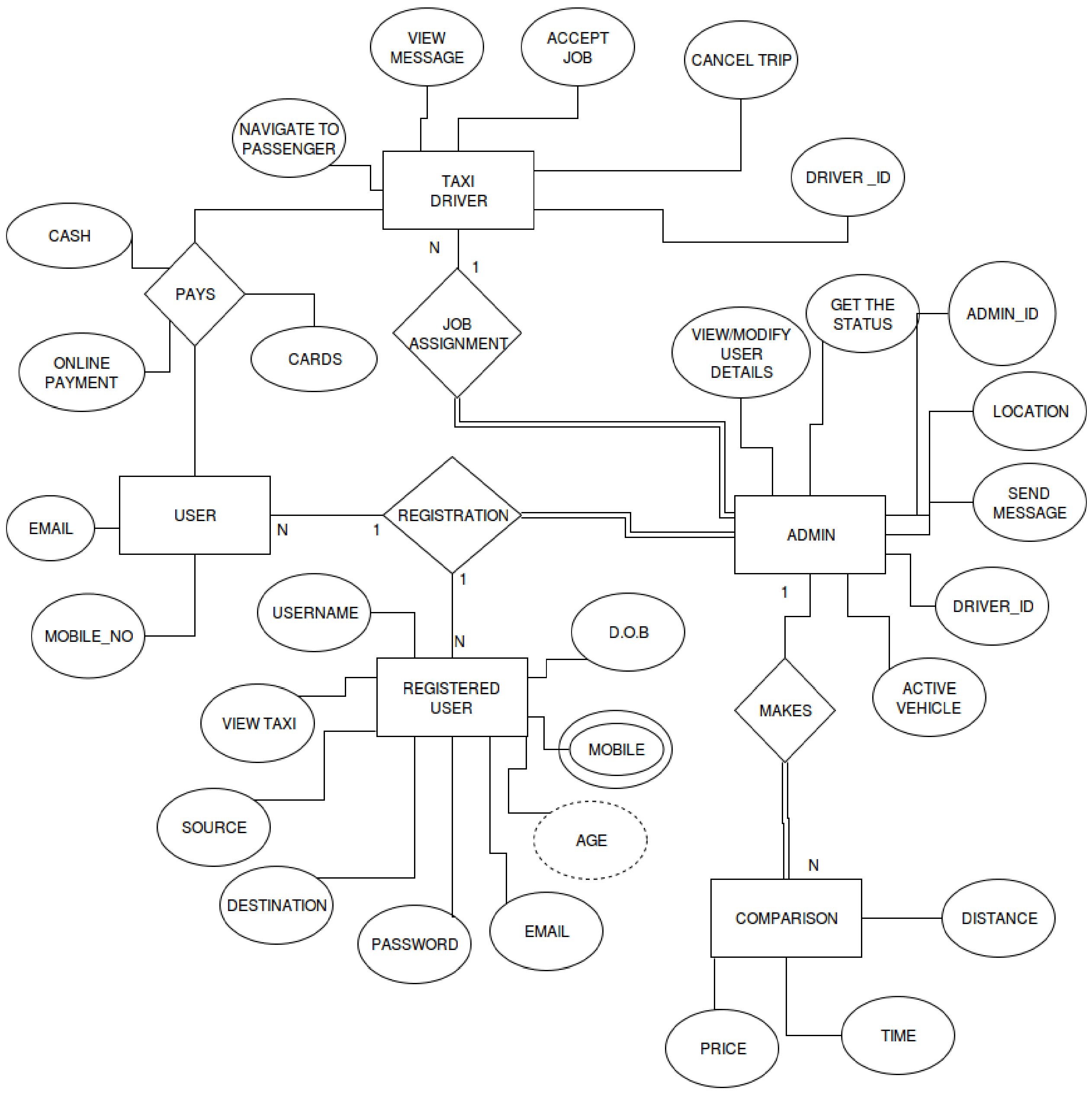


Figure 5.5: E-R Diagram

This is the E-R diagram of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

5.2 System Architecture Design

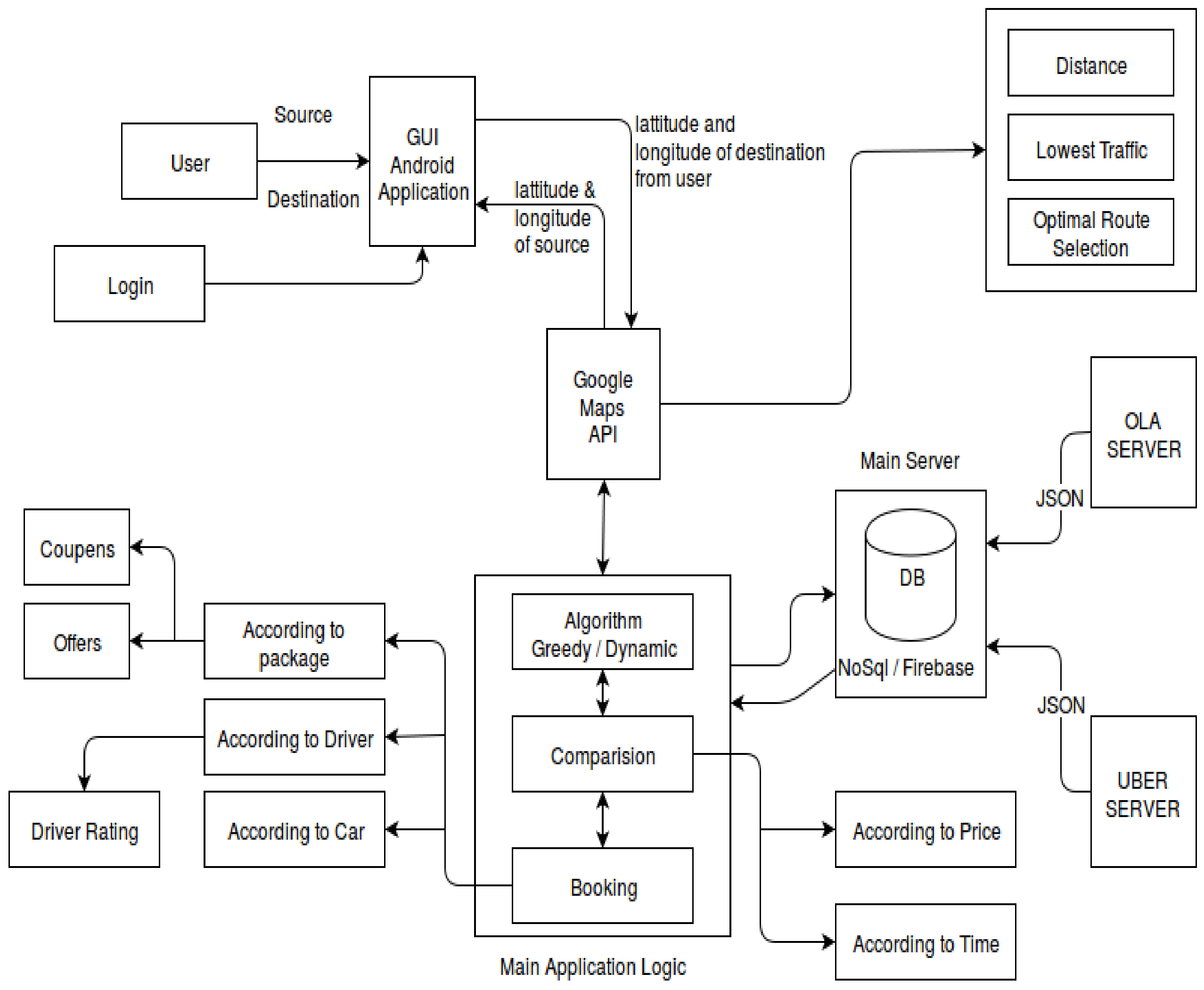


Figure 5.6: System Architecture

This is the System Architecture diagram of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

5.3 Sub-system Development

Login:-In login module the authentication part is covered and it provides a safe login into system via google id.

Maps:-The source and destination location provided by the user will be fetched to the Google maps and the output will be the distance calculated and route shown.

Data Fetch:-About the data fetch, the from the ola and uber server will be fetch about the cabs details in json format.

Main Application:-In the main application the internal working of the system is shown.The data comparison and the further bookig module is shown.

Booking Module:-In booking module the booking of the cab is shown via different parameters of based on car and driver rating or based on fare and ETA.

5.3.1 Booking Module:

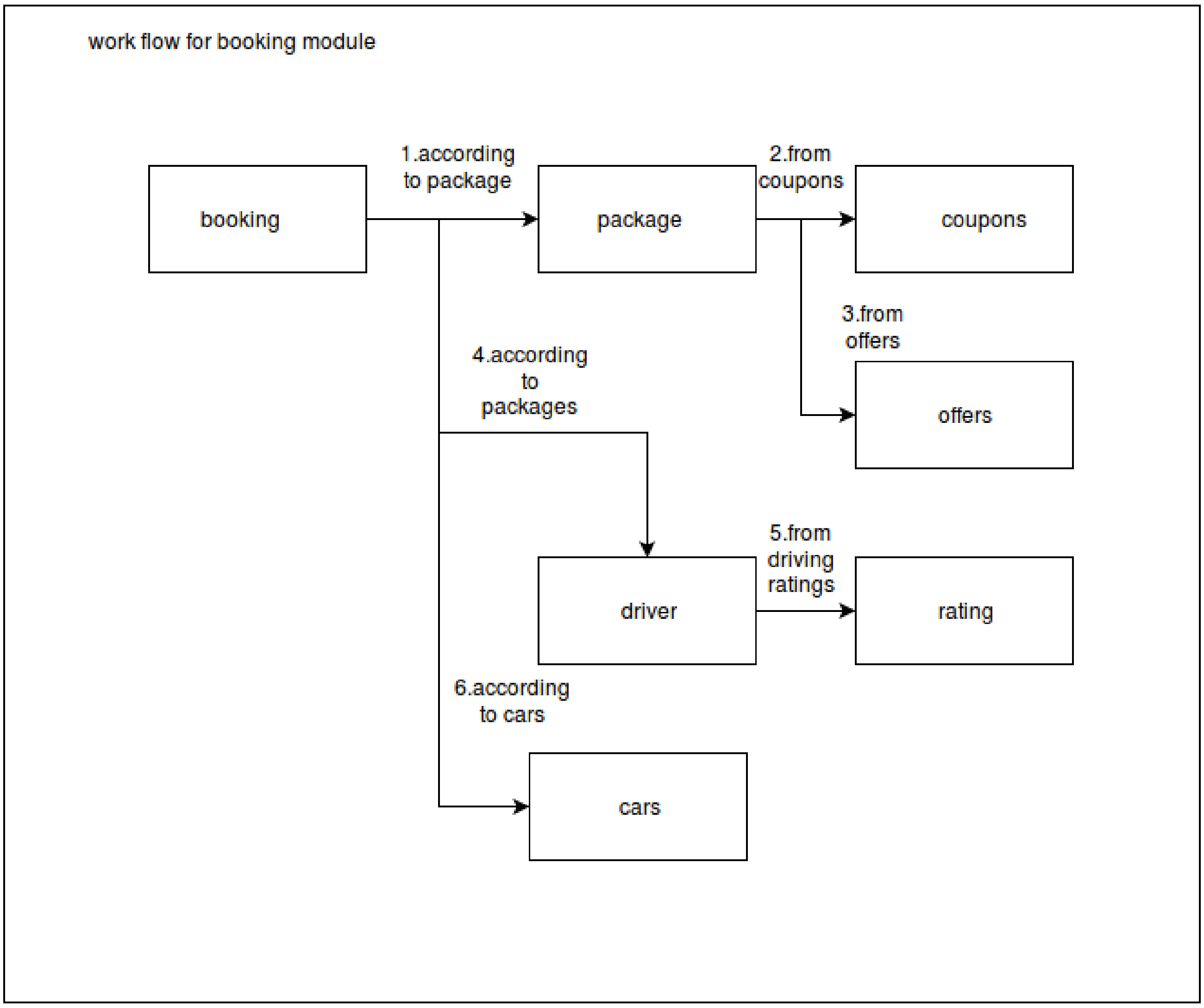


Figure 5.7: Flow Diagram For Booking Module

5.3.2 Main Application Logic Module

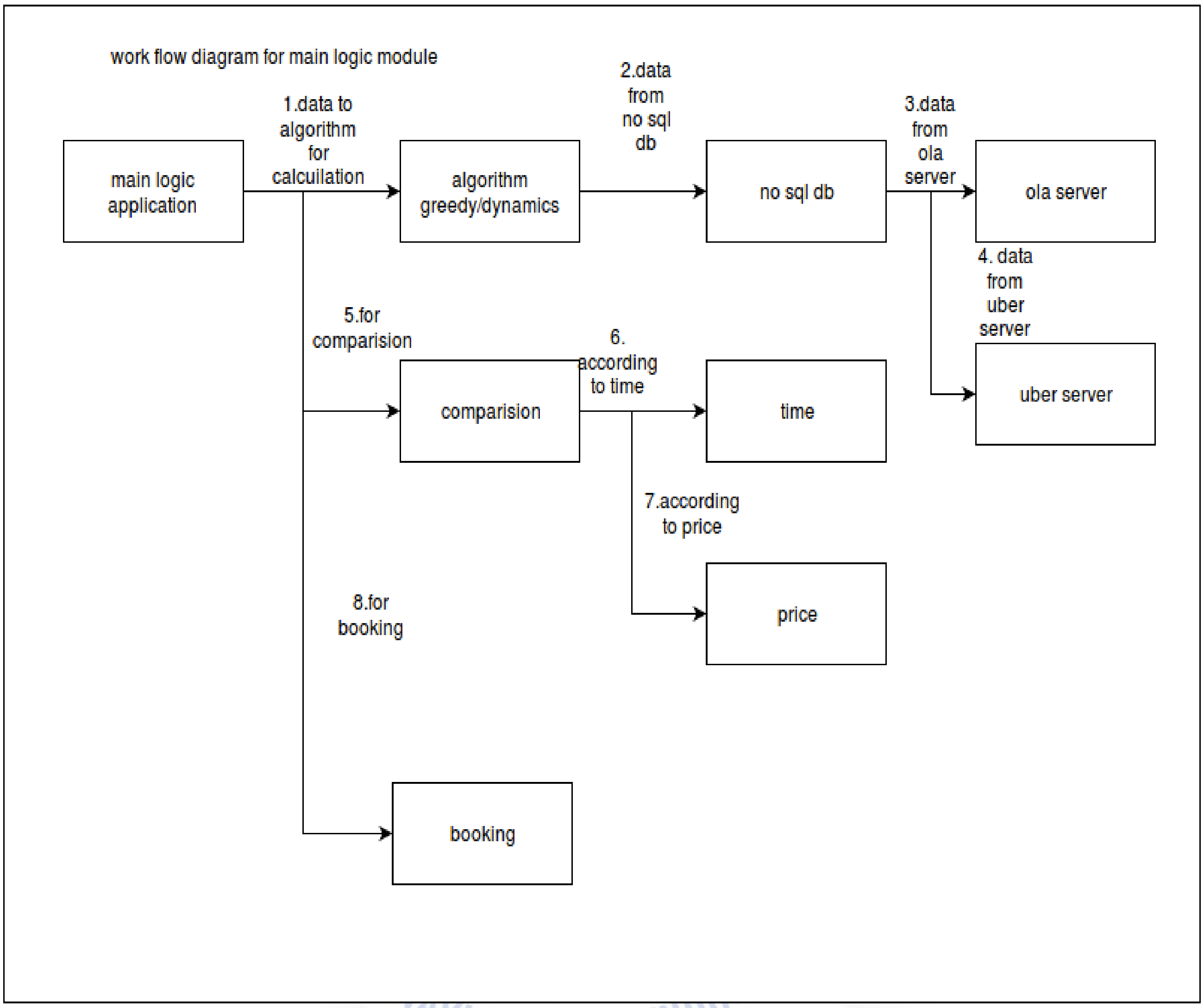


Figure 5.8: Main Application Logic Module Flow Diagram

5.3.3 Login Module

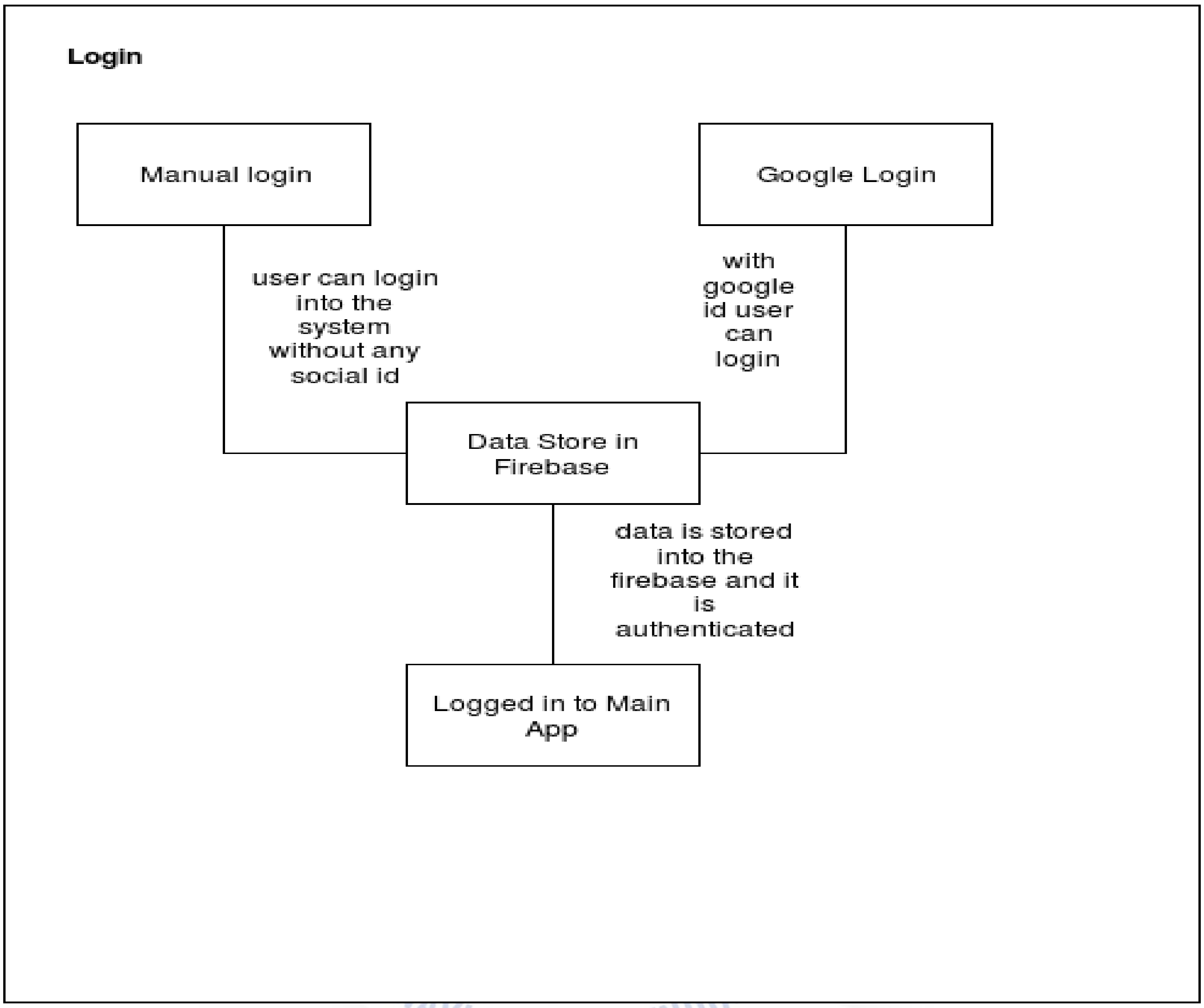

Login
Manual login
Google Login
user can login into the system without any social id
with google id user can login
Data Store in Firebase
data is stored into the firebase and it is authenticated
Logged in to Main App

Figure 5.9: Login Module Flow Daigram

5.3.4 Map Module

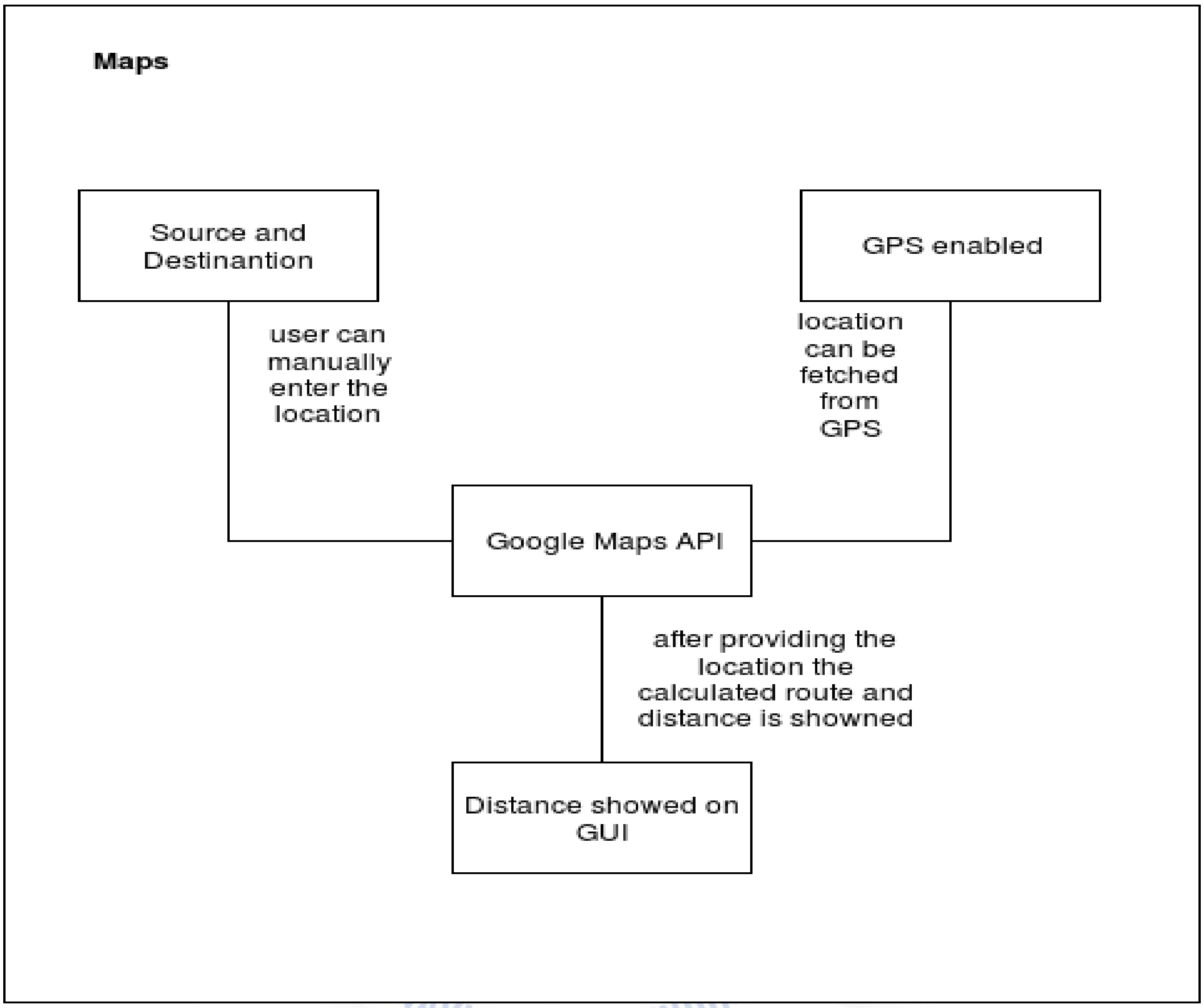

Maps
Source and Destinantion
GPS enabled
user can manually enter the location
location can be fetched from GPS
Google Maps API
after providing the location the calculated route and distance is showned
Distance showed on GUI

Figure 5.10: Map Module Flow Diagram

5.3.5 Data-Fetch Module

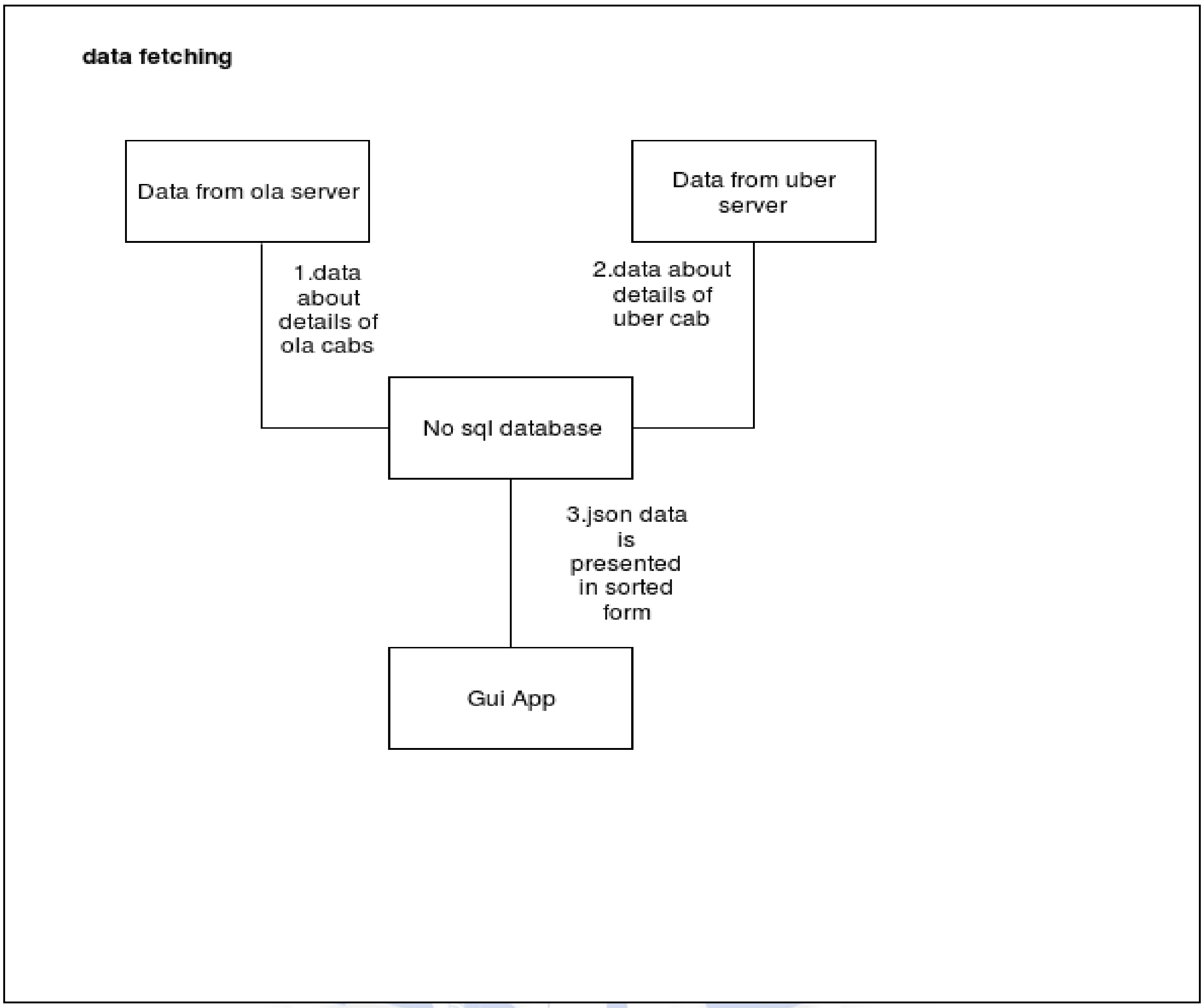

data fetching
Data from ola server
Data from uber server
1.data about details of ola cabs
2.data about details of uber cab
No sql database
3.json data is presented in sorted form
Gui App

Figure 5.11: Data-Fecth Module Flow Diagram

5.4 Systems Integration

First module of the system is login for which are using Google accounts. Second module of the system is the maps which are used to provide the distance for the location. Third module of the system is data fetching in which the json data about the cabs is fetch from the respective servers. all this module are integrated in to one whole app module using the android functionality of integrating.

5.4.1 Class Diagram

This is the Class diagram of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

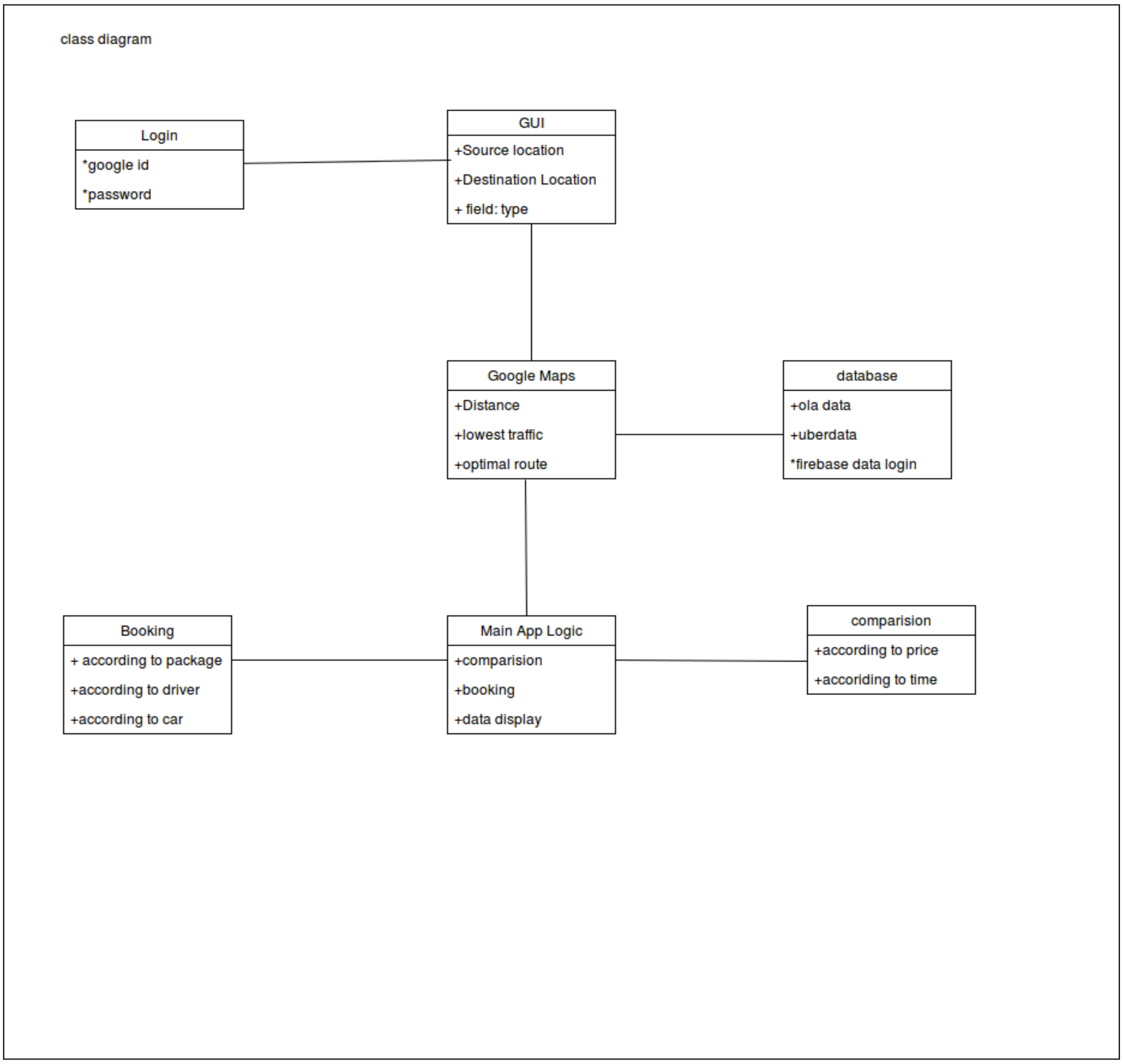


Figure 5.12: Class Diagram

5.4.2 Sequence Diagram

This is the Sequence diagram of the system in which the modules which will be there after the deployment are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

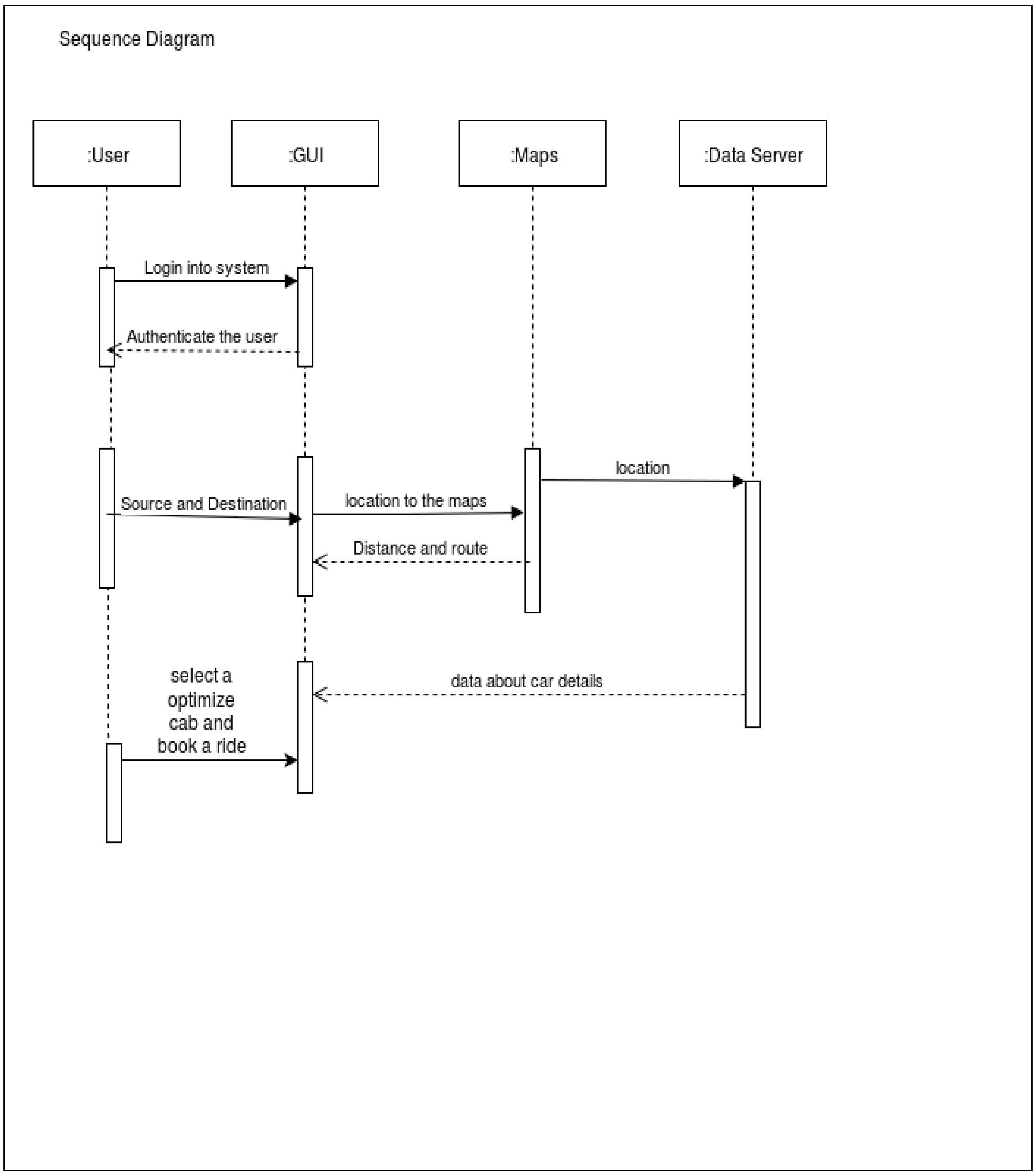


Figure 5.13: Sequence Diagram

5.4.3 Component Diagram

This is the Component diagram of the system in which the modules which will be there after the deployment of the project are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

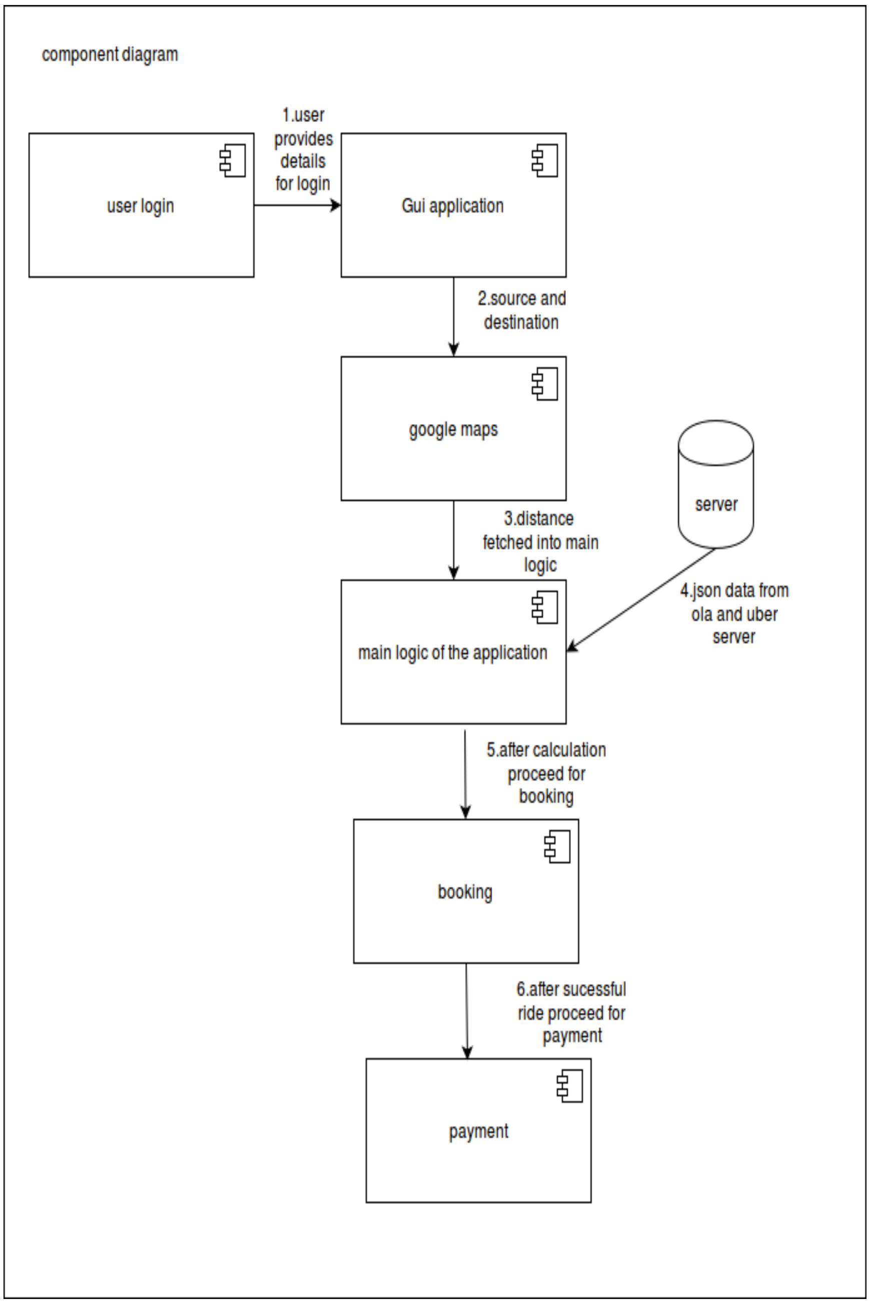

component diagram
user login
1.user provides details for login
Gui application
2.source and destination
google maps
3.distance fetched into main logic
server
4.json data from ola and uber server
main logic of the application
5.after calculation proceed for booking
booking
6.after sucessful ride proceed for payment
payment

Figure 5.14: Component Diagram

5.4.4 Deployment Diagram

This is the Deployment diagram of the system in which the modules which will be there after the deployment of the project are shown . the server of the respected cabs engine as well as the server of the firebase database is shown for easy understanding of the project.

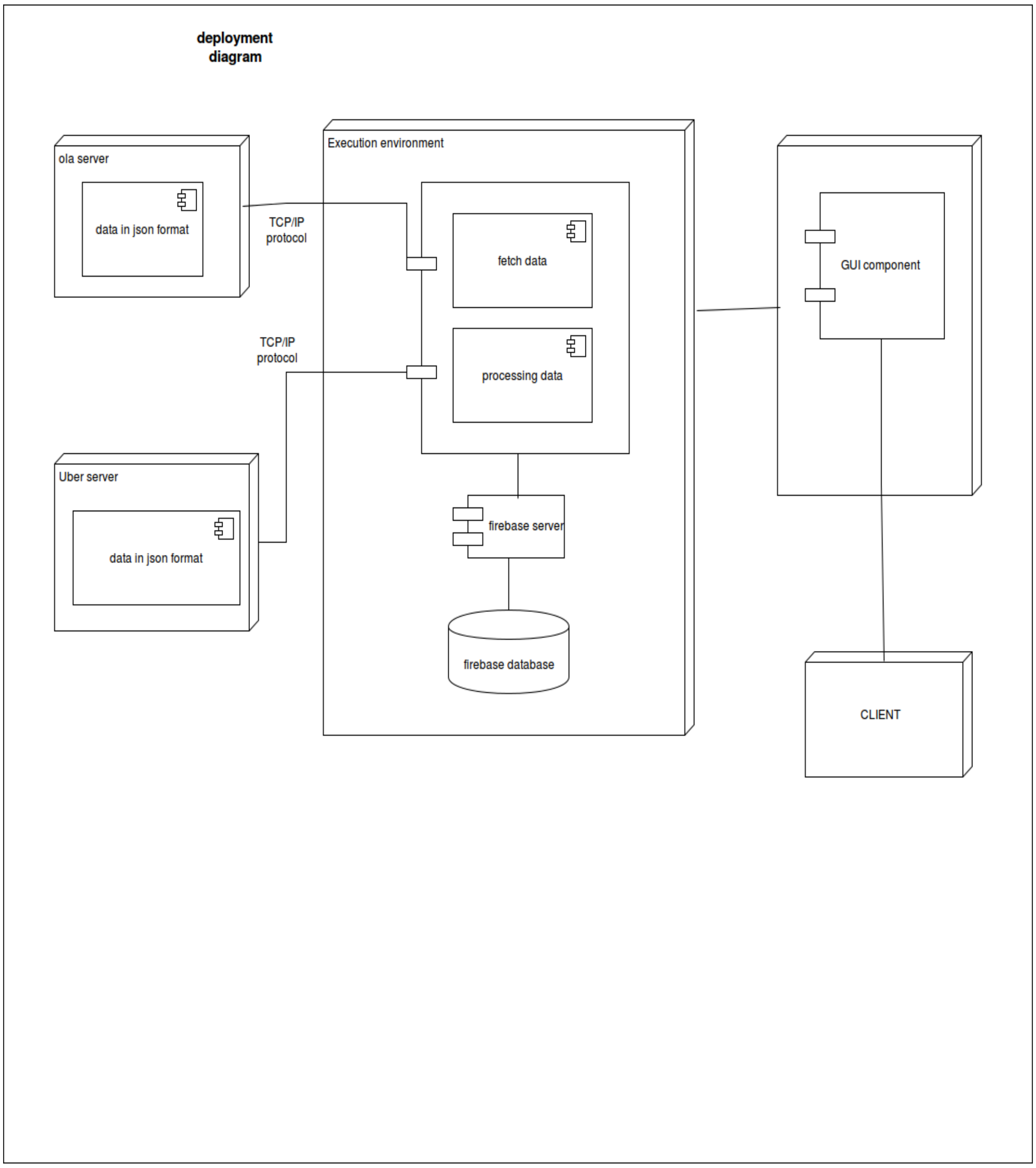


Figure 5.15: Deployment Diagram

Chapter 6

Implementation

6.1 Login Module

The Login module is the first module of the App and which give access to the other modules of the app.The login is provide through Google account or through manually through the database firebase is used as a database for the Login module.Google account is used for security and privacy of the User.if the User is successfully loged in then it can access the other modules of the App.

```
firebaseAuth = FirebaseAuth.getInstance();
        //if the objects getcurrentuser method is not null
         //means user is already logged in
         if(firebaseAuth.getCurrentUser() != null){
             //close this activity
             finish();
             //opening profile activity
             startActivity(new Intent(getApplicationContext(), MapsActivity.class
                 ));
         }
         //initializing views
         editTextEmail = (EditText) findViewById(R.id.editTextEmail);
         editTextPassword = (EditText) findViewById(R.id.editTextPassword);
         buttonSignIn = (Button) findViewById(R.id.buttonSignin);
         textViewSignup  = (TextView) findViewById(R.id.textViewSignUp);
         progressDialog = new ProgressDialog(this);
         //attaching click listener
         buttonSignIn.setOnClickListener(this);
         textViewSignup.setOnClickListener(this);
    //—our logic
        button = (Button) findViewById(R.id.google_btn);
         mAuth = FirebaseAuth.getInstance();
         button.setOnClickListener(new View.OnClickListener() {
             @Override
             public void onClick(View view) {
                 signIn();
             }
         });
        mAuthListener =new FirebaseAuth.AuthStateListener() {
             @Override
             public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {

                 if (firebaseAuth.getCurrentUser() != null) {
                     startActivity(new Intent(login.this, MapsActivity.class));
                     finish();

                 }   } };
         GoogleSignInOptions gso= new GoogleSignInOptions.Builder(
             GoogleSignInOptions.DEFAULT_SIGN_IN)
                 .requestIdToken(getString(R.string.default_web_client_id))
                 .requestEmail()
                 .build();
         mGoogleApiClient = new GoogleApiClient.Builder( this)
                 .enableAutoManage(this, new GoogleApiClient.
                     OnConnectionFailedListener() {
                     @Override
                     public void onConnectionFailed(@NonNull ConnectionResult
                         connectionResult) {
                          Toast.makeText( login.this,"Somenthing went wrong",Toast
                              .LENGTH_SHORT).show();
                     }
                 })
                 .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                 .build();
```

6.2 Maps Module

Maps Modules is used to get the current location of the user and also the required source and destination of the user and also to show the map on the UI of the map and also to Calculate the route between the source and destination.This data is then given to rest api to call the data fetching module to provide the data o available cabs of ola and Uber respectively.

```
    @Override
       public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.getUiSettings().setMapToolbarEnabled(false);
    mMap.getUiSettings().setZoomControlsEnabled(false);

    //Initialize Google Play Services
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mMap.setMyLocationEnabled(true);
        }
    }
    else {
        buildGoogleApiClient();
        mMap.setMyLocationEnabled(true);
    }
    //auto complete box code

    placeAutoComplete1 = (PlaceAutocompleteFragment) getFragmentManager().
        findFragmentById(R.id.place_autocomplete);
    placeAutoComplete1.setHint("Enter Source");
    placeAutoComplete1.setOnPlaceSelectedListener(new PlaceSelectionListener() {
            @Override
        public void onPlaceSelected(Place place) {
            Log.d("Maps", "Place selected: " + place.getName());
            //Toast.makeText(MapsActivity.this,place.getName(), Toast.
                LENGTH_SHORT).show();
            mMap.clear();
            MarkerPoints.clear();
            mMap.addMarker(new MarkerOptions()
                    .position(place.getLatLng())
                    .title(place.getName().toString()));
            MarkerPoints.add(place.getLatLng());
            MapsActivity.pickup_place= (List<Address>) place.getAddress();
        }
        @Override
        public void onError(Status status) {
            Log.d("Maps", "An error occurred: " + status);
        } });

    placeAutoComplete2 = (PlaceAutocompleteFragment) getFragmentManager().
        findFragmentById(R.id.place_autocomplete2);
    placeAutoComplete2.setHint("Enter Destination");
    placeAutoComplete2.setOnPlaceSelectedListener(new PlaceSelectionListener() {

        @Override
        public void onPlaceSelected(Place place) {

            Log.d("Maps", "Place selected: " + place.getName());
            //Toast.makeText(MapsActivity.this,place.getName(), Toast.
                LENGTH_SHORT).show();

            mMap.addMarker(new MarkerOptions()
                    .position(place.getLatLng())
                    .title(place.getName().toString())
                    .icon(BitmapDescriptorFactory.defaultMarker(
                        BitmapDescriptorFactory.HUE_GREEN))
            );
            MapsActivity.drop_place = (List<Address>) place.getAddress();
            MarkerPoints.add(place.getLatLng());
            if (MarkerPoints.size() == 2) {
                LatLng origin = MarkerPoints.get(0);
                LatLng dest = MarkerPoints.get(1);

                // Getting URL to the Google Directions API
                String url = getUrl(origin, dest);
                Log.d("onMapClick", url);
                FetchUrl FetchUrl = new FetchUrl();

                // Start downloading json data from Google Directions API
                FetchUrl.execute(url);
                //move map camera
                mMap.moveCamera(CameraUpdateFactory.newLatLng(origin));
                mMap.animateCamera(CameraUpdateFactory.zoomTo(16));   } }

        @Override
        public void onError(Status status) {
            Log.d("Maps", "An error occurred: " + status);
        }
    });
    Log.d("MyMarker","size"+MarkerPoints.size());

    Button clrbtn=(Button) findViewById(R.id.btnclr);
    clrbtn.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            mMap.clear();
            MarkerPoints.clear();
            placeAutoComplete1.setText("");
            placeAutoComplete2.setText("");
        }});

    Button subbtn=(Button) findViewById(R.id.btnsubmit);
    subbtn.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            if (MarkerPoints.size() == 2) {
                startActivity(new Intent(MapsActivity.this, MainActivity.class)
                        .putExtra("mylist", MarkerPoints)
                );

            }
            else
            {
                Toast.makeText(MapsActivity.this, "Please Enter 2 location",
                    Toast.LENGTH_SHORT).show();
            }

            //finish();
        }});

    // Setting onclick event listener for the map
    mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {

        @Override
        public void onMapClick(LatLng point) {

            // Already two locations
            if (MarkerPoints.size() > 1) {
                MarkerPoints.clear();
                mMap.clear();

            }

            // Adding new item to the ArrayList
            MarkerPoints.add(point);

            // Creating MarkerOptions
            MarkerOptions options = new MarkerOptions();

            // Setting the position of the marker
            options.position(point);
           /*
             * For the start location, the color of marker is GREEN and
             * for the end location, the color of marker is RED.
             */
            Log.d("MarkerPoints", "current: " + MarkerPoints);
            if (MarkerPoints.size() == 1) {
                options.icon(BitmapDescriptorFactory.defaultMarker(
                    BitmapDescriptorFactory.HUE_GREEN));
            } else if (MarkerPoints.size() == 2) {
                options.icon(BitmapDescriptorFactory.defaultMarker(
                    BitmapDescriptorFactory.HUE_RED));
            }
            // Add new marker to the Google Map Android API V2
            mMap.addMarker(options);
            // Checks, whether start and end locations are captured
            if (MarkerPoints.size() == 2) {
                LatLng origin = MarkerPoints.get(0);
                LatLng dest = MarkerPoints.get(1);
                // Getting URL to the Google Directions API
                String url = getUrl(origin, dest);
                Log.d("onMapClick", url);
                FetchUrl FetchUrl = new FetchUrl();
      //Code For Geo Cordinates
                Geocoder geocoder = new Geocoder(MapsActivity.this, Locale.
                    getDefault());
                List<Address> addresses = null;
                try {
                      addresses = geocoder.getFromLocation(origin.latitude,
                          origin.longitude,1);
                       pickup_place = addresses;
                       addresses = geocoder.getFromLocation(dest.latitude, dest.
                           longitude,1);
                       drop_place=addresses;

                   } catch (IOException e) {
                       e.printStackTrace();
                 }
```

6.3 Data Fetching Module

This Module is Used to fetch the Data from Ola and Uber server Respectively.the code given the Description on how the data is fetched.the data that is Fetched is in JSON format.data fetching process is dynamic in nature and REST API is used for it and real Time data is been fetched to provide accuracy to the user of the app.

UBER DATA FETCHING

```
public class UberClient {

  private static final String API_BASE_URL = "https://api.uber.com/v1.2/";
  private static HttpLoggingInterceptor logging = new HttpLoggingInterceptor()
      ;

  private static Retrofit.Builder builder =
      new Retrofit.Builder()
          .baseUrl(API_BASE_URL)
          .addConverterFactory(GsonConverterFactory.create());

  private static Retrofit getRetrofit(Interceptor interceptor) {
      OkHttpClient.Builder httpClientBuilder = new OkHttpClient.Builder();

      if (interceptor != null) {
          httpClientBuilder.addInterceptor(interceptor);
      }

      if (BuildConfig.DEBUG) {
          logging.setLevel(HttpLoggingInterceptor.Level.BODY);
          httpClientBuilder.addInterceptor(logging);
      }

      Retrofit retrofit = builder
              .client(httpClientBuilder
                      .build())
              .build();

      return retrofit;
  }

  public static <S> S createService(Class<S> serviceClass) {
      Retrofit retrofit = getRetrofit(null);
      return retrofit.create(serviceClass);
  }

  public static <S> S createBasicAppToken(Class<S> serviceClass) {
      UberTokenAuthorizationInterceptor olaAuthInterceptor = new
          UberTokenAuthorizationInterceptor();
      Retrofit retrofit = getRetrofit(olaAuthInterceptor);
      return retrofit.create(serviceClass);
  }}
```

OLA DATA FETCHING

```java
public class OlaClient {
  private static final String API_BASE_URL = "https://api.uber.com/v1.2/";
  private static HttpLoggingInterceptor logging = new HttpLoggingInterceptor()
      ;

  private static Retrofit.Builder builder =
      new Retrofit.Builder()
          .baseUrl(API_BASE_URL)
          .addConverterFactory(GsonConverterFactory.create());

  private static Retrofit getRetrofit(Interceptor interceptor) {
      OkHttpClient.Builder httpClientBuilder = new OkHttpClient.Builder();

      if (interceptor != null) {
          httpClientBuilder.addInterceptor(interceptor);
      }

      if (BuildConfig.DEBUG) {
          logging.setLevel(HttpLoggingInterceptor.Level.BODY);
          httpClientBuilder.addInterceptor(logging);
      }

      Retrofit retrofit = builder
              .client(httpClientBuilder
                      .build())
              .build();

      return retrofit;
  }
  public static <S> S createService(Class<S> serviceClass) {
      Retrofit retrofit = getRetrofit(null);
      return retrofit.create(serviceClass);
  }
  public static <S> S createBasicAppToken(Class<S> serviceClass) {
      UberTokenAuthorizationInterceptor olaAuthInterceptor = new
          UberTokenAuthorizationInterceptor();
      Retrofit retrofit = getRetrofit(olaAuthInterceptor);
      return retrofit.create(serviceClass);
  }}
```

6.4 Main Application Logic Module

This is the code for the main application logic modules.It Specify the integration of the above module into one. it created to keep all the modules in one place and to efficiently work with one another.it is also created to handles error and expect ion by the user while using the app as it is obvious and to protect the app from Crashing.

```
protected void onCreate(Bundle savedInstanceState) {
        supermGoogleApiClient = new GoogleApiClient.Builder(this)
                .enableAutoManage(this, 0, this)
                .addApi(Places.GEO_DATA_API)
                .build();
            ApiRequestHandler.fetchAllOlaRideEstimate(pickLatLng, dropLatLng, ""
                ,new Request.RequestDelegate<OlaRideEstimateResponseData>() {
            @Override
            public void onSuccess(OlaRideEstimateResponseData result) {
                Toast.makeText(MainActivity.this, "Data Loaded.......", Toast.
                   LENGTH_SHORT).show();
                Log.e("Ola Estimate", result.getRideEstimateList().size()+"");
                olaRideEstimateList = result.getRideEstimateList();
                fetchUberEstimates();
            }
            @Override
            public void onError(ErrorResponseData errorResponse) {
            }
        @Override
            public void onFailed(Throwable t) {
            }
        });
        mRideViewPager.beginFakeDrag();
        mSlidingTabLayout.setupWithViewPager(mRideViewPager);
    }
    private void fetchUberEstimates(){
        ApiRequestHandler.fetchAllUberRideEstimate(pickLatLng, dropLatLng, new
            Request.RequestDelegate<UberRideEstimateResponseData>() {
            @Override
            public void onSuccess(UberRideEstimateResponseData result) {
                Log.e("Uber Estimate", result.getRideEstimateList().size()+"");
                uberRideEstimateList = result.getRideEstimateList();
                Log.d("The Uber List", String.valueOf(uberRideEstimateList));
                mRideViewPager.setAdapter(new RidePagerAdapter(
                    getSupportFragmentManager(),
                        MainActivity.this));            }
            @Override
            public void onError(ErrorResponseData errorResponse) { }
            @Override
            public void onFailed(Throwable t) {
}});}
```

REST API HANDLER

REST API HANDLER handles all the query related to REST API.The code shown below give the description on how the REST API is used and how it is integrated to our App.rest api is called again again to provide the data that the user want and so the efficiency of the App is taken care to provide a smooth experience to user.

```java
public class ApiRequestHandler {
  public static OlaRideEstimateRequest fetchAllOlaRideEstimate(LatLng
      pickLatLng, LatLng dropLatLng, String categories, @NonNull Request.
      RequestDelegate<OlaRideEstimateResponseData> requestDelegate){
       OlaRideEstimateRequestData olaRideEstimateRequestData = new
           OlaRideEstimateRequestData();
       olaRideEstimateRequestData.setPickup_lat(String.valueOf(pickLatLng.
           latitude));
       olaRideEstimateRequestData.setPickup_lng(String.valueOf(pickLatLng.
           longitude));
       olaRideEstimateRequestData.setDrop_lat(String.valueOf(dropLatLng.
           latitude));
       olaRideEstimateRequestData.setDrop_lng(String.valueOf(dropLatLng.
           longitude));
       olaRideEstimateRequestData.setCategories(categories);
       OlaRideEstimateRequest request = new OlaRideEstimateRequest(
           olaRideEstimateRequestData, requestDelegate);
       request.execute();
       return request;
  }
  public static UberRideEstimateRequest fetchAllUberRideEstimate(LatLng
      pickLatLng, LatLng dropLatLng, @NonNull Request.RequestDelegate<
      UberRideEstimateResponseData> requestDelegate){

       UberRideEstimateRequestData uberRideEstimateRequestData = new
           UberRideEstimateRequestData();
       uberRideEstimateRequestData.setStart_latitude(String.valueOf(pickLatLng.
           latitude));
       uberRideEstimateRequestData.setStart_longitude(String.valueOf(pickLatLng
           .longitude));
       uberRideEstimateRequestData.setEnd_latitude(String.valueOf(dropLatLng.
           latitude));
       uberRideEstimateRequestData.setEnd_longitude(String.valueOf(dropLatLng.
           longitude));
       UberRideEstimateRequest request = new UberRideEstimateRequest(
           uberRideEstimateRequestData, requestDelegate);
       request.execute();
       return request;
  }}
```

Chapter 7

System Testing

In This Chapter the System is being tested to find out the accuracy of the system.The Tested result is shown in the Table and the Result image is shown below.Testing is used to find the error rate and to find the loop holes in the System.It gives the clear idea about the working of the system and the problems in the System.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Login	Should be register user	Will connect to firebase	Should login into system
T02	Maps	should accept location	Will connect to Google API	Calculate the route
T03	Data Fetching	Correct data	REST API call	Display of compare data

7.2 Sample of a Test Case

Title: Login Page – Authenticate Successfully on Google
Description: A registered user should be able to successfully login on APP.

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

Assumption: a supported App is being used.

Test Steps:

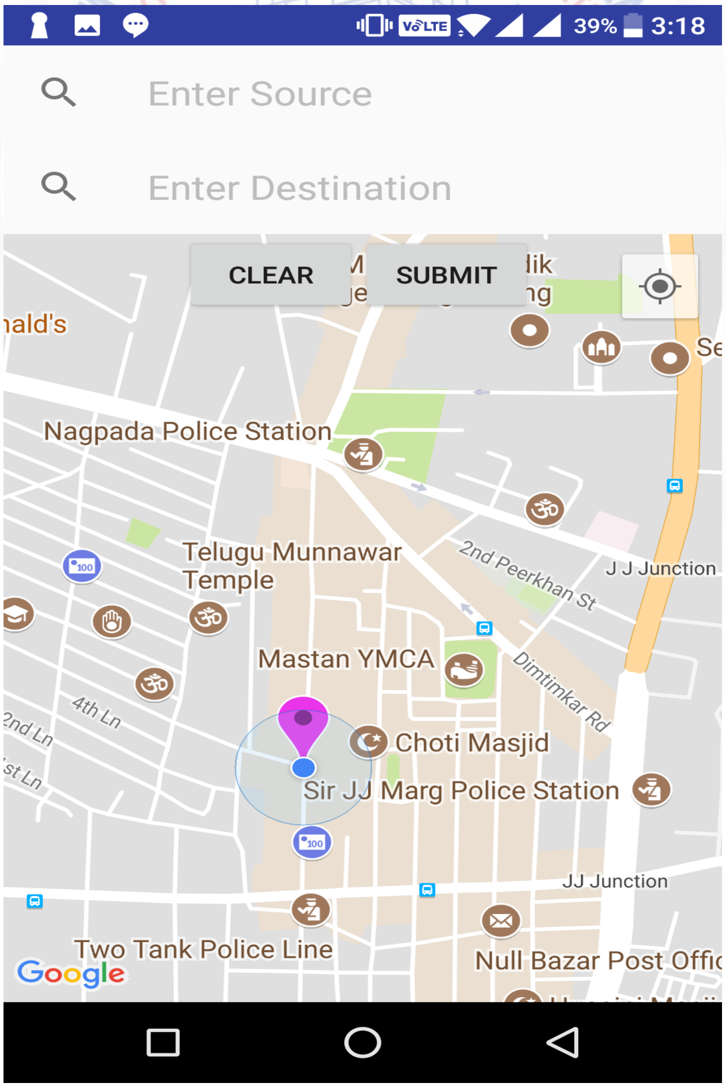
1. Navigate to App and click the Google Button

2. Or In the 'email' field, enter the email
3. In 'password' field enter password of the registered user.
4. Click submit
5. and if Not register click Sign Up link for registration.

Expected Result: A page displaying the maps with source and destination should load, showing the current location on the App.

Actual Result:

when Success-full login on the App through Google or firebase Login A new Page Should be Open to the User Displaying maps And providing flied for User to enter Source and Destination.
The image of result:



Title: Maps Page– Successfully shows your current location.
Description: A registered user should be able to see his current location using GPS as a Pointer.

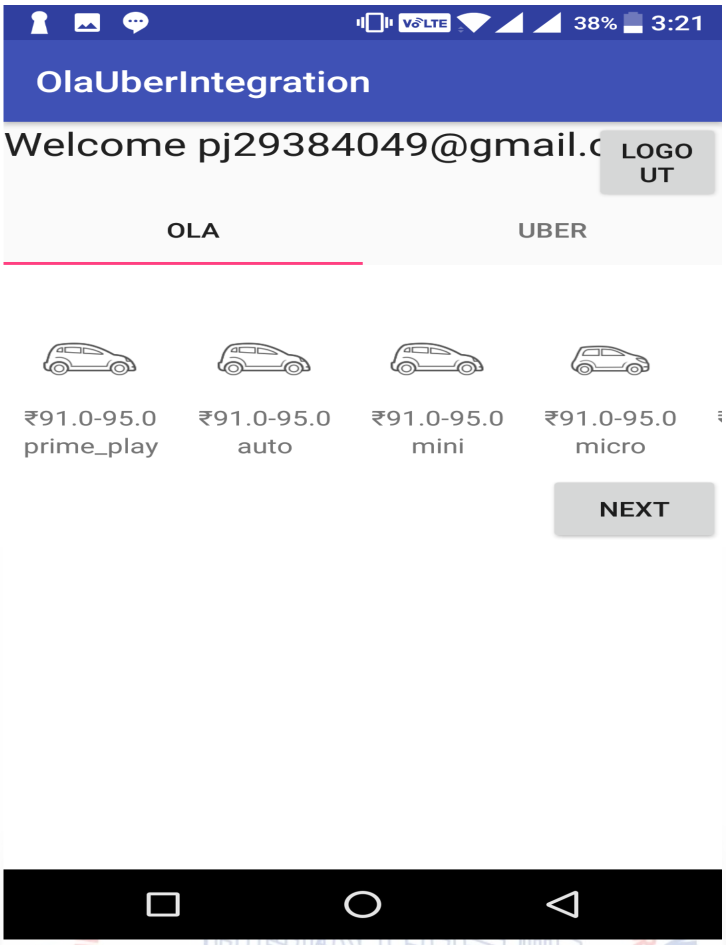
Precondition: the user must already be registered with an email address and password.

Assumption: a supported App is being used.

Test Steps:

1. User should allow its GPS to be open.
2. After showing location user should enter source and location.
3. Then it will show the cab details with compared rates.
4. Click submit and book a cab.

Expected Result: After putting source and destination in the fields the data of the cabs will be shown.
Actual Result: After putting source and destination in the fields the data of the cabs will be shown.



7.2.1 Software Quality Attributes

1. AVAILABILITY: The system should not be down,whenever the user use the system the specific data should be available to the user.
2. CORRECTNESS: As per the user search the correct data should be shown to the user like at time for searching the near by place the system should show only the places around the user.
3. MAINTAINABILITY: The administrators of the system will maintain the system with effective updates though on air update if needed.
4. Extensibility: The system is capable to be modified by changing some modules or by adding some features to the existing system

Chapter 8

Screenshots of Project

8.1 Google And Firebase Login

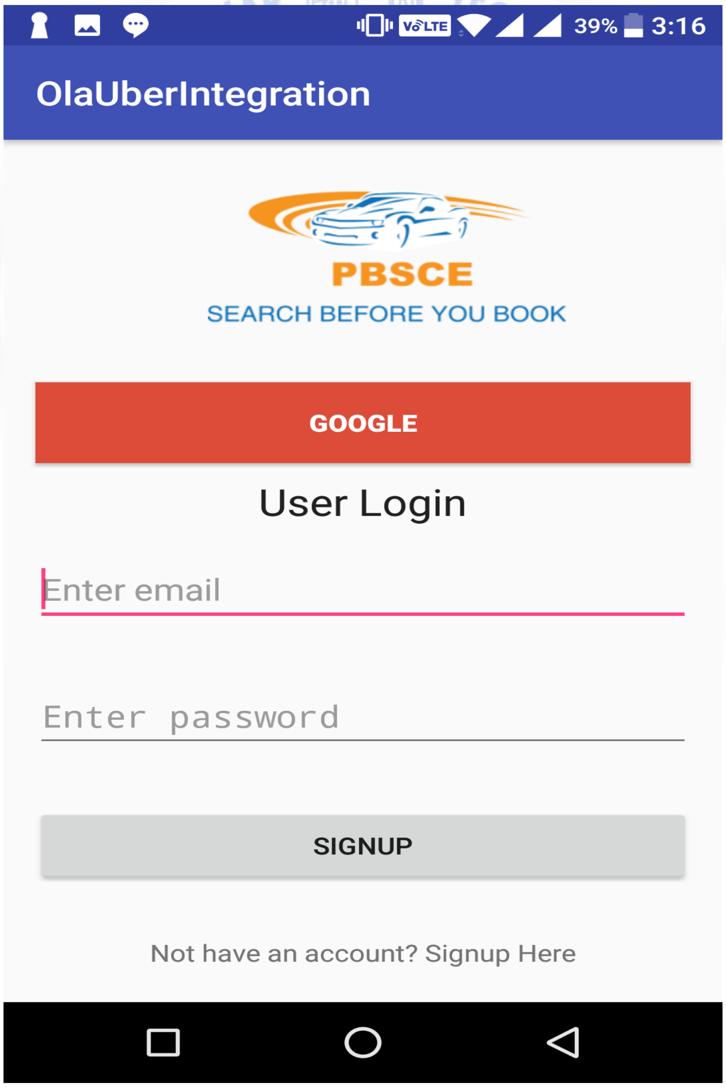


Figure 8.1: Screenshot for google and firebase login 1

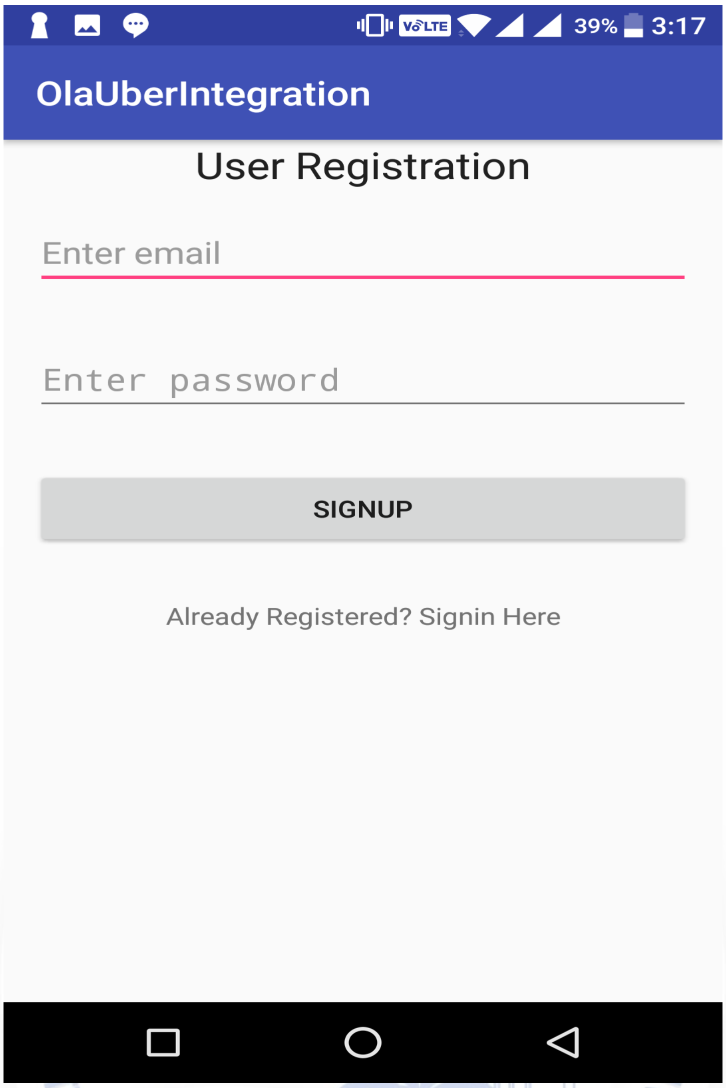
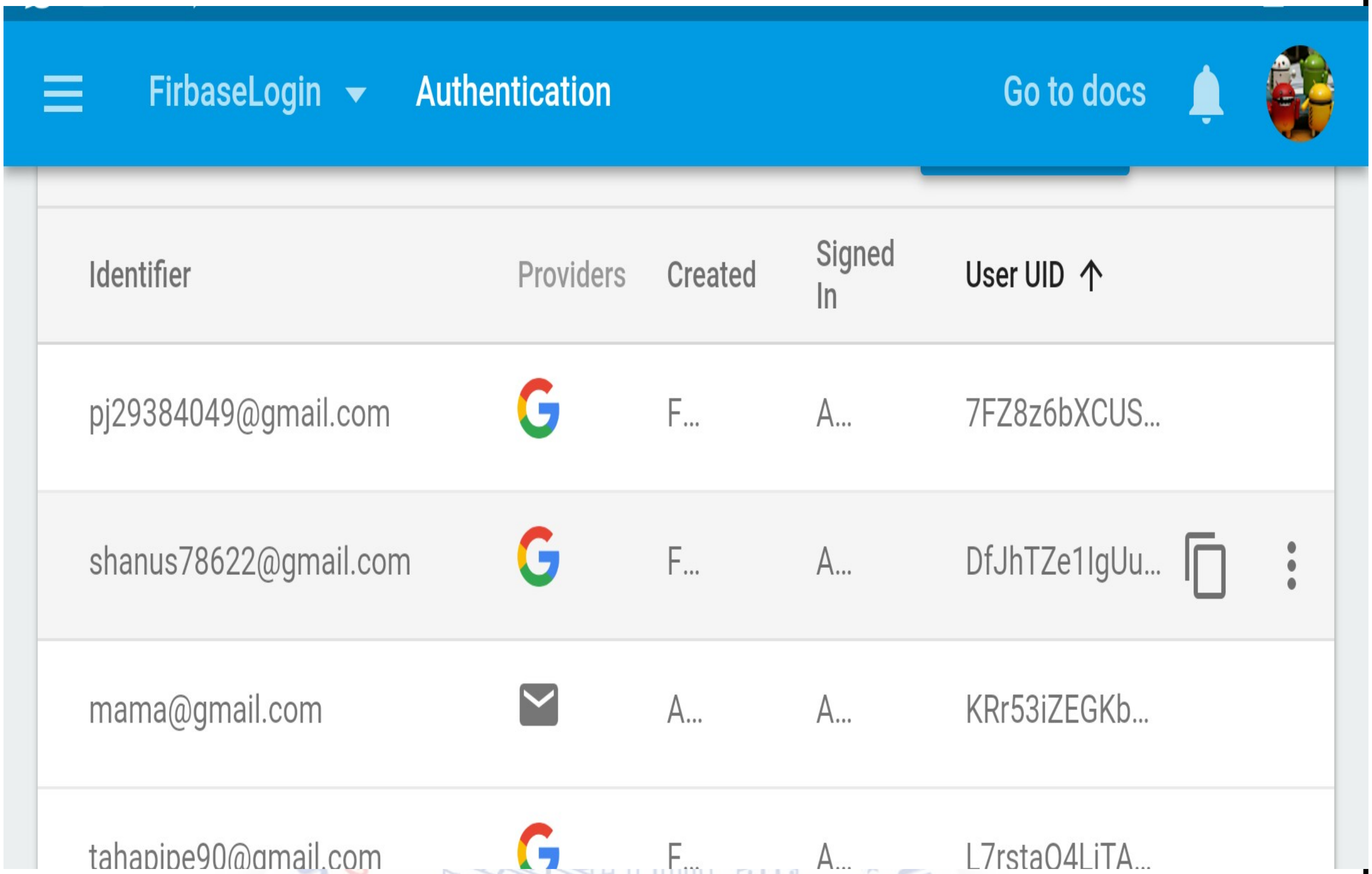


Figure 8.2: Screenshot for google and firebase login 2



Identifier	Providers	Created	Signed In	User UID ↑
pj29384049@gmail.com	G	F...	A...	7FZ8z6bXCUS...
shanus78622@gmail.com	G	F...	A...	DfJhTZe1IgUu...
mama@gmail.com	✉	A...	A...	KRr53iZEGKb...
tahapipe90@gmail.com	G	F...	A...	L7rstaO4LiTA...

Figure 8.3: Screenshot for google and firebase login 3

8.2 Working Of Maps

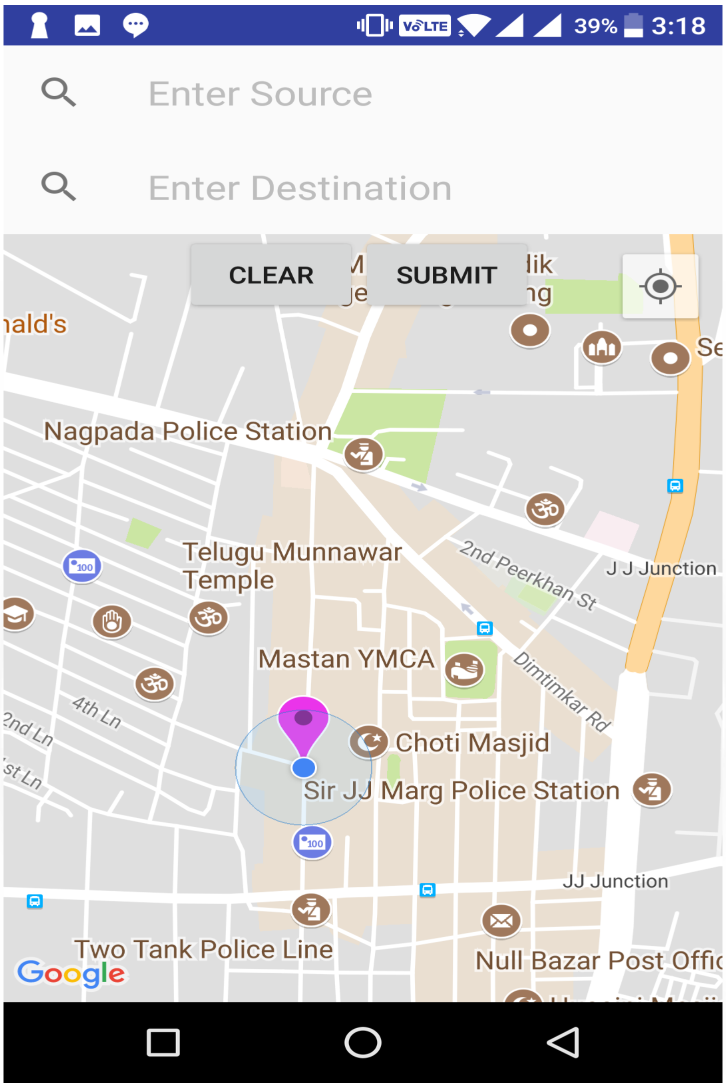


Figure 8.4: Screenshot for working of map 1

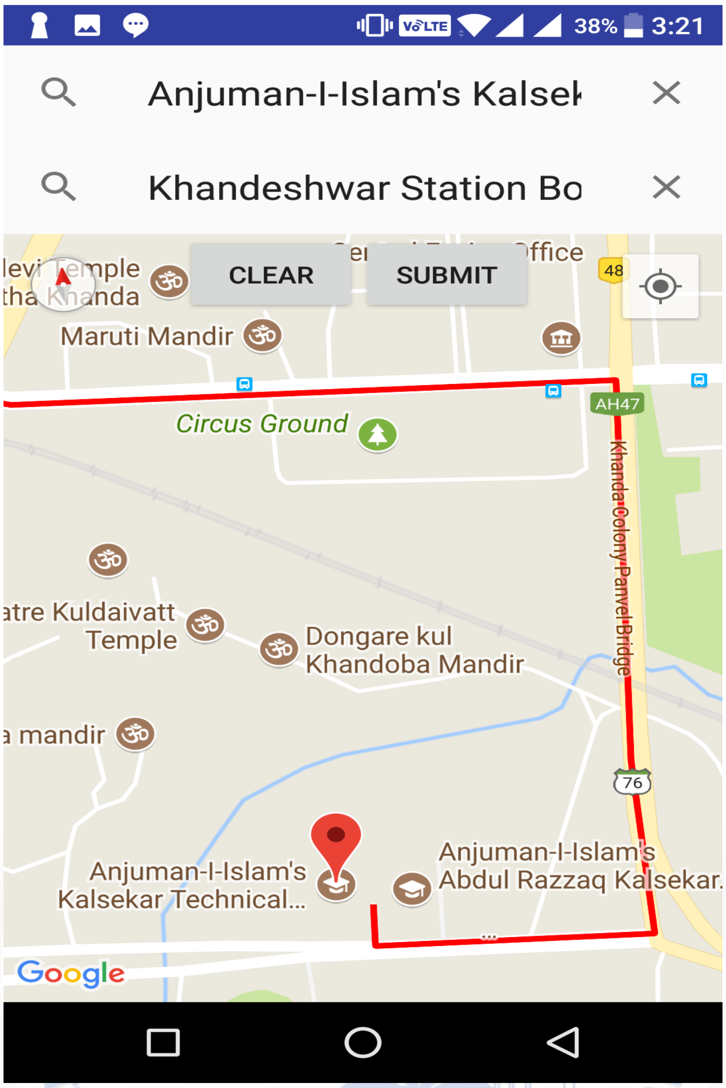

Anjuman-I-Islam's Kalsek
Khandeshwar Station Bo
CLEAR
SUBMIT
Maruti Mandir
Circus Ground
Dongare kul Khandoba Mandir
Anjuman-I-Islam's Kalsekar Technical...
Anjuman-I-Islam's Abdul Razzaq Kalsekar
Khanda Colony Panvel Bridge

Figure 8.5: Screenshot for working of map 2

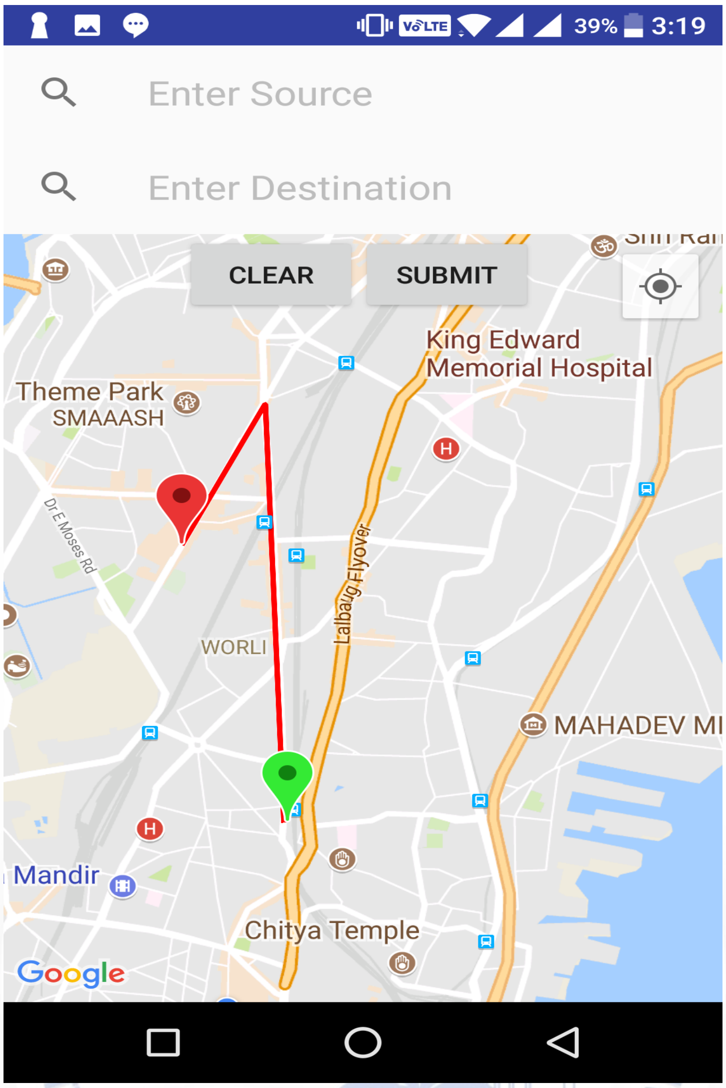
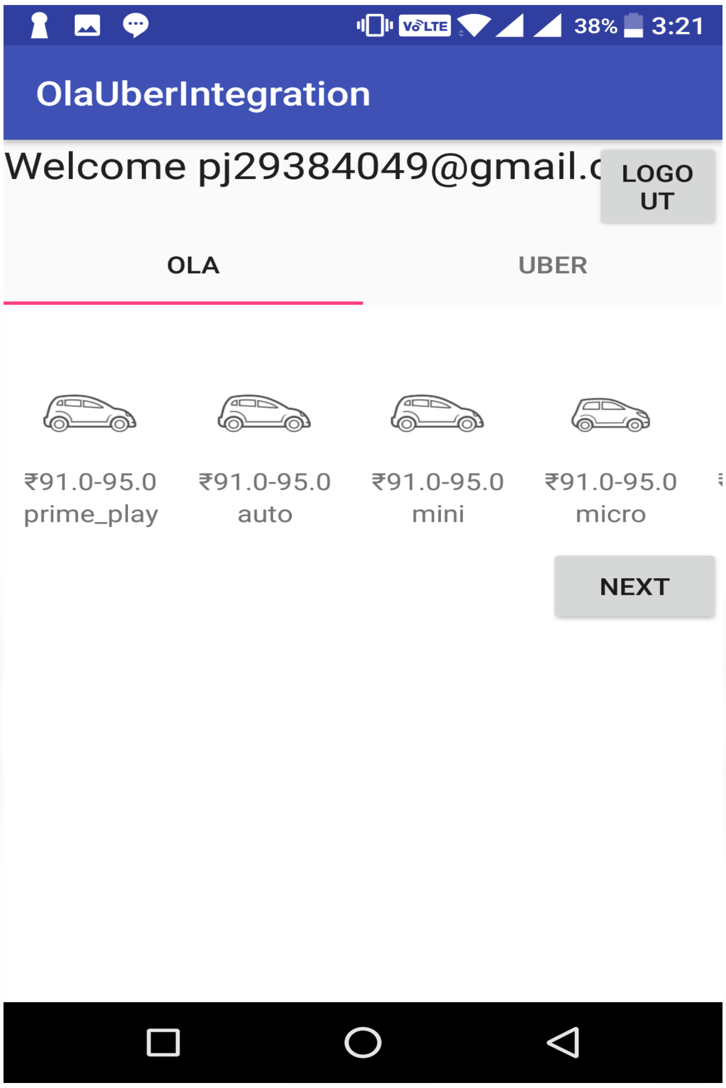


Figure 8.6: Screenshot for working of map 3

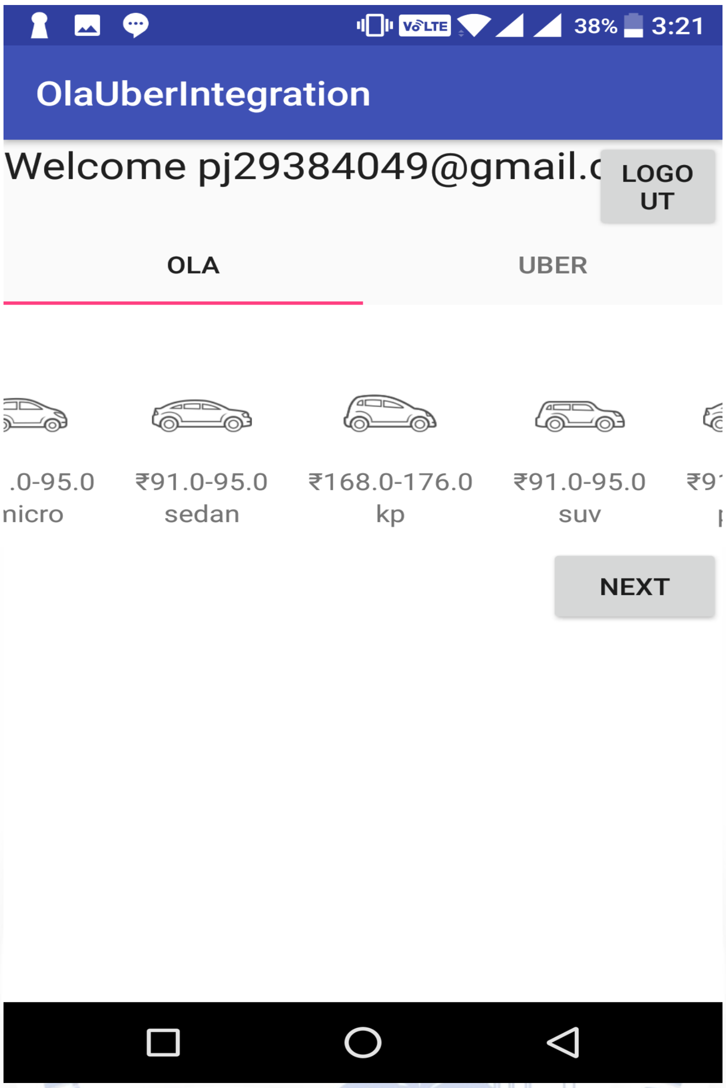
8.3 Data From Ola And Uber


OlaUberIntegration

Welcome pj29384049@gmail.c

₹91.0-95.0 prime_play
₹91.0-95.0 auto
₹91.0-95.0 mini
₹91.0-95.0 micro

Figure 8.7: Screenshot for data fetch from ola 1


OlaUberIntegration

Welcome pj29384049@gmail.c

.0-95.0 nicro

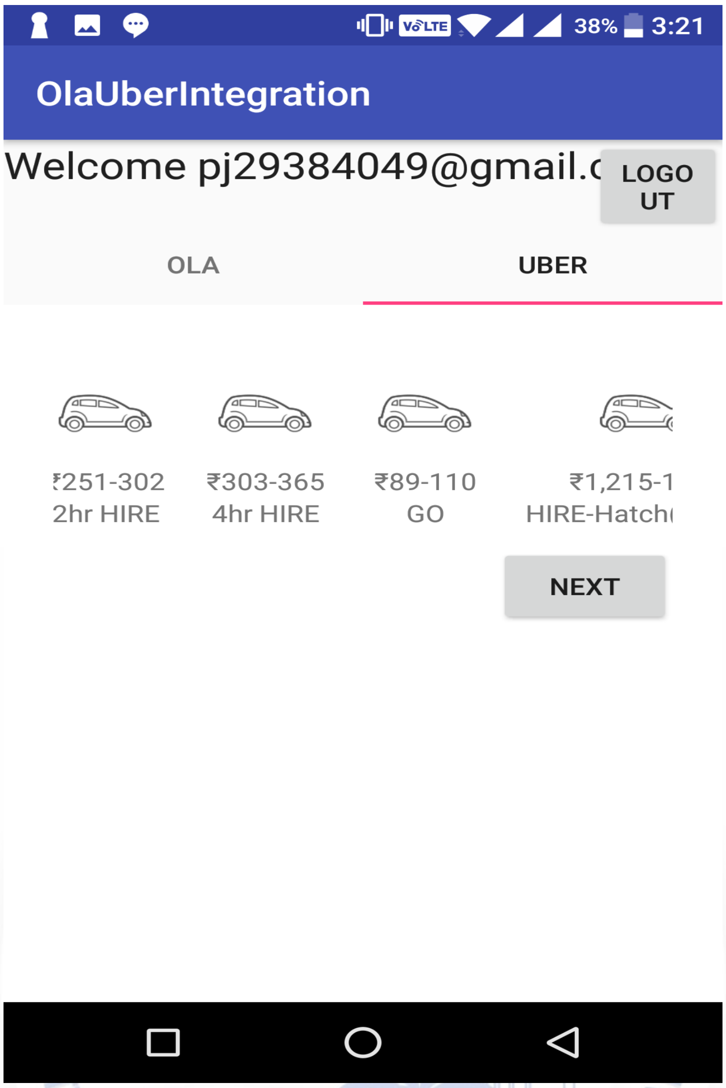
₹91.0-95.0 sedan

₹168.0-176.0 kp

₹91.0-95.0 suv

₹9

Figure 8.8: Screenshot for data fetch from ola 2


3:21 38%

OlaUberIntegration

Welcome pj29384049@gmail.c

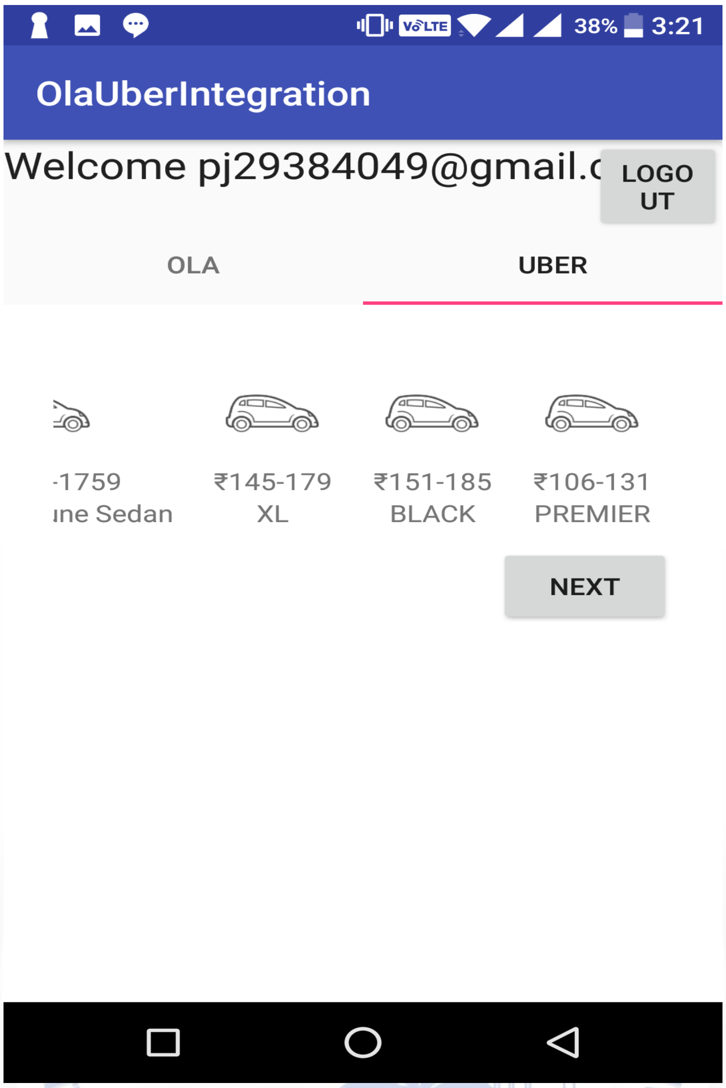
₹251-302
2hr HIRE

₹303-365
4hr HIRE

₹89-110
GO

₹1,215-1
HIRE-Hatch

Figure 8.9: Screenshot for data fecth from uber 1


OlaUberIntegration

Welcome pj29384049@gmail.c

Figure 8.10: Screenshot for data fetch from uber 2

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

We have selected this topic because as the Market of online cab business is booming and theneed of the customer who is willing to go from one place to another considering the factor oftime and money and these application integrate all these functions in one and keep them united.

9.2 Future Scope

This project has a wide scope.

- This would prove a Major breakthrough in reducing stress of user for booking the cab
- The app will provide a great experience for user and also for individual driver or employee of OLA and uber
- In future we will provide services of MERU,KAALI-PEELI and ETC, to this app thus creating a multi-Optional App.

References

[1] Ola documentation link `https://developers.olacabs`

[2] Uber Documentation link `https://developer.uber.com`

[3] Study of Rest API 2017 IEEE 3rd International Conference on Big Data Security on Clouding.

[4] The Comparison of Travel Patterns between Taxi and Private Car at Beijing Capital International Airpobhart Area Jiyuan Tan, Yibin Huang, Li Wang,Weiwei Guo 1 , Luxi Dong 1 , Jian

[5] Best Price Algorithm in Finding routes based On Unconventional Public Transportations: Indonesian Suburban Regions. By indra kuntara, muhammed ridha ,SIUS.

Achievements

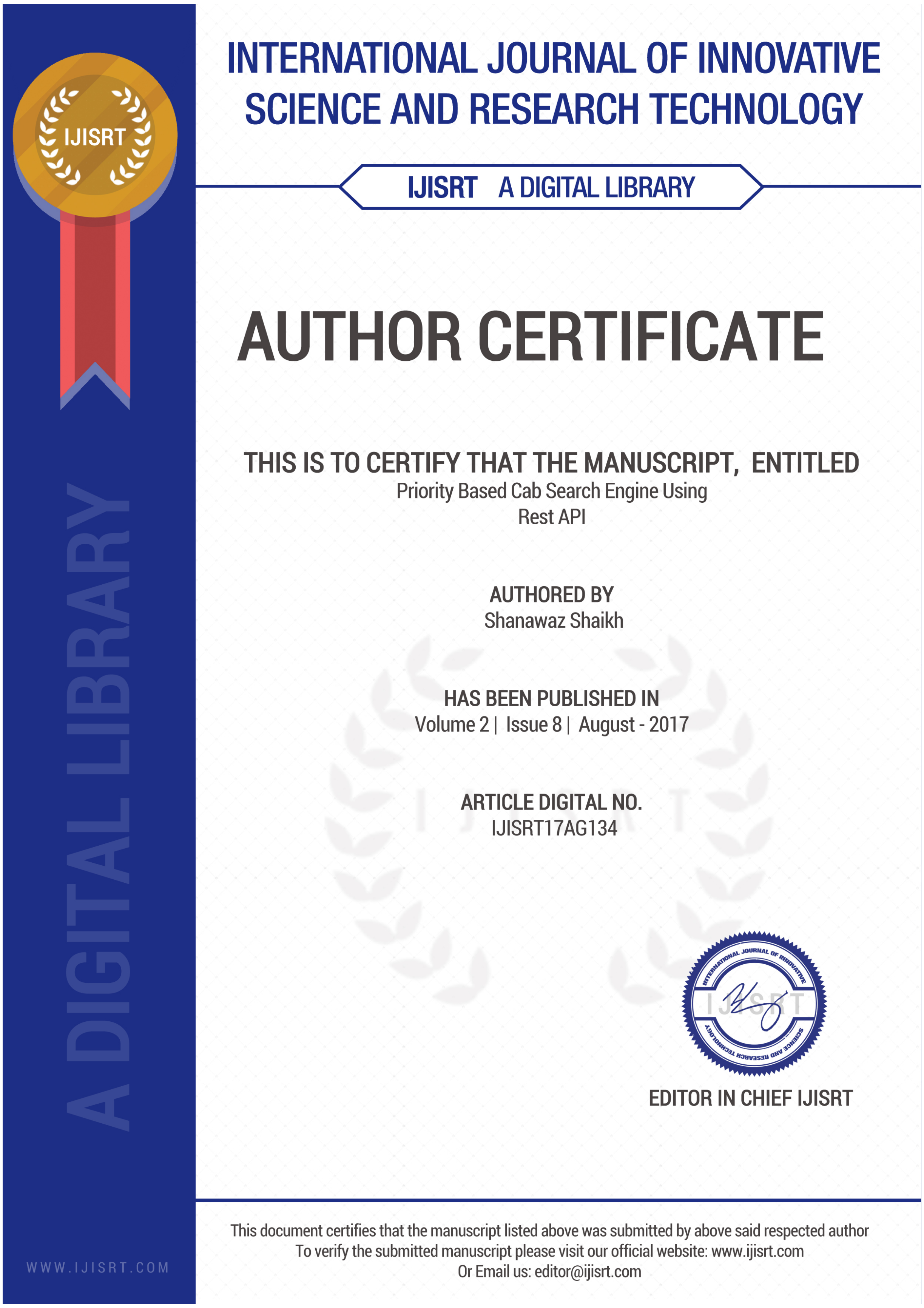
1. Publications

    (a) *Priority Based Cab Search Engine Using REST API*: Quid Zohar Morbiwala,Shanawaz Shaikh,Taha Pipewala,Prof.Javed Khan Sheikh,International Journal Of Innovative Science and Research Technology,August-2017 (*https://ijisrt.com/priority-based-cab-search-engine-using-rest-api*)

    (b) *Implementation Of Optimized Cab Search*: Quid Zohar Morbiwala,Shanawaz Shaikh,Taha Pipewala, Prof.Javed Khan Sheikh,International Journal Of Innovative Science and Research Technology,August-2018 (*https://ijisrt.com/priority-based-cab-search-engine-using-rest-api*)

2. Project Competitions

    (a) *Priority Based Cab Search Engine*: Quid Zohar Morbiwala, Shanawaz Shaikh,Taha Pipewala,TARTRAGYAN 2K18,march-2018(Venue :Lokmaniya Tilak college koparkhairane)


INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

IJISRT

A DIGITAL LIBRARY

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Priority Based Cab Search Engine Using
Rest API

AUTHORED BY
Shanawaz Shaikh

HAS BEEN PUBLISHED IN
Volume 2 | Issue 8 | August - 2017

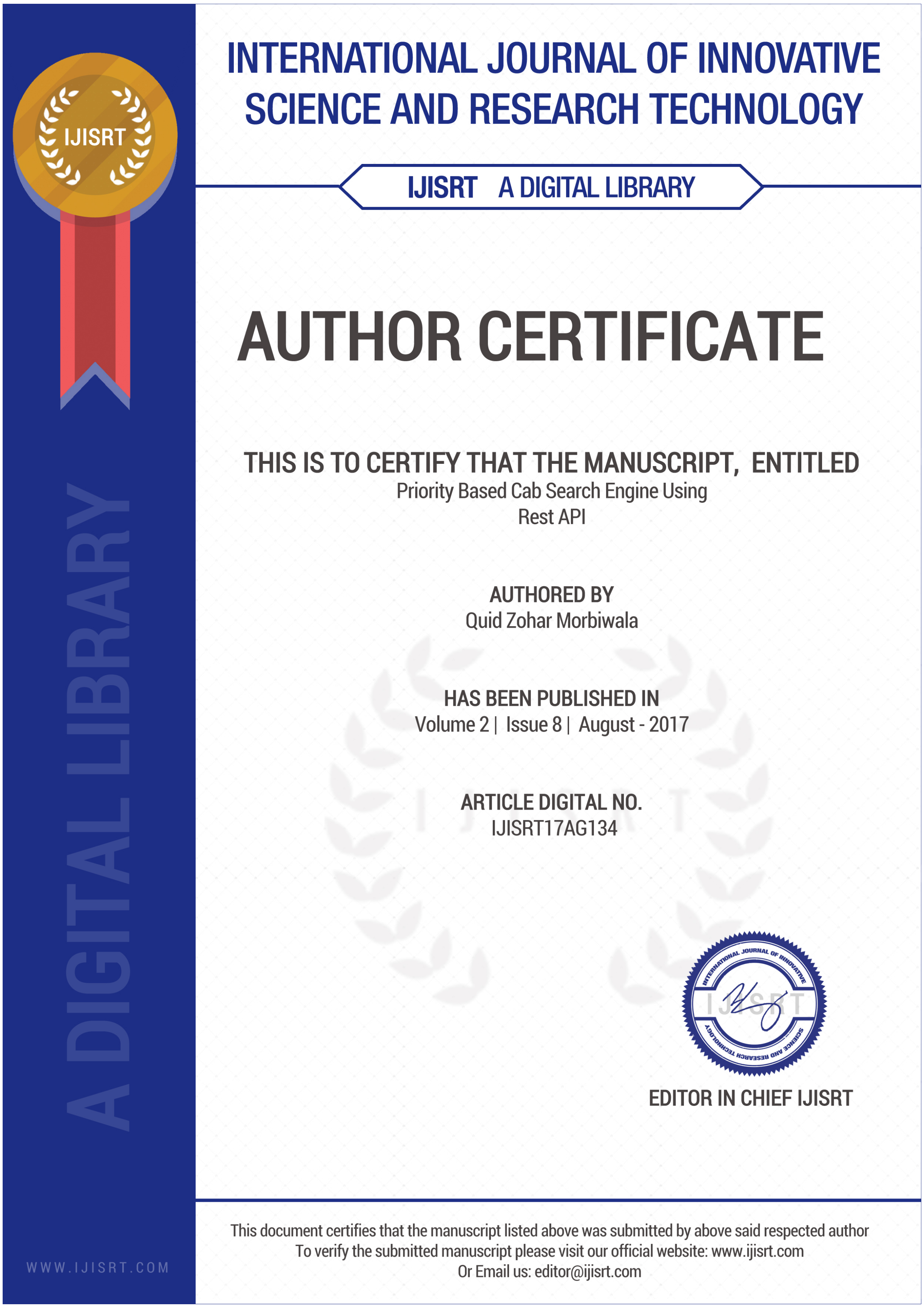
ARTICLE DIGITAL NO.
IJISRT17AG134

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY
IJISRT

EDITOR IN CHIEF IJISRT

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

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Priority Based Cab Search Engine Using
Rest API

AUTHORED BY
Quid Zohar Morbiwala

HAS BEEN PUBLISHED IN
Volume 2 | Issue 8 | August - 2017

ARTICLE DIGITAL NO.
IJISRT17AG134

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY IJISRT

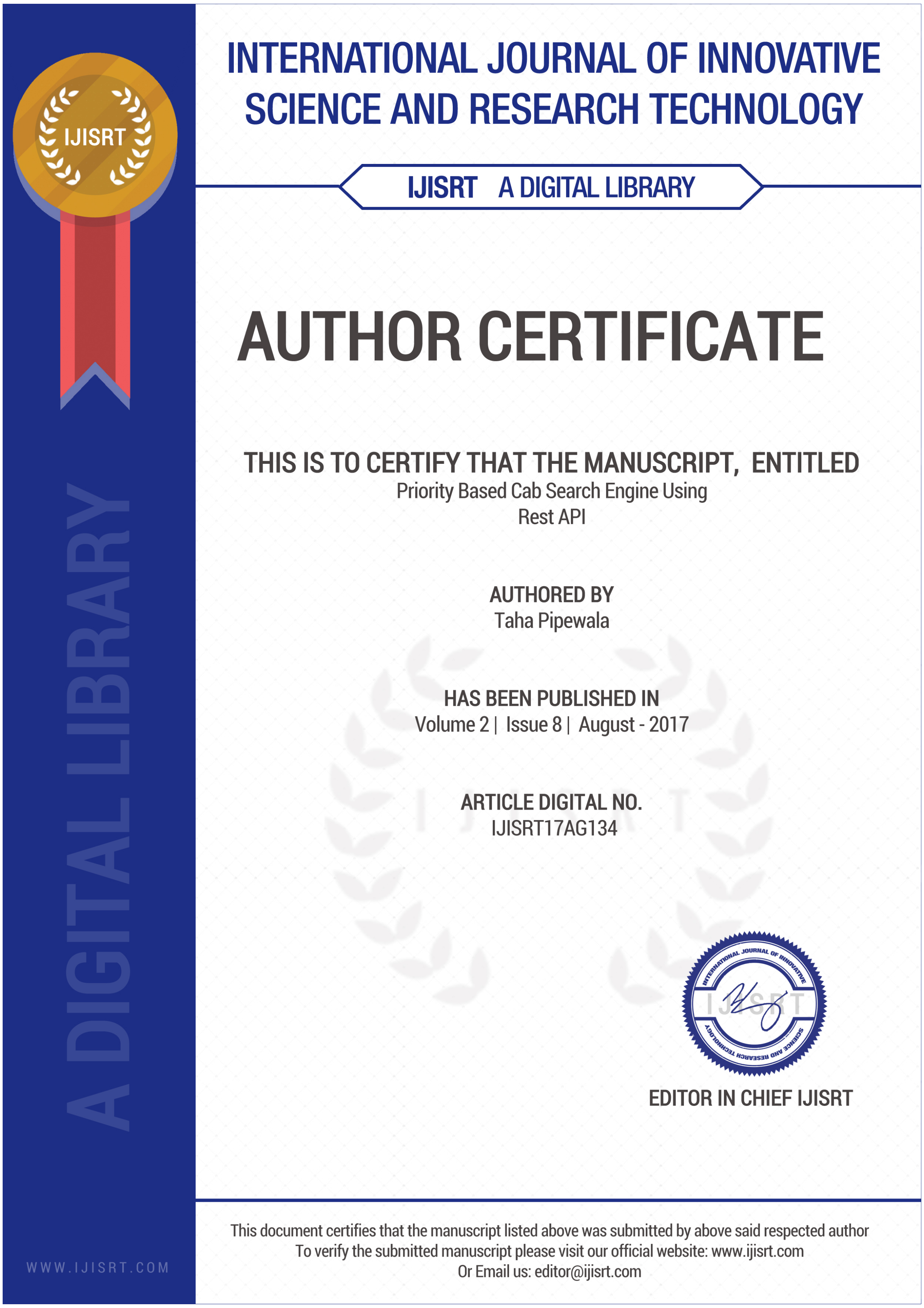
EDITOR IN CHIEF IJISRT

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

IJISRT

A DIGITAL LIBRARY

WWW.IJISRT.COM



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Priority Based Cab Search Engine Using
Rest API

AUTHORED BY
Taha Pipewala

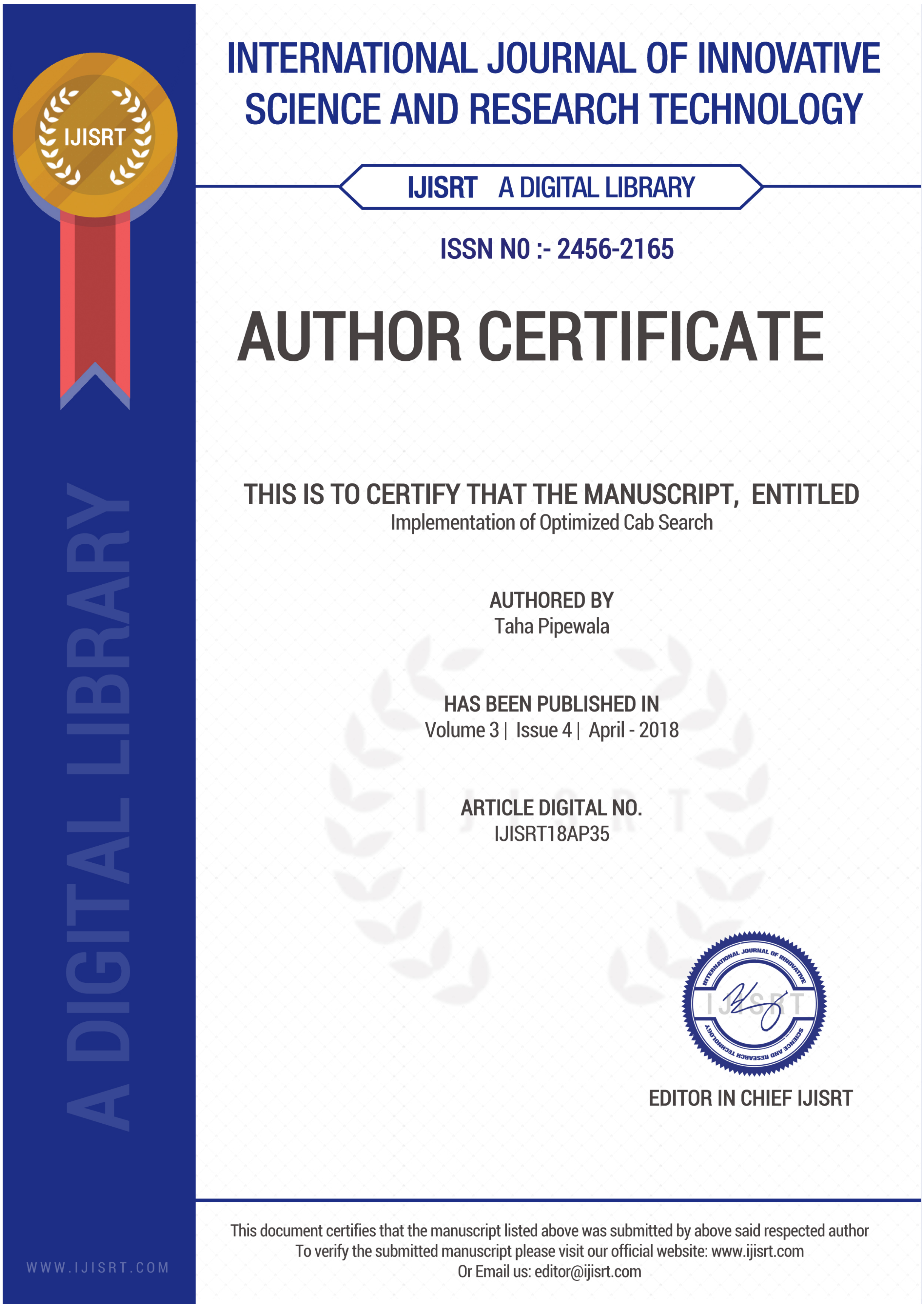
HAS BEEN PUBLISHED IN
Volume 2 | Issue 8 | August - 2017

ARTICLE DIGITAL NO.
IJISRT17AG134

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY
IJISRT

EDITOR IN CHIEF IJISRT

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
Implementation of Optimized Cab Search

AUTHORED BY
Taha Pipewala

HAS BEEN PUBLISHED IN
Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.
IJISRT18AP35

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY IJISRT

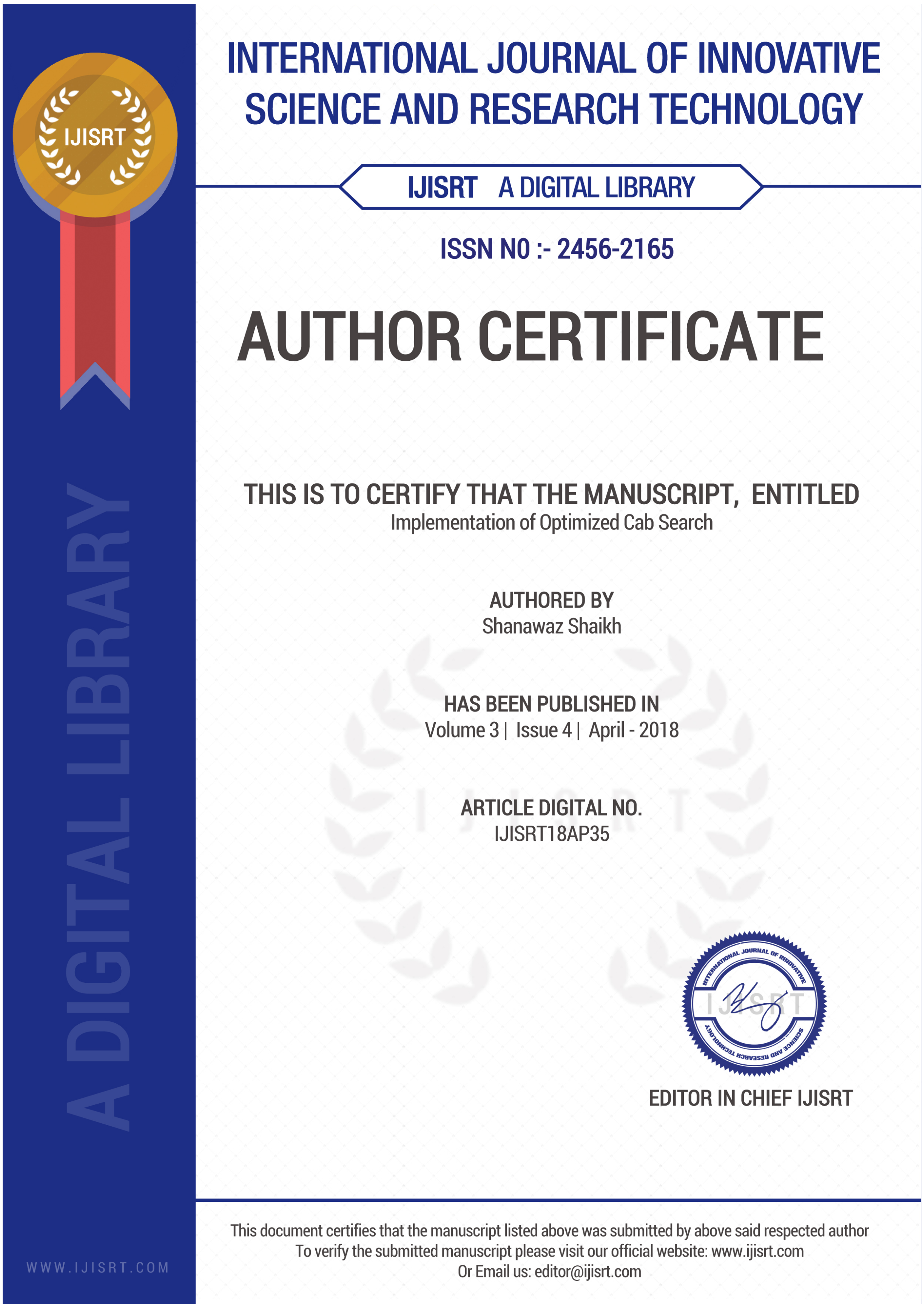
EDITOR IN CHIEF IJISRT

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

IJISRT

A DIGITAL LIBRARY

WWW.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
Implementation of Optimized Cab Search

AUTHORED BY
Shanawaz Shaikh

HAS BEEN PUBLISHED IN
Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.
IJISRT18AP35

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY IJISRT

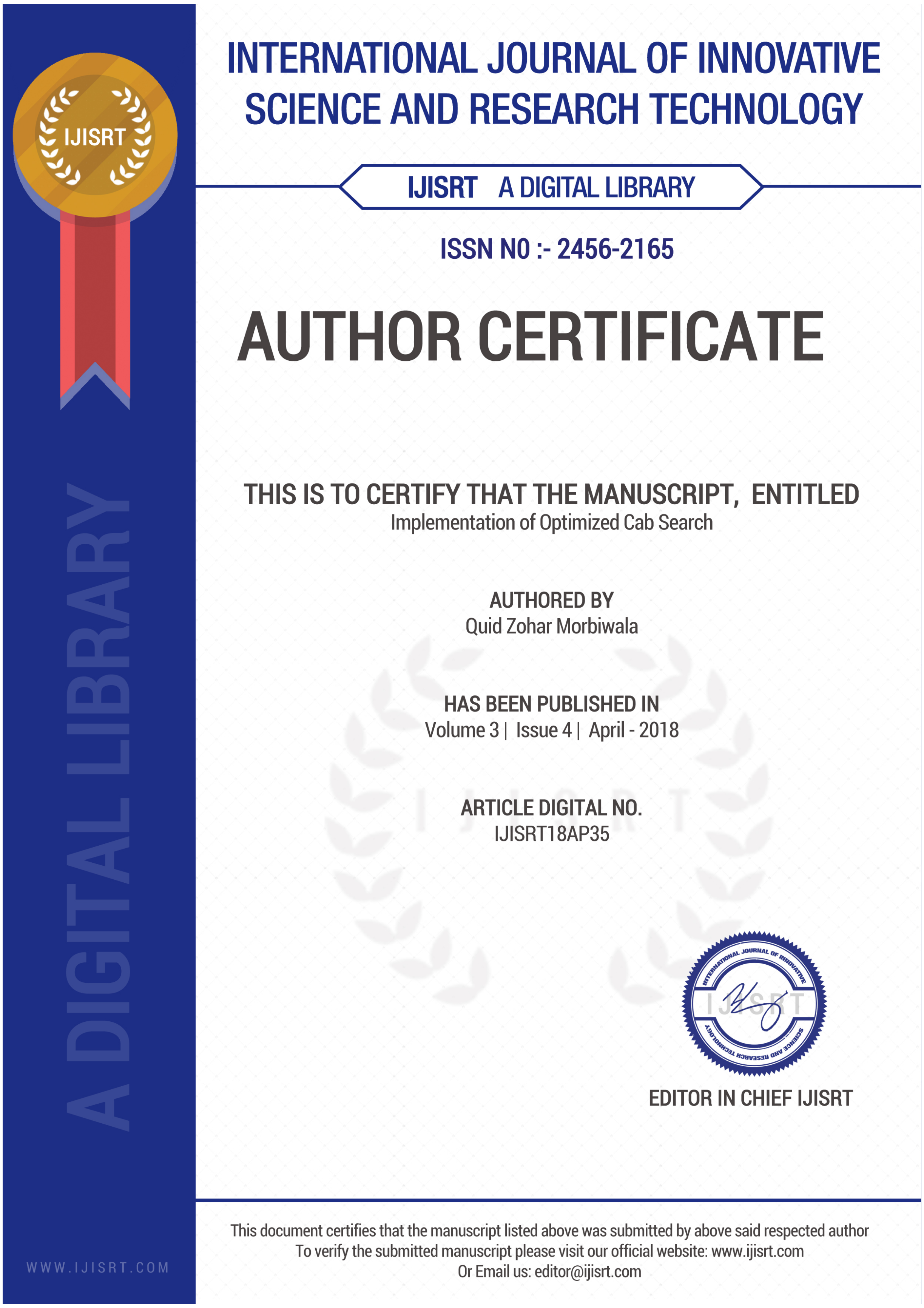
[illegible]

EDITOR IN CHIEF IJISRT

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

A DIGITAL LIBRARY

WWW.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
Implementation of Optimized Cab Search

AUTHORED BY
Quid Zohar Morbiwala

HAS BEEN PUBLISHED IN
Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.
IJISRT18AP35

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY IJISRT

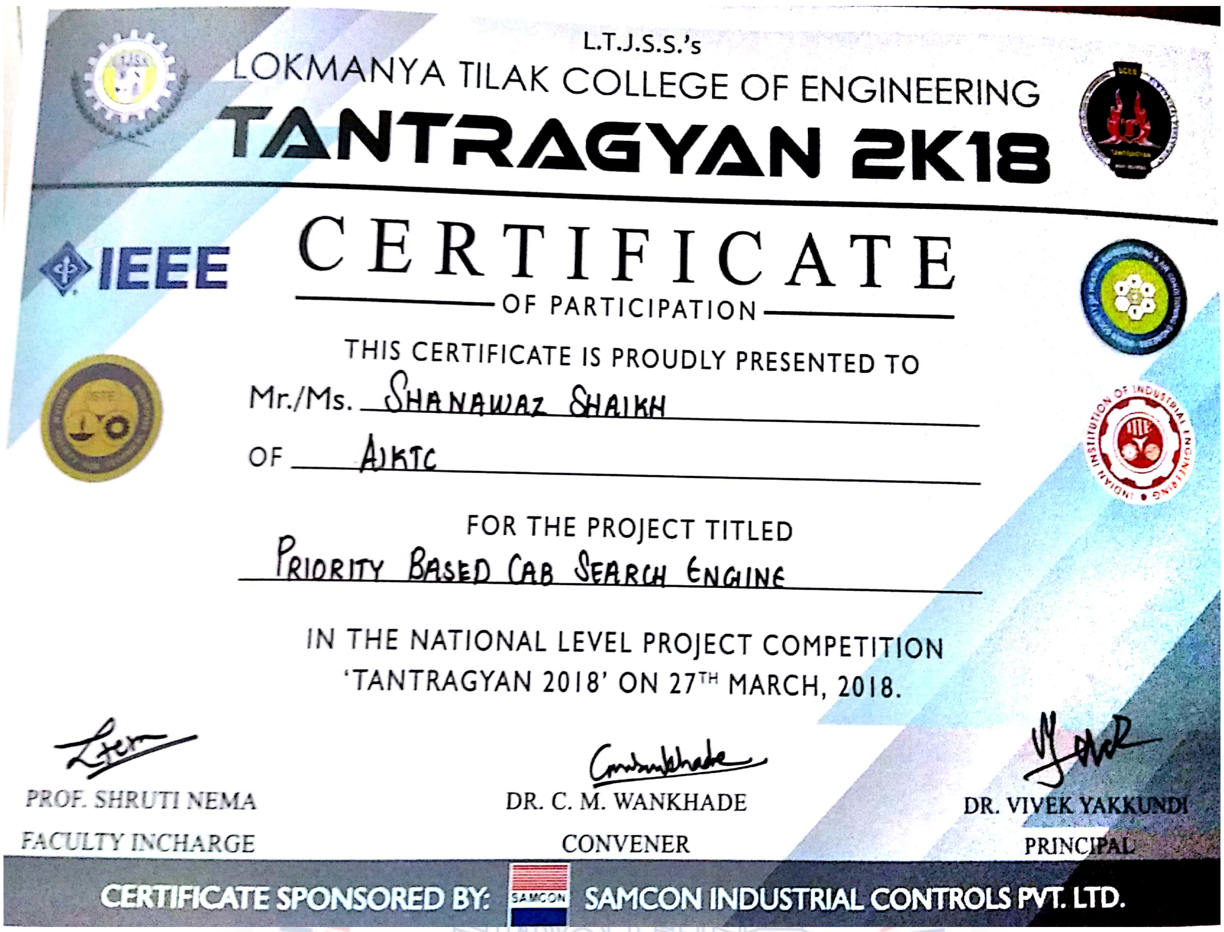
[illegible]

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



L.T.J.S.S.'s
LOKMANYA TILAK COLLEGE OF ENGINEERING
TANTRAGYAN 2K18

CERTIFICATE
OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO
Mr./Ms. SHANAWAZ SHAIKH
OF AIKTC

FOR THE PROJECT TITLED
PRIORITY BASED CAB SEARCH ENGINE

IN THE NATIONAL LEVEL PROJECT COMPETITION
'TANTRAGYAN 2018' ON 27TH MARCH, 2018.

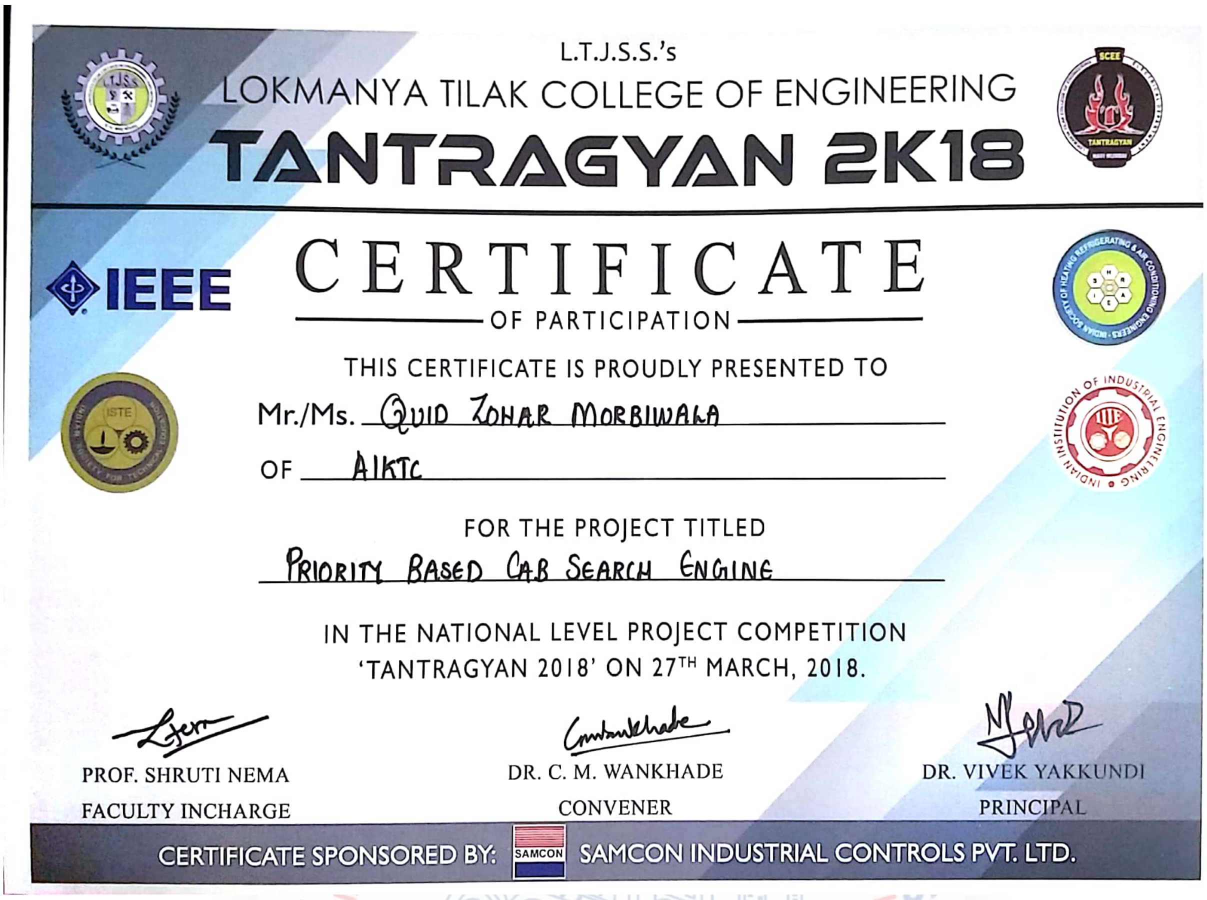
[illegible] [illegible] [illegible]

PROF. SHRUTI NEMA
FACULTY INCHARGE

DR. C. M. WANKHADE
CONVENER

DR. VIVEK YAKKUNDI
PRINCIPAL

CERTIFICATE SPONSORED BY: SAMCON INDUSTRIAL CONTROLS PVT. LTD.



L.T.J.S.S.'s
LOKMANYA TILAK COLLEGE OF ENGINEERING
TANTRAGYAN 2K18

CERTIFICATE
OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO
Mr./Ms. QUID ZOHAR MORBIWALA
OF AIKTC

FOR THE PROJECT TITLED
PRIORITY BASED CAB SEARCH ENGINE

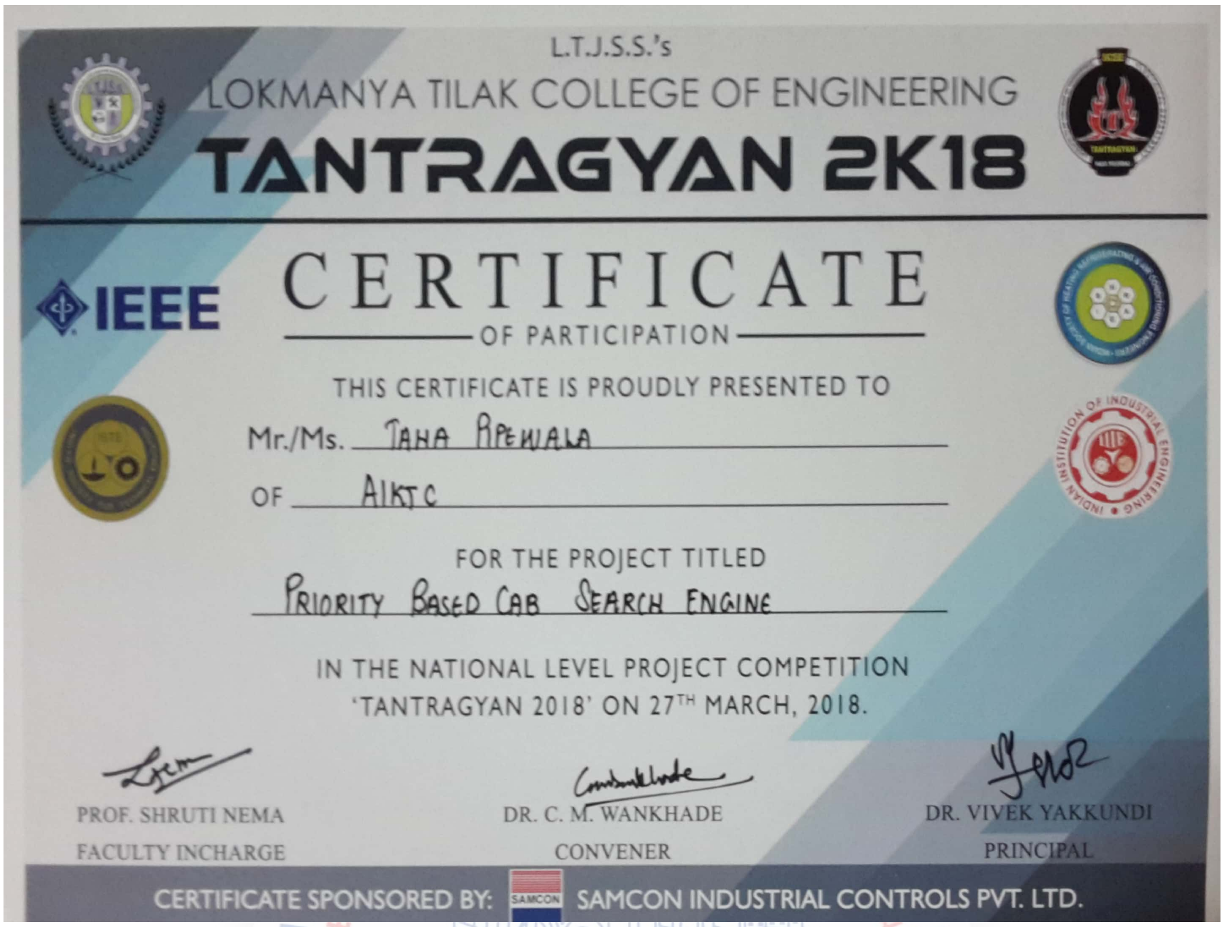
IN THE NATIONAL LEVEL PROJECT COMPETITION
'TANTRAGYAN 2018' ON 27TH MARCH, 2018.

[illegible]
PROF. SHRUTI NEMA
FACULTY INCHARGE

[illegible]
DR. C. M. WANKHADE
CONVENER

[illegible]
DR. VIVEK YAKKUNDI
PRINCIPAL

CERTIFICATE SPONSORED BY: SAMCON SAMCON INDUSTRIAL CONTROLS PVT. LTD.



L.T.J.S.S.'s
LOKMANYA TILAK COLLEGE OF ENGINEERING
TANTRAGYAN 2K18

CERTIFICATE
OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO
Mr./Ms. TAHA APEWALA
OF AIKTC
FOR THE PROJECT TITLED
PRIORITY BASED CAB SEARCH ENGINE
IN THE NATIONAL LEVEL PROJECT COMPETITION
'TANTRAGYAN 2018' ON 27TH MARCH, 2018.

[illegible] [illegible] [illegible]
PROF. SHRUTI NEMA DR. C. M. WANKHADE DR. VIVEK YAKKUNDI
FACULTY INCHARGE CONVENER PRINCIPAL

CERTIFICATE SPONSORED BY: SAMCON INDUSTRIAL CONTROLS PVT. LTD.