**Tribhuvan University**

Kathford International College of Engineering and Management

**A Final Project Report**

**On**

**“Ride Finder App”**

**Submitted To:**

**Department of Information Technology**

**Kathford International College of Engineering and Management**

**In partial fulfilment of the requirement for the Bachelor Degree in Computer Science and Information Technology**

**Submitted By:**

**Bivek Khatiwada (15216/074)**

**Ranjeeb Dahal(15235/074)**

**Samir Waiba (15240/074)**

**April 2022**

https://lh4.googleusercontent.com/4OimCUoYLaqsdS0HtgpVx7Lkizow_5Pj3UI2gsuPNE58xajhZZCjIfholdpEQbWPo3mnBT3_gTzBR20JPd54isb84bsqGyrFeykWZ8gGS4N2ipGsZFYR37Ar70Zb9E5I4lu74Qnn

**SUPERVISOR’S RECOMMENDATION**

I hereby recommend that this report has been prepared under my supervision by *Bivek Khatiwada(15216/074), Ranjeeb Dahal(15235/074)* and *Samir Waiba(15240/074)* entitled “Ride Finder App” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

..........................

Hari Prasad Pokharel

Supervisor

Department of Information Technology

Kathford International College of Engineering and Management

Balkumari, Lalitpur

https://lh4.googleusercontent.com/4OimCUoYLaqsdS0HtgpVx7Lkizow_5Pj3UI2gsuPNE58xajhZZCjIfholdpEQbWPo3mnBT3_gTzBR20JPd54isb84bsqGyrFeykWZ8gGS4N2ipGsZFYR37Ar70Zb9E5I4lu74Qnn

**CERTIFIATE OF APPROVAL**

This is to certify that this project prepared by *Bivek Khatiwada (15216/074) and Ranjeeb Dahal (15235/074)* *Samir Waiba (15240/074)*, entitled Ride Finder App in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Hari Prasad Pokharel**  Supervisor  Kathford International College of Engineering and Management  Balkumari, Lalitpur | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Dipendra Panta**  Deputy Head of Department  Department of Information Technology  Kathford International College of  Engineering and Management  Balkumari, Lalitpur |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (External Examiner)  Central Department of Information Technology  Tribhuvan University  Kirtipur, Nepal | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Internal Examiner)  Department Information Technology  Kathford International College of  Engineering and Management  Balkumari, Lalitpur |

# ACKNOWLEDGEMENT

We would like to take this opportunity to thank the department of computer science and information technology, Kathford International College of Engineering and Management for allowing us to use technology to explore our potential in the sector of our interest. We'd like to thank **Mr.Hari Prasad Pokharel** our project supervisor and **Mr. Mukesh Prasad Chaudhary** project coordinator, for advising us on widening the scope of our project and motivating us to think more about the project's viability.

We'd also like to express our gratitude to everyone who has helped and encouraged us with this endeavor.

# ABSTRACT

Implementation of a real-time ride finder system is proposed and developed to keep monitoring vehicles and to provide the information of vehicles for sharing a ride to the travelers via android application (app) in a more efficient manner. The idea behind this is to provide a quality ride sharing service in every crowded places specially in Kathmandu. Moreover, some limitations arise with the present existing ride sharing platforms & tracking systems in the universe-need hardware maintenance in the GSM/GPRS module that is expensive also, use conventional MySQL database that updates data within few seconds etc. Concerning these limitations, the developed system uses two smartphone applications for transmitting and receiving information using Firebase, an online server with real-time database updating data within few milliseconds that makes the system faster than others. The In-vehicle app receives latitude and longitude values from the satellite and transmits them to the firebase. Firebase updates the data continuously and transmits to the user app. A user can get the information of current location address, distance from any vehicle and time to reach a specific destination with the vehicle through the User app. Hence, the developed ride finder system provides more economical, velocious and user-friendly environment to the users. The users can check and book/reserve the seat in the vehicle.

***Keywords:* Tracking, Firebase, GSM/GPRS.**

Table of Contents

[ACKNOWLEDGEMENT i](#_Toc99919521)

[ABSTRACT ii](#_Toc99919522)

[List of Figure iv](#_Toc99919523)

[List of Tables v](#_Toc99919524)

[List of Abbreviations vi](#_Toc99919525)

[CHAPTER 1: INTRODUCTION 1](#_Toc99919526)

[1.1 Introduction 1](#_Toc99919527)

[1.2 Problem statement 1](#_Toc99919528)

[1.3 Objective 2](#_Toc99919529)

[1.4 Scope and Limitation 2](#_Toc99919530)

[1.5 Development Methodology 2](#_Toc99919531)

[1.6 Report Organization 3](#_Toc99919532)

[CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW 4](#_Toc99919533)

[2.1. Background Study 4](#_Toc99919534)

[2.2. Literature Review 4](#_Toc99919535)

[CHAPTER 3: SYSTEM ANALYSIS 7](#_Toc99919536)

[3.1. System Analysis 7](#_Toc99919537)

[3.1.1. Requirement Analysis 7](#_Toc99919538)

[3.1.2. Feasibility Study 8](#_Toc99919539)

[3.1.2.3. Economic Feasibility 8](#_Toc99919540)

[3.1.3. Analysis 9](#_Toc99919541)

[CHAPTER 4: SYSTEM DESIGN 15](#_Toc99919542)

[4.1. Database Design 15](#_Toc99919543)

[4.1.1. Component Diagram 15](#_Toc99919544)

[4.1.2. Deployment Diagram 15](#_Toc99919545)

[4.2. Algorithm Details 16](#_Toc99919546)

[4.2.1. Road identification 17](#_Toc99919547)

[4.2.2. Road features detection 18](#_Toc99919548)

[4.2.3. Road following 19](#_Toc99919549)

[4.2.4. Reliability and Integrity 20](#_Toc99919550)

[CHAPTER 5: IMPLEMENTATION AND TESTING 21](#_Toc99919551)

[5.1. Implementation 21](#_Toc99919552)

[5.1.1. Tools used 21](#_Toc99919553)

[5.1.2. Pseudo code 21](#_Toc99919554)

[5.2. Testing 22](#_Toc99919555)

[5.2.1. Test Cases for Unit Testing 22](#_Toc99919556)

[5.2.2. Test Case for System Testing 23](#_Toc99919557)

[CHAPTER 6: CONCLUSION AND RECOMMENDATION 25](#_Toc99919558)

[6.1 Conclusion 25](#_Toc99919559)

[6.2 Limitation 25](#_Toc99919560)

[6.3 Recommendation 25](#_Toc99919561)

[REFERENCES 26](#_Toc99919562)

[APPENDICES 27](#_Toc99919563)

# List of Figure

[Figure 1: Incremental Model of Ride Finder App 3](#_Toc99917153)

[Figure 2: Use Case Diagram of Ride Finder App 7](#_Toc99917154)

[Figure 3:Gantt Chart of Ride Finder App 9](#_Toc99917155)

[Figure 4:Class Diagram of Ride Finder App 10](#_Toc99917156)

[Figure 5: Sequence Diagram of Ride Finder App (Driver) 11](#_Toc99917157)

[Figure 6: Sequence Diagram of Ride Finder App(User/Customer) 12](#_Toc99917158)

[Figure 7:Activity Diagram of Ride Finder App(user) 13](#_Toc99917159)

[Figure 8:Activity diagram of Ride Finder App(driver) 14](#_Toc99917160)

[Figure 9: Component Diagram of Ride Finder App 15](#_Toc99917161)

[Figure 10: Deployment Diagram of Ride Finder App 16](#_Toc99917162)

[Figure 11: Map Matching Algorithm of Ride Finder App 17](#_Toc99917163)

[Figure 12: Road Identification 18](#_Toc99917164)

[Figure 13: Road Feature Detection 19](#_Toc99917165)

[Figure 14: Road Following 20](#_Toc99917166)

[Figure 15: Integrity of Map Matching Algorithm 20](#_Toc99917167)

# List of Tables

[Table 1:Unit testing for registration 22](#_Toc98541687)

[Table 2:Unit testing for add profile 23](#_Toc98541688)

[Table 3:Unit testing for searching the vehicle 23](#_Toc98541689)

[Table 4:Integration test for login and vehicle request 23](#_Toc98541690)

# List of Abbreviations

**DR:** Dead Reckoning

**GPRS**: General Packet Radio Service

**GPS**: Global Positioning System

**GSM**: Global system for mobile communication

**SMS:** Short Message Service

**UML:** Unified Modeling Language

# CHAPTER 1: INTRODUCTION

## Introduction

Our project ‘Ride Finder App’ is an android mobile based application solely designed to keep the track of the vehicle moving from one destination to another destination & to perform ride sharing function. It’s mainly focused on the people travelling in local routes because the arrival time of the local routes are random so, our project helps to save the time of the people travelling in the different location by giving the exact location of the vehicle and expected time of arriving in the destination regular time intervals. Implementation of a real-time ride finder system is proposed and developed to keep monitoring vehicles and to provide the information of vehicles to the travelers via android application (app) in a more efficient manner. The idea behind is to provide efficient ride sharing system in Kathmandu(major focus area). Moreover, some limitations arise with the present existing tracking systems in the universe-need hardware maintenance in the GSM/GPRS module that is expensive also, use conventional MySQL database that updates data within few seconds etc. Concerning these limitations, the developed system uses two smartphone applications for transmitting and receiving information using Firebase, an online server with real-time database updating data within few milliseconds that makes the system faster than others. The In-vehicle app receives latitude and longitude values from the satellite and transmits them to the firebase. Firebase updates the data continuously and transmits to the user app. A user can get the information of current location address, distance from any vehicle stop and time to reach a specific destination of the bus through the User app. Hence, the developed ride finder system provides more economical, velocious and user-friendly environment to the users. The users can check and reserve the seat in the vehicle.

## Problem statement

The main focus of our project is on the people travelling on the local routes. People have to wait a lot to reach to their destination. In the context of our country, traffic jam has been one of the major issue in making the person unpunctuality in reaching offices, schools, colleges and other different sectors. Most of their time get wasted in waiting for the vehicles. In order to reach office in time they have to use taxi and other means of transportation which is more expensive than the local routes. So our main concern is to track the location of the vehicle to know the exact location so that people can manage their time to reach to their destination. While travelling through the local routes, sometimes passengers forget their bags and other materials inside the vehicles. With the help of tracking system & ride finder system, one can easily find the location of the vehicle and will be helpful in finding the lost materials.

## Objective

The main objective of our project are:

1. To develop a mobile app to provide ride sharing functionality side by side along with tracking.
2. To reduce the waiting time and to create a convenient and easy-to-use application for passengers

## Scope and Limitation

1. It can track the live location of the vehicle which will be more helpful to the people travelling through the local routes.
2. This project is not only for tracking a vehicle but also people can reserve their seat via online process(ride sharing).
3. People can pay their fare through online payment gateway.

## Development Methodology

The software model used for this project is “Incremental development”. It involves developing an application incrementally and exposing it to customers (in this case, students) for comment, without necessarily delivering it and deploying it in the customer’s environment. Each increment or version of the application incorporates some of the functionality that is needed by the customer. As per the incremental model early increments are the most important functionality.

1. Developing the driver module and plotting the location of vehicles for the user.
2. Adding chat box.

The incremental development model gave an insight for the developers. It was easier to accommodate different user functionality without disturbing the previous increment. Keeping in mind the current features; using customer feedback further increments would be implemented.

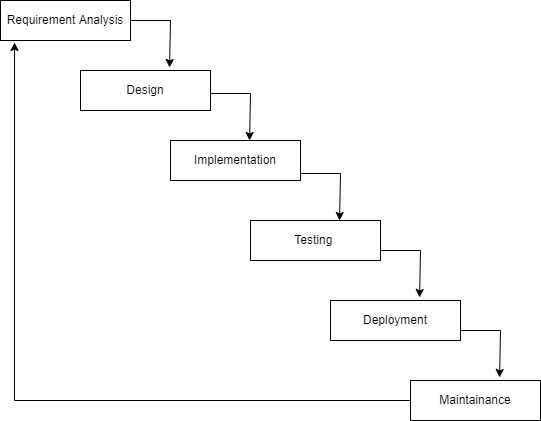


Figure 1: Incremental Model of Ride Finder App

## Report Organization

**Chapter 1**: In this section the overview of the project is given which includes problem statement, objectives, scopes and limitations.

**Chapter 2**: In this section, description of projects of similar nature that were collected and studied is provided.

**Chapter 3**: This section deals with the system analysis aspect of the project that involved functional, nonfunctional and feasibility study of the project.

**Chapter 4**: This section deals with system design. It includes architectural diagram, sequence diagram and algorithm. The detailed working of this project is described in this section.

**Chapter 5**: This section consists of overview of tools used, implementation and testing of the system. Testing includes unit test and integration test.

**Chapter 6**: In this section, the result and conclusion of this project is discussed and also the methods of improvement of the project

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## Background Study

The traditional technique for tracking the movement of vehicles in a commercial fleet involves the use of a central dispatch center and a two-way radio communication in each vehicle. This approach is commonly used worldwide by police departments, delivery companies, and taxi fleets. The use of a two-way radio communication forces driver to remove one hand from the wheel distracting them from the road and creates a serious potential hazard on today's busy roads. Vehicle Tracking Systems are commonly used by fleet operators for fleet management functions such as routing, dispatch, onboard information and security. Other applications include monitoring driving behavior, such as an employer of an employee, or a parent with a teen driver. Ride sharing systems are also popular in consumer vehicles as a theft prevention and retrieval device. Police can simply follow the signal emitted by the tracking system and locate the stolen vehicle. When used as a security system, a Vehicle Tracking System may serve as either an addition to or replacement for a traditional car alarm. The existence of vehicle tracking device then can used to reduce the insurance cost, because the lost risk of the vehicle drop significantly. These days, most of tracking applications use Global Positioning System (GPS) in their tracking systems to report position information automatically back to a central location. In such implementations of tracking applications, use Global Positioning System (GPS) as the Location Estimation technology and Network Transmission (SMS/GPRS) as the Location Transmission technology in those tracking systems. GPS based tracking applications & ride sharing systems are widely used in Europe and developed counties.

## Literature Review

According to, [1], Advanced Vehicle Tracking System on Google Earth Using GPS and GSM In this paper GPS based vehicle tracking/navigation system is implemented. This is done by fetching the information of the vehicle like location, distance, etc. by using GPS and GSM. The information can be transformed with the following features: The information of the vehicle like location, etc. is obtained after every specified time interval defined by the user. Then this periodic information of location is transmitted to monitoring or tracking server. This transmitted information is displayed on the display unit by using the Google earth to display vehicle location in the electronic Google maps.

According to [2],to solve the problem of unmanned ground vehicle leader-follower formation transportation in unstructured environment, we propose a novel target detection and tracking method based on multi-sensor fusion perception. Then during the movement, the dynamic leader is mainly tracked through millimeter wave Radar as this sensor can keep tracking the same target with a constant index and effectively distinguish dynamic vehicle from other static obstacles according to relative speed estimation. First, 3D-Lidar is used to detect the geometric model of the leader vehicle to complete the initialization of tracking target and it can also be assisted for target tracking.

According to [3], The mortality rate of patients who can be cured with on-time treatments is increasing annually, all because of rush-hour traffic and are unable to take the patients to the hospitals in the required time frame. Each ambulance will be equipped with GPS and GSM modem which in case of emergency will send its GPS coordinates to the cloud server, which will then mark the shortest distance from its present location to the hospital via the place from where the emergency call has been raised. The data from the modules will be stored in a cloud server from where the paramedic officials can access it using a unique ID and password that will be issued to them on the integration of this system into the infrastructure.

According to [4], This paper introduced an embedded system that designed and implemented for vehicle tracking based on an android application, the main contribution of this paper is to reduce the data that sent from the embedded system in the vehicle to the cloud server via picking only necessary data for vehicle tracking from Global Position System GPS and decreasing the number of Hypertext Transfer Protocol HTTP request that transmitted to the cloud server by construing the transmission of information with the movement of vehicles. This system is divided into three parts: embedded system that is attached with the vehicle, cloud/server part which has the database of every single move every car did, and the monitoring part which is the main user interface so they can monitor the vehicle.

According to [5],Second application is sending a gathering message for example ready messages to the understudies holding up at the following stop, changes in current course, transport number, and so forth. First application is setting up correspondence between school server and transport framework which is equipped for giving constant information with respect to the present area of transports. Last application is building up a crisis dealing with framework which will send ready messages all the while to school, police and rescue vehicle in the event of mishaps. This Vehicle Tracking System can likewise be utilized for Accident Detection Alert System, Soldier Tracking System and some more, by simply rolling out not many improvements in equipment and programming and broadly in following Cabs/Taxis, taken vehicles, school/universities transports and so on.

According to [6], So smart vehicle using an Internet of Things (IoT) technology is the solution to reduce the number of accidents and also to prevent theft. The GPS/GPRS system continuously keeps the track of vehicle location and can be used during an accident and theft. This paper presents a smart system design to detect the accident immediately. Though there are numerous different reasons behind car accidents, most injuries occur due to driver's unawareness and uncontrolled speed.

According to [7],During burglary of vehicles, the owner can deactivate engine vehicles ignition then consecutively owner can lock the gear panel, ABS brakes can also be locked if necessary. The complete prototype is designed on a single chip which is very simple and cost effective. The complete prototype is designed on a single chip which is very simple and cost effective. The proposed project aims to develop a portable embedded system to avoid burglary on vehicles.

According to [8],a system is developed using Arduino Uno R3 is being developed by our team, global system for mobile (GSM) device, and global positioning system (GPS) to track the exact and accurate position of the vehicle at any location. Furthermore, two software are utilized to display the data namely Thing Speak which is used to display the trend of the vehicle's motion by using longitude and latitude charts, and Freeboard which is utilized to display the same information in the form of a map that is accessible to all. The entire system has been tested thoroughly in real time and it has proven to function successfully in helping users to locate their vehicles in the event of a theft. The device is also equipped with display to show the information on the current location of the vehicle to the user.

# CHAPTER 3: SYSTEM ANALYSIS

## System Analysis

System analysis was carried out to investigate the system and its components in order to determine the goals. It was used as problem-solving strategy to enhance the system and to guarantee that all of the system’s components are work together to achieve our required goals. We collected and interpreted facts, identified the problems, and decomposition of a system and system’s components.

### Requirement Analysis

Software requirement can be analyzed, documented, validated, and managed using requirement analysis. The requirements are collected, elicited, analyzed, and modeled first. The requirement analysis is depicted in the use case diagram.

#### Functional Requirement

The below figure shows the functional requirements of the system as a use case diagram. The functional requirements are user registration which includes validation. When the user is registered, the system automatically checks if the input credentials are valid or not. Similarly, for login, the system checks if the credentials are valid or not. After login, the user can see the location of the vehicles of the different track. Even user can reserve the seat via online.

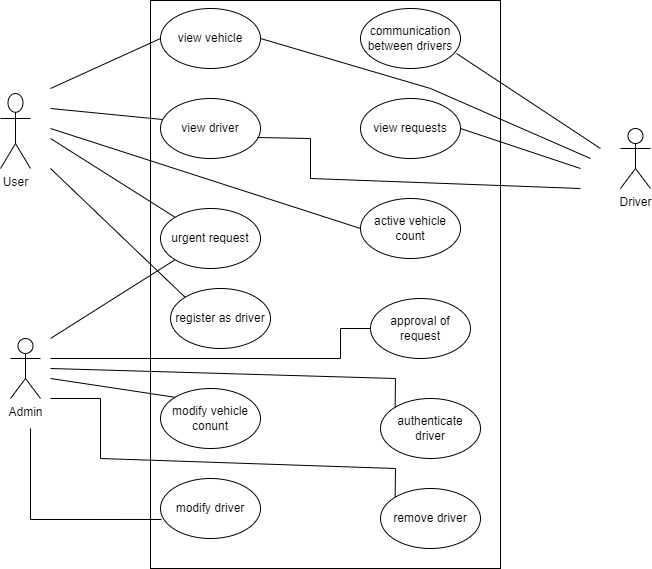


Figure 2: Use Case Diagram of Ride Finder App

The above figure 2 shows the functional requirements of the system as a use case diagram. The functional requirements are user registration which includes validation. When the user is registered, the system automatically checks if the input credentials are valid or not. Similarly, for login, the system checks if the credentials are valid or not.

Non-Functional Requirement

1. Security and Reliability: The system will make sure that the provided information of the users is stored reliably using the trusted database system structured systematically so their minimum chances of files being lost or leaked.
2. User Friendly: The system is very easy to operate which doesn't require high level knowledge for the user to use the application user with basic knowledge can easily login and enjoy the facility provided by the application.

### Feasibility Study

A feasibility study is an analysis that considers all of a project's relevant factors—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully

#### Operational Feasibility

The following factors were considered for operational feasibility:

* + - * 1. The ride finder system keeps a real time track of the active vehicles by dynamically updating the geo-location of the active vehicle on the firebase database and showing the updated location to the user’s end in real time
        2. This application has been developed to makes easier to passengers to know the location of the vehicle.

#### Technical Feasibility

The technologies used in this project like flutter, firebase are simple and are easily available. It can be easily hosted in the available database and even on application servers provided by the cloud providers which make the project technically feasible.

#### **Economic Feasibility**

The tools and technologies like flutter are free and open source are easily available. There is not any involvement of paid software or extensions which make it economically feasible.

#### Schedule Feasibility

Time schedule/Gantt chart has been create according to the methodology used for development. The total development task is sub-divided into various phases and allocated time scheduled as per requirements

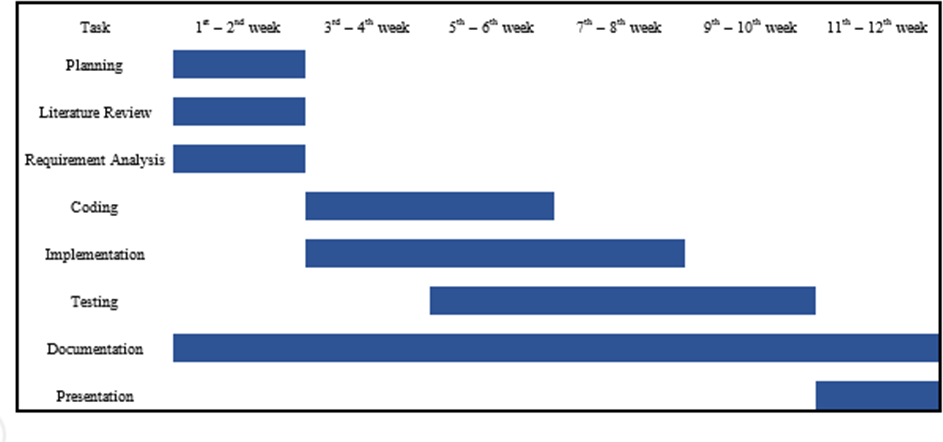


Figure 3:Gantt Chart of Ride Finder App

### Analysis

This project is based on Object Oriented Approach. In the object-oriented analysis phase of software development, system requirements are determined, classes are identified, and relationships between classes are identified.

#### Class Diagram

Class Diagram is the main building block of object oriented modeling. It describes the

Structure of system by showing the system’s classes, their attributes, operations and the

Relationship among objects. Our purpose of class diagram is to model the static view of

Of and application. Here we are describing the functionalities performed by the system.

Here we can see User as a superclass and there is generalization between superclass and

Subclasses of Location data, Database reference and Driver Data.

We can also see one to one relationship between register driver & driver data class, also

We have one to many relationship between driver data and the user.

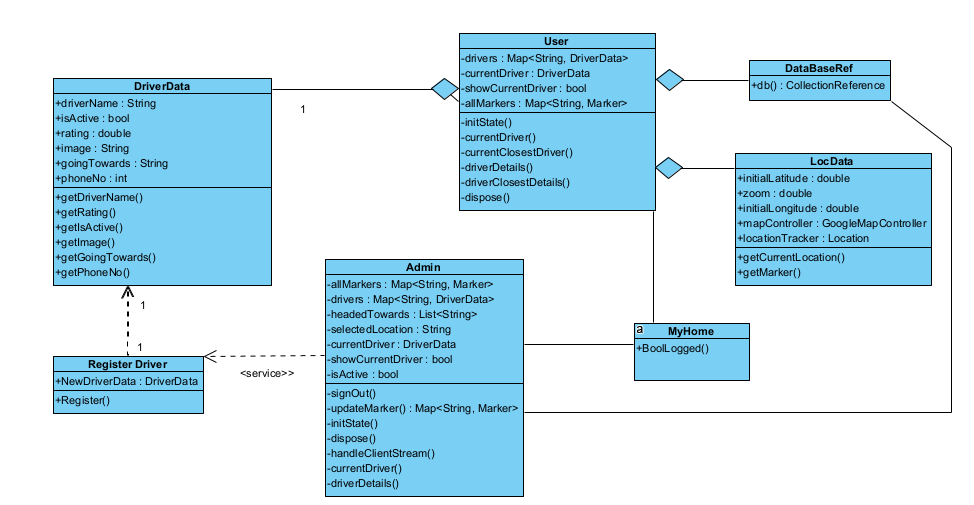


Figure 4:Class Diagram of Ride Finder App

#### Sequence Diagram

Sequence diagrams shows objects interactions arranged in time sequence. Below we can see an object interactions between applicant, interface & backend. Here in the figure there are two sequence diagram.

1. Sequence diagram for user
2. Sequence diagram for driver
3. **Sequence Diagram of driver**

We can see various lifeline notation for applicant and interface.

We can see various activation bars within life line & various asynchronous message between

Them like register, verify, request, location, etc.

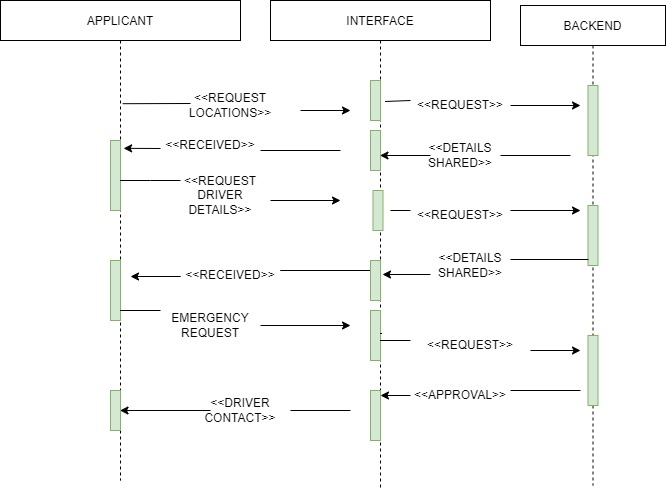


Figure 5: Sequence Diagram of Ride Finder App (Driver)

1. **Sequence Diagram of User/Customer**

We can see below for sequence diagram for user we have various lifeline notation for applicant,

Interface & backend. We can see various activation bars on lifeline & various asynchronous message between them like request location, request driver details, etc.

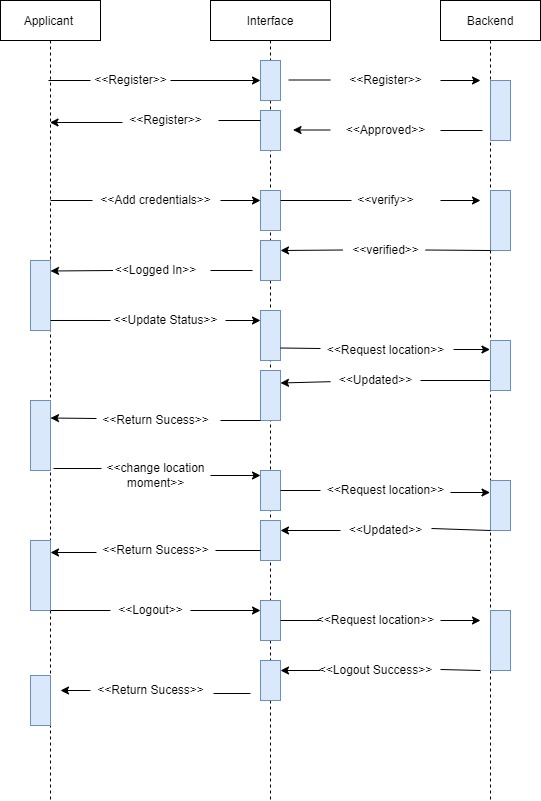


Figure 6: Sequence Diagram of Ride Finder App(User/Customer)

#### Activity Diagram

An activity diagram is a behavioral diagram, it depicts the behavior of a system. It portrays the control flow from a start point to a finish point showing the various decision paths that exists while the activity is being executed

1. **Activity Diagram of User**:

In figure below, we can see activities being carried out with two decision nodes along with the presence of one initial node and one final node.

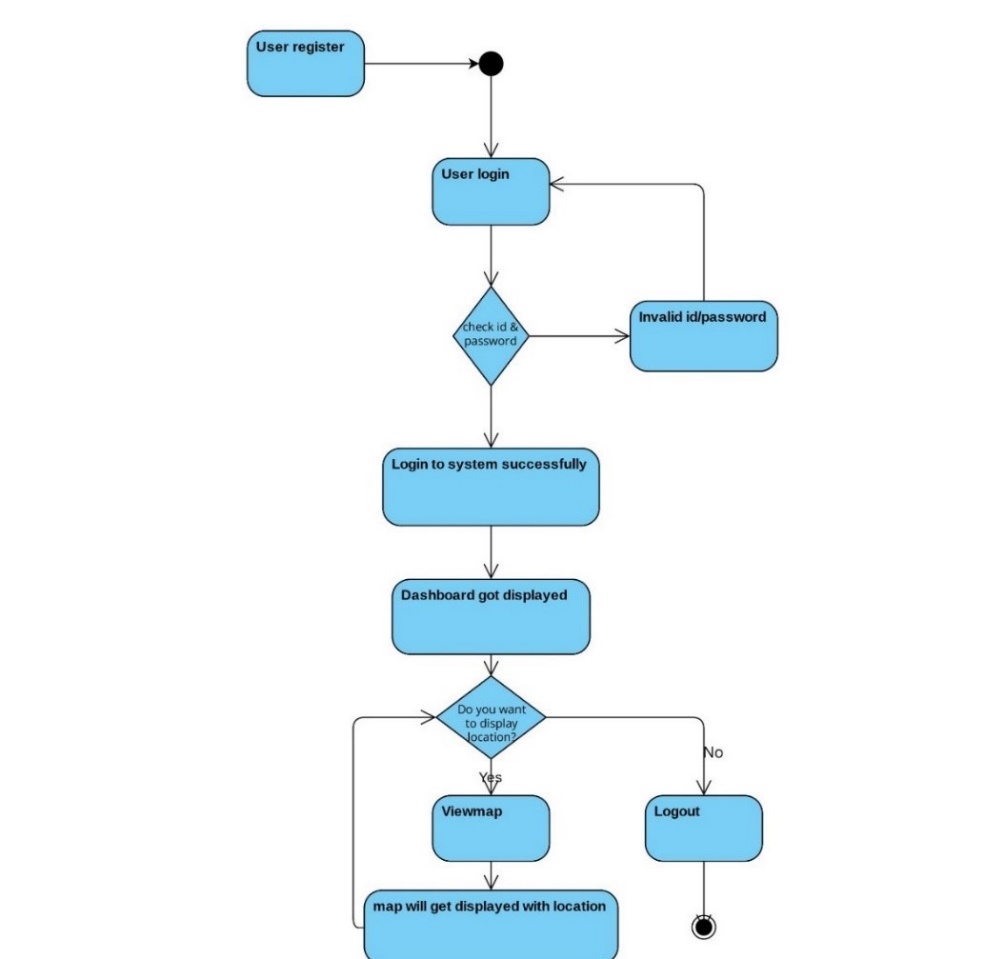


Figure 7:Activity Diagram of Ride Finder App(user)

1. **Activity Diagram of Driver:**

In the driver side we perform similar activities like that of  the user side with two different kinds of activities like sharing a location & map displayed to the user online. Similarly, we have use two decision nodes and one initial and final state.

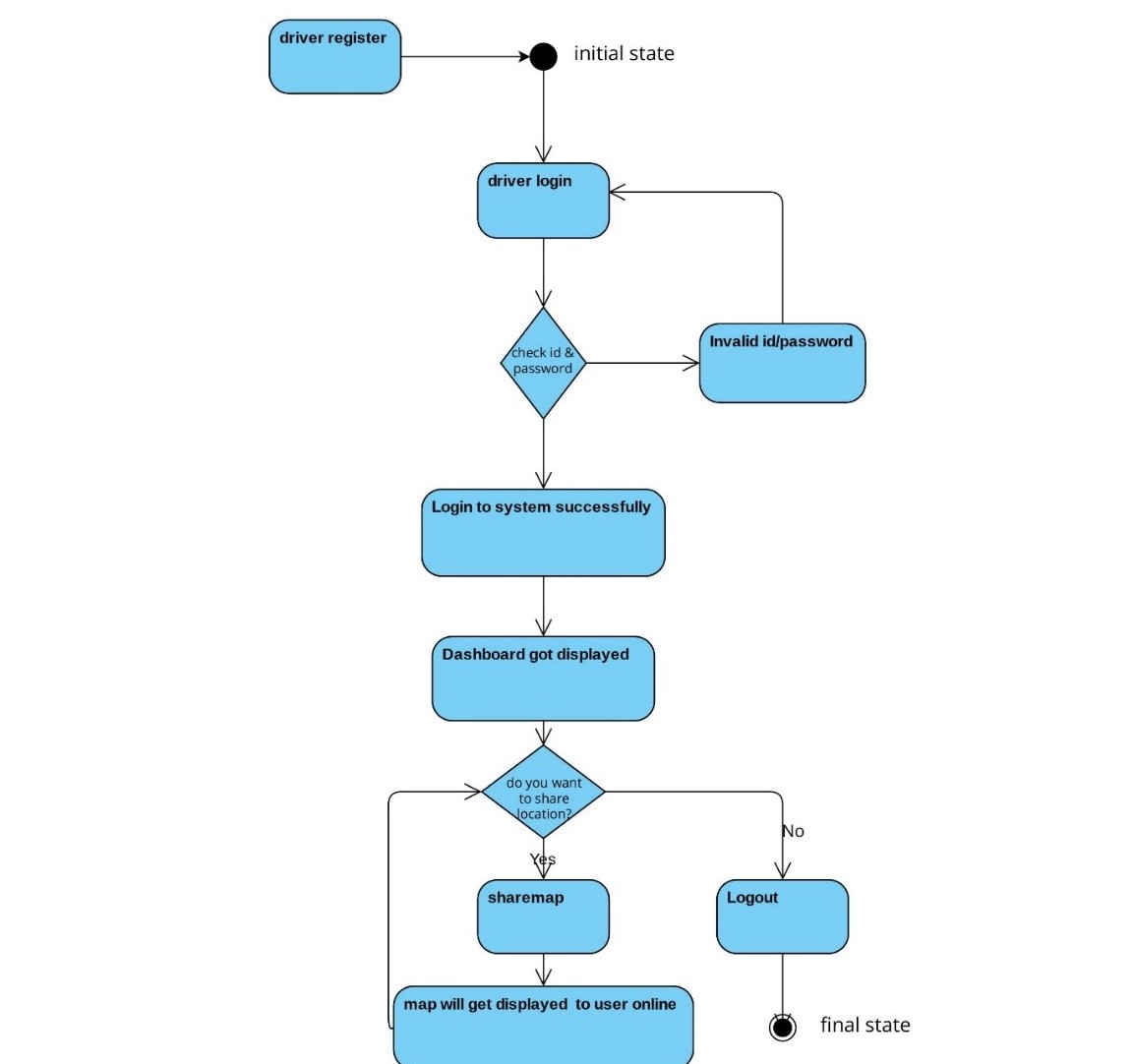


Figure 8:Activity diagram of Ride Finder App(driver)

# CHAPTER 4: SYSTEM DESIGN

## Database Design

### Component Diagram

A component diagram describes the organization and wiring of the physical components in a system. It is often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. In Ride Finder App, the components are user, driver, account, vehicle tracking. The provided interface of the system are security, persistence and database. Driver update the location in the vehicle tracking system and user get the information from the Ride Finder/vehicle tracking system.

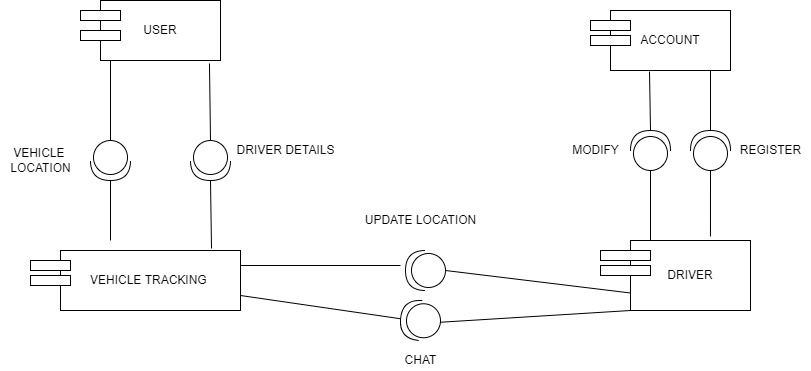


Figure 9: Component Diagram of Ride Finder App

### Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.

We have three nodes below with each embedded artifacts. Also we have nested nodes. Our major node includes mobile device, application server, database (firebase, real time database).

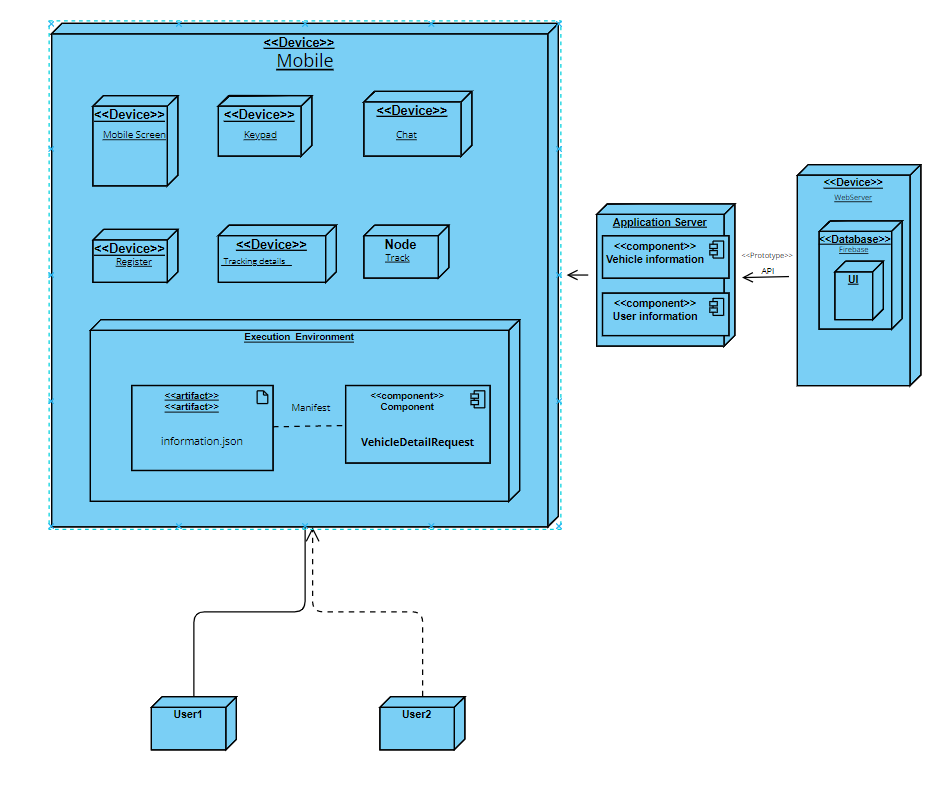


Figure 10: Deployment Diagram of Ride Finder App

## Algorithm Details

The algorithm used in our project is map matching algorithm. Map Matching algorithm integrate positioning data with spatial road network data to identify the correct link on which a vehicle is travelling and to determine the location of a vehicle on a link.

Map matching algorithm can be divided into four key operations.

1. Road identification
2. Road feature detection
3. Road following
4. Reliability and integrity

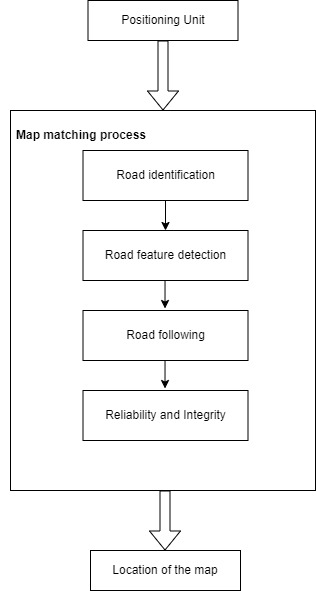
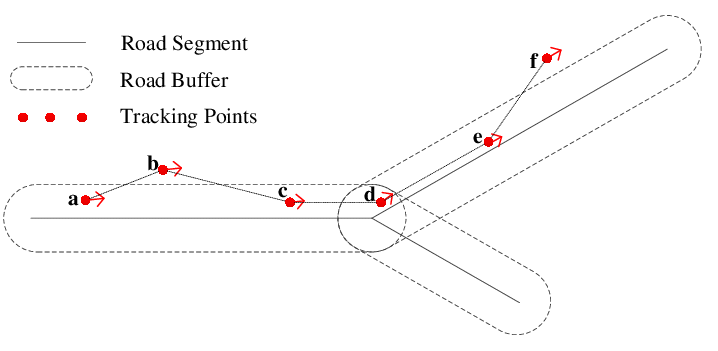


Figure 11: Map Matching Algorithm of Ride Finder App

### Road identification

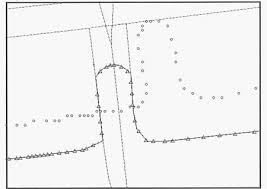
It is a procedure of finding the road vehicle is currently traveling on. Road feature detection is a procedure of detecting geometric and topologic feature of roads, such as road turn, road curvature, and road connection. After a road is identified, the vehicle position is relocated on the proper position relative to the identified road. This procedure is called road following. It would be serious problems if a vehicle were matching to a wrong road without notice. If' that were the case, the positioning function would be totally lost unless GPS is available. Therefore, special algorithms have designed for the reliability and integrity checks for map-matching.



**Figure 12: Road Identification**

### Road features detection

When a road segment is determined, we can extract the related information of the road, such as the road azimuth, road length, road connectivity, turn restriction of its junction. The information can then be used for next road identification and calibrate measurements (distance, bearing, and position) of the positioning unit. Moreover, such information is important for the reliability and integrity checks in data processing. Road azimuth can be used to correct the bearing measurements. As we determine the road segment, we can calculate the azimuth of the road segment. Because we know that the vehicle is now traveling on the road, we can use the road azimuth to calibrate gyro or compass measurements. The difference between the last relocated position and the last measured position can also be used to correct the current position measurement. Whenever a feature point of road segment such as the largest curvature point is detected, it can be used to correct the accumulated distance error.



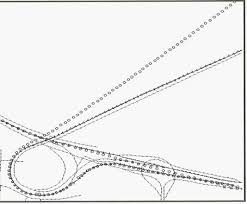
**Figure 13: Road Feature Detection**

### Road following

Road following is a process to determine the location of the vehicle on an identified road segment. After a road candidate is determined, a new position can be obtained from two sources. One is the predicted based on the vehicle velocity and the road azimuth and another from positioning sensors. The final position is obtained by the weighted average of the two solutions, as showed in the following equation:

**i = c1^\_'(c, +c2)-'x1 +c;l(cl +C,)yX,**

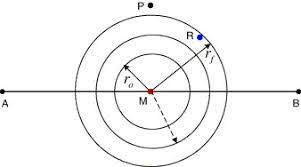
where X, is the projected position measurement and X2 predicted position. C, and C2 are the corresponding covariance matrices.



**Figure 14: Road Following**

### Reliability and Integrity

The reliability check is the procedure that makes sure the vehicle location is correctly matching to the map road network, while integrity check tries to identify any wrong matching. As the reliability and integrity requirements for land vehicle navigation is lower than civil aviation requirements, it is not a severe problem if GPS is available. However, this is very important for urban environment navigation because GPS may not be available for very long period, and position of vehicle is provided by the DR unit. If a vehicle location is wrongly matched on the road network, the following process will be completely wrong as DR uses this location to derive the next position unless reliable GPS is obtained again.



**Figure 15: Integrity of Map Matching Algorithm**

# CHAPTER 5: IMPLEMENTATION AND TESTING

## Implementation

### Tools used

For the development of this project, various tools and technologies were used. Additional hardware tools were not required. The following are the software tools used in this project:

1. Flutter: Flutter is a programming language which used develop the mobile app in both android and IOS. We have used flutter for the front end design.
2. Firebase: For back end of our app, we have used Firebase. Google firebase is a Google fire backed application development software that enables developers to develop android, IOS, and web application. It provides tool for tracking analytics, reporting and fixing crash apps, creating marketing and product experiment.

### Pseudo code

**void** displayRequestContainer() {  
 setState(() {  
 requestRideContainerHeight = 250;  
 rideDetailsContainerHeight = 0;  
 bottomPaddingOfMap = 230;  
 drawerOpen = **true**;  
 });  
 saveRideRequest();  
}  
  
DatabaseReference? rideRequestReference;  
  
BitmapDescriptor? nearByIcon;  
  
List<NearByAvailableDrivers>? availableDrivers;  
  
bool init = **true**;  
@override  
**void** didChangeDependencies() {  
 **super**.didChangeDependencies();  
 **if** (init) {  
 Provider.of<AppData>(context, listen: **false**).pickUpLocation;  
 }  
 init = **false**;  
}  
  
List<LatLng> pLineCoordinates = [];  
Set<Polyline> polyLineSet = {};  
  
@override  
**void** initState() {getCurrentUser();

}

## Testing

Software testing is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. We have done different types of testing in order to verify and validate.

### Test Cases for Unit Testing

For unit testing, we selected different modules of the project and tested them individually.

**Table 1:Unit testing for registration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Test input | Expected output | Actual output | Result |
| Valid registration | Email  (eg.samir@gmail.com)  Name (Samir) and password(Samirwaiba) | Registration successful message and redirect to dashboard | Registration successful message and redirect to dashboard | Test successful |

**Table 2: Unit testing for add profile**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Test input | Expected output | Actual output | Result |
| Add user profile | User address, name, contact no | Saved profile | Profile saved | Test successful |

**Table 3: Unit testing for searching the vehicle**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Test input | Expected output | Actual output | Result |
| Search for the vehicle | Request is sent by the user | Request Notification sent to the driver. | Request is accepted by the driver. | Test successful |

### Test Case for System Testing

For integration testing, we tested two modules together: login and create post

**Table 4: Integration test for login and vehicle request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Test input | Expected output | Actual output | Result |
| Login and search for the vehicle. | Email([samir@gmail.com](mailto:samir@gmail.com))  and password eg  . (samir123) for login and search for the vehicle. | Logged in request notification to the driver | Logged in request is accepted by the driver. | Test successful |

# 

# CHAPTER 6: CONCLUSION AND RECOMMENDATION

### 6.1 Conclusion

As a result, our project enables travelers and vehicle drivers to interact and connect with one another in order to receive fast and efficient service for travelling from one destination to another. Our system also provides a reliable medium for tracking a vehicle you desire. Our system is indeed useful, user friendly and adaptable that everyone can use efficiently.

## 6.2 Limitation

* Improvement on rating with driver’s sentiment analysis.
* Improvement on pre booking services with more payment gateway integration.
* Ride finder system for the entire country of Nepal.

## 6.3 Recommendation

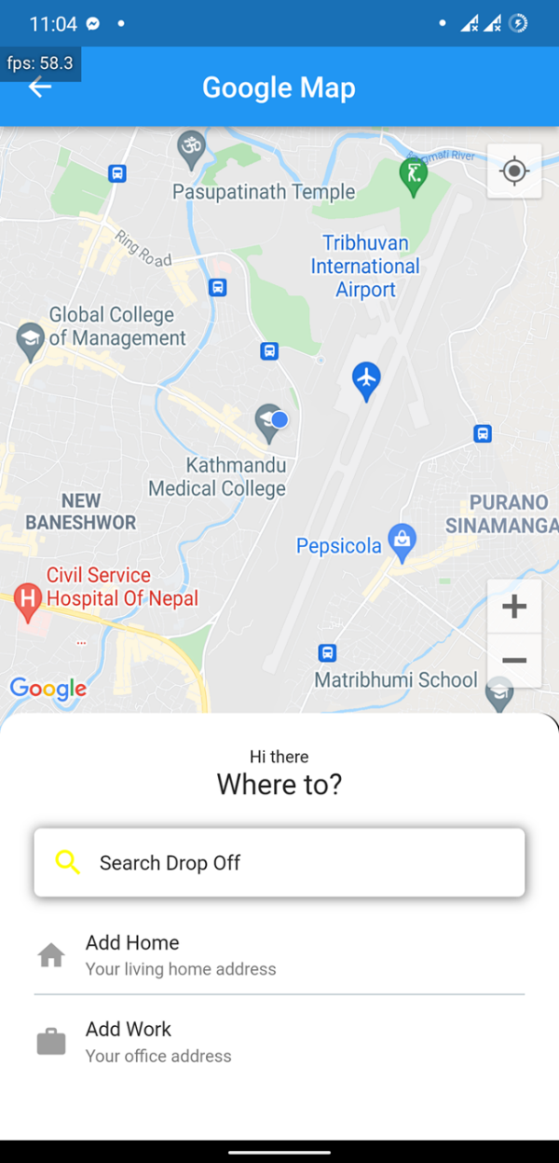
* People who adore travelling are highly recommended to use this app.
* People can establish time and cost effective way of travelling. Through this platform

# REFERENCES

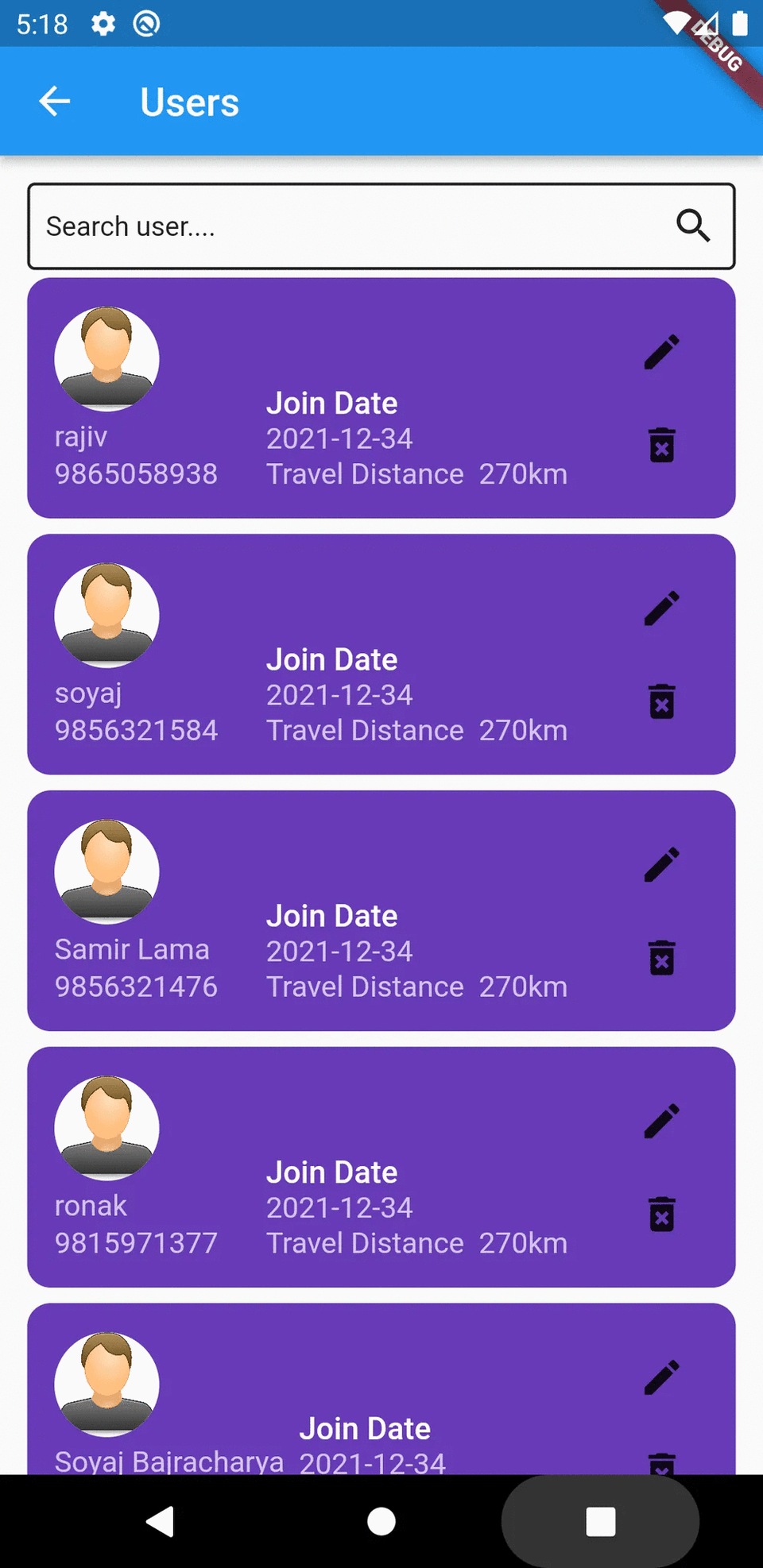
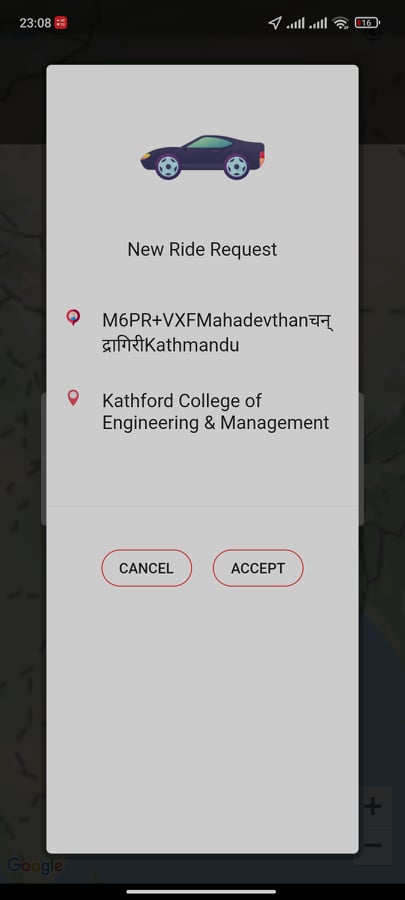
|  |  |
| --- | --- |
| [1] | S. k. &. H. B. Yanamadala, Advanced Vehicle Tracking System on Google Earth Using GPS and GSM, Eluru, AP, India: International Journal of Computer Trends and Technology (IJCTT), Dec 2013. |
| [2] | B. N. R. N. K. S. Vigneshwaran, "Design of Bus Tracking and Fuel Monitoring System," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 6-7 March 2020. |
| [3] | C. L. X. Y. X. L. B. S. Ning Li, "Real-time 3D-Lidar, MMW Radar and GPS/IMU fusion based vehicle detection and tracking in unstructured environment," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 30 May-5 June 2021. |
| [4] | A. M. I. a.-N. a. O. A. A. Mustafa, "A Smart real-time tracking system using GSM/GPRS technologies," in *2019 First International Conference of Computer and Applied Sciences (CAS)*, Baghdad, Iraq, 2019. |
| [5] | A. A. Megha Dewan, "IOT Based Smart Vehical Monitoring and Tracking System," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, Moradabad, India, 4-5 Dec. 2020. |
| [6] | S. S. S. R. R. Ganesh Prabhu S., "Tracking Real Time Vehicle And Locking System Using Labview Applications," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 6-7 March 2020. |
| [7] | A. B. M. G. S. B. S. K. &. C. A. Baksi, "Internet of Things (IOT) Based Ambulance Tracking System Using GPS and GSM Modules," in *2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, Kolkata, India, 2-4 Oct. 2020. |
| [8] | M. R. Y. A. S. M. Abdullah H. Alquhali, "IOT Based Real-Time Vehicle Tracking System," in *2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET)*, Penang, Malaysia, 7-9 Nov. 2019. |

# APPENDICES

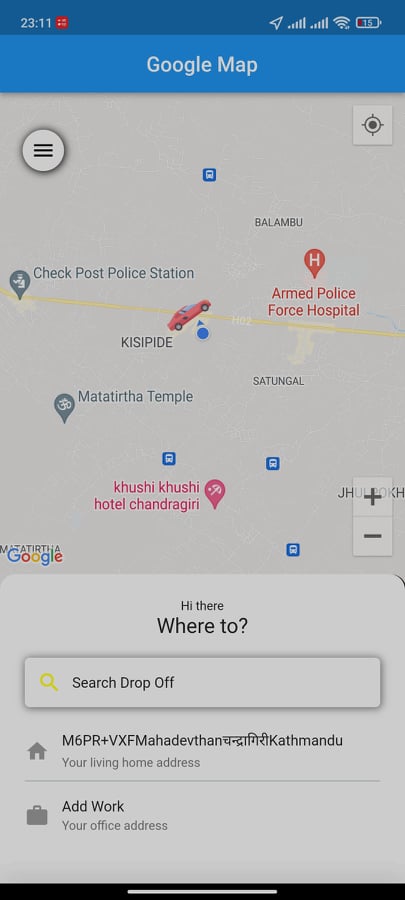
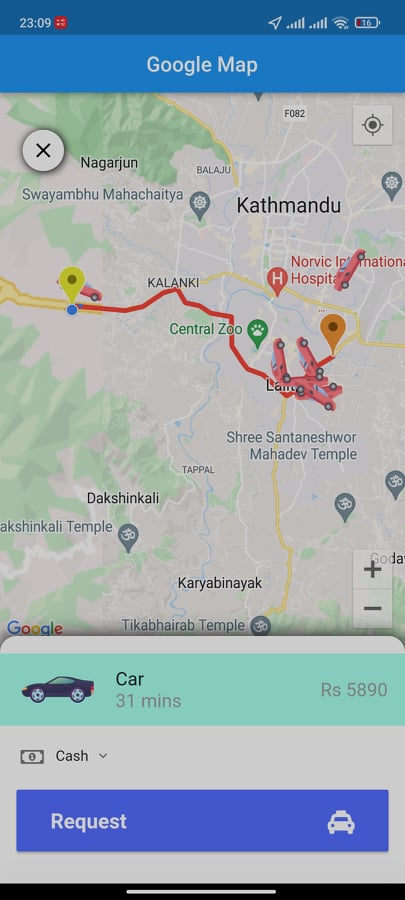
**Appendix for Homepage:**

****

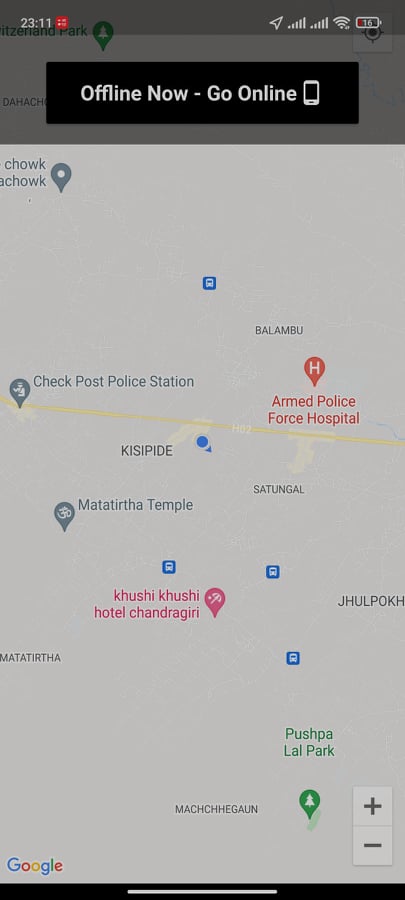
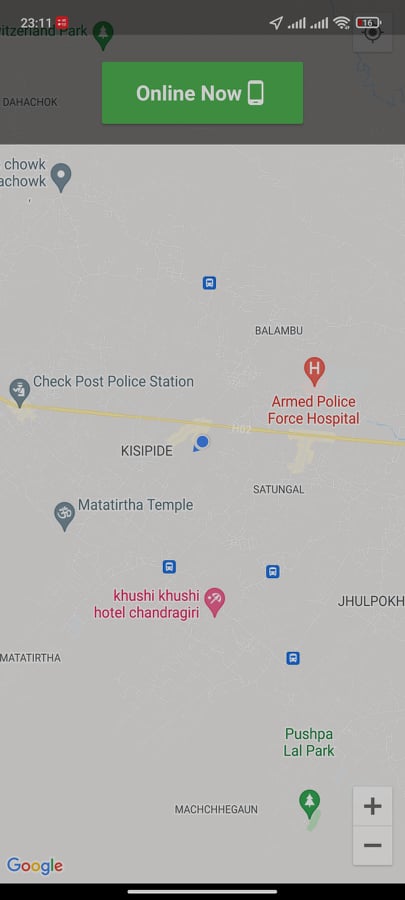
**Appendix for user and driver:**

** **

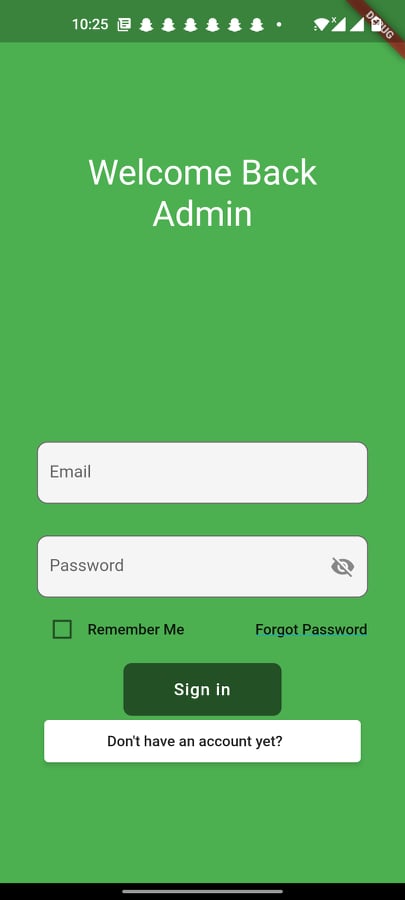
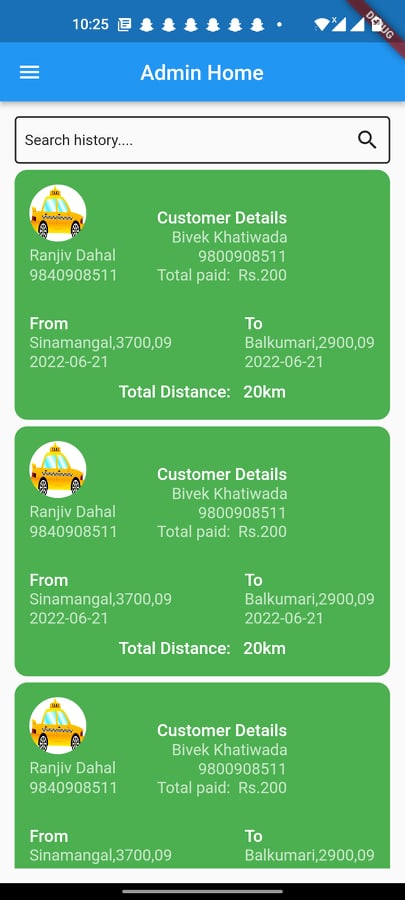
**Appendix for Map:**

****

**Appendix for Rider:**

****

**Appendix for Admin**

****