A REPORT

on

POLO-S

Submitted by

NIKHIL JOHN JOSE

PILLAI MIDHUN UNNIKRISHNA

NAVEEN WILSON

Bachelor of Technology

in

Computer Science & Engineering

Department of Computer Science and Engineering

Saintgits College of Engineering

ABSTRACT

Efficient and reliable transportation is crucial in today's fast-paced world, necessitating innovative solutions to overcome the limitations and concerns associated with traditional car booking systems. With the help of a user-friendly system, passengers would be able to travel more happily, safely, and affordably thanks to the PoLo project. This is based on the idea of getting a ride with a stranger that is safe. It enables the drivers and passengers to keep an eye on one another while also allowing the passengers to choose their fellow travelers based on personal information. The main advantage of this strategy is that all available rides and passengers with the same destination and source route may be listed nearby, saving time for travelers who would otherwise have to scour the entire road for a transport. An individual will be able to view the travel companions while also splitting the cost equally among all passengers hence reducing the actual cost. Additionally, the application is successful in keeping track of each rider's trip information for whom rides have been completed. One of the projects that will improve and streamline the sharing of travel experiences is POLO-S.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Booking cars is quickly becoming a recognised and utilized business model. Numerous companies are using software that enables customers to book cabs with only a few clicks of their fingers. It is not an overstatement to suggest that the procedure of reserving a ride on the Internet is growing simpler, according to the online Ride booking system project. This ride-sharing system project aims to create a welcoming environment for carpooling where you can pick the passengers you ride with. Customers are drawn to today's ride-sharing applications not only because they are very handy, but also because they offer a broader variety, comparable rates, better ride information (including user reviews), and incredibly straightforward navigation.

Ride-sharing has emerged as a cutting-edge and thrilling technology globally. By providing effortless access to rides, it offers unparalleled convenience to individuals, making it one of the most widely adopted applications today. The system has successfully incorporated online ride functionality, expanding beyond anonymous ride-sharing and allowing users to easily comprehend and accept riders of their choice. This progressive approach has greatly accelerated the development of ride-sharing applications, especially with the advent of platforms like Uber and Ola, which have revolutionized the industry with their user-friendly and convenient services.

This project is a riding system for similar ride searches that runs on both Android and iOS. Users can directly search for available drivers on the Internet in real time without the use of middleman services by using online ride sharing and prebooking. The goal of this initiative is to use the advantages of online rider and driver searches to physically avoid time- and money-consuming searches. helps you use your Android mobile to get transportation anywhere via the internet. Riders can therefore request rides and select the services they want from their preferred or chosen drivers.

1.1 Project Objective

Through a single internet gateway, PoloS for ride-sharing aims to offer comprehensive solutions for both users and drivers. It will allow drivers to acquire direct riders without having to physically wait on the highways for hours, and it would allow passengers to choose the driver and other passengers they choose to take the ride with. Both users and drivers will be able to accept and refuse rides using the user-driver module.

1.2 Project Scope

With acceptable terms of application, this system can be applied in any tiny local regions or across borders. On demand, we locate the appropriate rides, drivers, and ride systems for you. Riders can choose from a variety of rides from any location with the help of an online interface, saving them time from having to navigate through traffic. Both iOS and Android users will be able to download the software.

1.3 Project Overview

The technology offers a simple way for drivers to find riders without spending hours driving around looking for them with an added benefit for riders to select the other riders with whom they want to share the ride with along the way. The suggested system can be used by any inexperienced user and does not require any training, prior experience, or technical understanding in the field of computers, but it is helpful if the user is familiar with how to use a phone.

CHAPTER 2

LITERATURE REVIEW

This study of the literature compares and contrasts the many systems frequently employed in the creation of ride-sharing apps, taking into account their features, scalability, security, and adaptability. Systems used to build ride-sharing applications show the significance of dependable geolocation and mapping services, strong backend infrastructure, and secure payment processing. Scalability and dependability are provided by cloud-based platforms like AWS or GCP, while effective backend development is facilitated by frameworks like Node.js or Ruby on Rails. Effective ride matching and navigation support are ensured by integration with precise geolocation and mapping services. The usage of secure payment processing technologies also fosters consumer convenience and confidence. Ride-sharing software may provide consumers streamlined and dependable transportation experiences by using these systems efficiently.

India's ride-sharing services demonstrate how they might improve urban mobility, manage traffic issues, and handle other transportation-related issues. For ride-sharing services to expand sustainably, however, infrastructural, safety, and regulatory issues must be resolved. Design and usage of ride-sharing programmes may be further enhanced by understanding user behavior, preferences, and cultural considerations unique to India. Ride-sharing has important effects on urban mobility in India and has the potential to develop a more integrated, open, and sustainable transportation system. To maximize the advantages of ride-sharing apps in the nation, future research should concentrate on solving the special difficulties and possibilities in the Indian setting.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The system feasibility can be divided into the following sections:

3.1.1 ECONOMIC FEASIBILITY

The concept is financially viable because all that is required is an Android or iOS device that satisfies the aforementioned basic criteria. The sole expense incurred by users in order to access the programme is Internet connectivity.

3.1.2 TECHNICAL FEASIBILITY

To deploy the application, the only technical aspects needed are mentioned below:

- OS: Windows 10 or above

For Users:

- Android 6 and Ios 13 or above
- Internet Connection

3.1.3 BEHAVIOURAL FEASIBILITY

The application doesn't need any particular technological assistance, and each view it offers is self-explanatory. Users are given clear instructions on how to use the system, and all actions made are accompanied by warning and error notifications.

CHAPTER 4

OBJECTIVES AND FUNCTIONALITIES

4.1 OBJECTIVES

- Our project's goal is to provide a user interface that allows users to either receive a ride as a driver or a ride with other regular riders through online mode.
- A user can login into the system and then search for rides to their desired locations.
- The drivers can then view corresponding requested rides within a proximity.

4.2 FUNCTIONALITIES

- Provides the searching facilities based on various factors. Such as area, user, destination and source, ride confirmation.
- PoloS also manages the request details online for Request ride details, Confirm ride details.
- It tracks the information of Riders, Distance traveled.
- Manage the information of Location.
- Manage the User information.
- It deals with monitoring the information and motion of the current rides.
- It deals with monitoring the information and rides within a users' login.
- Searching and requesting is made easy with proper management of distances.

CHAPTER 5

DESIGN

5.1 TECHNOLOGY STACK

The technology stack used to create our system are as follows:

FRONTEND

- **FLUTTER:**

  Flutter is an open-source UI toolkit developed by Google, revolutionizing the way developers build high-performance, visually stunning applications across multiple platforms. This one-page description provides an overview of Flutter's key features, benefits, and its impact on cross-platform app development.

- **ANDROID STUDIO:**

  The official Integrated Development Environment (IDE) for developing Android apps is called Android Studio, and it offers programmers a wide range of tools and resources to help them produce top-notch Android applications. Android Studio, which is based on the IntelliJ IDEA platform, provides a user-friendly interface and a wealth of capabilities that speed up the development process. It features a sophisticated code editor with automatic code completion, debugging tools, and integrated emulators for testing and running programmes. Additionally, Android Studio easily interacts with the Android SDK, giving developers access to a wide range of frameworks, APIs, and development tools, making it a crucial tool for building effective and reliable Android applications.

BACKEND

- **FIREBASE:**

  Google offers Firebase, a strong and complete platform for creating and managing online and mobile apps.

It provides a wide range of tools and services that streamline and quicken the development process. Without the need for complicated server architecture, developers can quickly set up authentication, real-time database storage, cloud messaging, hosting, and more using Firebase. Additionally, it has strong analytics and crash reporting features that give developers insightful knowledge on how users behave and how well their applications run. For developers wishing to create feature-rich and scalable applications, Firebase is a fantastic option due to its scalability, usability, and thorough documentation.

5.2 REQUIREMENT SPECIFICATION

5.2.1 FUNCTIONAL REQUIREMENTS

- Users must register if they want to search for rides. Unregistered users will not be able to access the application.
- The User logs in to the system by entering a valid email ID, Phone number and password.
- A User can search for rides by giving a source and drop off location or cancel the search after logging in or registering.
- A User can search for other riders with the approximate same pickup and drop off for requesting shared rides by pinging them and adding them as friends.
- In this system, the payment type is cash payment. Extend this to upi payments ,credit cards, debit cards, PoloPay etc. in the future.
- After placing a ride, the system sends the information of driver to rider and rider to driver by the connected system database.

5.2.2 NON-FUNCTIONAL REQUIREMENTS

- Secure access to consumer’s confidential data.
- Driver based availability.
- Better component design to get better performance at peak time.
- Flexible rider-base architecture will be highly desirable for future extension.
- The system shall provide an easy and attractive graphical interface for flexible use
- The Application should be easily maintainable for the users.

5.3 ACTIVITY DIAGRAM:

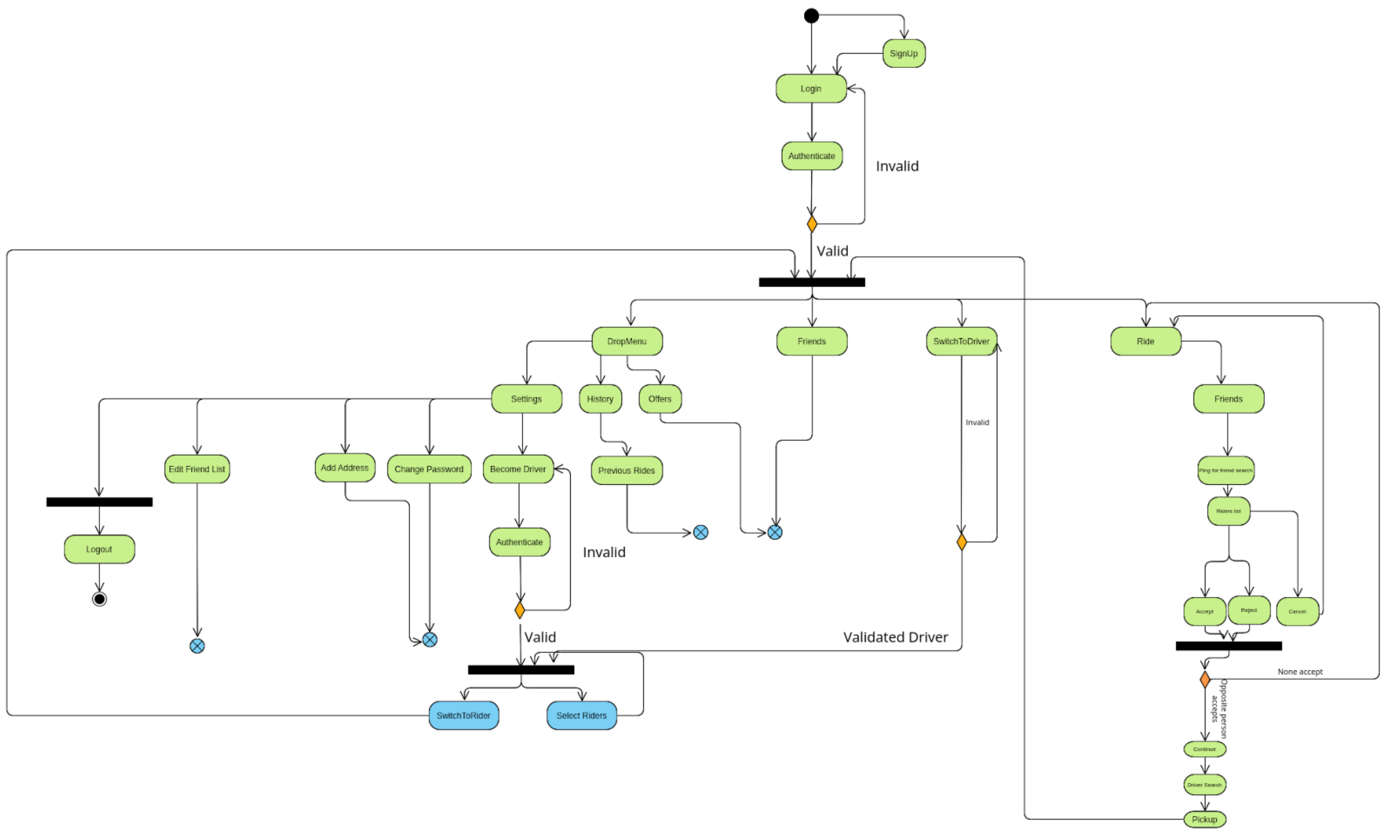


Fig 5.1: User Activity Diagram

The activity diagram in *Figure 5.1* shows the different way in which the user can interact with the application. Once a user starts the application, they can either login into their account if they have already registered or else, they can register as a new user by filling up the details in the sign-up page.

Upon logging in, the user will be taken to their respective Home pages. A registered user can search rides, view and edit profile and view ride reports or statistics. For activating the driver interface, the rider can apply and register as driver in the settings menu by providing the required documents. From there on, the rider can either request a ride or can themselves become drivers and get riders from their close proximity region.

Users can edit their personal details and change passwords. Users can also view ride reports and other rider profiles. At the end the user can post ratings for fellow riders and drivers.

5.4 RELATIONAL SCHEMA

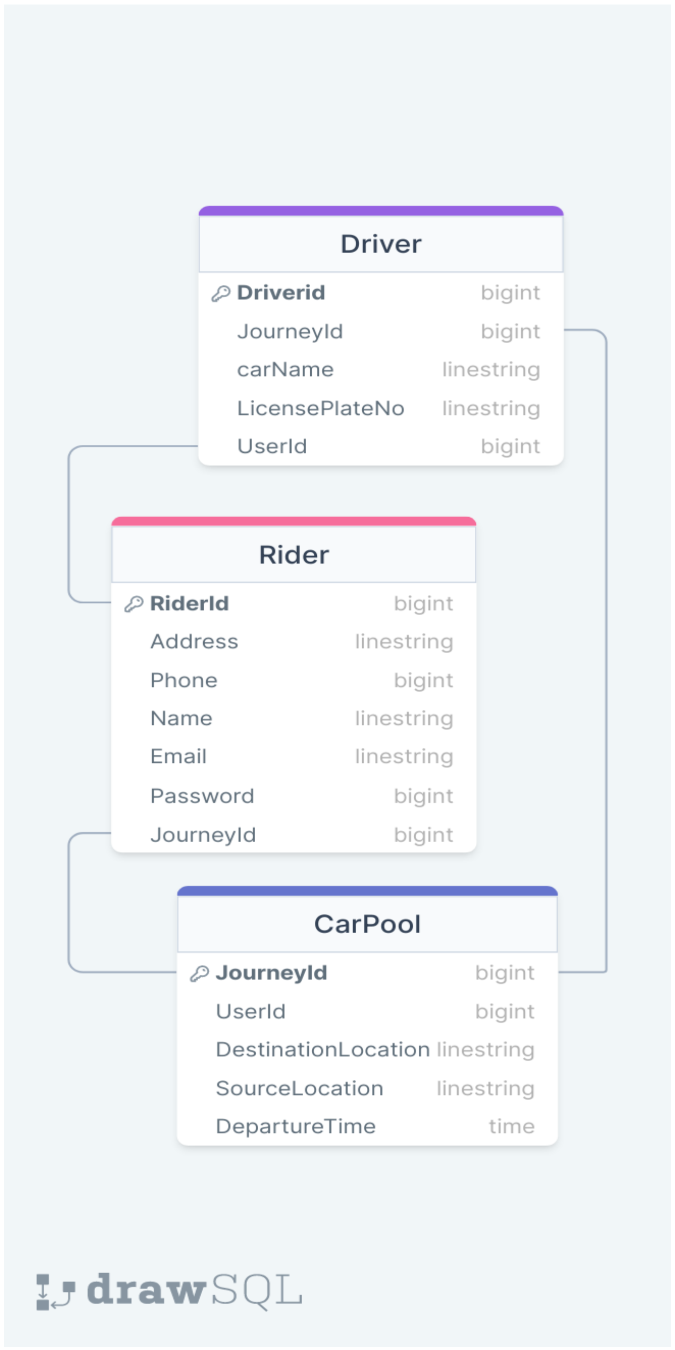


Fig 5.2: Relational Schema

A relational schema is a structure that represents a logical view of the entire database. It defines how data is organized and how relationships between data are linked. It formulates the limits applied on the data.

5.5 E-R DIAGRAM:

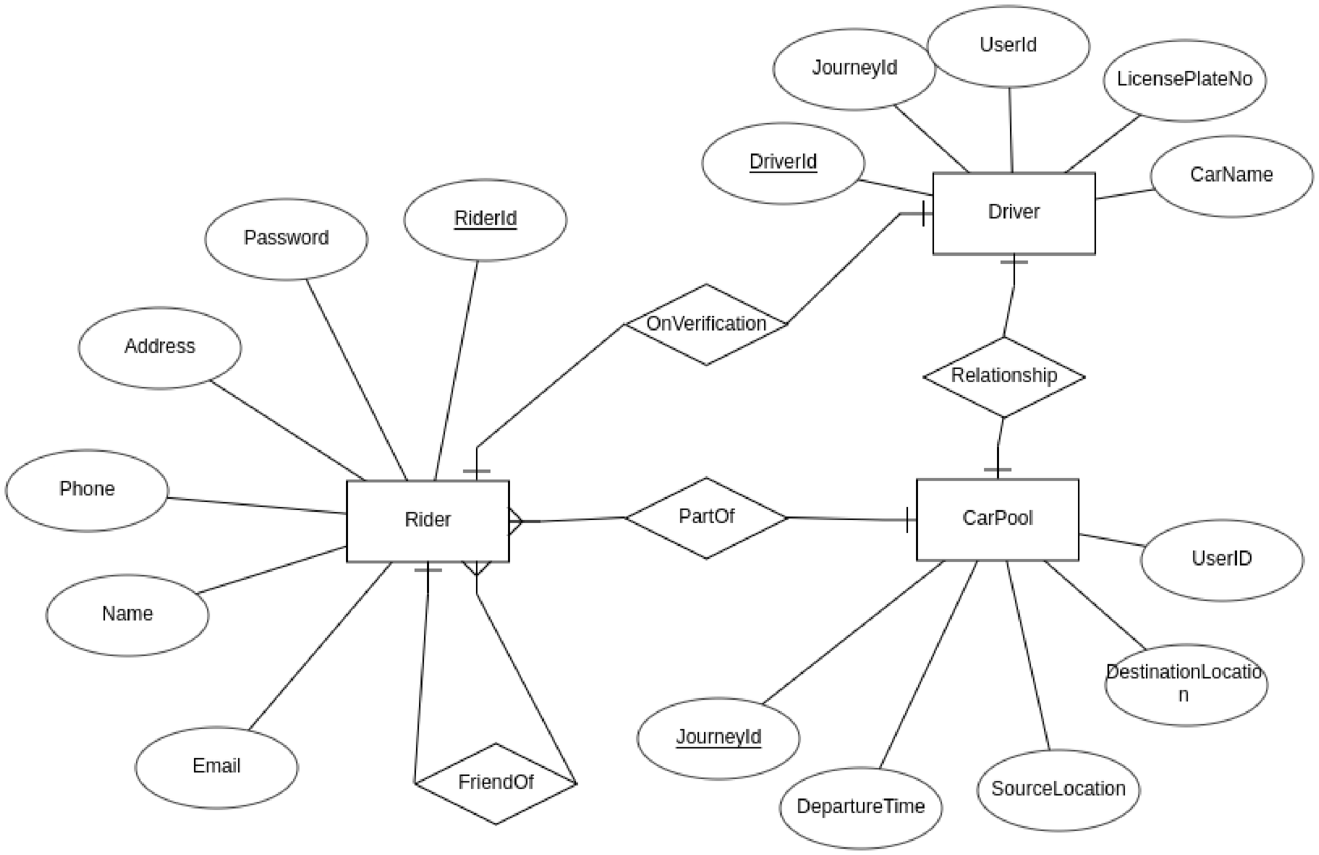


Fig 5.3: E-R Diagram

The project mainly contains 4 tables of which rider and driver wishlist are with mainly the primary key of rider as username, primary key of driver as driver_id. Other tables include Rating and Status of rides with respective primary keys. Rides, History and Type of ride can be controlled and managed by the user. Driver status is monitored by Rider and Rider status is monitored by Driver. Userlog monitors the sign in and log out of the registered users. Driver requests and Fellow passenger requests can be done by the Rider user. Drivers can ping their location and check for riders nearby by real time location stored of each requested user. Each ride may or may not get other riders. Each driver may or may not get many requests from riders. Users can accept or deny the rides. Users can also Cancel ride requests. Each User has a user rating entered by other users. Users place the requests. Requests are tracked by Status_monitors.

5.6 USE CASE DIAGRAM :

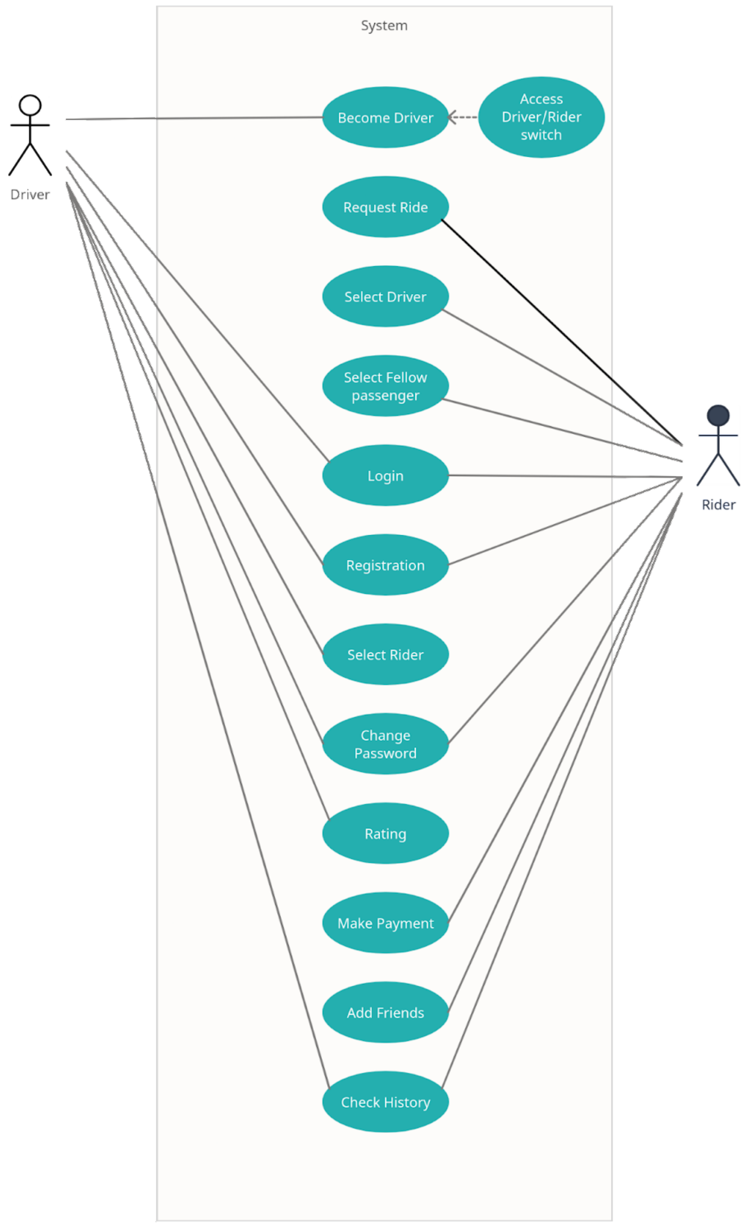


Fig 5.4: Use Case Design

Polo-S supports this use case:

• *User Characteristics*:

Primary Users should be familiar with the terms like login, register, ride, ratings, etc.

Users registered as Drivers should be familiar with terms like proximity, location mapping, payment, etc.

• *Principle Actors*:

2 Principle Actors are Driver and Rider.

5.7 DATABASE DESIGN

A database is a storehouse of data used by a software system. Data is stored in tables within a database. Several tables are created to manipulate data in the system.

The two required settings are:

Primary Key - field in the database, unique for every record occurrence.
Foreign Key - field that sets relation between tables.

Normalization is a technique for avoiding table redundancy.

5.8 SYSTEM ARCHITECTURE DESIGN:

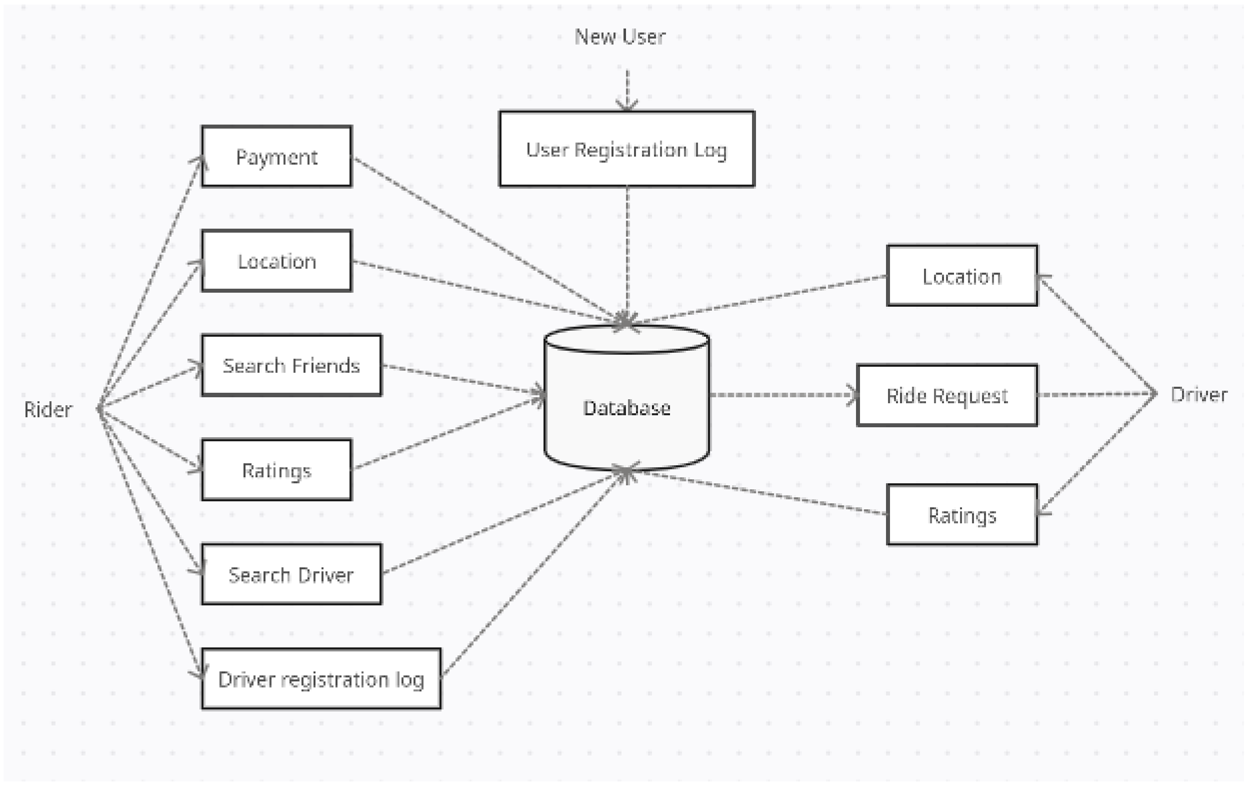


Fig 5.5: System Architecture Design

The process of defining a software system's overall structure, components, and interactions is known as system architecture design. It involves making deliberate choices about how the system's many components will be put together, integrated, and work as a unit to accomplish the intended functionality, performance, and scalability. Designing the system architecture includes determining the important modules, their roles, and the interfaces that connect them. It also requires making the right technology, framework, and pattern choices while taking flexibility, maintainability, security, and cost-effectiveness into account to make sure the system satisfies both functional and non-functional criteria. Laying the groundwork for a reliable and effective software system that can be effectively implemented, maintained, and expanded over time requires careful consideration during the system architecture design process

CHAPTER 6

DEVELOPMENT

6.1 MODULAR DESCRIPTION:

Root page is the Index page. This is the home page that displays all the types of rides that are available at the current moment. There is a navbar that navigates to Home Screen, Driver-Rider switch, Friends and Ride. There is an icon menu that opens up 3 options: Settings, Offers(not for driver), History of rides.

<u>There are 2 panels: User and Driver panel</u>

Driver:

All drivers who wish to become drivers have to firstly register as riders. After which the driver can then enable the switch to driver option by adding the required documents and information for becoming a driver in the settings page. The Drivers will have a map of the region and the rider requests will be mapped on the map in the drivers close proximity. This map will be directly provided in the Home page for drivers. Drivers can choose the riders by selecting them and viewing their profiles.

Rider:

- Registration module: Used for registering the new users of the system.
- Login module: Used for managing the login details.
- Ride module: Used for requesting rides and managing the ride details.
- Payment module: Used for managing the details of payment.
- History module: Used for managing the Previous and upcoming ride history.

6.2 SET CODING STANDARDS

To improve the readability of code,

- Used appropriate naming conventions,
- To understand and maintain code, headers of different modules aligned with a singular format.
- We used different identifiers for various purposes.
- We wrote comments at describing the code function at various points in the script

6.3 ENVIRONMENT SETUP

We are using Android Code as our Environment

Android Studio provides:

- Intelligent Code editor
- Android SDK manager
- Instant app support
- Integration with Firebase

6.4 SOURCE CODE CONTROL SETUP

- We use GitHub to track and manage our Code
- GitHub has in-house version management and other features to ensure Source code control
- It is widely used among developers

CHAPTER 7

TESTING

7.1 TYPES OF TESTINGS DONE:

• **Functional testing**:

Functional testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements. Functional testing mainly involves black box testing and it is not concerned about the source code of the application. This testing checks User Interface, APIs, Database, Security, Client/Server communication and other functionality of the Application Under Test. The testing can be done either manually or using automation.

• **Unit testing**:

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

• **Integration testing**:

Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Integration Testing focuses on checking data communication amongst these modules.

7.2 RIDERS INTERFACE:

Registered users can log-in to their account using their email/phone number and password and after logged in users can view their ride page for selecting the type of ride with the pickup and drop locations.

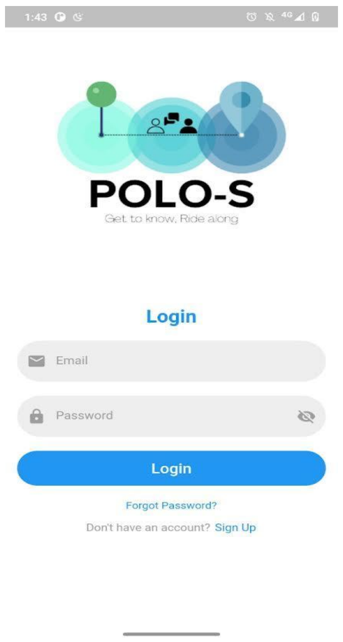


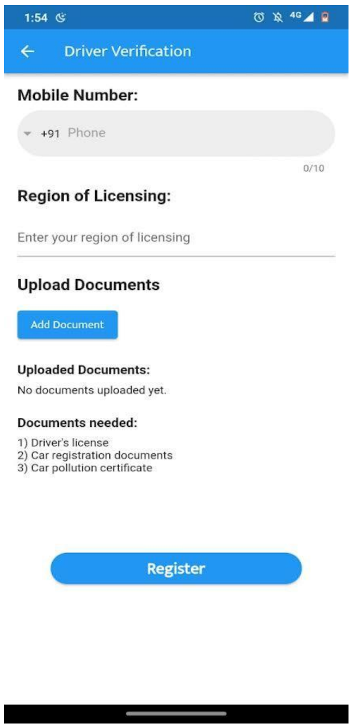
Figure 7.1: User Login page

The user is redirected to the home page with their account already logged in after successfully logging in. On the home page, a new profile icon will be available for viewing and editing the profile. The ride option is available in the bottom navigation bar, from which the user can order a taxi using one of the three modes offered and check the status of their ride. It has a Friends section where we can see every friend who has been added and has shared rides with us up to the current date. It has a switch-to-driver option, but it won't work until you present the necessary paperwork. An extra drop down menu provides an extra distance traveled feature that helps you in getting coupons that can be used in for claiming rider discounts.

The interface of settings is common to both the rider as well as driver which consist of all options including the sign-out option.

7.3 DRIVER INTERFACE:

This page directs to the Driver Home. Drivers get a map of their realtime locations and the riders or set of shared riders as ping marks on the map. The Drivers can then select which riders to accept rides from. This interface is enabled only after submitting required documents and information for becoming a driver.



Mobile Number:

Region of Licensing:

Enter your region of licensing

Upload Documents

Uploaded Documents:

No documents uploaded yet.

Documents needed:

1) Driver's license
2) Car registration documents
3) Car pollution certificate

Figure 7.2: Driver Document Request Page

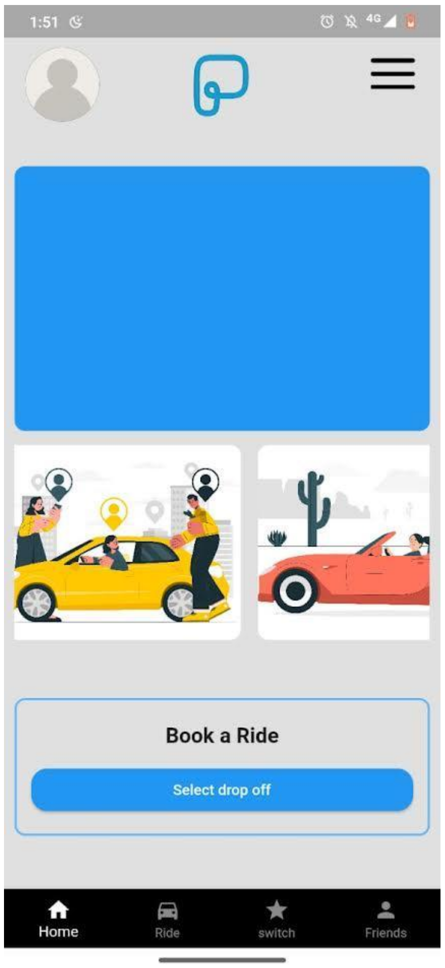


Figure 7.3: Rider Homepage

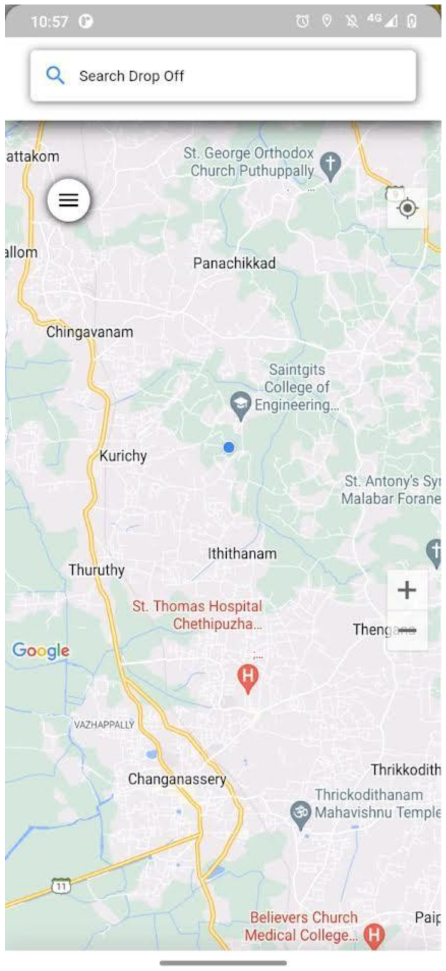

10:57
Search Drop Off
Panachikkad
Chingavanam
Saintgits College of Engineering
Kurichy
Ithithanam
Thuruthy
St. Thomas Hospital Chethipuzha
Changanassery
Thrickodithanam Mahavishnu Temple
Believers Church Medical College
Google

Figure 7.4: Rider location page

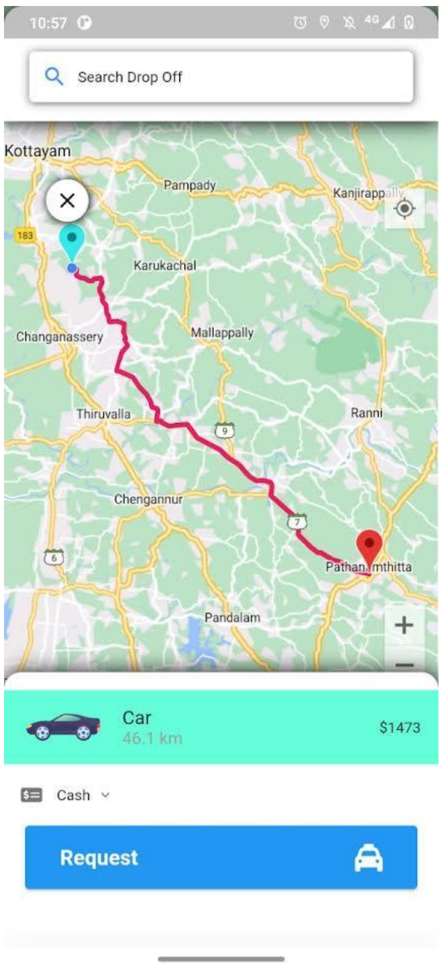


Figure 7.5: Rider search (ping) page

7.3 FUNCTIONAL INTERFACE:

Test Cases	Test result	Tested By
LOGIN SCREEN		
Is the logo visible?	Pass	Midhun
Is the email text box working?	Pass	Midhun
Is the password text box working?	Pass	Midhun
Is the forgot Password link working?	Pass	Midhun
Is the signup link working?	Pass	Midhun
Is the Login button working?	Pass	Midhun

REGISTRATION SCREEN		
Is the Name and email text input working?	Pass	Midhun
Is the Phone number input text working?	Pass	Midhun
Are the address box, Password, Confirm Password working?	Pass	Midhun
Is the Register button working?	Pass	Midhun

HOME PAGE		
Is the ride button working?	Pass	Naveen
Is the List View for items scrollable?	Pass	Naveen
Is the switch button working?	Pass	Naveen
Is the friends button working?	Pass	Naveen
Is the Polo logo being displayed?	Pass	Naveen
Does the select drop off take you to location input?	Pass	Naveen
Are all the alignments done right?	Pass	Naveen
Is the menu drawer working?	Pass	Naveen
Is the profile picture being displayed?	Pass	Naveen

SETTINGS PAGE		
Is the Profile Picture aligned correctly?	Pass	Nikhil
Is the user name and phone number displayed?	Pass	Nikhil
Does the verify driver button work?	Pass	Nikhil
Does the sign out button work?	Pass	Nikhil
Are all the button spaces kept properly?	Pass	Nikhil

RIDE PAGE		
Is the Page displaying the map?	Pass	Nikhil
Are the locations displayed right?	Pass	Nikhil
Are the pickup and drop location being marked on map?	Pass	Nikhil
Is the minimum distance between both points being shown by Line?	Pass	Nikhil
Is the start journey button working?	Pass	Nikhil

INTEGRATION TESTING		
Do all the modules work together properly?	Pass	Midhun
Upon clicking signin does the homepage load?	Pass	Midhun
Are all transitions errorfree?	Pass	Midhun
Are the details for all users being saved in flutter?	Pass	Midhun

DRIVER VERIFICATION PAGE		
Is the country selection working?	Pass	Naveen
Is the number validation working?	Pass	Naveen
Are documents being uploaded?	Pass	Naveen
Is the text box for region of licensing accepting values?	Pass	Naveen
Are all elements aligned in order?	Pass	Naveen

UNIT TESTING		
Do all pages run individually?	Pass	Midhun
Upon clicking signin does the homepage load?	Pass	Midhun
Are the screen being overlayed on clicks?	Pass	Midhun

LOCATION INPUT SCREEN		
Is the size ok?	Pass	Midhun
Is the pickup bar working?	Pass	Midhun
Is the drop down location bar working?	Pass	Midhun
Is the location filtering working?	Pass	Midhun
Is the set location working?	Pass	Midhun

CHAPTER 8

RESULTS AND DISCUSSIONS

8.1 RIDER PANEL

8.1.1 HOME PAGE

This is the first page that appears when we open the application. It has a navigation bar where we can switch for selecting rides, open friends list, switch to driver(authorized users). It has a drop down menu for getting into settings and Ride history. Login page helps the user to login to the account. If the user has already logged in to his account, he/she can view ride history, track current rides, contact friends and modify his/her profile. If the user hasn't already signed up, then user must register his account, in order to get these services.

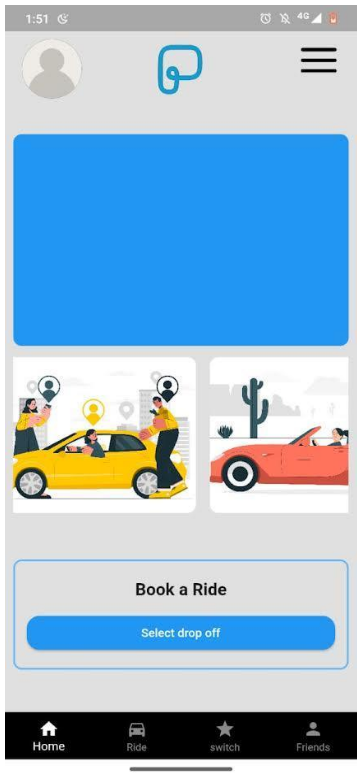


Figure 8.1: Home Page

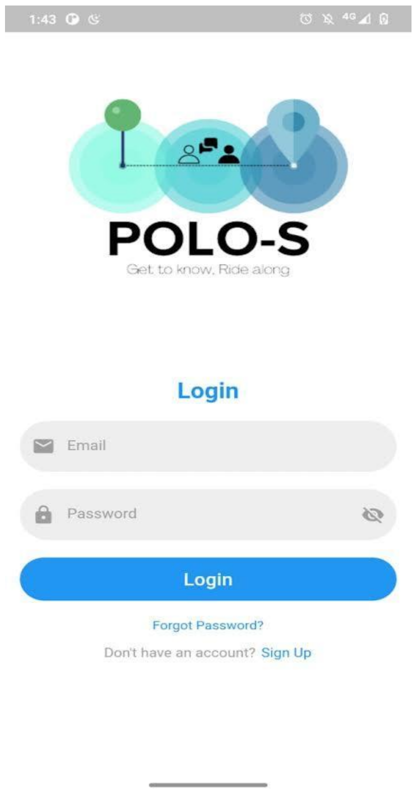


Figure 8.2: Login Page

8.1.2 LOGIN PAGE

The login page is the page where the existing user can login in with their account to avail the ride services. Riders can choose ride options only when the user logs into his/her account otherwise it is not possible.

The ability to create a new account is granted to users who are brand-new to the application. Name, email address, phone number, password, and other user information are required for registration. To avoid errors, the user must enter accurate information into their account.

After logging in, it redirects the user to the home page with their account logged in. There will be an additional profile icon in home page for viewing and changing the profile and the adding addresses of the user. The switch to drivers can be done through settings.

8.1.3 RIDE PAGE

Ride Page with a Map improves the user experience by giving a visual representation of the ride's route. The map shows the entire route in addition to the crucial information like the places of departure and arrival, the date, and the time. Users can quickly view the route, potential stops, and estimated travel time. Passengers can better plan their journey and visualize the journey ahead of them thanks to this interactive map feature, which adds convenience and clarity. The carpooling application provides users with an intuitive and user-friendly way to comprehend the route and make decisions about joining or creating rides by integrating the map into the Ride Page.

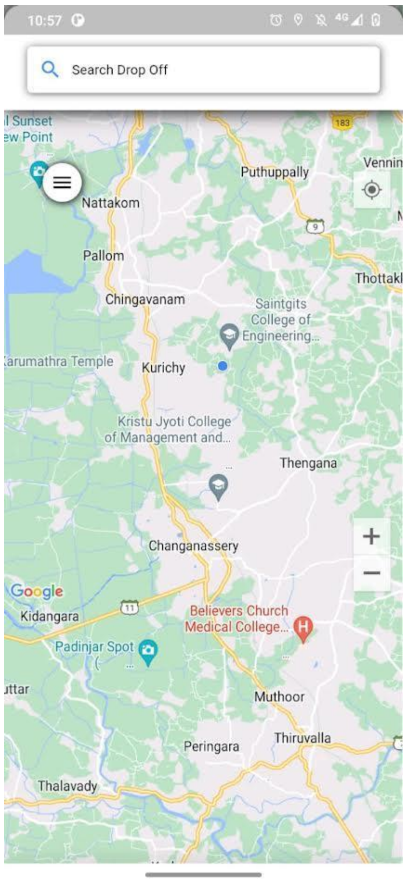


Figure 8.3: Ride page

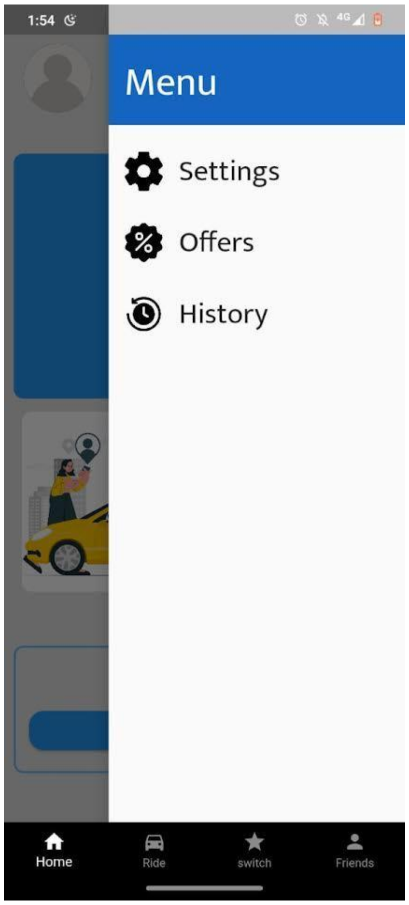


Figure 8.4: Home page drop screen

8.1.4 SETTINGS PAGE

The home screen's drop menu can be used to access the settings page. It comprises of five options that offer the users several alternatives.The features mostly consist of signout, password changing, address addition, driver verification, and friend management.

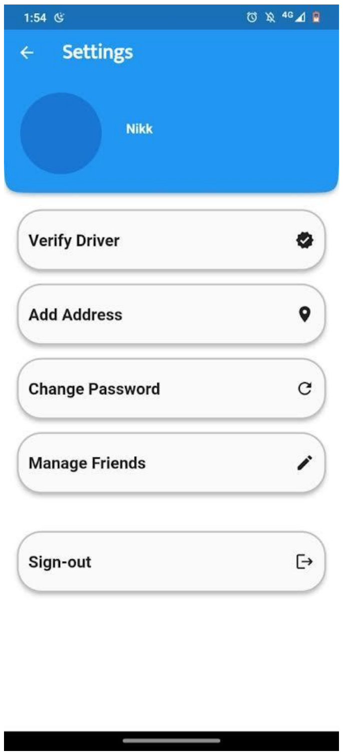


Figure 8.5: Settings page

8.1.5 OFFERS PAGE

The major component of it is a track meter that measures the overall distance you have covered with Polo. The acquired trip distance can be used as points to redeem a variety of offers depending on the time frame from this location.

8.1.6 LOCATION INPUT PAGE

The user has two input options on this page. The drop-off site is one, while the pickup place is another. Additionally, it features a built-in automatic location filter depending on typing.

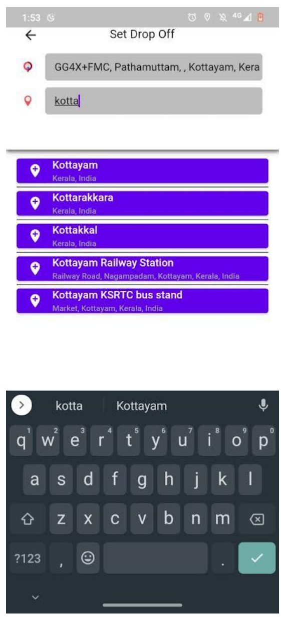


Figure 8.6: Location input page

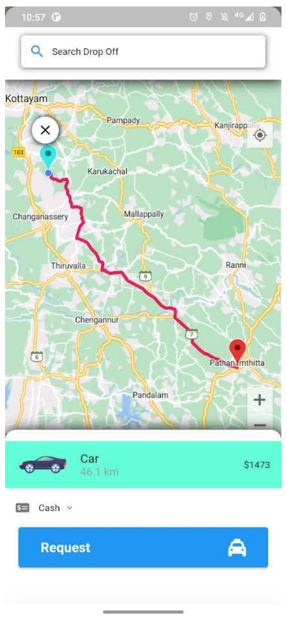


Figure 8.7: Map Points page

8.1.7 MAP POINTS AND SEARCH PAGE

This section displays the positions of pointed spots on the map along with their distances from one another. Here, a second search option is visible to look for nearby users who have the same pickup locations as the user's pickup and display them on a map. The users can then request them for ride sharing by looking at their profile.

CHAPTER 9

CONCLUSION

In conclusion, the integration of Flutter, Firebase, and Google Maps API offers a powerful platform for developing a carpooling application with searched and accepted people. By leveraging these technologies, carpooling services can provide a seamless and efficient experience for users seeking to share rides. With Flutter's cross-platform capabilities, the application can be developed for both Android and iOS, ensuring a wide reach to potential users.

The combination of Firebase and Google Maps API further enhances the carpooling experience. Firebase provides a reliable and scalable backend infrastructure, allowing for real-time communication and data synchronization between drivers and passengers. The integration of Google Maps API enables precise location tracking, route optimization, and interactive mapping functionalities, ensuring users can easily search for nearby drivers or passengers and view the optimal route for their journey.

Overall, the amalgamation of Flutter, Firebase, and Google Maps API empowers carpooling services to create a user-friendly and feature-rich platform. By connecting individuals seeking rides with compatible drivers or passengers, this technology stack offers an efficient and eco-friendly solution for commuting, reducing traffic congestion, and fostering a sense of community through shared transportation experiences.

CHAPTER 10

REFERENCES

[1] https://firebase.flutter.dev/

[2 https://drawsql.app/teams/polos/diagrams/polos

[3] https://erdplus.com/

[4 https://app.creately.com/

[5] https://online.visual-paradigm.com/diagrams/solutions/free-activity-diagram-tool/

[6] https://console.cloud.google.com/apis/

[7] https://www.javatpoint.com/adding-firebase-to-app

[8 https://docs.flutter.dev/

[9] https://pub.dev/

[10] https://firebase.google.com/docs

[11] https://www.javatpoint.com/sql-tutorial

[12] https://codelabs.developers.google.com/codelabs/google-maps-in-flutter#0