LAIBA IHSAN
01-135201-032

Smart Carpool

Bachelor of Science in Computer Science

Supervisor: Dr. Asghar Ali Shah

Department of Computer Science
Bahria University, Islamabad

December 2023

Certificate

We accept the work contained in the report titled “*Carpoolify*”, written by Mrs. LAIBA IHSAN, as a confirmation of the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Dr. Asghar Ali Shah

Internal Examiner:

External Examiner:

Project Coordinator: Dr. Faisal Imran

Head of the Department: Dr. Arif ur Rehman

October 20th, 2023

Abstract

In the rapidly evolving need for transportation, the Carpool App has witnessed the remarkable growth, offering convenience and accessibility for transportation in need. However, the current rise in fuel prices, increasing environmental concerns and the fall in the economy has made car ownership a dream for many salaried individuals, students and other citizens. In response, Car- pooling industry has emerged as a hope, providing a cost-effective solution by sharing empty seats of car hence, eliminating the need of car ownership. In this Project, I aim to provide a technology solution to tackle problems in theconventional Carpool paradigm by using the powerful impact of blockchain technology. By using Blockchain, I aim to create a fair and secure payment method where passengers and drivers can interact directly without anyintervention. They often don't get their full share of money due to third party involvement. [1] In this regard, decentralization eliminates the need forcentral authority. This benefits both passengers and drivers in terms of saving money, ensuring privacy, and enhancing security. I tested the idea using Ethereum Blockchain and executed Smart Contracts to facilitate seamless payment method. This model has gained popularity in urban areas because it allows people to virtually own a car by paying a small fare. It's a cost-effectiveand advantageous solution for both passengers and drivers by merging the Blockchain technology with Carpool App.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to the Almighty. I would also like to extend my heartfelt appreciation to my supervisor, Lecturer Mr. Asghar Ali Shah, a Senior Lecturer at the Department of Computer Science, Bahria University, Islamabad. My profound thanks go out to my parents, families, friends, and all my teachers who have guided me throughout our academic journey.

I am deeply indebted to my course instructors who have imparted knowledge and skills that have brought me to this significant moment. Without their valuable contributions, the completion of this project would not have been possible.

LAIBA IHSAN
Islamabad, Pakistan

December 2023

“We cannot solve problems with the kind of thinking we employed when we came up with them.”
Albert Einstein

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

OTP One-time password
UX User Experience
APIs Application Programming Interfaces

Chapter 1

Introduction

1.1 Overview

Due to hike in petrol prices, owning and maintaining a car is expensive. Also, a lot of cars on the road results in traffic congestion and increasesthe amount of fuel burned by the cars. Hence people are looking for alter- nate ways of transportation that are affordable and convenient. Carpooling App provides a solution to reduce the transportation cost by sharing the empty seats of the ride with other passengers. [4]. While there are numer- ous ride-sharing applications available, my objective is to eliminate the needfor third-party involvement and enhance payment transparency to prevent fraud. [5] By sharing rides with other passengers, fuel consumption is signifi- cantly reduced, contributing to environmental sustainability and cost savings for everyone involved. The main purpose of this app is to provide two inter-faces to individuals, driver interface and passenger interface and to provideease to passengers to request shared rides or go for individual rides. Addi- tionally, this Carpooling App enables users to connect with nearby drivers within minutes, eliminating the need to wait for traditional taxis and en- hancing convenience.

1.2 Problem Description

Carpool App is a convenient means of transportation across many cities. Before the advent of carpool apps or ride sharing apps, the traditional taxi systems had their own set of problems. Meanwhile, the world has already shifted its transportation means towards ride sharing apps but they majorly include the third-party involvement. The need to develop a carpool app arose due to following challenges:

1. Centralized Payment methods do not facilitate direct peer-to-peer trans- actions between drivers and passengers as they include intermediary fees and commissions.
2. Centralization rely on a single central authority which require multiple resources to manage and operate the system.
3. User's transactions data is collected and stored in centralized system. This increases the risk of data breaches, raises concerns for user privacy and security. Moreover, users have limited visibility into the processes. This undermines their trust into the system.
4. Before Ride sharing apps, there were more cars on the road, increases the traffic congestion and air pollution due to more fuel consumption.

5. Significant number of empty seats leads toward the under utilization of resources, hence indicating the need for optimization and efficiencyby sharing rides.

1.3 Objective

My primary objective is to develop a carpooling application that helps people reach their destinations at the lowest possible fares and prevents the under utilization of resources. Here are the key aspects of my app:

1. To connect users who share similar routes, allowing them to share rides and reach their destinations on time. By utilizing the available seats in cars, optimizing transportation resources and reduce the number of vehicles on the road.
2. To eliminate the need for third-party in payment method, ensuring that users' personal data remains confidential. By leveraging the decentralized nature of blockchain, I enhanced the security and privacy of the users' information.
3. Carpooling with fellow passengers helps reduce traffic congestion by decreasing the number of individual cars on the road. As a result, transit times are also reduced, providing a more efficient and time- saving travel experience.

4. I prioritize the safety of our users by implementing real-time location tracking. This feature ensures that drivers and passengers can track the progress of their journey, providing peace of mind and security throughout the ride.
5. The app offers the flexibility for users to choose their carpooling com- panions based on gender preferences. This feature addresses the com- fort and safety concerns of passengers, allowing them to select ridesonly with females or males, as per their preference.
6. To streamline the carpooling process, this app allows users to schedule rides for weeks or even months in advance. This feature eliminates the hassle of posting trips multiple times for the same destination, providing convenience and efficiency for both drivers and passengers.

By focusing on these objectives, I aim to create a carpooling app that pro- motes efficient, cost-effective, safe, and convenient transportation while pri- oritizing security, privacy, and user preferences.

1.4 Scope

The core purpose of the application is to decentralize payments through blockchain, ensuring safe and transparent transactions. It aims to provide an affordable and convenient mode of transportation, especially when peo-ple are grappling with rising transport expenses. Car owners can post ride requests for their routes, and passengers can easily find others heading in the same direction. Passengers have the flexibility to request individual rides or schedule them in advance. Drivers can view scheduled rides and decide whether to accept or decline passenger requests. The application places a strong emphasis on safety by incorporating real-time location tracking for both passengers and drivers.

1.5 Feasibility Analysis

1. Risks Involved:

    - Fraudulent Fake requests can be made by the passenger.
    - Driver can demand for more fare. This will be catered as our app will calculate the fare for the particular route according to current petrol prices.

2. Resource Requirement:

    - Laptop.
    - Internet Connection
    - Digital Wallet

1.6 Constraints

Peer Connect Application requires an active internet connection to display accurate routes for emergency contact. The Application uses Blockchain technology, there might be some delays in transaction processing that may cause latency issues. Moreover, In Pakistan people are not familiar with blockchain technology, that may cause trust issues among users. Moreover, users of application might need the sufficient storage space in their device to install the application. Additionally, the gas fees associated with blockchain transactions can be relatively high that may affect the cost-effectiveness of using the application.

1.7 Stakeholders

Stakeholders involved in this project are:

1. General Public:
   The general public are the primary stakeholders who will be using the application for their daily transportation needs. These individuals will download and utilize the app to share rides and travel to their desired destinations efficiently and conveniently.They can have the role of a driver or a passengers based on their needs they can register for both profiles.
2. Admin:

The admin represents a crucial stakeholder responsible for overseeing and managing all aspects of the application. This includes monitor- ing ongoing rides, tracking fare transactions, and ensuring the overall smooth functioning of the platform.

3. Staff Members:
   The staff members are essential stakeholders who play a vital role in the successful operation of the application. They are responsible for managing all the rides, ensuring a seamless experience for both drivers and passengers. Their involvement helps in providing customer sup- port and resolving any problems that may arise during the use of the application.

1.8 Solution Application Areas

The proposed project will be valuable for general public. People will be able to earn and will be able to travel across places within less amount. Hence, the app is providing a long-term solution to optimize the utilization of privatecars on the road. It would be cost effective solution for general public.

Chapter 2

Literature Review

To gain a comprehensive and detailed understanding of Carpooling apps and the associated challenges,it’s important to read and study what experts and others have written about them in books, articles, and reports. This reviewof literature helped to learn from what’s already been done and figure out what I need to know to build my own carpooling app. The knowledge gained through this literature review helped to make better decisions for developing an cost effective and user friendly carpooling app.

2.1 Why Carpooling?

According to the researchers that the composition of carpool groups playsa significant role in the formation of carpools. There are different types of carpooling exists. One of them is ”family pools” which are easily formed because the participants are family members. Another type is ”co-worker carpooling,” which involves employees who know each other from the same workplace. Therefore, carpooling can be done among friends, family members, colleagues or among mutual friends.

2.1.1 A Research Hypothesis

Type of factor	Factors
Demographic	Age, income, number of people in household, number of cars in household
Psychological	Saving money and time, re- duce congestion, sustainability, comfort, convenience, socialising, trust
Interventions	Parking availability, parking cost, finding potential partner, reserved parking, high occupancy vehicle lanes
Situational	Fixed/regular work schedule, commute distance, time commuting, population density, fuel costs

Table 2.1: Factors in the table based on sources [3].

2.1.2 Carpooling Impact

Various Researches indicated the advantages of sharing rides, including reducing fuel cost, saving time and environmental sustainability. These advantages encompass easing traffic congestion, curbing energy consumption, and lowering carbon emissions [6]. Consequently, people should be motivated to foster carpooling. Before, policymakers have expressed a need for deeper insights into the psychological factors that drive carpooling participation, with the aim of refining their strategies to promote and encourage carpooling.

2.2 Idea Generation

In this current evolving time, the cost effective transportation is a basic need of every individual. The idea behind the carpooling addresses the problems related to traffic congestion, fuel consumption and cost savings. As it can be seen there are a lot of carpool apps exist, this project enhances the idea behind those carpool apps. [1]
My project not only creates a carpooling app but also leverages the blockchain technology for making secure payments between driver and passenger. By integrating ethereum blockchain wallet, the carpool applications can provide convenient and seamless payment integrations.

2.3 Blockchain Technology

A blockchain is a decentralized and distributed digital ledger that records all transactions occurring within a network. It consists of number of nodes and each node has its own list of transactions. Each block contains a unique identifier, namely hash and hash of the previous block. It operates in a de-

centralized manner, meaning that copies of the ledger are replicated among all participants in the network, who collaborate with each other. Whenever a new block is added to the chain, it cannot be altered and tampered. [7] Decentralized consensus mechanisms including Proof of Stake and Proof of Work ensures the transparency, integrity and security of the network. Ethereum is a decentralized blockchain platform that allows developers to build an deploy their DApps and smart contracts, and it serves as a digital currency (ETH) for DApps. [8] This currency compensate miners for validating transactions and executing smart contracts on the network.
Miners are the participant in the network who validate and add new transactions in the blockchain. When users initiate transactions, the network might contain the pool of transactions. Miners compete with each other to perform Proof of Work Consensus mechanism. The first miner to solve the puzzle get the right to add the transaction to the blockchain. In return of this, miners get some cryptocurrency (Ethereum) and transaction fee from the transactions they added to the network.

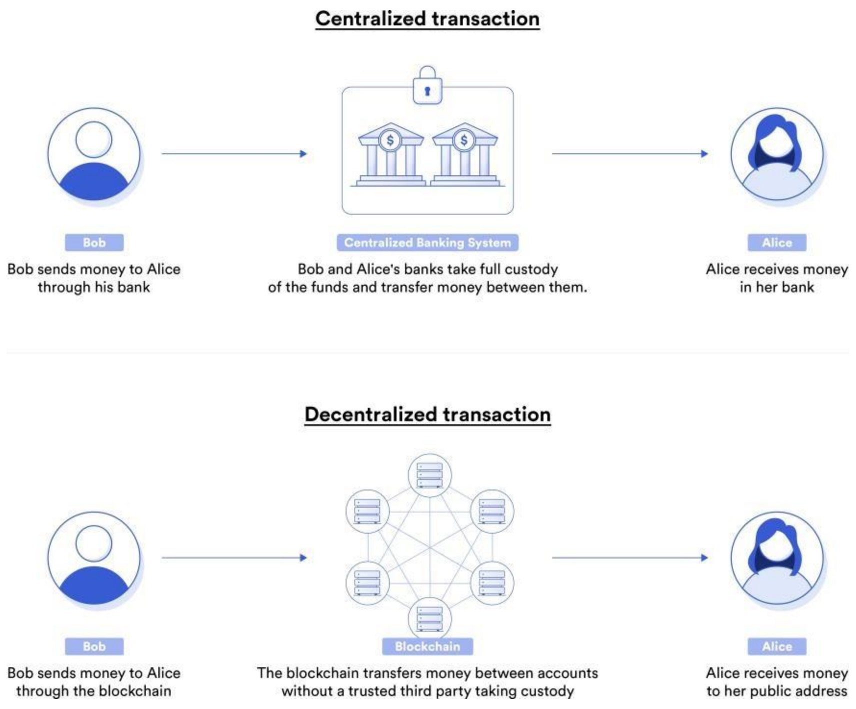

Centralized transaction
Bob
Bob sends money to Alice through his bank
Centralized Banking System
Bob and Alice's banks take full custody of the funds and transfer money between them.
Alice
Alice receives money in her bank
Decentralized transaction
Bob
Bob sends money to Alice through the blockchain
Blockchain
The blockchain transfers money between accounts without a trusted third party taking custody
Alice
Alice receives money to her public address

Figure 2.1: Figure is based on sources [2]

2.4 Smart Contracts

Blockchain uses smart contract to provide controlled access to ledger. Smart contract is a key mechanism for encapsulating information and keeping it simple across the network they also allow the participants to execute certain transactions automatically. In a smart contract, contract clauses written in computer programs will be automatically executed when predefined con- ditions are met. Smart contracts consisting of transactions are essentially stored, replicated, and updated in distributed blockchains. In contrast, conventional contracts need to be completed by a trusted third party in a centralized manner consequently resulting in long execution time and extra cost. The integration of blockchain technology with smart contracts will make the dream of a “peer-to-peer market” come true.

2.5 How Blockchain Payments work?

Blockchain is a decentralized system that enables the network of computers to maintain single, updated and secure ledger. [7] Ledger is a digital file that keeps the track of all transactions performed in the network. This file is not stored on any central entity but is distributed across the chain of computers, nodes. For a single transaction to be performed, a message is broadcast to the network including the amount of Eth coins. For this transaction, the deduction from the sender wallet should be done and the same amount in the receiver’s wallet should be increased.
Additionally, in order to perform transactions, a digital wallet is required that allows to store the digital currency i.e Ethereum. Each wallet is protected with cryptographic methods that uses unique pair of connected keys i.e Public and Private keys. If a message is encrypted with the public key, the owner of private key would be able to decrypt and read the message, vice versa. Each node in the network checks the transaction request is coming from legitimate node by decrypting the message with public key of his wallet.

2.6 Existing Systems

There are many applications that provide the ride sharing facilities.There are some similar applications that provide the ride sharing facilities like my idea. I took inspiration from those systems and implement a few more features to make it more user friendly and cost effective. Some of the existing systems are InDriver, Careem, Yango.

1. InDrive
   InDrive introduced its application with minimal features including bargaining of fare and allowing the users to choose the driver from the list. But it does not provide the payment method integration. Also it does not have the strong passenger profile. My application provides the seamless and secure payment method and a strong driver and passenger profiles.

Figure 2.2: InDrive

2. Careem
   Careem being the first Ride booking app became popular in Pakistan. But after some time, due to non-availability, Careem loses 40 percent of their customers for not having the rides on time. Careem still struggles to have their coverage in many cities because of not having enough cars or customers.

Figure 2.3: Careem

3. Yango
   Yango is newly launched ride booking app that provides route hailing, routing and navigation features. It only has the basic functionalities of ride Also, Yango provides seperate app for drivers and passengers.

YANGO

Figure 2.4: Yango

2.7 Analysis

Carpoolify is a mobile application that provides the facility of ride sharing to users. Users going towards the same location can hail a cab and shareit and at the end split the fare. The driver will receive the full share of fare. The payment method of application is based on blockchain that is adecentralized technology that means there’s no third party involved in the operation of payment transfer. The application facilitates users by offering lower fares and it facilitates the drivers by providing full share of fare as there’s no third party that needs it fair share. There are many cab hailing applications that are performing their operations successfully like Careem, Uber and InDriver etc. But they lack payment modules.

Chapter 3
Requirement Specification

A requirement specification is a comprehensive document that includes all the specific requirements that are to be imposed on the design and verification of the software. It not only covers the technical aspects but also includes, functional requirements, non-functional requirements, design constraints, and other quality attributes.

3.1 Application Overview

This Application will allow users to signup using their Emails. OTP willbe send to their email and upon verification, users will be allowed to make their profiles with generic information. Then the user will be asked to Login with OTP. User can also signup for the driver mode. For the driver's pro- file, a complete verification will be done based on their CNIC and Liscene. Ridesharing service will also be provided at the main page of application. User can opt for solo or can request shared rides based on their needs. User can also schedule the rides and these rides will be shown at the driver’s pro- file. Driver will be able to post carpool rides. Moreover, the app integratesthe secure payment gateway through blockchain technology.

3.2 Systems Requirement

The carpool app system requirements encompass a robust architecture and key features. The backend leverages Node.js for server-side logic, interfacing with a MongoDB database to store and manage user and driver profiles, ride information, and feedback. React Native, the cross-platform framework, powers the Android app’s frontend providing user interfaces for both passengers and drivers.

Geolocation and mapping Application Programming Interfaces (APIs), such as Google Maps, enable real-time location tracking and route optimization. Push notifications are integrated to inform users of ride requests, updates,and important alerts. A unique aspect of this app is its blockchain-based payment system, ensuring secure and transparent transactions.

In-app messaging and chat functionality allow seamless communication between passengers and drivers to provide a user-friendly experience. Reviews and feedback mechanisms are added to empower users to provide input on their ride experiences. The system adheres to the principles of scalability,data security, and user privacy, offering a reliable, feature-rich, and respon- sive carpooling solution.

3.3 Functional Requirements

3.3.1 User Signup

The application provides the signup for the user. The user will be asked to enter the email address for signup. On which an One-time password (OTP) will be send to the user to verify the email address.

3.3.2 User Login

The application offers a user login feature, where each time a user logs in, an OTP is sent to their registered email address for authentication. This login process will eliminate the need for users to remember passwords, enhancing the overall User Experience (UX).

3.3.3 Profile Management

The application enables the user to create and manage their profiles. User can provide their basic information to be recognized by others including, name, contact details, gender, work and home addresses.

3.3.4 Book a Ride

Users will have the option to book a ride by either manually entering their source and destination addresses or by selecting them directly from a map interface.

3.3.5 Search for nearby Drivers

When a user requests a ride, the app will initiate a search for nearby drivers and display their locations on a map interface, allowing the user to see the available drivers in their locality and choose the most convenient option.

3.3.6 Live Locations to contacts

Users have the capability to share their real-time location with their in app emergency contacts. This feature allows users to enhance their safety by providing their chosen contacts with live updates about their whereaboutswhen necessary.

3.3.7 Payment Preferences

The app will offer users the flexibility to choose between two payment methods: making payments directly from their blockchain wallet or opting for cash payments.

3.3.8 Notifications

Both users and drivers will receive notifications regarding carpool rides, including ride confirmations, updates, and cancellations. This notification system ensures that all parties involved are kept informed and can effectively manage their ride-sharing arrangements.

3.3.9 Ride Sharing

Users can search for carpool rides shared by drivers and sendrequests to occupy available seats in the car. This feature enables a stream-lined process for users to find suitable rides and request a spot in the vehicle.

3.4 NonFunctional Requirements

3.4.1 Response time

The app is designed to provide early response times for basic operations such as searching for rides, navigating through the app, and booking rides. These improved response times are achieved through the utilization of React Native components, which enhance the efficiency and speed of these key functions. By doing so, users can experience a more responsive and seamless interaction with the app.

3.4.2 Security

Following are the ways in which I enhanced the security of my App:

User Authentication

To authenticate users, the app employs an Email OTP verification method. Upon user registration or sign-up, a unique OTP is sent to the user's registered email address. After successful verification, a token is generated, users gain access to the app's features and services. This authentication process enhances security and ensures that only verified individuals can utilize the application.

Secure Transaction

I enhanced the payment security by integrating the blockchain payment wallet with in the app. For each user who want to make transactions, a private key is generated for secure identity and public key.Once the transaction is agreed between the users, it needs to be approved, or authorised, before it is added to a block in the chain. [9]

Token based Login

The token that is generated at the time of Signup, is validated each time user logins. This validation mechanism ensures that only authenticated users with valid tokens can access the application's services,

3.4.3 Compatibility on Android version 12 or higher

The app is designed on minsdkVersion 31, which is suitable for Android version 12 and future versions. This way compatbility is ensured for carpool application with current and future versions of android.

3.5 Use Cases

3.5.1 Splash Screen

Use Case ID	UC-001
Use Case Name	Splash Screen
Actors	User
Description	The splash screen will be displayed to user after use clicks on the app icon immediately
Trigger	User launches the application
Precondition	User should have installed the app.
Basic Flow	User launches the app by clicking on the icon. Immediately, a Welcome splash screen is displayed to user.
Postcondition	The app is now launched and ready to use by user.

Table 3.1: Splash Screen UseCase

3.5.2 Signup

Use Case ID	UC-002
Use Case Name	Signup
Actors	User
Description	The user is allowed to signup in the application using email and an OTP will be send to user on their email.
Trigger	User enters the email.
Precondition	App is already launched by user. And user is creating the account first time.
Basic Flow	User enters the email in the input field and then click on the verify button. A code has been send to user on their email. A verification screen will appear to user. User will enter the valid verification code. App will verify whether user entered code is matched by the code send to the user.
Postcondition	Email has been verified.

Table 3.2: Signup UseCase

3.5.3 Profile Creation

Use Case ID	UC-003
Use Case Name	Profile Creation
Actors	User
Description	User can create their passenger profile by entering their information including firstname, lastname, gender, contact detail, homes address and work address.
Trigger	User fill all the input fields.
Precondition	Email has been verified
Basic Flow	User enters the all required input fields and then click on the create profile button.
Exception	User has not entered required fields. The app will throw an error ”All fields are required”
Postcondition	Profile has been created and saved in Database with email.

Table 3.3: Profile Creation UseCase

3.5.4 Login

Use Case ID	UC-004
Use Case Name	Login
Actors	User
Description	The user is allowed to Login in the application using email and an OTP will be send to user on their email.
Trigger	User enters the email.
Precondition	App is already launched by user. And user is Already registered in the database.
Basic Flow	User enters the email in the input field and then click on the Login button. A OTP has been send to user on their email. A verification screen will appear to user. User will enter the valid verification code. App will verify whether user entered code is matched by the code send to the user.
Postcondition	Login has been logged in.

Table 3.4: Signup UseCase

3.5.5 Book Ride

Use Case ID	UC-005
Use Case Name	Book Ride
Actors	User
Description	The user wants to book a ride and provides details such as source, destination, car type, number of passengers, and scheduling prefer- ences.
Trigger	User selects the ”Book Ride” option in the app.
Precondition	The app is already launched, and the user is logged in.
Basic Flow	1. User selects the ”Book Ride” option. 2.User enters the source and destination ad-dresses. 3. User selects the car type from available options. 4. User specifies the num- ber of passengers. 5. User sets the schedul- ing for the ride (immediate or scheduled). 6. User confirms the booking.
Postcondition	The ride is booked successfully.

Table 3.5: Book Ride Use Case

3.5.6 Google Places Autocomplete

Use Case ID	UC-005
Use Case Name	Autocomplete
Actors	User
Description	User will enter their source and destination addresses in input fields where Autocomplete api has been called. After entering 3 key-words, user will get location suggestions
Trigger	User entered the source and destination ad-dresses.
Precondition	User has valid login in the app
Basic Flow	User enters the the source and destination addresses in input fields. The autocomplete api will show location suggestions to user af- ter entering 3 keywords. User will click onthe desired location.
Exception	The subscription plan of google maps is ex-pired or user has exceeded the per click limit of the plan.
Postcondition	Source and destination addresses has been saved to databases.

Table 3.6: Google places autocomplete UseCase

3.5.7 Choose Location on map

Use Case ID	UC-006
Use Case Name	Choose Location on Map
Actors	User
Description	The user will enter location coordinates onmap including source and ip addresses. The markers will be displayed on the map and addresses lattitudes, longitudes will be saved in the databases for future use.
Trigger	User click on the map.
Precondition	Map is displayed with valid user login.
Basic Flow	User clicks on the map and green marker will be displayed for source address and red marker will be displayed for destination address. After that, source and destination lattitudes, longitudes will be saved in the database
Postcondition	Lattitudes, longitudes coordinates has been saved in the database.

Table 3.7: Choose location on map UseCase

Chapter 4

Design

This chapter includes all the design for the said model, having an in-depth analysis of the needs and requirements. The chapter itself is divided into different modules, all of which are discussed in detail, below.

4.1 System Architecture

System architecture represents the application as it is a high-level logical representation of the system. It always shows components and their relationships. System Architecture of this application is:

- Application Interface.
- Database
- Middleware

This describes the high level architecture of carpool application. The app offers a seamless registration process using email OTP, creating user profiles.

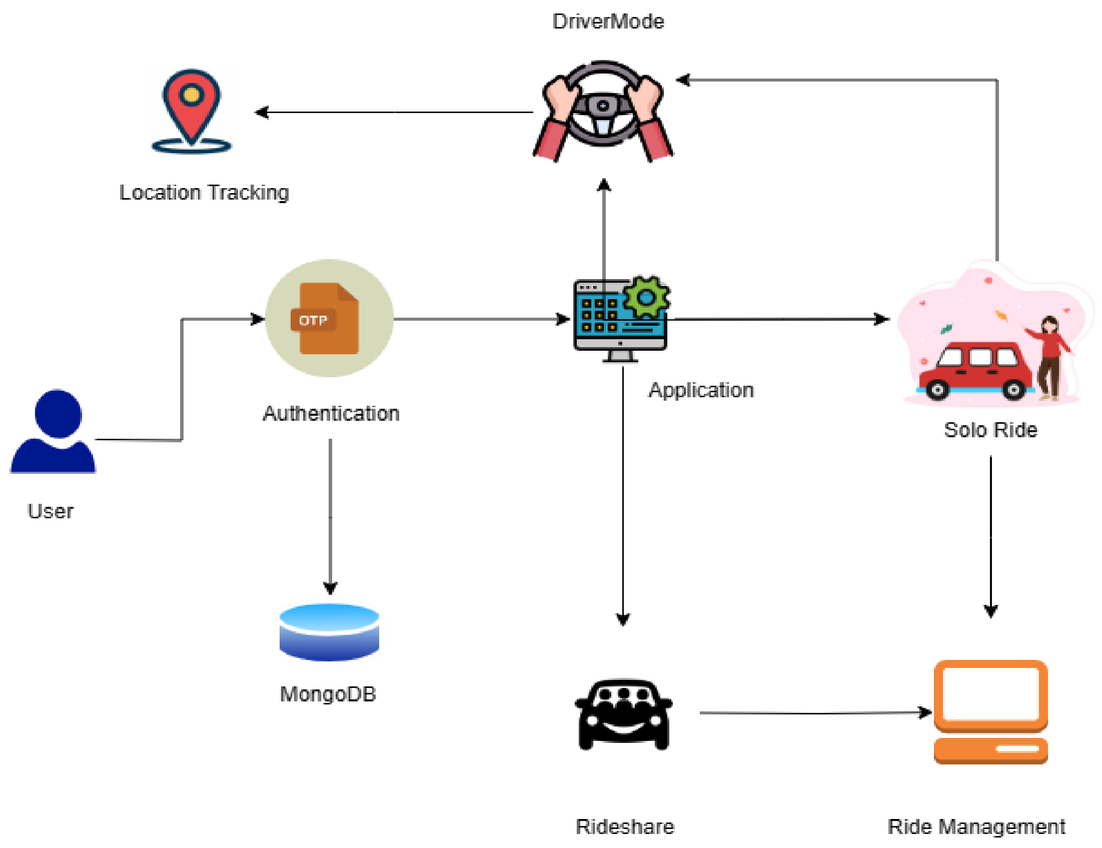


Figure 4.1: System Architecture

The main interface provides choices of ”Driver Mode” and ”Passenger Mode,” allowing passengers to request rides by specifying locations and choosing from available cars. Drivers in ”Driver Mode” can accept ride requests. Ride negotiation, sharing ride details, and flexible payment options enhance the user experience. Cancellation is allowed before the ride begins.

4.2 Use Case Diagram

4.2.1 Login Use case Diagram

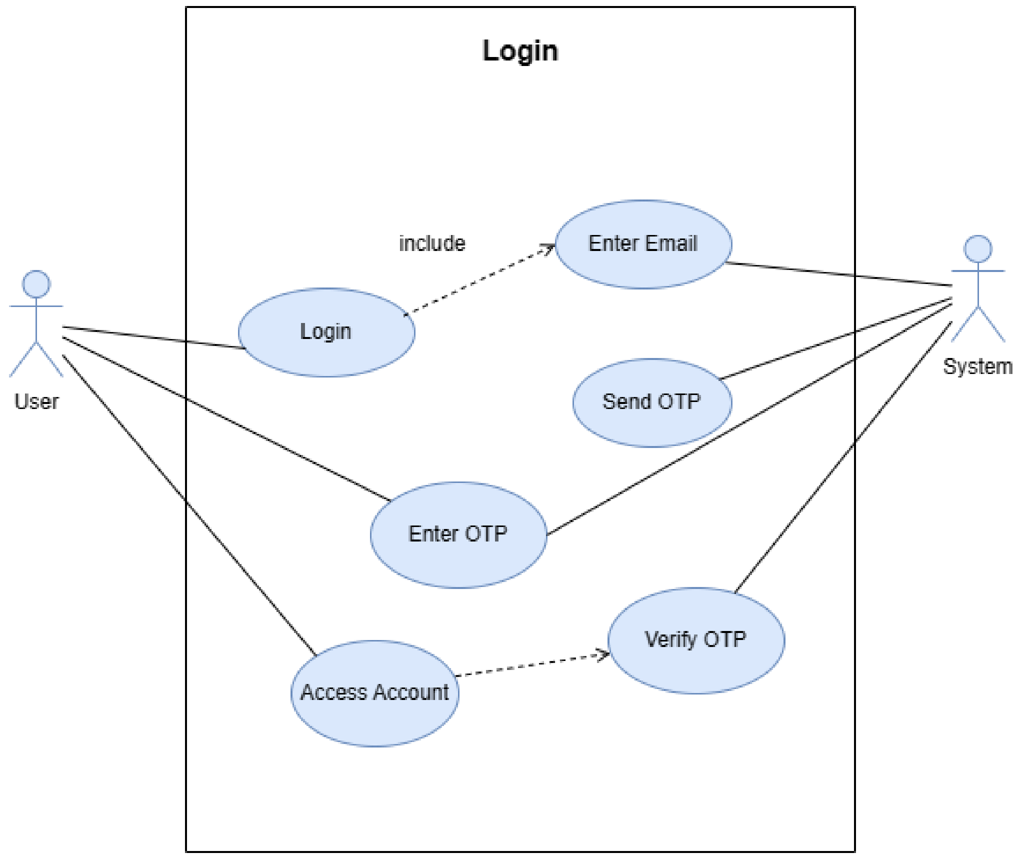


Figure 4.2: Login Use Case

4.2.2 Signup Use case Diagram

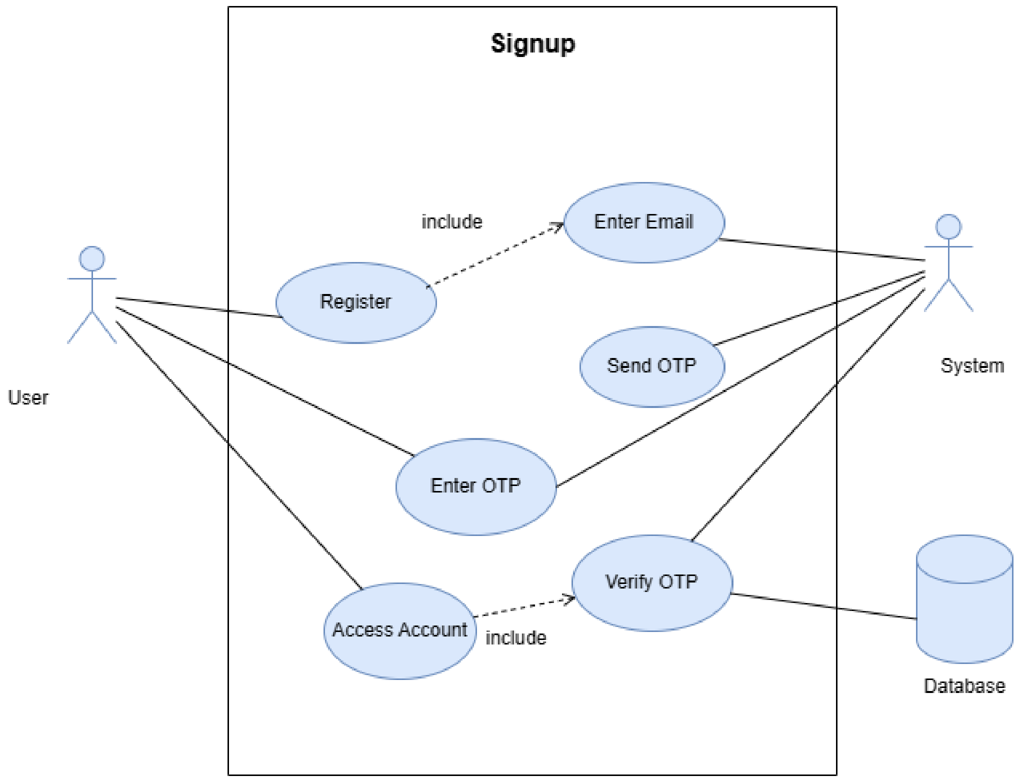


Figure 4.3: Signup Use Case

4.2.3 Profile Creation Use case Diagram

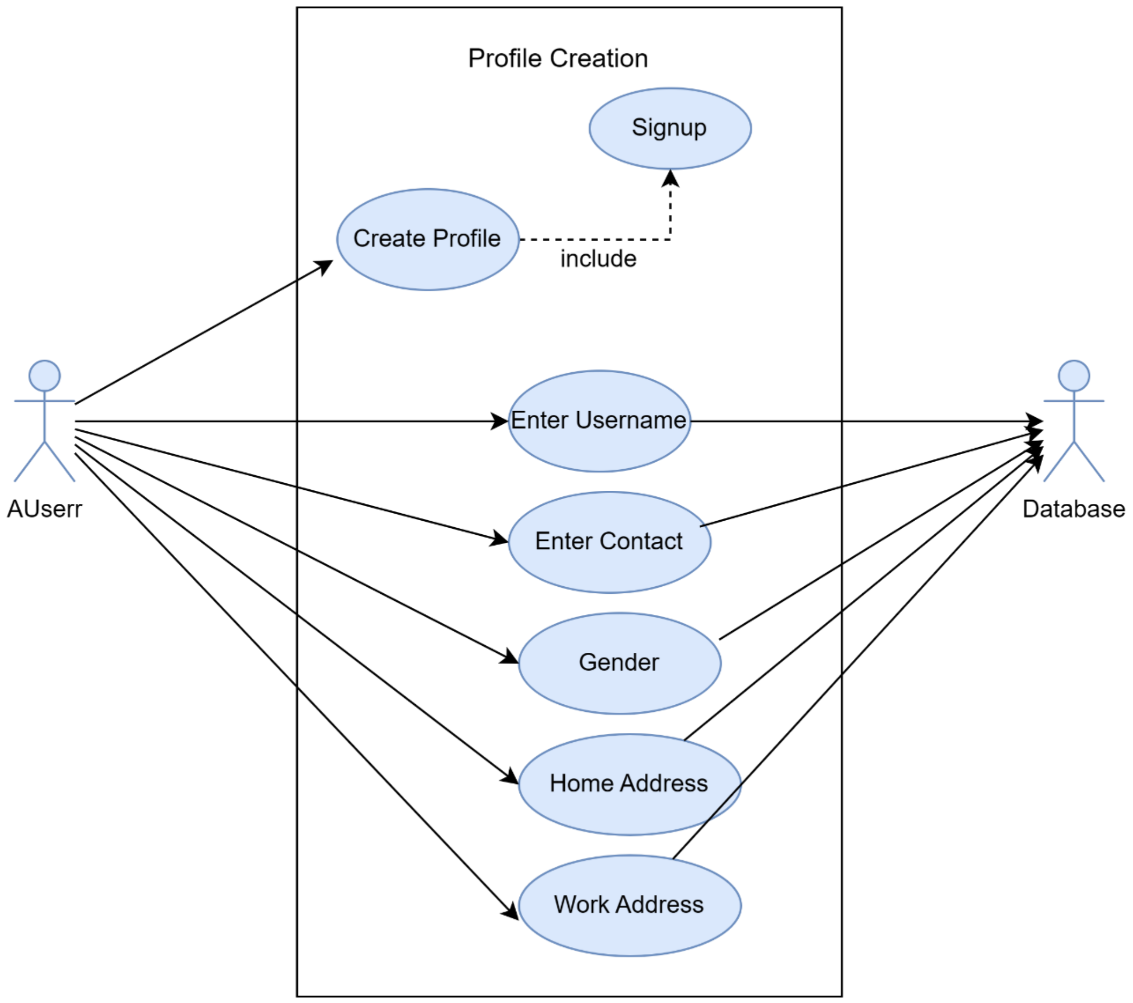


Figure 4.3: Profile Creation Use Case

4.2.4 .4 Main Use case Diagram

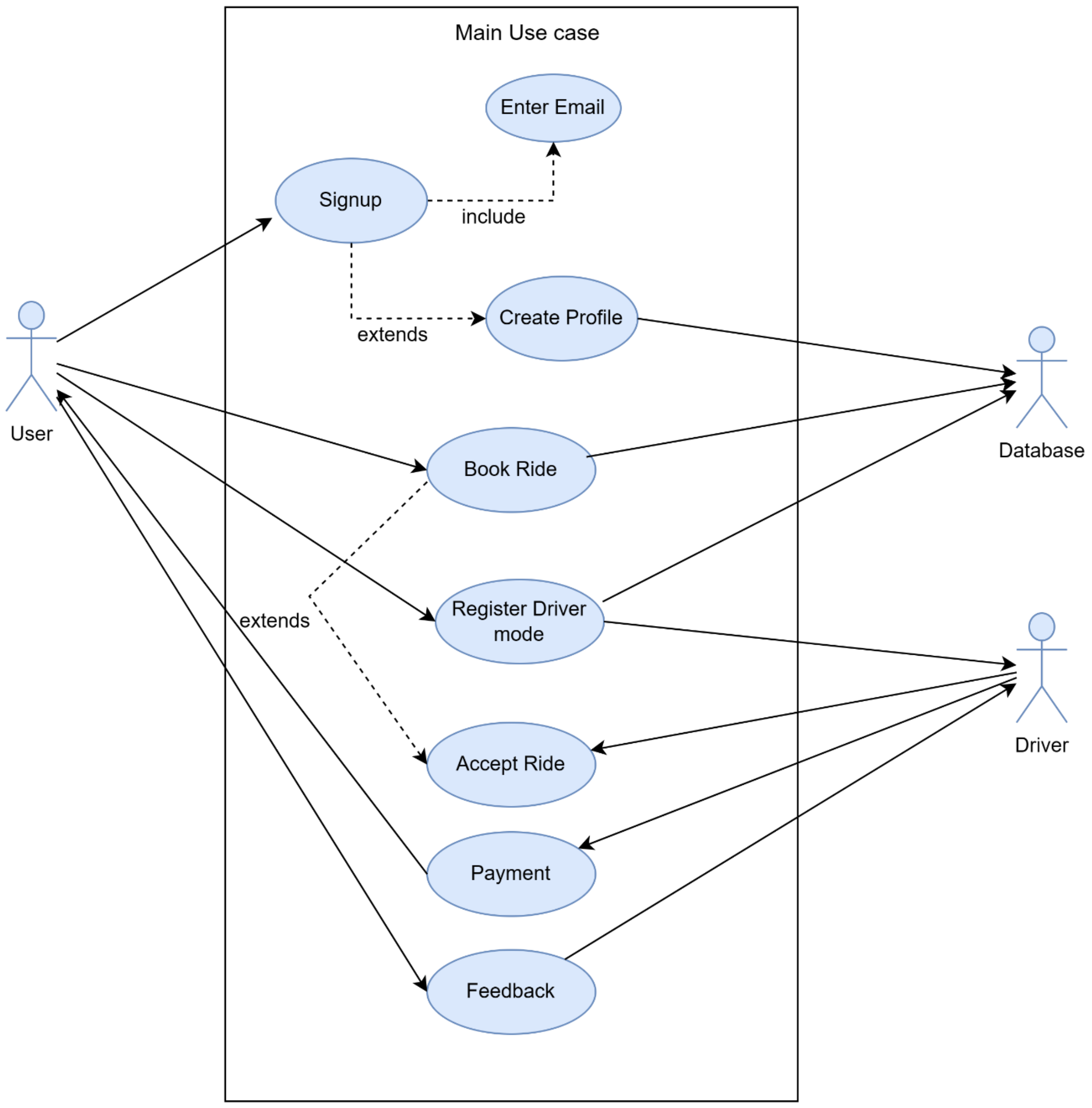


Figure 4.3: Main Use Case

4.3 Sequence Diagram

Sequence diagram describes the interaction between components. They show how operations are carried out in an application or system. It shows that what messages and send and when.

4.3.1 Main Sequence Diagram

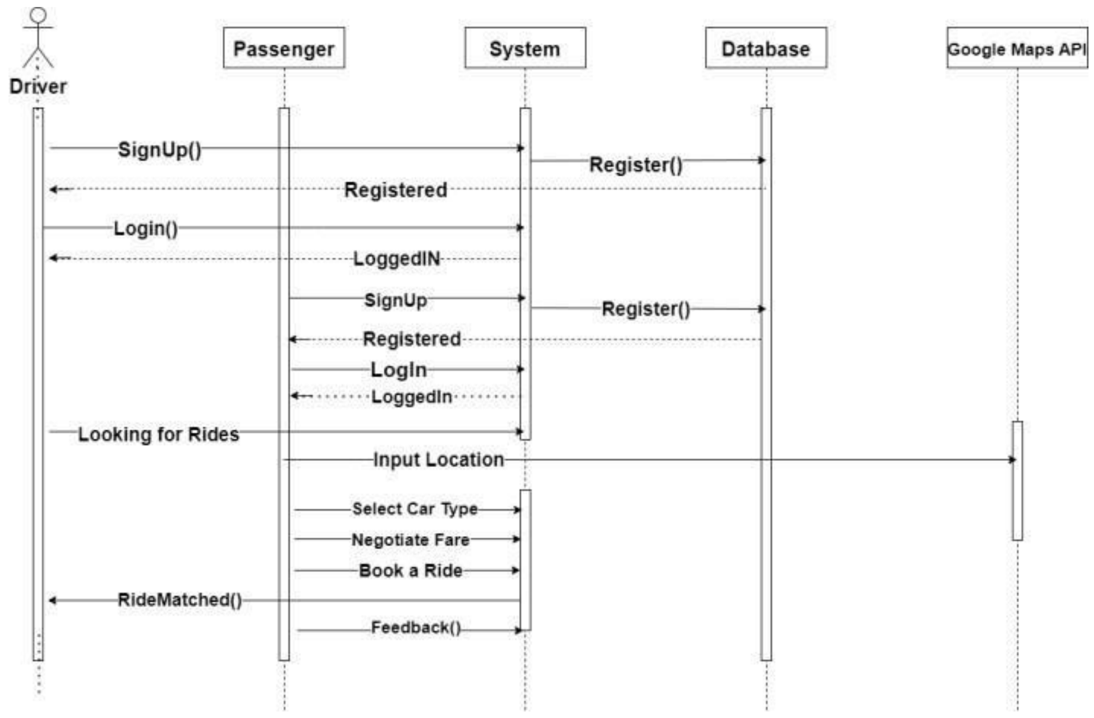


Figure 4.4: Main Sequence Diagram

4.3.2 Signup Sequence Diagram

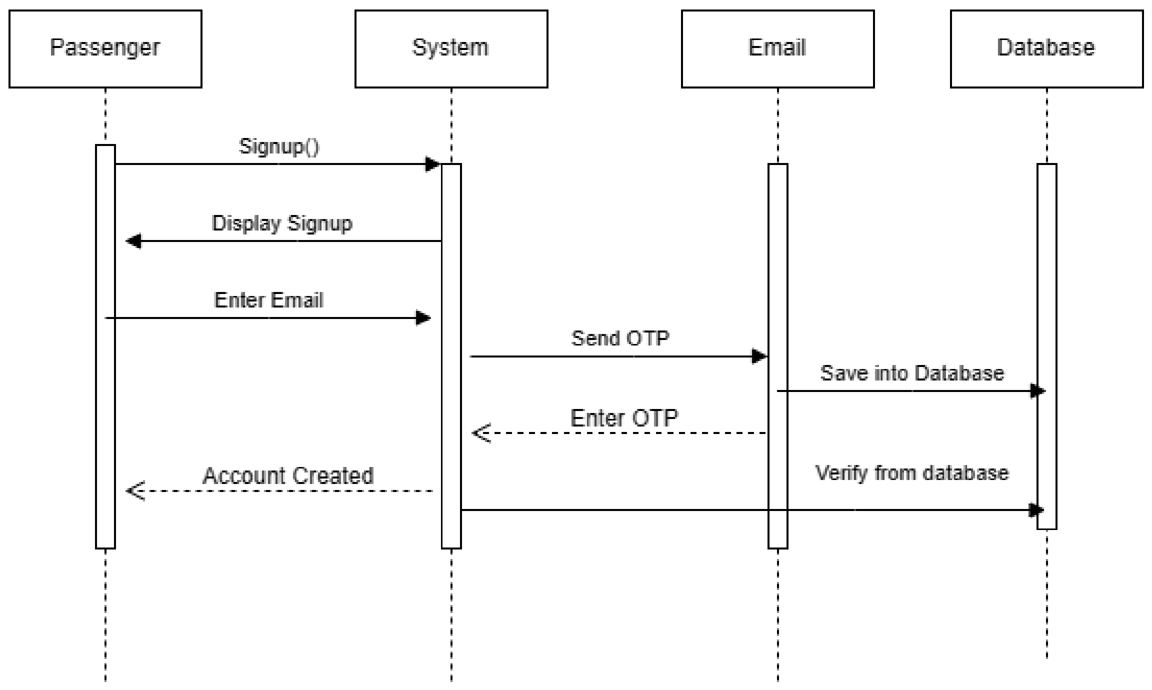


Figure 4.5: Signup Sequence Diagram

4.3.3 Login Sequence Diagram

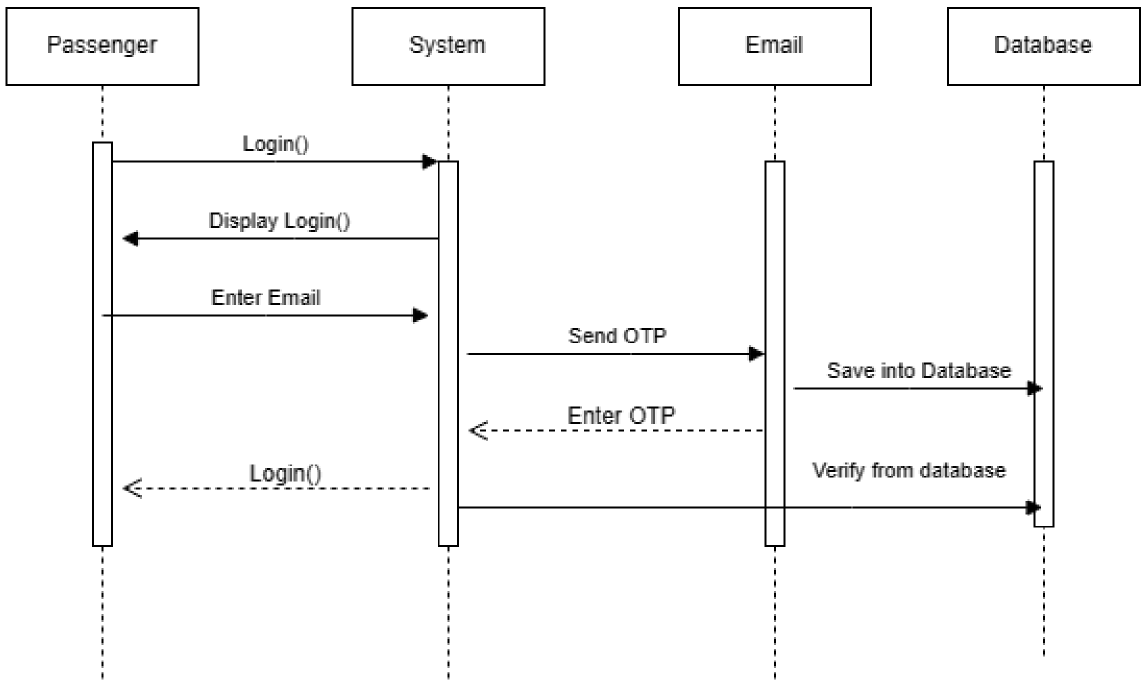


Figure 4.6: Login Sequence Diagram

4.3.4 Book Ride Sequence Diagram

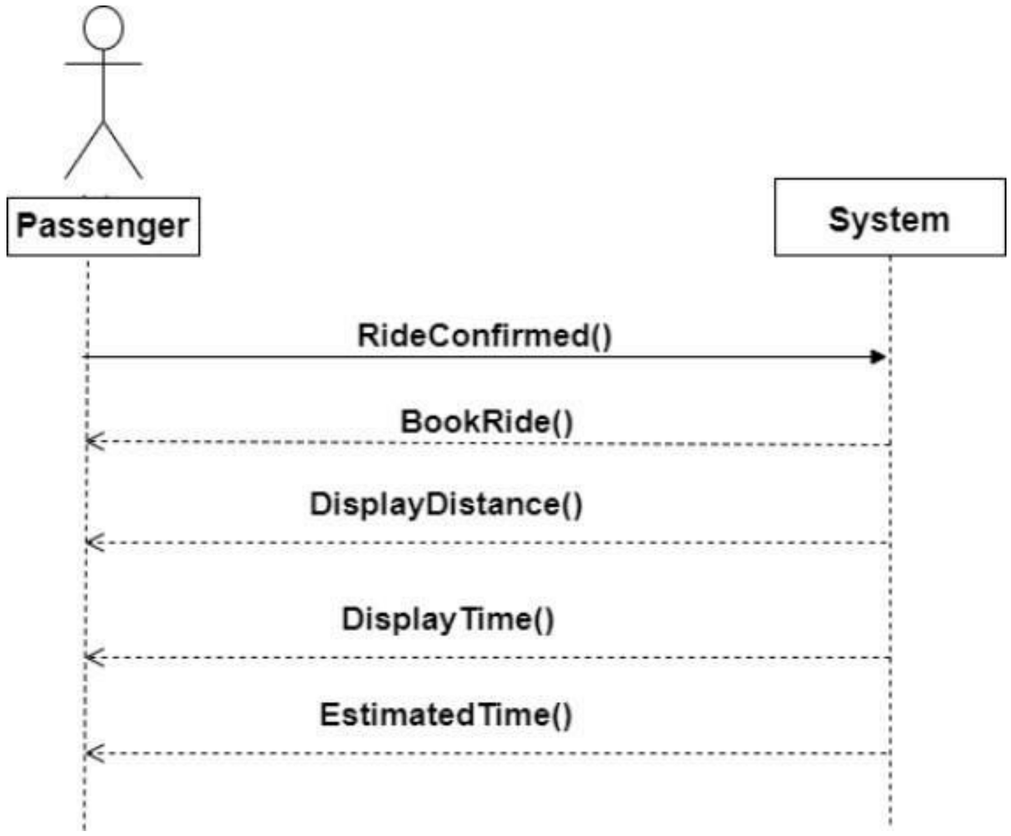


Figure 4.7: Book Ride Sequence Diagram

4.3.5 End of Ride and Feedback Sequence Diagram

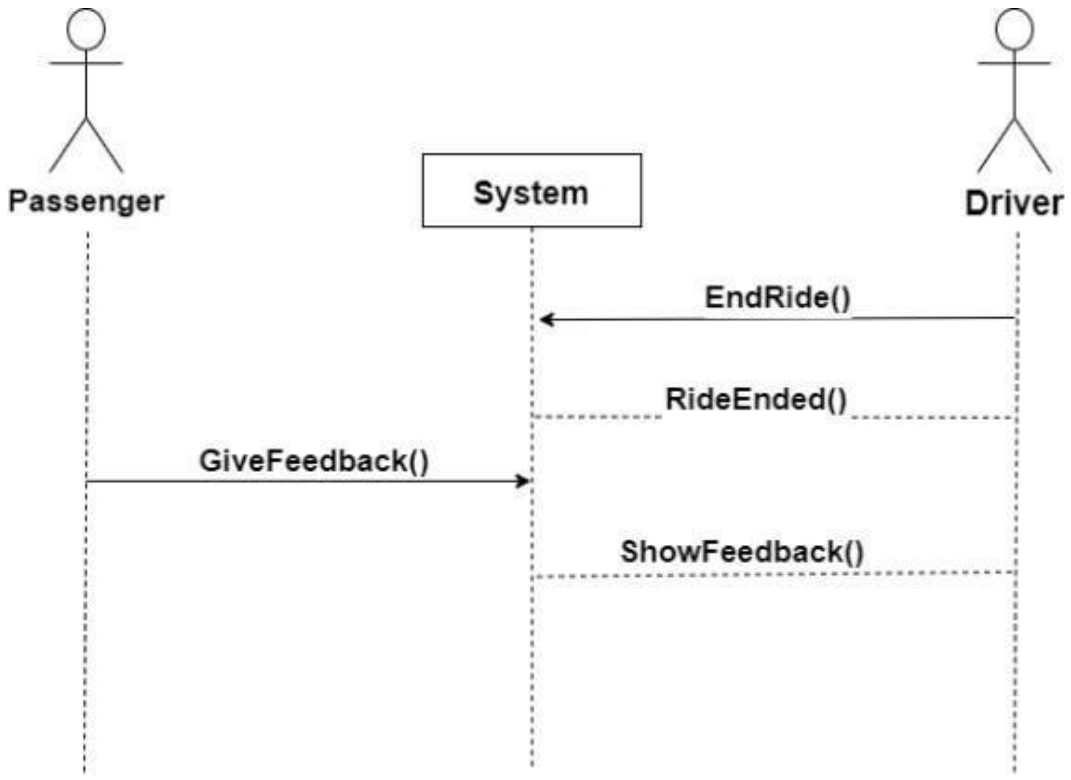


Figure 4.8: End of Ride and feedback Sequence Diagram

4.4 Activity Diagram

Activity Diagrams is a behavioral diagram used to illustrate the flow of control in the system and includes all the steps involved in the execution. Sequential and concurrent activities are modelled using activity diagrams. So, workflows are depicted visually using an activity diagram. [10]

4.4.1 Main Activity Diagram

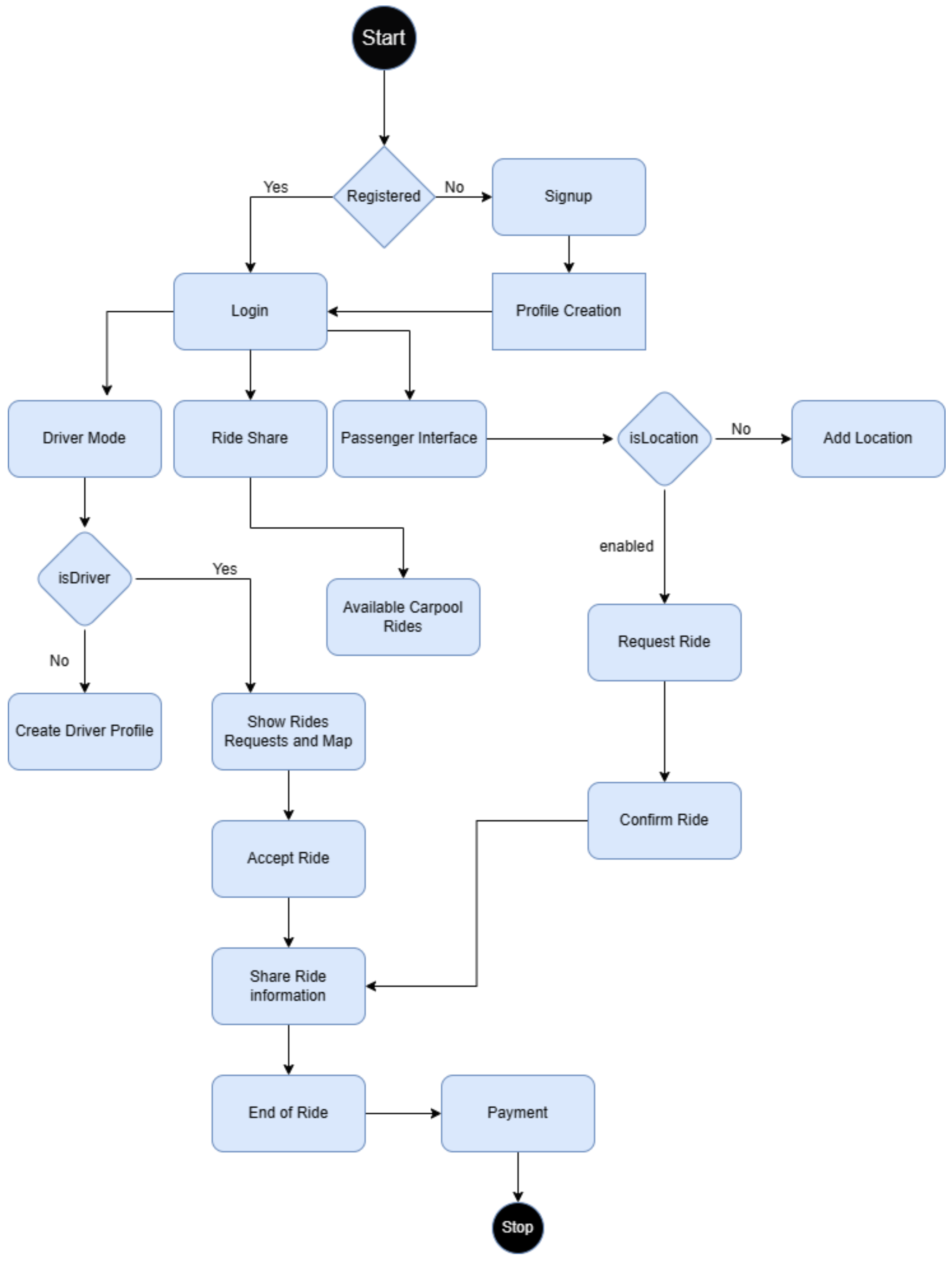


Figure 4.9: Main Activity Diagram

4.4.2 Login Activity Diagram

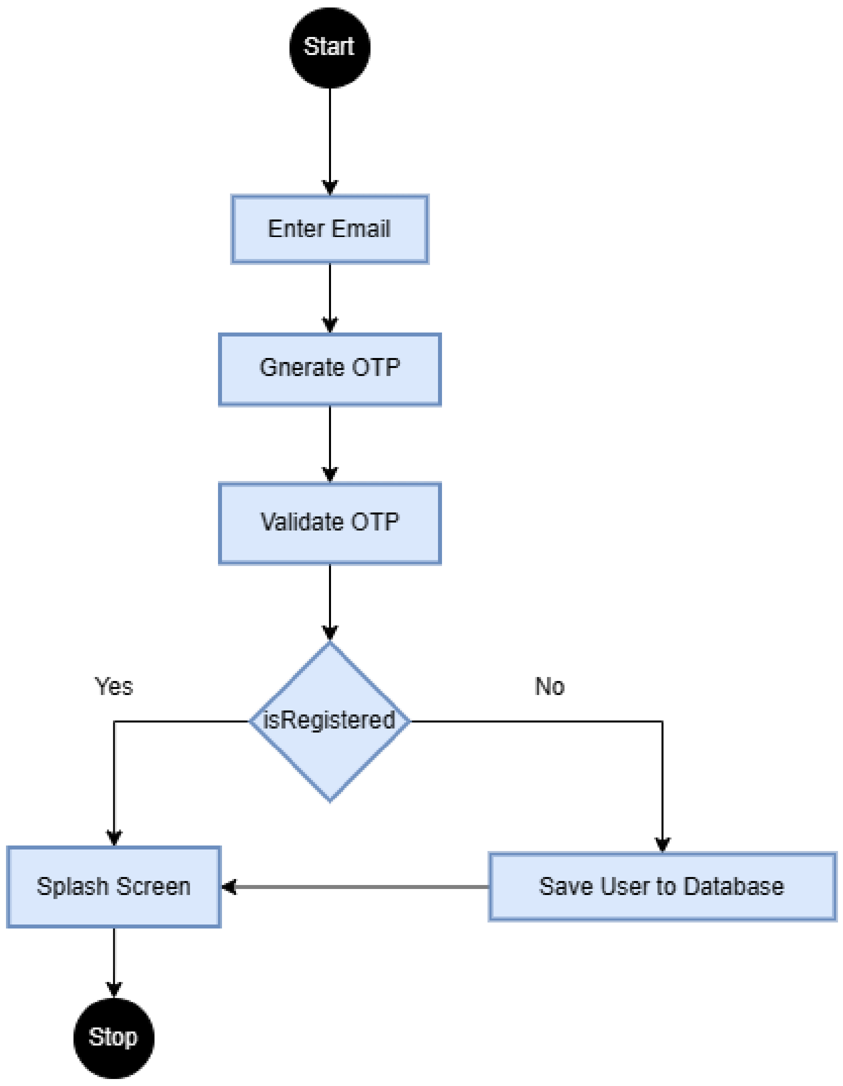


Figure 4.10: Login Activity Diagram

4.4.3 Request Ride Activity Diagram

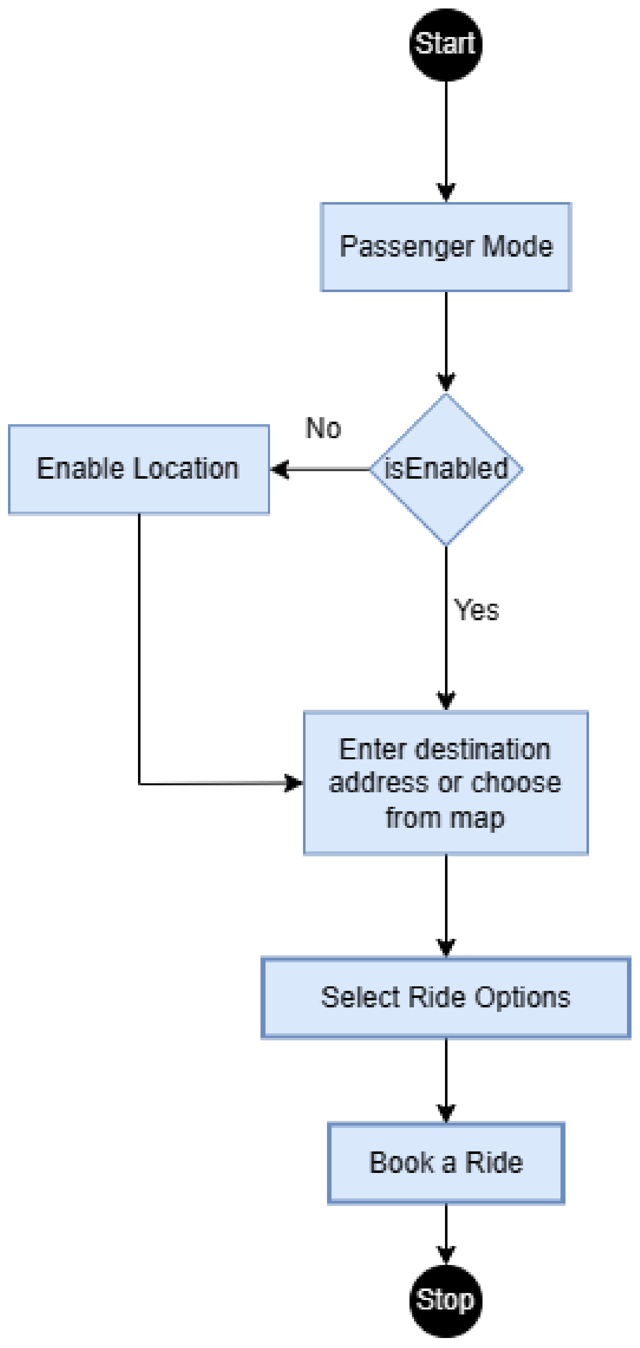


Figure 4.11: Request Ride Activity Diagram

4.4.4 Carpool Ride Activity Diagram

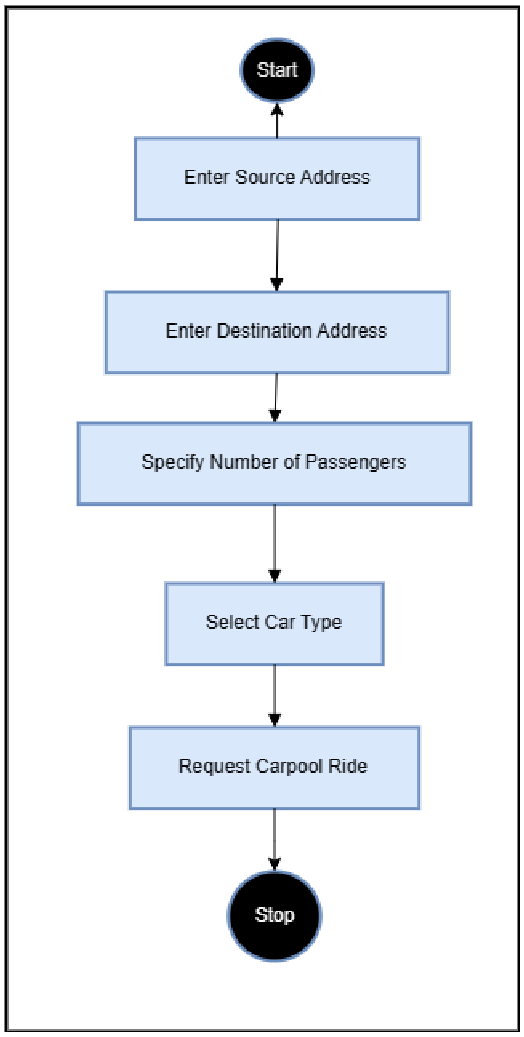


Figure 4.12: Carpool Ride Activity Diagram

Chapter 5
System Implementation

This chapter gives the detailed overview of how system is implemented, which technologies, database design and system architecture are used to develop a robust system. It is the process of defining and explaining how the system should be build and ensuring that system is performing the desired tasks efficiently. [11]

5.1 System APIs

Google Places Autocomplete API	This API is used to display user suggestions of Google map locations after they enter min3 keywords in the input type.
Reverse Geocoding API	**Reverse Geocoding:** This API is used to convert lattitude and longitude coordinates into human readable addresses. **Forward Geocoding:** This API is used to convert human readable addresses into lattitude and longitude coordinates.

Table 5.1: System APIs

5.2 Tools Used

5.2.1 Visual Code

Visual Code IDE is used to code the entire project repository inlcuding the server side code and front end. VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. [12]

5.2.2 Android Emulator

The android studio emulator is used to test the application on mulitple android versions and screens. It provided almost all the capabilities of a real Android device. [13]

5.3 Technologies Used

5.3.1 React Native

React Native is an open-source mobile application development framework that allows developers to create high-performance mobile apps for multiple platforms, including iOS and Android, using a single codebase. [14]

5.3.2 Node js

Node js is used to connect frontend with database server using post and get requests. Database is connected with nodejs server and then data has been saved in the router using node js commands. This data is displayed at the frontend using fetch APIs from node js.

5.3.3 Mongo db

MongoDB is a popular NoSQL database management system known for its flexibility and scalability. It is used to store users profiles and their rides informations.

5.3.4 Blockchain Ethereum Wallet

A Blockchain Ethereum wallet is used to securely manage payment operations among passengers and drivers. It ensures the safe storage of digital assets and facilitates transactions within the Ethereum network. Users can store, send, and receive Ethereum, access DApps, and interact with smart contracts using their Ethereum wallet.

5.4 System GUI

5.4.1 Splash Screen

The user will be welcomed by a splash screen after which the user will have to provide his/her credentials and gain access to our application’s features.

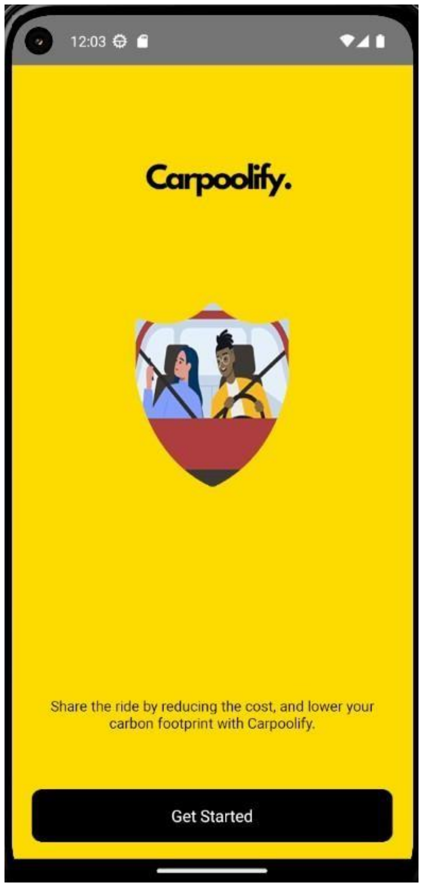

12:03
Carpoolify.
Share the ride by reducing the cost, and lower your carbon footprint with Carpoolify.
Get Started

Figure 5.1: Splash Screen

5.4.2 Login Screen

The user will be able to Login after which the Welcome screen. User will have to provide his/her credentials and gain access to the application's features.

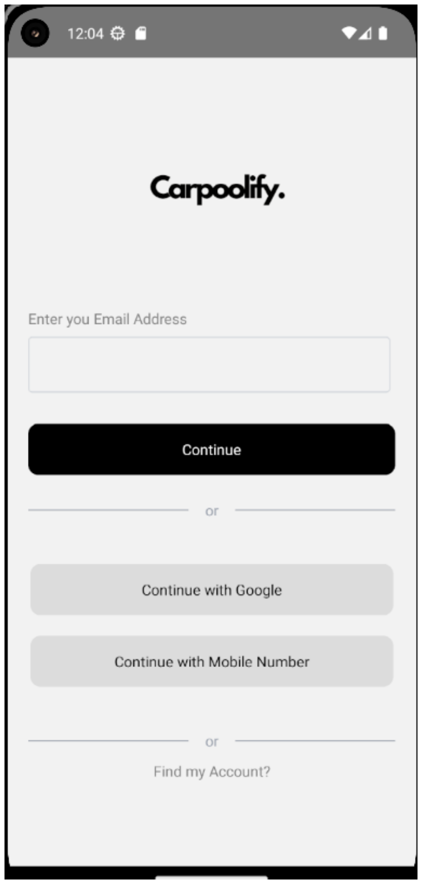


Figure 5.2: Login Screen

5.4.3 Location Permission

The user can give access to the application for current location that can be used as source address.

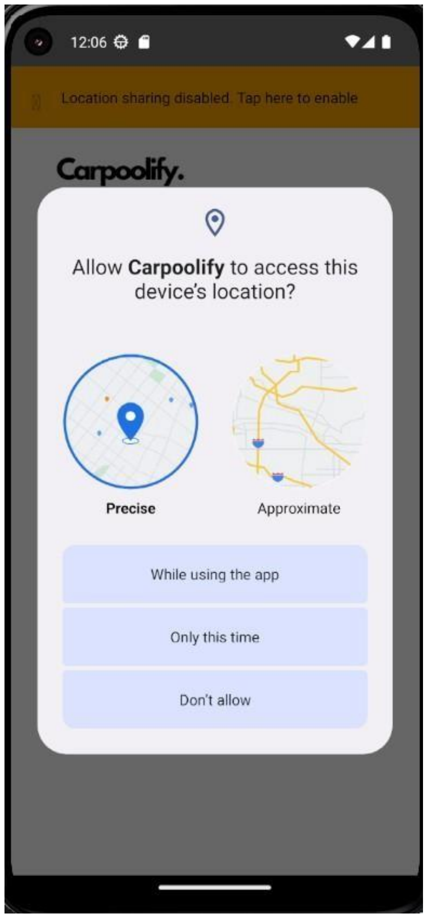
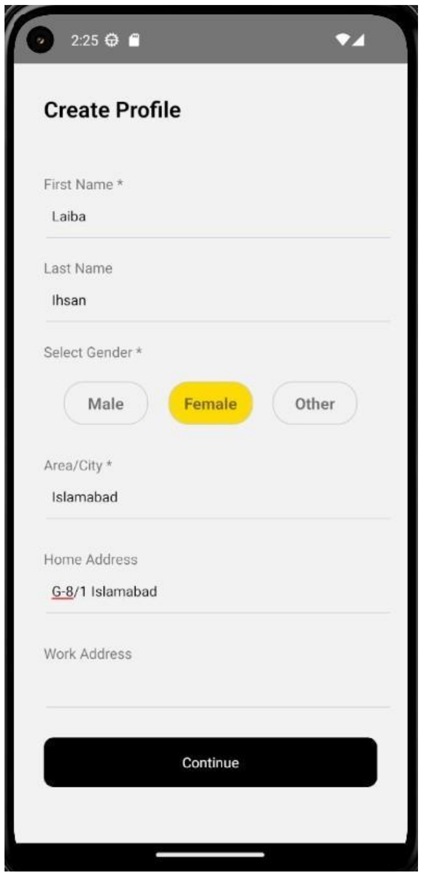

12:06
Location sharing disabled. Tap here to enable
Carpoolify.
Allow **Carpoolify** to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow

Figure 5.3: Enable Location Screen

5.4.4 Create User Profile

The user can create their profile with basic information, firstname, lastname, gender, area and addresses that will be saved in database.



First Name *
Laiba

Last Name
Ihsan

Select Gender *
Female

Area/City *
Islamabad

Home Address
G-8/1 Islamabad

Work Address

Figure 5.4: Create User Profile Screen

5.4.5 Main Screen

The user will be displayed with the main screen after successful login to the app. Here user can access the app’s services.

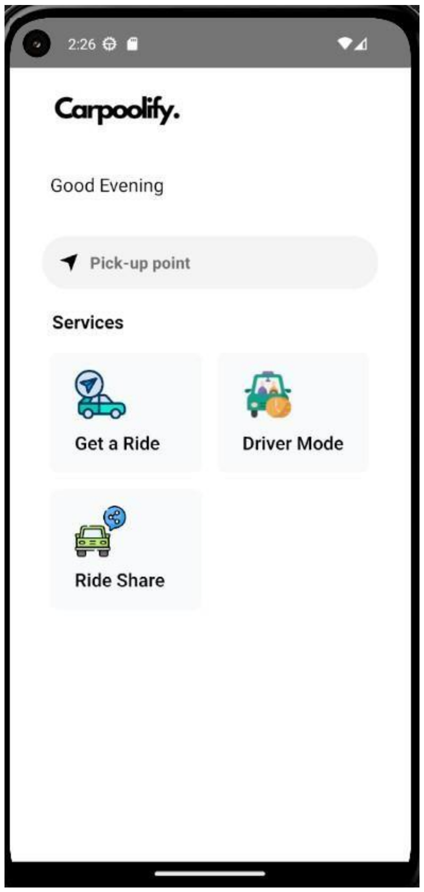


Figure 5.5: Main Screen

5.4.6 Choose Location Screen

The user can manually enter the location in input fields.Here Google Places Autocomplete Api will be called after entering 3 keywords, user will get suggestions of relevant locations.

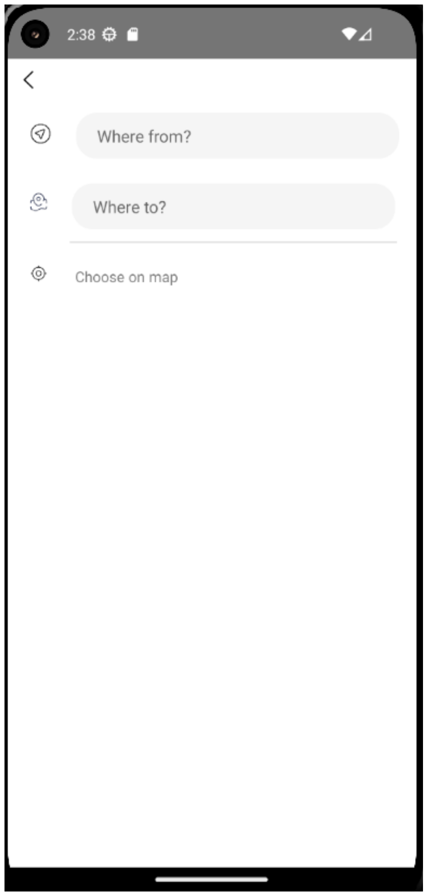


Figure 5.6: Choose Location Screen

5.4.7 Choose Location from Map Screen

The user can also select location from map by touching the desired points. The selected source and destination latitude and lonitudes will be converted to addresses and saved in databases.

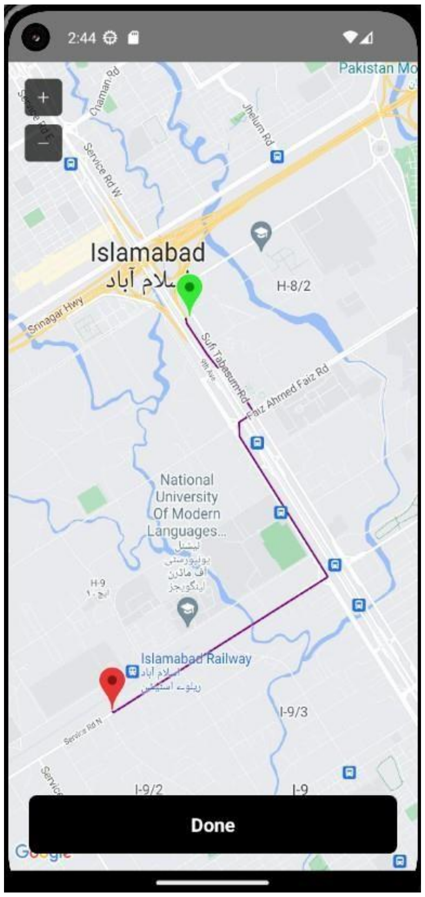


Figure 5.7: Map Location Screen

5.4.8 Book Ride Screen

The user can entered location addresses will be displayed in input fields. User can select the type of car to travel.

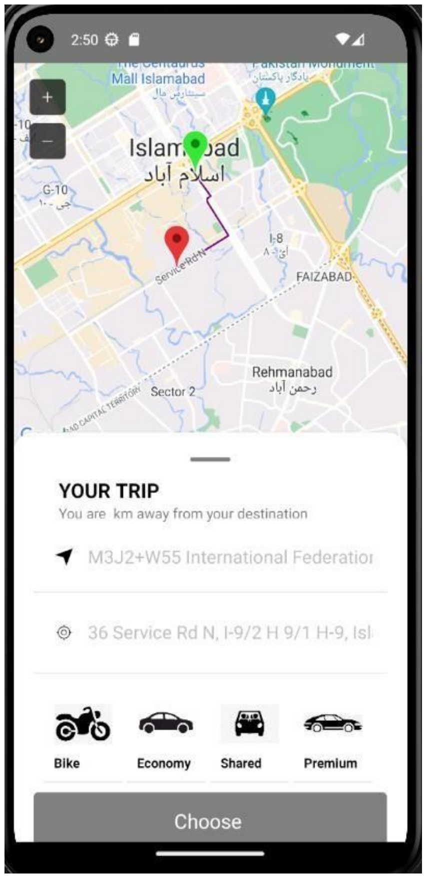

2:50
Islamabad
Sector 2
Rehmanabad
FAIZABAD
YOUR TRIP
You are km away from your destination
M3J2+W55 International Federatior
36 Service Rd N, I-9/2 H 9/1 H-9, Isl
Bike
Economy
Shared
Premium
Choose

Figure 5.8: Book Ride Screen

5.4.9 Ride share Screen

The user can manually request for the shared ride at the separate component.

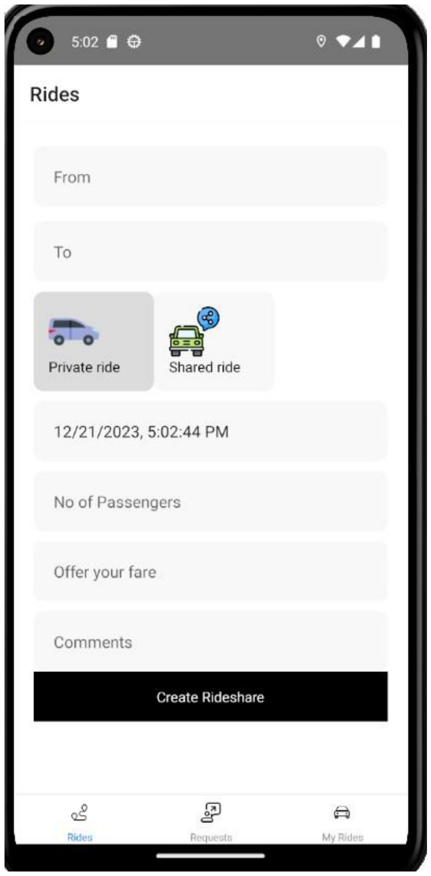


Figure 5.9: Rideshare Screen

5.4.10 Driver Screen

If the current user is not registered as a driver, then the user has to create their driver profile first.

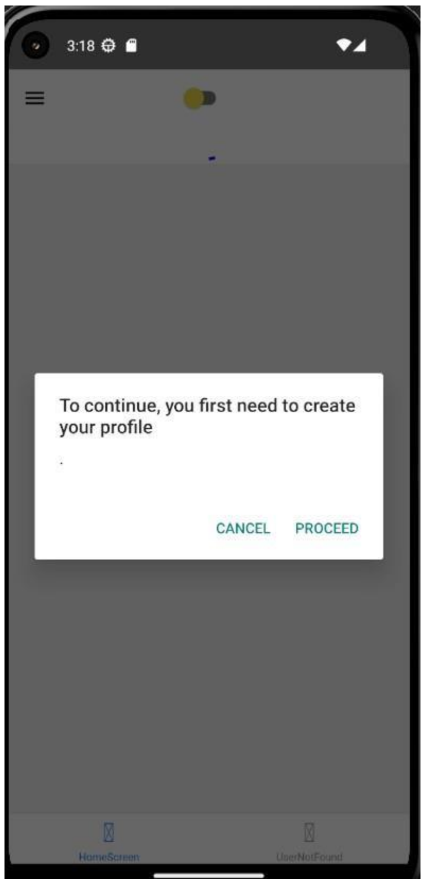


Figure 5.10: Driver Screen

5.4.11 Driver Registration Screen

Driver has its seperate registration with essential fields. Driver will generate its own token, based on which driver profile will be rendered.

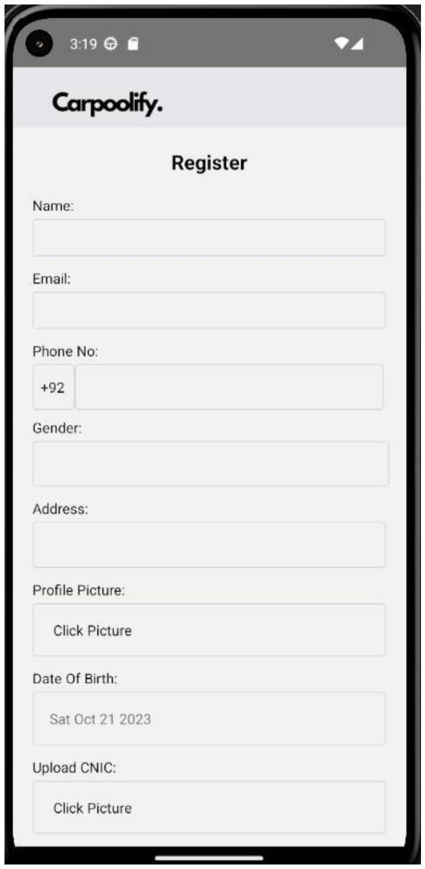


Figure 5.11: Driver Registration Screen

5.4.12 Driver’s Map Screen

After successful profile creation driver will be redirected towards the Map screen, where a red marker will be displayed sowing the driver’s current location and this marker changes with the driver’s location.

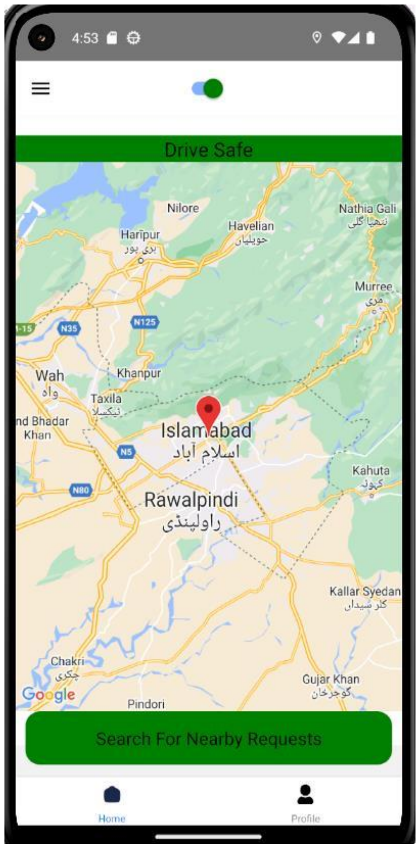


Figure 5.12: Driver Map Screen

Chapter 6
System Testing and Evaluation

6.1 Approaches for Testing

In this section approaches to testing have been specified through which the system has been tested and evaluated. The test cases covered in this section will be:

1. GUI Testing
2. Usability Testing
3. Performance Testing
4. Security Testing
5. Documentation Testing
6. Integration Testing

6.2 GUI Testing

6.2.1 Signup Test case

Test Case ID	Signup-EmailOTP
Tester Name	Laiba Ihsan
Test Case Description	To verify that users can successfully sign up using their email and OTP.
Preconditions	• The app is installed on an Android device.
Test Steps	1. Tap on the ”continue” button on welcome screen. 2. Enter a valid email address and request an OTP. 3. Check your email for the OTP. 4. Enter the received OTP in the app. 5. Tap the ”Sign Up” button.
Expected Results	• The app successfully registers the user’s account, and the user is logged in. • The user can access the app’s services.
Actual Results	User successfully created an account.
Status	Pass

Table 6.1: Test Case Signup with Email and OTP

6.2.2 Existing Email Signup Test case

Test Case ID	Signup-ExistingEmail
Tester Name	Laiba Ihsan
Test Case Description	To verify that users receive an appropriate error message when attempting to sign up with an email address that already exists in the system.
Preconditions	• An account with the email address used in this test case already exists in the system.
Test Steps	1. Enter an email address that is already associated with an existing account. 2. Request an OTP for registration. 3. Check your email for the OTP. 4. Enter the received OTP in the app. 5. Tap the ”Sign Up” button.
Expected Results	• The app should detect that the email address is associated with an existing account. • An error message should be displayed, such as ”Account already exists. Please log in.”
Actual Results	Error ” Account already exists. Please Log in”
Status	Pass

Table 6.2: Test Case Signup with Existing Email

6.2.3 Login Email Test Case

Test Case ID	Login-EmailOTP
Tester Name	Laiba Ihsan
Test Case Description	To verify that users can successfully log in using their email and OTP.
Preconditions	• A registered account with email exists.
Test Steps	1. Enter a valid email address. 2. Request an OTP for login. 3. Check your email for the OTP. 4. Enter the received OTP in the app. 5. Tap the ”Log In” button.
Expected Results	• The app successfully logs the user in. • The user can access the app’s services.
Actual Results	User is successfully in the main screen.
Status	Pass

Table 6.3: Test Case Login with Email and OTP

6.2.4 Location Permission Test case

Test Case ID	Location Permission
Tester Name	Laiba Ihsan
Test Case Description	To verify that the app requests location permission and correctly fetches the latitude and longitude of the user's current location and places a marker on the map.
Preconditions	• Stable internet connection is available. • The device's location services (GPS) are enabled.
Test Steps	1. The app prompts the user to grant location permission. 2. Allow location permission. 3. The app accesses the device's GPS and fetches the current latitude and longitude. 4. The app displays a map with a marker at the fetched coordinates.
Expected Results	• The app should request location permission from the user. • After permission is granted, the app should access the device's GPS and successfully fetch the current latitude and longitude. • A marker should be placed on the map at the fetched coordinates.
Actual Results	Successfully fetches the current location and marker is displayed on the map.
Status	Pass

Table 6.4: Test Case Location Permission and Marker Placement

6.2.5 Reverse Geocoding Test case

Test Case ID	Reverse Geocoding
Tester Name	Laiba Ihsan
Test Case Description	To verify that the app can correctly convert latitude and longitude coordinates into human-readable addresses using a reverse geocoding API.
Preconditions	• Stable internet connection is available. • The geocoding API key is included in .env file.
Test Steps	1. Launch the app. 2. Add source and destination addresses. 3. Initiate a async request to the geocoding API to convert the coordinates into an address. 4. Save the addresses into database and display them.
Expected Results	• The API should respond with a human-readable address that corresponds to the provided latitude and longitude coordinates.
Actual Results	Successfully converted the latitudes and longitudes into addresses and saved into database.
Status	Pass

Table 6.5: Test Case Convert Lat/Lon to Address

6.2.6 Request Ride Test case

Test Case ID	Ride Request
Tester Name	Laiba Ihsan
Test Case Description	To verify that a user can successfully request a ride.
Preconditions	• The user is logged into their account.
Test Steps	1. Ensure user is logged into their account. 2. Tap on the ”Get a Ride” button. 3. Enter the source and destination addresses or choose from map. 4. Choose the car type for your ride. 5. Confirm the ride request. 6. Wait for the app to find a driver.
Expected Results	• The app should allow user to enter the source and destination addresses or choose from map. • Passenger should be able to choose the car type for ride. • After confirming the ride request, the app should find a driver for the solo ride. • The passenger should be able to select the driver from the list.
Actual Results	Ride request has been made successfully and driver is arriving within some time depend upon the distance.
Status	Pass

Table 6.6: Test Case Solo Ride Request

6.2.7 Rideshare Test case

Test Case ID	Rideshare Request
Tester Name	Laiba Ihsan
Test Case Description	To verify that a user can successfully request a rideshare ride.
Preconditions	• The user is logged into their account.
Test Steps	1. Enter the source and destination addresses. 2. Specify the number of passengers and car type 3. Wait for the app to find rideshare drivers. 4. Verify that rideshare drivers are assigned and displayed.
Expected Results	• The app should allow user to enter the source and destination addresses. • User should be able to specify the number of passengers. • User should be able to choose the car type for your rideshare ride. • After confirming the rideshare ride re-quest, the app should find rideshare drivers.
Actual Results	Passenger has successfully requested for the shared ride.
Status	Pass

Table 6.7: Test Case Rideshare Request

6.2.8 Driver Mode Activation

Test Case ID	DriverMode-Activation Test case
Tester Name	Laiba Ihsan
Test Case Description	To verify that users can successfully activate the driver mode and create a driver profile.
Preconditions	• The app is installed on an Android device. • The user is already registered in the system.
Test Steps	1. Launch the app. 2. Access the ”Driver Mode”. 3. Initiate the driver mode activation. 4. Fill in the driver profile details (e.g., name, car details). 5. Verify and confirm the driver mode activation. 6. Check for a successful activation message.
Expected Results	• The user should be able to activate the driver mode and create a driverprofile successfully. • A confirmation message should be displayed.
Actual Results	Driver mode activated, and a confirmation message displayed.
Status	Pass

Table 6.8: Test Case Driver Mode Activation

6.3 Usability Testing

It is the process of testing how application will be used by its target users. Usability Testing identifies that the system follows certain rules and norms to help identify design and user-related issues in the System. [15]

Task	**Description**	**Usability Rating (1-5)**
Task 1	Navigating to Main Screen - Login to the application and access main screen - Evaluate how easily you can return to the Login screen from the main.	5
Task 2	Booking a Ride - Navigate to the Map screen from the main screen. - Assess the process of returning to the main screen after completing a ride booking.	4
Task 3	Accessing Driver Mode - Find and access the Driver Mode feature, - Assess the ease of returning to the passenger mode from the driver mode.	4

Table 6.8: Navigating between screens Usability testing

6.4 Performance Testing

6.4.1 Reverse Geocoding Response Time

Test Case ID	Performance-ReverseGeocoding
Tester Name	Laiba Ihsan
Test Case Description	To evaluate the performance of the reverse geocoding feature by measuring the response time it takes to convert latitude and longitude coordinates into an address.
Preconditions	The app has the necessary permissions to access location information.
Test Steps	1. Manually input a set of latitude and longitude coordinates or choose from map to obtain current coordinates to book ride. 2. Trigger the reverse geocoding process by click on book ride option to convert the coordinates into an address. 3. Measure and record the time it takes for the app to display the human-readable address. 4. Repeat the test for multiple coordinates or addresses. 5. Calculate the average response time.
Expected Results	- The reverse geocoding feature should convert latitude and longitude coordinates into addresses within a few seconds.
Actual Results	The lat and longs are converted to their corresponding addresses within seconds.
Status	Pass

Table 6.10: Reverse Geocoding Performance Test case

6.5 Security Testing

6.5.1 Token Expiry on Logout

83**Test Case ID**	Security-LogoutTokenExpir6y.4. Performance Testing
Tester Name	Laiba Ihsan
Test Case De-scription	To verify that the token expires upon user logout, ensuring the user cannot remain logged in.
Preconditions	1. The user has successfully logged in and obtained a valid token.
Test Steps	1. Log in to the app using valid credentials to obtain a token. 2. Verify that the token allowsaccess to the app’s services. 3. Log out of the app. 4. Attempt to access app resources using the same token. 5. Verify that the token is no longer valid
Expected Results	The token is no longer valid after logout.
Actual Results	The system correctly recognizes that the token is no longer valid after logout, denying access to app resources.
Status	Pass

6.6 Documentation Testing

In this phase of testing, I have checked that the proposed document is according to the requirements of the application. This ensured that how touse the system matches with what system does by adding all the necessary information from preparatory phase to execution phase.

6.7 Integration Testing

This system has two components i.e., Driver and the Passenger where Driver is responsible for providing services to the Passenger and Passenger can choose from ride- sharing or solo-rides. Therefore, a successful integration test was initiated to test the functionality on both sides. I booked a ride from the Passenger side end and checked that it is being shown to the Driver and can successfully accept the Ride.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

In conclusion, Carpoolify is an innovative mobility application that has the potential to reshape the way people commute in urban areas. The test cases we've developed demonstrate its robustness and reliability, ensuring that users can access a safe and efficient transportation service.

Carpoolify sets itself apart through its pioneering use of blockchain technology. This not only enhances security but also provides transparency in transactions, offering users peace of mind in a digital world where data privacy is of utmost importance.

Furthermore, this application addresses the issue of rising transportation costs. By eliminating the need for third-party fees, both drivers and passengers can enjoy significant cost savings. Carpoolify is not just a convenient mode of transportation; it's a financial relief for all its users.

Another noteworthy feature of Carpoolify is its commitment to the environment. By promoting carpooling, it actively contributes to reducing the number of vehicles on the road, thereby decreasing carbon emissions. This eco-friendly approach not only benefits the planet but also lightens the loadon users' wallets.

In this era of evolving transportation services, Carpoolify stands out as a comprehensive solution that addresses multiple concerns—security, affordability, and sustainability. It's poised to become a go-to choice for those who seek efficient, secure, and budget-friendly mobility options. Carpoolify isn't just a ride; it's a journey towards a smarter, more sustainable urban future.

7.2 Future Enhancements

In terms of future enhancements, there are several key areas to consider for the app's growth and improvement. Firstly, integrating public transportation options into the app can provide users with a seamless and comprehensive transportation experience. This feature would enable users to combine carpooling with public transit, offering even more flexibility and convenience in their daily commutes.

Secondly, leveraging user data is vital for gaining insights into travel patterns, user behavior, and demand. By analyzing this data, the app can op- timize route suggestions and carpool matching, ultimately providing a more efficient and tailored service to users.

To expand the app's reach, it's important to consider offering multi- language support. This enhancement allows the app to cater to a more diverse user base, ensuring that people from different regions and language backgrounds can comfortably use it.

Lastly, accessibility features are crucial to ensure that the app is user-friendly and inclusive for everyone. By incorporating features like voice commands and screen reader compatibility, the app can be used by individuals with disabilities, making it accessible to a broader audience and improving the overall user experience.

Appendix A

User Manual

This manual is your go-to guide for using the app effortlessly. It explains everything step by step, making it easy for you to understand and use all the features. It’s like having a helpful friend to show user around the app, so user can get the best experience.

A.0.1 Splash Screen

The Splash Screen is the first screen users see when they open the Carpoolify app. It provides a warm welcome and sets the tone for their journey. The Splash Screen appears for a brief moment, introducing users to the app’s world of hassle-free ridesharing.

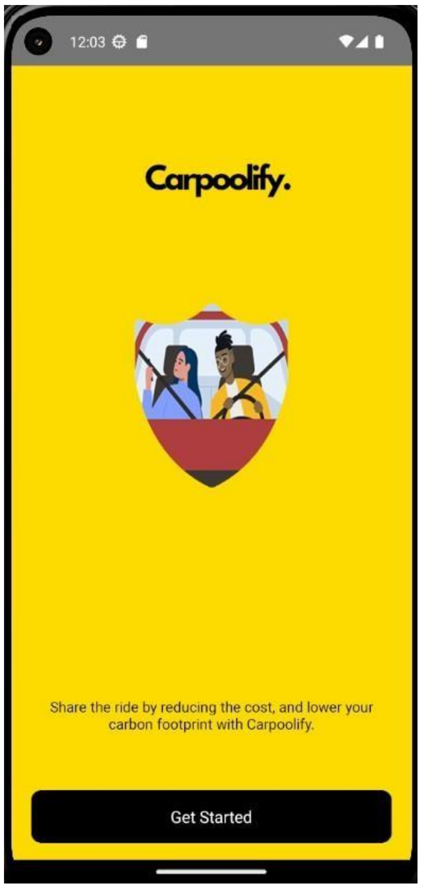

12:03
Carpoolify.
Share the ride by reducing the cost, and lower your carbon footprint with Carpoolify.
Get Started

Figure A.1: Splash Screen

A.0.2 Login Screen

The Login Screen is where users start their journey into the Carpoolify app. To access the app's features, users must enter their login credentials, ensuring a secure and personalized experience.

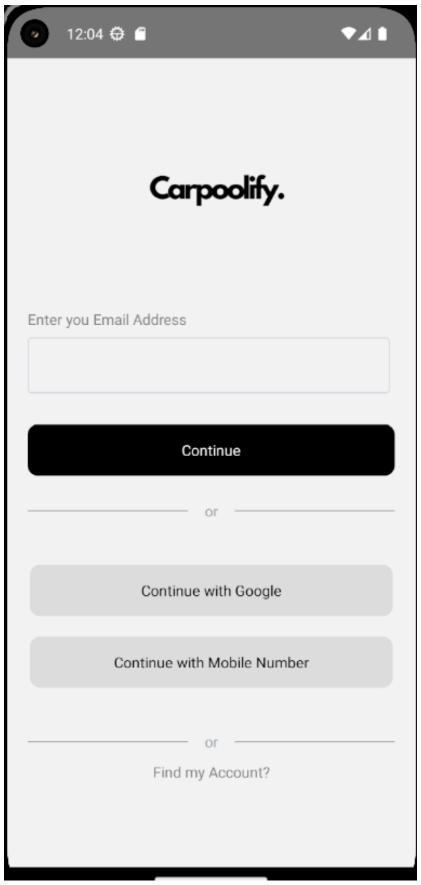


Figure A.2: Login Screen

A.0.3 Location Permission

Carpoolify offers a feature that allows users to share their current location, ensuring an accurate pickup experience. Users can grant the app access to their device's location services, providing a smooth and convenient way to set their pickup location.

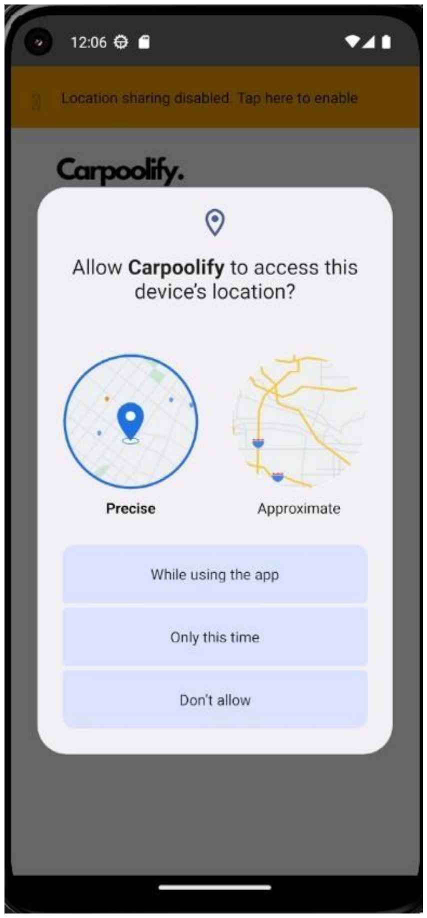
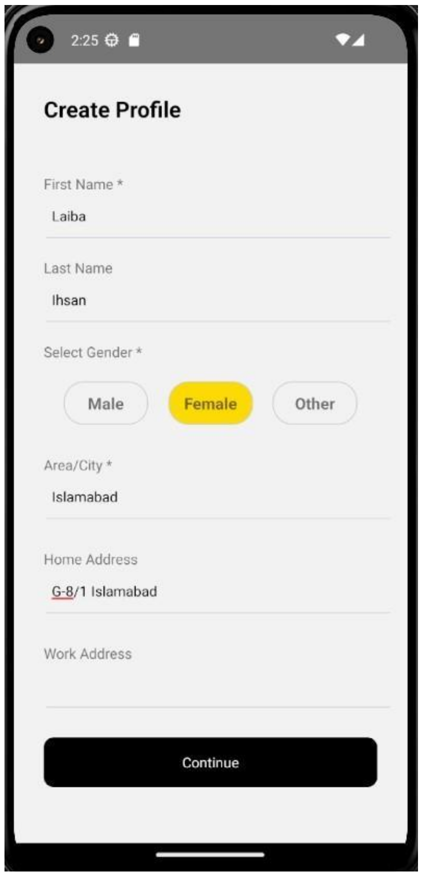

12:06
Location sharing disabled. Tap here to enable
Carpoolify.
Allow **Carpoolify** to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow

Figure A.3: Enable Location Screen

A.0.4 Create User Profile

The user can create their profile with basic information, firstname, lastname, gender, area and addresses that will be saved in database.



Create Profile

First Name *
Laiba

Last Name
Ihsan

Select Gender *
Male [✓ Female] Other

Area/City *
Islamabad

Home Address
G-8/1 Islamabad

Work Address

Figure A.4: Create User Profile Screen

A.0.5 Main Screen

The user will be displayed with the main screen after successful login to the app. Here user can access the app’s services.

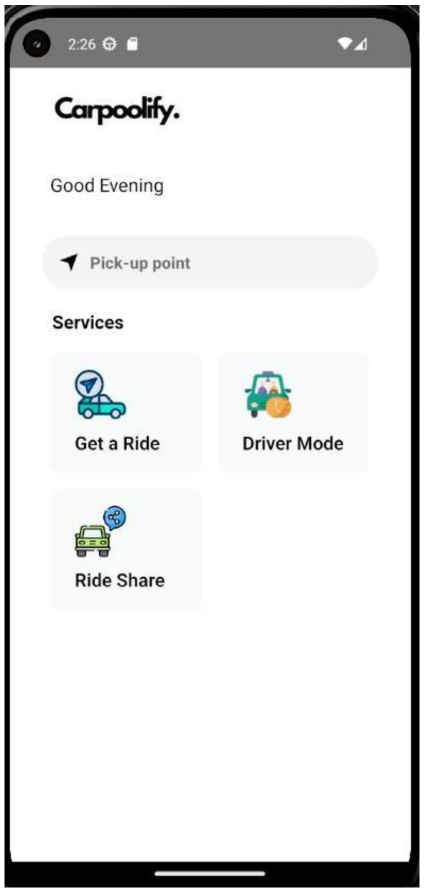


Figure A.5: Main Screen

A.0.6 Choose Location Screen

The user can manually enter the location in input fields.Here Google Places Autocomplete Api will be called after entering 3 keywords, user will get suggestions of relevant locations.

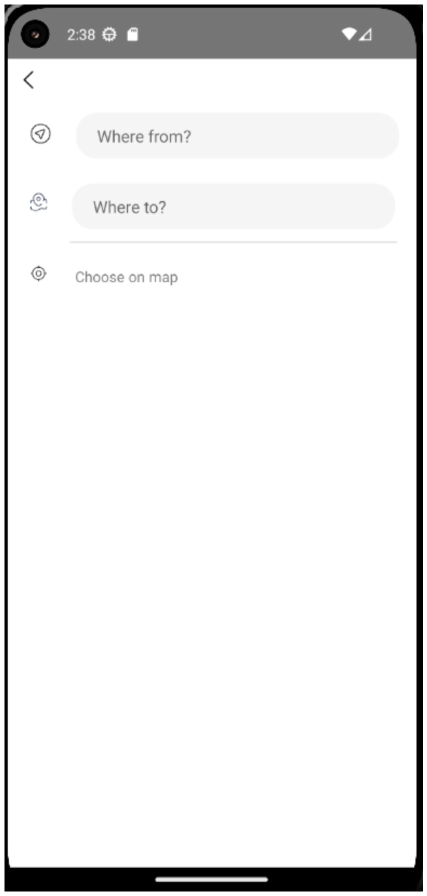


Figure A.6: Choose Location Screen

A.0.7 Choose Location from Map Screen

The user can also select location from map by touching the desired points. The selected source and destination latitude and longitudes will be converted to addresses and saved in databases.

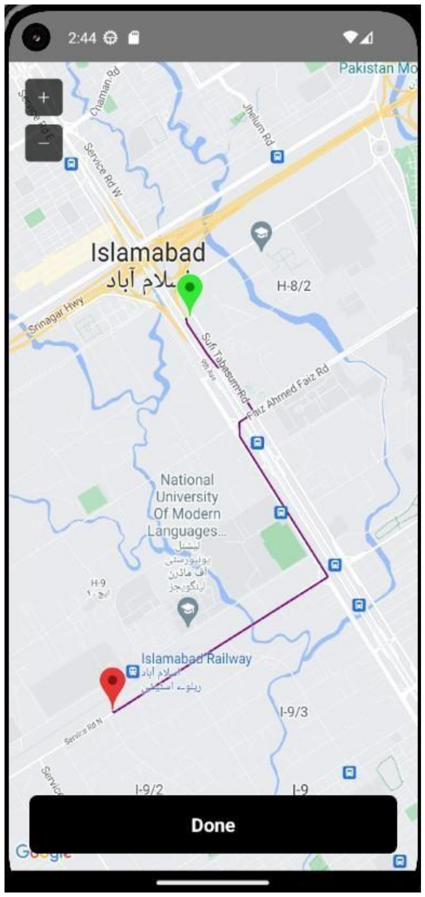
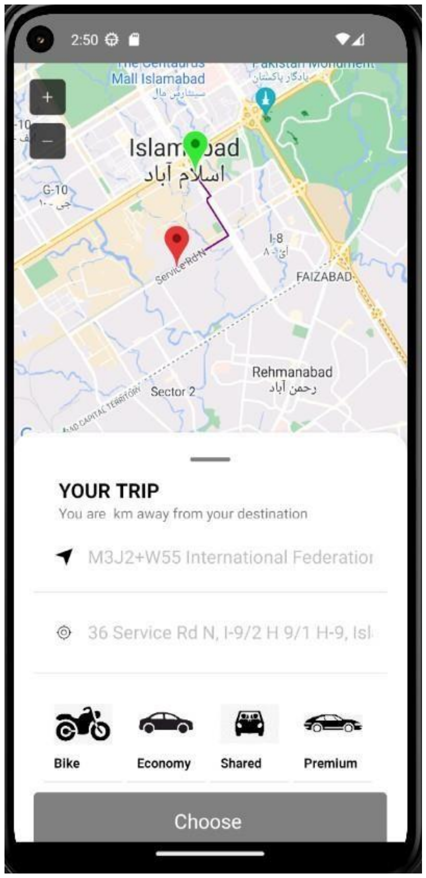


Figure A.7: Map Location Screen

A.0.8 Book Ride Screen

Users can conveniently input their desired source and destination addresses, which will then be presented in the designated input fields. Additionally, users have the option to select their preferred car type for their upcoming travel, offering a tailored and personalized experience.



You are km away from your destination

M3J2+W55 International Federatioı

36 Service Rd N, I-9/2 H 9/1 H-9, Isl

Figure A.8: Book Ride Screen

A.0.9 Ride share Screen

Users have the flexibility to manually initiate a shared ride request by ac-cessing a dedicated component designed specifically for this purpose. Thiscomponent provides a seamless and user-friendly way for individuals to re- quest a shared ride, connecting them with other users who are traveling inthe same direction.

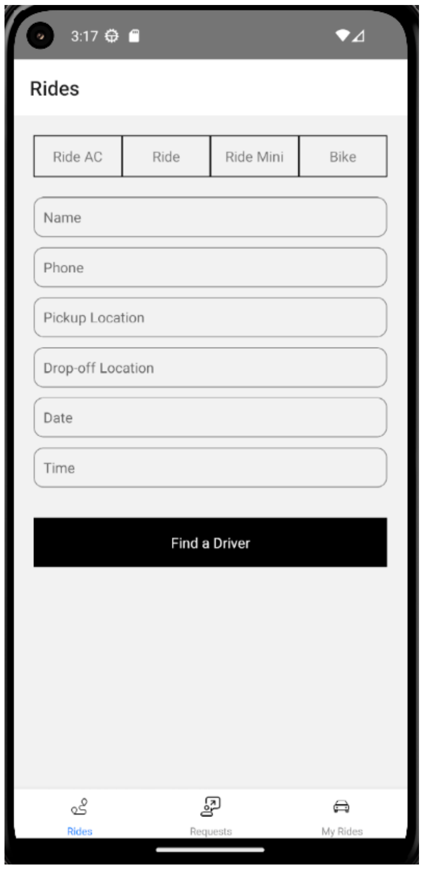


Figure A.9: Rideshare Screen

A.0.10 Driver Screen

In case the current user has not yet registered as a driver, they will need to initiate the creation of their driver profile as the initial step. This driver profile setup is crucial to access and utilize the full range of services and features tailored for drivers within the application.

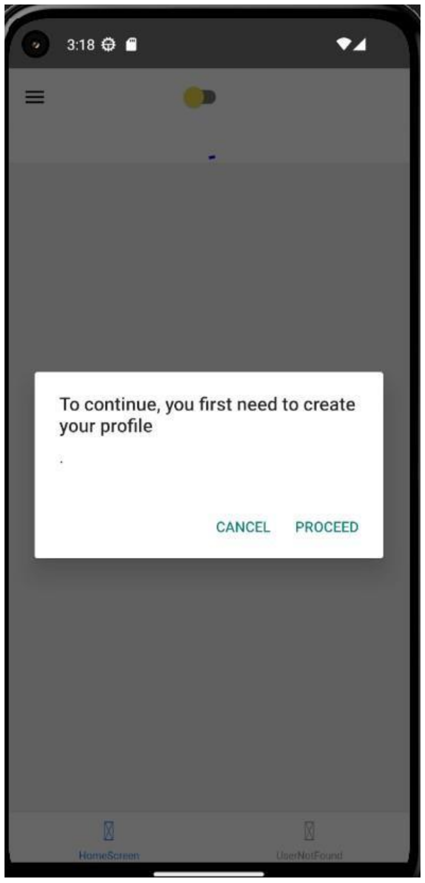

To continue, you first need to create your profile

Figure A.10: Driver Screen

A.0.11 Driver Registration Screen

Drivers will initiate their registration by generating a unique token, which serves as the key to unlocking and personalizing their driver profiles. This seamless process ensures that drivers can provide rideshare services with ease and convenience.

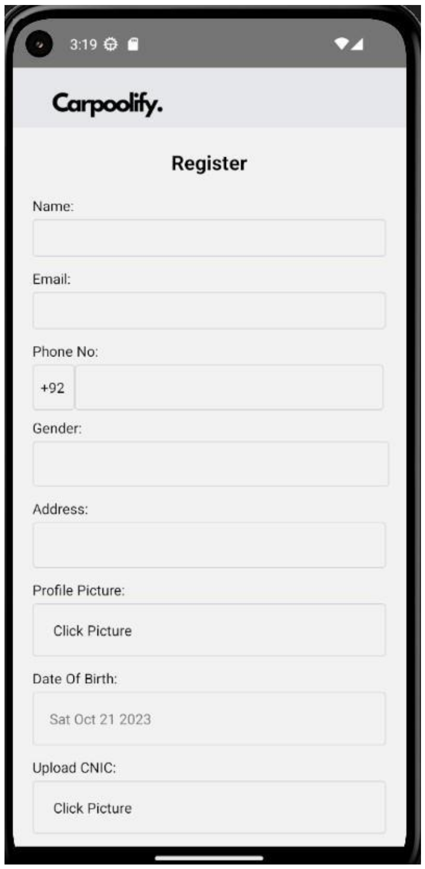


Figure A.11: Driver Registration Screen

A.0.12 Driver’s Map Screen

After successful profile creation driver will be redirected towards the Map screen, where a red marker will be displayed sowing the driver’s current location and this marker changes with the driver’s location.

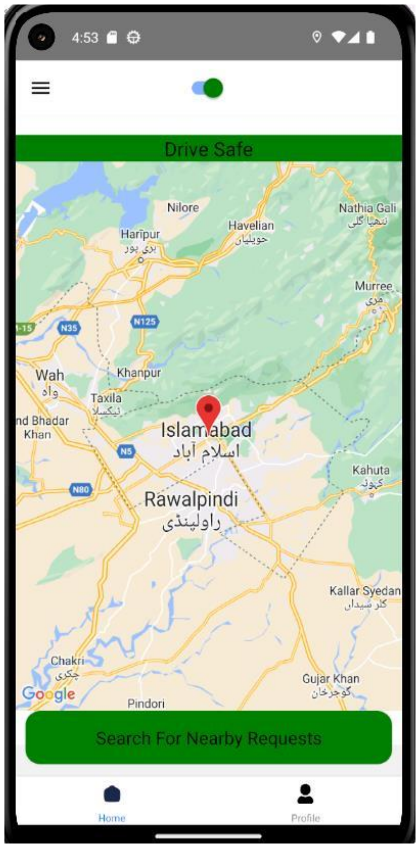


Figure 5.12: Driver Map Screen

Bibliography

[1] D. B. V, “Secure decentralized vehicle sharing system using blockchain,” *ResearchGate*, 2023.

[2] “What is blockchain technology?,” 2023.

[3] P. Julagasigorn, “What encourages people to carpool? a conceptual framework of carpooling psychological factors and research propositions,” *ScienceDirect*, 2021.

[4] P. Eleonore, “The future and sustainability of carpooling practices. an identification of research challenges,” *ResearchGate*, 2021.

[5] V. Valastin, “Blockchain based car-sharing platform,” *ResearchGate*, 2019.

[6] L. Bocker, “Sharing for people, planet or profit? analysing motivations for intended sharing economy participation,” *ScienceDirect*, 2017.

[7] A. Hayes, “Blockchain facts: What is it, who it works and how it can be used,” 2023. [Accessed: 17-August-2023].

[8] O. Technologies, “The evolution of consensus mechanisms: From proof of work to proof of stake and beyond,” July. Accessed: 17-August-2023.

[9] H. Hakke, “Blockchain based payment method for secure transactions.” Accessed: 18-August-2023.

[10] “Unified modeling language (uml) — activity diagrams.” Accessed: 20-August-2023.

[11] “Project implementation overview.” Accessed: 23-August-2023.

[12] "Why did we build visual studio code?." Accessed: 23-August-2023.
[13] "Run apps on the android emulator." Accessed: 23-August-2023.
[14] "Why choose react native for mobile app development?," 2023.
[15] D. Krasovskaya, "Mobile usability testing – the complete guide," 2023.