



Faculty of Engineering and Technology

Department of Computer Science

COMP 4200 - Introduction to Graduation Project

**Section - A**

**Title of Project:**

**CarPal: Ridesharing App**

**Group Members with IDS:**

---

BELAL HMEIDAT, 1202295

LOTFI HAJI, 1202064

BASEL ABU HAMED, 1202397

**Supervisor:**

NAEL QARAEEN

## **Section - B**

**Title of Project:**

**CarPal: Ridesharing App**

**Project No.:**

9

**Supervisor:**

NAEL QARAEEN

**Key Areas:**

Transportation Service, Carpooling, Ride-sharing, Mobile App, Flutter, Firebase, Java Spring Boot

## **Abstract**

The project aims to develop a carpool mobile app that facilitates sharing rides between people going to the same destination. There are two parties involved, drivers and passengers, and the app will facilitate their ride transactions so they split the travel costs. The app will benefit both drivers and passengers, as drivers can cover gas fees for their trip or even make a small profit while the passengers get to their destination faster, in one commute, and when public transportation is not available. The project's goal is to reduce travel expenses and improve access to rides in areas with limited public transportation. The application will be supporting both Android and iOS as we are going to be mainly using cross-platform technologies such as Flutter coupled with Java Spring to manage the backend, and Firebase database.

# Contents

<b>I First Semester</b>	<b>1</b>
1 Introduction . . . . .	2
1.1 Overview . . . . .	2
1.2 Aims and Objectives . . . . .	3
1.3 Technologies . . . . .	3
2 Literature Review and Similar Projects . . . . .	7
2.1 Similar Ideas . . . . .	7
2.2 Relevant Research . . . . .	12
3 Approach . . . . .	16
3.1 Methodology . . . . .	16
3.2 Resources . . . . .	16
3.3 Considerations . . . . .	18
3.4 Work Schedule . . . . .	19
4 System Analysis . . . . .	20
4.1 Functional Requirements . . . . .	20
4.2 Non-Functional Requirements . . . . .	23
4.3 Use Cases . . . . .	24
4.4 Sequence, UML, State, and Deployment Diagrams . . . . .	32
5 Initial Prototype . . . . .	41
6 Conclusion and Future Work . . . . .	42
6.1 Conclusion . . . . .	42
6.2 Future Work . . . . .	42
<b>II Second Semester</b>	<b>44</b>
1 Introduction . . . . .	45

2	Work Plan . . . . .	45
3	Changes to Part 1 Requirements . . . . .	45
4	Implementation . . . . .	47
4.1	Firebase Setup . . . . .	47
4.2	User Sign up and Login . . . . .	48
4.3	Driver Trip Creation . . . . .	49
4.4	Passenger Trip Search . . . . .	50
4.5	User Live Location . . . . .	50
4.6	Profile Picture Picker . . . . .	52

# List of Figures

1.1	Firebase Logo [3] . . . . .	5
1.2	Flutter Logo [5] . . . . .	5
1.3	Spring Logo [7] . . . . .	6
2.1	BlaBlaCar App Pages Overview [9] . . . . .	8
2.2	Pictures from withing Gett Taxi App [12] . . . . .	9
2.3	Screens showcasing inDrive app [14] . . . . .	10
3.1	Test Case Route <sup>1</sup> . . . . .	17
4.1	Use Case Diagram . . . . .	31
4.2	Login Workflow . . . . .	32
4.3	Showing Passenger's Accepted Trips Action . . . . .	33
4.4	Passenger Payment Workflow . . . . .	34
4.5	Showing Driver's Scheduled Trips Action . . . . .	35
4.6	Passenger Request Workflow . . . . .	36
4.7	UML Class Diagram . . . . .	37
4.8	State Diagram for Passenger . . . . .	38
4.9	State Diagram for Driver . . . . .	39
4.10	Deployment Diagram . . . . .	40
5.1	CarPal App Initial Prototype <sup>1</sup> . . . . .	41
2.1	Refined Work Plan . . . . .	46
4.1	Firebase Console . . . . .	47
4.2	User Sign Up and Login Screens . . . . .	48
4.3	Driver Trip Creation Screens . . . . .	49
4.6	User Live Location Demo - iOS . . . . .	50
4.4	Passenger Trip Search Screens . . . . .	51
4.5	Passenger Trip Search Screens . . . . .	51

4.7 Profile Picture Picker Screens . . . . .	52
--	----

# List of Tables

1.1	Relevant Firebase Features [2] . . . . .	4
2.1	Our app compared to similar other known apps. . . . .	11
3.1	Work plan for the left tasks and the time needed to achieve them starting from 08/11/2023 . . . . .	19
6.1	Work plan for the tasks and the initial estimated time needed to achieve them. .	43

# **Part I**

## **First Semester**

# **1 Introduction**

## **1.1 Overview**

When it comes to commuting from one place to another in Palestine, options can be narrowed down into two options, public and private transportation services and privately owned automobiles; people relying on the latter might carry with them colleagues, family, or friends or may be traveling alone. Here we will give an overview of the downsides involving these two norms of commute, briefly introduce the idea of our app, and how it fits among these two.

Public transportation is very dependent on in Palestine. Thousands of people use them every day. However, it suffers from a lot of problems that vary in severity from one area to another. The service is only available up to a certain time in the day. In addition to the long time the roads take, passengers waste a lot of time waiting for cars to come by or fill up with passengers. During rush hours when there are a lot of people and not so many buses available, people have to compete to get into a car. Moreover, and while not a big deal, it can be a hustle to find where these buses are stationed in each city/ town, even costing another transportation to get to them. On the other hand, there are a lot of people who use their cars every day to get to their jobs. They can pick up a friend or relative along their way but that's not always the case as many will go empty. This method can provide individuals with flexibility and freedom, and save them the wait time. However, it can be very pricey given the road traffic situation here and the gas prices. Here we come to introduce our idea of organized yet very accessible carpooling, where anyone with a car can sign up for an account to offer to share their ride with people going to the same destination as them and split the cost of the trip together.

The implementation of this project revolves around a mobile app that allows people to book trips with other people driving to the same destination. The two parties interacting with the app are the passengers and the drivers. The app will facilitate scheduling a trip, booking it, paying for it, and rating it at the end of each trip. The app will feature an easy setup process for both the passengers and drivers to get started using it quickly. The app will let all users create an account while users who want to become drivers will need to undergo a verification process. After signing up, drivers can schedule trips whilst passengers can request trips for passing by drivers to accept and upon getting accepted proceed to pay. The app will have intuitive features

involving suggesting matching trips, displaying live driver location on the map, tracking trip progress, user profiles, and the ability to provide ratings and feedback from both sides. The app will let users pay using easy digital payment methods available via local payment services e.g., PalPay.

We believe this project will help people with travel expenses, especially those who have to travel to work/ study frequently and would like to make up for travel expenses without sacrificing much of the freedom and independence of owning a private car. Similarly, people who rely on public transportation to commute regularly can avoid the waiting times and inconveniences of public transportation services while still paying the same affordable fee. This app will also greatly help people find a ride at times when public transportation services are lacking.

## **1.2 Aims and Objectives**

### **Aims**

Developing a Mobile application that enables users to be able to have multiple options when traveling from one place to another and to give them what we hope is a better experience during their trips.

### **Objectives**

To develop a carpooling community in Palestine through the application.

To develop an application that is easily accessible to a wide range of users.

To achieve a safe carpooling community that is supported by people.

To improve the traveling experience and choices for users both passengers and drivers.

## **1.3 Technologies**

We are trying to use relevant technologies that are easy to implement, common, up-to-date, and compatible with our previous experiences.

### **Firebase**

”Firebase is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication and integration for a variety of

applications.” -Firebase Wiki [1]

**Why Firebase?** Our app needs to have a real-time connection and the ability to synchronize data throughout different devices, in addition to having the Google Analytics support just by activating Firebase on the application, which will help with tracking customers events and get feedback on the application which will help us with scaling our project in the future, and by that we would be having a huge benefit especially when using something like Firebase which is developed and supported by Google. Here are some benefits of using Firebase:

<b>Real Time Updates</b>	Firebase uses data synchronization rather than typical HTTP requests. It allows all connected devices to receive immediate updates every time data changes. This helps with the real-time location and messaging through our app without having to dive into abstract networking.
<b>Offline Data Availability</b>	Firebase Realtime Database SDK persists the data to disk keeping it available offline. It syncs with the client device once connectivity is reestablished.
<b>Ease of Access</b>	The Firebase Realtime Database can be accessed directly from a mobile device or web browser. There's no need for an application server. Moreover, it has security and data validation available through the Firebase Security Rules.
<b>Google Integration</b>	Google Analytics for Firebase will allow for tracking users' trips through real-time and custom reporting. According to Google, ”Firebase provides unlimited free reporting on up to 500 distinct events. Just like the regular Google Analytics, Google Analytics for Firebase automatically tracks certain key events and user parameters straight out of the box, and allows you to define custom events that are important to your application.” -Firebase
<b>Scalability</b>	Firebase can scale well for real-time updates, which is crucial for a car-pooling app with a potentially large user base.

Table 1.1: Relevant Firebase Features [2]



Figure 1.1: Firebase Logo [3]

## Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, Mac OS, Windows, Google Fuchsia, and the web from a single code base. [4]



Figure 1.2: Flutter Logo [5]

**Why Flutter?** With Flutter, we can save a lot of time and work by writing the code once knowing it will work on both of our target platforms iOS and Android. The main reason we are using Flutter over other cross-development frameworks such as React Native is due to our previous experience developing in Flutter. Flutter while it is still not as fast as native development, still outperforms React Native. Also since we are using Google Maps API for routes, and Google's Firebase as a Database, we expect Flutter will work more easily and reliably with them.

## Java Spring Boot

Java Spring Framework is a popular, open-source framework used for creating standalone, production-grade applications that run on Java Virtual Machine. [6]



Figure 1.3: Spring Logo [7]

**Why Java Spring Boot** All of our team members have current and past experience working with Java and Java Spring Boot. This will make it easy for us to develop, debug, and maintain the back end of our app. Given the reliability of this framework, we can expand our app with new features and take it to new places such as the web. This is not an inclusive list. Other technologies and APIs as well as the specific libraries that are going to be used will be mentioned here when we discuss further details.

## 2 Literature Review and Similar Projects

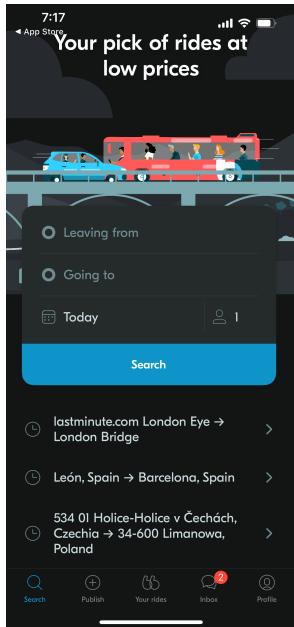
In this chapter, we give a brief overview of other similar projects that we took inspiration from and how will they compare to our project. We also list some research papers we found helpful to our project, how they are related, and how they shaped the way we are going to take certain approaches in our project.

### 2.1 Similar Ideas

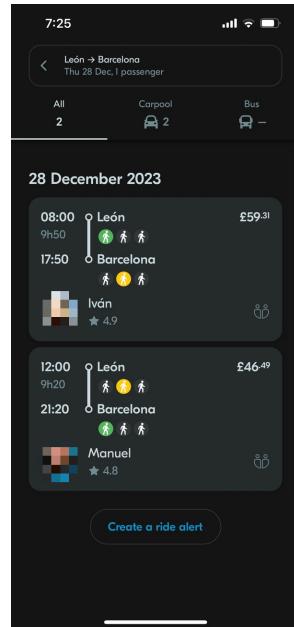
Here is how our app will compare to other similar projects. We have selected these projects each for a different reason. BlaBlaCar is the most similar app as it is mainly a carpooling app for all the people to share a trip and split the cost. Gett is a carpooling app that is widely known as it's active in our target area while inDrive is a very feature-rich carpooling app.

#### BlaBlaCar

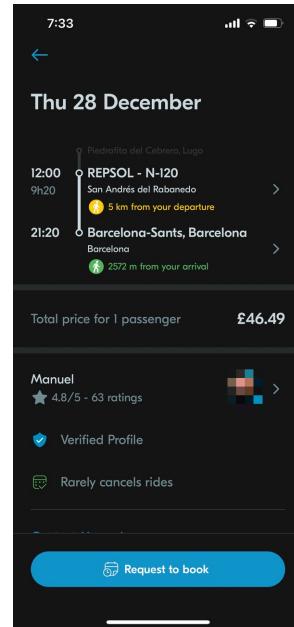
French carpooling company that lets everyone make trips and share the expenses with those who are going to the same destination. BlaBlaCar is available in 21 countries and has an easy setup and booking process. The app has a clean and easy-to-navigate interface that lets passengers and drivers book and schedule trips easily. It offers a unique aspect built solely around organized social ride-sharing to reduce trip costs. It doesn't contract drivers working for it which allows everyone to become a verified driver just by uploading some papers. The app makes it clear that its purpose is ride-sharing to reduce costs; it prohibits the use of its service as a means of bringing income to the driver. The company also offers its bus services from within the same app. Since their app is operational in European countries where hitchhiking and expense-sharing trips are widely accepted, the app has few restrictions and requirements for its users. [8] BlaBlaCar app is the closest to our idea and is its primary source of inspiration; therefore, our app shares many features and implementations with it such as checking the profile of other users to get to know who you will be riding with and setting specific rules for the trip. Additionally, our app will offer new features like on-the-way pickup, live map location services, Arabic language support, and a localized framework that fits into the roads situation in the West Bank.



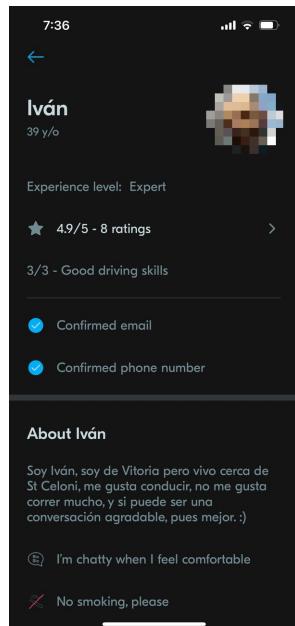
(a) Search Page



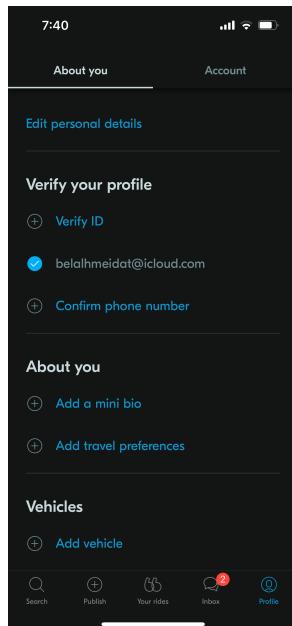
(b) Trip Search Results Screen



(c) Trip Detail Screen



(d) User Profile Screen

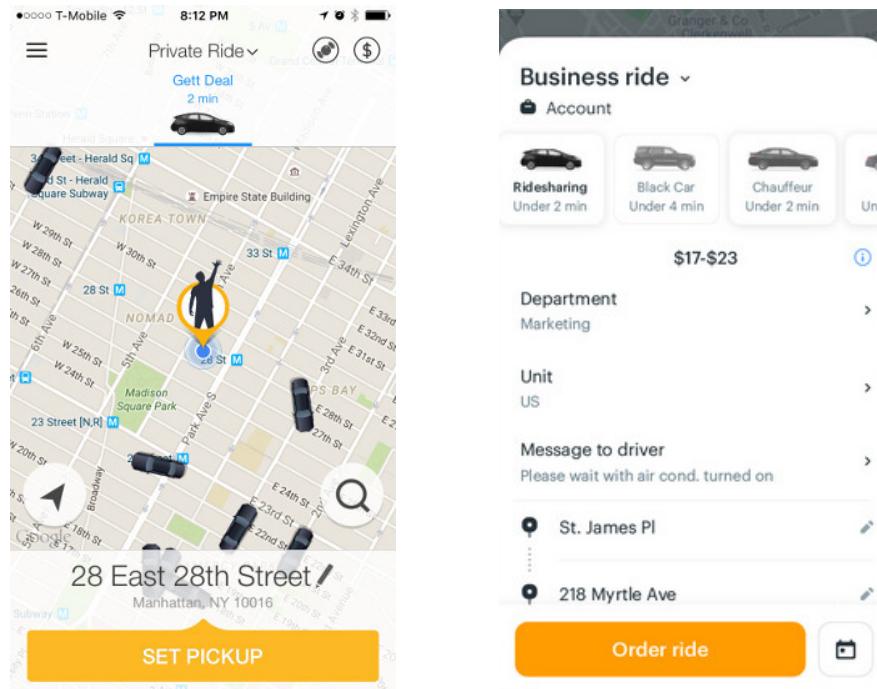


(e) Settings Screen

Figure 2.1: BlaBlaCar App Pages Overview [9]

## Gett

Also called Gett Taxi, which is a B2B Ground Transportation Management (GTM) platform, which was designed to help people save a lot of time and effort as it offers a faster, easier, and cheaper way for people to find transportation from one place to another. [10, 11] Gett is available and operating in Palestine particularly inside pre-1948 borders. The app offers many quality features such as being able to see drivers on the map and utilizing it to choose exact pickup and drop-off locations. Moreover, the app allows users to choose the type of ride and car size they want: shared ride, chauffeur, SUVs, and more. While being fairly popular around, the company's business model is primarily providing cab services rather than ride-sharing, hence the name Gett Taxi. It can offer a better experience for passengers going on long trips more than public transportation services do but it doesn't have other noteworthy advantages over them.



(a) Ride Location on Map

(b) Trip Detail Screen

Figure 2.2: Pictures from withing Gett Taxi App [12]

## inDrive

International ride-hailing service and one of the most popular carpooling apps in the world. It sits as the second most downloaded ride-sharing and taxi app. [13] In addition to being a taxi alternative, inDrive is used for many services such as courier delivery, freight transport, and professional services such as home repairs and tutoring. inDrive appeals to its user base by

allowing them to negotiate a price with different drivers and thus helping them find the price they're looking for. The app itself allows users to select a service and select locations on the map. It also shows nearby drivers on the map. It also gives users access to message or start a call with the driver from within the app.[14] Despite being very popular, inDrive is not available in Palestine. The business operates differently from BlaBlaCar and our app as it falls in line with the workflow of other famous carpooling apps such as Uber which is primarily a carpooling app that drivers rely on for bringing in income rather than merely sharing a ride. It also offers more general use cases such as professional maintenance services.

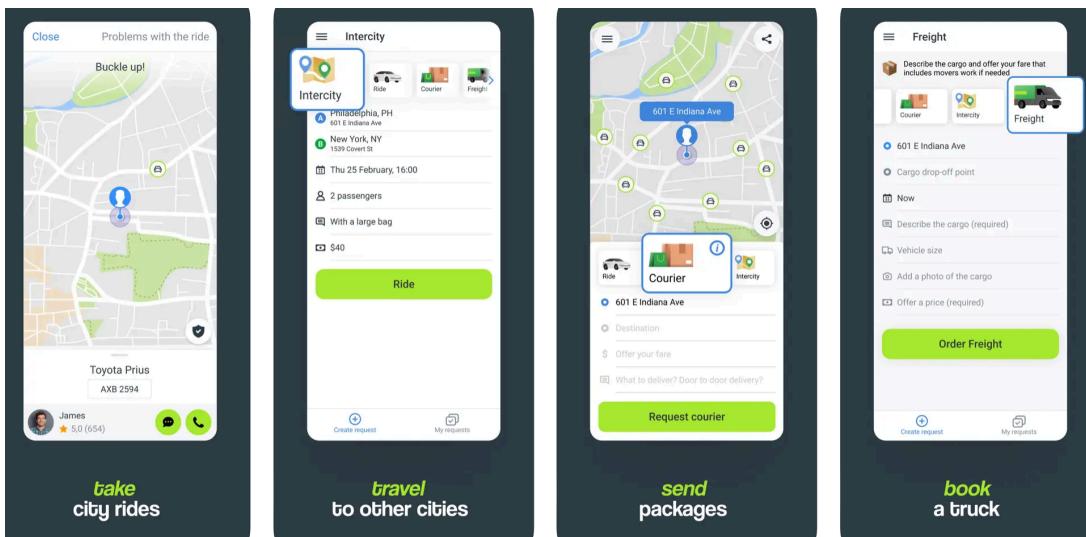


Figure 2.3: Screens showcasing inDrive app [14]

Here is how they compare to our app:

Table 2.1: Our app compared to similar other known apps.

Feature	inDrive	BlaBlaCar	Gett	Our App
Anyone can make a trip after verification (no contract)	✓	✓	✓	✓
Ability to pay beforehand through the app	✓	✓	✓	✓
Driver can schedule trips days prior	✓	✓		✓
Passengers can browse as a guest		✓		✓
Passengers can filter through drivers	✓			✓
Users can set rules visible on their profile		✓		✓
Cancellations allowed	✓	✓	✓	✓
Pickup Locations				
Pickup on the way				✓ (from stations)
List and map of trips from nearby locations <sup>1</sup>				✓
Alternative routes				
Provide other trip suggestions to complete the trip*				✓
Provide public transportation station information to be able to complete the trip *				✓
Map integration				
Trip route on the map		✓	✓	✓ (limited)
Live location of drivers <sup>2</sup>	✓		✓	✓
Communication and Trip Sharing				

## Our app compared to similar other known apps (continued)

Feature	inDrive	BlaBlaCar	Gett	Our App
User profile with rules, cars, trip counting, and ratings	✓	✓		✓
Users can see other passengers' profiles in the same trip <sup>3</sup>		✓		✓
In-app chat for passengers and driver*	✓	✓		✓
Rating and feedback for both passengers and drivers		✓		✓

\* Feature is not of high priority. We will try to implement this feature if time allows but don't promise to.

<sup>1</sup> The passengers can see a list of drivers heading to their destination sorted by certain criteria as well as a map showing nearby locations of drivers in the list.

<sup>2</sup> location snapshots if live location feature is found not viable.

<sup>3</sup> Users can see the name and rating of a passenger who is on the same trip when browsing trips. They can visit their profiles after booking the trip.

## 2.2 Relevant Research

### Carpooling, Who, Why, and How

In his article about who and why people do carpooling, ROGER F. TEAL [15] outlines some of the factors that make carpooling appealing to people or not. He mentions numerous statistics and research done combined with comprehensive analysis on the topic. While the article is based on the U.S. environment, there are general conclusions that can be drawn from the paper that serve as useful indicators to guide us in how we approach our app. One of which is his conclusion that socio-economic factors have little effect on one's propensity to carpool. This conclusion has been echoed multiple times by other researches he mentioned in his paper and ones that we came across in our further readings as well. That being said, he later states that people of lower income are more willing to carpool. The paper also suggests that women are more likely to carpool than men. He also brings up the agreement among researchers he mentions in his paper that carpoolers travel much bigger distances to work than people who commute alone.

This might be the case for Palestine as well since examples of people carpooling with others going to their work in further cities are fairly common. This information is helpful since our app relies on stations of pickup. The range these stations cover should be larger if people are more inclined to carpool to far places. This will streamline the station selection to fewer stations that people can use reliably instead of creating many other substations that people would more likely use other traditional transport methods to get to. He also finds that the percentage of unrelated people who carpool together is larger than those who are related. This is to be expected as the chance of family members working in the same place or at the same time is unlikely. However, in Palestine, families are fairly larger than American families; so the difference between these two percentages might be smaller. There is still a sizeable percentage of people who carpool with unrelated people and could benefit from our app to find them. Moreover, Dr. Teal finds that the number of unrelated people carpooling is higher than that of internal carpooling, where people carpool with their relatives, by 2.25 to 2.63 people respectively. He finds that internal carpooling is incentivized by vehicle shortage whereas external ones are motivated primarily by cost burden.[15] Since our app is mainly focused on external - non-related - carpooling, we will have to focus on ways to make the app cost-cutting, maybe by limiting the price people set or suggest a recommended price tag that is proportional to the travel cost.

## **What Motivates Carpooling**

In their paper about what encourages people to carpool [16], Jun Guan Neoh, Maxwell Chipulu, and Alasdair Marshall dive into the factors that affect people's willingness to carpool. They reviewed 908 papers from previous studies and filtered them down to 19 eligible articles to perform a meta-analysis on them. They summarize the literature on the carpooling field as "rich in studies but doubts remain about the generalisability of the results, making it difficult for policy makers to translate the findings into practice." Their research splits the factors affecting carpooling into internal individual and judgemental factors and external environmental and situational factors. Like Teal (1984) [15] and many other publications on the topic, this research agrees that socio-demographics don't strongly influence carpooling. One key point drawn from their research regarding socio-demographic factors and one which agrees with Teal (1984) [15] is that women carpool more than men do. We don't know if this translates well to Palestine but we aim that our app will be usable for both men and women easily by taking some measures that are friendly with the gender rules in Palestine such as giving female users the ability to filter

female drivers only. In terms of situational factors, the research also confirms the previously discussed findings that carpools favors longer distances. It also claims that the inconvenience of waiting for other carpools members can be a major factor in pushing away people from carpools. We see that this issue might be exacerbated in our app since we are allowing drivers to pick up people on the way. We need to bring countermeasures that will make pickup easier and faster. One such measure is limiting the number of people a person can pick up. In addition to the ability of drivers and passengers to communicate and decide on a pickup location, we also discussed utilizing the map so passengers can send their exact pickup location from the start when requesting the trip to the driver who can, in return, agree or send a new waiting location for the passenger. As for judgemental factors, the research notes that psychological factors play a major role in making carpools decisions. More so than socio-demographic factors. People feel more comfortable carpools with their relatives and co-workers than when they are carpools with strangers. Commuter privacy is also a big factor as carpoolers feel a sense of giving up some privacy and personal space when carpools. However, since public transportation is very popular in Palestine, we feel carpools will be just as acceptable in this regard as it offers similar, if not better, privacy levels. The study also points out that a sense of control can encourage people to carpool, we believe we can achieve this in our app by offering the option to find trips that agree with the user's set rules. We expect people's willingness to carpool can be negatively affected by their inclination to socialize. We are trying to mitigate the effect of this by designating a rule for chatting and socializing. We can also benefit from the fact that our app is oriented around picking up people along the way, which in turn, encourages shy people to participate knowing that other people will be on the trip with them providing them similar environment to familiar public transportation. Pollution is also a factor that encourages people to carpool according to this research but while reducing carbon footprints is one of our goals, we don't believe it is a major incentive for carpoolers in Palestine. Nonetheless, we see that traffic plays a similar role in encouraging people but we don't know to what extent since the traffic and road situation in Palestine can be abnormal most of the time. Moreover, the research results point out the important and the not-so-significant factors that affect carpools. It finds that transportation costs, the number of employees in the carpooler's workplace, and finding potential partners have the biggest effect. Some other less important factors include the availability of carpools lanes, fixed working schedules, and urban areas. Factors such as age, marital status, parking costs, parking availability, and being a university student, have negligi-

ble effects on carpooling. [16] Overall, we believe this research gives very good insight for us to decide on how we implement certain aspects of our app and will try to make good use of it in this project and future works to come.

## **Carpooling Challenges**

In their research paper on carpooling and its economic implications for drivers [17], Hai-Jun Huang, Hai Yang, and Michael G.H. Bell address the challenges associated with carpooling, including inconvenience for those with flexible hours, increased travel time, and concerns about privacy and independence. The study primarily focuses on the economic effects of carpooling, employing two main models: deterministic and stochastic.

Where deterministic mode, explores passengers' behavior and thoughts about carpooling, and driving alone modes under various scenarios, including no-tool equilibrium and social conditions. On the other hand, the stochastic model takes the path of logit-based modeling, considering different factors such as fuel cost, assembly cost, value of time, and traffic congestion.

The research adopts three main approaches to studying carpooling which are: estimating ride-sharing potential, predicting ride-sharing demand, and considering demand and supply effects. Several numerical examples are shown in the paper that demonstrate the impact of carpooling on car trips for users, which indicates a decrease in the overall cost for both passengers and drivers. To conclude, this research provides valuable insights into the economic aspects of carpooling, showing users' behavior and preferences, and offering a good understanding of the challenges and benefits that this mode of transportation can bring with it. [17]

## 3 Approach

### 3.1 Methodology

To ensure an effective development process, we have chosen to adopt the Agile development methodology [18] as the software development process we will be taking throughout our journey.

Where this approach is known for its success in delivering reliable software, we will go through our project using iterative cycles, allowing for flexibility during the development process.

Following the selection of the software development process, our initial step was surveying to gather data from potential users to see what people would expect and want from such an application, then we compared existing carpooling applications to see what we could improve or add to our application to make it stand out. Following that we looked into different technologies suitable for our cross-platform mobile application, ensuring they best suit our idea.

With all that, we transitioned to the formulation of system requirements. In this phase, we gathered both functional and non-functional requirements, and then we moved to the creation of diagrams such as use cases, etc.

Moving forward in our development road, we start working on the application development process, starting with the establishment of the database, interfaces, and backend of our application, with technologies such as Flutter for the front end and Spring Boot for the back, Firebase for the database.

Where this methodology ensures that each phase of the development is well planned and executed, which will lead us to have a more robust and reliable development process.

### 3.2 Resources

#### West Bank Cities Roads Dataset

A dataset of all the roads commonly used to commute from one Palestinian town or city to another in the West Bank. This will be used to define fixed stations that allow for passengers to be picked up along the way from towns and cities that drivers normally pass by to arrive at their destination. There is no already made database that we know of as of the writing of this draft, 08/02/2024. We are to ask the responsible parties for one or start collecting it ourselves.

Due to the narrow time window of the project, the scope of operational routes, or the routes the app will work on, is reduced to a smaller area to use as proof of concept and a test case for development; we chose the route of Qalqilya - Ramallah for this, see the figure 3.1 showcasing possible routes and possible stations in between. We aim to scale the app's operational scope from this to the whole of the West Bank wherever possible.



Figure 3.1: Test Case Route<sup>1</sup>

---

<sup>1</sup>Orange denotes a branch.

## **Public Opinion Questionnaire**

A questionnaire to get public opinion on the idea. We will assess people's acceptance of the idea, whether they will partake in it as drivers or as regular users, what their concerns are, and ask them for any input or suggestions.

### **3.3 Considerations**

#### **User Privacy Consideration**

We will make sure the app respects user privacy. The app will be asking for user permission to access all needed device sensors and data such as camera access, location, calendar, and contacts if needed. It will not partake in collecting user data but for their necessary information such as phone number, real name, and payment method information. We will maintain the secrecy of this data and store it in a secure encrypted database. An in-app chat to allow passengers an easy channel with the driver is discussed. In case it is implemented we do not promise utilizing an end to end encrypted messaging. Lastly, users will be allowed the option to delete their accounts from our database at any time.

#### **Effect on Public Transportation Sector Consideration**

Our app is not meant to replace the public transportation sector and is not intended to be used as a main source of income for drivers. We have taken into consideration measures to combat that such as suspending accounts that do partake in that.

#### **Legal Consideration**

In case of a reported felony or abuse committed by the driver or the passenger, we are willing to provide the driver's information to the legal authorities upon request. Access to users' (passengers and drivers) information can be acquired in case required by law. We are **not** to be held accountable for any of the drivers and passengers' actions.

### 3.4 Work Schedule

Table 3.1: Work plan for the left tasks and the time needed to achieve them starting from 08/11/2023

Task	Time
Proposing an idea and writing an abstract	1 week
Writing literature review and similar projects	1 week
Writing Introduction and searching tools to be used	1 week
Finding/creating a suitable dataset for transport	1 week
Reading relevant articles/ papers	1 week
Defining system requirements and use cases	2 weeks
Completing system analysis with class diagrams, sequence diagrams, and state chart diagrams	2 weeks
Creating system design with system structure and deployment diagram	1 week
Finalizing the report	1 week
Total	11 weeks

## 4 System Analysis

### 4.1 Functional Requirements

UR1. Users shall be able to log in and log out after creating an account.

SR1.1 The system shall allow users to sign up for the same type of account through the same page and app.

SR1.2 The system shall enable both drivers and passengers to log in, with the ability to stay logged in till logout is requested.

UR2. User shall have a profile where they can edit personal information, and add a set of rules for their trips.

SR2.1 The system should allow the user to add/edit their personal information.

SR2.2 The system should allow users to choose from a variety of rules to add to their trip preferences, such as smoking, music, etc.

UR3. Users shall search for trips by selecting the departure location, destination, and date for a trip as a member/guest.

SR3.1 The system shall look for drivers going to the selected destination on the same date and passing by the user's departure location.

SR3.2 The system shall display relevant trip details, including driver information, route, and rules, and expected arrival time.

SR3.3 The system shall sort the drivers list depending on the time and route taken whether they respect the user's set rules or not.

SR3.4 The system shall allow users to browse trips without signing in (as guests).

SR3.5 The system shall display a list of trips based on the user's given requirements.

SR3.6 The system shall provide locations of nearby drivers making the requested trip on a map mirroring the information provided in the list.

UR4. Users should be able to become drivers after undergoing a verification process.

SR4.1 System shall allow users with cars willing to collect passengers along their way to undergo a verification process to become verified drivers.

SR4.2 The system shall notify the Users if they were accepted or not.

UR5. Admins shall be notified about new applying Drivers.

SR5.1 The system shall send a notification to the responsible department about newly applied drivers

SR5.2 The responsible department shall evaluate the user's request and approve or deny their request.

UR6. Users shall be able to file complaints and reports which shall be received by the responsible department to resolve issues or provide support.

SR6.1 The system shall notify Admins about drivers' being reported more than 3 times.

SR6.2 The admins can review the complaints and take action if they are valid complaints.

UR7. Verified drivers shall be allowed to schedule trips.

SR7.1 The system shall allow drivers to schedule trips by selecting the departure and destination locations and time of the trip.

SR7.2 The system shall ask the driver to specify available seats and luggage rooms.

SR7.3 The system should allow the driver to select the route throughout the trip.

UR8. Users shall be able to report other users (drivers/ passengers) for any complaints.

SR8.1 The system shall allow passengers to report drivers

SR8.2 The system shall allow drivers to report passengers.

SR8.3 The system shall allow passengers to report each other if they were on the same trip.

UR9. Users shall be able to give feedback and ratings to different users.

SR9.1 The system shall allow passengers to provide feedback and ratings for their driver.

SR9.2 The system shall allow drivers to provide feedback and rating for their passengers.

SR9.3 The system shall allow passengers to provide feedback and ratings to other passengers on the same trip.

UR10. Passengers shall be able to book trips.

SR10.1 The system shall allow only signed-in members to book trips.

SR10.2 The system shall display relevant trip details, including driver information, route, and set rules.

SR10.3 The system shall notify the driver of this booking.

SR10.4 The system should wait for the driver to accept the request before confirming the booking.

SR10.5 The system shall request the passenger to pay for the trip.

UR11. Passengers shall be able to pay for the trip.

SR11.1 The system shall allow passengers to pay through different online payment methods.

SR11.2 The system shall complete the payment securely.

SR11.3 The system shall confirm the booking for the driver/passenger.

SR11.4 The system shall complete the payment for the driver when trip end confirmation is done.

UR12. Users shall be able to cancel trips (Both Drivers and passengers') and get refunds.

SR12.1 The system shall allow users to cancel the trip.

SR12.2 The system shall start the refund process for this trip.

SR12.3 The system shall notify the driver/passengers when cancellation is completed.

SR12.4 The system shall calculate the refund based on the time of the cancellation.

SR12.5 The system should allow for a full refund if the driver is too late from his planned arrival time.

UR13. The passenger shall be notified sometime before the driver's scheduled/ estimated arrival time.

UR14. Users (driver and passenger) should confirm that the trip is finished and will have the option to provide feedback when the trip is complete.

SR14.1 The system should receive confirmation from both the driver and passenger at the end of the trip.

SR14.2 The system will prompt the driver to rate the passenger.

SR14.3 The system shall prompt the passenger to rate the trip, the driver, and other passengers.

UR15. The user shall be able to change their account's settings e.g. change email, password, and privacy settings.

## 4.2 Non-Functional Requirements

1. The app must submit to the average acceptable response time for mobile apps which is 2-3 seconds.
2. The app must be reliable, this means that it must work correctly at 97%, and have a 3% downtime. This means 12 hours of downtime per year.
3. The app must submit to the industry standards security practices to protect collected and analyzed data.
4. The app must be designed with well-documented code to allow future maintenance and enhancements.
5. The app must be compatible with widely used Android and iOS versions, namely Android 10, iOS 13, and later.
6. The app must feature an intuitive UI that is easy to use and navigate.
7. The app must be scalable such as it can support the ability to scale the program in the future to keep up with rising demand.
8. The app must be recoverable, as the database will be backed up with scheduled plans.

## 4.3 Use Cases

### Actors Description

- Admin: This actor is responsible for verifying drivers, and reviewing reports/complaints from app users.
- Driver: This actor is responsible for scheduling trips, accepting passengers to his trip, and operating the trip.
- Passenger: This actor can search for trips, and request to join trips that are available on the application.
- Guest User: This actor can download the application, but is limited to only searching for trips and seeing them (can not request to join trips).

### External Actors Description

- Notification Manager: This service handles notifying actors about events such as informing admins of a newly filed report/ complaint from users, notifying admins about new requests from users wishing to become drivers, informing drivers of new ride requests from passengers, and reminding drivers and passengers of their upcoming trips.
- Map Services: This service is fired up whenever a new trip is added with its final destination located on the map, passenger checks to see a driver's live location on the map, or when passengers choose a point on the map to be picked up from and the driver approves or sends back a different point for pick up.
- Payment Services: This service is responsible for facilitating payments between passengers and drivers. It is used whenever a payment transaction is made.

### Use Cases Description

This is a description of the use cases shown in 4.1.

#### 1. Login/ Registration

Actors: Driver, passenger

The page users are welcomed the first time they launch the app. It also offers an option to browse trips in guest mode.

## 2. Login/Registration Verification

Actors: Driver, passenger

When a user signs up for a new account, they are required to verify their phone number and email. When they sign in, a verification code is sent to their confirmed phone number

## 3. Create an Account

Actors: Unregistered users

An unregistered user can create an account by entering the needed information: email, phone number, and password.

## 4. Login

Actors: Driver, passenger

A user can log in by entering his email and password to the login page.

## 5. Logout

Actors: Driver, passenger

A registered user can log in by clicking the logout button on the main screen.

## 6. Become Verified

Actors: Driver, Passenger

Users can choose to upload a picture of their ID card to earn a verified account badge.

This serves as a measure to increase other user's trust in them.

## 7. Show/Edit Profile

Actors: Driver, passenger

A user can see his profile details, and edit them from adding preferences to his trips to updating their name and profile picture. They can also view their trip history from their profile or apply to become drivers.

## 8. Apply to Become Driver

Actors: passenger

A passenger can apply to become a registered driver by uploading their car registration, and their driving license, which then is revised by the admins.

## 9. Notify driver about approval/decline for his application

Actor: notification manager

The notification manager should send a notification to the user to notify them about their driver's application being approved or declined by the admin.

#### 10. Upload Driver Documents

Actors: Passenger

A driver needs to submit certain documents to prove eligible to start making trips. These documents are their driver's license and their car license. They are also required to take a photo of their face to undergo a verification process after they are sent to the admins.

#### 11. Notify Admins about new Drivers

Actors: Notification Manager

The notification manager is responsible for notifying admins about new driver applications.

#### 12. Unlock Driver Features

Actors: admin

When a user's submitted paper to become a driver is approved by the admin, the driver will unlock the ability to schedule trips for other users to book.

#### 13. Notify Users of Approved or Decline Requests

Actors: Admin, Notification Manager

When a user's request to become a driver is examined by admin and approved or declined, the user will get a notification back telling him about the decision and the reasons behind refusal in case of one.

#### 14. Add Payment method

Actors: Passenger, Driver

The User can choose his preferred payment method that is presented in the application for future payments.

#### 15. Report/ Submit Complaint

Actors: Passenger, Driver

The user can report/submit a complaint about another user if he has recently been on a trip with them.

#### 16. Notify About Upcoming Trips

Actors: Notification Manager, Passenger, Driver

The notification manager is responsible for reminding a passenger/driver about their upcoming trips before a reasonable amount of time.

17. Show Scheduled Trips

Actors: Driver, passenger

Drivers and passengers can see a page with their upcoming trips sorted by the time they are scheduled to start.

18. Manage Account Settings

Actors: Driver, passenger

Driver and passengers can access their account settings through the app to manage settings like changing their email, password, and payment method.

19. Search Trips

Actors: passenger, Driver, Guest

Drivers, passengers, and application guests can search for trips by entering the departure, destination, and time of the trip he is looking for.

20. View available trips

Actors: Driver, passenger, guest

Drivers, passengers, and application guests can view available trips after searching for them.

21. Apply Trip Filter/ Sorting if Any

Actors: Driver, passenger, guest

Driver, passengers, and guests can filter their trip search results based on different criteria, such as driver rating, rules they would like the trip to abide by, or the verified status of the drivers.

22. Show Chosen Trip Details

Actors: Driver, Passenger, Guest

Users searching for a trip can see more details about available trips, like driver information and rating, driver's rules, names of other passengers on the trip and their ratings, and access to the driver profile from there.

23. Schedule Trips

Actors: Driver, Map Service

The driver can schedule a trip by adding departure, destination, and time of the trip, preferred routes, and additional trip rules.

24. Set trip route

Actors: Driver, Map Service

The driver can select the trip route using predefined stations allowing passengers to be picked up along the way. If the driver chooses not to select the routes, their trip will show up only to passengers coming from the same starting destination and going to the same final destination.

25. Suggest New Stations

Actors: Driver, Map Service

The driver can suggest a new station to be added to the set of stations throughout a certain route.

26. Validate Trip

Actors: Driver

The system checks if the scheduled trip by the driver doesn't conflict with other trips they already set up.

27. Request to Join Trip

Actors: Driver, passenger

Driver and passenger can send a request to the trip driver to book a seat on the trip.

28. Validate Request

Actors: passenger

A user's request to join a trip is validated so that it does not conflict with other scheduled Trips.

29. Review Join Trip Request

Actors: Driver

The driver can see requests from passengers wanting to ride with them and accept or decline depending on the ratings of the passenger, and other factors such as the verified account status of the passenger and their location.

30. Notify Passenger of Driver Response to Trip Request

Actors: Driver, Notification Manager

When a driver accepts or declines a passenger's request to be picked up, the system notifies the passenger and enables them to proceed to payment in case of approval.

31. Send Payment to Driver for Trip

Actors: payment service, Admin

At the end of trip confirmation, The system processes the payment to the driver's account.

32. Receive Payment

Actors: payment Service, Admin

When a payment is made by a passenger the payment service transfers the money into a median pocket owned by the application admins which holds the payment until trip completion.

33. Confirm Trip Completion

Actors: Driver, passenger

At the end of the trip:

The driver should ask and ensure that the passenger confirms that his trip has ended.

The driver should confirm that the trip has ended when he reaches his destination.

34. Cancel a Reservation

Actors: passenger

passengers are allowed to cancel a trip reservation within a limited amount of time estimated xx hours, with a full refund, and a partial refund accounted for the remaining time of the trip after that.

35. View Accepted Trips

Actors: Passenger

Passengers can view the trips that they have been accepted to and decide which one they are going to attend.

36. Perform Payment

Actors: Passenger, Payment Service

Passenger can pay for the trip they have chosen to attend and their seat is booked on successful payment. Additionally, all previous requests sent to other drivers will be canceled.

### 37. Validate Payment

Actors: Payment Service

The payment service is responsible for validating the payment and transferring the money from the passenger to the admin's account.

### 38. Cancel a Reservation

Actors: Passenger

The passenger is allowed to cancel an already booked trip or a trip request. If the trip is paid for, a refund is issued back to the passenger's payment method. The refunded amount will be deferred depending on how late the passenger requests to cancel.

### 39. Ask for Rating/ Feedback

Actors: Passenger, Driver

On the trip's end, both passengers and drivers are able to rate each other and provide feedback if they wish.

### 40. View trip history

Actors: Passenger, Driver

Passengers and drivers can see their trip history from their profile section, and if a user is registered as a driver he can choose to display trips he made or trips he has been part of.

## Use Case Diagram

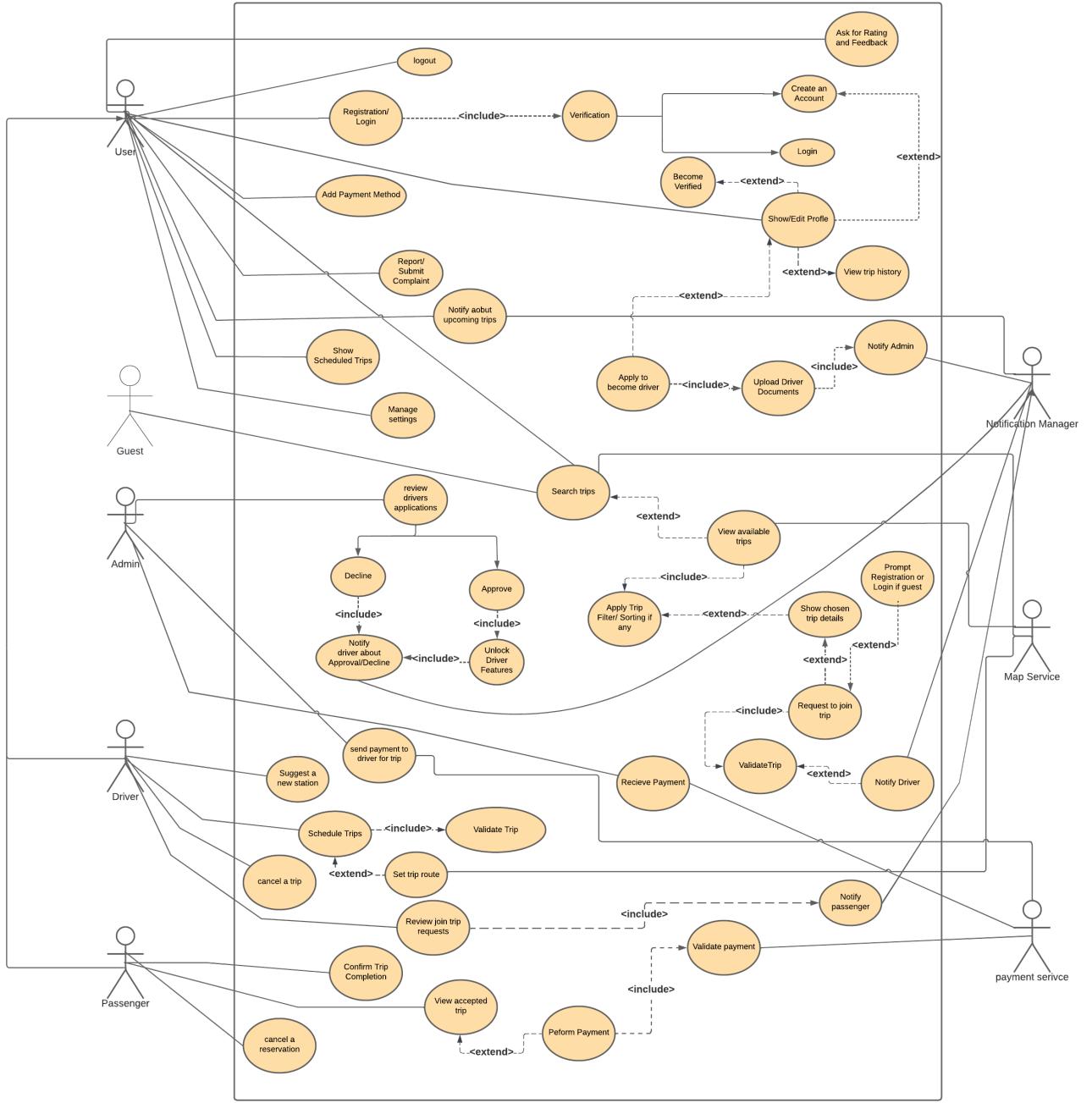


Figure 4.1: Use Case Diagram

## 4.4 Sequence, UML, State, and Deployment Diagrams

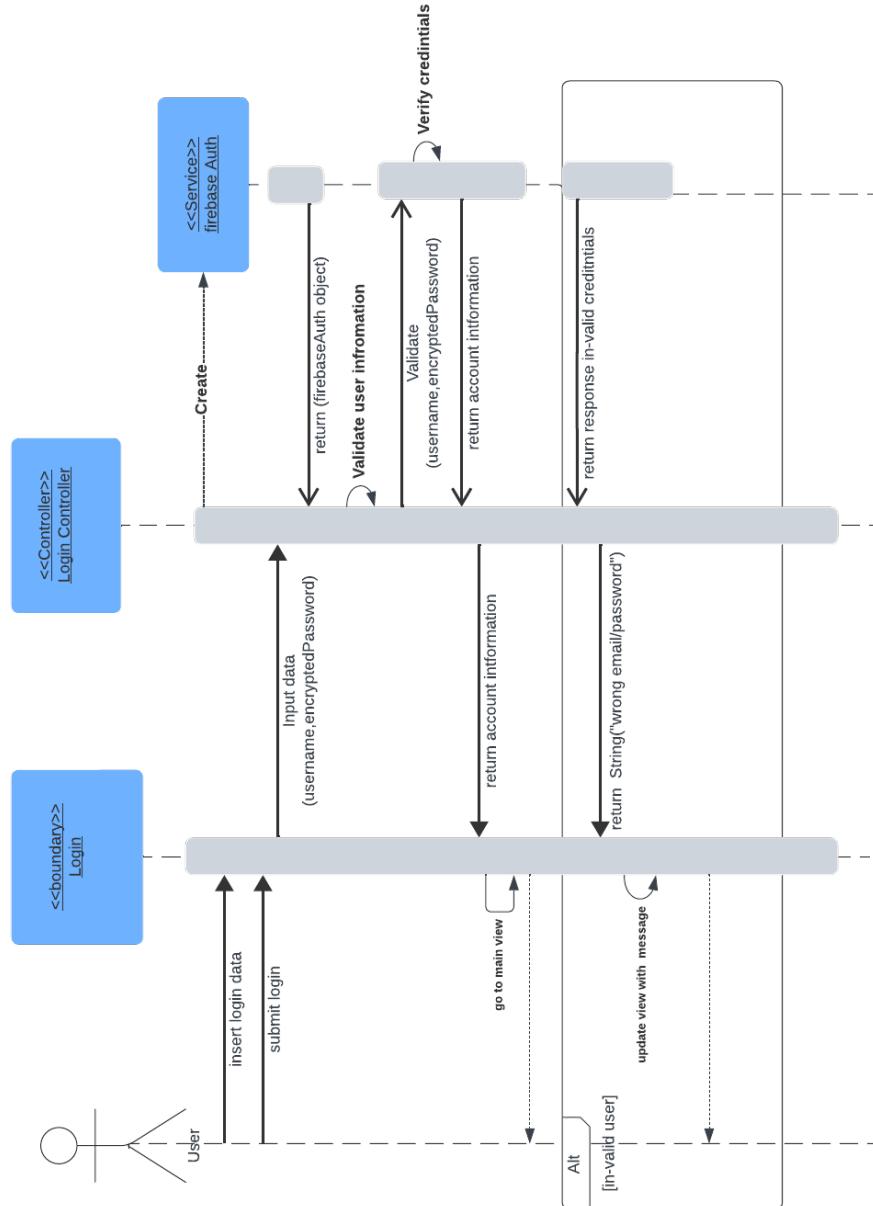


Figure 4.2: Login Workflow

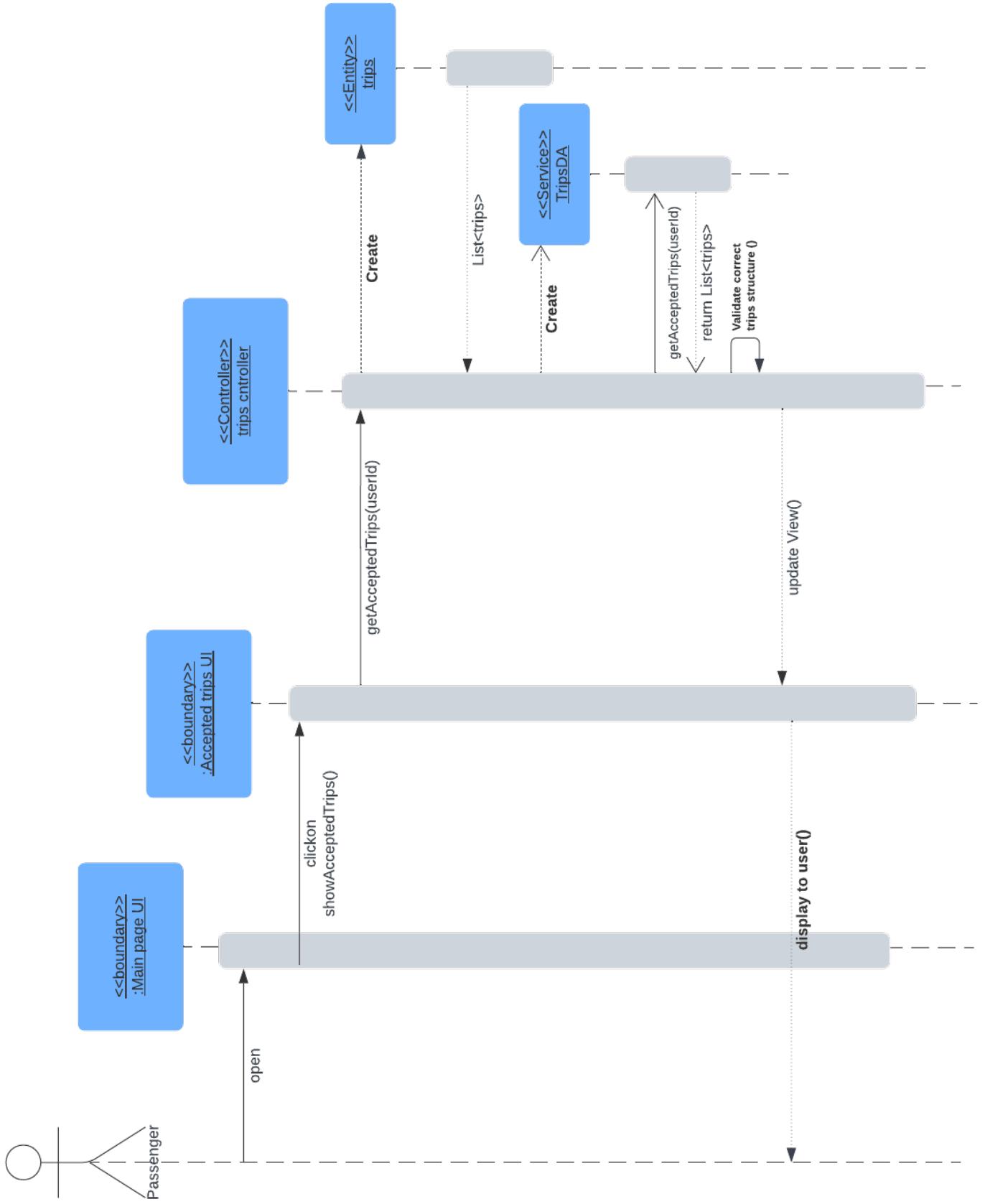


Figure 4.3: Showing Passenger's Accepted Trips Action

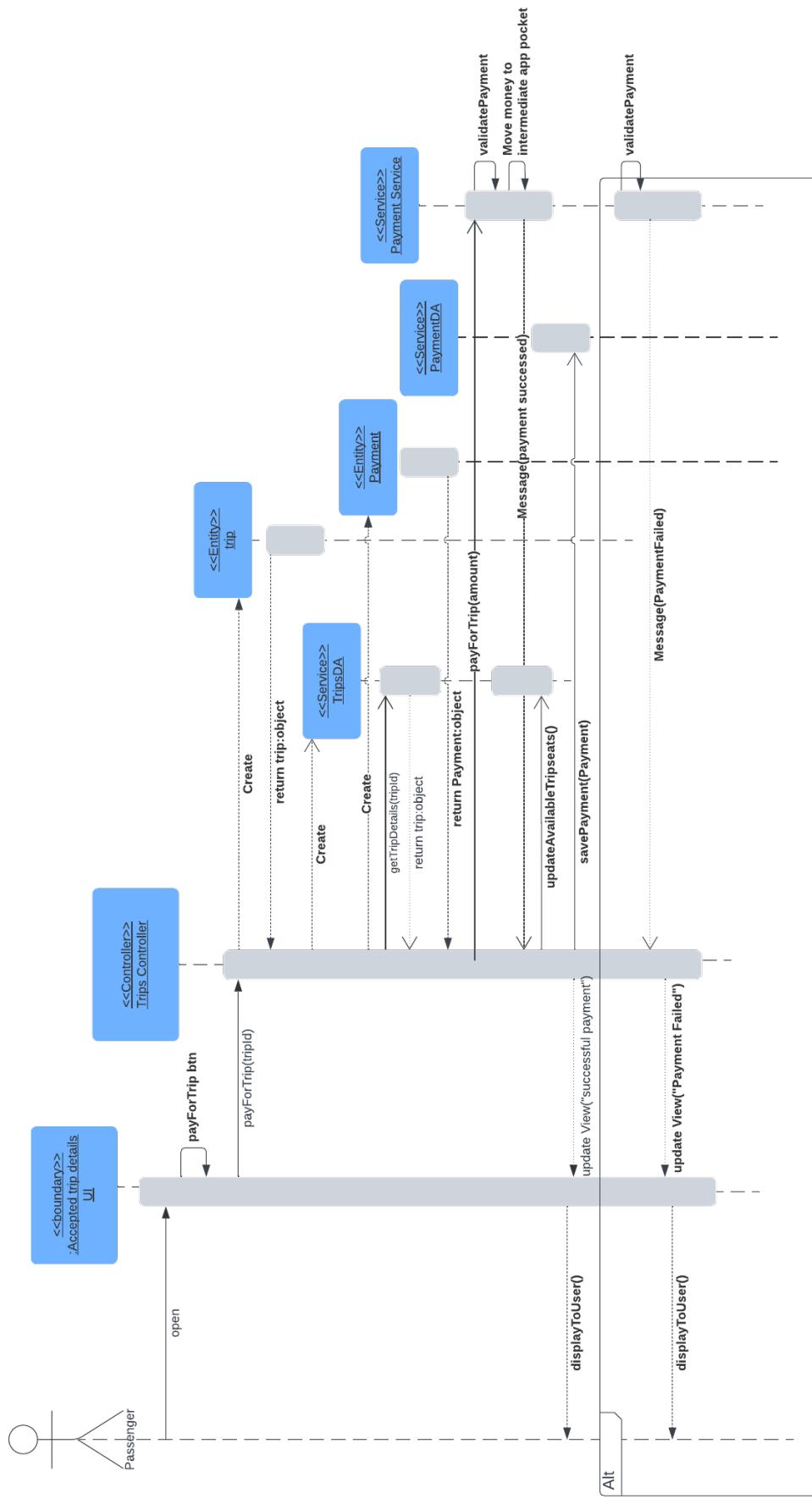


Figure 4.4: Passenger Payment Workflow

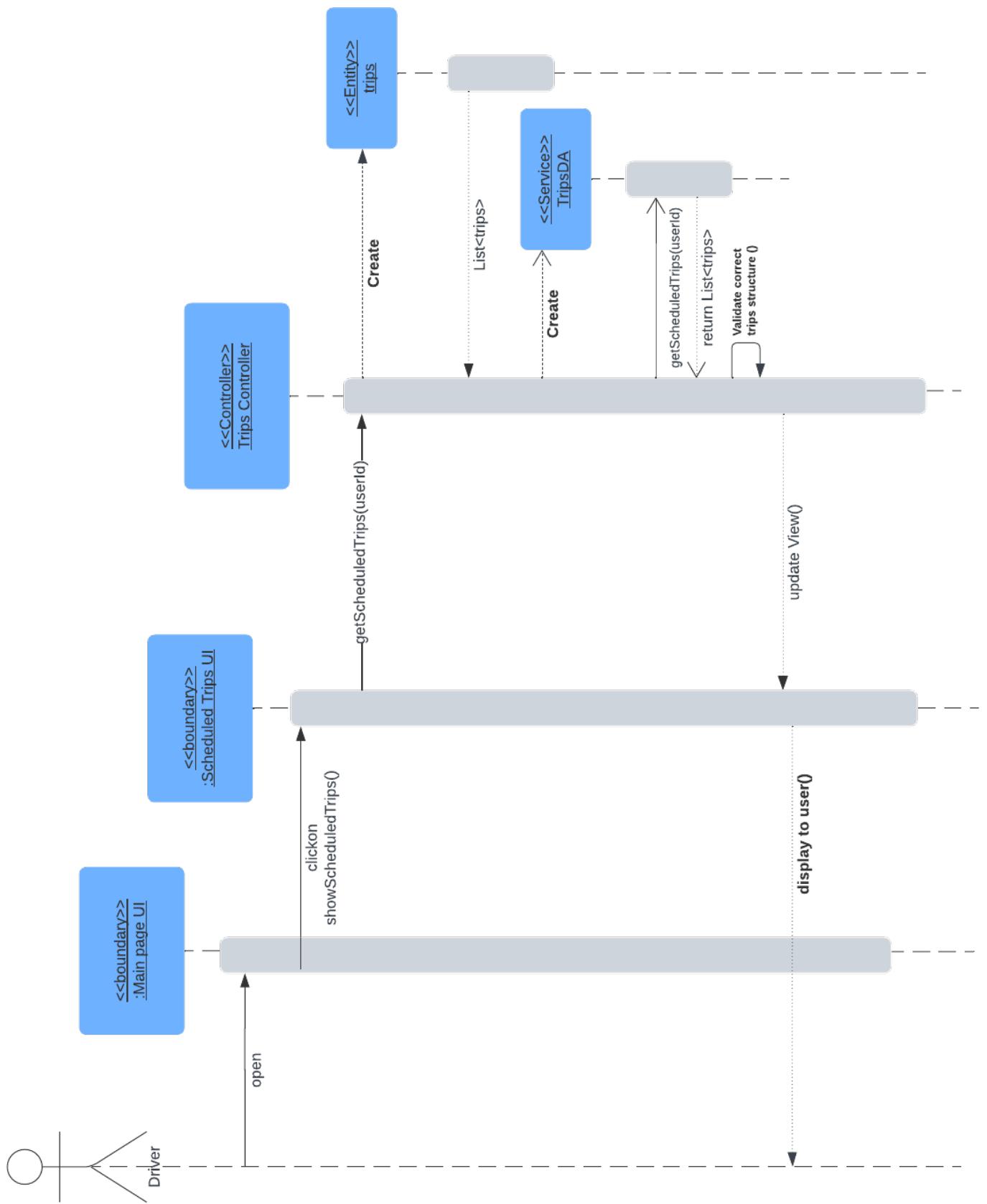


Figure 4.5: Showing Driver's Scheduled Trips Action

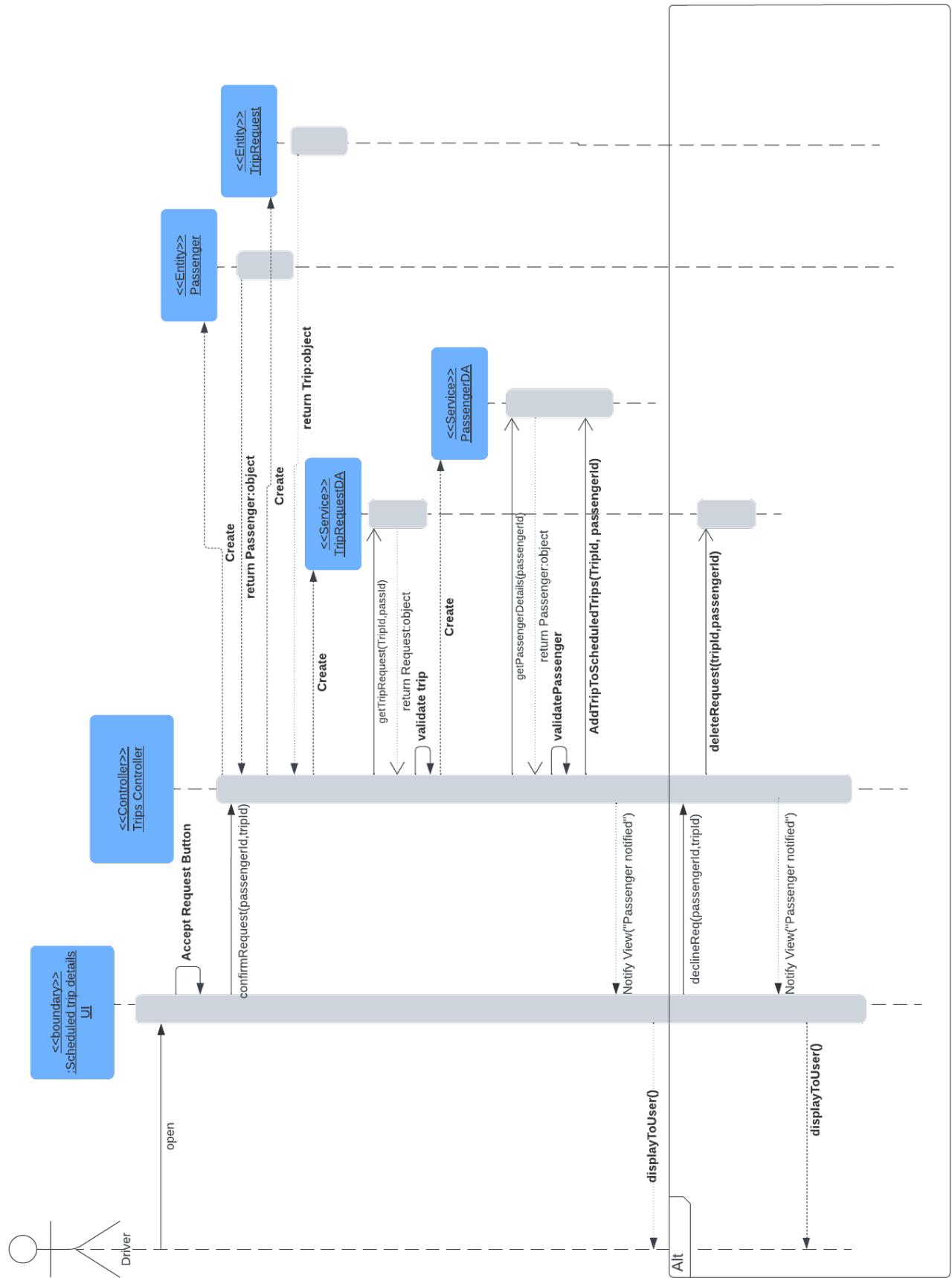


Figure 4.6: Passenger Request Workflow

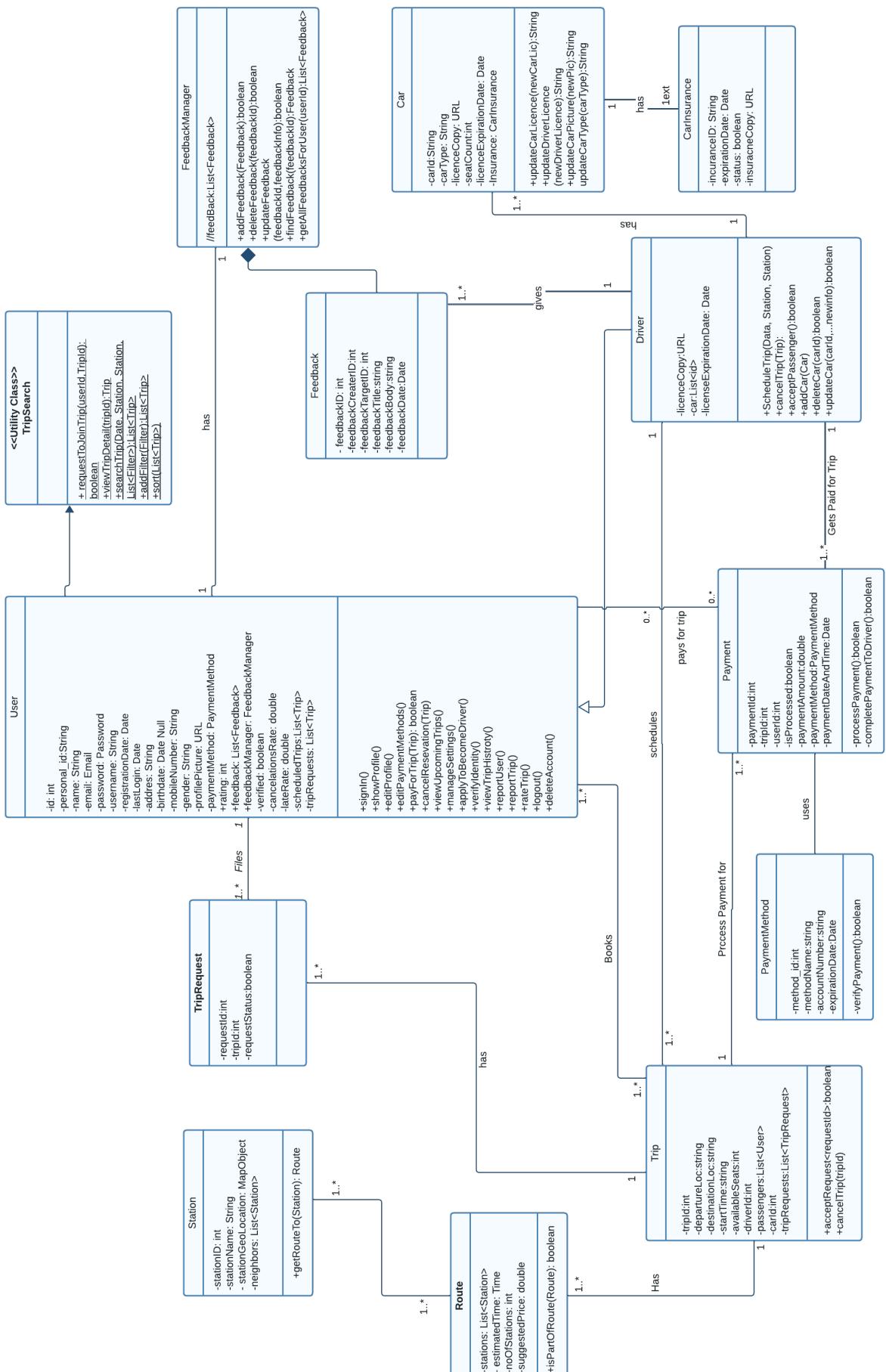


Figure 4.7: UML Class Diagram

### State diagram passenger view

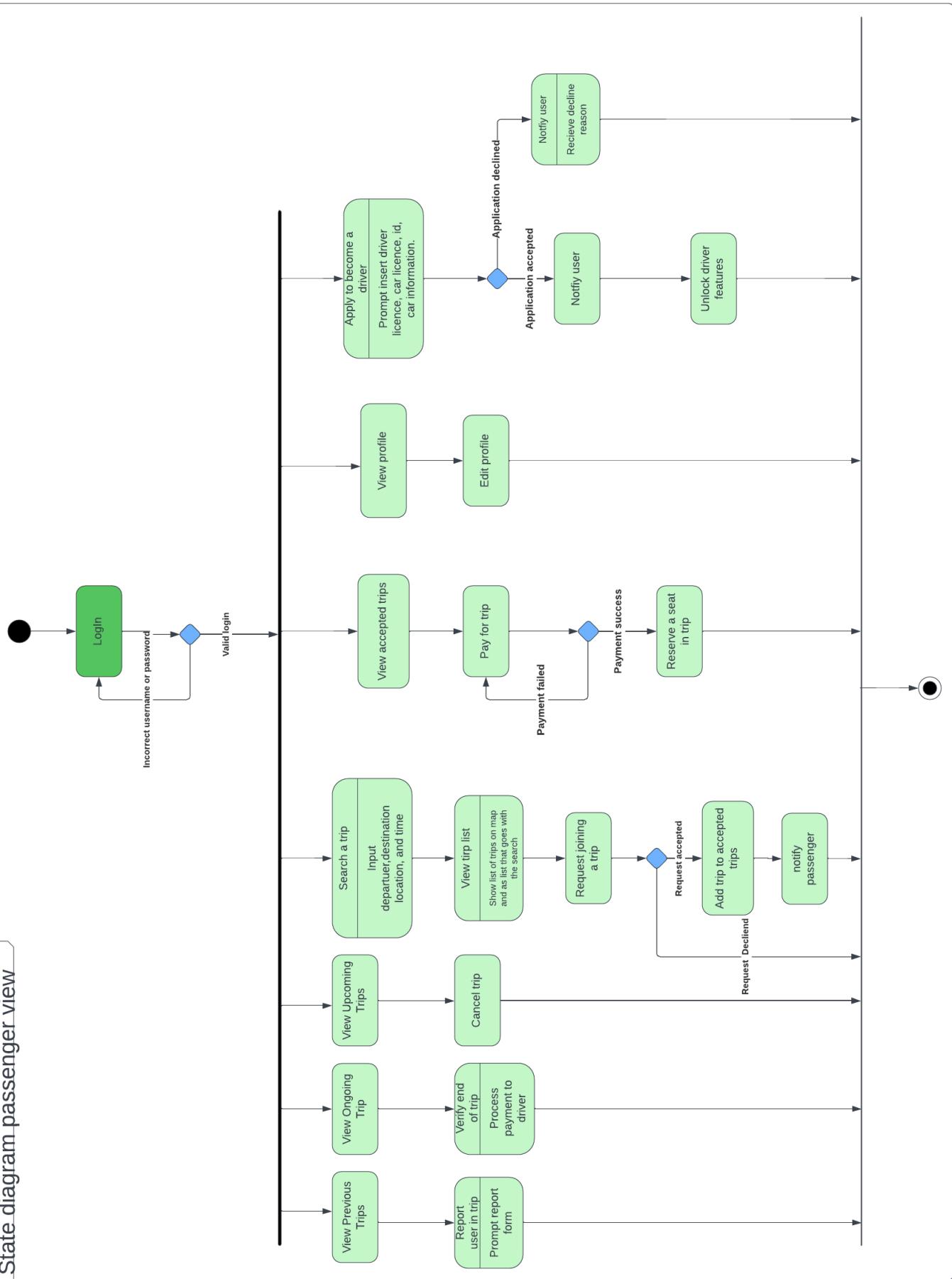


Figure 4.8: State Diagram for Passenger

## State Diagram for Driver

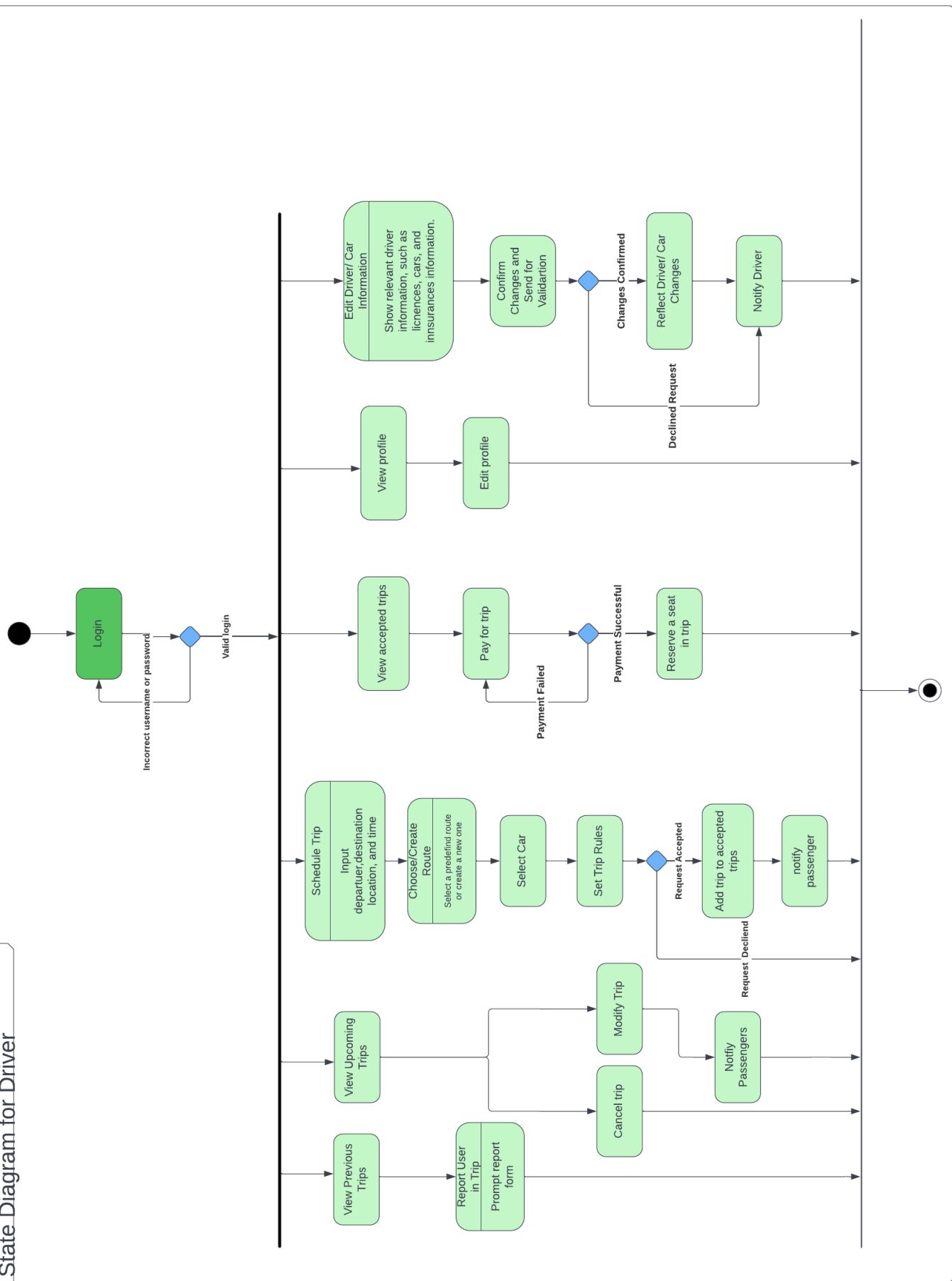


Figure 4.9: State Diagram for Driver

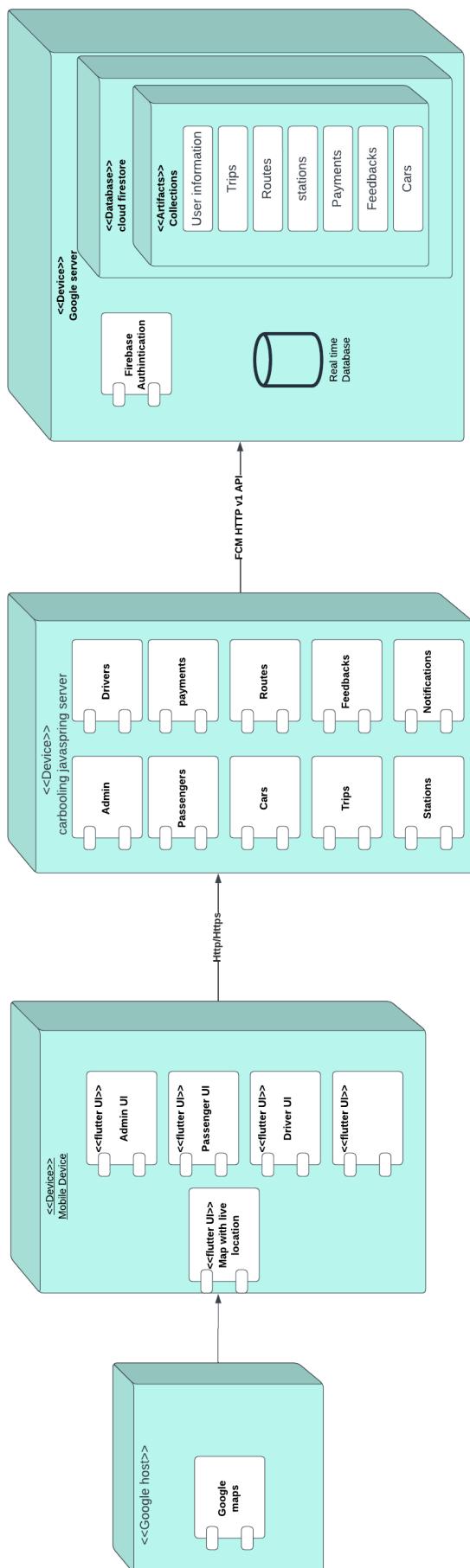


Figure 4.10: Deployment Diagram

## 5 Initial Prototype

This is an initial prototype. The final app look is subject to change.



(a) Login Screen



(b) Passenger Search Screen



(c) Driver Trip Creation Screen



(d) Passenger Search Result Screen

Figure 5.1: CarPal App Initial Prototype<sup>1</sup>

<sup>1</sup>Made using figma.com. Icons credit: flaticon.com

## **6 Conclusion and Future Work**

### **6.1 Conclusion**

This project aims to develop a ridesharing app tailored to work with the roads in Palestine, where thousands of people rely on some form of transportation to commute every day. Given that conventional transportation methods suffer from problems regarding time, availability, and comfort, our app comes in to provide people to give people another option that tackles these issues. CarPal Ridesharing App allows anyone with a driver's license and a car to pick other people up along the way anywhere and at any time. It allows people to cut the costs of their trip, get there in fewer commutes, and save them a lot of waiting time, all while being affordable and easily accessible through a mobile app. The app will let people create accounts, become verified drivers who can start scheduling trips, or look for rides with others as passengers. Passengers will be able to book and pay through the app. At the end of the trip, both the passengers and the drivers can provide feedback on the trip.

### **6.2 Future Work**

Since the workflow of the project is incremental, we will start by working on the core features that our app depends on first. The core features of our app mostly reside in the back-end API and the database, i.e. routes and user database, the trip scheduling and booking transactions, and map services API. We will work on the UI and UX for each feature simultaneously. We will then move on to work on user accounts, profiles, social features, and payment methods. The final stage will be to test the app, add more routes, and deploy a version of the app to the public or a specific party for beta testing. See table 6.1.

Table 6.1: Work plan for the tasks and the initial estimated time needed to achieve them.

Feature	Time
Identify the priority of each functional requirement and create sprints.	1 week
Setting up the routes and users database with test cases for both.	2 weeks.
Working on the trip scheduling logic and UI (driver logic).	2 weeks
Working on trip booking logic and UI (passenger logic).	2 weeks
Integrate map and location services.	2 weeks
Add user profiles, user rules, trip filters, feedback, and rating features.	4 weeks
Set up payment environment.	2 weeks
Polishing and fine-tuning period.	1 week
Deploy the app for testing and collecting feedback.	1 week
Total	17 weeks

# **Part II**

## **Second Semester**

# **1 Introduction**

This part of the project is the continuation of the first part, which was dedicated to the analysis and design of the project, while this part is dedicated to the implementation and testing of the project. The project is a ridesharing app that allows people signed up as drivers to pick up passengers along the way with them to a mutual destination.

# **2 Work Plan**

We have refined the work plan found at the end of the last part in table 6.1 to better reflect the actual work needed to be done. The new work plan clearly outlines the different stages need to be done to complete the project. Figure 2.1 shows the new work plan. The new work plan starts with setting up the Firebase database with the collected sample routes, then moves to work on the trip scheduling logic and UI, then the trip booking logic and UI, then integrating the map and location services, then adding user profiles, user rules, trip filters, feedback, and rating features, then setting up the payment environment, then a polishing and fine-tuning phase. Please note we have done the first 3 stages and we are working on stage 4 and 5 concurrently.

# **3 Changes to Part 1 Requirements**

As we started working on the application, we found that Firebase has a lot of features that cover the functionalities of a designated backend. Therefore, for the time being, there is no need to create a Spring Boot API for our application. Firebase has a built in authentication system with password reset and email and phone verification. The database has filtering queries backend that enables us to send specialized requests to get, update, or create specific data. There are no other changes are currently for the rest of the requirements in part 1.

## Implementation Timeline

Implementation timeline for the project

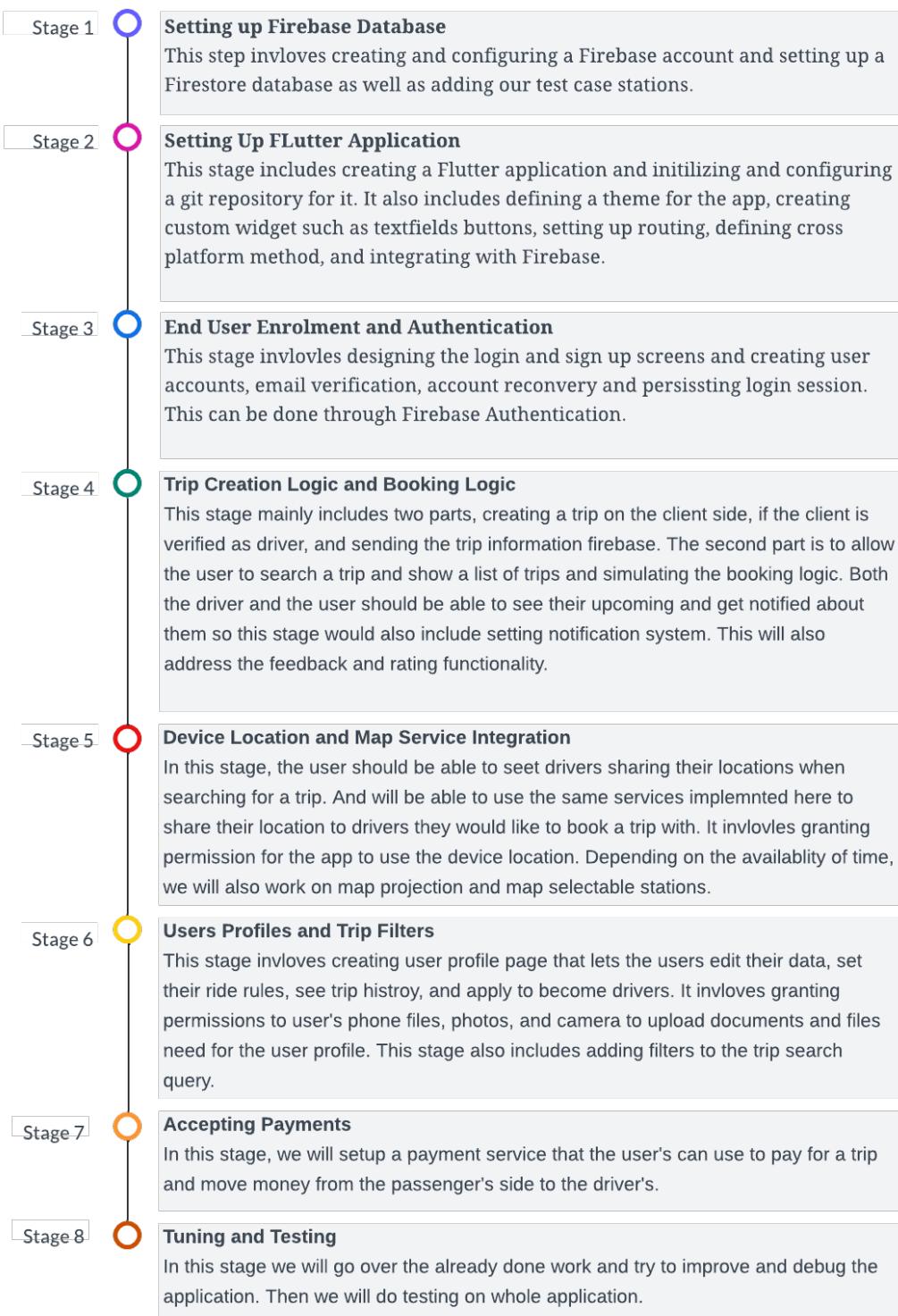


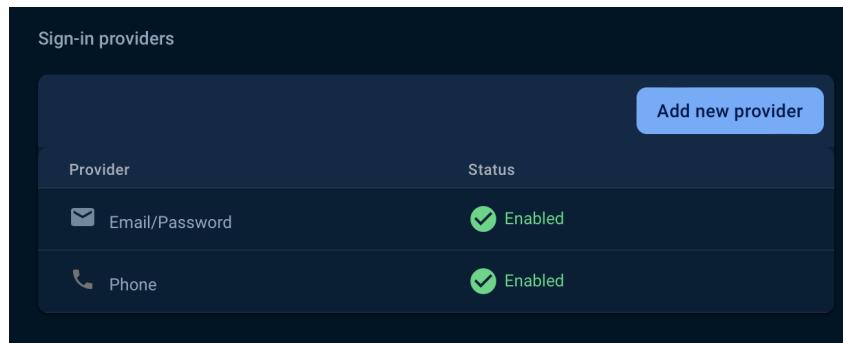
Figure 2.1: Refined Work Plan

## 4 Implementation

This section outlines some of the work we have done across stage 1 - 5. It includes screenshot from the app running on both Android and iOS presenting the user interface and the features implemented.

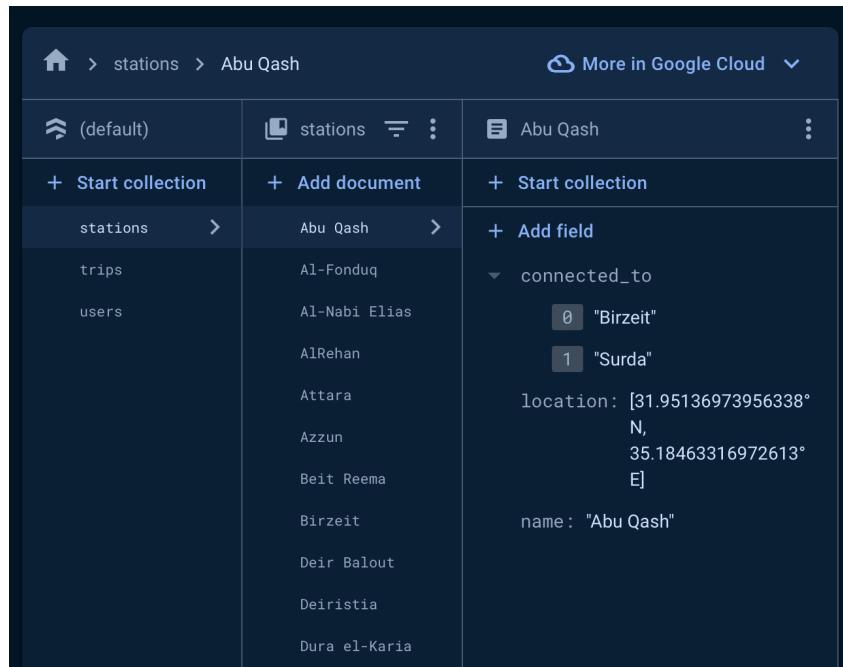
### 4.1 Firebase Setup

We have set up the Firebase project and integrated it with the app. We have set up the Firebase Authentication, Storage, and FireStore Database. Figure 4.1 shows the Firebase console with the Authentication methods enabled for our and some of the documents stored in the FireStore Database.



The screenshot shows the 'Sign-in providers' section of the Firebase console. It lists two providers: 'Email/Password' and 'Phone', both of which are marked as 'Enabled' with green checkmarks. There is a blue button at the top right labeled 'Add new provider'.

(a) Firebase Authentication



The screenshot shows the 'stations' collection for the document 'Abu Qash' in the Firebase FireStore Database. The collection contains several documents: 'Al-Fondouq', 'Al-Nabi Elias', 'AlRehan', 'Attara', 'Azzun', 'Beit Reema', 'Birzeit', 'Deir Balout', 'Deiristia', and 'Dura el-Karia'. One document, 'Birzeit', has its 'location' field expanded, showing coordinates [31.95136973956338° N, 35.18463316972613° E] and its 'name' field set to "Abu Qash".

(b) Firebase FireStore Database

Figure 4.1: Firebase Console

## 4.2 User Sign up and Login

We have implemented the user sign up and login functionalities using Firebase Authentication. The user can currently sign up using their email. They will need to verify their email to be able to login. Email verification as well as password reset are implemented. Figure 4.2 shows the start screen, the user sign up screen and login screens. Guest accounts, and phone sign up are not implemented yet.

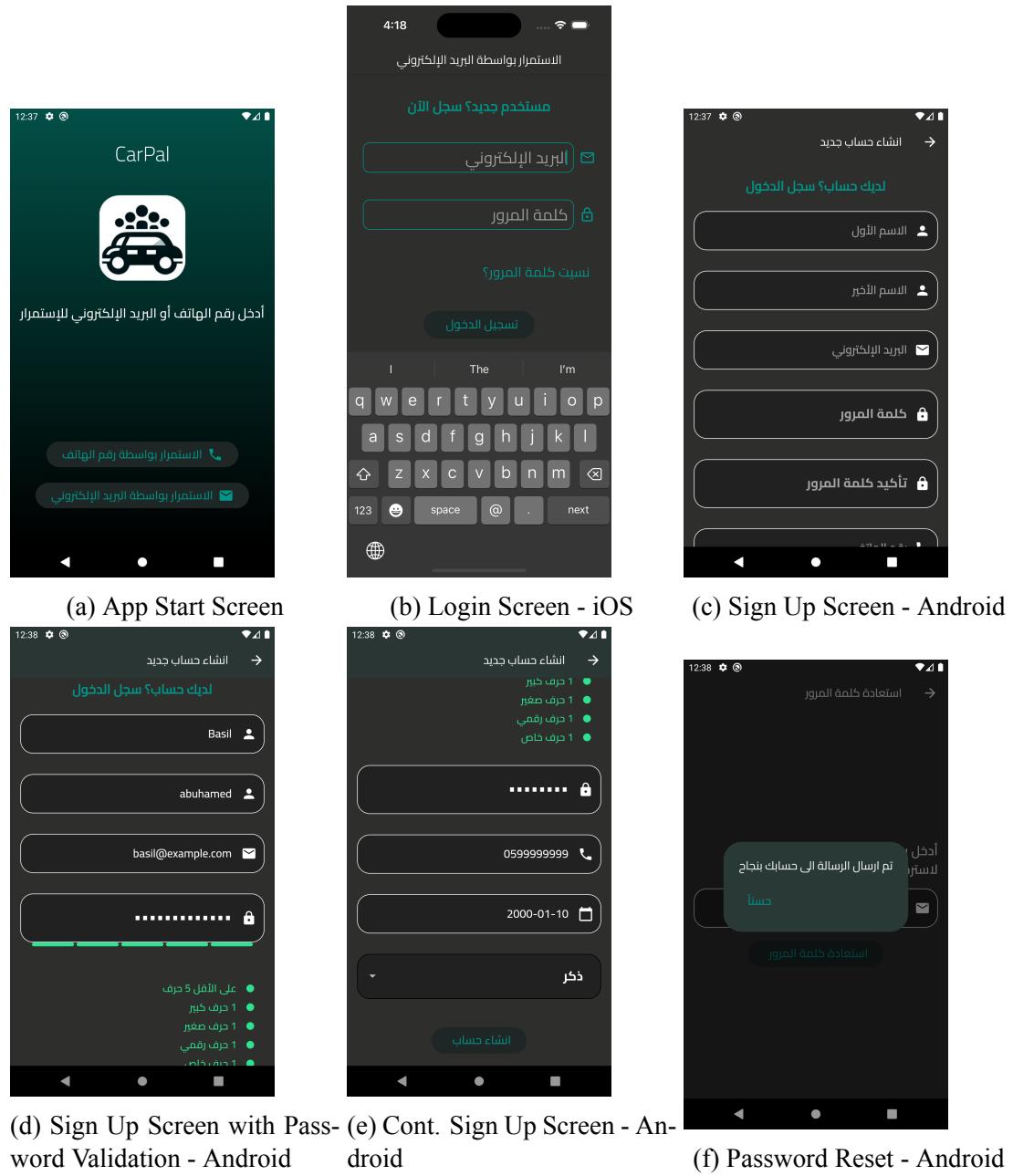


Figure 4.2: User Sign Up and Login Screens

### 4.3 Driver Trip Creation

Drivers can create trips by specifying the departure and destination locations, time of the trip, available seats, and the price of the trip. Further options such as trip rules and the stations to pass by are not implemented yet. Figure 4.3 shows the trip creation screens.

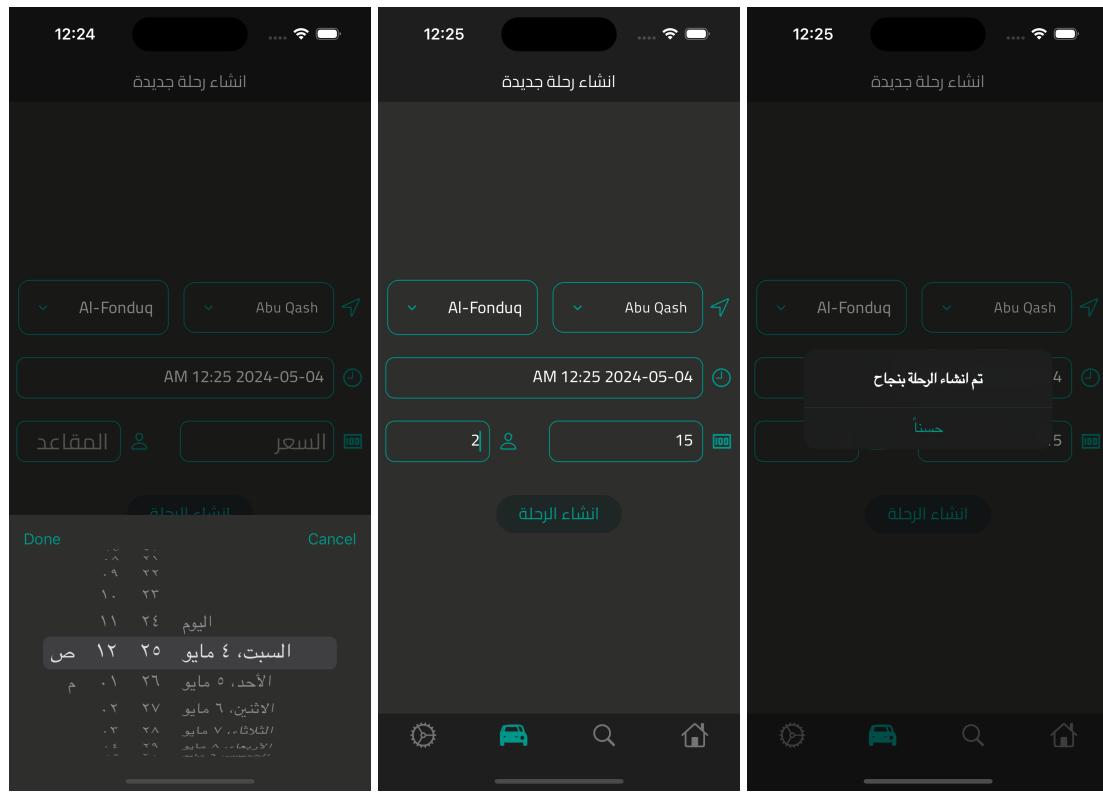


Figure 4.3: Driver Trip Creation Screens

## 4.4 Passenger Trip Search

The passengers can search for a trip using the *Trip Search Screen*. Similar to the *Trip Creation Screen* in UI and parameters, the passenger can search for the trip by entering the departure and destination locations, date of the trip, and the number of passengers that need to be accommodated. When a search query is submitted, the app will open a new screen where the passenger can see the trips available and a map of the last location the driver was seen in. The passenger can pull down to refresh the page and update the trip list. In addition to the information visible in the trip list which can be seen in Figure 4.5, the trip list will also include information about the current driver location such the last time the location was updated. The trip list should also be sorted by different parameters (chosen by user) such time, price, distance, filters, and ratings. Each item in the trip list is clickable. The click action will open a new screen with more details about the trip, driver, and booked passengers if any. The *Trip Detail Screen* is not yet implemented.

## 4.5 User Live Location

The app provides features to see the live location of the driver and the passenger. We have configured the app so that it asks for device location permissions on both Android and iOS. We are using Google Maps API to show the live location of the driver and the passenger on an instance of an intractable Google Maps Widget built into the app. We have established a Stream of realtime locations of a group of users stored in FireStore. 4.6 shows the live location feature in action, streaming the location data of users stored in FireStore. We already used this feature to show live location of all relevant drivers to the passenger when booking a trip. We will use this feature for the passenger to track the driver's location and for the driver to see the passenger's location after booking a trip.

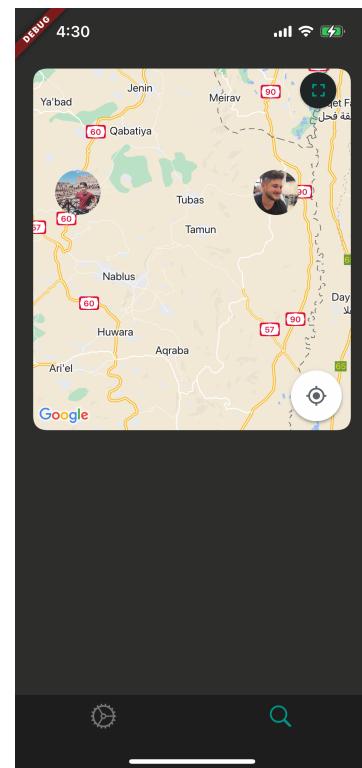


Figure 4.6: User Live Location Demo - iOS

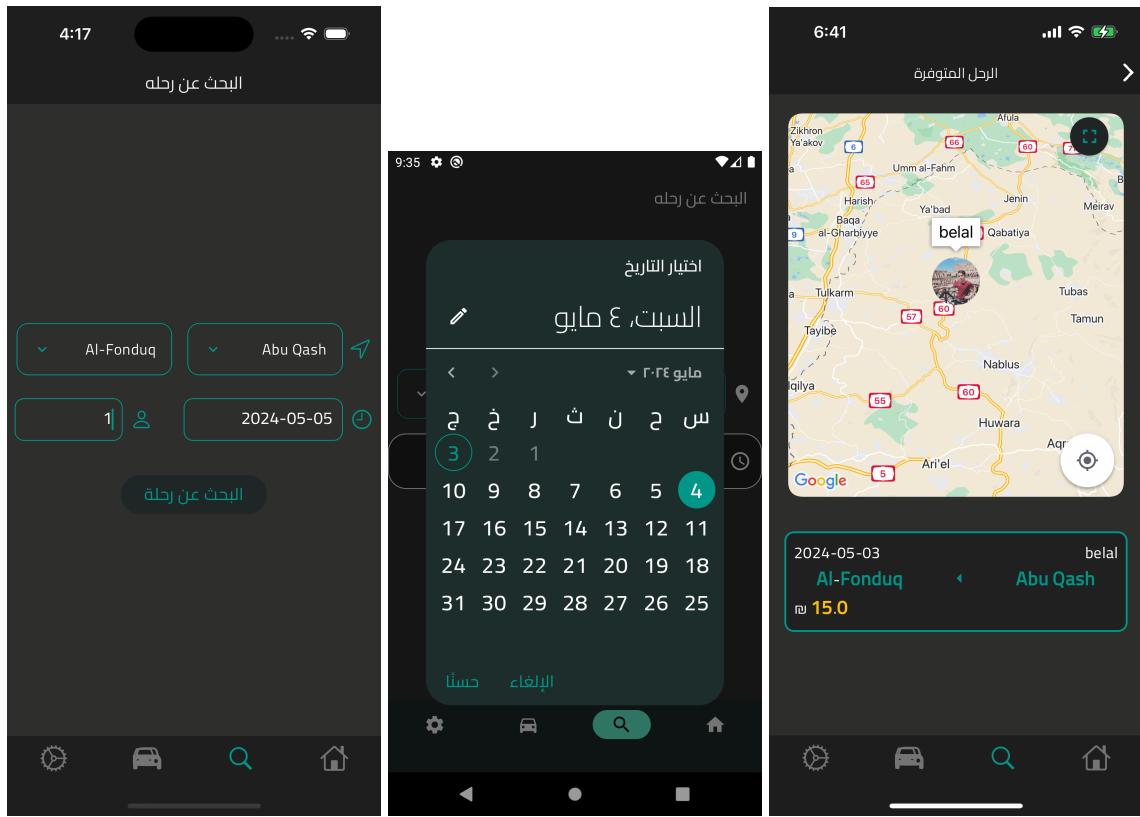
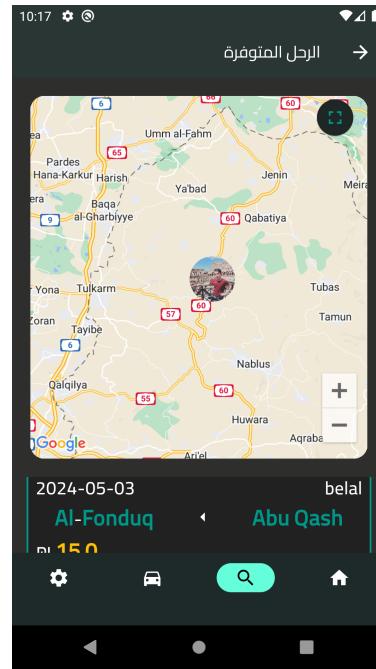


Figure 4.4: Passenger Trip Search Screens

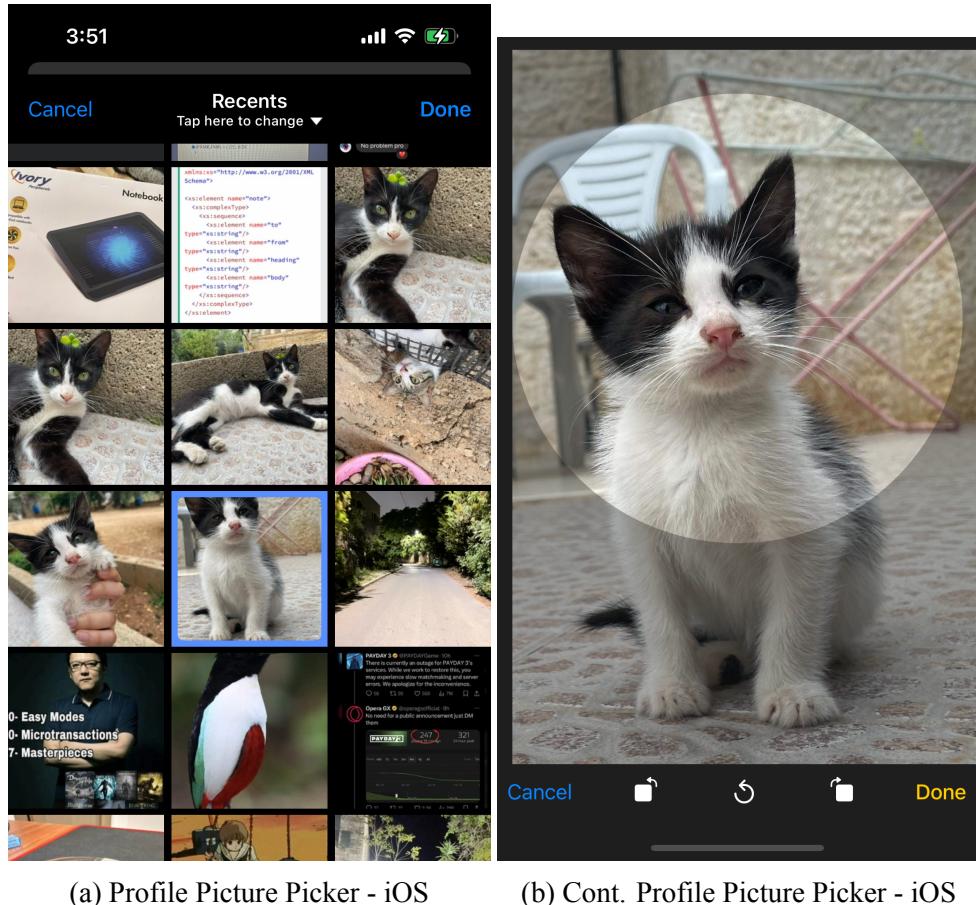


(a) Trip Search Result - Android

Figure 4.5: Passenger Trip Search Screens

## 4.6 Profile Picture Picker

We have added a functionality for users to add profile pictures using their camera or photo gallery using the Flutter library HL Image Picker [19]. After getting relevant permissions, it allows the user to take/pick photos, crop them, and then set them as their profile picture.



(a) Profile Picture Picker - iOS

(b) Cont. Profile Picture Picker - iOS

Figure 4.7: Profile Picture Picker Screens

# Bibliography

- [1] Wikimedia Foundation Wikipedia. *Firebase*. URL: <https://en.wikipedia.org/wiki/Firebase> (visited on 10/16/2023).
- [2] Google. *Firebase Realtime Database*. URL: <https://firebase.google.com/docs/database> (visited on 01/06/2024).
- [3] Firebase. *Firebase Logo*. [https://commons.wikimedia.org/wiki/File:Firebase\\_Logo.png](https://commons.wikimedia.org/wiki/File:Firebase_Logo.png). Oct. 2018.
- [4] Wikimedia Foundation Wikipedia. *Flutter (Software)*. URL: [https://en.wikipedia.org/wiki/Flutter\\_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software)) (visited on 11/14/2023).
- [5] Google. *Google Flutter Logo*. <https://commons.wikimedia.org/wiki/File:Google-flutter-logo.png>. Sept. 2018.
- [6] Wikimedia Foundation Wikipedia. *Spring Framework*. URL: [https://en.wikipedia.org/wiki/Spring\\_Framework](https://en.wikipedia.org/wiki/Spring_Framework) (visited on 10/13/2023).
- [7] Pivotal Software. *Spring Framework Logo*. [https://commons.wikimedia.org/wiki/File:Spring\\_Framework\\_Logo.png](https://commons.wikimedia.org/wiki/File:Spring_Framework_Logo.png). Dec. 2018.
- [8] Wikimedia Foundation Wikipedia. *Blablacar*. URL: <https://en.wikipedia.org/wiki/BlaBlaCar> (visited on 09/27/2023).
- [9] Blablacar App for iOS.
- [10] Wikimedia Foundation Wikipedia. *Gett*. URL: <https://en.wikipedia.org/wiki/Gett> (visited on 10/18/2023).
- [11] Bloomberg Bloomberg.Com. *Gett Inc - Company Profile and News*. URL: <https://www.bloomberg.com/profile/company/0852787D:IT> (visited on 11/15/2023).
- [12] Max Eddy PCMAG. *Gett (for iPhone) Review*. URL: <https://www.pc当地.com/reviews/gett-for-iphone> (visited on 01/10/2024).

- [13] Wikimedia Foundation Wikipedia. *InDrive*. URL: <https://en.wikipedia.org/wiki/InDrive> (visited on 11/06/2023).
- [14] inDrive App Store. *InDrive: Ride & Drive With Us*. URL: <https://apps.apple.com/ma/app/indrive-ride-drive-with-us/id780125801> (visited on 01/10/2024).
- [15] Roger F. Teal. “Carpooling: Who, How, and Why”. In: *Pergamon Journals Ltd.* (1984).
- [16] Jun Guan Neoh, Maxwell Chipulu, and Alasdair Marshall. “What encourages people to carpool? An evaluation of factors with meta-analysis”. In: *Transportation* 44.2 (Mar. 2017), pp. 423–447. URL: <https://eprints.soton.ac.uk/381789/>.
- [17] Huang Hai-Jun, Yang Hai, and G.H. Bell Michael. “The models and economics of car-pools”. In: *The Annals of Regional Science* (1997).
- [18] Kate Brush and Valerie Silverthorne. *What Is Agile Software Development (Agile Methodologies)?* URL: <https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development> (visited on 12/06/2023).
- [19] howljs. *hl image picker*. URL: [https://github.com/howljs/hl\\_image\\_picker/tree/main](https://github.com/howljs/hl_image_picker/tree/main) (visited on 05/04/2024).