LONDON METROPOLITAN UNIVERSITY

ITAHARI INTERNATIONAL COLLEGE

**Module Code & Module Title**

CS6PO5NT Final Year Project

**Assessment Weightage & Type**

25% FYP Interim Report

**Year and Semester**

2022-23 Autumn

**Student Name:** Susan Shrestha

**London Met ID:** 20048536

**College ID:** NP05CP4S210004

**Internal Supervisor:** Mr. Rabi Rouniyar

**External Supervisor:** Mr. Shekhar Timisina

**Assignment Due Date:** 2022-12-28

**Assignment Submission Date:** 2022-12-28

**Project Title:** Ride Hero - Ride Sharing Web App

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

## Table of Contents

## List of Figures

## Table of Tables

# 1. Introduction

Ride-sharing refers to the use of a digital platform to connect passengers with drivers for the purpose of transporting passengers from one location to another. Ride-sharing services have become increasingly popular in recent years as a convenient and cost-effective alternative to traditional taxi services. A ride-sharing web app is a software application that allows users to request and pay for rides through their internet browser. These apps typically use GPS technology to match passengers with nearby drivers and to track the progress of the ride. Passengers can usually rate their experience and leave reviews for drivers. This feedback system helps to ensure and maintain the overall quality of the service.

Ride-sharing web apps offer a number of benefits to both passengers and drivers. For passengers, they provide an easy way to get around without the need to own a vehicle or deal with the inconvenience of public transportation. For drivers, they offer the opportunity to earn money by providing rides to people in their community. Overall, ride-sharing web apps have revolutionized the way people think about transportation and have made it easier for people to get around in urban areas.

This project is a web-based app through which is a popular alternative to traditional modes of transportations, such as taxis and public transportation. This web app provides on-demand ride-sharing services through several modes of transportation such as cars, bikes and tuk-tuks. This web app provides services to the users and drivers as well. This web app uses GPS technology to interconnect passengers and drivers. Initially, the passenger must register their personal information as requested by the app, and then they must login to the app to request a ride. A passenger can enter their pick-up and drop-off locations and can request for a ride. A passenger can also bargain over the fare of a ride. If a driver accepts the fare offered by the passenger then the server will send the message of confirmation to the passenger. This web application manages user bookings in the fastest and easiest way possible. With a single click, passenger can request or cancel a vehicle for a ride and a driver can accept or cancel passengers request for a ride with a single click. The admin has entire authority over this web application. The admin authenticates the passenger's and driver's personal information as requested by the application. The admin has the authority to add, delete, and update passenger and driver information. When registering a driver for the application, the admin verifies to see if the driver has a valid driver's license. A driver can’t work as a driver in the application if he/she doesn’t have a valid driver's license. Payment transactions in this web application is digitalized. All the payment transactions are monitored by the admin.

## 1.1. Problem Scenario

The following are the problem scenarios:

- Limited transportation alternatives, such as limited public transportation routes or a lack of taxi services, can make it difficult for people to travel around and access necessary services.
- Transportation can be costly which can be burden for who need to travel frequently or have a limited income.
- Transportation may not be accessible to everyone, especially people with disabilities or mobility challenges, making it difficult for them to travel around and obtain important services.
- Public transportation may experience, congestion, particularly during peak hours, which can lead to delays and reduced reliability.

## 1.2. Project as a Solution

This project is considered as problem-solving web app because this web app address a variety of problems  related to transportation, convenience, cost, the environment, and income generation.

This project can solve a number of problems, including:

- **Transportation:** This project can offer an alternate means of transportation for individuals who do not own a car or choose not to drive. This is especially useful in urban areas where public transportation may be limited or inconvenient.
- **Convenience:** This project make it simpler for people to get around because they can request rides and pay for them. This eliminates the need to hail a taxi or pre-arrange a ride.
- **Cost-effectiveness:** This project can be more cost-effective than other forms of transportation, especially for longer distances or for multiple passengers. This can make it more affordable for people to get around, especially in areas where taxi fares are high.
- **Income generation:** This project can enable people to supplement their income by driving for the app. This is particularly appealing to people who own a car and are seeking for a flexible side career.
- **Accessibility:** This project can make transportation more accessible to persons who may have trouble accessing traditional modes of transportation, such as those with disabilities or mobility issues.

## 1.3. Aims and Objectives

This project is developed to promote sustainable transportation service and mitigate the problems generated from public and private transportations by developing ride sharing web app which helps the passenger to request a vehicle for a ride with just a single click. The main aim of this project is to provide a convenient, affordable and safe ride for people to get from one place to another. This web app generally works by connecting passengers with drivers who are willing to provide a ride. Passengers can request and pay for their ride, allowing them to track their driver in real-time, while at the same time the driver can see the exact location of the passenger and have access to the payment once the ride is complete.

In order to achieve this aim, the following objectives have been set up which are listed below:

**Objectives**

- To conduct an extensive and comprehensive analysis of the major terms and resources required for the development of the web app.
- To develop a complete web application.
- To create proper format for the project documentation.
- To build a web application using the agile methodology's development process.
- To detect various items required to create wireframes and GUI.
- To facilitate convenient way to request a vehicle ride for the users.
- To determine whether the web app is impactful for the users or not.
- To develop a better understanding of web platforms.
- To learn how to use various tools necessary for the development of the project.
- To record the precise details of this web app's development and execution.

## 1.4. Web Application Features

1. **Admin**
    - Login to the app using their email address and password.
    - Set up and manage their password if necessary.
    - View and manage all users (passengers and drivers) and their profiles, including personal information, ride history, ratings and reviews, and any other relevant details.
    - Set policies and rules for drivers and verify the driver based on their driver KYC details which includes driving license as a necessary credentials.
    - Monitor and analyze ride data.
    - Manage payment processing and billing for rides.
    - Set pricing and pricing policies for rides, including dynamic pricing based on factors such as demand, time of day, and distance travelled.
    - View and manage driver ratings and reviews to ensure that drivers are meeting the required standards of service and safety.
    - Log out of the app when they are finished using it to protect their account and personal information and prevent unauthorized access.

2. **Passenger**
    - Login to the app using their email address and password.
    - Register for a new account by providing their personal information, such as name, and email address.
    - Set up and manage their password if necessary.
    - Create and manage their personal profile.
    - Request rides by specifying a pickup and drop-off location, and view quotes and estimated arrival times for available drivers.
    - View a map with real-time location tracking of their assigned driver and the estimated time of arrival at their destination.
    - Rate and review drivers after a ride to provide feedback and help other passengers make informed decisions.

- View and manage their ride history.
- Accept or cancel a ride request if needed.
- Set up and manage digital payment options.
- View and manage their preferences or special requests, such as preferred vehicle types or accessibility options.
- Log out of the app when they are finished using it to protect their account and personal information and prevent unauthorized access.

3. **Driver**
   - Login to the app using their email address and password
   - Register for a new account by providing their personal information and vehicle details.
   - Set up and manage their password if necessary.
   - Create and manage their personal profile.
   - View a map with real-time updates on ride requests and their current location.
   - Accept or decline ride requests.
   - View and update their availability status to let passengers know when they are available to pick up a ride.
   - View and update their earnings and payout information, including a summary of their total earnings.
   - View and manage their ride history.
   - Log out of the app when they are finished using it to protect their account and personal information and prevent unauthorized access.

# 2. Background

## 2.1. Technology used

This section of the report describes the overall analysis of the technologies used or will be using for the development of the **Ride Hero - Ride Sharing Web App**.

### 2.1.1. Web Browser

i. **Google Chrome:** Google Chrome is a free web browser created by Google that can be used to browse web pages on the internet. It is also a cross-platform browser, which means that various versions of the browser work on different computers, mobile devices, and operating systems. It is fast, secure, and easy to use (Moreau, 2022).

ii. **Microsoft Edge:** Microsoft Edge is the official name for the new and updated Web browser that was released in Microsoft's Windows 10 operating system to replace the ancient Internet Explorer Web browser. It shares several functionalities and menu options with Chrome while having a distinct overall design and structure. It provides more security features (Stroud, 2021).

### 2.1.2. Server Side Scripting Language – PHP

PHP is a server-side programming language used to create static, dynamic, and web-based applications. PHP is an abbreviation for Hypertext Preprocessor, which was previously known as Personal Home Pages. PHP code can be integrated into HTML code or used in conjunction with a wide range of web template systems, content management systems, and online frameworks (Jackson, 2022).

### 2.1.3. IDE – Visual Studio Code

Microsoft's Visual Studio Code (often known as VS Code) is a free, open-source text editor. VS Code is compatible with Windows, Linux, and macOS. Although the editor is lightweight, it contains several powerful capabilities that have helped VS Code become one of the most popular development environment tools in recent years (Mustafeez, 2022).

### 2.1.4. Web Design

i. **HTML:** HTML stands for Hyper Text Markup Language. HTML is a markup language that is used to create appealing web pages that seem nice on a web browser with the help of styling. An HTML document is comprised of several HTML tags, each of which includes unique content. It is simple to understand and modify (javatpoint, 2022).

ii. **CSS:** CSS stands for Cascading Style Sheet. CSS is simple to comprehend and understand, but it offers users a great deal of control over how an HTML document appear. CSS is most typically used in conjunction with the markup languages HTML or XHTML. CSS allows developers to isolate content from visual components, giving them more website flexibility and control (techopedia, 2018).

iii. **JavaScript:** JavaScript is a text-based programming language that can be used on both the client and server sides to render web pages dynamically. It is a lightweight object-oriented programming language. JavaScript is a translated language, not a compiled language. The JavaScript Translator is in responsible of interpreting JavaScript code for web browsers (javatpoint, 2022).

iv. **React JS:** React JS is a JavaScript library used to create user interfaces. React JS is an extremely powerful library that allows for efficient front-end development by dividing the page into several building elements known as components. The Model View Controller (MVC) architecture is used by React JS, and the view layer is in charge of dealing with mobile and web apps (tutorialspoint, 2022).

### 2.1.5. Framework – Laravel

Laravel is a PHP framework for web applications with powerful, attractive syntax. Laravel aims to make development easier by simplifying typical tasks seen in the majority of online projects, such as authentication, routing, sessions, and caching. Laravel promises to make the development process more enjoyable for developers while maintaining application functionality (Laravel, 2022).

### 2.1.6. Web Server – XAMPP

XAMPP is a popular cross-platform web server that enables programmers to write and test their programs on a local webserver. It was created by Apache Friends, and the audience can update or modify its native source code. It includes the Apache HTTP Server, MariaDB, and interpreters for many programming languages such as PHP and Perl (javatpoint, 2022).

### 2.1.7. Database Management System – MYSQL

MySQL is a relational database management system that is free and open source. MySQL, like other relational databases, stores data in tables comprised of rows and columns. MySQL operates on almost every platform, including Linux, UNIX, and Windows. Although it can be used for a variety of purposes, MySQL is most commonly linked with web applications and online publishing (Moore, 2022).

## 2.2. Methodology

Software methodology refers to the systematic, theoretical analysis of the methods applied to the development of software. It is the way in which software is developed and includes the principles and procedures that guide software developers in their work, and it helps to ensure that the software is developed in a systematic, rigorous, and objective manner. There are different software methodologies, and the appropriate methodology for a particular project will depend on the requirements of the software, the resources available, and the preferences of the development team development (Alliance Software, 2022). Every day, a massive chunk of software is planned, designed, built, and deployed, and each of these activities needs intense concentration and teamwork. Software businesses contemplate software development approaches to examine client requirements and construct a strong business solution on time (Positiwise, 2022).

Some benefits of using a software methodology are as follows:

- It provides structure and predictability to the software development process.
- It improves communication and collaboration within the development team.
- It reduces the risk of project failure.
- It increase the quality of the software.
- It increases efficiency and optimizes workflows.
- It helps to manage scope, schedule, and budget.

### 2.2.1. Methodology Consideration

When considering a software methodology for a project, it is essential to consider a number of factors in order to select the methodology that is best suited to the project. The nature of the software being produced, the preferences and demands of the development team, the resources available, the software's specifications, and the needs of the end users are all important factors to consider. The ideal software methodology will be selected based on a combination of these and other criteria, and it is necessary to thoroughly assess all of these factors in order to select the best methodology for the project.

#### i. Scrum Methodology

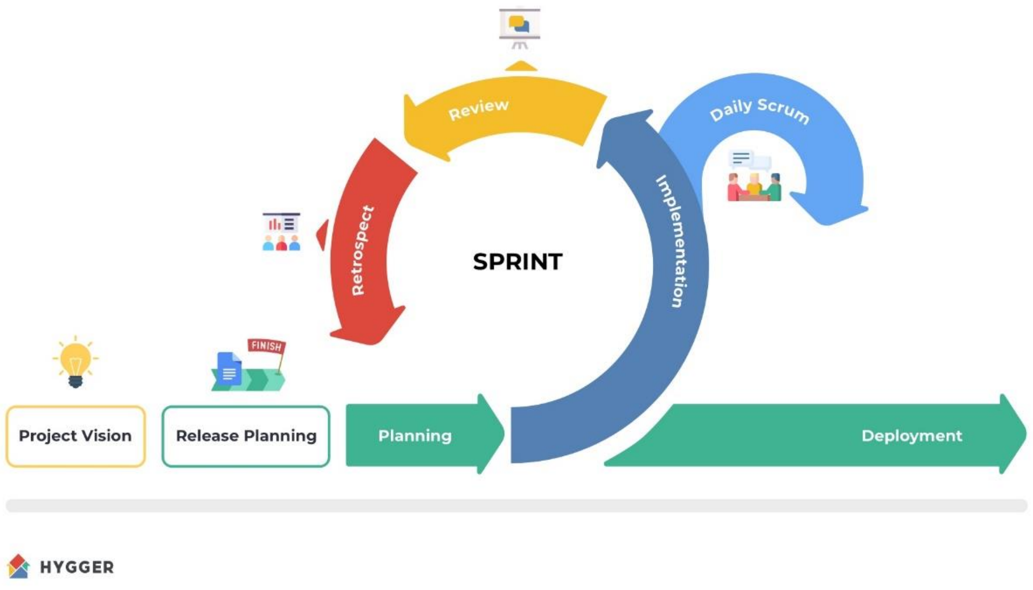


*Figure 1: Scrum Methodology*

Scrum is an agile methodology for managing large projects, particularly those involving software development. It entails working in short, iterative cycles known as "sprints" to develop functional software. The approach highlights transparency, inspection, and adaptability, and it encourages team members to communicate and collaborate frequently. In Scrum, the team meets every day to verify with one another and guarantee that the task is on track. The team holds a feedback session at the conclusion of each sprint to showcase completed work and address any issues that have arisen. The team holds a summary meeting following the review to reflect on the sprint and identify areas for improvement. Scrum is designed to be flexible and adaptable, and it is well-suited for projects where requirements are likely to change. Scrum emphasizes the significance of providing value to the customer and supports decision-making based on scientific facts. It also highlights self-management and self-organization among team members and fosters an environment of continual improvement (javatpoint, 2022).

Some benefits of Scrum methodology are as follows:

- **Self-organizing teams:** Scrum teams are self-organizing, which means they must decide how to successfully complete the task. This might lead to enhanced team spirit and commitment.
- **Flexibility and Adaptability:** Scrum is designed to be flexible and adaptable, allowing teams to adjust more easily to changing priorities and requirements.
- **Improved collaboration:** Scrum recognizes the significance of frequent interaction and collaboration among team members, which can result in better teamwork and increased production.
- **Faster delivery:** Working in short, iterative cycles enables teams to release working software faster and gather feedback from stakeholders faster in the process.
- **Continuous improvement:** The concentration on producing working software and receiving frequent feedback supports teams in recognizing and resolving issues early in the process, which can result in higher-quality software.
- **Greater transparency:** Scrum facilitates clear communication among the team and with stakeholders, which can boost trust and cooperation.

## ii. DSDM

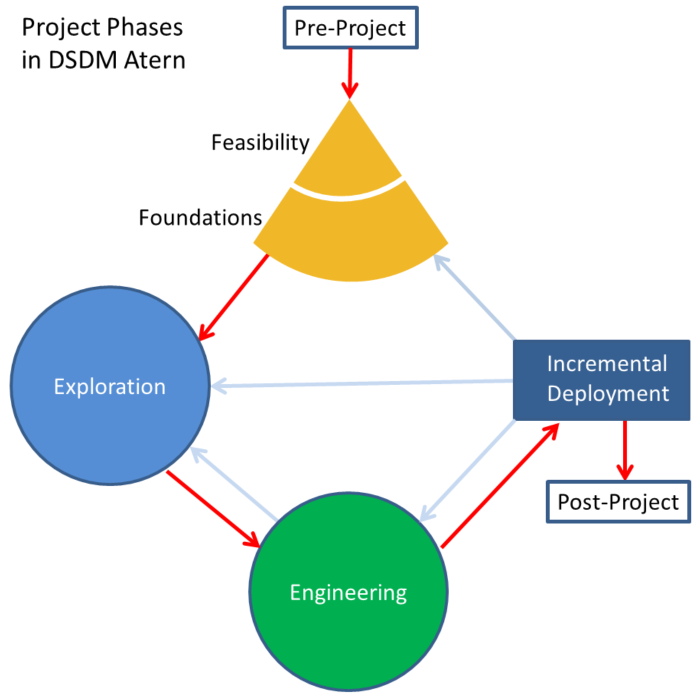


*Figure 2: DSDM Methodology*

DSDM, or Dynamic Systems Development Method, is an agile methodology used for managing the development of software and other systems. It is based on the principles of active user involvement, frequent delivery, and focus on the business need, and emphasizes the importance of delivering working software quickly and getting frequent feedback from stakeholders. Teams work in short, iterative cycles called "timeboxes" to deliver working software, holding regular progress reviews and review meetings to ensure that the work is on track (QA LEAD, 2022). In addition to these features, DSDM is also known for its flexibility and adaptability, which make it an ideal choice for projects where there is a need for frequent communication and collaboration between team members and where requirements are likely to change. Overall, DSDM is a powerful and effective methodology that can help teams deliver high-quality working software efficiently and effectively. So, it is a highly attractive methodology for managing complex projects (ProductPlan, 2022).

Some benefits of DSDM methodology are as follows:

- **Active user involvement:** DSDM highlights the significance of actively incorporating consumers in the development process, which can result in a final product that better meets their needs.
- **Frequent delivery of working software:** Teams implementing DSDM can gather input from stakeholders earlier in the process by working in short, iterative cycles and releasing working software consistently.
- **Focus on the business need:** DSDM places a significant emphasis on achieving business requirements, which can assist teams in delivering products with higher impact and relevance.
- **Flexibility and adaptability:** DSDM is designed to be flexible and adaptive, allowing teams to quickly manage to change requirements and priorities.
- **Continuous improvement:** The primary focus on providing working software quickly and receiving frequent feedback enables DSDM teams to continually enhance the process and provide higher-quality products.

### iii. Extreme Programming (XP)

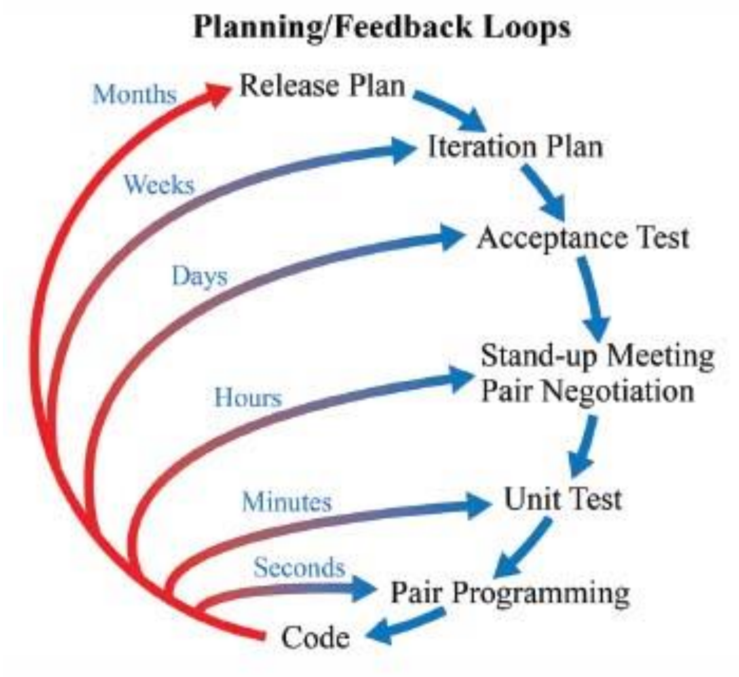


*Figure 3: Extreme Programming Methodology*

Extreme Programming (XP) is an agile software development process that focuses on simplicity, collaboration, and feedback to produce high-quality products on time. It was created in the late 1990s as a response to the more traditional, Waterfall approach to software development, which involved extensive planning and documentation before any actual coding took place. XP consists of a set of practices and attitudes designed to support development teams in working quickly and effectively. Among these methods are continuous integration, pair programming, frequent releases, and automated testing. XP also highlights the importance of communication and collaboration within the development team as well as with partners to identify that everyone is on the same page and collaborating toward the same objective. It is often used in conjunction with agile methodologies, which share similar principles and practices. Overall, the purpose of XP is to provide high-quality software that efficiently and effectively serves the needs of end users (AgileAlliance, 2022).

Some benefits of Extreme Programming (XP) are as follows:

- **Faster delivery of working software:** XP highlights the necessity of delivering functional software quickly, which can assist organizations in bringing new products or services to market more quickly.
- **Higher-quality software:** XP demonstrate regular testing and feedback, which can assist teams in detecting and resolving issues early in the development process, resulting in higher-quality software.
- **Greater customer satisfaction:** XP focuses on client interaction and input, which can enhance the guarantee that the final product satisfies the client's expectations and needs.
- **Increased flexibility and adaptability:** XP drives teams to be flexible and adaptable, which can organizations respond effectively to market or business environment developments.
- **Improved team morale:** XP places a strong emphasis on respect, communication, and collaboration, which can result in a good and friendly team culture.
- **Cost savings:** By delivering working software quickly and effectively, XP can help organizations in reducing operating costs and avoid the setbacks and overruns that affect traditional software development projects.

### 2.2.2. Selected Methodology

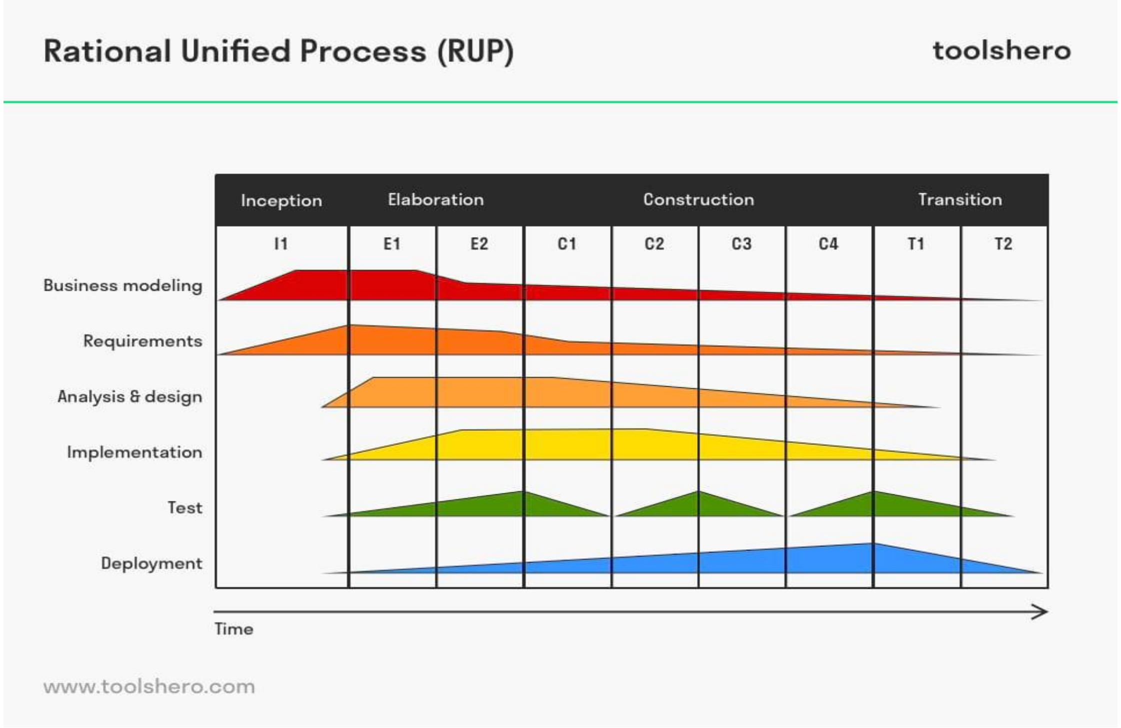


*Figure 4: RUP Methodology*

The RUP (Rational Unified Process) is a popular software development methodology that provides a structure for organizing and managing software system development. It is based on the idea of incremental and iterative development, which means that the program is developed in a succession of increments, each of which builds on the preceding one. The RUP methodology comprises a set of best practices and methodologies for managing the development process, which includes requirements gathering, design, implementation, testing, and deployment. It also contains templates and tools for documenting and tracking the development effort's progress. The RUP methodology is flexible and can be adjusted to a project's individual requirements, making it suitable for a variety of software development projects (Pedamkar, 2022). This methodology's role is to guarantee the timely and cost-efficient production of effective software that meets the needs of its end users (GeeksforGeeks, 2022).

Here are the several reasons why I selected to use the RUP methodology for my project:

- It is well-documented and well-structured, making it simple to follow and understand.
- It is flexible and adaptable, enabling customization based on project requirements.
- It is based on software development best practices, which help to guarantee that the software being developed is of high standard.
- It provides tools and templates for documenting and tracking the performance of the development effort.
- It is broadly utilized and has a large practitioner community, making it easy to access information and support.
- It is appropriate for a wide variety of software development projects.

**Phases of RUP Methodology**

The RUP (Rational Unified Process) methodology is based on the concept of iterative and incremental development, in which software is created in a series of increments, each of which builds on the preceding one (Pedamkar, 2022). The RUP methodology has four phases throughout its lifecycle:

i. **Initiation:** This phase is the preliminary or the first phase of the development process that focuses on clarifying the project's initial boundaries and highlighting critical stakeholders. It entails collecting and documenting the software's requirements as well as developing a high-level strategy for the development effort (GeeksforGeeks, 2022).
   The following are the key process that takes place during the initiation phase:
   - Project Idea
   - Project Finalization
   - Research about project related topics in depth
   - Requirement Gathering
   - Research about the technology used

ii. **Elaboration:** This phase is the second phase of the development process that entails fine-tuning the initial software design and developing a more thorough development strategy. It consists of actions such as prototyping, risk assessment, and the creation of a thorough project timeline (GeeksforGeeks, 2022).

The following are the key process that takes place during the elaboration phase:

- Identify project tasks and its completion date
- Project Architecture
- Project UML Diagrams
- Risk Analyzation
- Proposal Confirmation

iii. **Construction:** This phase is the third phase of the development process that includes the actual software development, which includes coding, testing, and integration (GeeksforGeeks, 2022).

The following are the key process that takes place during the elaboration phase:

- Development
- Testing
- Prepare Documentation

iv. **Transition:** This phase is the last or final phase of the development process that deals with deploying the software and delivering it to the production environment. User acceptance, testing, training, and deployment are all part of the process (GeeksforGeeks, 2022).

The following are the key process that takes place during the elaboration phase:

- Supervisor Reviews and Feedbacks
- Updates and Upgrades
- Update Project Documentation
- Project Submission

### 2.2.2. Methodology Section Justification

Here is a methodology section justification of the considered methodology and chosen methodology which is presented in the table below:

| Methodology | Key Features |
|---|---|
| Rational Unified Process (RUP) | • Comprehensive Documentation<br>• Formal reviews and inspections<br>• Focus on architecture and design<br>• Suitable for large, complex projects |
| Scrum | • Flexibility and adaptability<br>• Emphasis on rapid iteration and delivery<br>• Self-organizing teams<br>• Appropriate for smaller, agile projects |
| Extreme Programming (XP) | • Focus on customer satisfaction<br>• Collaborative development<br>• Continuous integration and testing<br>• Appropriate for small, agile projects |

## 2.3. Similar Systems

### 2.3.1. Similar System Consideration

### i. System 1

System Name: Uber

URL: https://www.uber.com/

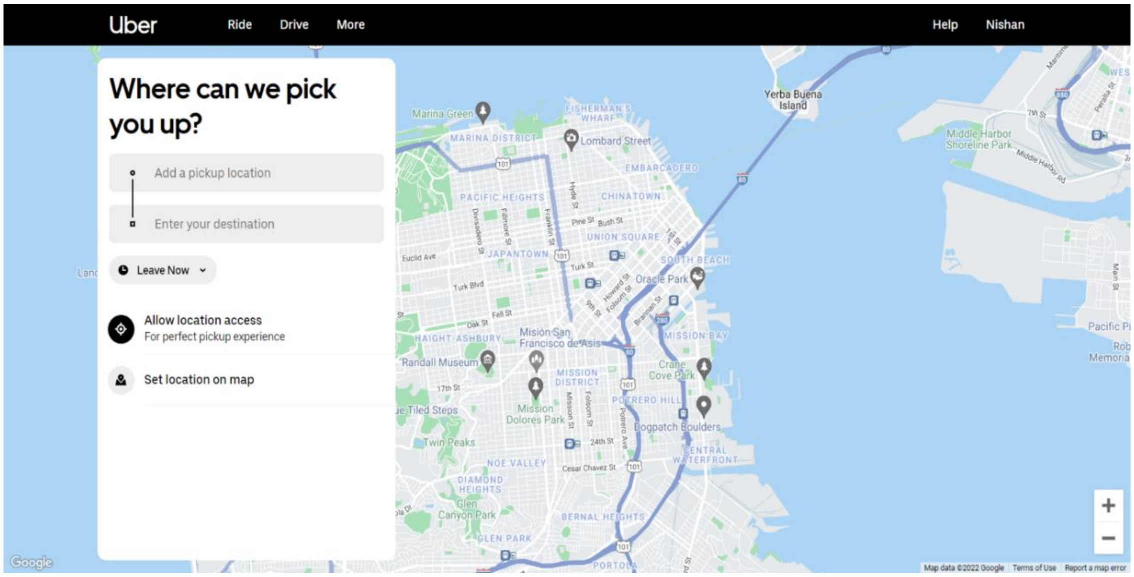


*Figure 5: Uber*

Uber is a technology company that offers a range of transportation services to consumers around the world. The company was founded in 2009 and has since become a major player in the sharing economy, operating in more than 10,000 cities in over 65 countries. To use Uber's ride-hailing service, users download the Uber app on their smartphone and create an account. They can then request a ride from a nearby Uber driver by entering their destination into the app. Payment is automatically charged to the user's credit card or PayPal account, and users can rate their driver and provide feedback about their experience. In addition to its ride-hailing service, Uber also offers a food delivery service called Uber Eats, which allows users to order meals from local restaurants and have them delivered to their location (BLYSTONE, 2022).

### ii. System 2

System Name: OLA

URL: https://www.olacabs.com/

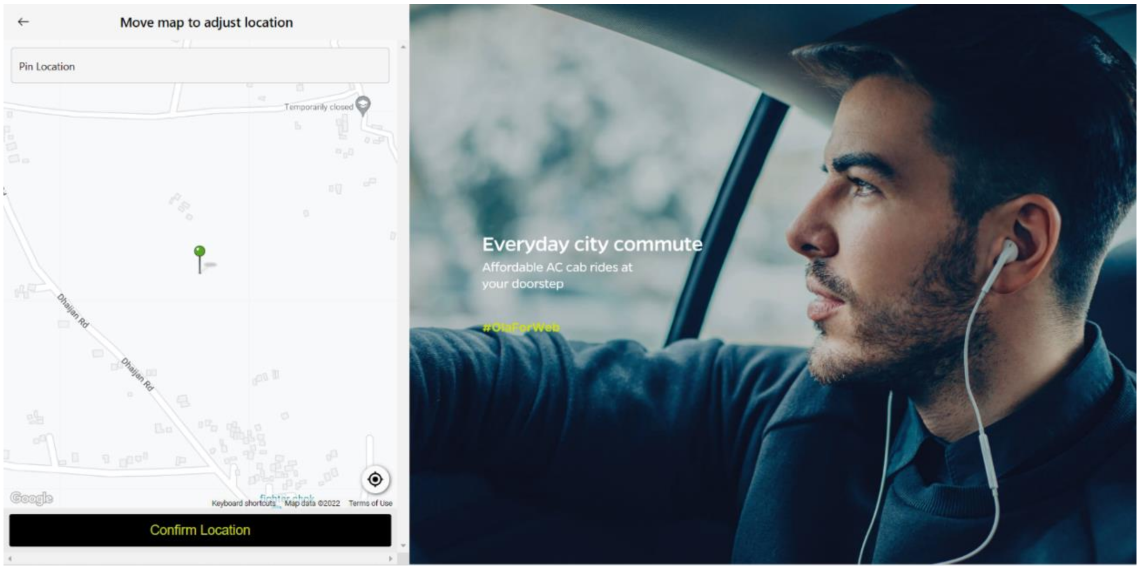


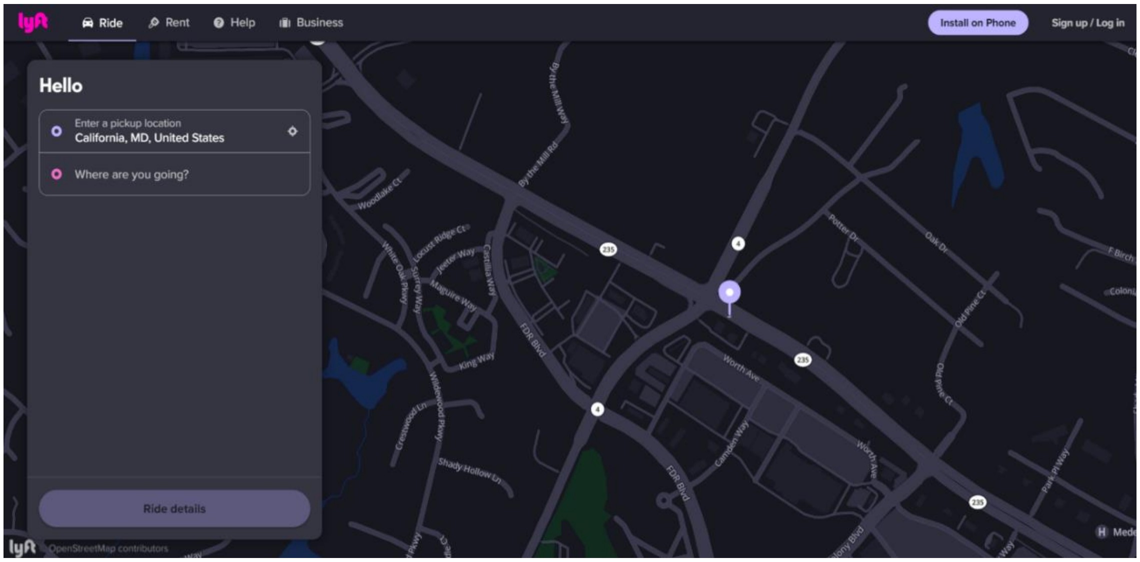
*Figure 6:OLA*

Ola is a technology company that offers a range of transportation services, including ride-hailing, auto-rickshaw hailing, and food delivery. The company was founded in 2011 in Mumbai, India and has since expanded to operate in over 250 cities in India and a number of other countries including Australia, New Zealand, and the United Kingdom. To use Ola's ride-hailing service, users download the Ola app on their smartphone and create an account. They can then request a ride from a nearby Ola driver by entering their destination into the app. Payment is automatically charged to the user's credit card or linked bank account, and users can rate their driver and provide feedback about their experience. In addition to its ride-hailing service, Ola also offers a food delivery service called Ola Food, which allows users to order meals from local restaurants and have them delivered to their location (engineeringforchange, 2022).

### iii. System 3

System Name: Lyft

URL: https://www.lyft.com/



Hello

Enter a pickup location
California, MD, United States

Where are you going?

*Figure 7:Lyft*

Lyft is a technology company that offers a ride-hailing service to consumers in the United States and Canada. The company was founded in 2012 and has since become a major player in the North American ride-hailing market, operating in more than 600 cities across the two countries. To use Lyft's service, users download the Lyft app on their smartphone and create an account. They can then request a ride from a nearby Lyft driver by entering their destination into the app. Payment is automatically charged to the user's credit card or linked bank account, and users can rate their driver and provide feedback about their experience. In addition to its standard ride-hailing service, Lyft also offers a number of other options, including shared rides, luxury rides, and electric scooter and bike rentals (DAVIS, 2022).

### 2.3.2. Similar System Comparison

There are several ride-sharing system that have similar models to my FYP project which I have mentioned above. Here is the comparison of my system to the above mentioned system are shown in the table below:

*Table 1: Comparison between similar projects*

| S.N | Features | Uber | Ola | Lyft | My project |
|---|---|---|---|---|---|
| 1 | Registration and login | ✓ | ✓ | ✓ | ✓ |
| 2 | Tuk-tuk service | ✗ | ✗ | ✗ | ✓ |
| 3 | Online payment | ✓ | ✓ | ✓ | ✓ |
| 4 | Offline payment | ✓ | ✓ | ✓ | ✓ |
| 5 | Rating and feedback | ✓ | ✓ | ✓ | ✓ |
| 6 | Bargain option | ✗ | ✗ | ✗ | ✓ |

In conclusion, my system is a better option than the abovementioned system as it offers a unique feature that they don't have: the ability to bargain on fare prices. This can be particularly useful for passengers who are looking to save money on their rides and are willing to negotiate with drivers to get a lower price. Additionally, my app also offers tuk-tuk service, which is not available on the other platforms. This allows passengers in certain regions to request rides on the traditional three-wheeled vehicles that are popular. These features make project more attractive option for passengers who are looking for cost-effective and culturally relevant transportation options. Along with its other advanced features and efficient and reliable system for matching passengers with drivers, my project offers a better experience for both passengers and drivers. Overall, my project “Ride Hero” will undoubtedly be successful if all of the features are implemented successfully.

# 3. Work To Date

## 3.1. Requirement Gathering

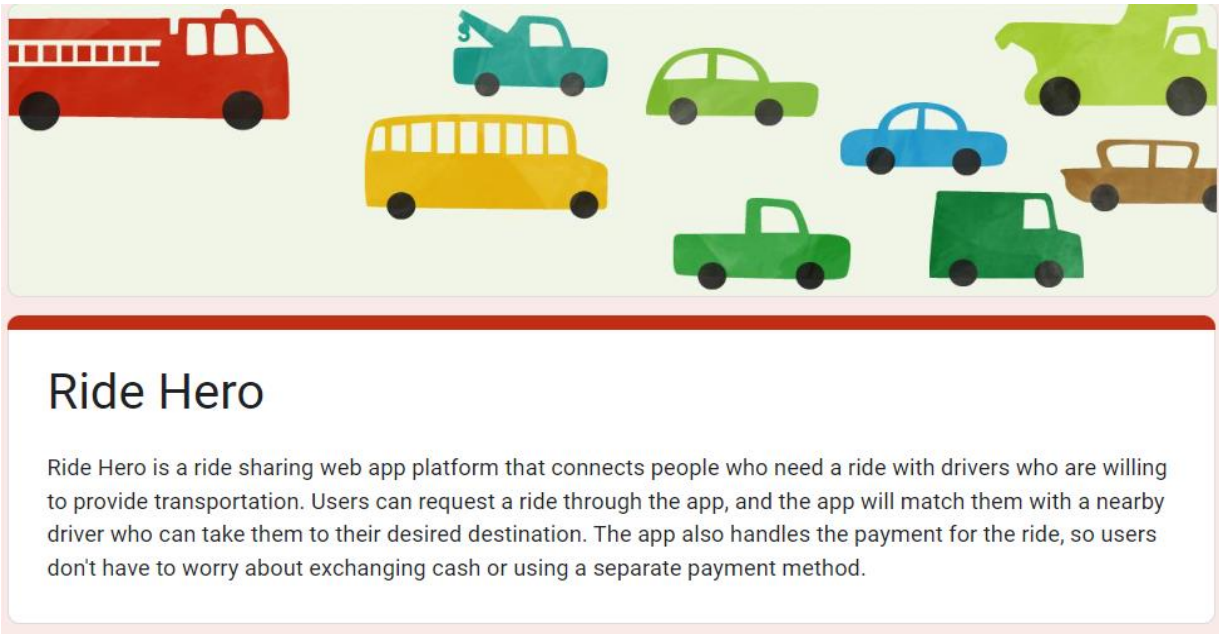
The main aim of this system is to provide a convenient, affordable and safe ride for people to get from one place to another. This web app generally works by connecting passengers with drivers who are willing to provide a ride. Passengers can request and pay for their ride, allowing them to track their driver in real-time, while at the same time the driver can see the exact location of the passenger and have access to the payment once the ride is complete.

There are several key requirements that should be gathered before developing a ride-sharing web app in order to create a successful and user-friendly service. These include:

- **Market research:** It is important to gather market research to understand the needs and preferences of potential riders and drivers, as well as the competitive landscape of existing ride-sharing services. This can be achieved through surveys, focus groups, and other market research methods.
- **User experience:** The web app should be simple to use and accessible for both passengers and drivers, with a clear and user-friendly interface and clear instructions for how to use it. This can be achieved through user testing and feedback, as well as through designing a basic and straightforward interface.
- **Reliability:** The web app should be dependable, with little downtime and errors, and it should be tested and debugged to guarantee it works properly. This can be handled through extensive testing and quality assurance procedures.
- **Integration:** The web app should be compatible with several technologies such as mapping and navigation systems, payment processors, and driver and passenger rating systems. This can contribute to a more seamless and convenient user experience.

### 3.1.1. Conducted Online Survey

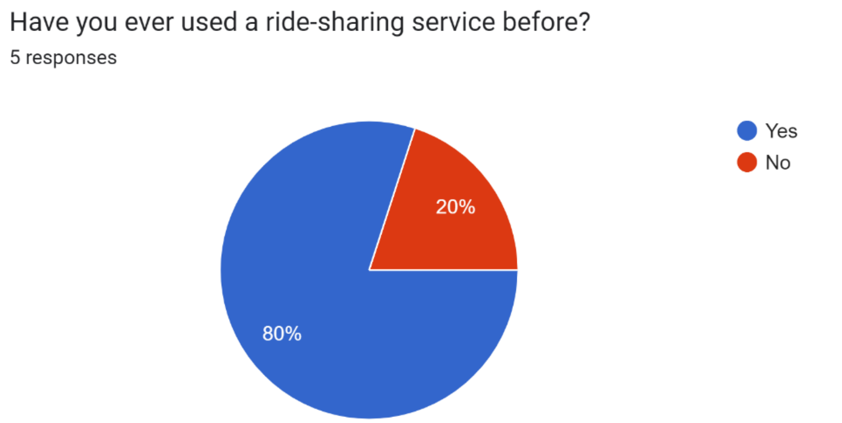
I conducted an online survey form to comprehend and understand the current situation of the ride-sharing web app in Nepal and to collect the feedback from the people to make better decisions and implementation while developing the web app.



Ride Hero

Ride Hero is a ride sharing web app platform that connects people who need a ride with drivers who are willing to provide transportation. Users can request a ride through the app, and the app will match them with a nearby driver who can take them to their desired destination. The app also handles the payment for the ride, so users don't have to worry about exchanging cash or using a separate payment method.

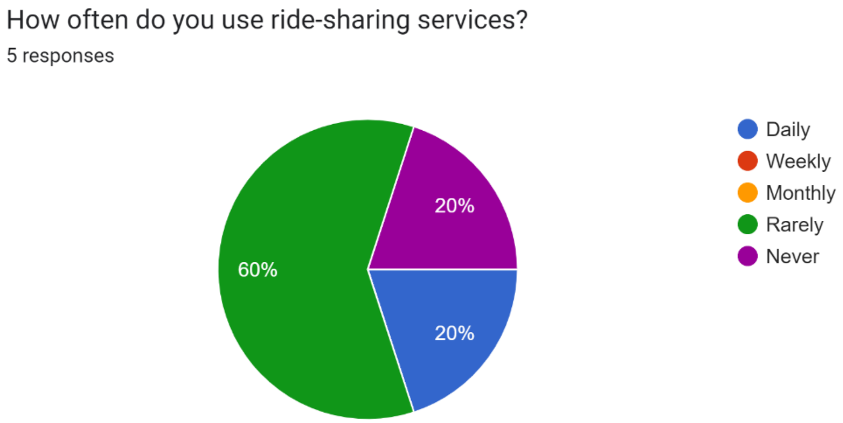
*Figure 6: Survey 1*



Have you ever used a ride-sharing service before?

5 responses

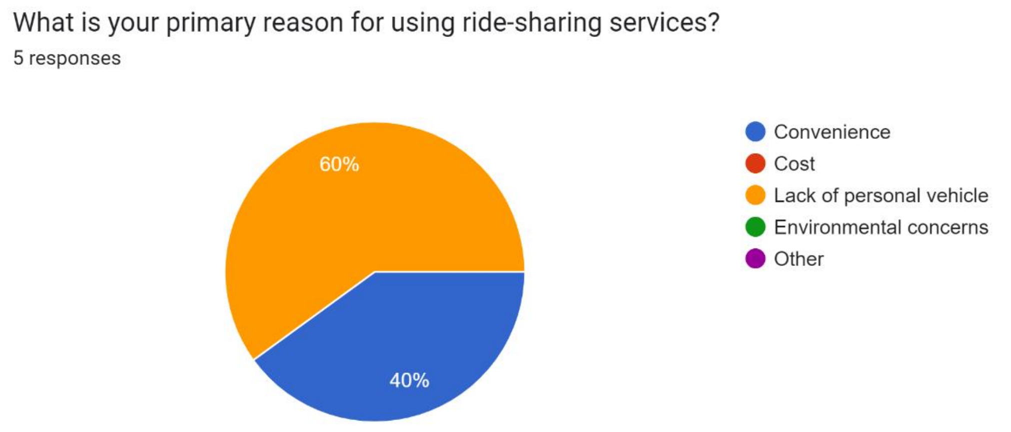
*Figure 7: Survey 2*



How often do you use ride-sharing services?

5 responses

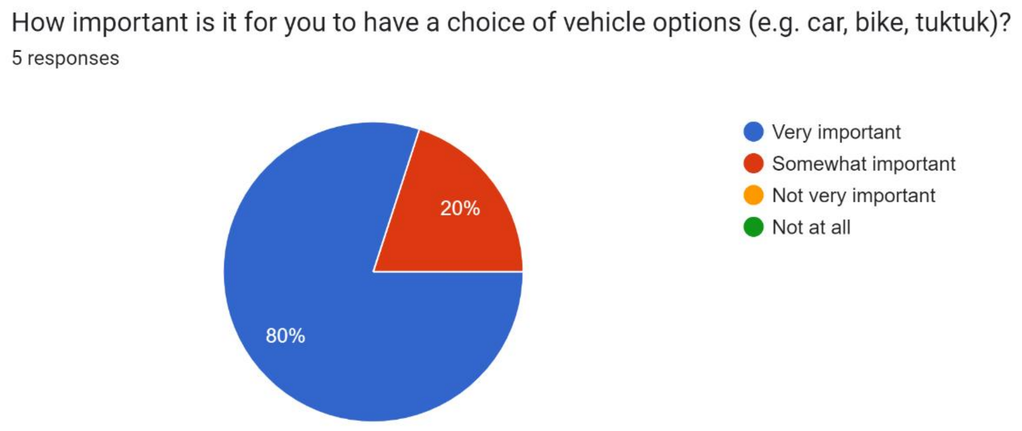
*Figure 8: Survey 3*



What is your primary reason for using ride-sharing services?

5 responses

*Figure 9: Survey 4*



How important is it for you to have a choice of vehicle options (e.g. car, bike, tuktuk)?

5 responses

*Figure 10: Survey 5*

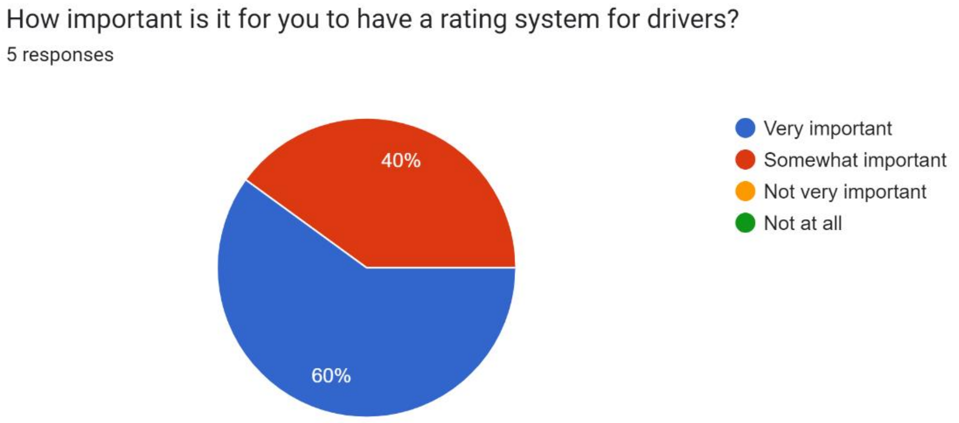


*Figure 11: Survey 6*

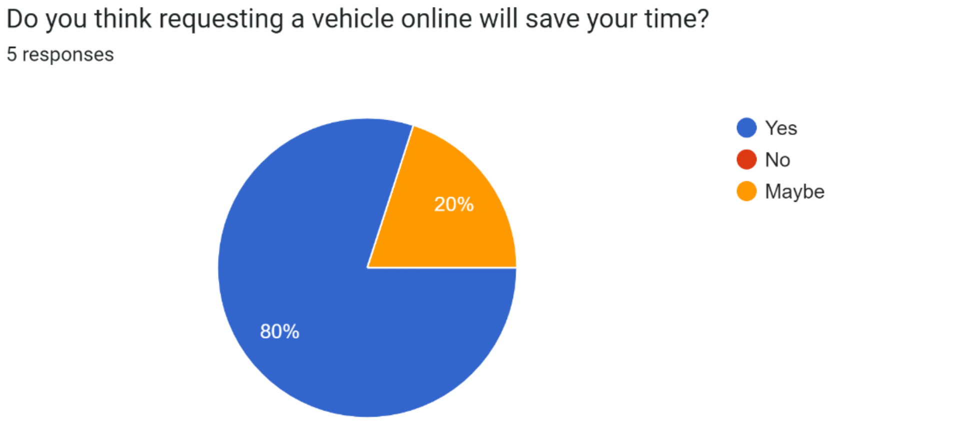


*Figure 12: Survey 7*

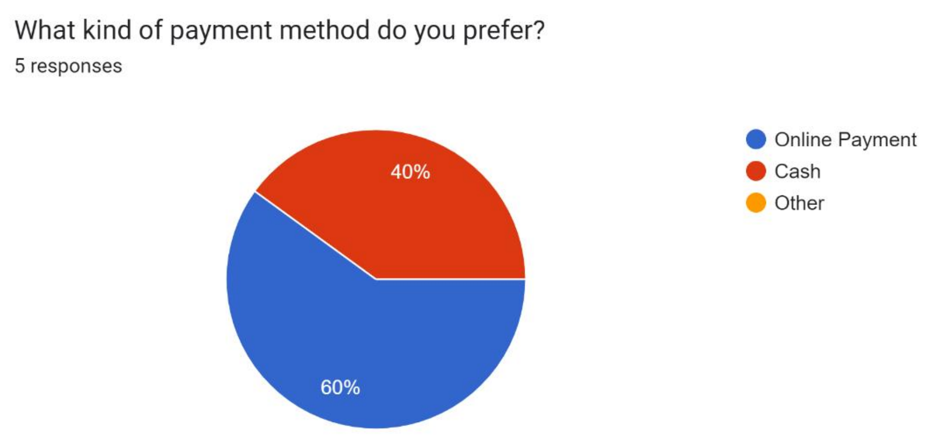


*Figure 13: Survey 8*

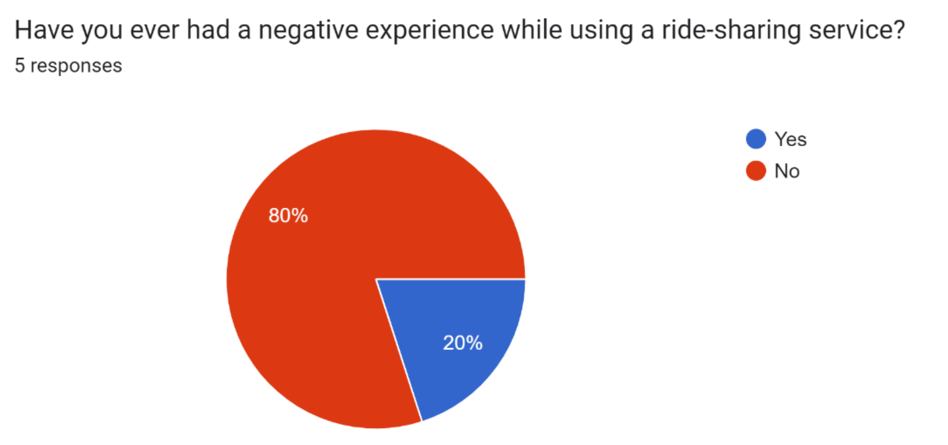


*Figure 14: Survey 9*

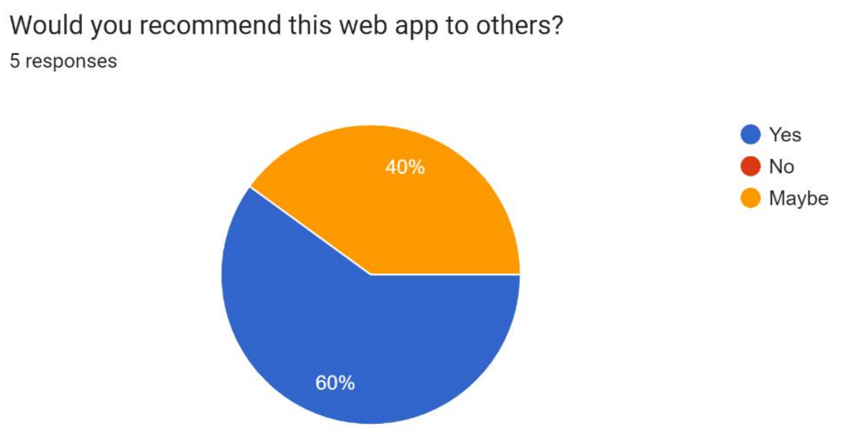


*Figure 15: Survey 10*

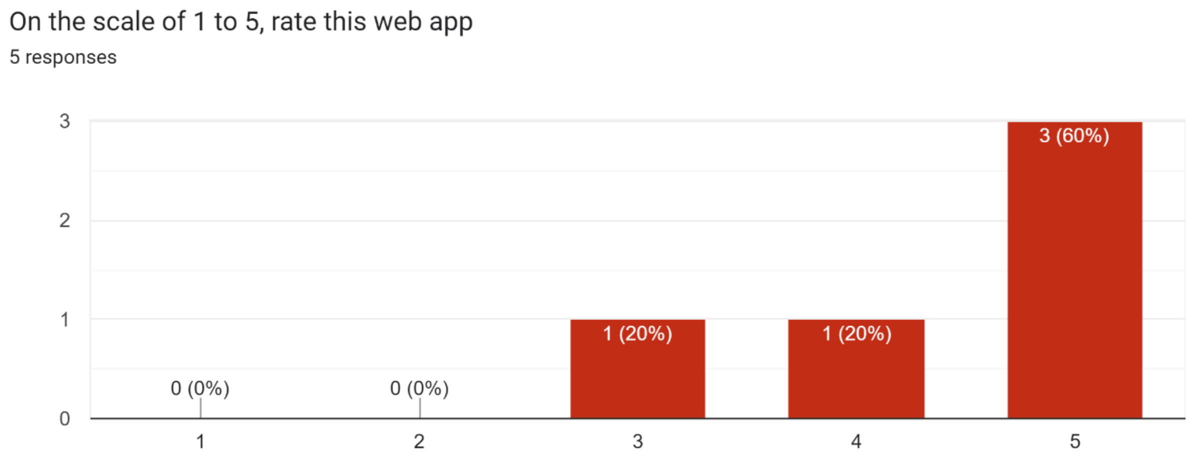


*Figure 16: Survey 11*

### 3.2. Use Case Diagram

A UML use case diagram is the principal form of system/software specifications for an undeveloped software program. Use cases define the intended behavior (what) rather than the exact technique of achieving it (how). A key concept of use case modelling is that it allows us to build a system from the perspective of the end user. It is an effective technique for explaining system behavior in user terms. It is made up of use cases, individuals, or other objects that call the actor features, and the sections that are responsible for putting the use cases into action. It illustrates how an outside entity interacts with the system. It depicts external entities that interact with the system's component (Walker, 2022).

The main purposes of use case diagrams are as follows:

- It is used to collect the requirements of a system.
- It is used to determine the external and internal influences on the system.
- It is used to show how the requirements interact with each other.
- It is used to represent the goals of system-user interaction.

### 3.2.1. Individual Use Case Diagram

### 1. Admin

#### a) Login

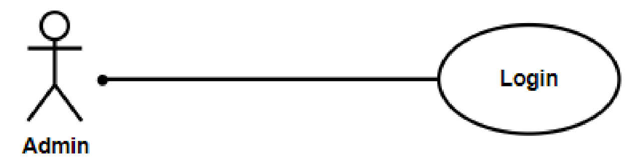


*Figure 177: Use Case of Admin Login*

#### b) Logout

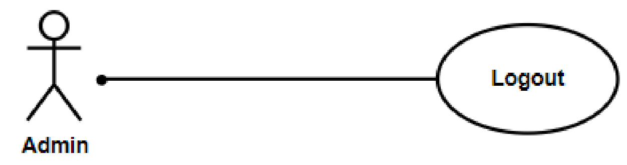


*Figure 18: Use Case of Admin Logout*

#### c) Manage Users

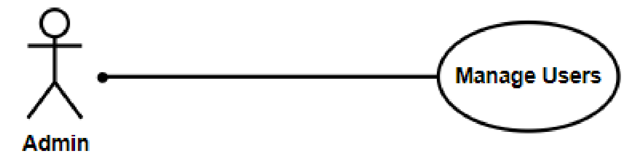


*Figure 19: Use Case of Admin Manage Users*

### d) Payment Management

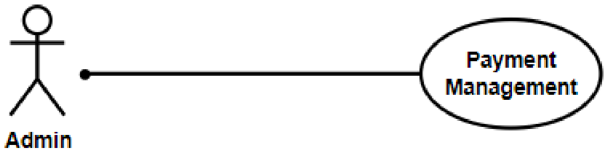


*Figure 20: Use Case of Admin Payment Management*

## 2. Passenger

### a) Registration

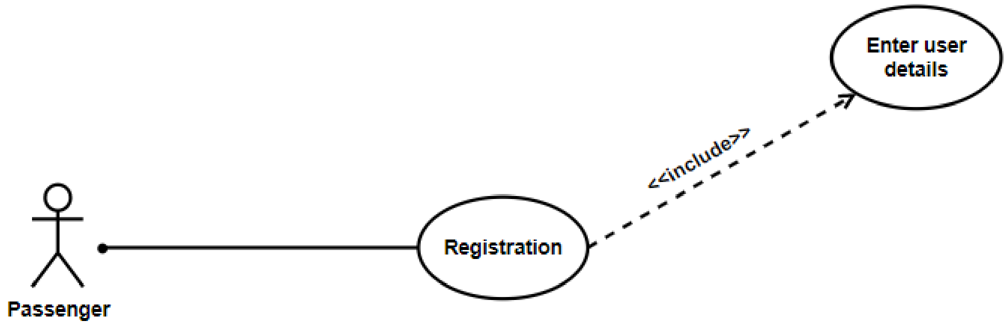


*Figure 21: Use Case of Passenger Registration*

### b) Logout

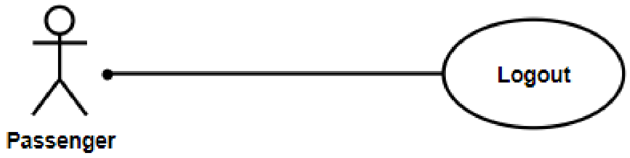


*Figure 22: Use Case of Passenger Logout*

## c) Login

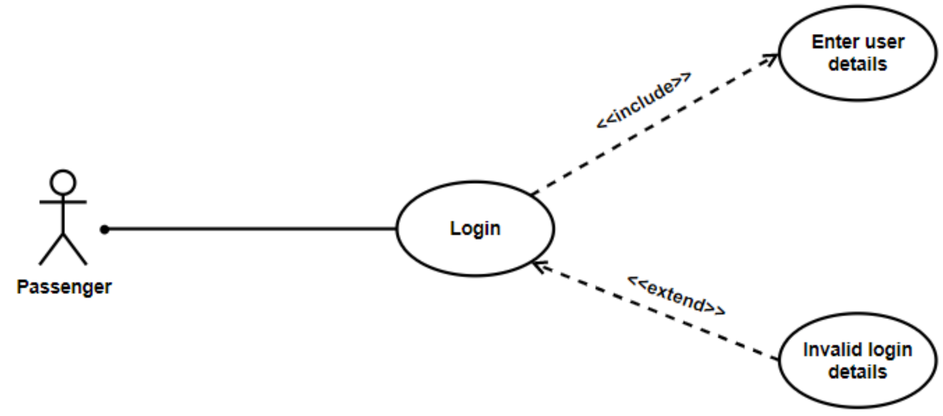


*Figure 23: Use Case of Passenger Login*

## d) Ride

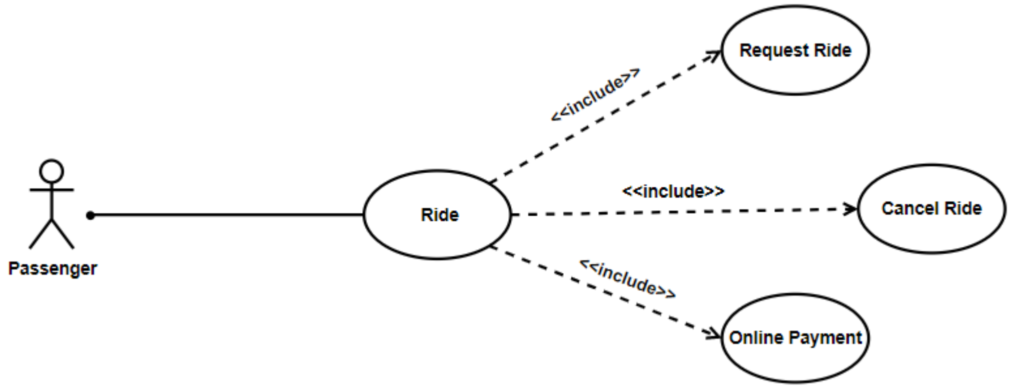


*Figure 24: Use Case of Passenger Ride*

### e) Ride History

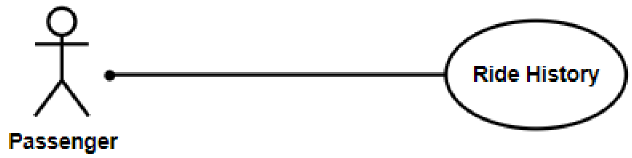


*Figure 25: Use Case of Passenger Ride History*

## 3. Driver

### a) Registration

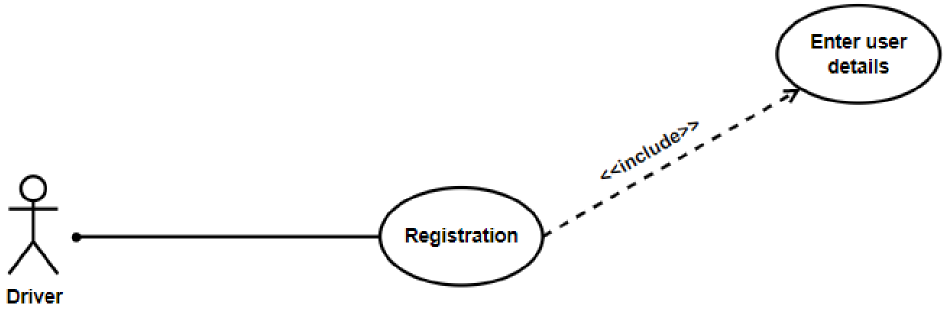


*Figure 26: Use Case of Driver Registration*

## b) Login

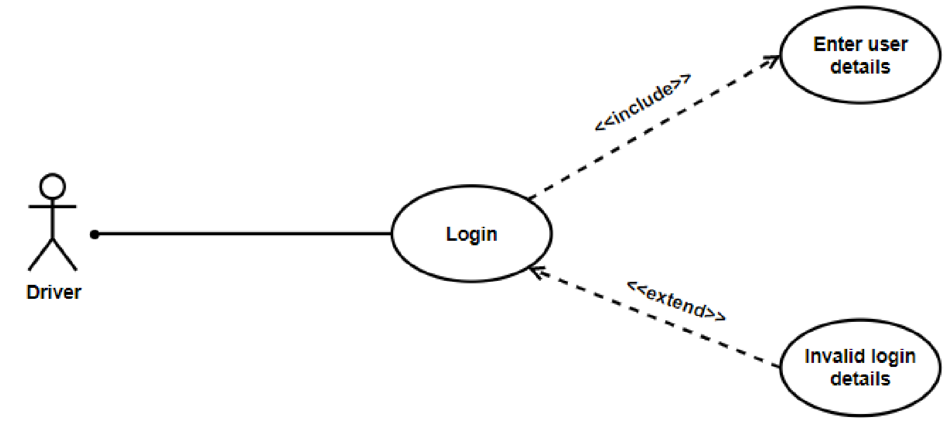


*Figure 27: Use Case of Driver Login*

## c) Logout

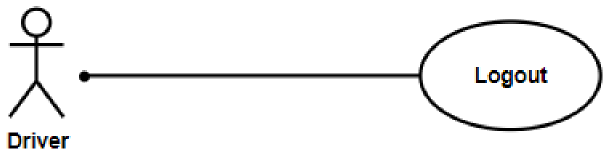


*Figure 28: Use Case of Driver Logout*

### d) KYC Verification

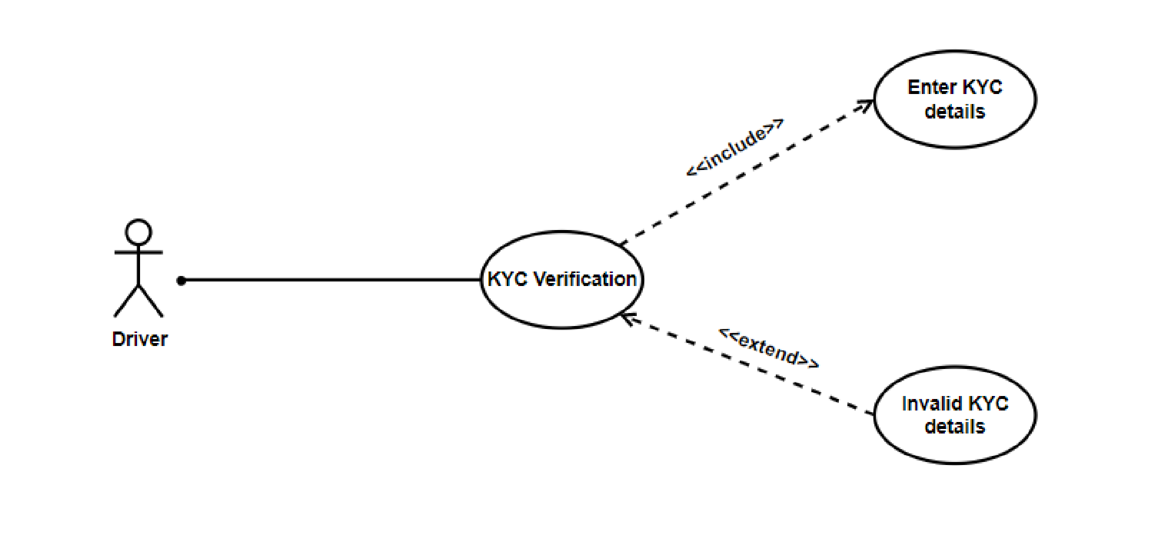


*Figure 29: Use Case of Driver KYC Verification*

### e) Manage Ride

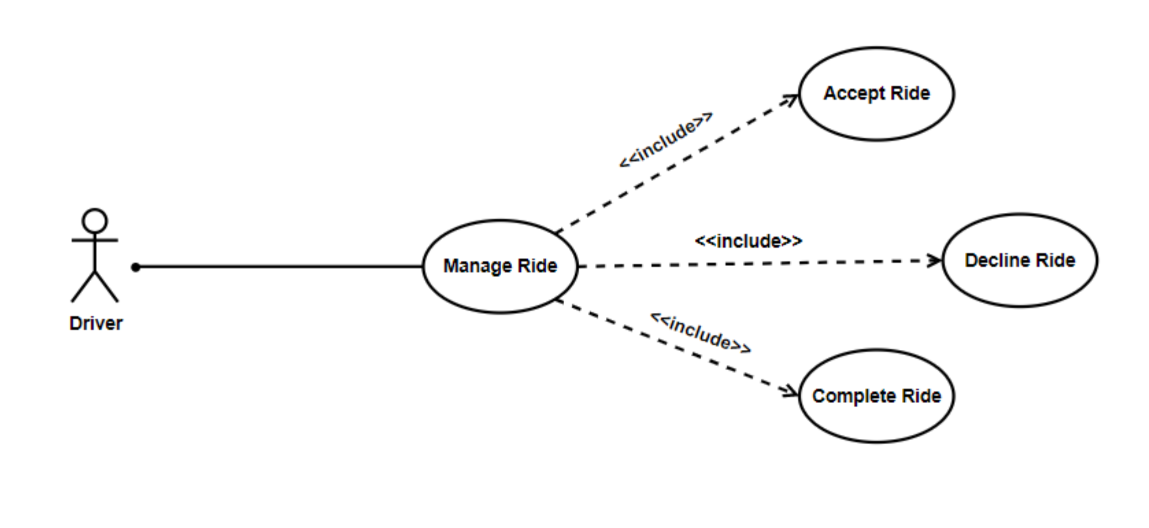


*Figure 30: Use Case of Driver Manage Ride*

### f) Ride History

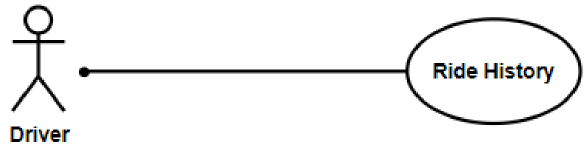
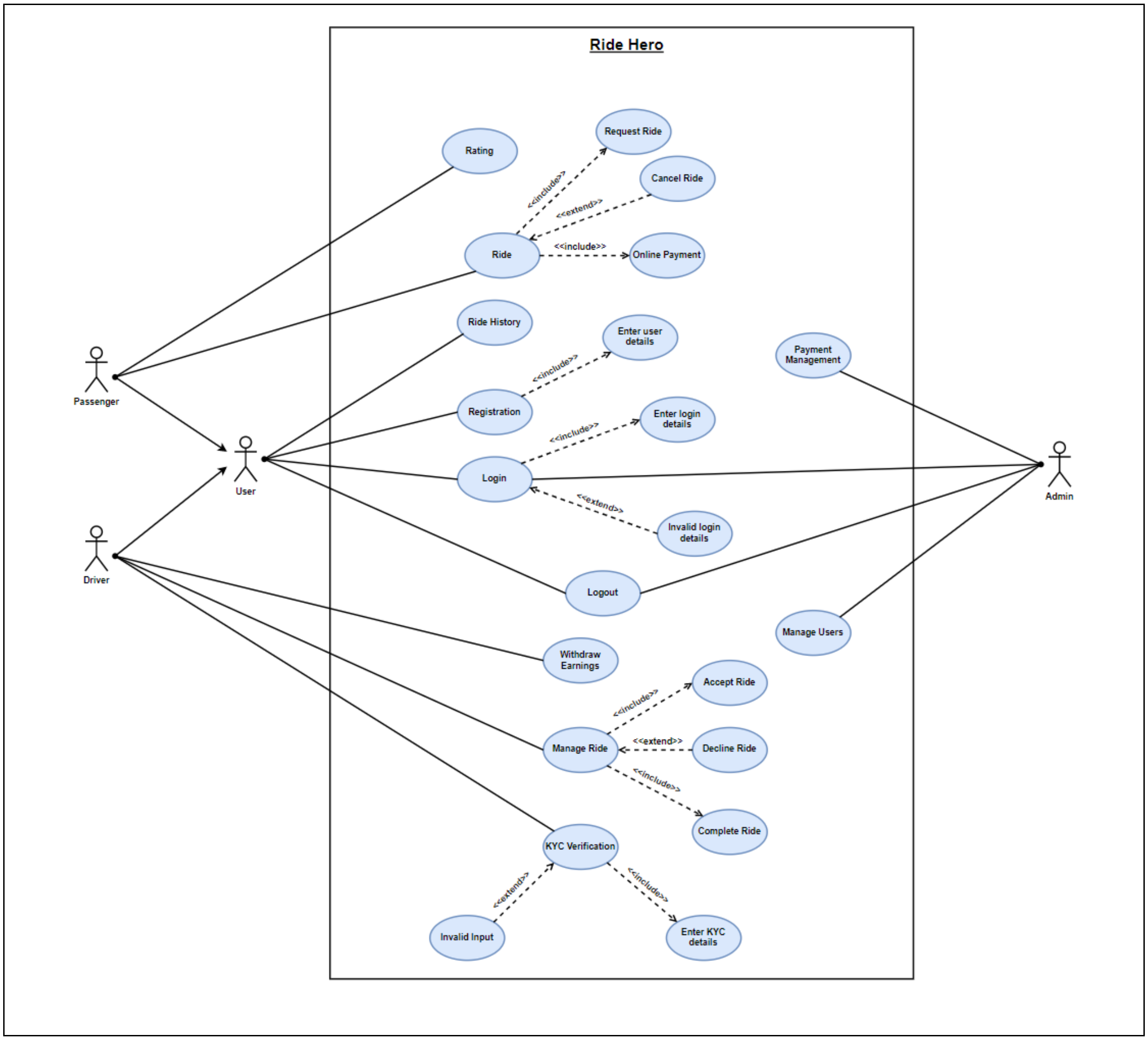


*Figure 31: Use Case of Driver Ride History*

### 3.2.2. System Use Case Diagram



## 3.3. High Level Use Case

A high-level use case is a summary of the steps required to accomplish a certain goal within a system or process. It provides an overview of the primary actors or roles involved, as well as the activities they undertake to achieve the goal, and is frequently used in software development to describe system requirements and to assist in the design of the overall architecture. High-level use cases can help stakeholders clarify their system requirements and expectations by enabling them to comprehend the big picture of a system or process and highlighting any potential difficulties or opportunities for improvement (GEOG 468: GIS Analysis and Design, 2022).

A high-level use case in a ride-sharing app, for example, might define the workflow for a passenger to request and complete a ride, including activities taken by the passenger and driver. A high-level use case normally does not go into great detail regarding the technical implementation of the activities or the system, but rather focuses on the overall flow of events and the primary aims and objectives.

Here is the high level use case of each actors with their description:

*Table 2: High Level Use Case*

| S.N. | Use Cases | Actors | Description |
|---|---|---|---|
| **1.** | Registration | Passenger, Driver | Passenger and driver provides necessary credentials requested by the web app to register into the web app. |
| **2.** | Login | Admin, Passenger, Driver | Admin, Passenger, and Driver must login with their registered email address and password. |
| **3.** | Logout | Admin, Passenger, Driver | Admin, Passenger, and Driver can logout from the web app. |
| **4.** | Manage Users (Passengers and Drivers) | Admin | Admins can manage passenger and driver accounts, including creating and deleting accounts, |

| | | | |
|---|---|---|---|
| | | | reviewing user information, and resetting passwords. |
| **5.** | Payment Management | Admin | Admins can view and manage the overall payment transaction. |
| **6.** | Ride (Accept, Cancel and Online Payment) | Passenger | Passengers can enter their pickup and drop-off locations, select a type of vehicle, and request a ride through the web app and can pay the fare digitally. |
| **7.** | Rating | Passenger | Passenger can rate the driver after the ride is completed. |
| **8.** | Ride History | Passenger, Driver | Passenger and Driver both can view their ride history. |
| **9.** | Withdraw Earnings | Driver | Driver can view and withdraw their earnings. |
| **10.** | Manage Ride (Accept, Decline, and Complete) | Driver | Driver can accept, decline or complete a ride. |
| **11.** | KYC Verification | Driver | Driver must provide the KYC details to be a verified driver. |

## 3.4. Wireframes

A wireframe is a visual representation of the structure of a website or application's user interface design. It is a user experience blueprint that shows how the interface will look and perform. Wireframes are often developed early in the design process and are used to demonstrate to stakeholders the structure, hierarchy, and functionality of a design. One of the distinctive characteristics of wireframes is that they concentrate on the structure and functioning of a design rather than its appearance. This allows designers and stakeholders to concentrate on the user experience and functionality of the interface rather than the visual aspects (ProductPlan, 2022).

### 3.4.1. Admin Dashboard

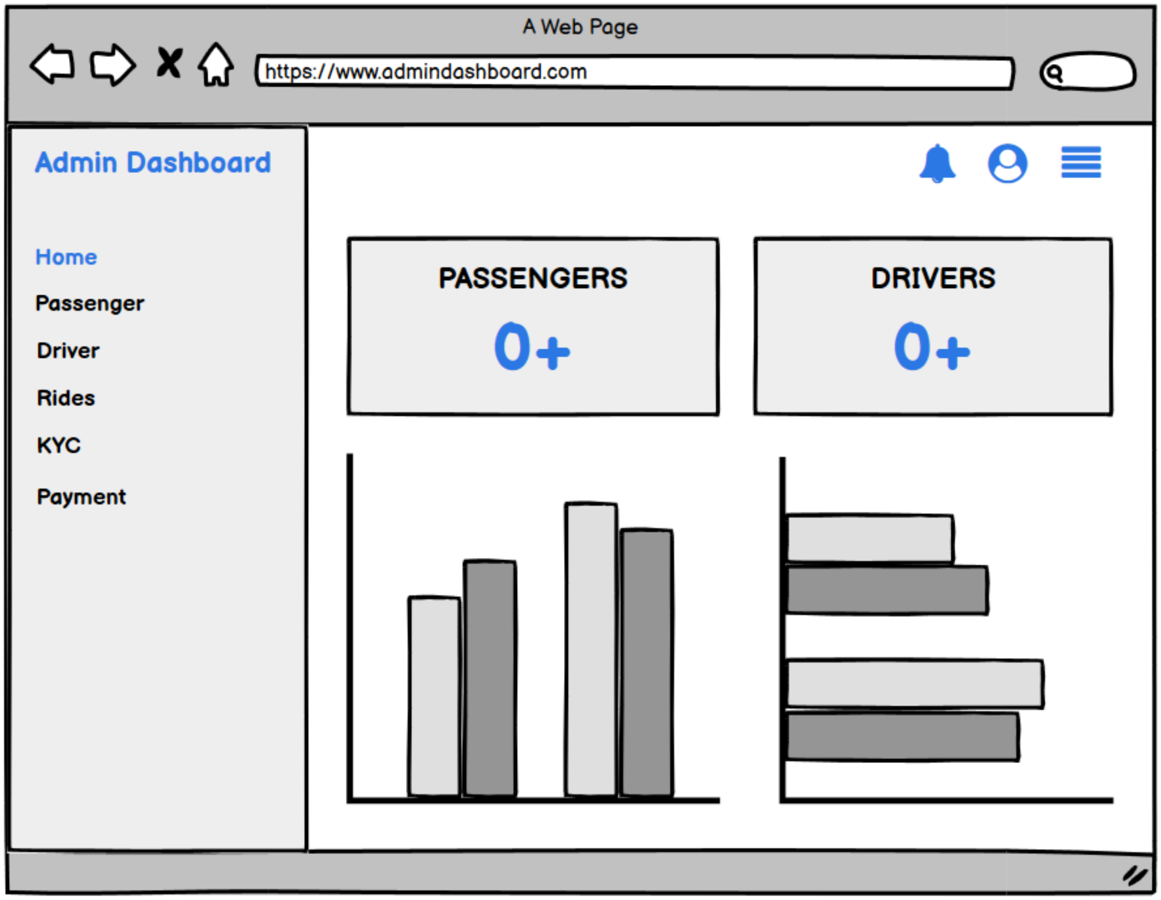


*Figure 32: Admin Dashboard*

### 3.4.2. Home Page

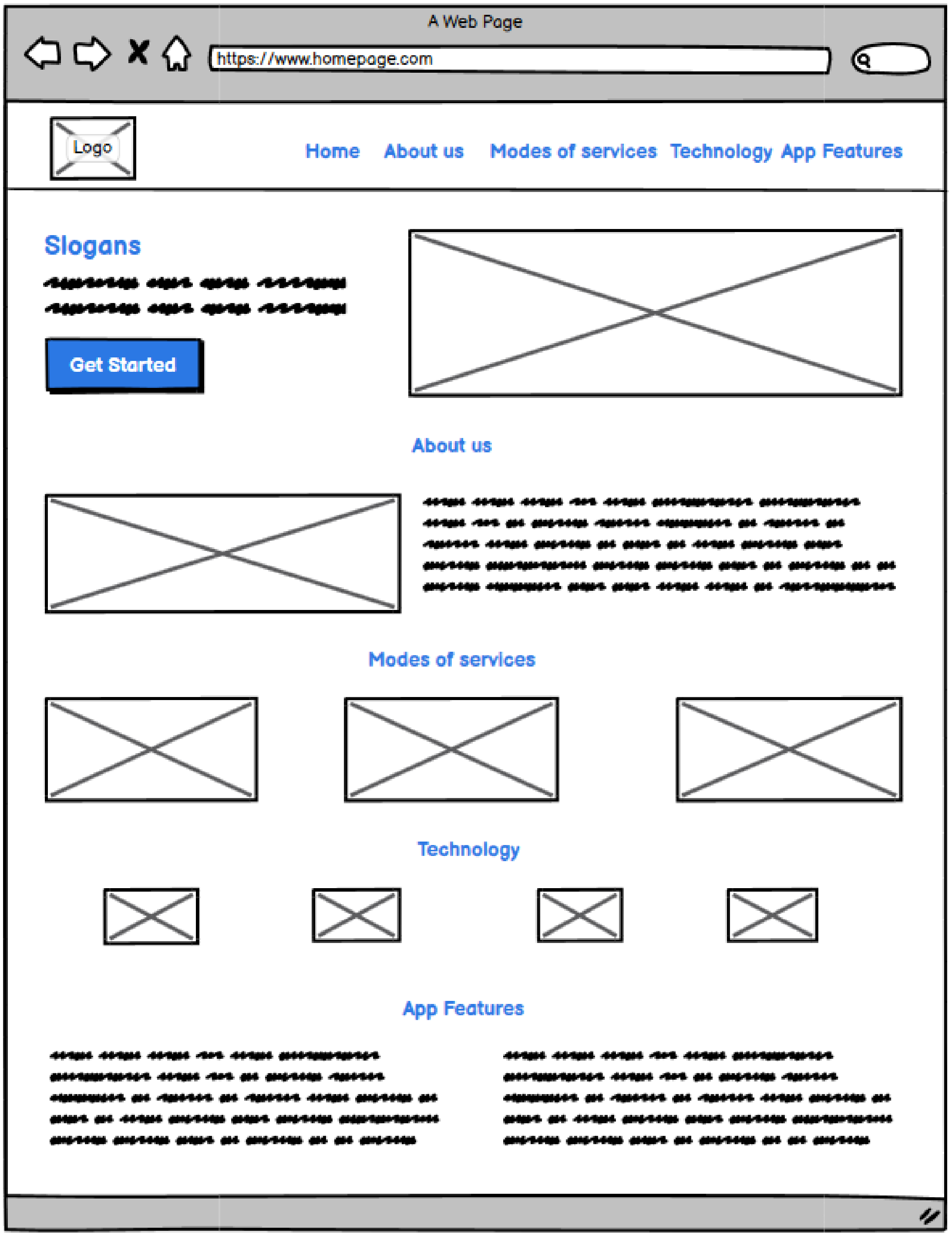
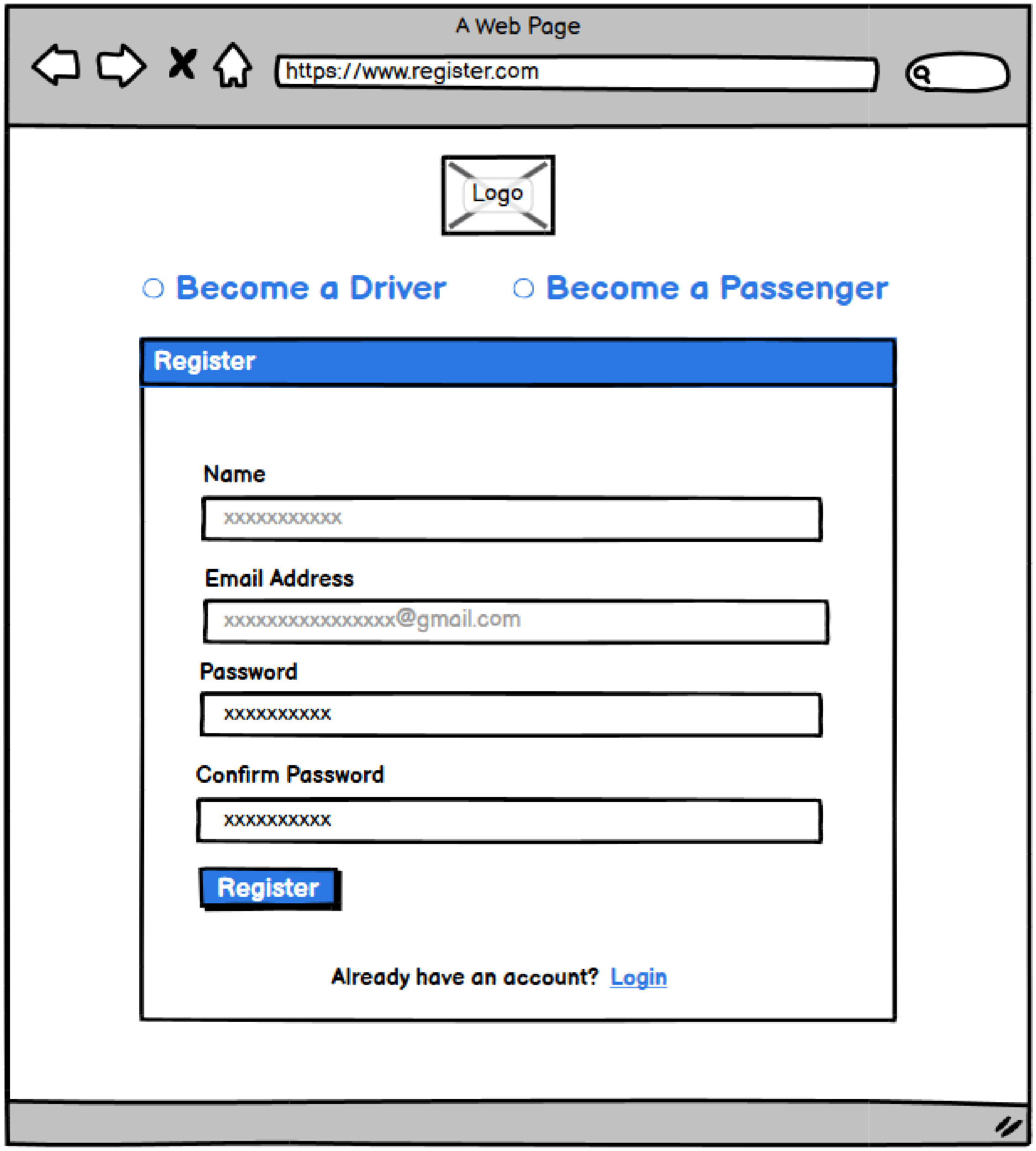


*Figure 33: Home Page*

### 3.4.3. Register Page



Name

xxxxxxxxxxx

Email Address

xxxxxxxxxxxxxxxx@gmail.com

Password

xxxxxxxxxx

Confirm Password

xxxxxxxxxx

*Figure 34: Register Page*

### 3.4.4. Login Page

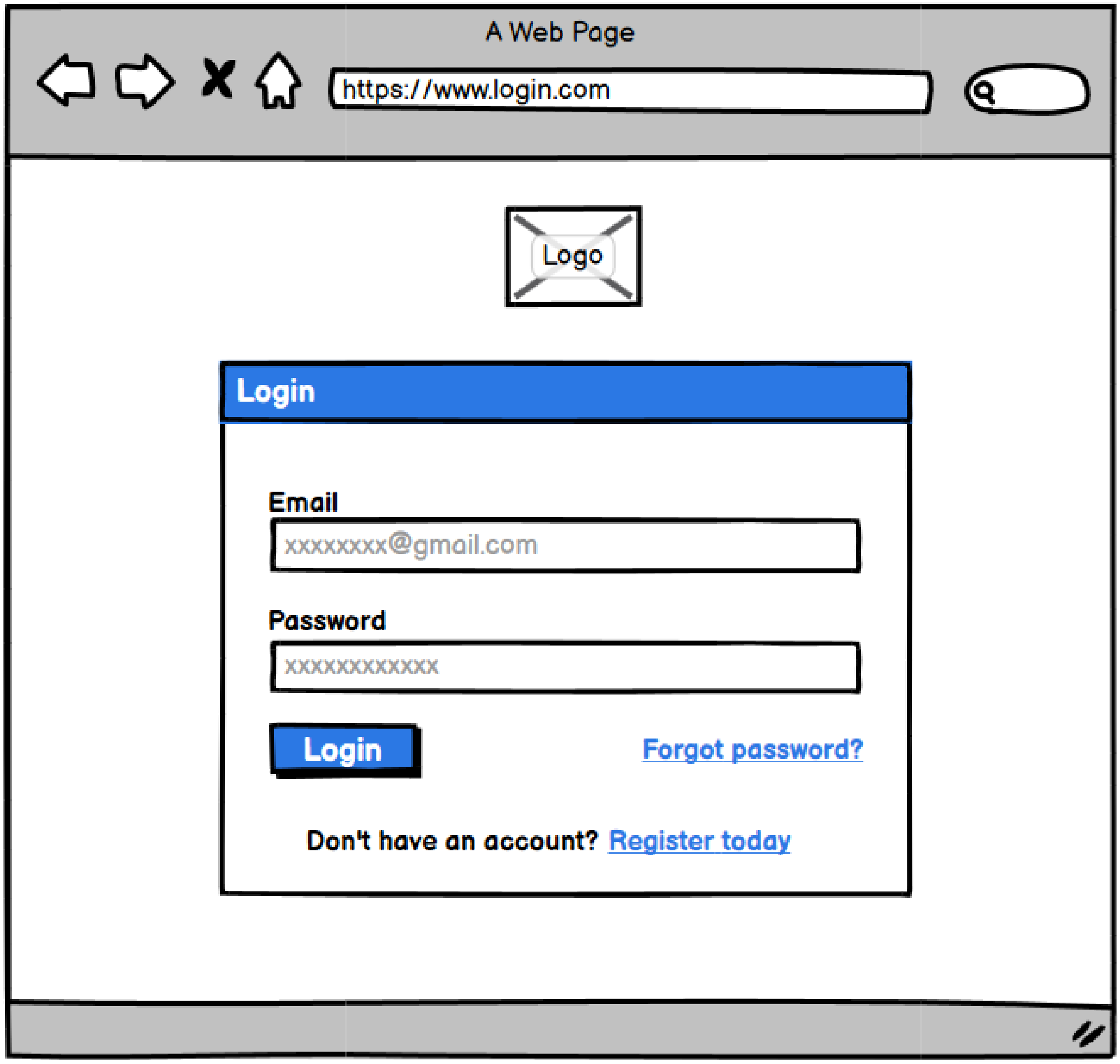


*Figure 35: Login Page*

### 3.4.5. Reset Password Page

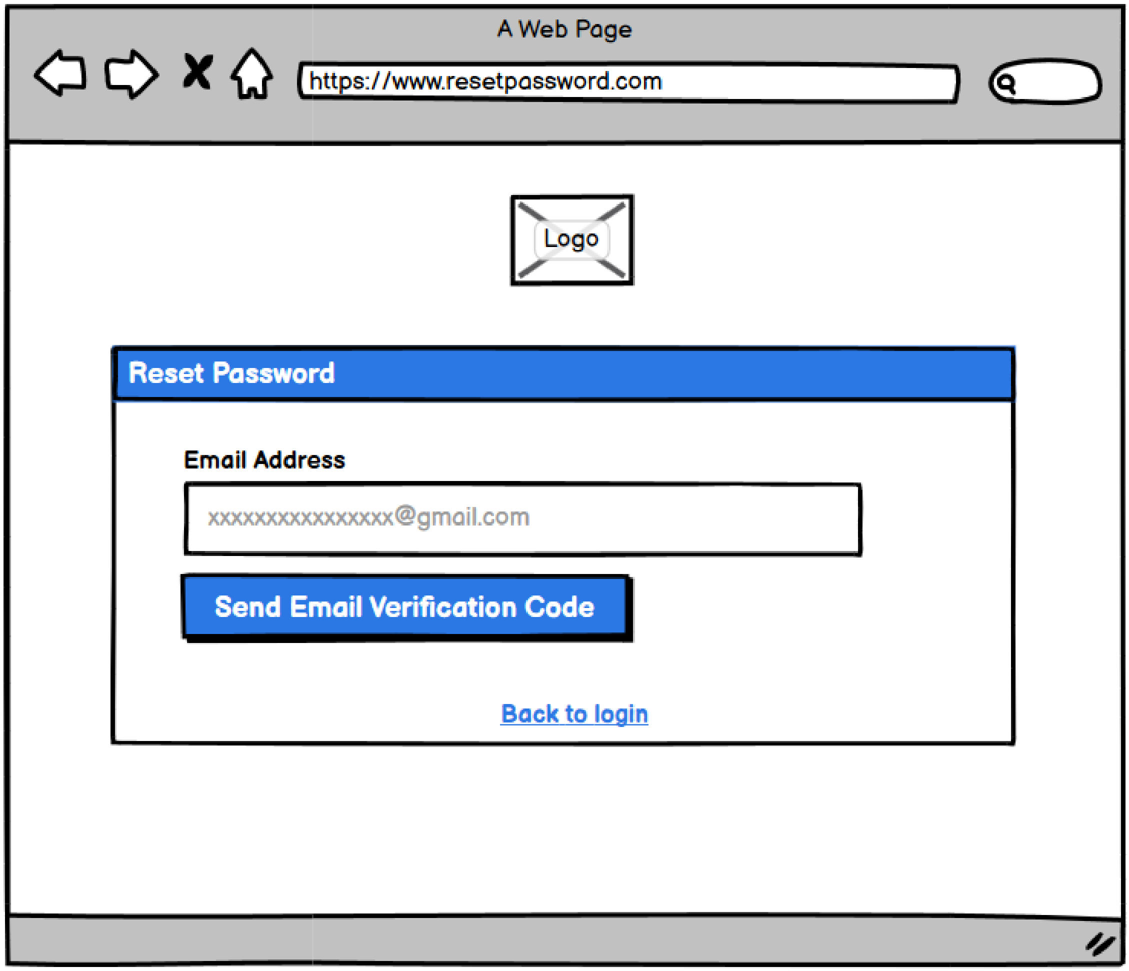

A Web Page
https://www.resetpassword.com
Logo
Reset Password
Email Address
xxxxxxxxxxxxxxxx@gmail.com
Send Email Verification Code
Back to login

*Figure 36: Reset Password Page*

### 3.4.6. Passenger Interface Page

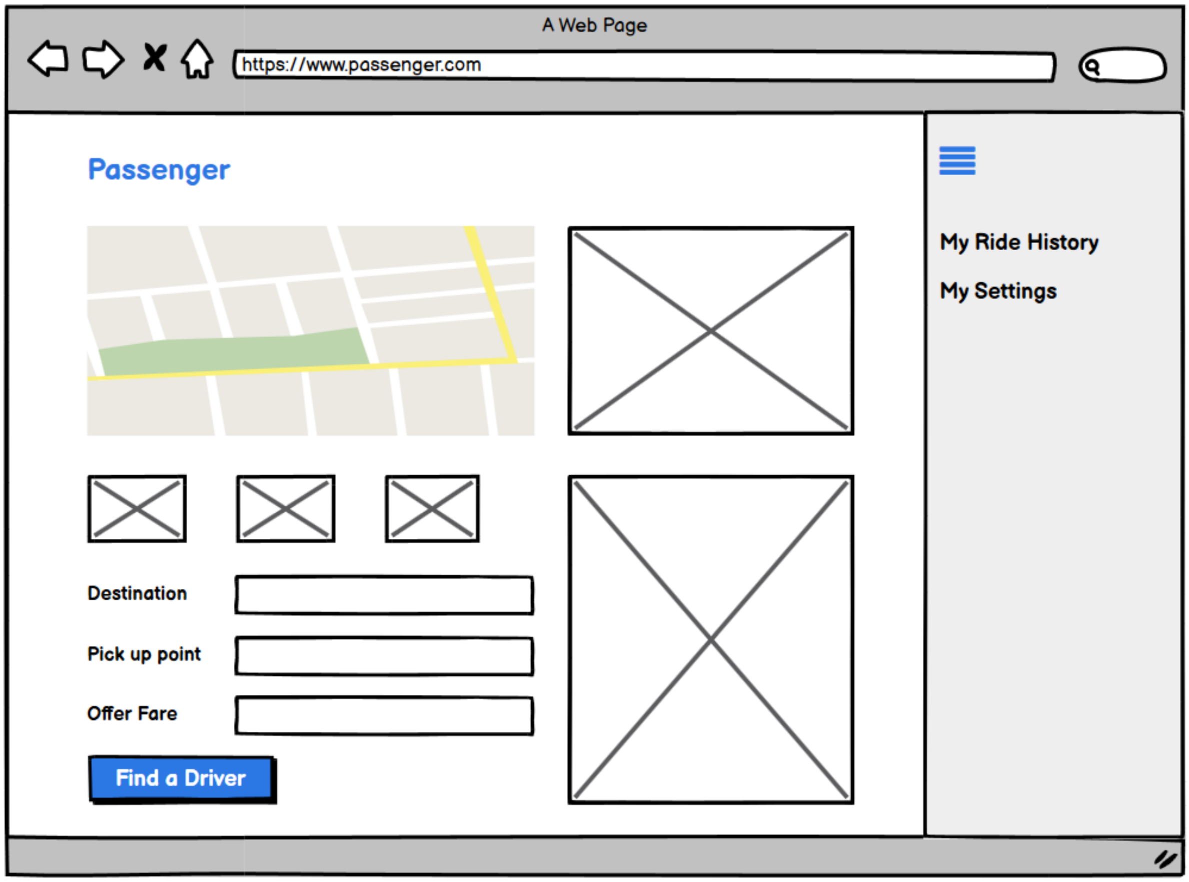


*Figure 37: Passenger Interface Page*

### 3.4.7. Driver Interface Page

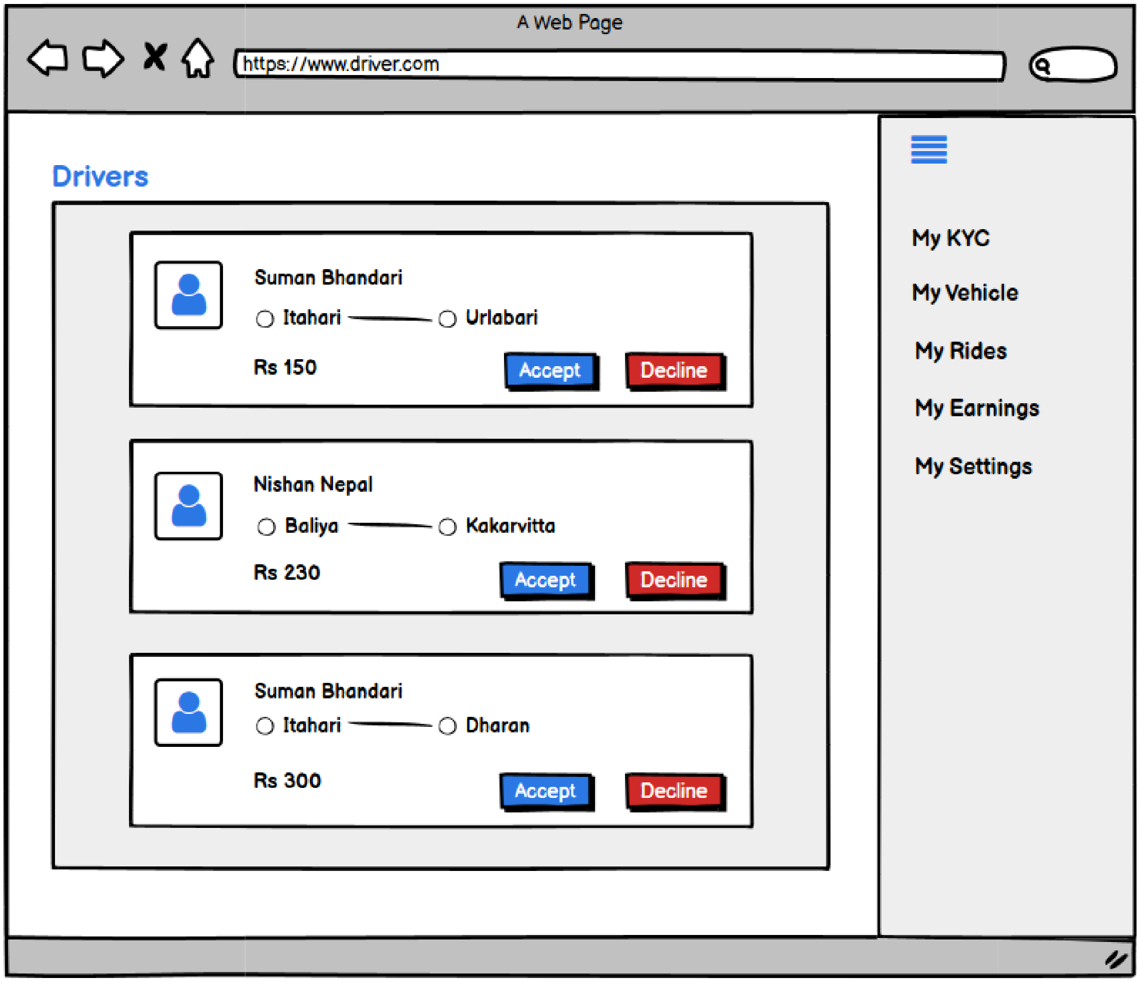


*Figure 38: Driver Interface Page*

## 3.5. Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is a visual representation of the relationships between entities in a database. It is used to model and design the structure of a database and is an essential tool for database designers and developers. An ERD is a graphical representation of the data relationships and attributes and is used to model the data requirements and design the database structure. The ERD is a powerful tool for designing and documenting databases, as it provides a clear, visual representation of the data structure and the relationships between different entities. It is a valuable tool for understanding the data requirements of an organization and for communicating those requirements to others (Peterson, 2022).

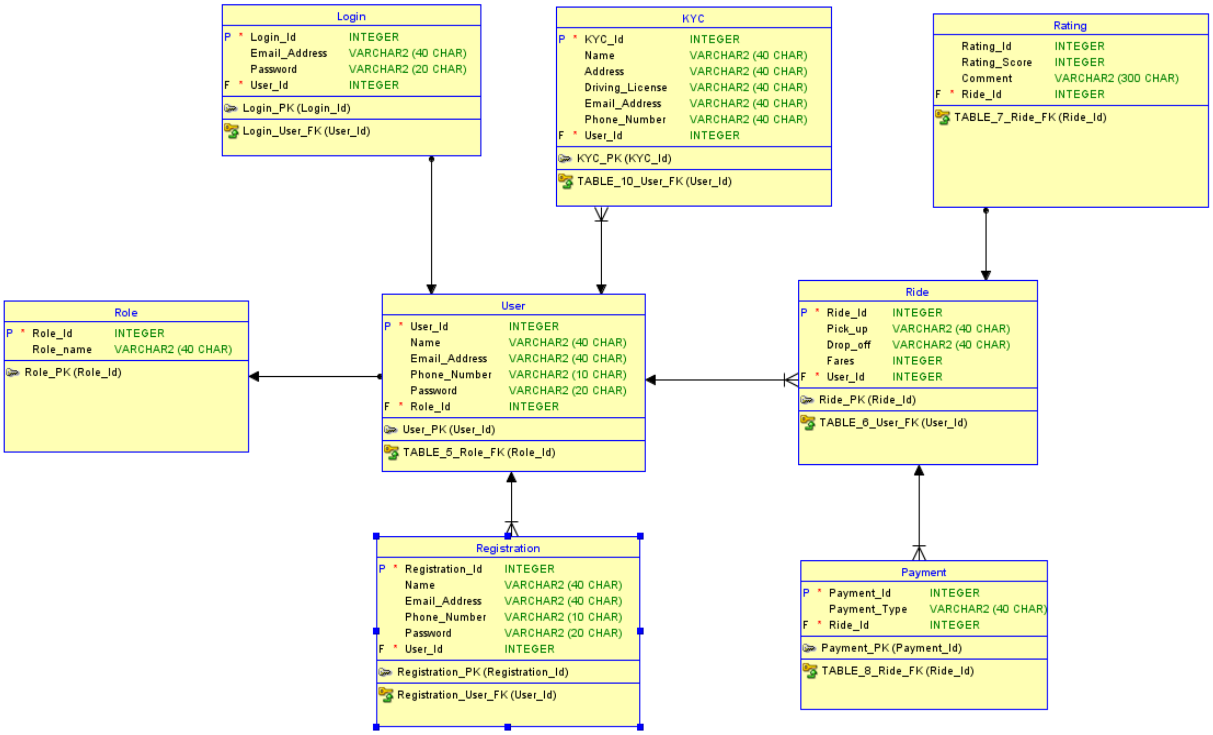

Login
P * Login_Id INTEGER
Email_Address VARCHAR2 (40 CHAR)
Password VARCHAR2 (20 CHAR)
F * User_Id INTEGER
Login_PK (Login_Id)
Login_User_FK (User_Id)
KYC
P * KYC_Id INTEGER
Name VARCHAR2 (40 CHAR)
Address VARCHAR2 (40 CHAR)
Driving_License VARCHAR2 (40 CHAR)
Email_Address VARCHAR2 (40 CHAR)
Phone_Number VARCHAR2 (40 CHAR)
F * User_Id INTEGER
KYC_PK (KYC_Id)
TABLE_10_User_FK (User_Id)
Rating
Rating_Id INTEGER
Rating_Score INTEGER
Comment VARCHAR2 (300 CHAR)
F * Ride_Id INTEGER
TABLE_7_Ride_FK (Ride_Id)
Role
P * Role_Id INTEGER
Role_name VARCHAR2 (40 CHAR)
Role_PK (Role_Id)
User
P * User_Id INTEGER
Name VARCHAR2 (40 CHAR)
Email_Address VARCHAR2 (40 CHAR)
Phone_Number VARCHAR2 (10 CHAR)
Password VARCHAR2 (20 CHAR)
F * Role_Id INTEGER
User_PK (User_Id)
TABLE_5_Role_FK (Role_Id)
Ride
P * Ride_Id INTEGER
Pick_up VARCHAR2 (40 CHAR)
Drop_off VARCHAR2 (40 CHAR)
Fares INTEGER
F * User_Id INTEGER
Ride_PK (Ride_Id)
TABLE_6_User_FK (User_Id)
Registration
P * Registration_Id INTEGER
Name VARCHAR2 (40 CHAR)
Email_Address VARCHAR2 (40 CHAR)
Phone_Number VARCHAR2 (10 CHAR)
Password VARCHAR2 (20 CHAR)
F * User_Id INTEGER
Registration_PK (Registration_Id)
Registration_User_FK (User_Id)
Payment
P * Payment_Id INTEGER
Payment_Type VARCHAR2 (40 CHAR)
F * Ride_Id INTEGER
Payment_PK (Payment_Id)
TABLE_8_Ride_FK (Ride_Id)

*Figure 39: Entity Relationship Diagram*

In the above ERD, the entities are:

- **User Entity:** It represents a user of the app, who can be an admin, driver, or passenger. The attributes for this entity includes name, email address, phone number, and password. The primary key for this entity is the "User_Id" attribute. The "Role_Id" attribute is a foreign key that references the "Role" entity.
- **Login Entity:** It represents a login session for a user. The attributes for this entity includes email address and password. The primary key for this entity is the "Login_Id" attribute. The "User_Id" attribute is a foreign key that references the "User" entity.
- **Registration Entity:** It represents the process of a user signing up for the app. The attributes for this entity includes name, email address, phone number, and password. The primary key for this entity is the "Registration _Id" attribute. The "User _Id" attribute is a foreign key that references the "User" entity.
- **Role Entity:** It represents the role of a user. The attributes for this entity include role name. The primary key for this entity is the "Role _Id" attribute. The "User_Id" attribute is a foreign key that references the "User" entity.
- **Ride Entity:** It represents a ride that is requested and completed through the app. The attributes for this entity includes pick up, drop off and fares. The primary key for this entity is the "Ride _Id" attribute. The "User_Id" attribute is a foreign key that references the "User" entity.
- **Rating Entity:** It represents a rating given by a passenger to a driver after a ride. The attributes for this entity includes rating score and comments. The primary key for this entity is the "Rating _Id" attribute. The "Ride _Id" attribute is a foreign key that references the "Ride" entity.
- **Payment Entity:** It represents a payment made by a passenger to a driver for a ride. The attributes for this entity include payment type. The primary key for this entity is the "Payment _Id" attribute. The "Ride _Id" attribute is a foreign key that references the "Ride" entity.

- **KYC Entity:** It represents the process of verifying a driver's identity. The attributes for this entity include name, address, email address, driving license and phone number. The primary key for this entity is the "KYC _Id" attribute. The "User _Id" attribute is a foreign key that references the "User” entity.

The relationship between these entities are as follows:

- A "User" can have many "Login" (one-to-many relationship).
- A "User" can have one "Registration" (one-to-one relationship).
- A "User" can be the driver for many "Rides" and can be a passenger for many "Rides" (many-to-many relationship). This relationship is represented by the "Driver" and "Passenger" entities, which have a one-to-many relationship with the "Ride" entity.
- A "Ride" can have one "Payment" (one-to-one relationship).
- A “Ride” can have one “Rating” (one-to-one relationship).
- A “User” can have one “KYC” record (one-to-one relationship).

### 3.6. System Architecture Diagram

An architecture diagram is a visual representation of the structure and components of a system or application. It is a high-level view of the overall design and organization of a system and is used to communicate the design and structure to stakeholders and team members. An architecture diagram is a helpful tool for comprehending a system's general design and is an important aspect of the design and development process. It is frequently used to record and explain the design to others since it gives a clear and straightforward way to comprehend the relationships and dependencies between different components of a system. An architecture diagram is an important tool for ensuring that a system is well-designed, scalable, and maintainable, and is an invaluable resource for software architects and developers (Wondershare EdrawMax, 2022).

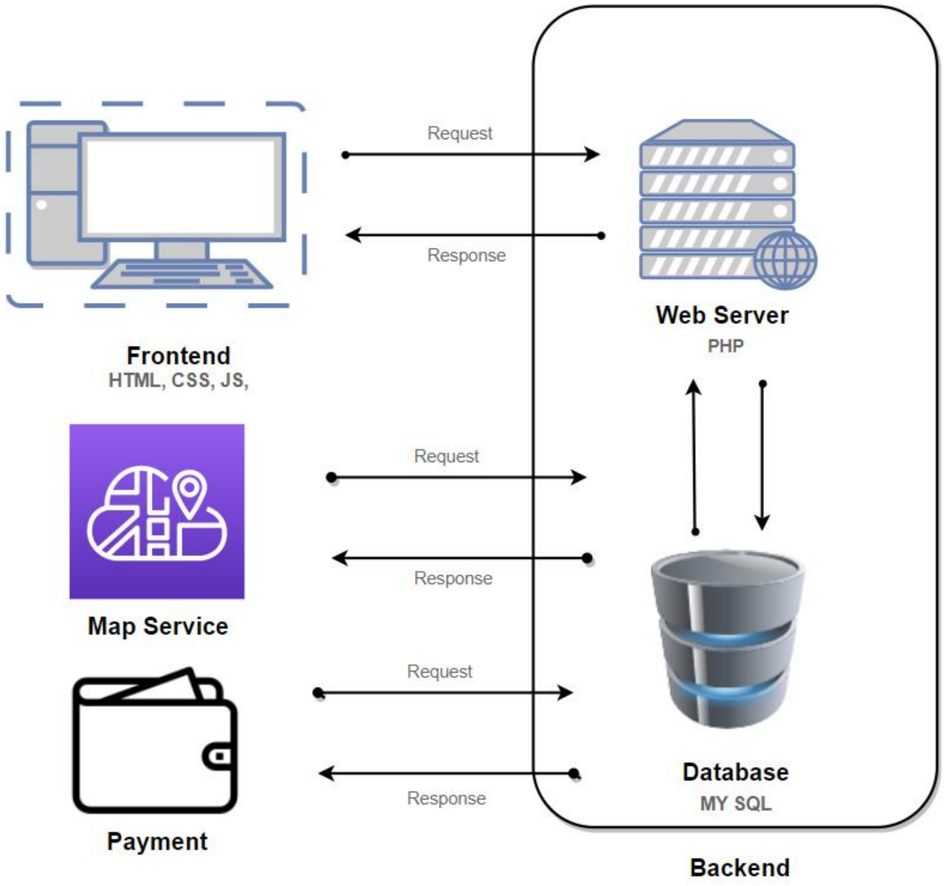


*Figure 40: Architecture Diagram*

### 3.7. Class Diagram

Class diagram is a type of Unified Modeling Language (UML) diagram that depicts the structure of a system by displaying the classes, properties, operations, and relationships between them. Class diagrams are used to represent the static aspects of a system, and they are highly useful for comprehending and explaining the design of object-oriented systems. In a class diagram, classes are represented as boxes with three sections: the top section lists the name of the class, the middle section lists the attributes of the class, and the bottom section lists the operations or methods of the class. Class relationships are shown by lines with arrowheads showing the direction of the relationship. Class diagrams can help to comprehend and express the design of a class (IBM, 2021).

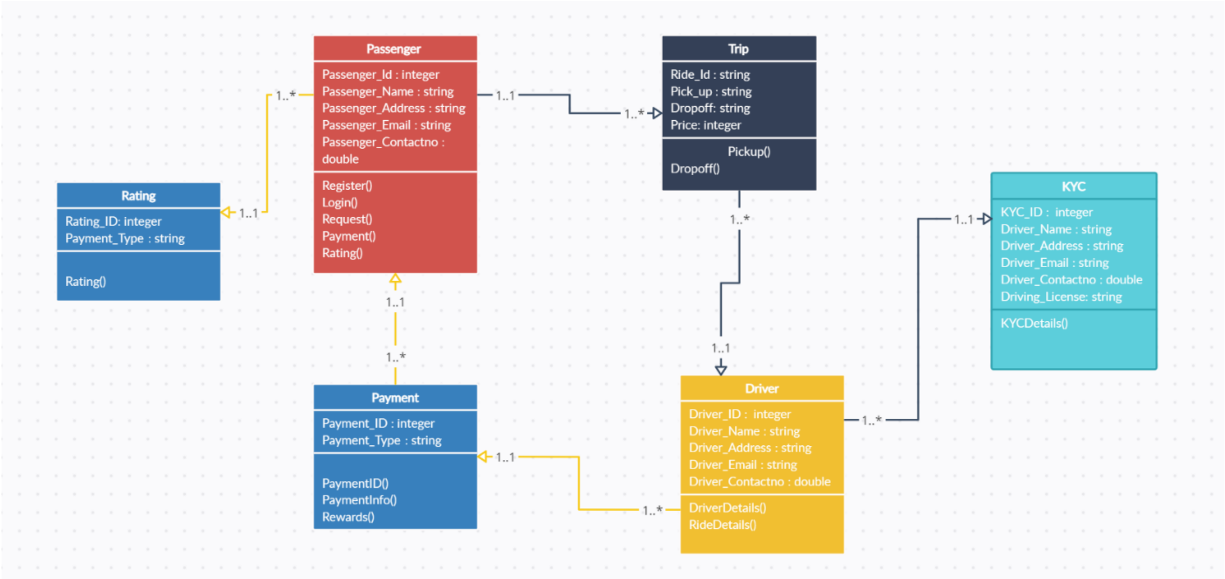


*Figure 41: Class Diagram*

## 3.8. Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a graphical depiction of data flow in a system. It is used to understand and convey how data is processed and altered as it goes through a system, and it is especially useful for recognizing and modeling the processes, data storage, and external entities involved. Data flows are depicted in a DFD as lines with arrowheads indicating the flow direction. Processes that indicate data transformation are represented as circles or boxes. Data stores, which indicate data storage, are shown as rectangles with double lines. External entities, such as data sources and destinations, are represented by rectangles with a single line. DFDs are commonly used in software development and systems analysis, and can be helpful in a variety of contexts, including business processes, information systems, and software applications  (Visual Paradigm, 2022).

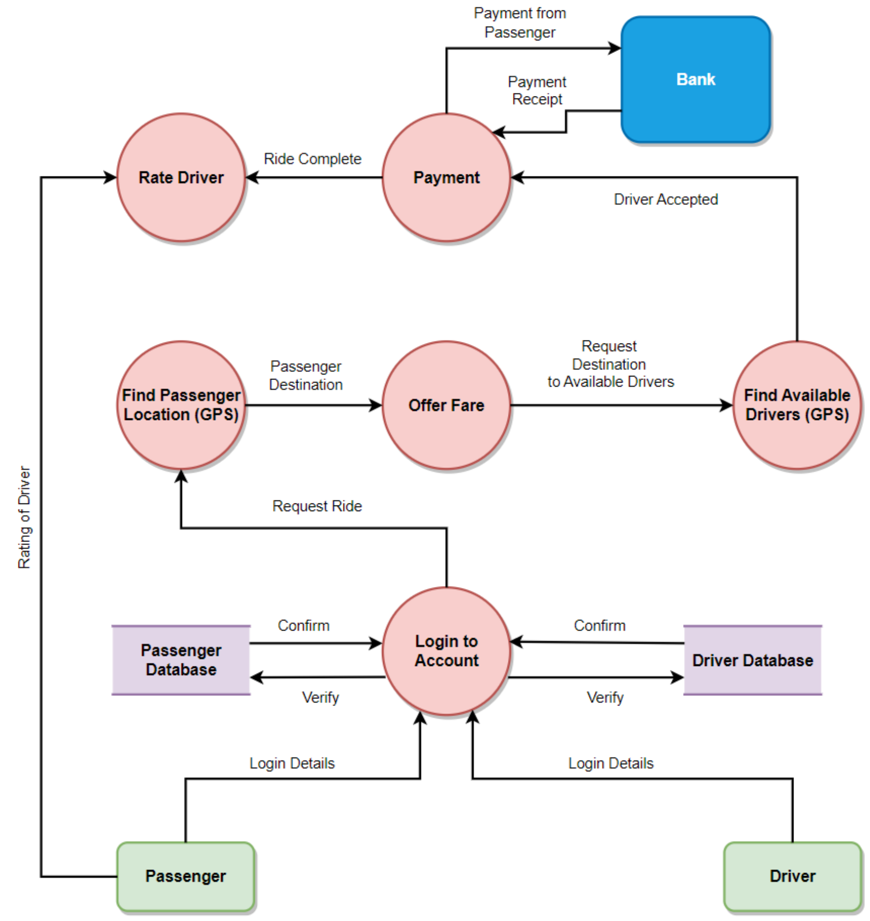


*Figure 42: Data Flow Diagram*

The above DFD in the context of a passenger and driver service could refer to an individual who is either a passenger or a driver using the service. The processes associated with a passenger and driver service may include:

- **Login account:** This process involves creating an account with the passenger and driver service, which may require providing personal information such as an email address and phone number.
- **Find passenger location:** This process involves using GPS technology to locate the passenger's current location so that a driver can pick them up.
- **Offer fare:** This process involves presenting the passenger with a fare estimate for their ride based on the distance and duration of the trip.
- **Find available driver:** This process involves using the passenger and driver service's platform to match a passenger with a nearby driver who is available to provide a ride.
- **Payment:** This process involves the passenger paying for the ride which may can be offline or digital payment.
- **Rate driver:** This process involves the passenger providing feedback on their experience with the driver, which may include rating the driver on a scale and writing a review. This feedback can be used by the passenger and driver service to improve the quality of service provided to passengers and to help drivers improve their performance.

## 3.9. Milestone Chart

A milestone chart is a graphical representation of the progress of a project or process over time. It is often comprised of a timeline indicating the planned and actual completion dates for a sequence of major milestones or events. Milestones are often visualized as points on a timeline, with the expected completion date displayed as a solid line and the actual completion date represented as a dotted line. The difference between the two lines reflects whether the milestone was accomplished on time or if a delay occurred. A milestone chart can be used to track project progress, identify potential delays or concerns, and communicate project status to stakeholders. It can be a useful tool for managing and tracking the progress of projects and processes, particularly when there are a large number of tasks or milestones involved (McAbee, 2021).

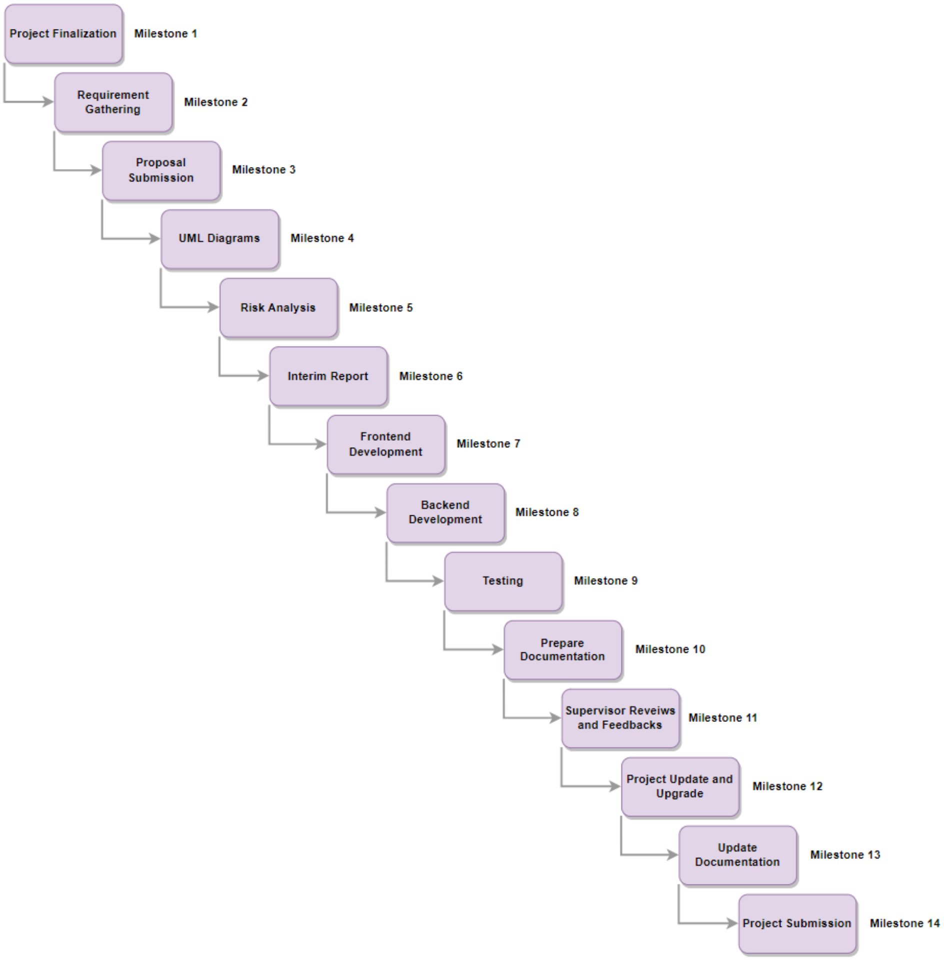


*Figure 43: Milestone Chart*

### 3.10. Work Breakdown Structure (WBS)

A Work Breakdown Structure (WBS) is a hierarchical illustration of the project work that must be accomplished. It is used to break down a project into smaller, more manageable units of tasks, as well as to organize and coordinate the project team's work. A work breakdown structure (WBS) generally consists of a top-level overview of the project, with lower-level tasks and activities stacked below. Each level of the work breakdown structure reflects a more precise breakdown of the work to be accomplished, with the lowest level reflecting individual tasks that must be completed. A work breakdown structure (WBS) can be an effective tool for organizing and directing the work of a project team, especially for big or complex projects with several tasks and activities (wrike, 2022).

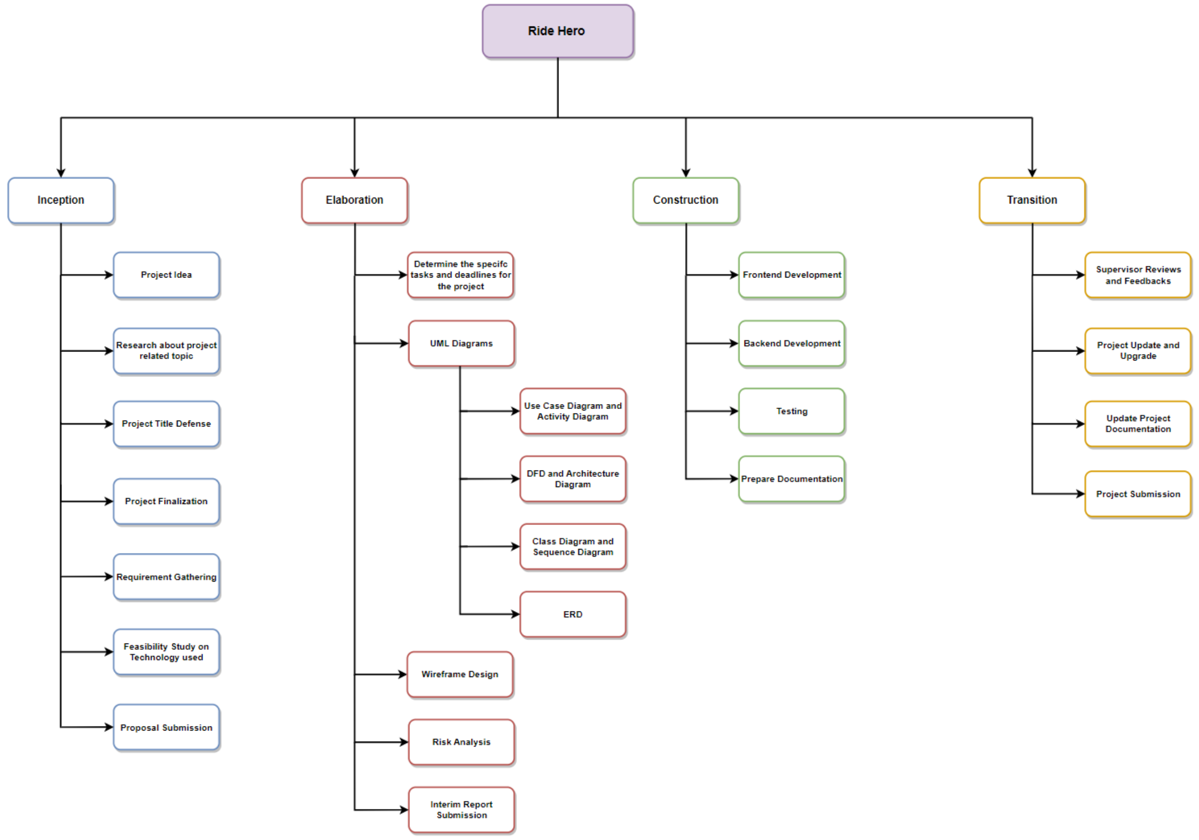
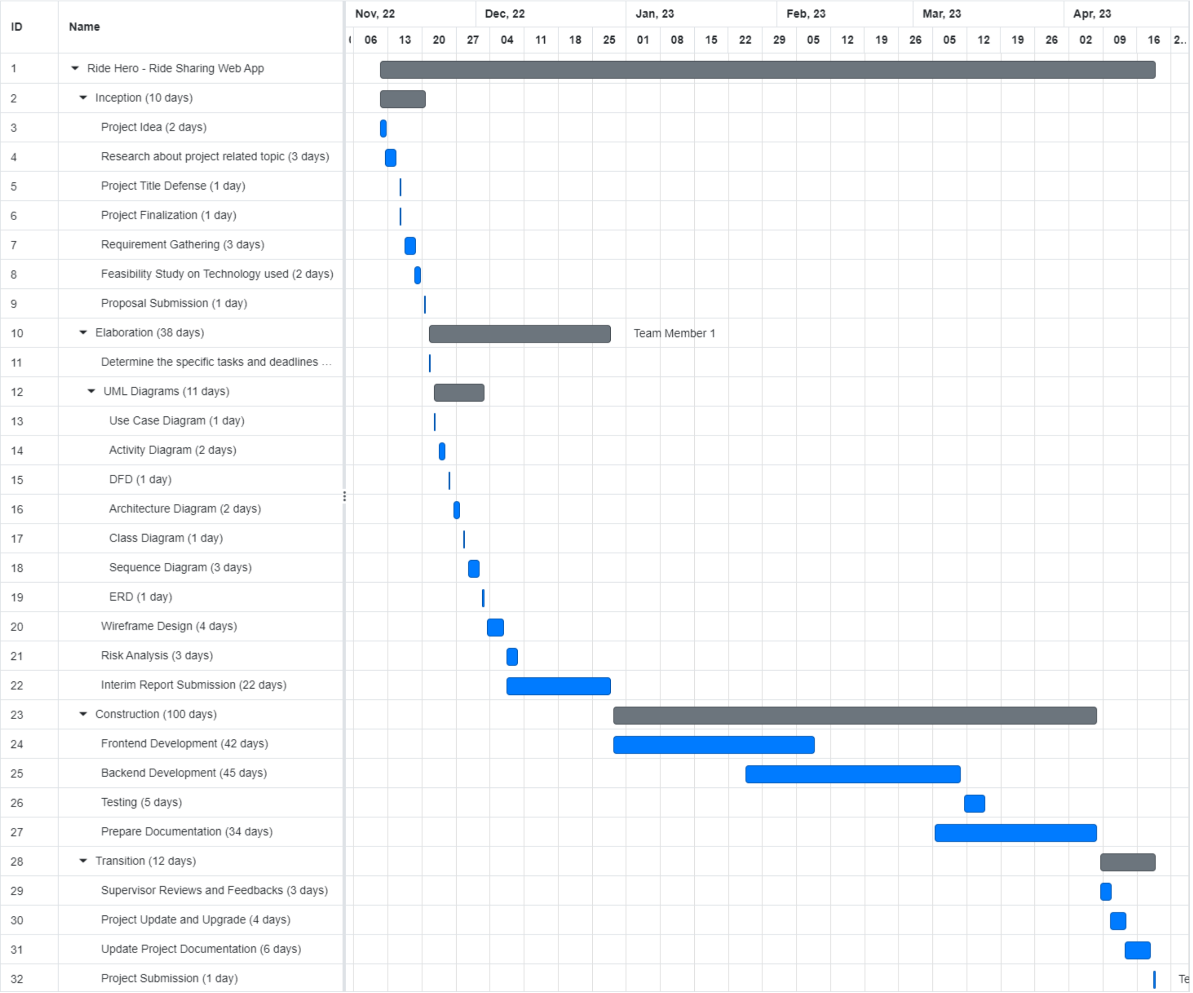

Ride Hero
Inception
Project Idea
Research about project related topic
Project Title Defense
Project Finalization
Requirement Gathering
Feasibility Study on Technology used
Proposal Submission
Elaboration
Determine the specifc tasks and deadlines for the project
UML Diagrams
Use Case Diagram and Activity Diagram
DFD and Architecture Diagram
Class Diagram and Sequence Diagram
ERD
Wireframe Design
Risk Analysis
Interim Report Submission
Construction
Frontend Development
Backend Development
Testing
Prepare Documentation
Transition
Supervisor Reviews and Feedbacks
Project Update and Upgrade
Update Project Documentation
Project Submission

*Figure 44: Work Breakdown Structure*

### 3.11. Gantt Chart

A Gantt chart is a graphical representation of the schedule and progress of a project or process. It is commonly used to illustrate the tasks and activities associated with a project, as well as to manage and coordinate the project team's activity. A Gantt chart is composed of a horizontal timeline that specifies the duration of each task or activity, as well as vertical bars that specify the start and end dates of each task. The bars can be color-coded to indicate the status of each task, such as whether it is in progress, complete, or behind schedule. Gantt charts are effective for organizing and managing the work of a project team, as well as tracking project progress. They can help to identify potential issues or roadblocks and can be used to communicate the status of a project to stakeholders (Shweta, 2022).



| ID | Name |
|---|---|
| 1 | ▾ Ride Hero - Ride Sharing Web App |
| 2 | ▾ Inception (10 days) |
| 3 | Project Idea (2 days) |
| 4 | Research about project related topic (3 days) |
| 5 | Project Title Defense (1 day) |
| 6 | Project Finalization (1 day) |
| 7 | Requirement Gathering (3 days) |
| 8 | Feasibility Study on Technology used (2 days) |
| 9 | Proposal Submission (1 day) |
| 10 | ▾ Elaboration (38 days) |
| 11 | Determine the specific tasks and deadlines … |
| 12 | ▾ UML Diagrams (11 days) |
| 13 | Use Case Diagram (1 day) |
| 14 | Activity Diagram (2 days) |
| 15 | DFD (1 day) |
| 16 | Architecture Diagram (2 days) |
| 17 | Class Diagram (1 day) |
| 18 | Sequence Diagram (3 days) |
| 19 | ERD (1 day) |
| 20 | Wireframe Design (4 days) |
| 21 | Risk Analysis (3 days) |
| 22 | Interim Report Submission (22 days) |
| 23 | ▾ Construction (100 days) |
| 24 | Frontend Development (42 days) |
| 25 | Backend Development (45 days) |
| 26 | Testing (5 days) |
| 27 | Prepare Documentation (34 days) |
| 28 | ▾ Transition (12 days) |
| 29 | Supervisor Reviews and Feedbacks (3 days) |
| 30 | Project Update and Upgrade (4 days) |
| 31 | Update Project Documentation (6 days) |
| 32 | Project Submission (1 day) |

Figure 45: Gantt Chart

### 3.12. Analysis of Progress

Analysis of progress is the process of evaluating the progress of a project or task over time in order to identify areas of success and areas for improvement. This analysis can be helpful for identifying any areas where progress is lagging and for taking corrective action to get the project back on track.

#### 3.12.1. Progress Table

In this section, the progress of individual task of the project is tracked and also determine the areas of tasks that need to be completed. The following table shows the progress of the task:

*Table 3: Progress Table*

| S.N. | Tasks | Progress | Progress % |
|---|---|---|---|
| 1. | Research about project topic | Completed | 100% |
| 2. | Requirement Gathering | Completed | 100% |
| 3. | Feasibility Study on Technology used | Completed | 100% |
| 4. | Proposal Submission | Completed | 100% |
| 5. | Use Case Diagram | Completed | 100% |
| 6. | Activity Diagram | Completed | 100% |
| 7. | Architecture Diagram | Completed | 100% |
| 8. | Sequence Diagram | Completed | 100% |
| 9. | Class Diagram | Completed | 100% |
| 10. | Data Flow Diagram (DFD) | Completed | 100% |
| 11. | Entity Relationship Diagram (ERD) | Completed | 100% |
| 12. | Wireframe Design | Partially Completed | 70% |
| 13. | Interim Report | Completed | 100% |
| 14. | Frontend Development | Incomplete | 0% |
| 15. | Backend Development | Incomplete | 0% |
| 16. | Testing | Incomplete | 0% |
| 17. | Review | Incomplete | 0% |
| 18. | Documentation | Incomplete | 0% |

### 3.12.2. Progress Review

The progress of the project so far has been very good, with important tasks such as research about the project topic, requirement gathering, feasibility study on the technology used, and proposal submission all completed successfully. The use case diagram, activity diagram, architecture diagram, sequence diagram, class diagram, data flow diagram (DFD), and entity relationship diagram (ERD) have also been completed, providing a clear understanding of the project's requirements and design.

Wireframe design has been partially completed, with progress made on defining the layout and functionality of the app. However, there is still work to be done in this area to ensure that the design is complete and meets the needs of the end users.

Overall, the project is on track and making good progress. However, it will be important to continue to monitor progress and make any necessary adjustments to stay on track and ensure the success of the project.

### 3.12.3. Progress Timeline

The following progress timeline table shows the tasks that have been completed and the tasks that are currently in progress:

*Table 4: Progress Timeline Table*

| S.N. | Tasks | Progress | Progress % |
|---|---|---|---|
| 1. | Research about project topic | Completed | 100% |
| 2. | Requirement Gathering | Completed | 100% |
| 3. | Feasibility Study on Technology used | Completed | 100% |
| 4. | Proposal Submission | Completed | 100% |
| 5. | Use Case Diagram | Completed | 100% |
| 6. | Activity Diagram | Completed | 100% |
| 7. | Architecture Diagram | Completed | 100% |
| 8. | Sequence Diagram | Completed | 100% |
| 9. | Class Diagram | Completed | 100% |
| 10. | Data Flow Diagram (DFD) | Completed | 100% |
| 11. | Entity Relationship Diagram (ERD) | Completed | 100% |
| 12. | Wireframe Design | Partially Completed | 70% |
| 13. | Interim Report | Completed | 100% |

### 3.12.4. Action Plan

An action plan is a detailed list of the tasks and activities that need to be completed in order to achieve goals of the project. It typically includes a timeline for when each task should be completed, a description of the tasks and the resources that are required to complete it. The task of the project is followed according to a new Gantt chart created for the interim report as reference for the tasks that must be accomplished and the deadlines for each activity. The action plan includes a description of each activity, the resources required to perform it, and the work's expected length. The Gantt chart can be used to track each task's progress and highlight areas where work is lagging. It is also a good idea to review and update the action plan and Gantt chart on a frequent basis to make sure that they remain precise and current as the project moves forward. I can effectively handle the project by following the action plan and referring to the Gantt chart.

# 4. Future Work

## 4.1. Phases to complete

There are typically four main phases remains to complete my project that includes:

- **Development:** In the development phase, I need to design and build the system. This includes frontend development and backend development by writing code and integrating various systems and technologies.
- **Testing:** In the testing phase, I need to conduct testing to ensure that my project is functioning as intended. This may include unit testing, integration testing, and user-acceptance testing.
- **Implementation:** In the implementation phase, I need to deploy my project and make available to the end users. This may involve installing the app, training users, and providing ongoing support. I need to ensure that I have a plan in place to handle any issues that may arise during this phase.
- **Documentation:** In the documentation phase, I need to create overall and proper documentation of my project. This includes everything regarding to the project. I need to ensure that all documentation is accurate and up-to-date.

## 5. Conclusion

In conclusion, the ride-sharing web app project has completed the elaboration phase, which means that the project goals and scope have been defined and the necessary resources have been identified. The remaining phases of the project, including development, testing, implementation, and documentation, will be critical for ensuring the success of the app. It is important to allocate sufficient time and resources for each of these phases in order to bring the app to completion and ensure that it meets the needs of the end users. By following a structured development process and conducting thorough testing, you can help ensure that the app is of high quality and performs as expected. The implementation and documentation phases will also be important for ensuring that the app is deployed and supported effectively. By following best practices and staying on track with the project plan, you can help ensure the success of the ride-sharing web app project.

## 6. References

AgileAlliance, 2022. *Extreme Programming (XP).* [Online] Available at: https://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1)
[Accessed 16 December 2022].

Alliance Software, 2022. *An Introduction To Software Development Methodologies.* [Online]
Available at: https://www.alliancesoftware.com.au/introduction-software-development-methodologies/#:~:text=What%20is%20a%20software%20development,non%20iterative%20kind%20of%20process.
[Accessed 16 December 2022].

BLYSTONE, D., 2022. *The Story of Uber.* [Online] Available at: https://www.investopedia.com/articles/personal-finance/111015/story-uber.asp
[Accessed 16 December 2022].

DAVIS, L., 2022. *Lyft vs. Uber: What's the Difference?.* [Online] Available at: https://www.investopedia.com/articles/personal-finance/010715/key-differences-between-uber-and-lyft.asp
[Accessed 16 December 2022].

engineeringforchange, 2022. *Ola Cabs.* [Online] Available at: https://www.engineeringforchange.org/solutions/product/ola-cabs/
[Accessed 16 December 2022].

GeeksforGeeks, 2022. *RUP and its Phases.* [Online] Available at: https://www.geeksforgeeks.org/rup-and-its-phases/#:~:text=Rational%20Unified%20Process%20(RUP)%20is,Method%20Composer%20(RMC)%20product.
[Accessed 16 December 2022].

GeeksforGeeks, 2022. *RUP and its Phases.* [Online] Available at: https://www.geeksforgeeks.org/rup-and-its-phases/#:~:text=Rational%20Unified%20Process%20(RUP)%20is,Method%20Composer%20(RMC)%20product.
[Accessed 16 December 2022].

GEOG 468: GIS Analysis and Design, 2022. *Use Cases in a Nutshell.* [Online] Available at: https://www.e-education.psu.edu/geog468/l8_p3.html [Accessed 18 December 2022].

IBM, 2021. *Class diagrams.* [Online] Available at: https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams [Accessed 18 December 2022].

Jackson, P., 2022. *What is PHP? Write your first PHP Program.* [Online] Available at: https://www.guru99.com/what-is-php-first-php-program.html [Accessed 16 December 2022].

javatpoint, 2022. *Learn JavaScript Tutorial.* [Online] Available at: https://www.javatpoint.com/javascript-tutorial [Accessed 16 December 2022].

javatpoint, 2022. *What is HTML.* [Online] Available at: https://www.javatpoint.com/what-is-html [Accessed 16 December 2022].

javatpoint, 2022. *What is Scrum?.* [Online] Available at: https://www.javatpoint.com/agile-scrum#:~:text=Scrum%20is%20a%20framework%20that,continuously%20shipping%20values%20to%20customers.
[Accessed 16 December 2022].

javatpoint, 2022. *XAMPP TUTORIAL.* [Online] Available at: https://www.javatpoint.com/xampp [Accessed 16 December 2022].

Laravel, 2022. *Introduction.* [Online] Available at: https://laravel.com/docs/4.2/introduction [Accessed 16 December 2022].

Lucidchart, 2022. *UML Activity Diagram Tutorial.* [Online] Available at: https://www.lucidchart.com/pages/uml-activity-diagram [Accessed 19 December 2022].

McAbee, J., 2021. *A Complete Guide to Milestone Charts.* [Online] Available at: https://www.wrike.com/blog/milestone-charts-guide/ [Accessed 18 December 2022].

Moore, L., 2022. *MySQL.* [Online] Available at: https://www.techtarget.com/searchoracle/definition/MySQL [Accessed 16 December 2022].

Moreau, E., 2022. *What Is the Google Chrome Browser?.* [Online] Available at: https://www.lifewire.com/what-is-google-chrome-4687647 [Accessed 16 December 2022].

Mustafeez, A. Z., 2022. *What is Visual Studio Code?.* [Online] Available at: https://www.educative.io/answers/what-is-visual-studio-code [Accessed 16 December 2022].

Pedamkar, P., 2022. *RUP Methodology.* [Online] Available at: https://www.educba.com/rational-unified-process/ [Accessed 16 December 2022].

Peterson, R., 2022. *Entity Relationship (ER) Diagram Model with DBMS Example.* [Online]
Available at: https://www.guru99.com/er-diagram-tutorial-dbms.html [Accessed 18 December 2022].

Positiwise, 2022. *What is Software Development Methodology?.* [Online] Available at: https://positiwise.com/blog/what-is-software-development-methodology/ [Accessed 16 December 2022].

ProductPlan, 2022. *Dynamic Systems Development Method (DSDM).* [Online] Available at: https://www.productplan.com/glossary/dynamic-systems-development-method/
[Accessed 16 December 2022].

ProductPlan, 2022. *Wireframe.* [Online] Available at: https://www.productplan.com/glossary/wireframe/
[Accessed 18 December 2022].

QA LEAD, 2022. *What Is DSDM (Dynamic Systems Development Method)?.* [Online] Available at: https://theqalead.com/development-devops-agile/dsdm-dynamic-systems-development-method/
[Accessed 16 December 2022].

Shweta, C. B., 2022. *What Is A Gantt Chart? The Ultimate Guide.* [Online] Available at: https://www.forbes.com/advisor/business/software/what-is-a-gantt-chart/
[Accessed 18 December 2022].

smartdraw, 2022. *Sequence Diagram.* [Online] Available at: https://www.smartdraw.com/sequence-diagram/
[Accessed 18 December 2022].

Stroud, F., 2021. *Microsoft Edge.* [Online] Available at: https://www.webopedia.com/definitions/microsoft-edge/
[Accessed 16 December 2022].

techopedia, 2018. *Cascading Style Sheet (CSS).* [Online] Available at: https://www.techopedia.com/definition/26268/cascading-style-sheet-css
[Accessed 16 December 2022].

tutorialspoint, 2022. *ReactJS - Overview.* [Online] Available at: https://www.tutorialspoint.com/reactjs/reactjs_overview.htm
[Accessed 16 December 2022].

Visual Paradigm, 2022. *What is Data Flow Diagram?.* [Online] Available at: https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-

flow-diagram/
[Accessed 18 December 2022].

Walker, A., 2022. *UML Use Case Diagram: Tutorial with EXAMPLE.* [Online] Available at: https://www.guru99.com/use-case-diagrams-example.html [Accessed 18 December 2022].

Wondershare EdrawMax, 2022. *Types of Architecture Diagram.* [Online] Available at: https://www.edrawmax.com/article/architecture-diagram-types.html [Accessed 18 December 2022].

wrike, 2022. *What Is Work Breakdown Structure in Project Management?.* [Online] Available at: https://www.wrike.com/project-management-guide/faq/what-is-work-breakdown-structure-in-project-management/
[Accessed 18 December 2022].

# 7. Appendix

## 7.1. Appendix 1: Sequence Diagram

A sequence diagram is a graphical representation that demonstrates the interactions of objects or components in a system through time. It shows the chain of messages or events exchanged between objects, as well as the order in which they occur. In software development, sequence diagrams are frequently used to visualize the flow of messages and data between objects or components in a system, as well as to facilitate in the design and debugging of complicated systems. They can be used to mimic a wide range of interactions, from simple request-response interactions to more sophisticated interactions involving several objects or components. Sequence diagrams can help to understand how a system works and identify potential problems or blockages. When working with complex systems including several objects or components interacting in detailed ways, sequence diagrams can be especially beneficial (smartdraw, 2022).

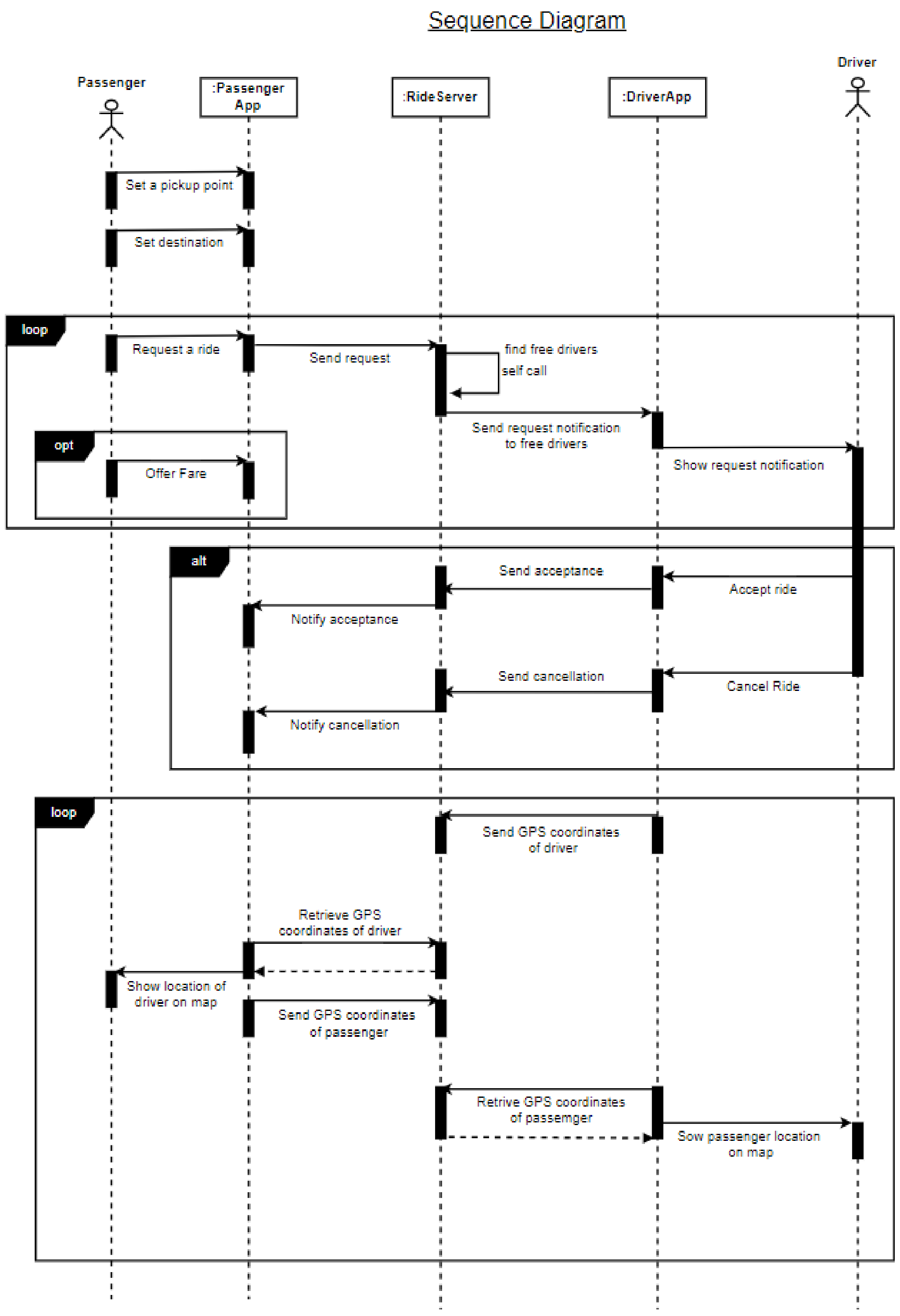


*Figure 46: System Sequence Diagram*

## 7.2. Activity Diagram

An activity diagram is a graphical representation of the flow of activities or actions within a system. It is typically used to model the behavior of a system or process and can be used to represent the flow of control in a software system, the processing of a request or order, or the steps involved in a business process. Activity diagrams are created using a set of standardized symbols and notation, which are connected by arrows to indicate the flow of control from one step to the next. Activity diagrams can be used to model both simple and complex processes and are often used in conjunction with other modeling techniques, such as sequence diagrams and state diagrams, to provide a more complete understanding of a system or process. The activity diagram is a useful tool for visualizing and documenting the flow of activities within a system, and can help to identify potential bottlenecks, inefficiencies, or opportunities for improvement (Lucidchart, 2022).

### 7.2.1. Passenger Activity Diagram

**a) Passenger Register Activity Diagram**

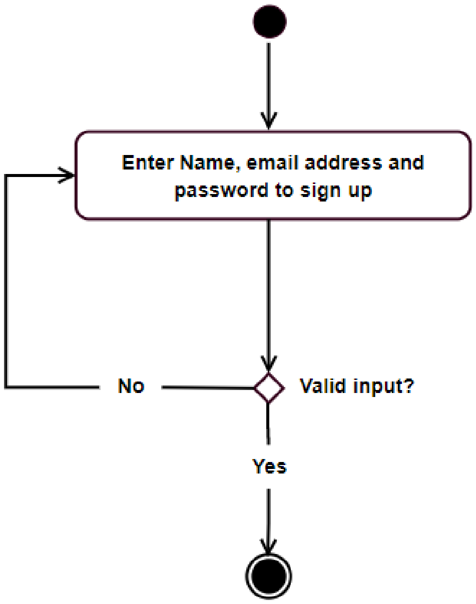


*Figure 47: Passenger Register Activity Diagram*

## b) Passenger Login Activity Diagram

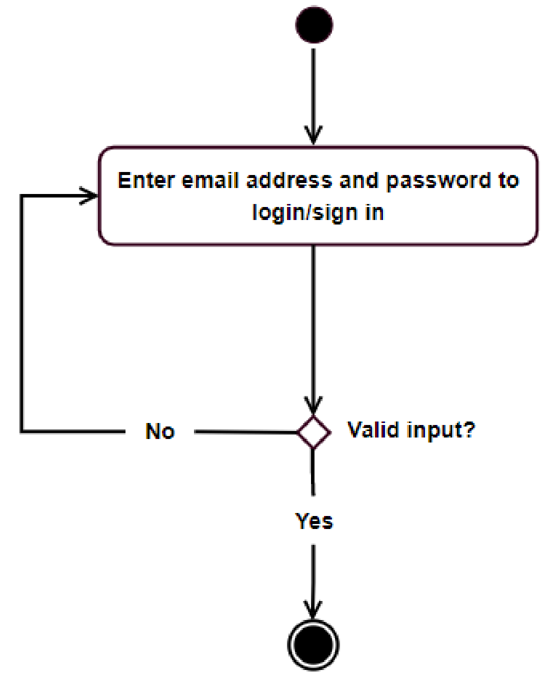


*Figure 48: Passenger Login Activity Diagram*

### c) Passenger Ride Request Activity Diagram

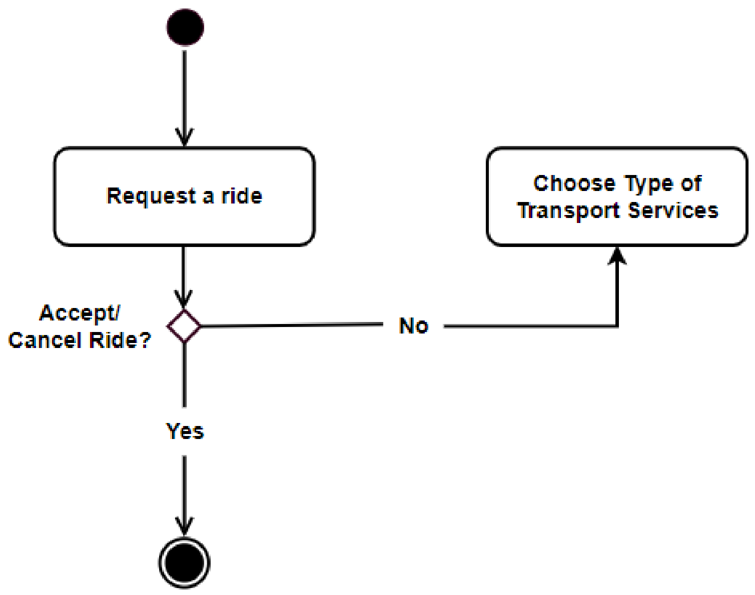


*Figure 49: Register Ride Request Activity Diagram*

### d) Passenger Location Detection Activity Diagram

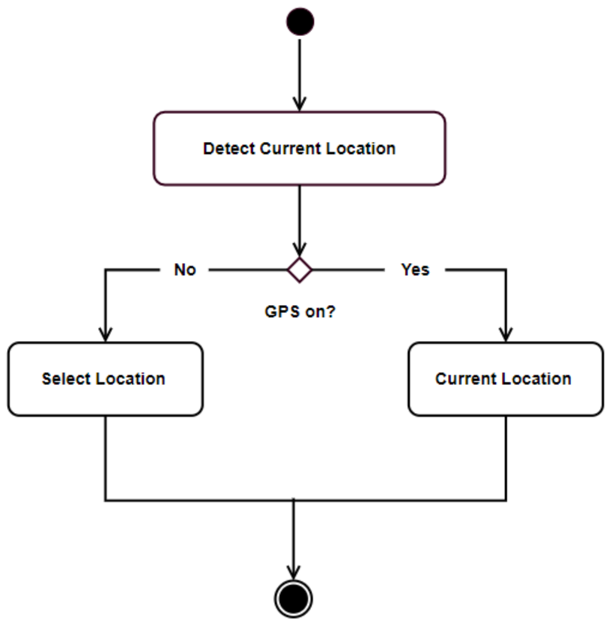


*Figure 50: Passenger Location Detection Activity Diagram*

## e) Final Passenger Activity Diagram

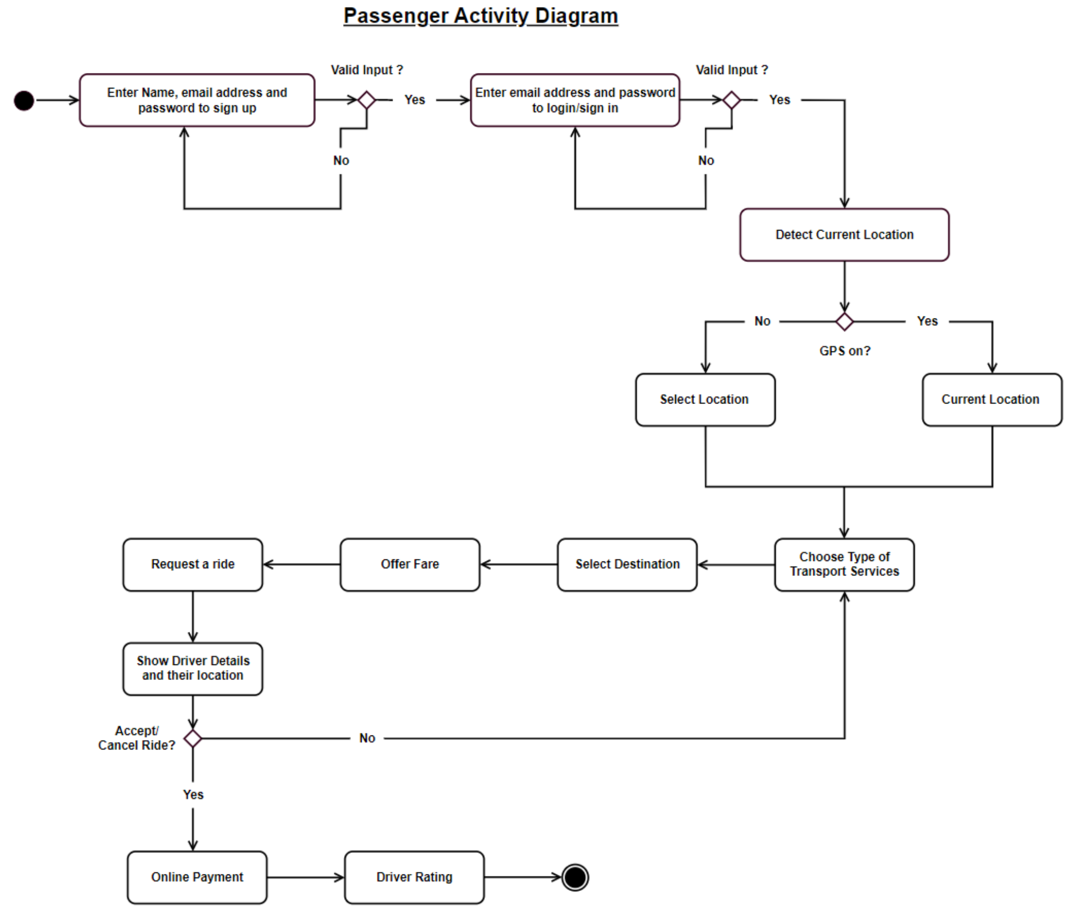


*Figure 51: Final Passenger Activity Diagram*

### 7.2.2. Driver Activity Diagram

#### a) Driver Register Activity Diagram

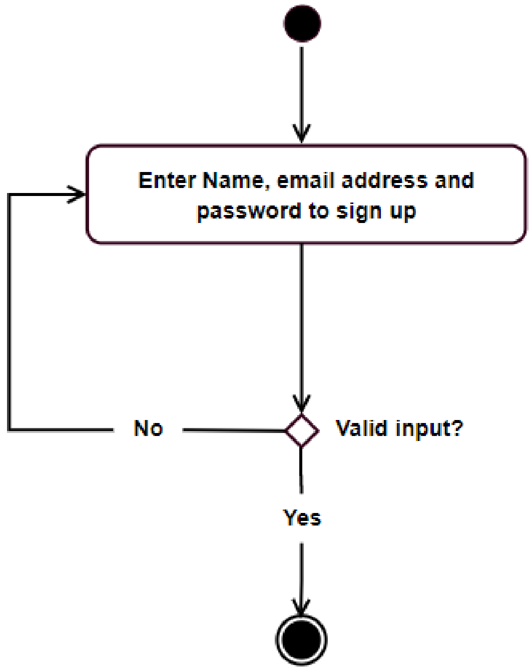


*Figure 52: Driver Register Activity Diagram*

## b) Driver Login Activity Diagram

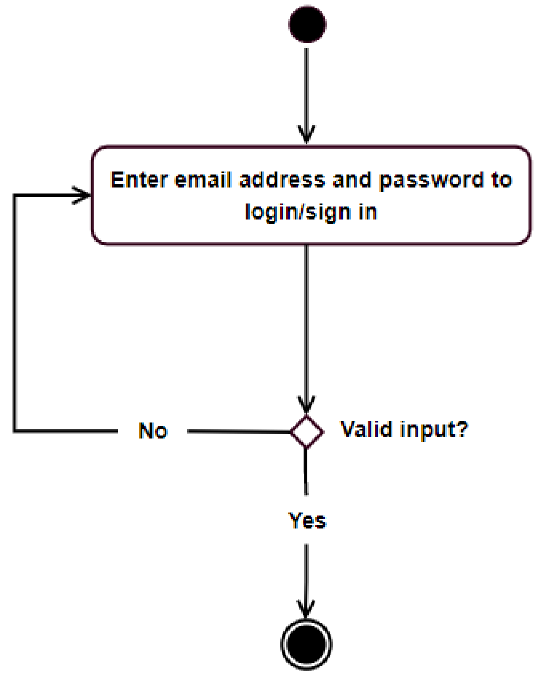


*Figure 53: Driver Login Activity Diagram*

### c) Driver KYC Verification Activity Diagram

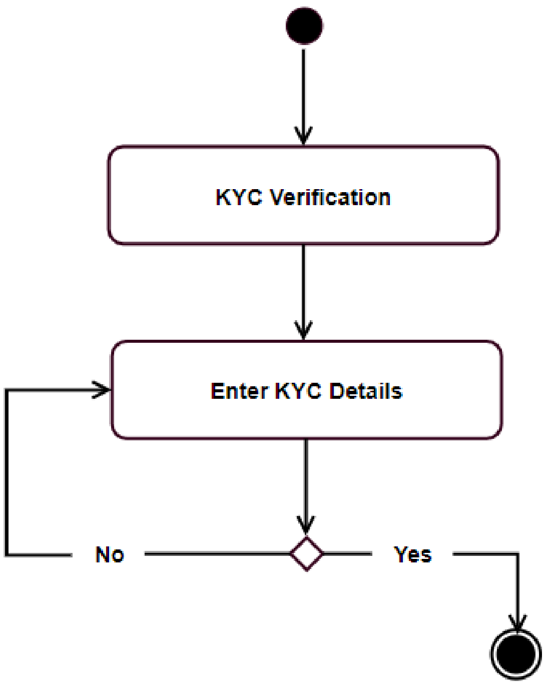


*Figure 54: Driver KYC Verification Activity Diagram*

### d) Driver Manage Ride Activity Diagram

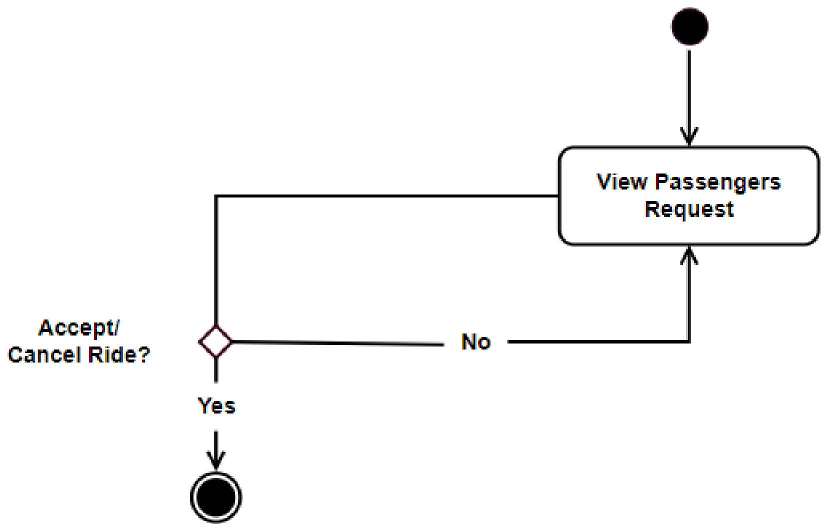


*Figure 55: Driver Manage Ride Activity Diagram*

## e) Final Driver Activity Diagram

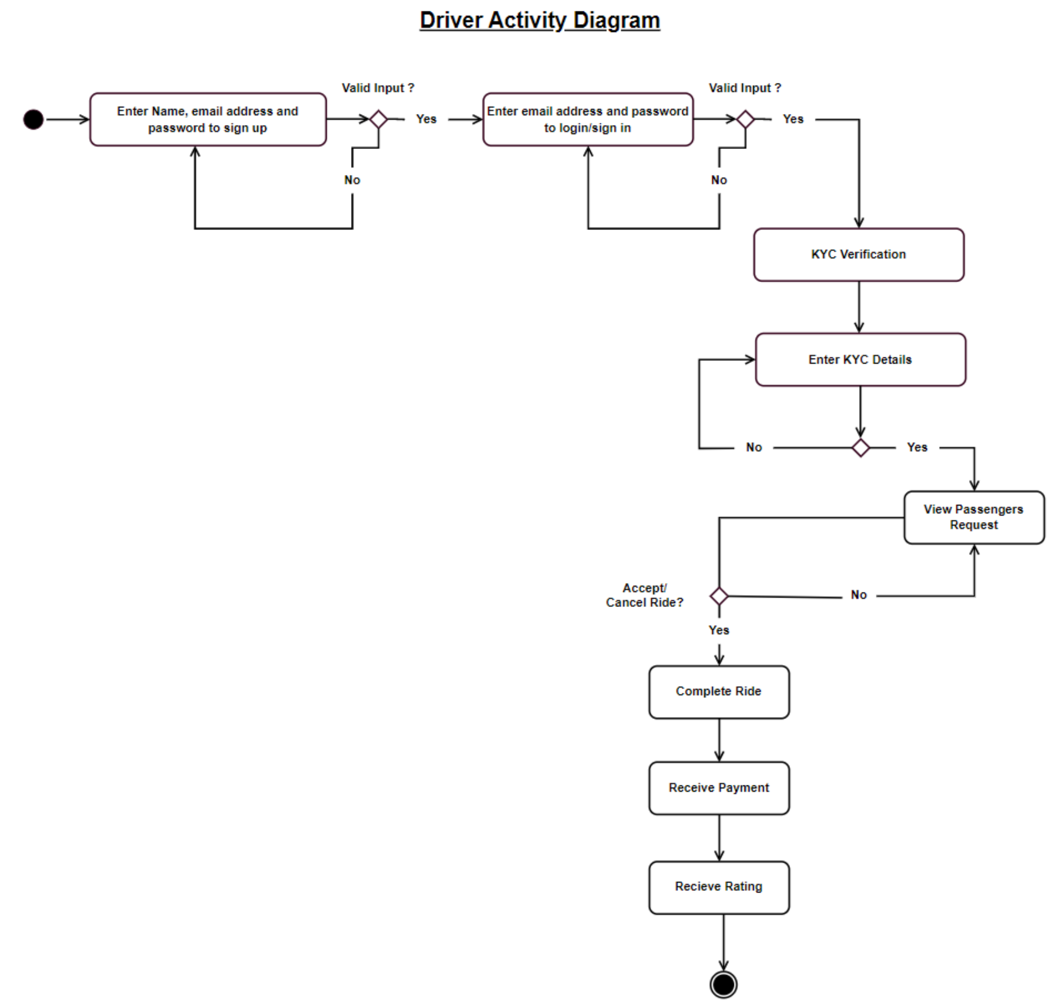


*Figure 56: Final Driver Activity Diagram*