CHALO RIDE PE

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2024-25)

Submitted by

MEENAKSHI BHARADWAJ
(2300290140102)
MURARI KUMAR JHA
(2300290140105)
NIKHIL CHAUDHARY
(2300290140107)
PAWAN GANGWAR
(2300290140113)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Monika Kansal
(Assistant Professor)

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(DECEMBER 2024)

CERTIFICATE

Certified that **Meenakshi Bharadwaj (2300290140102), Murari Kumar Jha (2300290140105), Nikhil Chaudhary (2300290140107),** and **Pawan Gangwar (2300290140113)** have carried out the project work having "**ChaloRidePe**" (**Mini-Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Monika Kansal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ChaloRidePe
(Meenakshi Bharadwaj)
(Murari Kumar Jha)
(Nikhil Chaudhary)
(Pawan Gangwar)

ABSTRACT

Today's urban settings demand dependable and effective transportation. As the need for smooth travel options in crowded places grows, ChaloRidePe shows up as a vibrant and easy-to-use ride-hailing service. Whether you're traveling to work, attending an important meeting, or exploring new areas of the city, it enables passengers to communicate in real-time with local drivers, guaranteeing a seamless and stress-free journey.

ChaloRidePe offers real-time transportation scheduling, precise fare predictions, and safe cashless payment processing. It provides a smooth and safe experience because of its user-friendly interface, which is improved with safety measures and robust validations to guarantee legitimate users.

Without the hassles of conventional transportation alternatives, users may easily book trips and complete transactions. ChaloRidePe is a useful tool for traversing today's hectic cities because of its emphasis on client comfort, safety, and efficiency. It offers a dependable, easily available alternative to urban commuting, perfect for both impromptu travel and routine commutes.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Monika Kansal (Assistant Professor)** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Meenakshi Bharadwaj

Murari Kumar Jha

Nikhil Chaudhary

Pawan Gangwar

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

With the growing need for budget-friendly and user-friendly ride-booking apps, ChaloRidePe seeks to offer a smooth solution for users to book rides effortlessly. It utilizes advanced APIs such as Here Maps, Clerk and Stripe to provide real-time location services, authenticated users and secure payment options.

ChaloRidePe is a mobile-based application that is available on mobile platforms such as Android or iOS. This application is developed with a clear idea of providing users a smooth application to book their rides effortlessly and conveniently.

1.2 Objectives

The core objective of **ChaloRidePe** is to develop a comprehensive mobile application that connects drivers and users for convenient rides, while ensuring the platform is simple, safe, and effective.

The primary objectives of the ride-booking application "ChaloRidePe" are:

1. **To design a user-friendly interface for ride booking:** The application interface is designed simple and straightforward to ensure the convenience of users of all age. Easy-to-use interface with vibrant color palette makes it more engaging and easier for users to adapt to it.

2. **To ensure proper ride matching passengers with available drivers:** The application will match available drivers with passengers who need a ride, all nearby drivers in a certain area around user are searched and fetched into a list of drivers that the user may select and proceed for ride-booking.

3. **To ensure secure user authentication and payment processing:** The platform integrated secure and convenient payment methods, allowing users to pay for their rides using a variety of payment options such as credit/debit cards, etc. Features

like profile verification, and driver selection based on matched requirements will ensure that the platform is safe and reliable for both parties.

4. **To implement real-time location-based services:** The real-time current location of user is accessed by the application after granted location access and then located on the map available in the application, ensuring transparency and consistent location access throughout.

5. **To facilitate features such as ride history and user profiles:** Features such as ride history helps the passenger to track their ride history with relevant information such as ride date, driver name, destination location, etc. It promotes transparency. Features such as user profiles helps the application to keep record of legitimate users and make them identifiable by the drivers.

1.3 Scope

The current application focuses solely on **user-centric** functionalities, with future plans to incorporate driver-side capabilities.

The features are designed to meet user requirements, providing them with an easy and efficient ride-booking experience, the key components include:

1. **User Registration and Authentication:** Passengers will be able to register and create an account using their personal details. The platform will offer a secure login system, ensuring the protection of user data.

2. **Profile Management:** Users will have access to manage their profiles, including personal information, ride history, and details about their location and drivers nearby.

3. **Ride Booking:** Passengers will be able to search for available rides and book them. Drivers can view their ride requests, accept or reject bookings, and manage their schedules.

4. **Real-Time Location Access:** Live location access functionality will allow passengers to get the destination as well as user location on the map in real time, improving safety and providing transparency for users. It will also help users to track the drivers nearby on the map.

5. **Payment Integration:** The platform will offer various payment methods, making transactions simple and secure. Users can choose their preferred method for paying for rides.

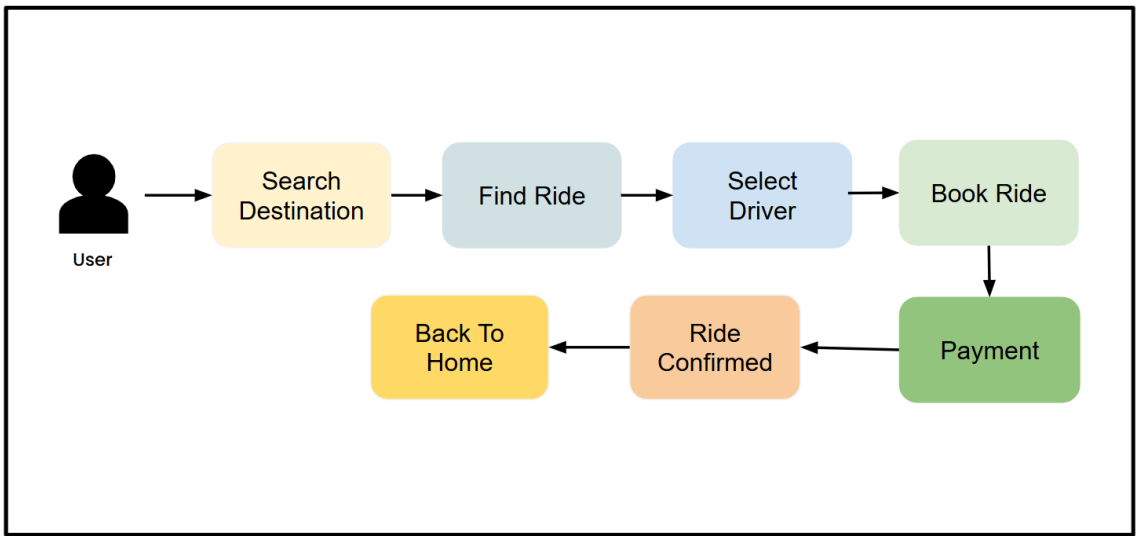


Fig. 1.1 ChaloRidePe Workflow

1.4 Features

ChaloRidePe offers a variety of features aimed at providing an excellent user experience primarily for focused users i.e., passengers:

1. **User-Friendly Interface:** The application has been designed with a clean and intuitive interface, making it easy for users to navigate and find what they need quickly. Whether on android or iOS, users will be able to access all features with minimal effort.

2. **Simple Ride Booking Process:** The feature allows users to book a ride. The process of booking a ride is straightforward, with users able to select their pick-up and drop-off locations, and confirm the booking with just a few clicks.

3. **Real-Time Location Access:** Live location access will be used to show the exact location of the vehicle to both, the nearby drivers and the passengers. This feature will provide real-time updates, allowing users to know when to expect their arrival.

4. **Payment Options:** The platform will integrate secure and easy payment options, including credit cards, debit cards, etc, so that users can pay for their rides in a way that suits them best.

CHAPTER 2

FEASIBILITY STUDY

A feasibility study is an essential part of any project as it helps assess whether the project is viable and whether it can be successfully executed within the given constraints. For ChaloRidePe, the feasibility study is divided into four key categories:

- Economical Feasibility Study
- Technical Feasibility Study
- Operational Feasibility Study
- Behavioral Feasibility Study

Each of these aspects will ensure that the project is not only practical but also sustainable in the long term.

2.1 Economical Feasibility Study

Economical feasibility study evaluates whether the project is financially viable and whether it can be developed and maintained within the available budget.

- **Cost Estimation:** The main costs for developing the initial version of ChaloRidePe will include application development, debugging via Expo Mobile App, free APIs, and marketing through social media. Additionally, there will be ongoing costs related to platform maintenance, user support, and software updates in later versions.
- **Profitability:** By ensuring a reasonable pricing model and offering multiple ways to generate revenue such as in-app advertisements, ChaloRidePe can be profitable. The app will also scale as the user base increases, which can lead to higher profits over time.
- **Budgeting:** Initial version of ChaloRidePe has a budget limiting to utilize only the freely available services. For later versions of application, a detailed budget plan will be created to ensure that the project remains financially sound. A careful estimate of development costs, marketing expenses, and expected revenues will guide the project's financial strategy.

Overall, the economic feasibility of ChaloRidePe suggests that it is a financially viable project, provided it maintains a strong user base and operates efficiently.

2.2 Technical Feasibility Study

Technical feasibility refers to the technology and infrastructure required to develop, launch, and maintain the ChaloRidePe platform.

- **Technology Stack:** The project will use proven and scalable technologies. The frontend will be developed using React Native Expo, a widely used framework for building interactive user interfaces. The backend will be powered by Neon Serverless database. NeonDB PostgreSQL will be used for storing data due to its flexibility and scalability, making it ideal for handling the dynamic nature of ride data.
- **API Integration:** For real-time location access, Here Maps API will be integrated. This API will allow the platform to display live locations of vehicles, calculate routes, and provide directions for both passengers and drivers.
- **Security:** The platform will implement robust security features, including google OAuth authentication for secure user authentication, and secure payment gateways for financial transactions.
- **Scalability:** The app's architecture is designed to be scalable. As the number of users grows, the platform can handle more rides without compromising on performance.
- **Development Tools:** Tools like Visual Studio Code for coding and Postman for testing APIs will be used throughout the development process to ensure efficiency and quality.

The technical feasibility of ChaloRidePe is high, as the required technology is available, scalable, and secure, ensuring the app can be built and maintained with minimal risk.

2.3 Operational Feasibility Study

Operational feasibility assesses whether the system can be efficiently operated on a day-to-day basis by users.

- **User Experience:** The platform will be designed to be intuitive and easy to use for users. Clear navigation, straightforward booking processes, and real-time location will make the system user-friendly.
- **Accessibility:** The platform is lightweight and supports the application with Android and iOS platforms.
- **Usability:** The application is usable for target audience and it very intuitive in its interface which makes it operate-able by people of all ages.

The application is user-friendly and addresses the needs of its target audience. Features like ride booking, payment, and ride history are intuitive, making it easy for users to adapt to the system.

2.4 Behavioral Feasibility Study

Behavioral feasibility examines whether users will accept and adopt the ChaloRidePe platform and whether it will be welcomed by the community.

- **User Acceptance:** Since people are becoming more conscious of minimal and user-friendly interfaces, there is a strong potential for acceptance of a ride-sharing platform. By providing a reliable, affordable transportation option.
- **Trust and Security:** Building trust is crucial for any platform that involves shared transportation. By implementing user verification, and secure payment methods, the platform will ensure that both drivers and passengers feel safe and confident in using the service. Proper verification will further engage more users and will build trust with enhanced security.
- **Community Support:** ChaloRidePe can benefit from word-of-mouth marketing, as satisfied users recommend the platform to friends and family. Local partnerships and collaborations with businesses could also help promote the platform within specific communities.

The feasibility analysis shows that the platform is financially viable, technically sound, operationally efficient, and behaviorally engaging. The project establishes the groundwork for effective implementation by resolving all essential issues for primary users with minimalistic and intuitive interface. By retaining scalability and adaptability for future improvements, this comprehensive feasibility assures that the platform will satisfy the wide range of user needs.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Users

The current application focuses solely on **user-centric** functionalities for passengers only, with future plans to incorporate driver-side capabilities.

3.2 Platform

The application is solely developed only for mobile platforms since it ensures easy access and convenience of users, the target platforms are –

- Android
- iOS

3.3 Functional Requirements

There are several primary functional requirements that are listed below which are achievable and necessary to provide an easy-to-use, transparent and secure application:

1. **User Authentication:** Sign-up and log-in functionality via email and Google OAuth using Clerk Authentication.
2. **Location Services:** Fetch current user location, geocode, and reverse geocode locations using Here Maps API.
3. **Ride Booking:** Search, confirm, and book rides based on user input.
4. **Ride History:** Display past ride details or notify if no history exists.
5. **Payment Processing:** Secure payment integration with Stripe.
6. **User Profile Management:** Manage user data, including name, email, and profile picture.
7. **Communication Tab:** Chat functionality to enhance user-driver interaction.

3.4 Non-Functional Requirements

The essential non-functional requirements achievable for the application are:

- **Scalability:** Ability to handle multiple concurrent users.
- **Security:** Ensure secure authentication and payment transactions.
- **Performance:** Provide a seamless and responsive user experience.
- **Portability:** Support for both Android and iOS platforms.

3.5 Hardware and Software Requirements

The development of ChaloRidePe requires a combination of hardware and software tools to ensure the platform's functionality and efficiency.

- **Hardware Requirements:** Hardware requirements for the ChaloRidePe application are listed below in Table 3.1, with each hardware component requirement and its specification description.

Component	**Specification**
Processor	Quad-core 2.5 GHz or Higher
RAM	8 GB or more
Storage	50 GB or more disk space
Database	PostgreSQL

Table 3.1 Hardware Requirements

- **Software Requirements:** Software requirements for the ChaloRidePe application are listed below in Table 3.2, with each software component requirement and its specification description.

Component	**Description**
Expo Mobile Application	SDK version 51 for debugging
Frameworks	React Native Expo, Tailwind CSS
Operating System	Windows 11/MacOS/Linux
APIs	**Location:** Here Maps API **Authentication:** Clerk Services **Payment:** Stripe Payments **Database:** NeonDB Serverless
IDE	VS Code
API Testing Software	Postman

Table 3.2 Software Requirements

CHAPTER 4

DESIGN

4.1 System Architecture

The system architecture closely aligns with a Client-Server architecture (shown in Fig. 4.1) with elements of modular architecture, ensuring scalability and ease of maintenance. The client side is built using React Native Expo, while backend functionalities rely on Clerk Authentication, Here Maps API, Stripe, and Neon Serverless Database.

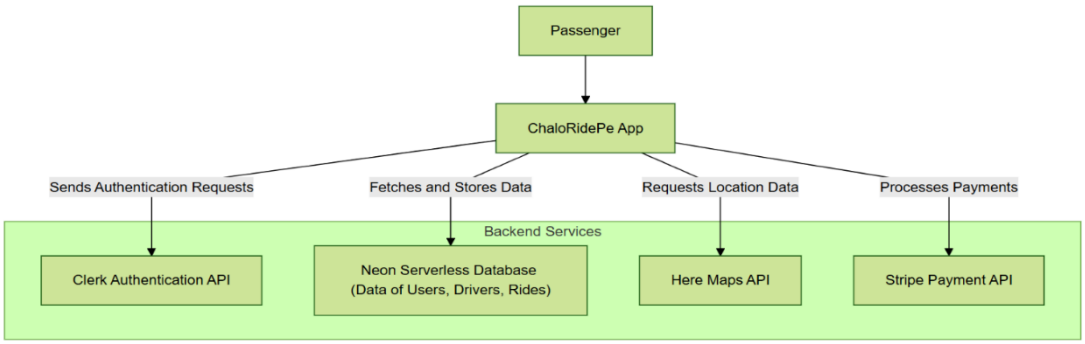


Fig. 4.1 Architecture of ChaloRidePe Application

1) Client-Server Architecture:

a) **Frontend (Client):**

i) Built using **React Native Expo**, it handles the user interface and interactions. It communicates with external APIs and services like Clerk for authentication, Stripe for payment processing, Here Maps for location, and NeonDB for serverless database.

ii) Responsible for rendering the UI, capturing the user input, and displaying the necessary data retrieved from backend or APIs.

b) Backend (Server / Services):

i) Integrated backend services (e.g., Clerk Authentication, Neon serverless database, and Stripe Payment).

ii) These services manage user authentication, database operations, and payment processing without requiring to maintain own server infrastructure.

This client-server architecture allows a clear separation of concerns, where the client focuses on user interaction and the backend services handle data processing, authentication, and transactions.

2) Modular Architecture:

This modularity enhances scalability, maintainability, and ease of integration for new features. The modules developed for ChaloRidePe are as follows:

1) **Authentication:** Manages user signup, login, and logout through Clerk.
2) **Location:** Fetches current location and provides geocoding and reverses geocoding using Here Maps API.
3) **Ride Management:** Enables ride booking, confirmation, and history display.
4) **User Profile:** Manages user-specific data, including name, email, and profile picture.
5) **Payment:** Handles secure transactions via Stripe.

4.2 Functionalities

1. **User Authentication:** Sign-up and log-in functionality via email OTP verification and Google OAuth using Clerk Authentication.
2. **Location Services:** Fetch current user location, geocode, and reverse geocode locations using Here Maps API.
3. **Ride Booking:** Search, confirm, and book rides based on user input.
4. **Ride History:** Display past ride details or notify if no history exists.
5. **Payment Processing:** Secure payment integration with Stripe.
6. **User Profile Management:** Manage user data, including name, email, and profile picture.
7. **Communication Tab:** Chat functionality to enhance user-driver interaction.

4.3 User and Characteristics

For this application, we defined passengers as primary users, and dummy data is taken for the drivers:

1. **Passengers:** Passenger refers to an individual or entity who engages with the platform to find, confirm, and book ride using card payment method offered by the application.
   The characteristics of the user are:
   1) **Registration:** Anyone can register as a passenger.

2) **For Registered user:** A registered user can-
   a. Find ride by searching destination location.
   b. Confirm rides by selecting the driver that best matches as per user requirements.
   c. Book ride by paying for the ride through card.
3) **Tech-savvy:** Familiar with mobile applications and online payment methods.
4) **Time-sensitive:** Require efficient and timely rides.
5) **Cost-conscious:** Seek affordable ride options.

2. **Users in Future Scope:** The current level of development of the application is solely for the users, i.e., passengers. The application aims to cover users such as admin and drivers in future. The current application is using drivers in the dummy data. The characteristics for these future users are same as that of passengers except some additional characteristics, i.e.,

   a. Admins will be responsible for managing the platform, users, and overall system operations.

   b. Drivers will be responsible for accepting ride; fares will be shown to the drivers based on their current location; they can accept or cancel a ride request; they can chat with the fare whose ride they will accept.

4.4 Features of User

1. **User Registration and Login:** The feature allows new users to securely create accounts on the ChaloRidePe application platform. This feature will allow users to:

   a. Sign Up using Email ID and Password:
   By using Clerk Email Authentication, new user can create account on the platform by providing Email ID and password. OTP-based email verification, clerk authentication service is integrated in the application to ensure that only legitimate users create account on the platform.

   b. Sign Up using Google Accounts: By using Clerk OAuth Authentication Services-
      i. New user can sign up on the platform using his/her google accounts.
      ii. Registered user can directly login using his/her google accounts.

   c. Login using Email ID and Password: Registered users can login the application using the credentials created during registration.

2. **Profile Management:** The current level of application has a dedicated tab for the user profile that displays user information such as name, phone number, email id, and profile

picture, which are saved in the database at the time of account creation via email id or google accounts. The application aims to develop the ability for users to update or change their information at any point of time in future even after registration.

3. **Ride Booking:** The feature allows users to book rides by entering pickup and drop locations. It provides a real-time fare estimation based on distance and time.

4. **Live Location Access:** The feature gives real-time live location access when allowed user location access. It allows user to see nearby driver locations with his/her current location also mapped on the map available in the application.

5. **Ride History:** The application maintains all past rides information. It maintains a log of paid rides with details like fare, date, and route, etc.

6. **Online Payment:** The feature allows users to pay online via card payment methods, etc. It uses Stripe payment integration that makes online payment easy, fast, and secure.

4.5 System Requirements

The main requirements of the application are: managing users; proper booking system that wraps features ensuring ride search, ride confirmation and booking redirection; and secure payment gateway which is integrated using Stripe payment services.

1. **User Management:**
    a. Database to store user credentials, profiles, and ride history.
    b. Real-time updates for active users and rides.

2. **Booking System:**
    a. Integration with a geolocation API such as Here Maps API for precise pickup/drop locations.
    b. Driver allocation system based on proximity.

3. **Payment Gateway Integration:**
    a. Secure and scalable payment processing.
    b. Support for different payment methods such as cards- debit/credit, etc.

4.6 Design Goals

Design goals of ChaloRidePe focus on the usability and ease of access of the system that prioritizes security.

1. **Usability:**

- The user interface is designed to be intuitive and minimalistic which promotes increased user-interaction and grabs the attention of the maximum market with users of all ages.

- The application is built with **React Native Expo** and **Tailwind CSS** to create clean and responsive interfaces.

2. **Accessibility:**
   - The application only requires internet connectivity to be able to connect users globally.

   - The application uses Stripe services for payment options which is accessible all over the world hence it enhances the accessibility of system to global users.

3. **Security:**
   - The application uses Clerk Authentication which is robust and ensures secure and verified users with legitimate access.

   - As the application uses Stripe payment, it ensures safe payment which is globally acceptable.

4.7 UML Diagrams

1. Context Diagram

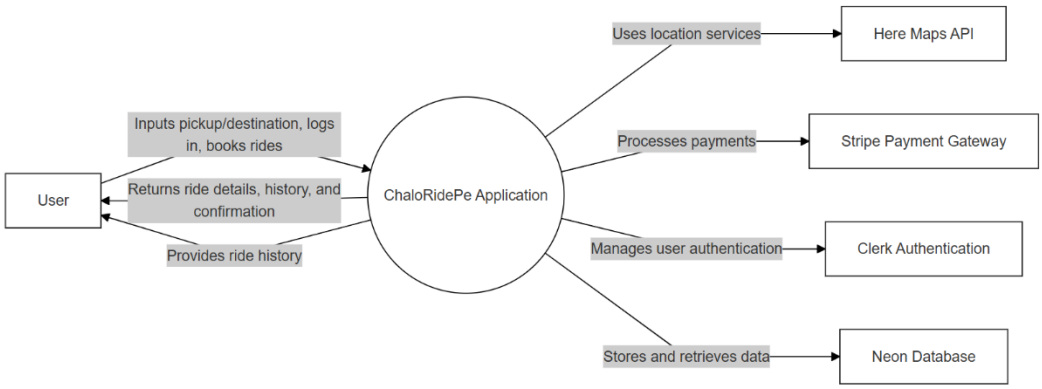


Fig. 4.2 Context Diagram

2. Use Case Diagram

a. For Authentication:

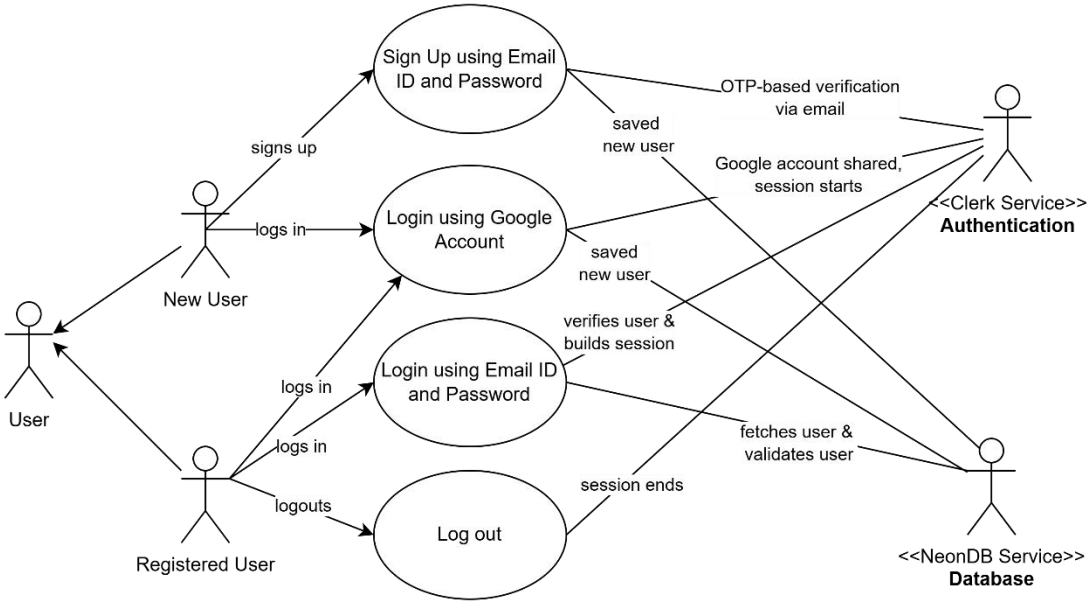


Fig. 4.3 Use Case Diagram: User Authentication

	Event	**Trigger**	**Initiator**	**Use Case**	**Extends / Includes / Services**
1	Sign up via email & password	To create account using email	New User	Sign Up using Email ID and Password	Uses Clerk and NeonDB Service
2	Login via google	To log into the system via google	New User, Registered User	Login using Google Account	Uses Clerk and NeonDB Service
3	Login via email & password	To log into the system via email	Registered User	Login using Email ID and Password	Uses Clerk and NeonDB Service
5	Logout of the system	To log-out of the system	Registered User	Logout	Uses Clerk Service

Table 4.1 Use Case Table: User Authentication

b. **For Ride Booking:**

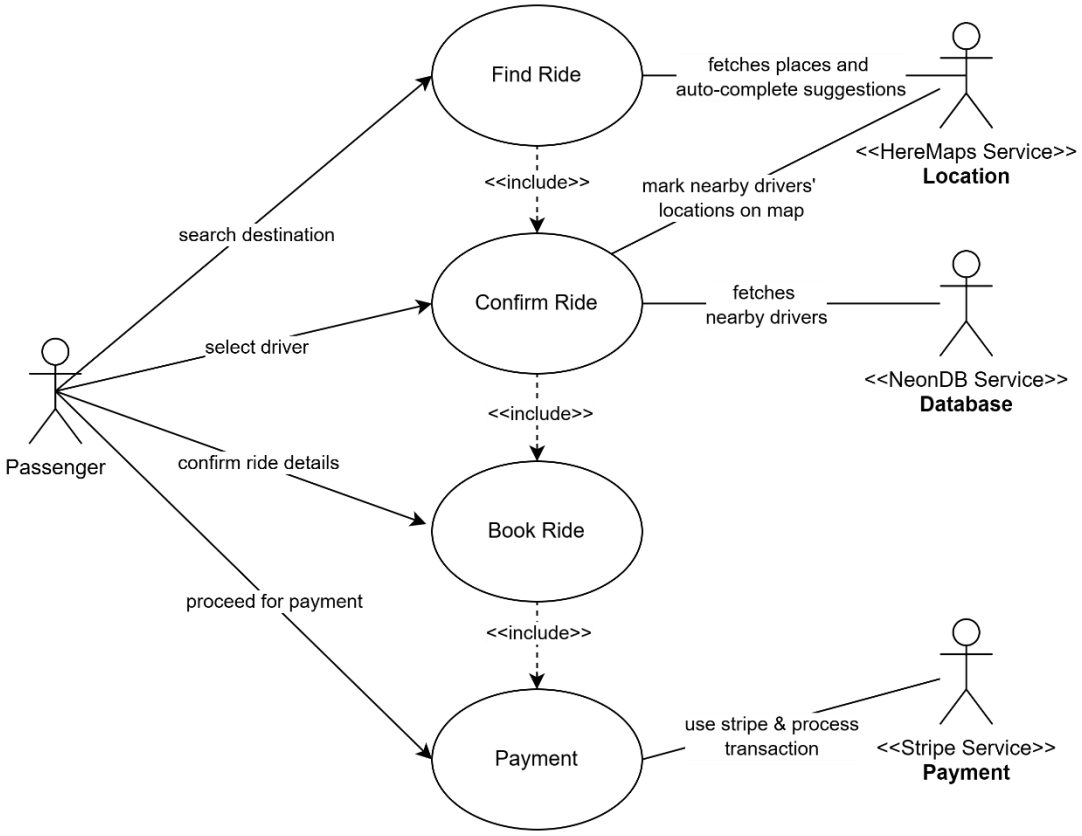


Fig. 4.4 Use Case Diagram: Ride Booking

	Event	**Trigger**	**Initiator**	**Use Case**	**Extends / Includes / Services**
1	Search for input destination	Destination input	Passenger	Find Ride	Uses Here Maps service
2	Driver Selection	Selecting driver from a list of drivers	Passenger	Confirm Ride	Uses Here Maps and NeonDB service Includes Find Ride use case
3	Confirm ride details	Verify ride details	Passenger	Book Ride	Includes Confirm Ride use case
4	Proceed for payment	Continue to payment gateway	Passenger	Payment	Uses Stripe Payment service Includes Book Ride use case

Table 4.2 Use Case Table: Ride Booking

c. **For User Functionalities:**

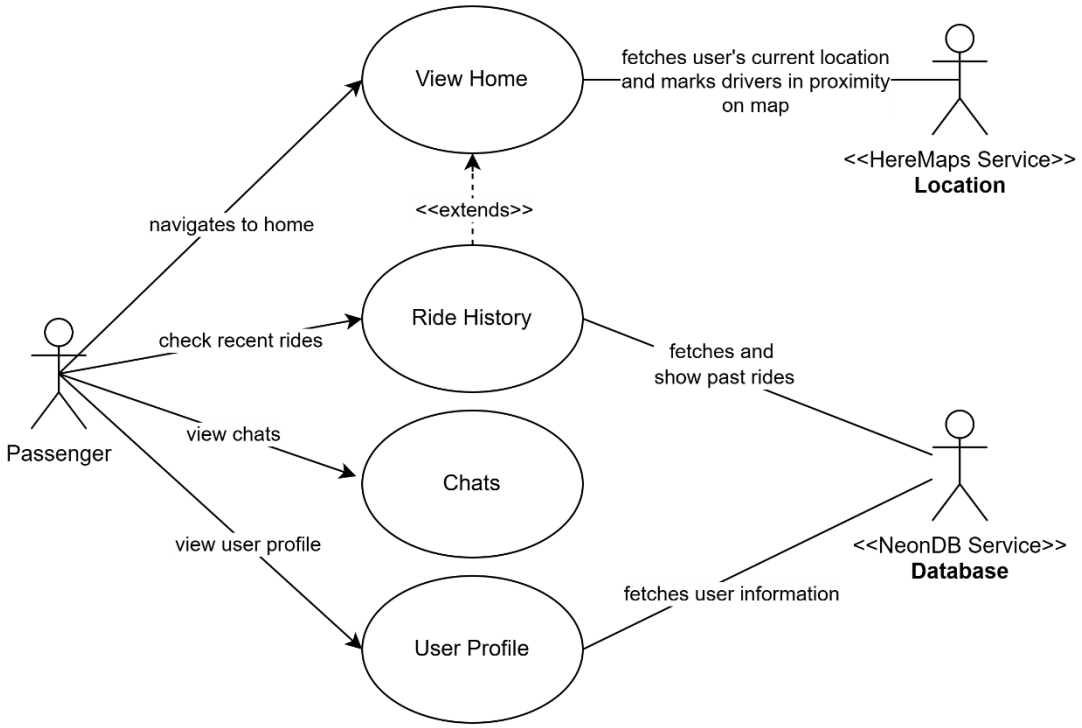


Fig. 4.5 Use Case Diagram: User Functionalities

	Event	**Trigger**	**Initiator**	**Use Case**	**Extends / Includes / Services**
1	Home navigation	Home tab	Passenger	View Home	Extends Location Services and Ride history use cases
2	Check recent rides	Ride history tab	Passenger	Ride History	Uses NeonDB service
3	View chats	Chat tab	Passenger	Chats	
4	View user profile	User profile tab	Passenger	User Profile	Uses NeonDB service

Table 4.3 Use Case Table: User Functionalities

3. ER Diagram

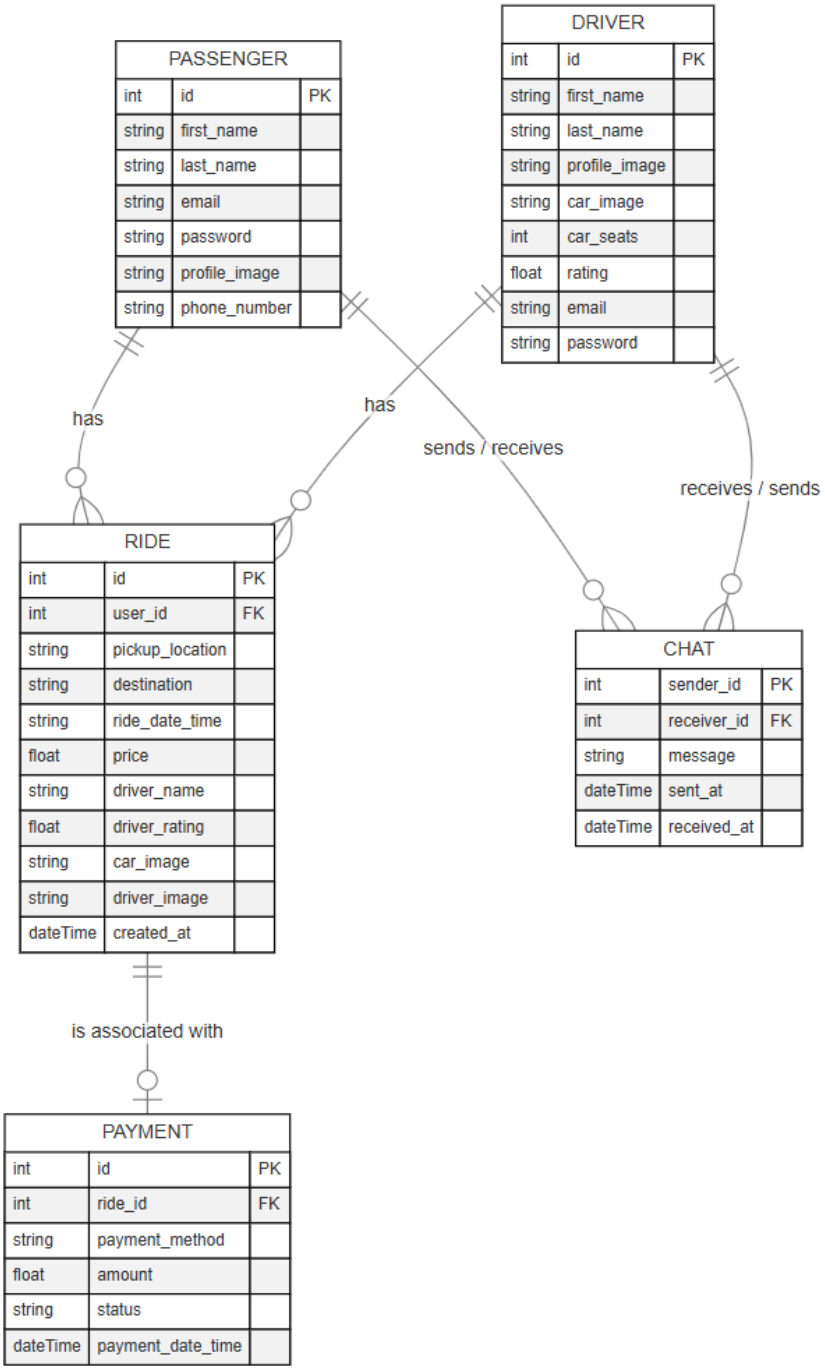


Fig. 4.6 ER Diagram

4. Class Diagram

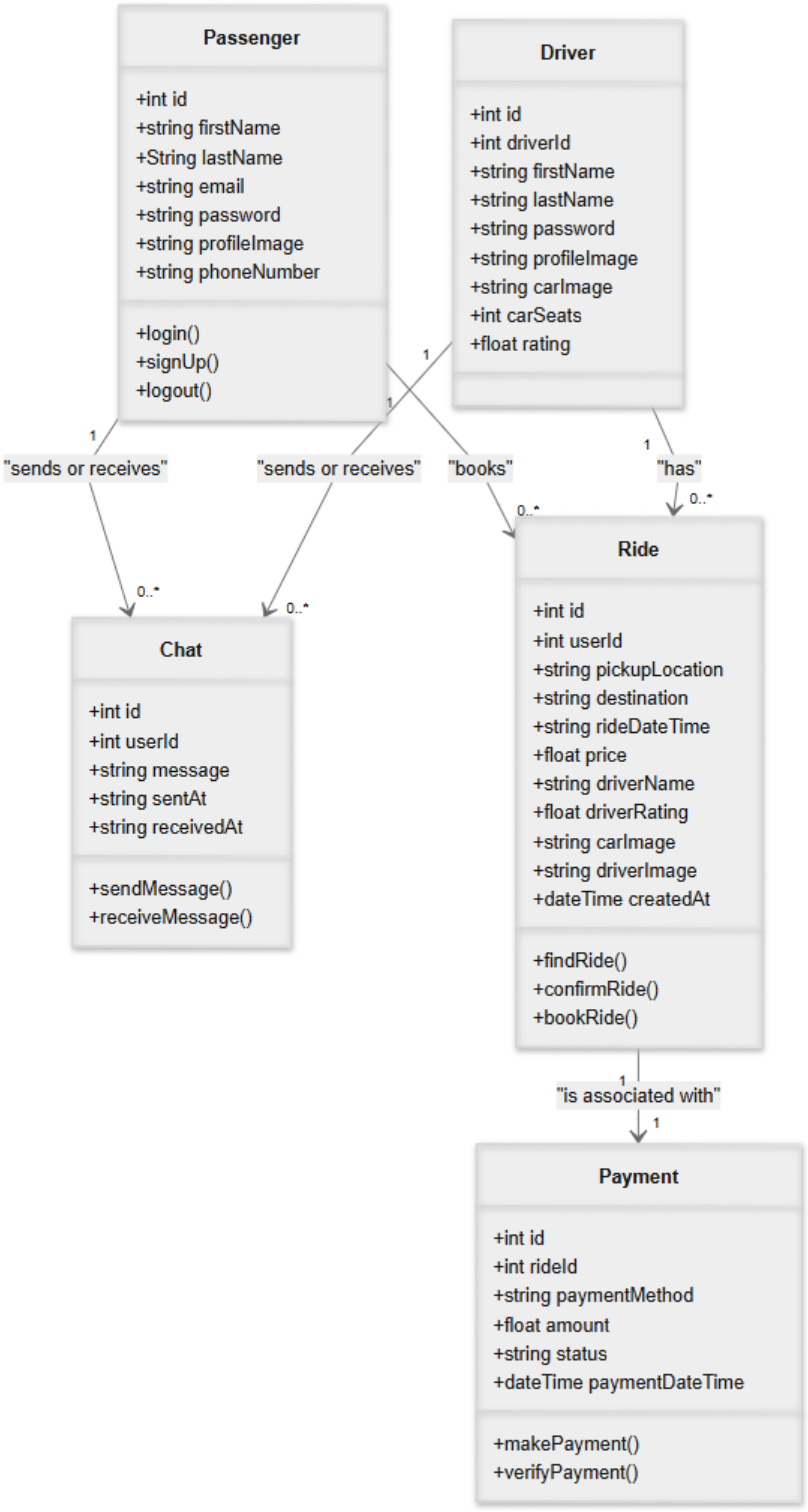


Fig. 4.7 Class Diagram

5. Sequence Diagram

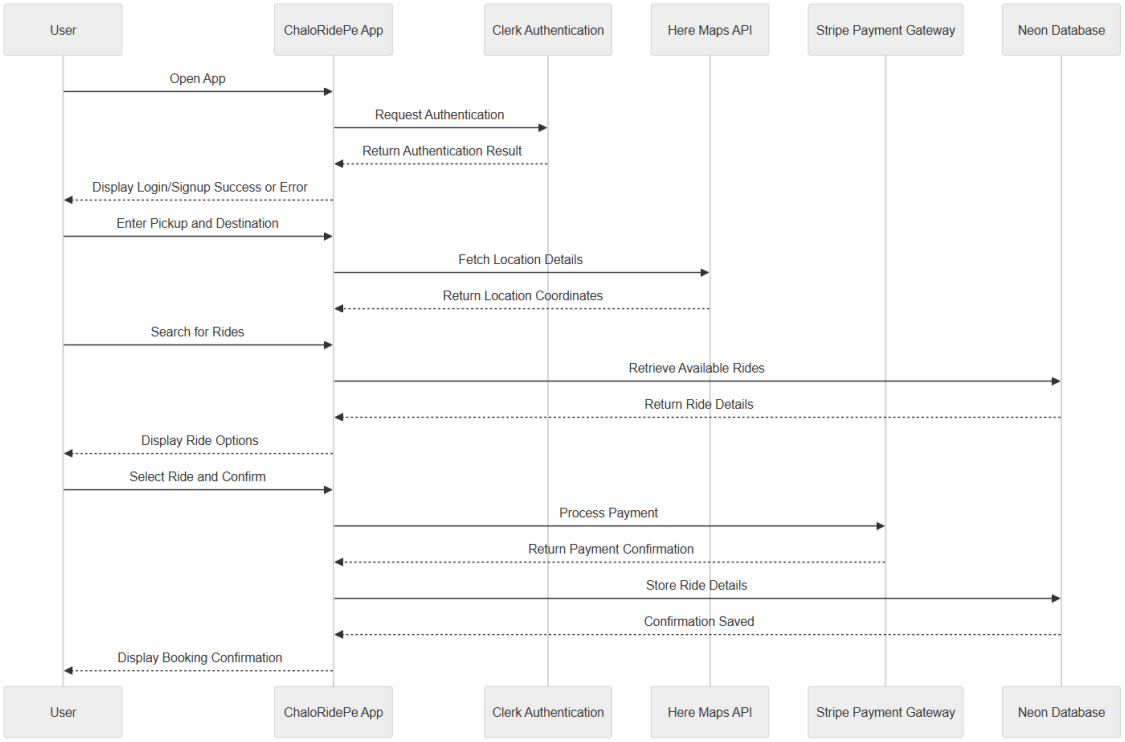


Fig. 4.8 Sequence Diagram

6. Level 0 DFD

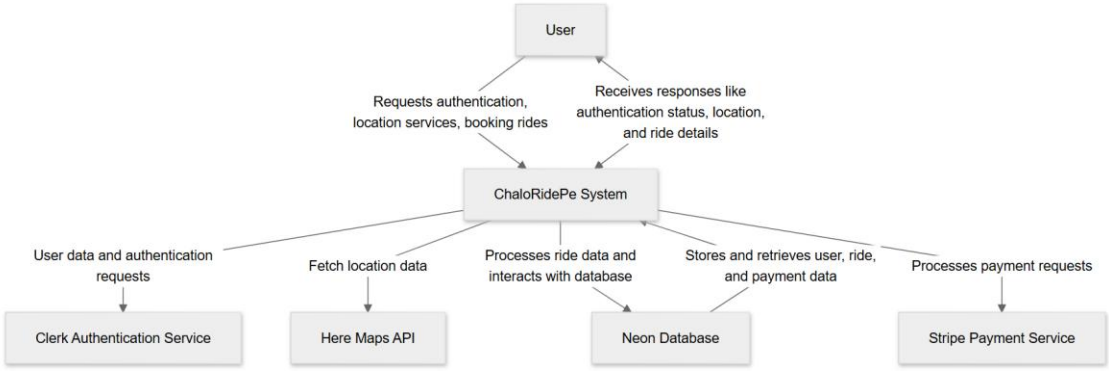


Fig. 4.9 Level 0 DFD

7. Level 1 DFD

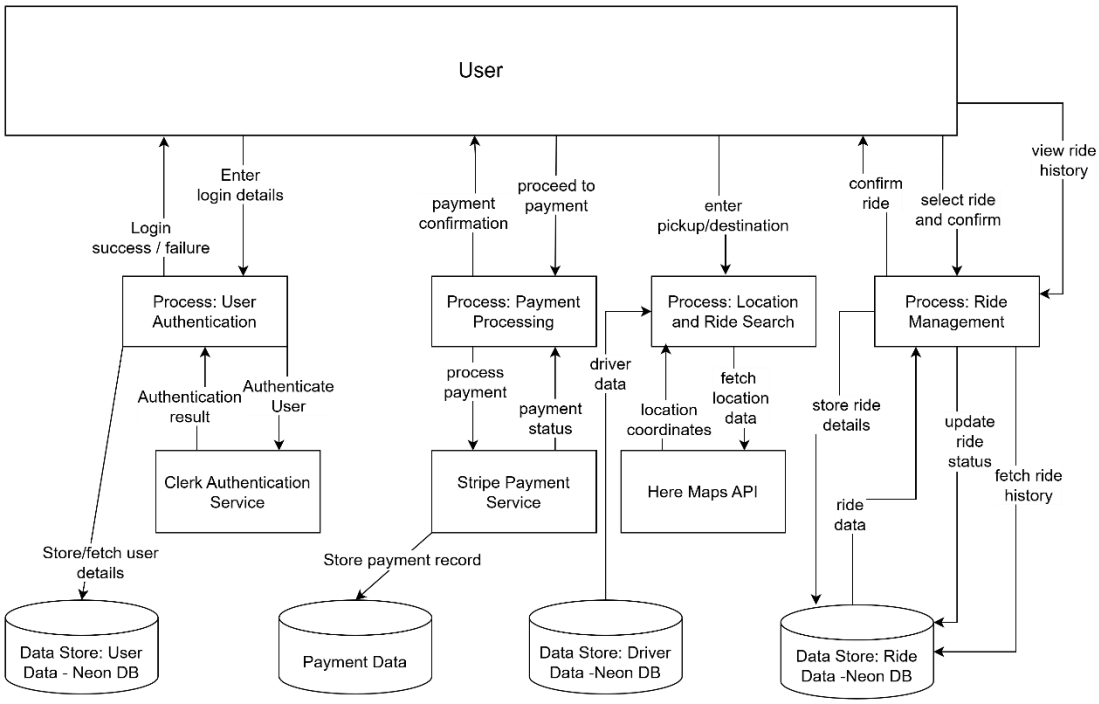


Fig. 4.10 Level 1 DFD

8. Level 2 DFD

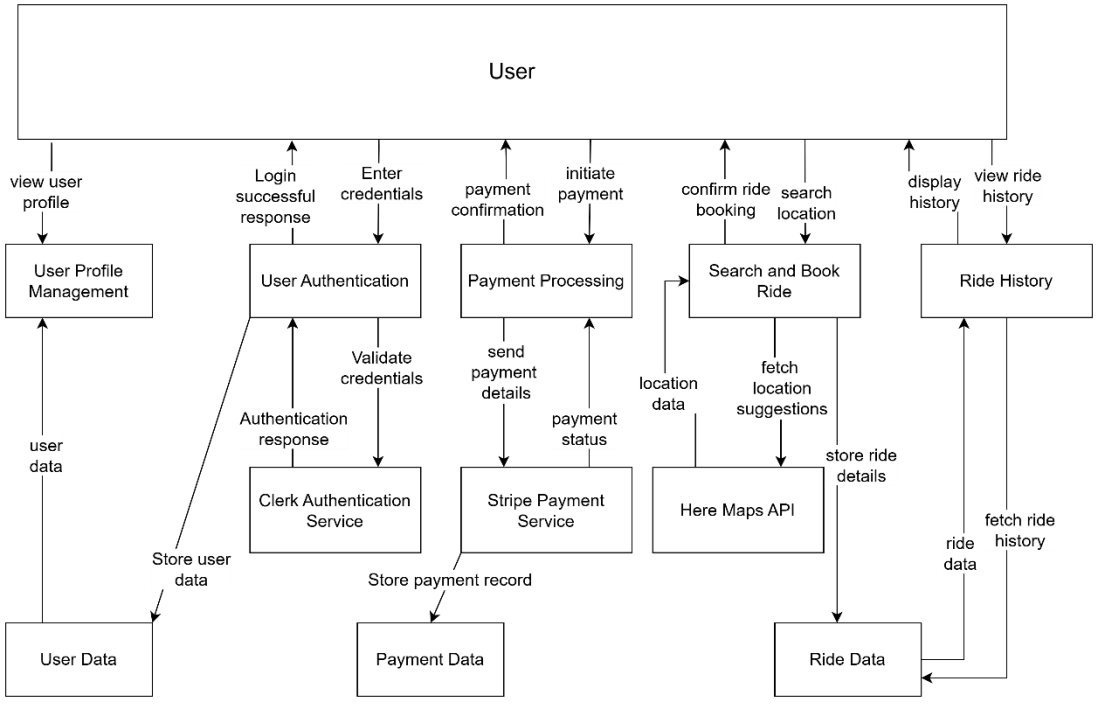


Fig. 4.11 Level 2 DFD

CHAPTER 5

PROJECT FLOW

5.1 Flow of Project

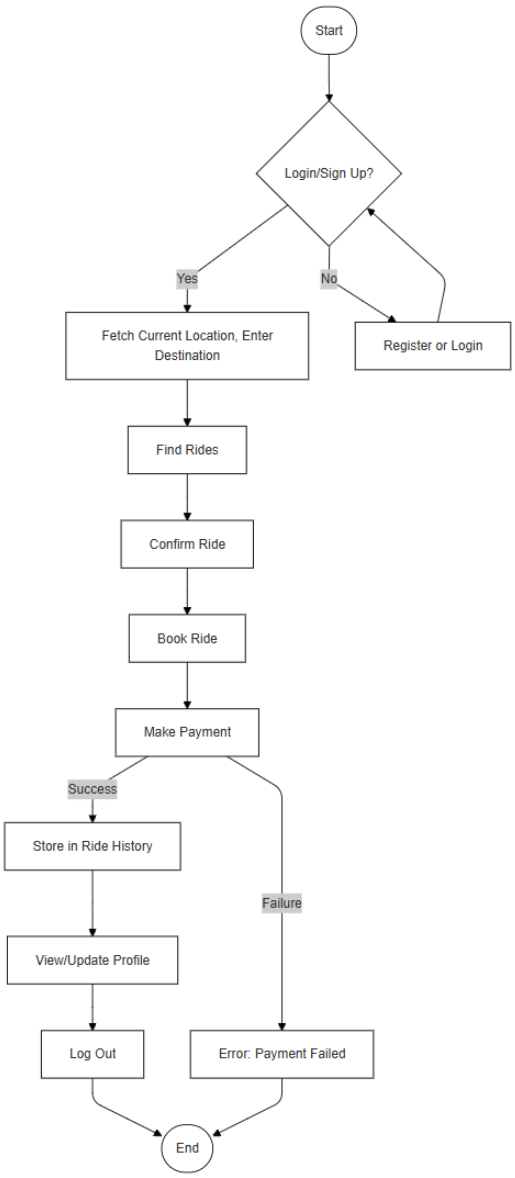


Fig. 5.1 Flow of application

The flowchart in the Fig. 5.1 shows the complete flow of the application wherein the application utilizes all technologies with the front end developed using React Native Expo and Tailwind CSS and backend facilities are achieved using services such as Neon Serverless Database, Stripe Payments and Clerk Authentication.

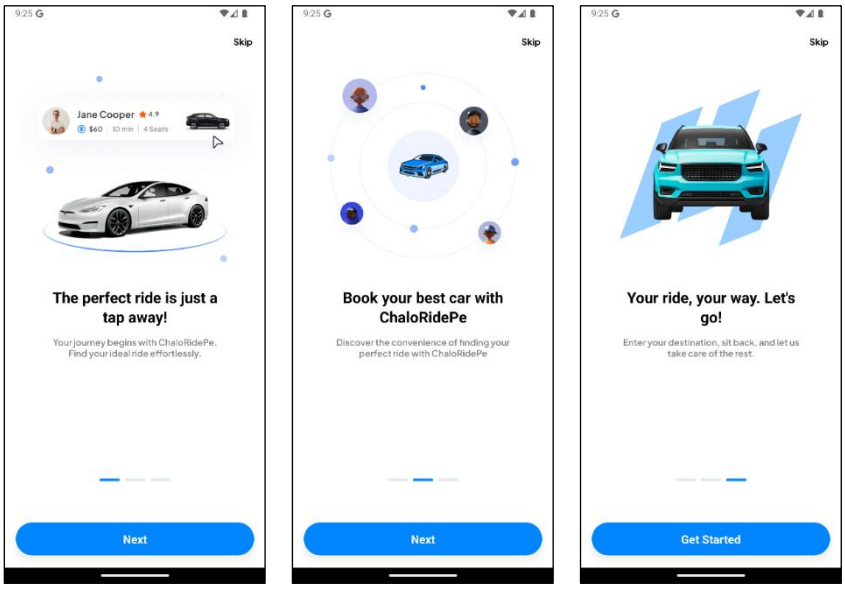
5.2 Implementation

1. **Logged out Users / New Users:** Initially, when application starts, a splash screen appears at the start shown in Fig. 5.2.

   For new users, the application redirects to the welcome screen which is also known as application's onboarding screen that onboards new users to the application. Fig. 5.3 shows onboarding screen, it has a slider attached that allows it to slide in between all the slides related to application features shown in Fig. 5.3 (a), (b) and (c).

   For already registered users who are not logged in to the platform, they are redirected to the Sign-Up screen (refer to Fig. 5.4). And for logged in users, the application resumes their previous sessions.

Fig. 5.2 Splash screen



(a) **(b)** **(c)**

Fig. 5.3 Onboarding screens

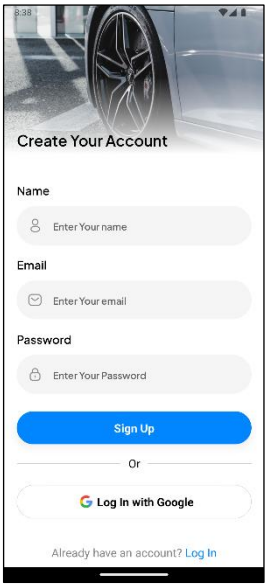


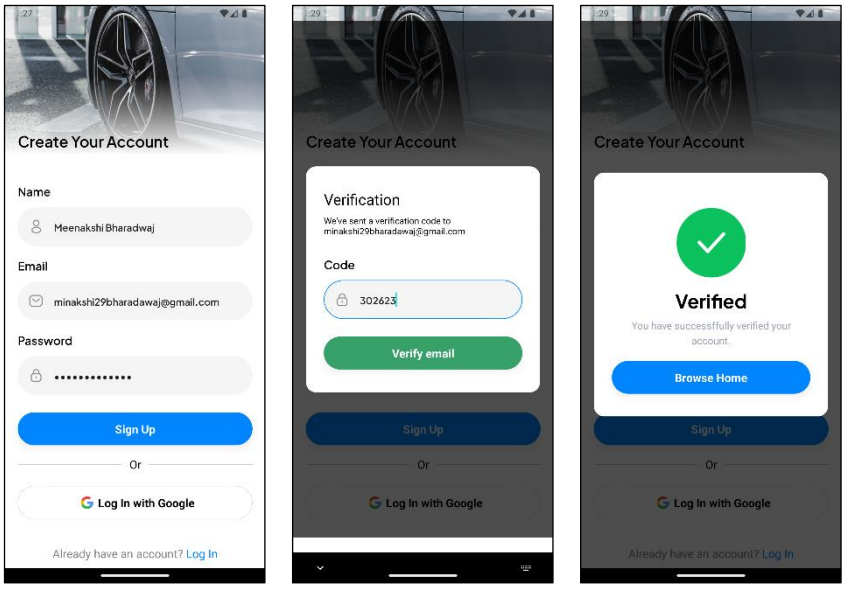
Fig. 5.4 Sign-up screen

2. **User Authentication:** Sign-up and log-in functionality via email OTP verification and Google OAuth using Clerk Authentication service. The user can Sign-Up on the platform using two ways:

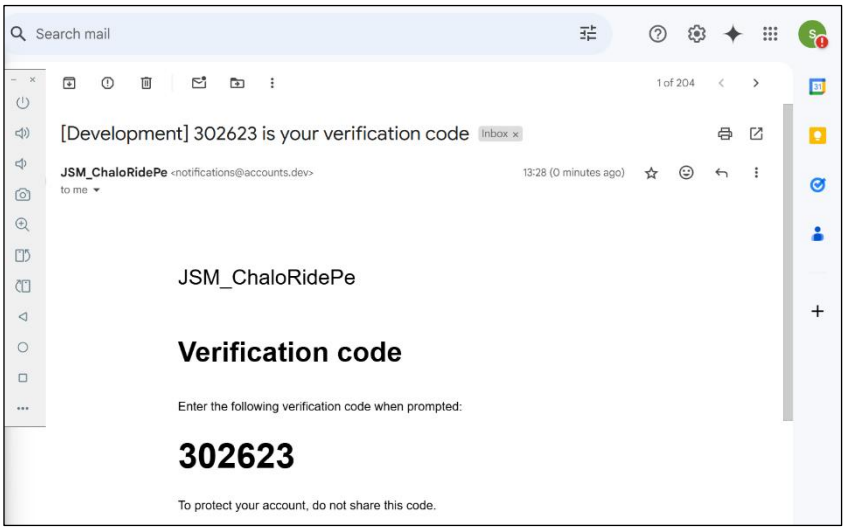
   - Using Email ID and Password
   - Using Google Account

   In Fig. 5.5 (a), the user signs up using his/her email id and password. The user is authorized and verified via email OTP verification authentication service provided by Clerk Services. After successful account

creation, the user is redirected to the application home page (refer Fig 5.6 (c)).



(a) Sign Up screen **(b) Email OTP** **(c) Account verified**



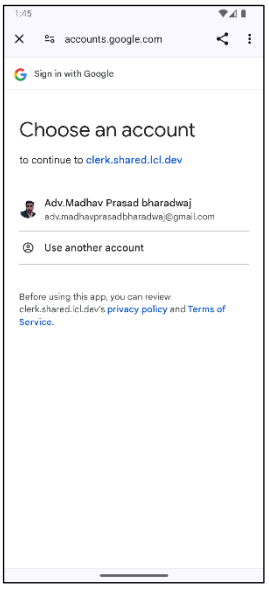
(d) OTP received on email

Fig. 5.5 Sign-up via email and password

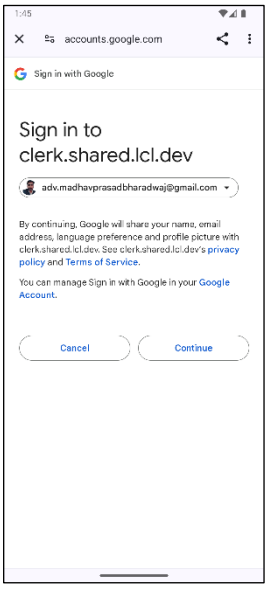
In Fig. 5.6, it shows the process of sign up via google account. The user signs up (or can directly log in whether already registered or not) using the google account and his/her google account details are shared with the clerk services which are then later utilized in maintaining the user profile.

In Fig 5.6 (a), asks to choose google account to continue login. After selecting account, asks for permissions to share google account information with Clerk services so that profile picture, name and email can be read by the application (refer to Fig. 5.6 (b)). After granting permissions, the application successfully authenticates the user as shown in Fig. 5.6 (c) and redirects him

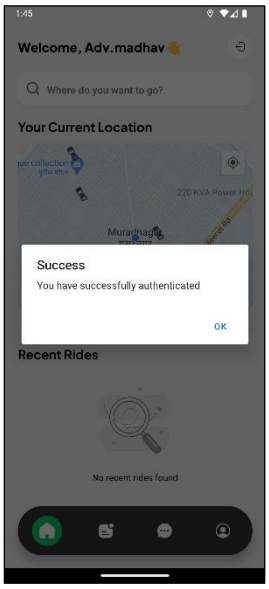
to the user home page of application. The user profile is updated with details shared by the google account (refer to Fig. 5.6 (d)).



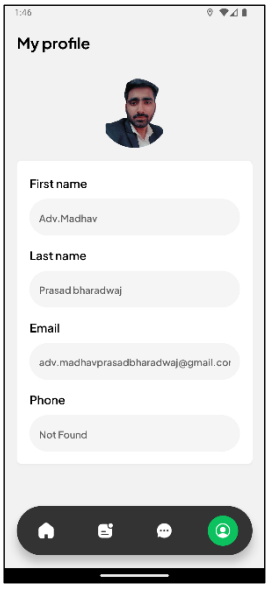
(a) Choose account



(b) Asks for permissions



(c) Login successful



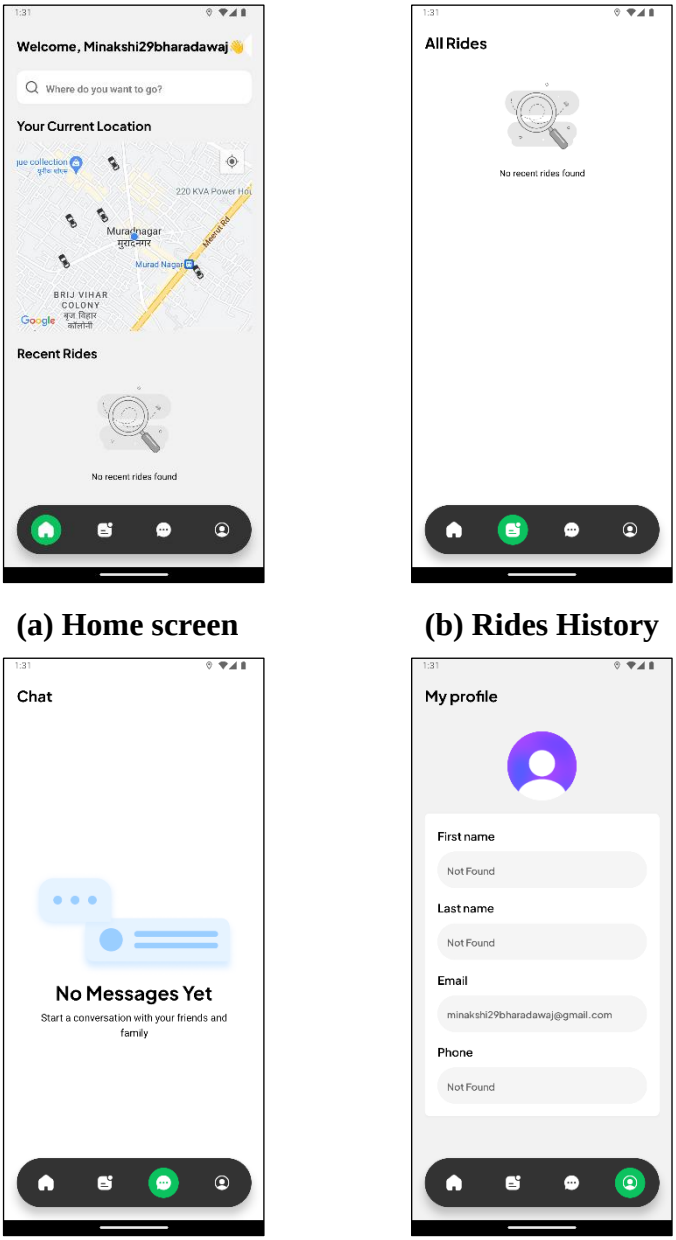
(d) User Profile

Fig. 5.6 Login/Sign-up via google account

3. **User Profile Management:** Logged in user can access the home screen, ride history, chats, and profile by navigating the respective tabs.

In Fig. 5.7 (a), user home screen is displayed wherein user's current location, ride history are displayed, and a feature to input destination is also provided. In Fig. 5.7 (b), a dedicated tab for ride history is designed. Each past ride has related information such as pickup location, destination, arrival time, payment status, driver name, etc. In Fig. 5.7 (c), Chats screen is also

available as a part of user functionalities which is aimed to be developed in future. The application fetches current user information from google accounts or neon database and displays it in the user profile screen (refer to Fig. 5.7 (d)). User profile has user data, including name, email, and profile picture.



(a) Home screen **(b) Rides History**

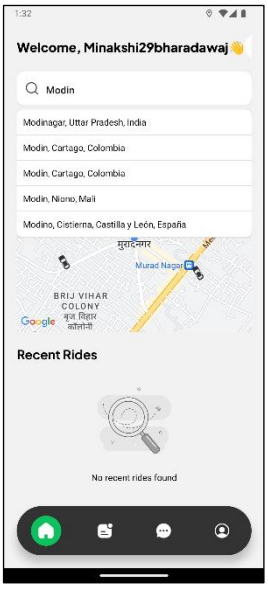
(c) Chats screen **(d) Profile screen**

Fig. 5.7 User profile management interfaces

4. **Ride Booking:** All ride booking functionalities such as search, confirm, and book rides based on user input.

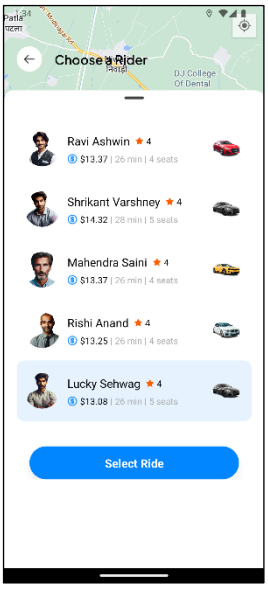
In Fig. 5.8 (a), input destination gives places auto-complete suggestions which when selected one of the location redirects to the find ride screen (refer to Fig 5.8 (b)). After proceeding to find ride, the application asks user to confirm the ride that best fits and is appropriate for him on the confirm ride

screen (refer to Fig. 5.8 (c)). Finally, in Fig. 5.8 (d), the user is asked to confirm the ride details to continue to payment process.

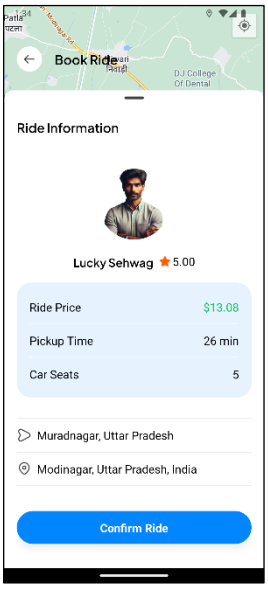


(a) Input destination

(b) Find Ride screen



(c) Confirm Ride screen



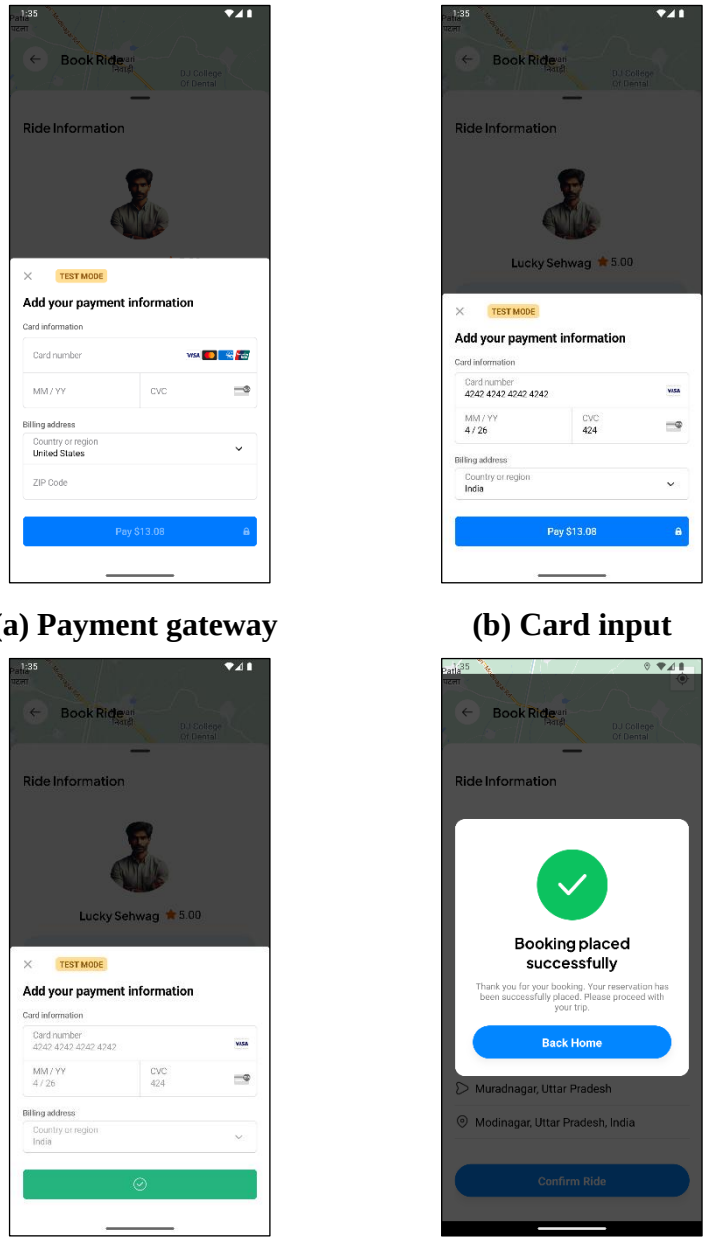
(d) Book Ride screen

Fig. 5.8 Ride booking process interfaces

5. **Payment Processing:** After confirming ride details, the user is asked for the payment.

Secure payment is implemented by integrating Stripe payments services. In Fig. 5.9 (a), the user is asked to enter the card details. In Fig. 5.9 (b), the development mode has ensured test mode card details to process and verify payments. After clicking on Pay button, the API processes the payment and after successful process shows interface as in Fig. 5.9 (c).

The application also ensures payment verification by showing a confirmation of successful ride booking as shown in Fig. 5.9 (d).



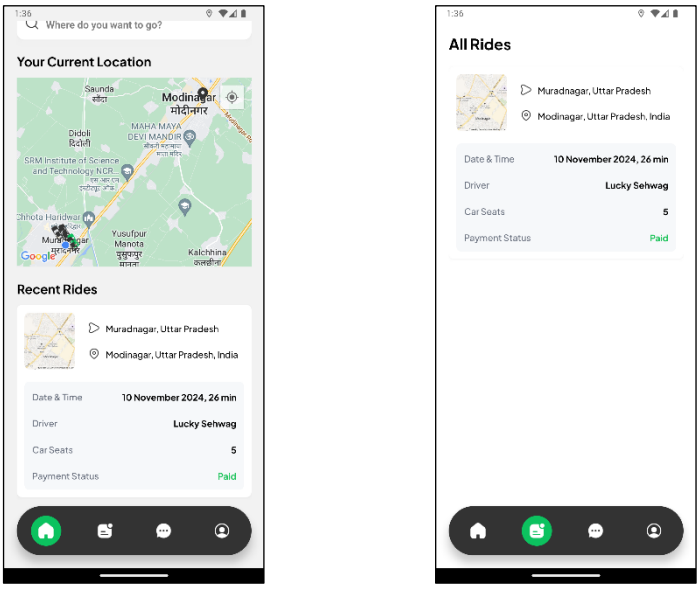
(a) Payment gateway **(b) Card input**

(c) Success **(d) Confirmation**

Fig. 5.9 Payment process interfaces

6. **Ride History:** As the ride booking is processed and completes after successful payment, it also updates the rides database as well as updates the interfaces for ride history on the home screen (refer to Fig. 5.10 (a)) and ride history tab as well (refer to Fig. 5.10 (b))).

All ride related information is shown such as pick-up location, destination location, date and time, driver name, car seats, and payment status as well along with as snapshot of route of pick-up to destination location.



(a) Home screen **(b) Ride History**

Fig. 5.10 Updated interfaces for past rides

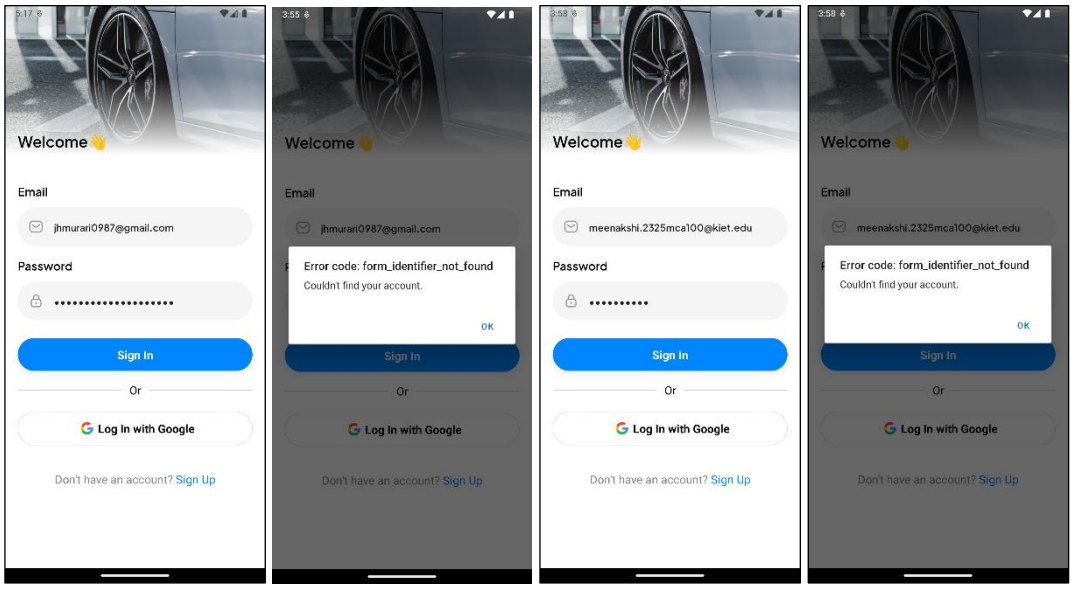
CHAPTER 6

TESTING

For testing and debugging, all the boundary test cases were covered to ensure proper working of the application. Test cases that would cause the application to crash or raise exceptions and errors were tested, handled and removed during debugging.

6.1 Test Cases

- **User doesn't not exist in DB**



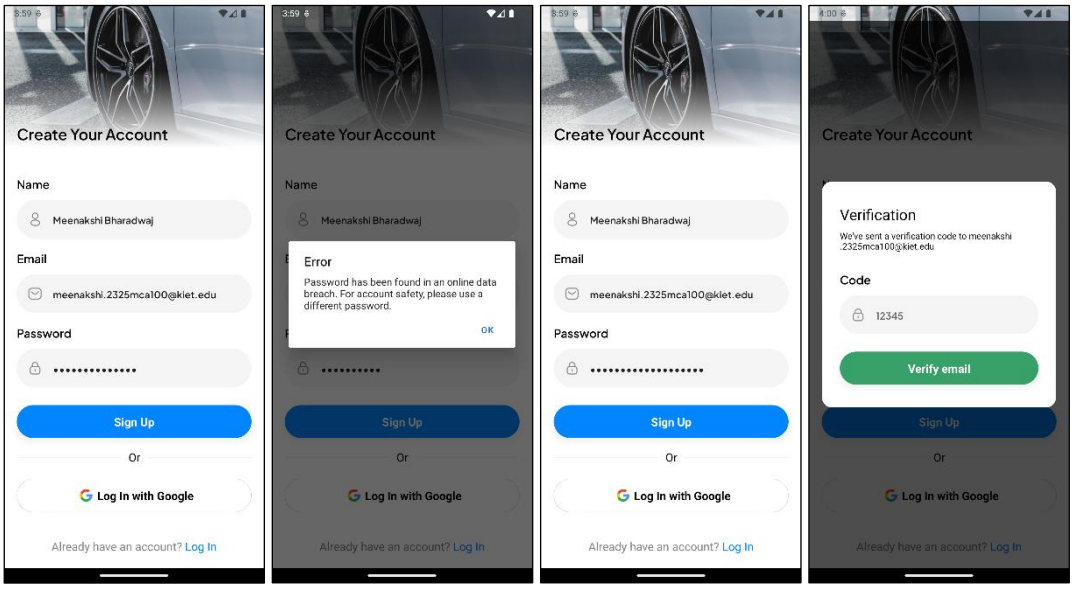
(a) **(b)** **(c)** **(d)**

Fig. 6.1 User not exists: test cases for two users

In Fig. 6.1 (a) and (c), user 1 and user 2 both try to log into the account but since their account is not created on the application, they are shown an alert message informing them that the system in unable to find their account in the database. Even if their accounts exist, additional checks and tests are tested to ensure proper flow of application.

- **Secure Password Creation**

The Clerk Authentication covers additional functionalities such as creating a strong password that it unique and secure. In Fig. 6.2 (b), the application suggests user to enter a strong password since the entered password was found in an online data breach and is not secure enough to protect user account.

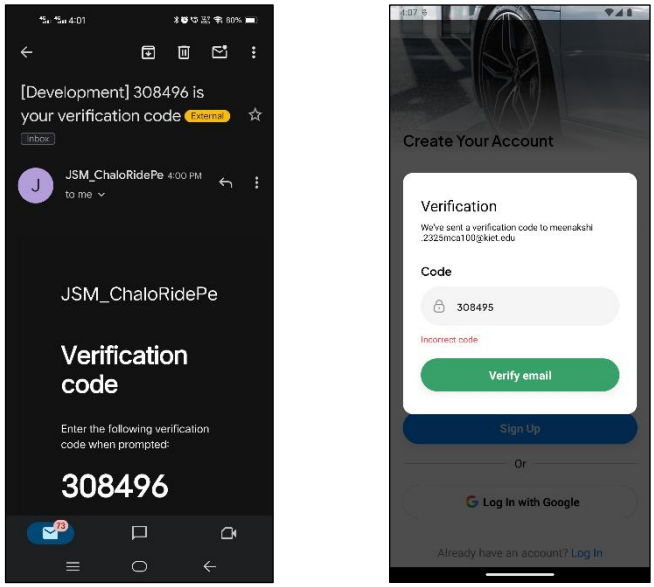


(a) (b) (c) (d)

Fig. 6.2 Secure and Strong password creation

The application does not proceed until the user enters a secure and strong password that is not found in data breach. When entered strong password as shown in Fig. 6.2 (c), the application proceeds and navigates to OTP verification as shown in Fig. 6.2 (d).

- **Incorrect OTP entered while email verification**



(a) (b)

Fig. 6.3 Incorrect OTP

As the user inputs incorrect OTP, the application displays warning text about incorrect code entered hence verification does not proceed until the user enters correct code and verifies. Each time when user clicks on the **Verify email** button, a new code is generated and sent over email but it has a limited number of attempts, if crossed the limit of attempts, it will ask user to wait for some time and then try again later.

- **Too many attempts with incorrect OTP**

In Fig. 6.4, the application handles too many OTP attempts as well and displays a warning in red text suggesting user to try later in sometime.

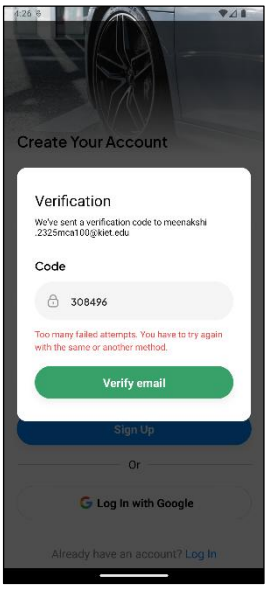
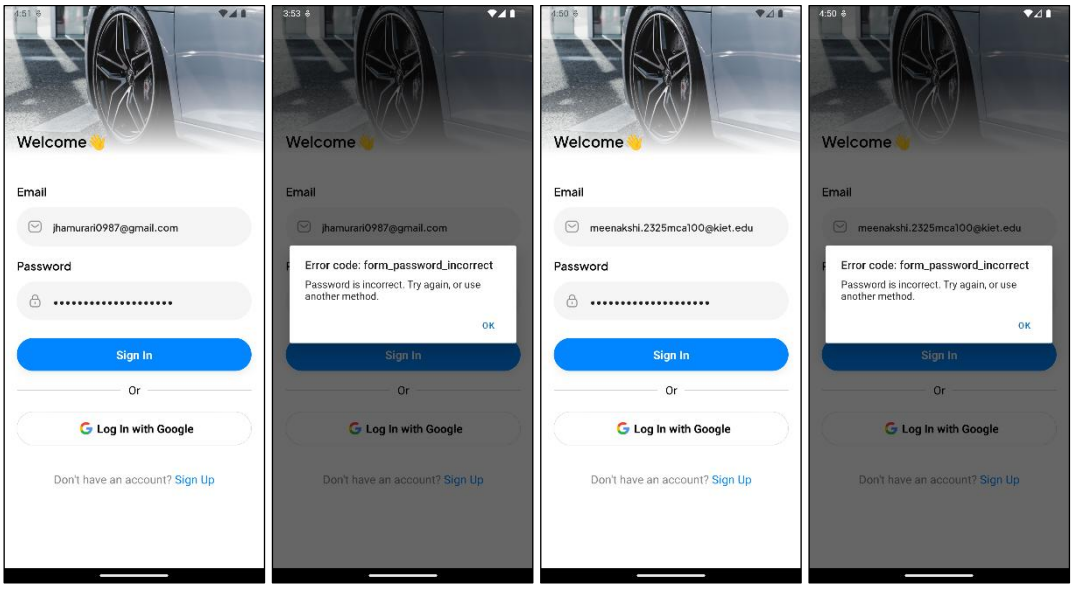


Fig. 6.4 Incorrect OTP: Too many attempts

- **Incorrect password**



(a) (b) (c) (d)

Fig. 6.5 Incorrect password: test cases for two users

- **Invalid destination input**

Invalid destination input is already handled by the application. The application ensures to generate auto-complete suggestions of the valid places which needs to be selected by the user to proceed, hence the application never proceeds on any invalid destination location as there is no suggestion for such location. In Fig. 6.6, it is clearly visible that there is no autocomplete suggestion for invalid destination input.

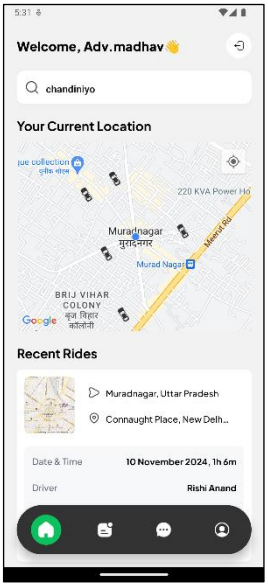
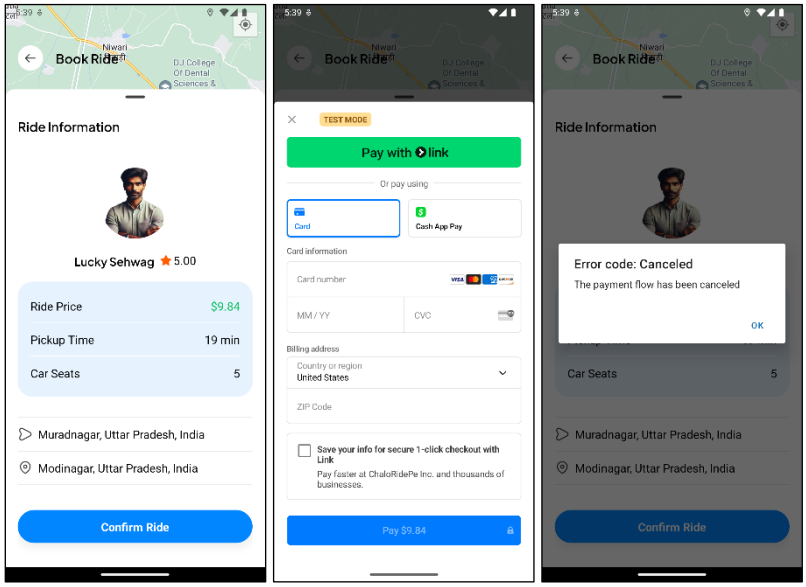


Fig. 6.6 Invalid destination input

➢ **Payment Failed or Canceled**

When the user clicks back button before clicking on **Pay** button, the application cancels the payment flow as shown in Fig 6.7, and displays alert message for the same.



(a) **(b)** **(c)**

Fig. 6.7 Payment canceled by user before processing

CHAPTER 7

FUTURE SCOPE

The future scope involves developing the application in a full-fledged way that would ensure the complete workflow of the application and would also encourage economical productivity with revenue generating capabilities. To develop the application complete and robust the following features can be added in later versions:

- Implementing driver-side functionalities.
- Real-time ride tracking for users.
- Push notifications for ride updates.
- Enhanced user chat with multimedia support.
- Driver verification.
- Incorporating payment options such as UPI, and Cash on delivery, etc.
- Incorporating system admin functionalities to manage and operate the application.
- Ratings for drivers.
- Customer support for users.
- Chat support for users and drivers to track, query, and chat when required.
- Feature to be able to book for someone other.
- Car type and car's interior images with all other details related to car.

Incorporating all these features will make the application robust, and complete, and it will also improve the credibility and high customer satisfaction.

CHAPTER 8

CONCLUSION

The application is developed with aim to primarily serve the passenger users. It aims to provide a user-friendly interactive interface that promotes secure, safe and faster ride-booking.

The development of ChaloRidePe has proved the feasibility and effectiveness of the use of modern tools and APIs in the development of scalable and user-centric ride-booking apps. The project managed to meet all its core objectives: secure authentication, seamless ride booking, real-time location tracking, as well as seamless payment processing. Utilizing React Native Expo, the app achieves cross-platform compatibility, enhanced security, and user trust due to the integration of Clerk Authentication and Stripe.

The application uses a modular design, in conjunction with an easy-to-use interface, to meet the needs of its target audience while ensuring that it is easily usable and appealing. The serverless database solutions such as Neon help to keep operational costs low while upscaling both availability and performance. The ability for users to look up ride histories and profiles proves advantageous in providing personalized experiences.

However, future improvements can still be made to the application, such as adding features like driver-side features, real-time ride tracking, and push notifications. These additions will work towards summarizing the breadth of functionality and use associated with the app to create and sustain competitiveness in the race with evolving user expectations.

In summary, ChaloRidePe is a solid base for building this ride-booking platform, albeit one that can be upscaled in response to future requirements and technology trends. It also stands as testimony to the power of blending open-source tools, responsive APIs, and innovative principles of design to address real-world issues ably.

REFERENCES

1. React Native Documentation. [https://reactnative.dev/]
2. Clerk Authentication Services. [https://clerk.dev/]
3. Here Maps API Documentation. [https://developer.here.com/]
4. Stripe Payment Integration. [https://stripe.com/docs/]
5. Neon Database Documentation. [https://neon.tech/]