
Design Document

for

PedalPal

Version 1.0

Prepared by

Group # 4

Raghav Manglik	220854
Amogh Bhagwat	220288
Srishti Chandra	221088
Wadkar Srujan Nitin	221212
Anaswar K B	220138
Khushi Gupta	220531
Ananya Singh Baghel	220136
Pathe Nevish Ashok	220757
Debraj Karmakar	220329
Kaneez Fatima	220496

Group Name: Bit Brewers

raghavkmanglik@gmail.com
amogh.2004b@gmail.com
chandra.srishti2403@gmail.com
srujanwadkar@gmail.com
anaswarkb013@gmail.com
khushi07g@gmail.com
ananyabaghel2004@gmail.com
nevu.pathe1234@gmail.com
debraj2003jsr@gmail.com
kaneezfatimamehdi7@gmail.com

Course: CS253
Mentor TA: Mr. Bharat
Instructor: Prof. Indranil Saha
Date: February 9, 2024

Contents

1 Revisions	2
2 Context Design	3
2.1 Context Model	3
2.2 Human Interface Design	3
2.2.1 Login and Registration Pages	4
2.2.2 Viewing Hubs and Booking a Ride	4
2.2.3 Starting a Ride	5
2.2.4 User Feedback	5
2.2.5 Subscription Model	6
2.2.6 Navigation Bar and Settings	6
2.2.7 Analytics and Wallet	7
2.2.8 Admin Interface	7
3 Architecture Design	9
4 Object Oriented Design	10
4.1 Use Case Diagrams	10
4.1.1 Use Case 1 (UC1) - Start Ride and End Ride	11
4.1.2 Use Case 2 (UC2) - View Nearby Hubs on a Map	11
4.1.3 Use Case 3 (UC3) - Create a new Account	12
4.1.4 Use Case 4 (UC4) - View Analytics	12
4.1.5 Use Case 5 (UC5) - Feedback Mechanism	13
4.1.6 Use Case 6 (UC6) - Use and Recharge Wallet	13
4.2 Class Diagrams	14
4.3 Sequence Diagrams	15
4.3.1 Login Sequence	15
4.3.2 Registration Sequence	15
4.3.3 Reset Password Sequence	16
4.3.4 Fetch Cycle Hubs Sequence	16
4.3.5 Book Ride Sequence	17
4.3.6 Payment Sequence	17
4.3.7 Feedback Sequence	18
4.3.8 View Analytics Sequence	18
4.3.9 Admin Sequence	19
4.4 State Diagrams	20
4.4.1 Overall state diagram	20
4.4.2 Ride Begin state diagram	21
4.4.3 Ride End state diagram	21
4.4.4 Maintenance State Diagrams	22
4.4.5 Account Closing	22
5 Project Plan	23
5.1 Project Planning	23
5.2 Code Collaboration	23
5.3 Communication	24

5.4 Work Division	24
Appendices	25
Appendix A Group Log	25

1. Revisions

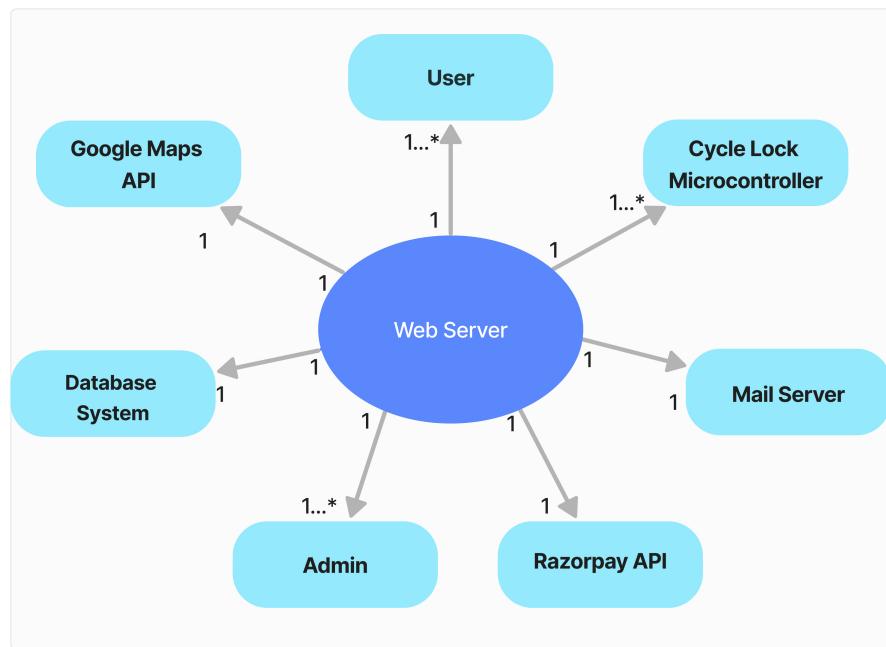
Version	Primary Author(s)	Description of Version	Date Completed
v1.0	Raghav Manglik Amogh Bhagwat Srishti Chandra Wadkar Srujan Nitin Pathe Nevish Ashok Debraj Kamakar Khushi Gupta Ananya Baghel Anaswar K B Kaneez Fatima	First version of the Software Design Document	09/02/23

2. Context Design

2.1 Context Model

The context model shows the various entities that interact with the system, and the relationships between them. It plays a crucial role in supporting efficient context management. A brief explanation of the context model is as follows -

- **User:** interacts with the server via a mobile application
- **Admin:** has various privileges and interacts via a web application
- **Database:** stores all data of the system, including the user credentials, statistics and wallet information
- **Payment Gateway:** integrates payment functionality, enabling financial transactions within the system
- **Google Maps API:** provides functionality of displaying a map, particularly useful for showing cycle hub locations within the IIT Kanpur campus to users.
- **Mail Server:** facilitates communication by sending password reset and reminder emails to users.
- **Cycle Lock Microcontroller:** hardware component communicating with the system to relay the status of cycle locks, crucial for monitoring and management.

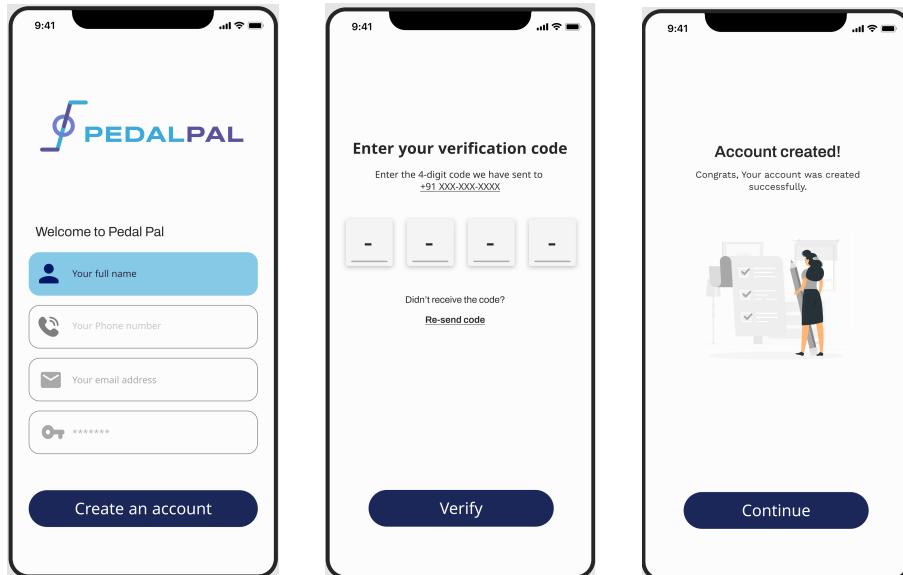


2.2 Human Interface Design

Users will engage with the system via a mobile application featuring an intuitive user interface (UI) that is accessible on Android devices, ensuring a seamless experience regardless of platform.

2.2.1 Login and Registration Pages

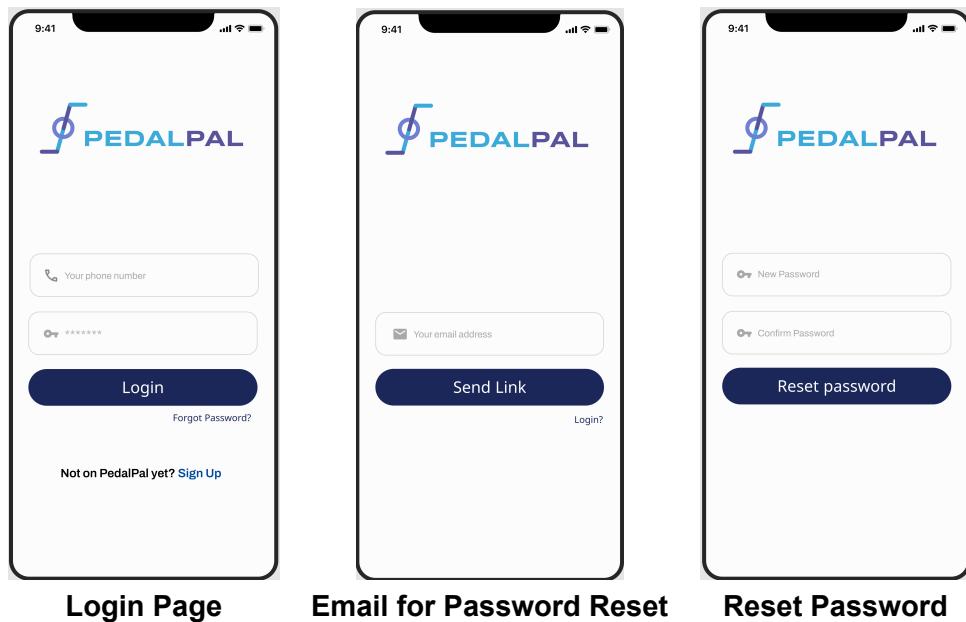
In order to use PedalPal, the User has to login first. If the user has not created an account yet, he/she can click on Sign Up to create a new account. If the user has forgotten his/her password, he/she can reset it using the "Forgot Password" button.



Registration Page

Phone Authentication

Account Created



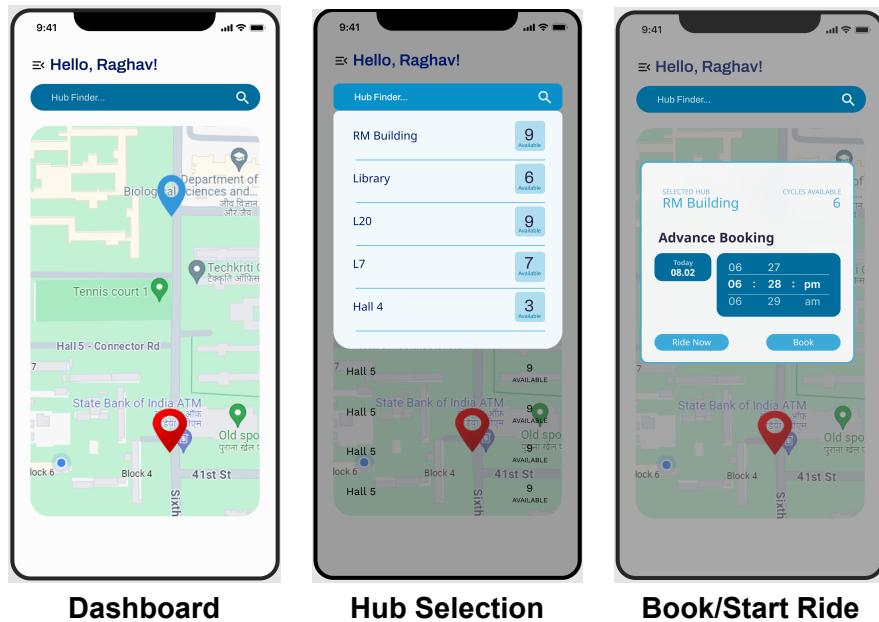
Login Page

Email for Password Reset

Reset Password

2.2.2 Viewing Hubs and Booking a Ride

The user sees a map which shows the location of the user along with the hubs present. They can either choose the hub on their map, or search it from the search box. Once a hub is selected, the details of the hub are displayed, including the number of available cycles. The user can choose to start a ride instantly, or book a cycle for a later time (available only for subscribed users). If the user wishes to book for later, they are presented with a time picker.



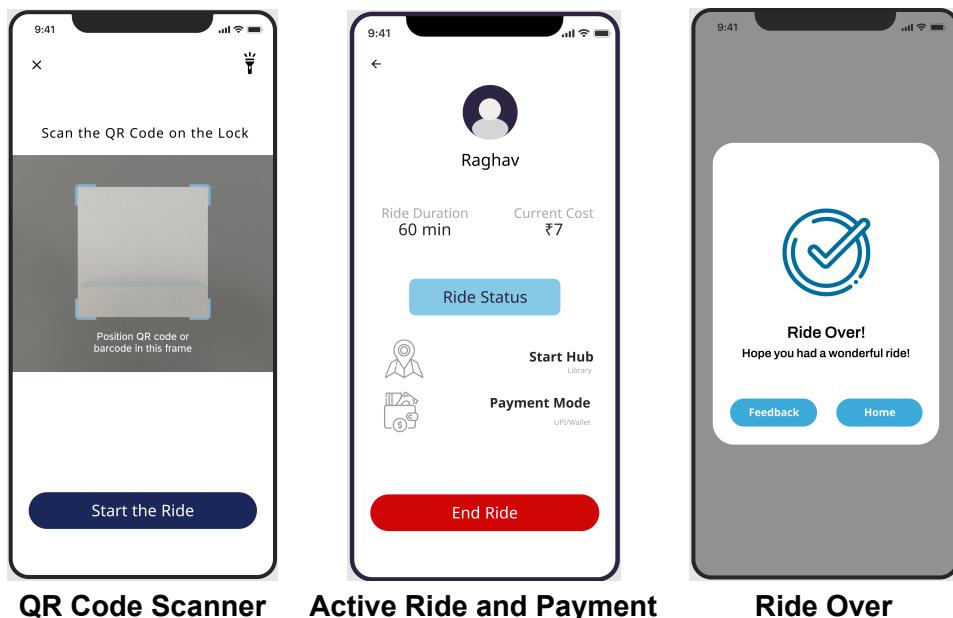
Dashboard

Hub Selection

Book/Start Ride

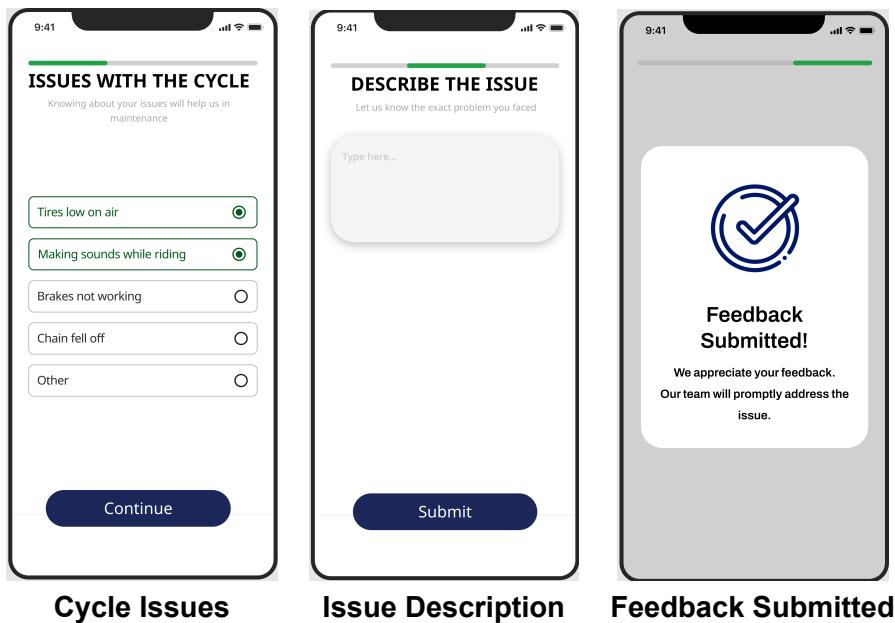
2.2.3 Starting a Ride

If the user wishes to ride instantly, they are presented with a QR code which they can scan to unlock the cycle. During the ride, statistics like ride time and current cost are shown to the user. Once the ride is over, the user can end the ride and the app will show the user the fare for the ride, and ask for feedback. The user can also report any issues with the cycle during the ride.



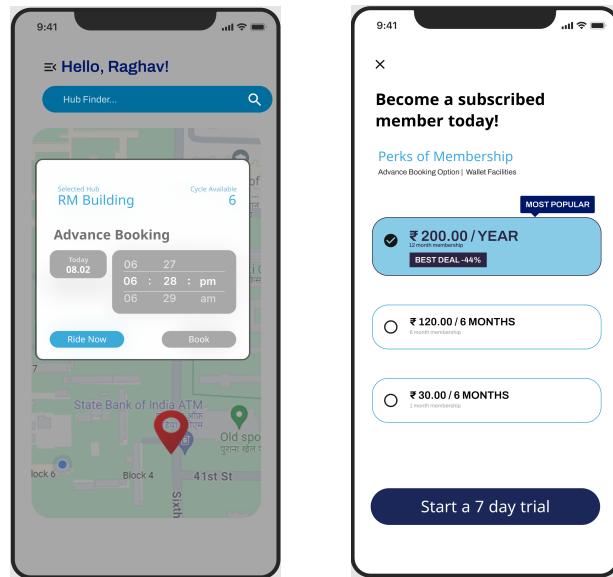
2.2.4 User Feedback

User has the option to give feedback on their riding experience and choose from various possible issues encountered during their ride and furnish a description thereof.



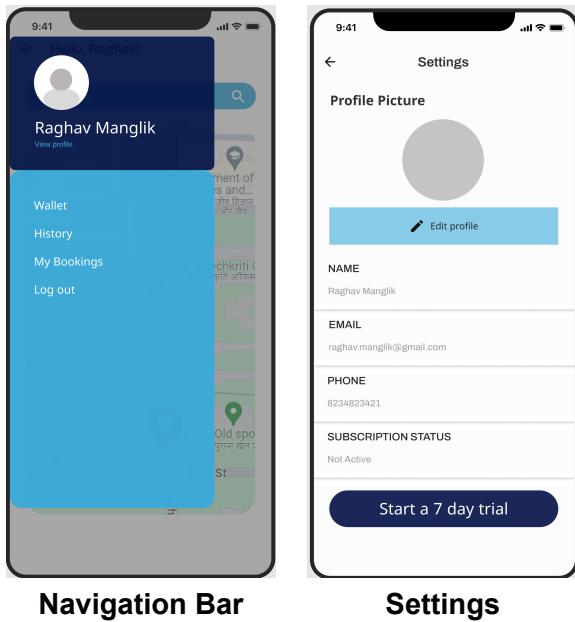
2.2.5 Subscription Model

The option of booking rides in advance is only available for subscribed users. This option is greyed out in case the user is not subscribed. If the user taps on any greyed out option, an advertisement for subscribing is shown.



2.2.6 Navigation Bar and Settings

The navigation bar offers users a selection of choices, including accessing their wallet details, viewing ride history, adjusting settings, managing bookings, and logging out. Within the settings section, users can modify personal information and also have the option to subscribe for additional features.

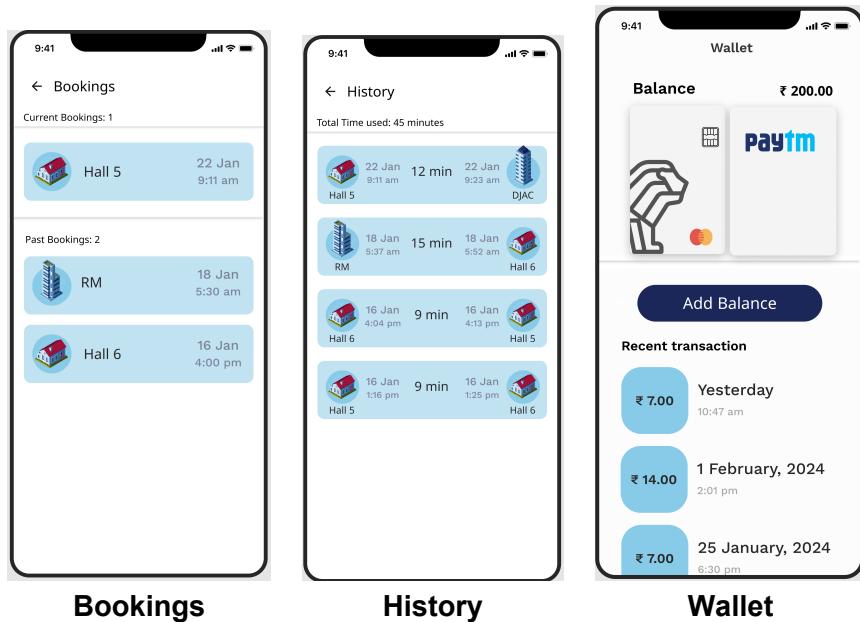


Navigation Bar

Settings

2.2.7 Analytics and Wallet

In the bookings page, users have access to both current and previous bookings, enabling them to track their activity. The history page provides users with a comprehensive overview of their ride history, detailing each ride's specifics. On the wallet page, users can monitor their balance and conveniently add funds to their wallet as needed.



Bookings

History

Wallet

2.2.8 Admin Interface

The admin interface serves as a control hub for managing both hubs and cycles. Admins possess the capability to introduce new hubs, inspect hub specifics, and regulate cycle operations. Furthermore, admins have access to comprehensive statistics, including total revenue metrics. They can also monitor user feedback and address reported issues effectively.



Admin Login



Admin Login

LOGIN

PEDALPAL

User Management

Average rating : 4.75/5 Bookings : 420 Total Revenue : ₹6699

Name	Booking Date	Booking Time	Payment	Other Details
Debraj	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>
Raghav	03/02/2024	11:45 - 13:45	₹14.00	<button>VIEW</button>
Khushi	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>
Shristi	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>
Srujan	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>
Nevish	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>
Kaneez	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>
Ananya	03/02/2024	12:25 - 13:25	₹7.00	<button>VIEW</button>

hector@email.com

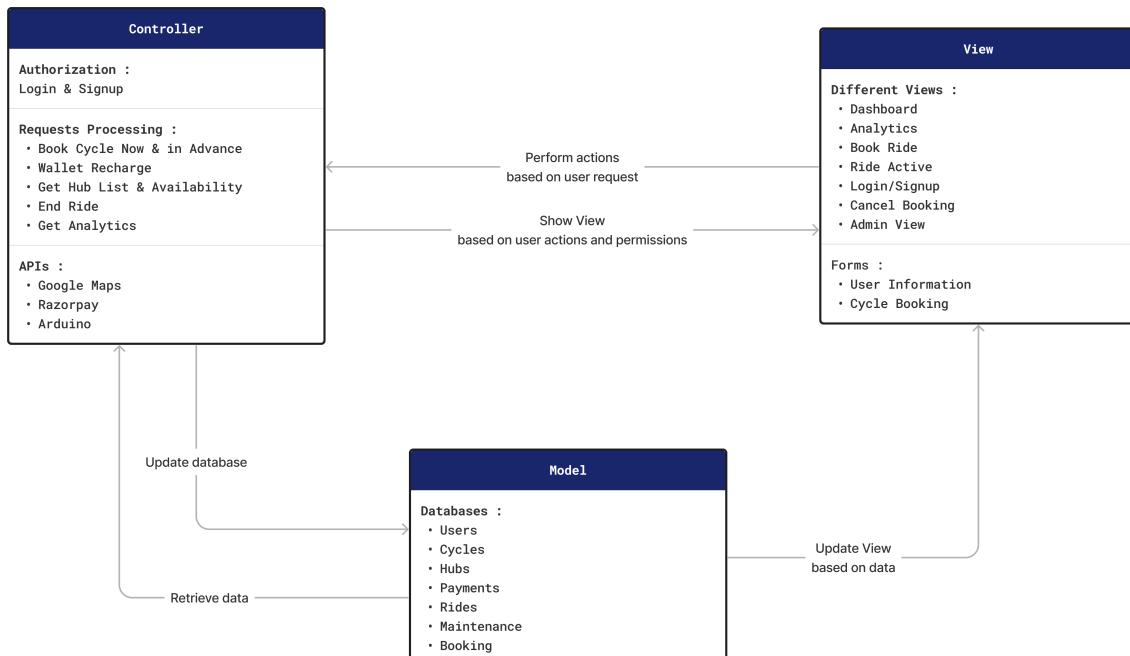
Admin Dashboard

3. Architecture Design

The overall architecture of our app follows the Model-View-Controller model. Given the complexity of the app, each functionality is sub-divided into smaller tasks, which are achieved by the Pipe-Filter architecture model.

The model-view-controller architecture is as follows:

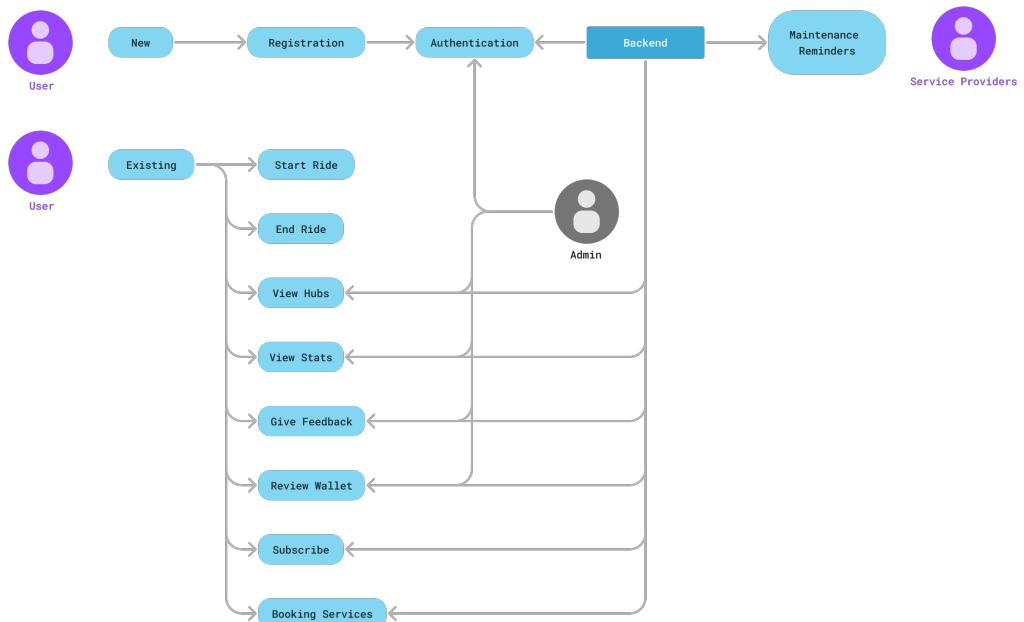
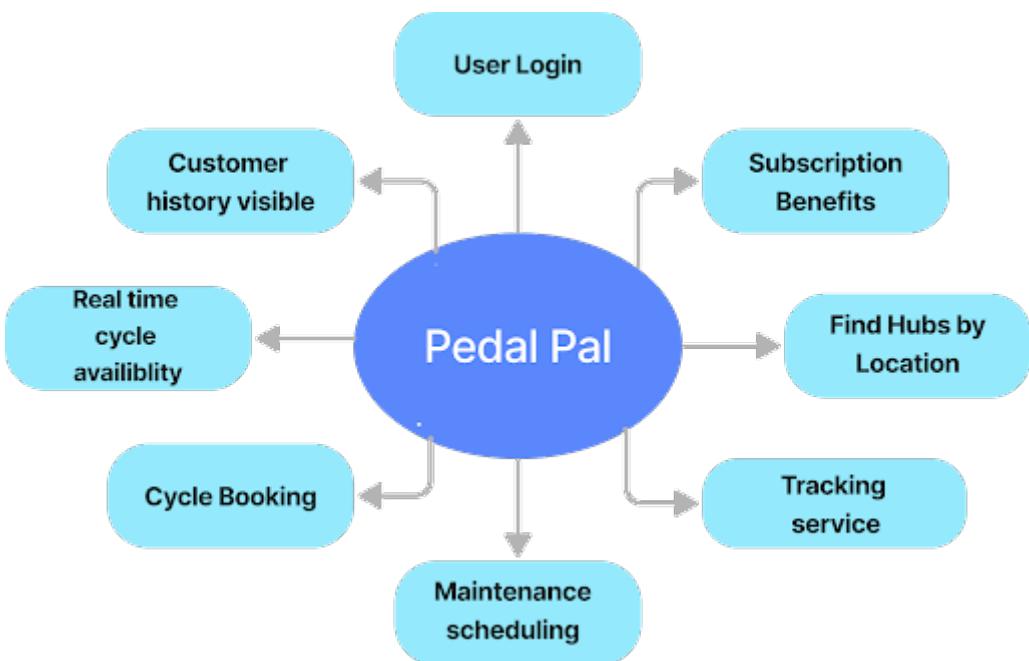
- **Model:** The model serves as the data centric component of the app, acting as the sole component directly interfacing with the database. Its primary role entails storing and retrieving data from the database. We plan to use PostgreSQL as our database, while using Django as our backend framework to interact with the database.
- **View:** The view acts as the presentation component of the app, responsible for rendering data to the user and handling user input. For app users, the view will be a mobile application, while for administrators, it will be a web application. The mobile app will be developed using Flutter while the admin webpage will be automatically generated by Django.
- **Controller:** The controller serves as the logic hub of the app, tasked with data processing and determining the app's flow. It will be developed using Django in Python. The integration of Google Maps API will provide map functionality, while RazorPay API will handle payment processes. For the hardware aspect of opening cycle locks, a microcontroller handled by Arduino will be employed.



We've opted for the **Model-View-Controller** (MVC) model because it accommodates our diverse data interaction needs. This model facilitates various operations such as viewing ride history, managing user data and bookings, and providing specialized admin views. Additionally, in the near future, one might think of adding some functionalities to it, which makes the Model-View-Controller model the best fit.

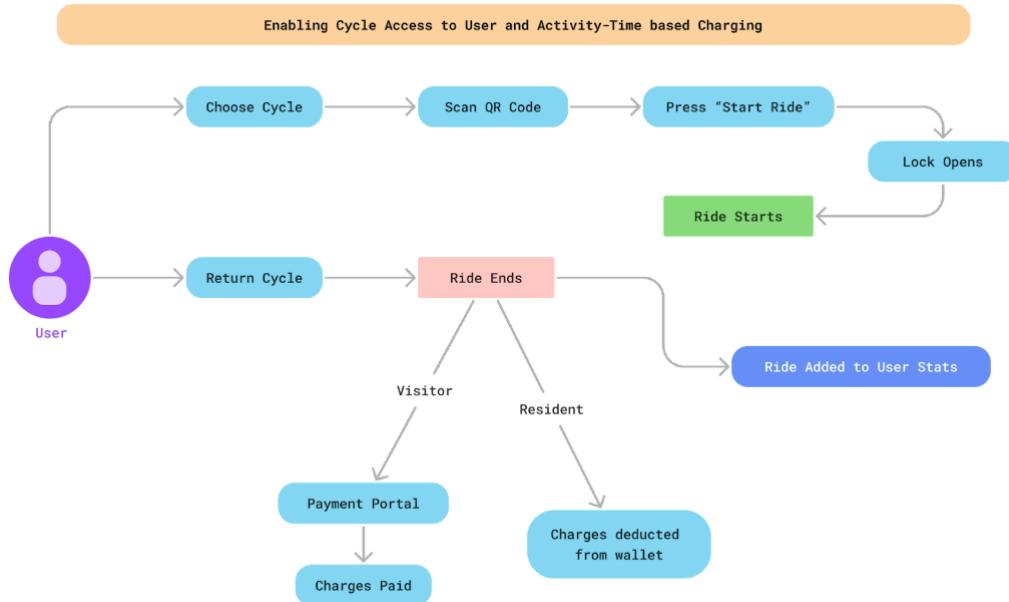
4. Object Oriented Design

4.1 Use Case Diagrams

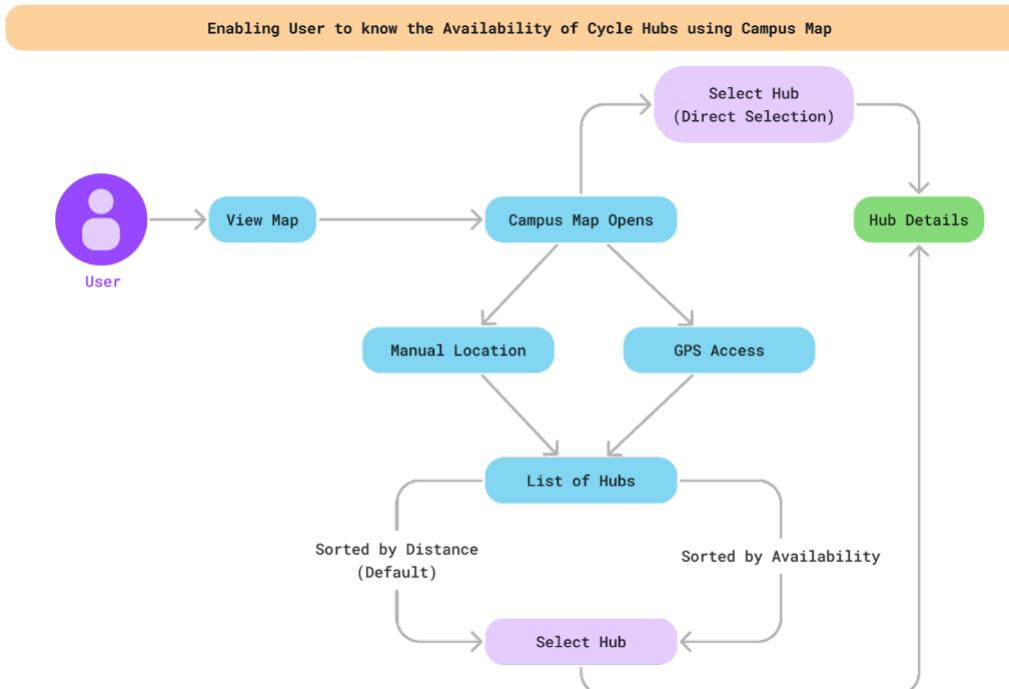


Overview of the Use Cases

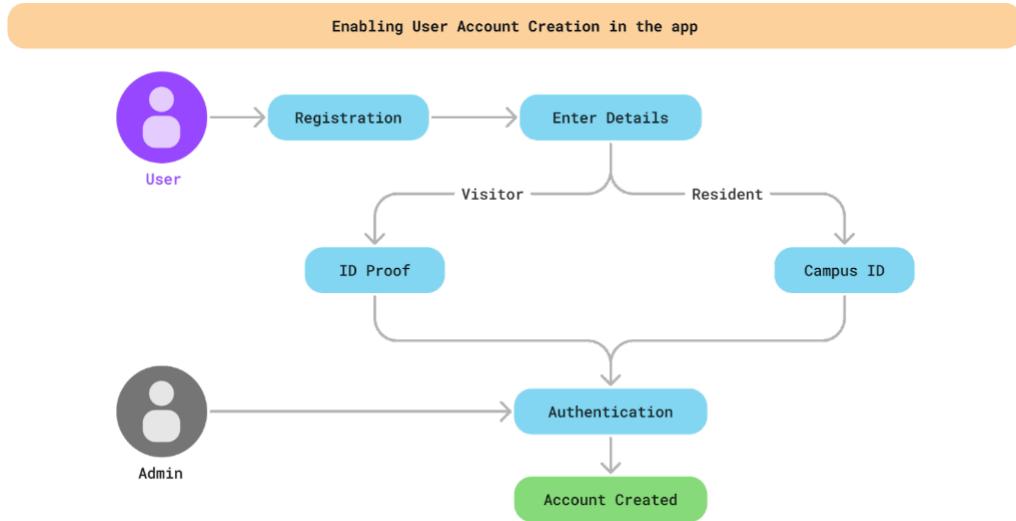
4.1.1 Use Case 1 (UC1) - Start Ride and End Ride



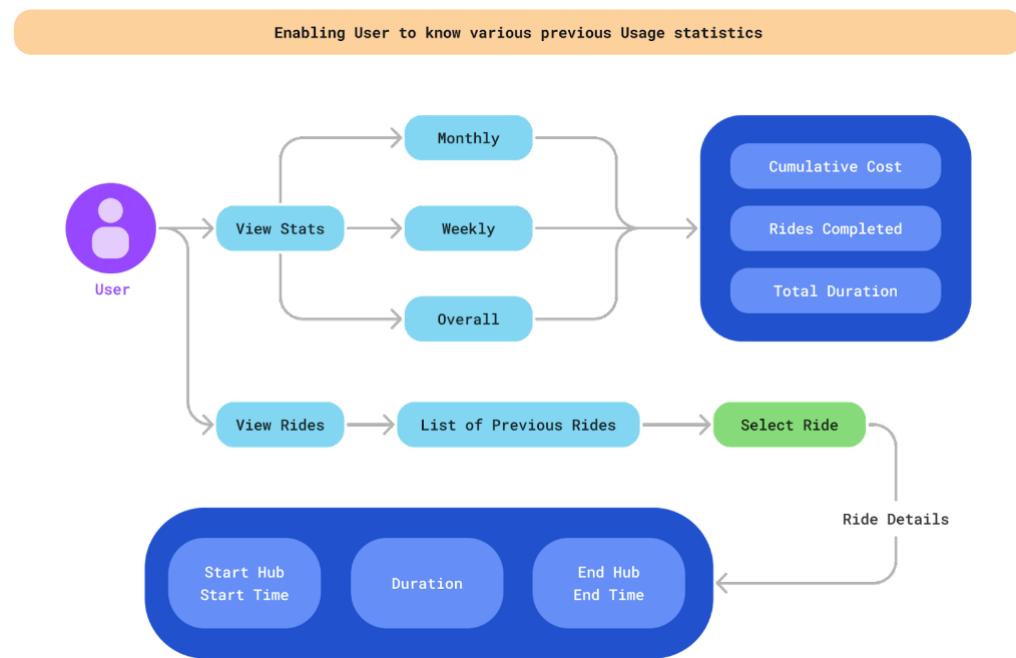
4.1.2 Use Case 2 (UC2) - View Nearby Hubs on a Map



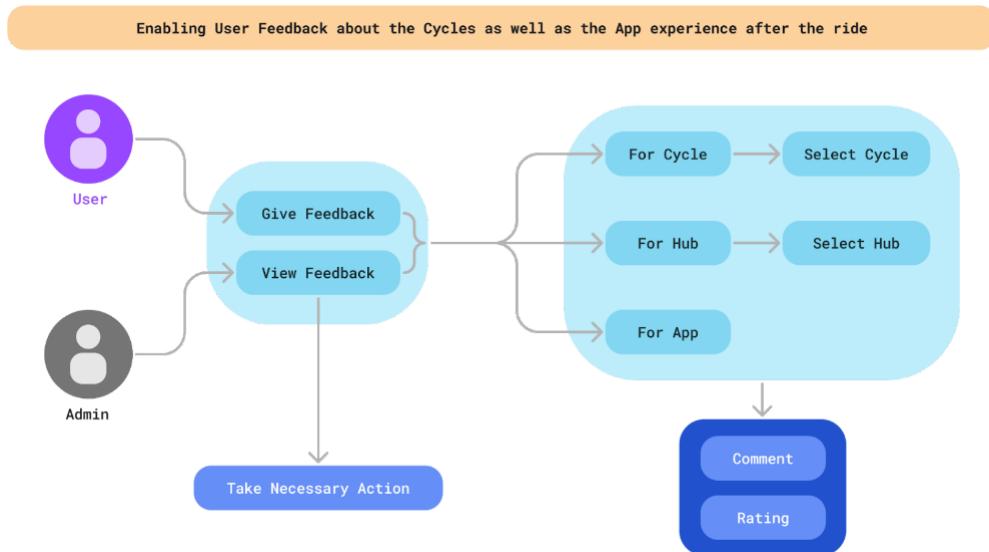
4.1.3 Use Case 3 (UC3) - Create a new Account



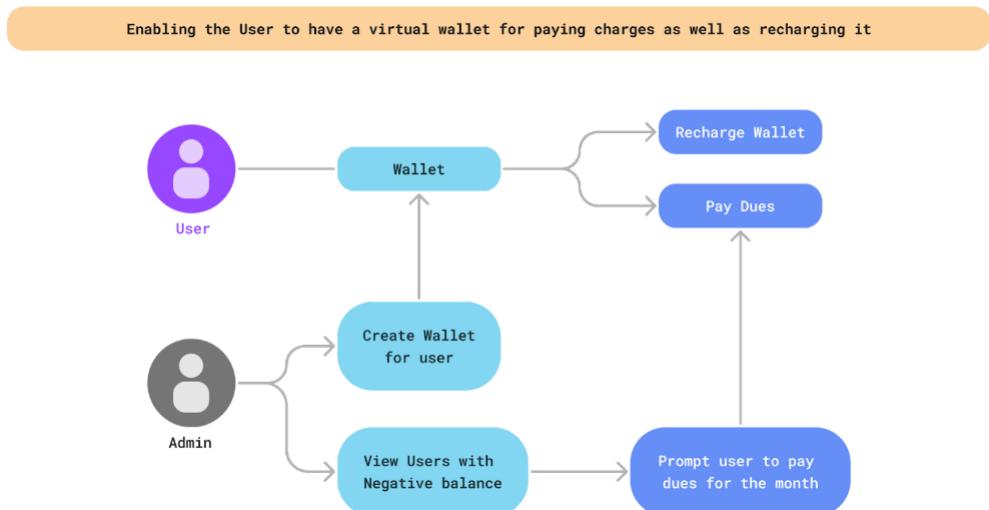
4.1.4 Use Case 4 (UC4) - View Analytics



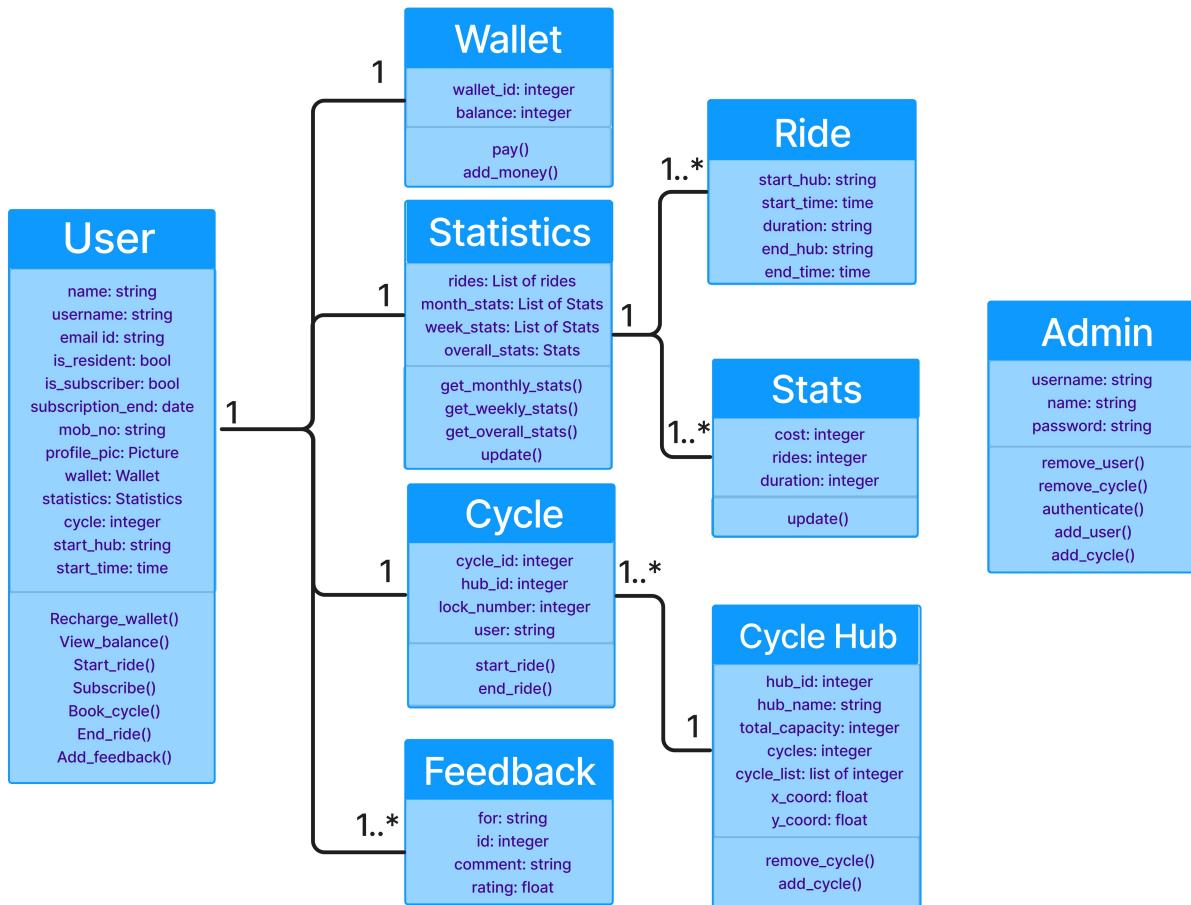
4.1.5 Use Case 5 (UC5) - Feedback Mechanism



4.1.6 Use Case 6 (UC6) - Use and Recharge Wallet



4.2 Class Diagrams

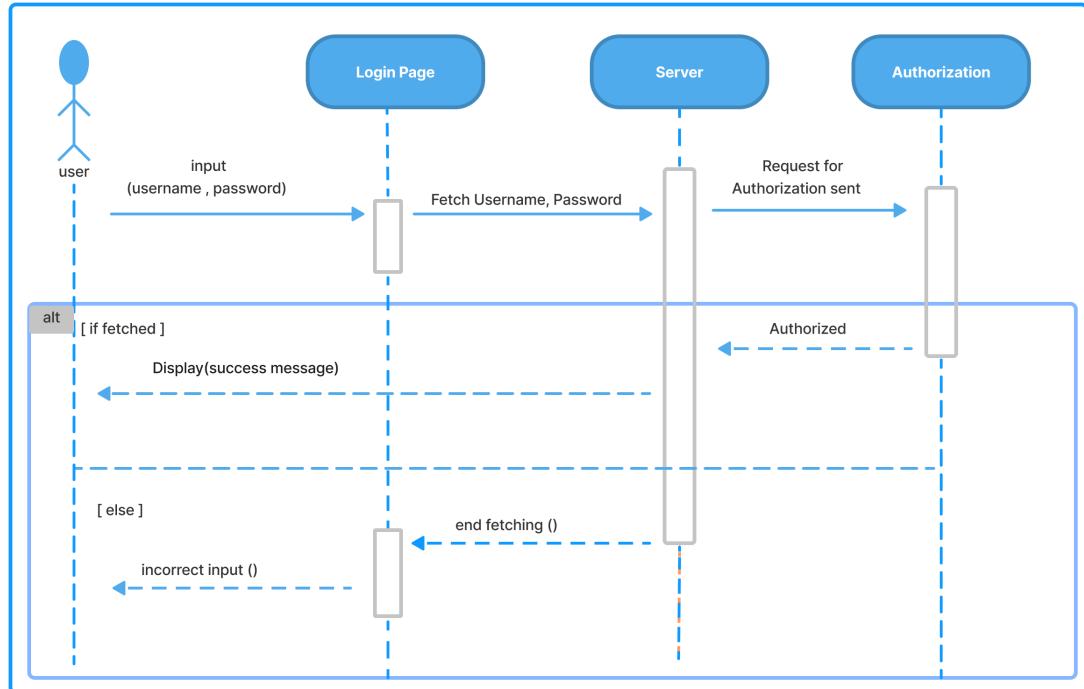


Class Diagram for PedalPal

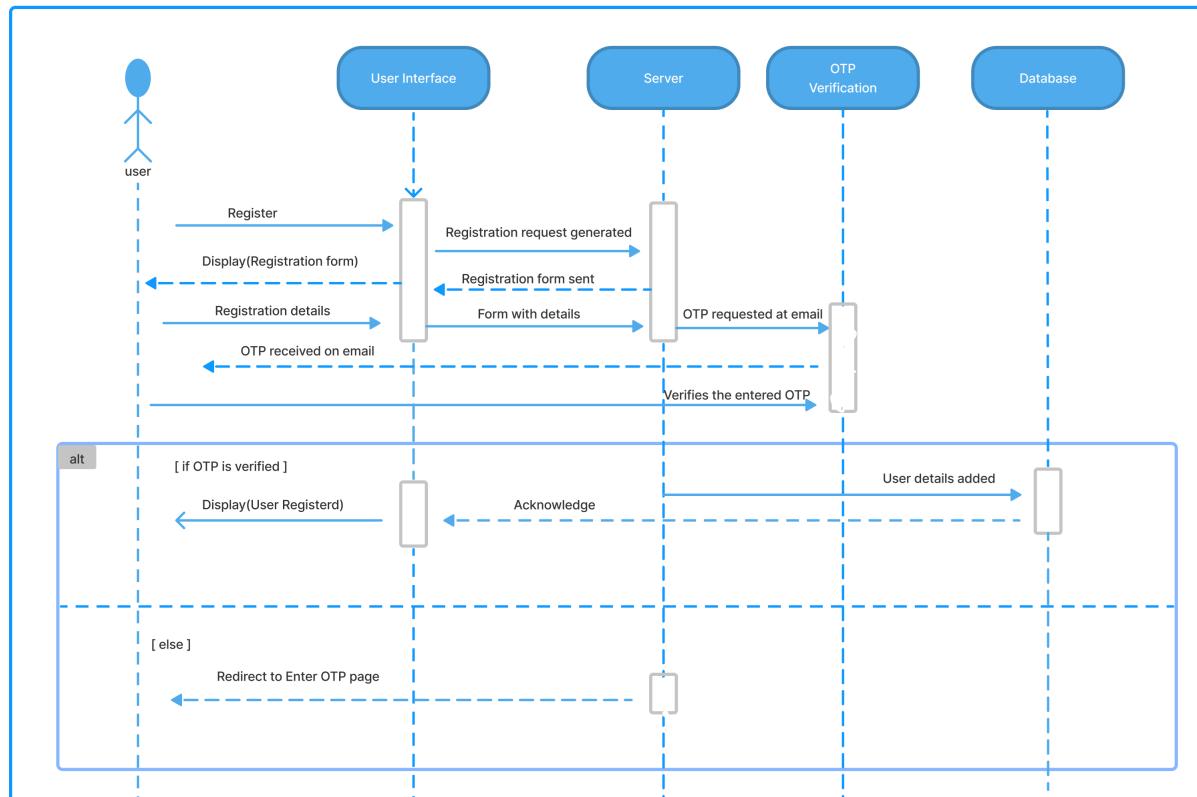
This is how all the objects in the project will interact with each other. Admin object can access all other objects.

4.3 Sequence Diagrams

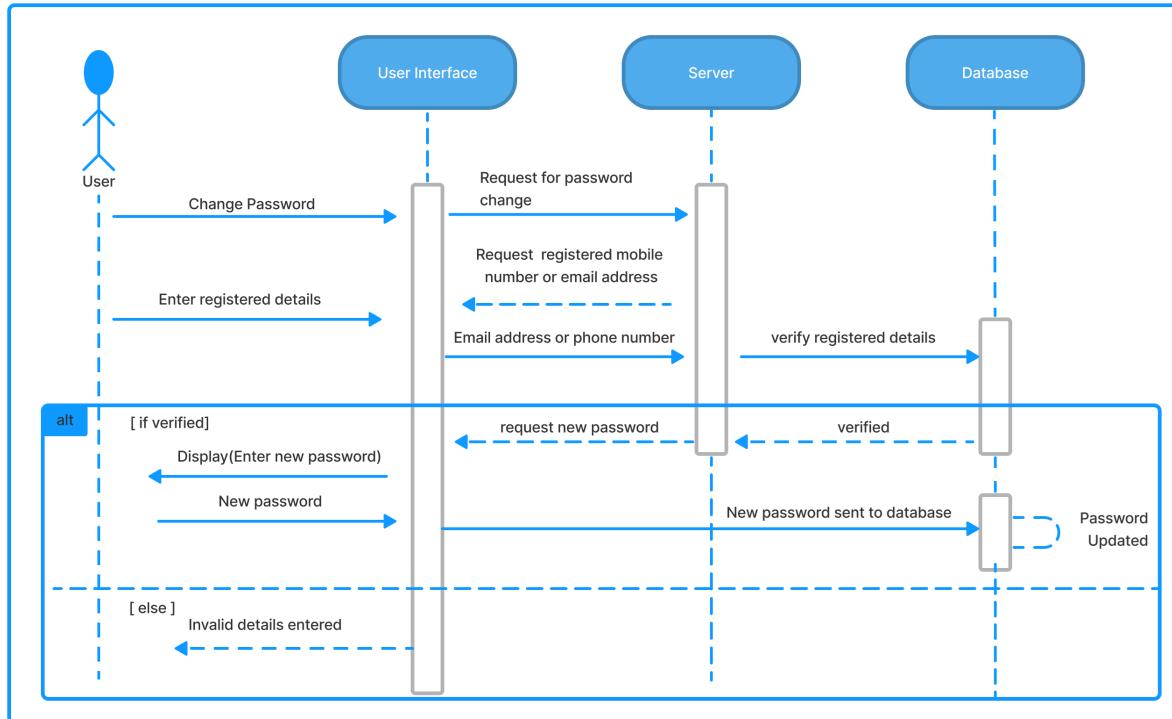
4.3.1 Login Sequence



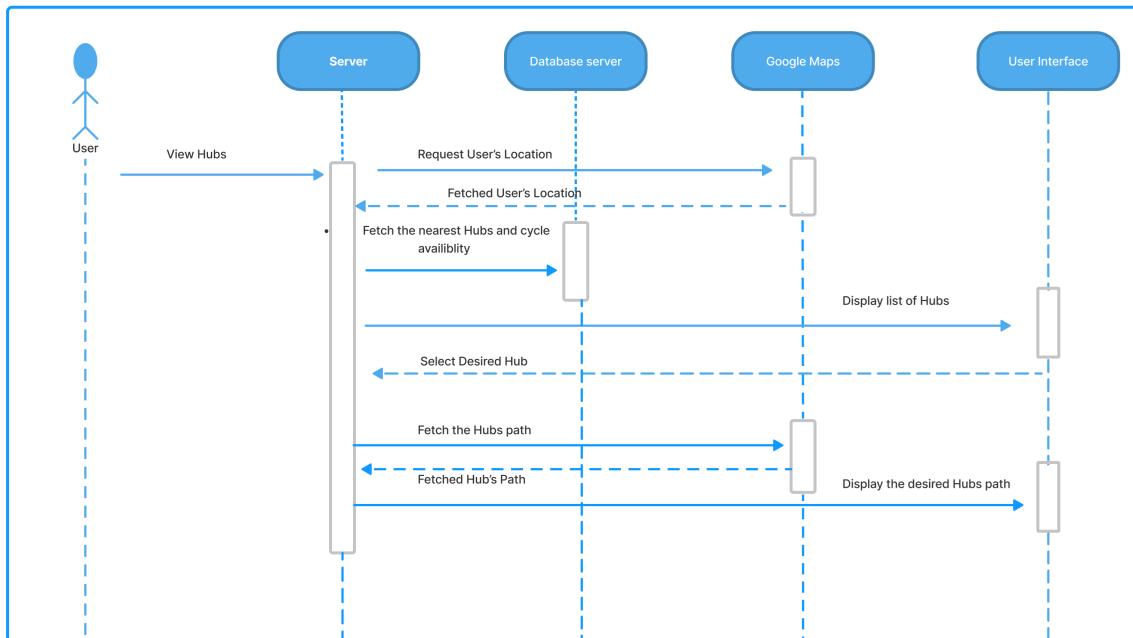
4.3.2 Registration Sequence



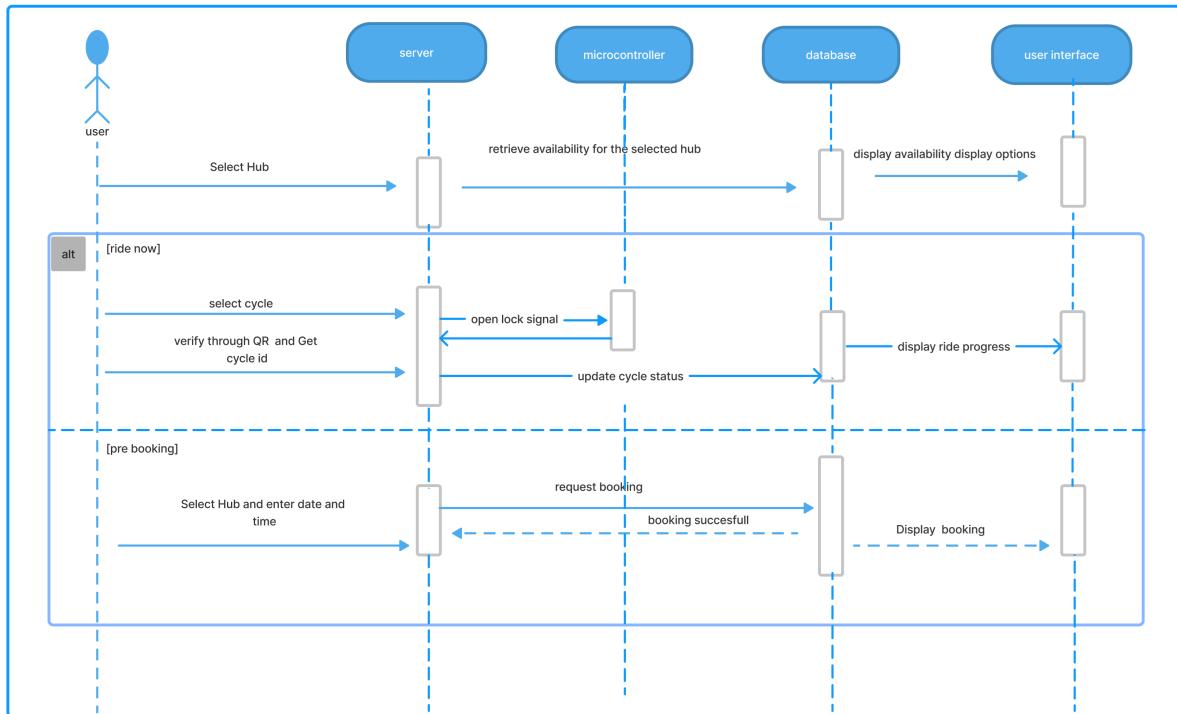
4.3.3 Reset Password Sequence



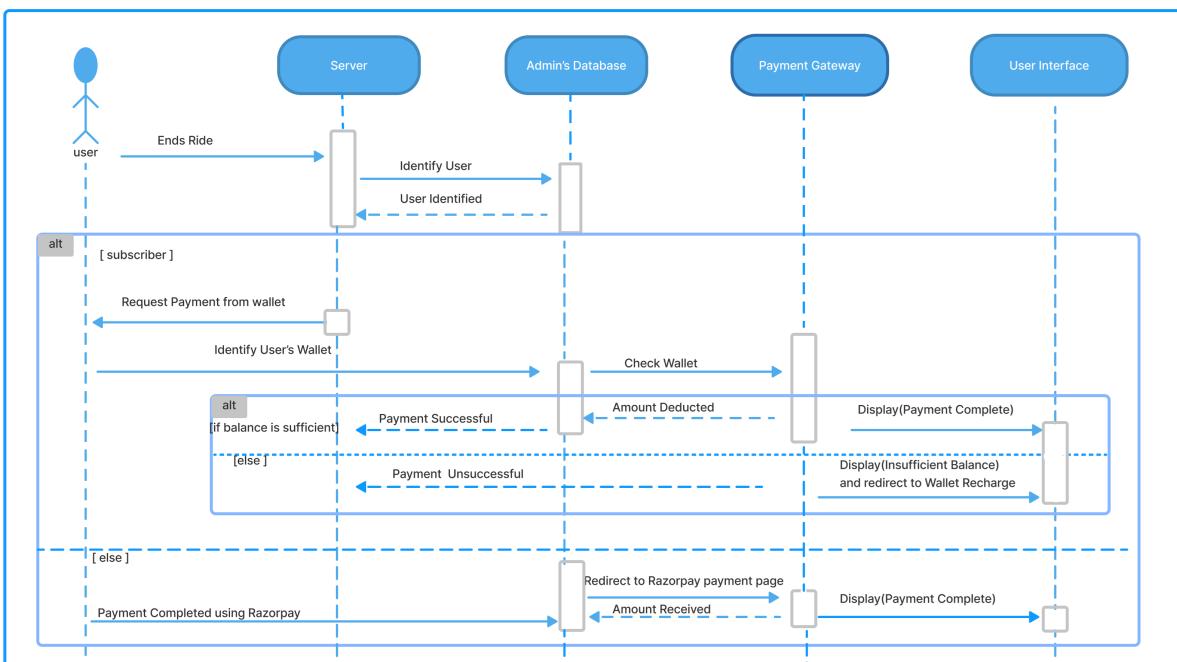
4.3.4 Fetch Cycle Hubs Sequence



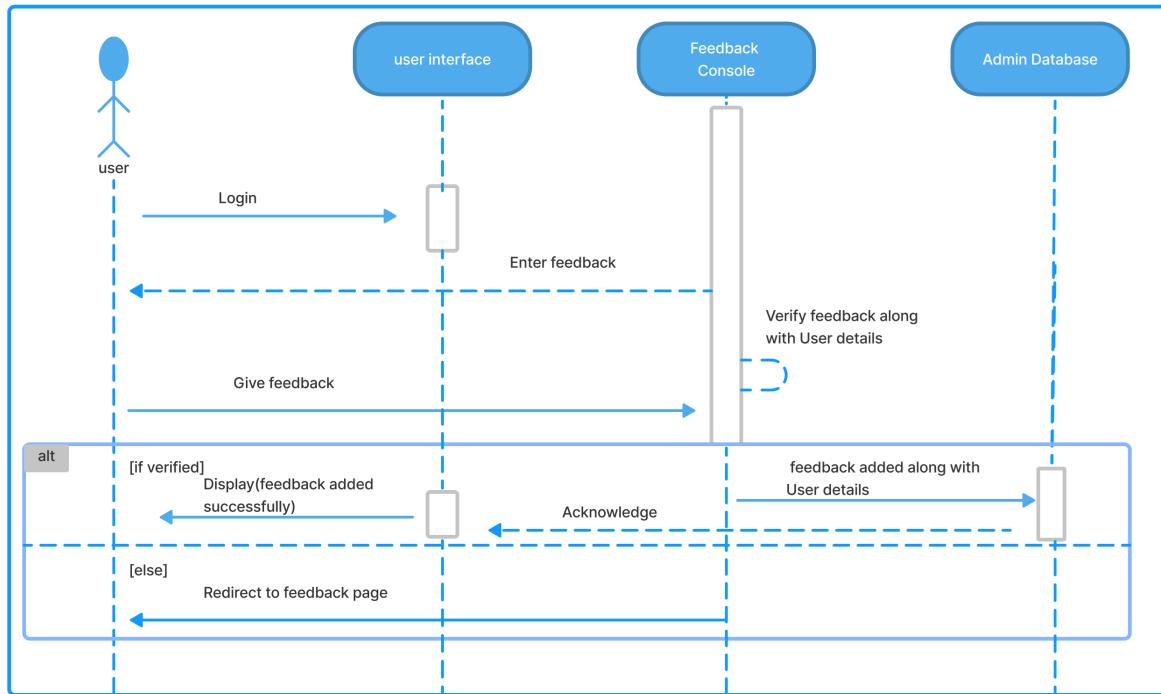
4.3.5 Book Ride Sequence



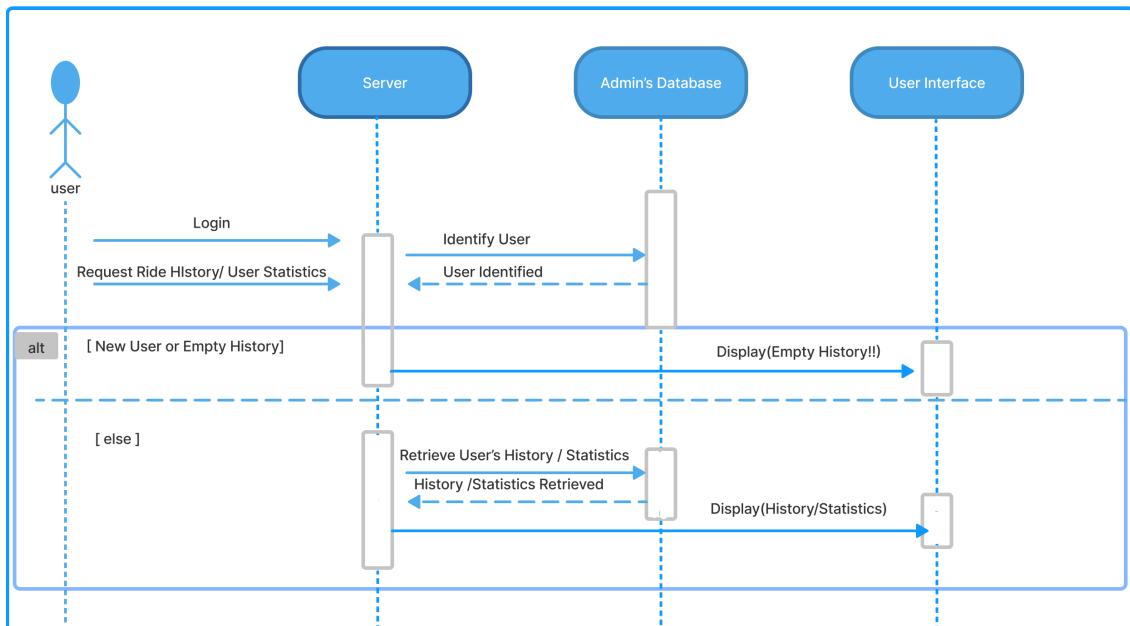
4.3.6 Payment Sequence



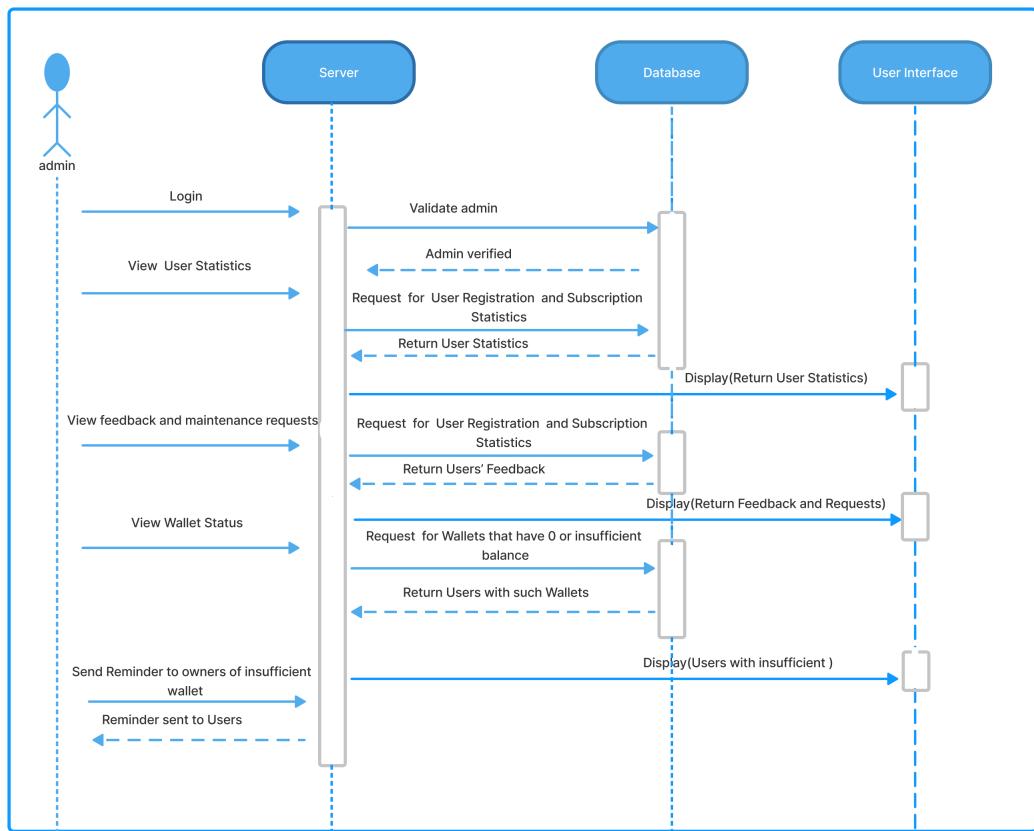
4.3.7 Feedback Sequence



4.3.8 View Analytics Sequence

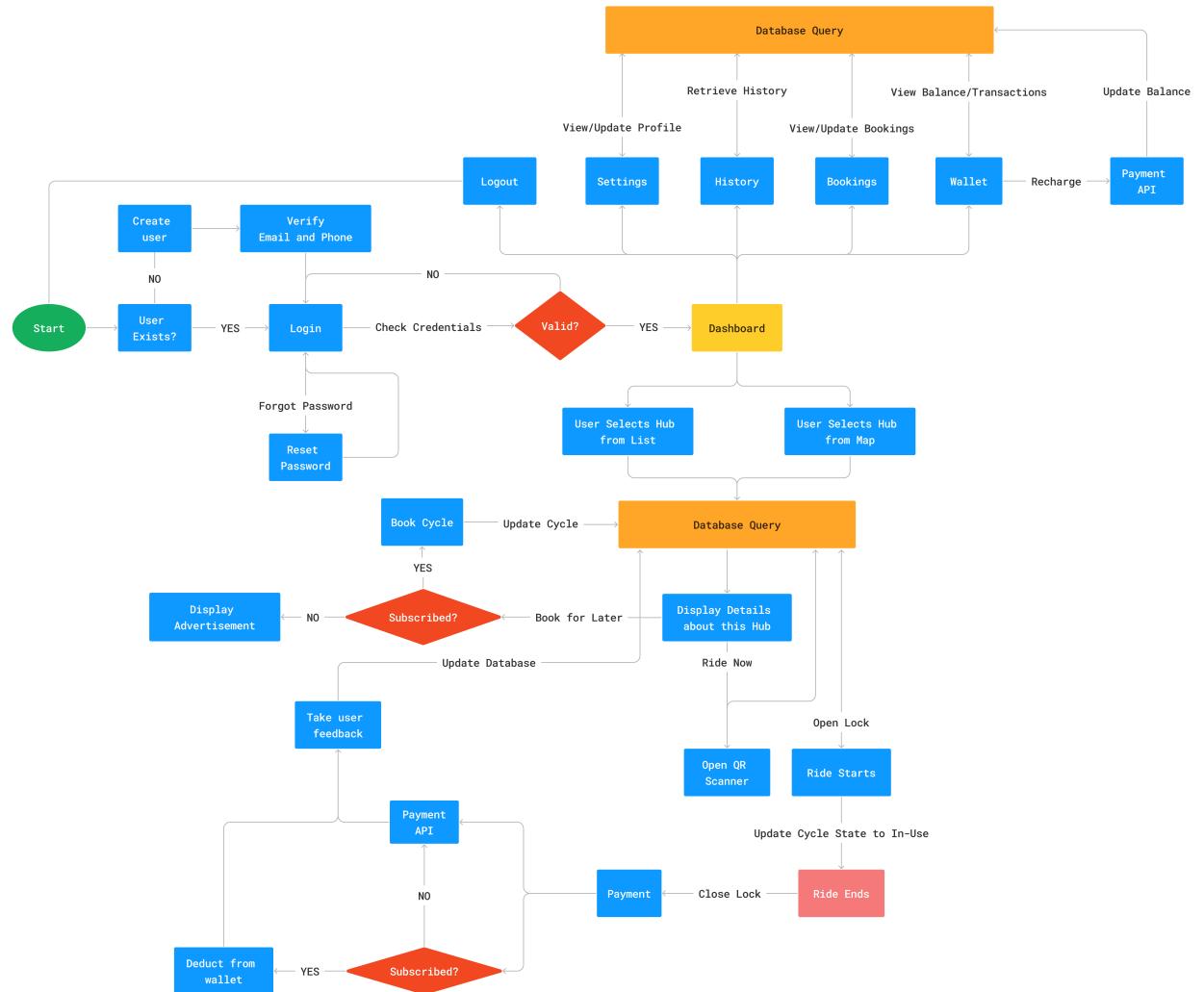


4.3.9 Admin Sequence



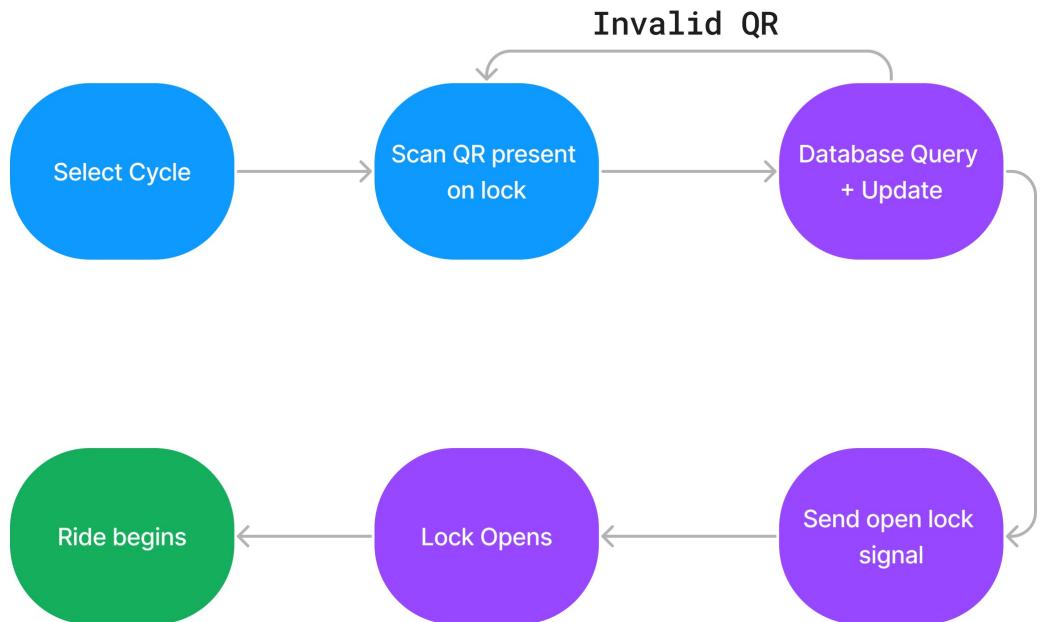
4.4 State Diagrams

4.4.1 Overall state diagram

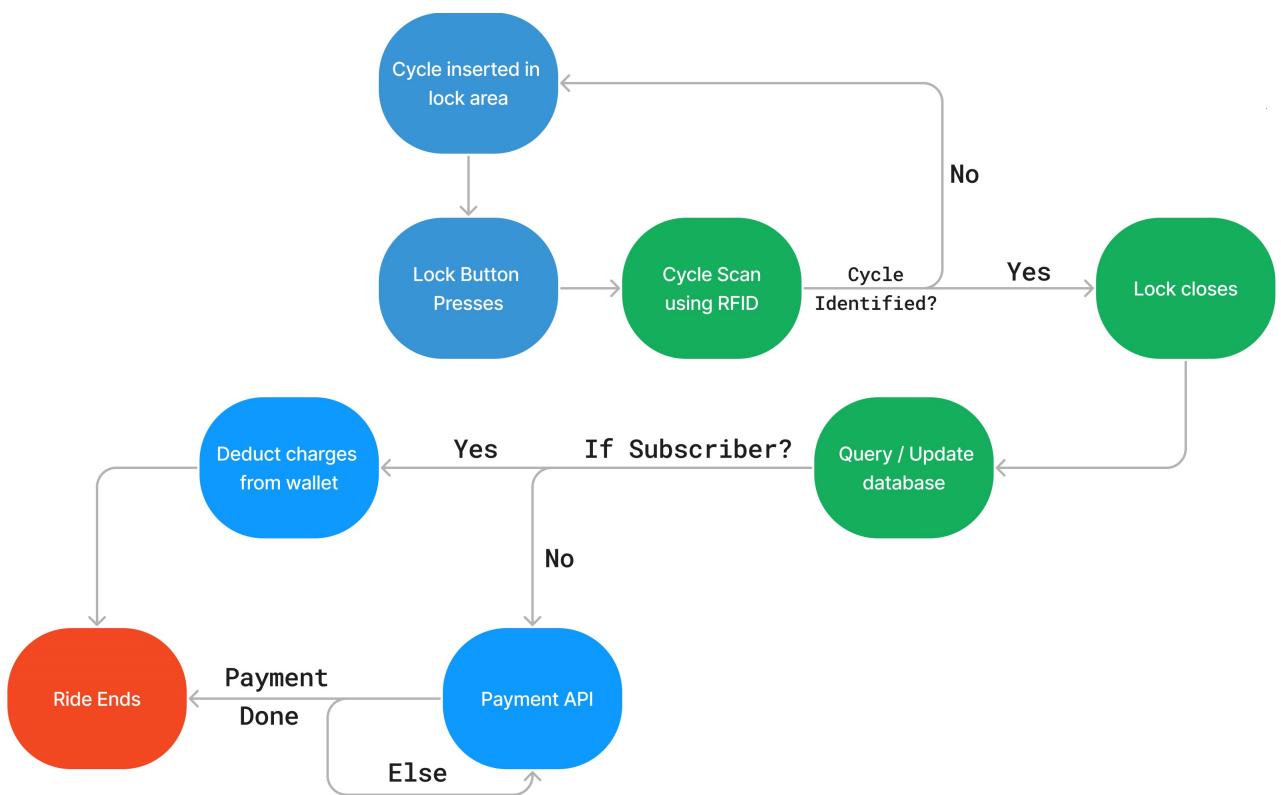


The image describes the high-level state diagram of the application from a user perspective. The customer can either be a subscribed customer or an unsubscribed customer. The customer can book a ride, view ride history, view wallet, view settings, view booking history, and give feedback. The orange boxes show a query to the database and the red diamonds show a decision point.

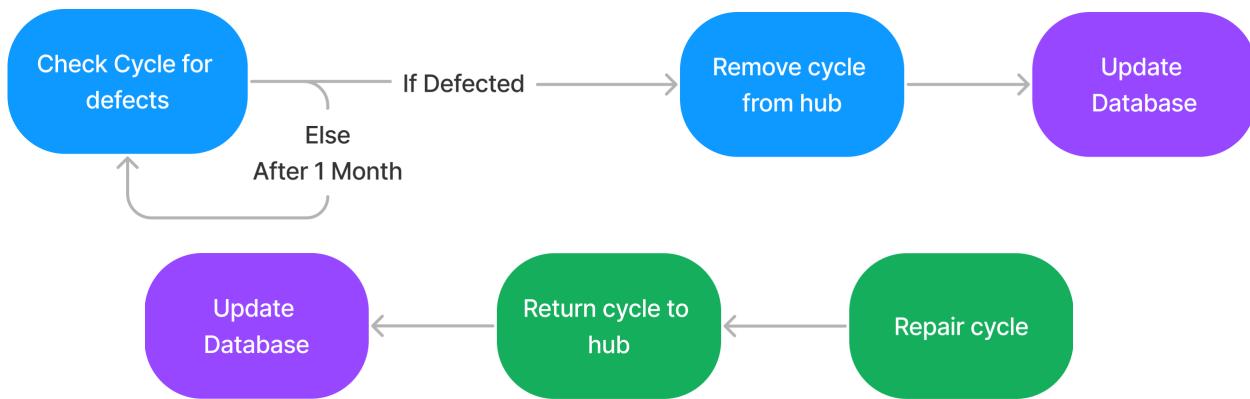
4.4.2 Ride Begin state diagram



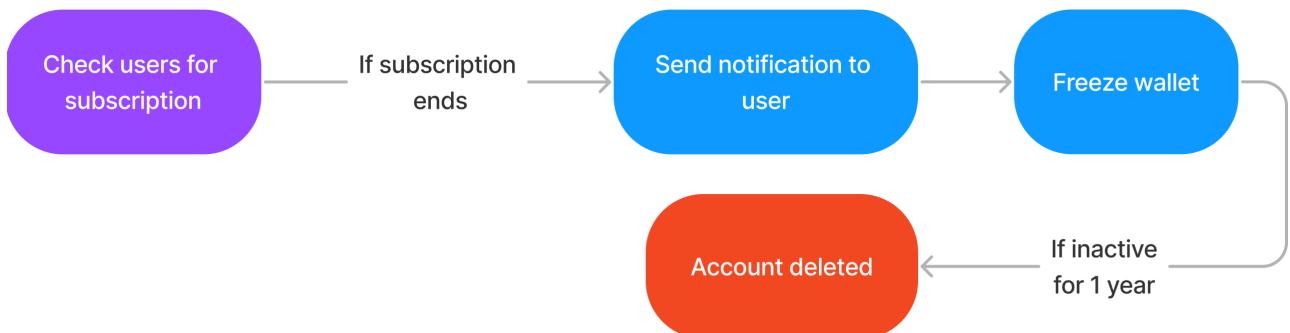
4.4.3 Ride End state diagram



4.4.4 Maintenance State Diagrams



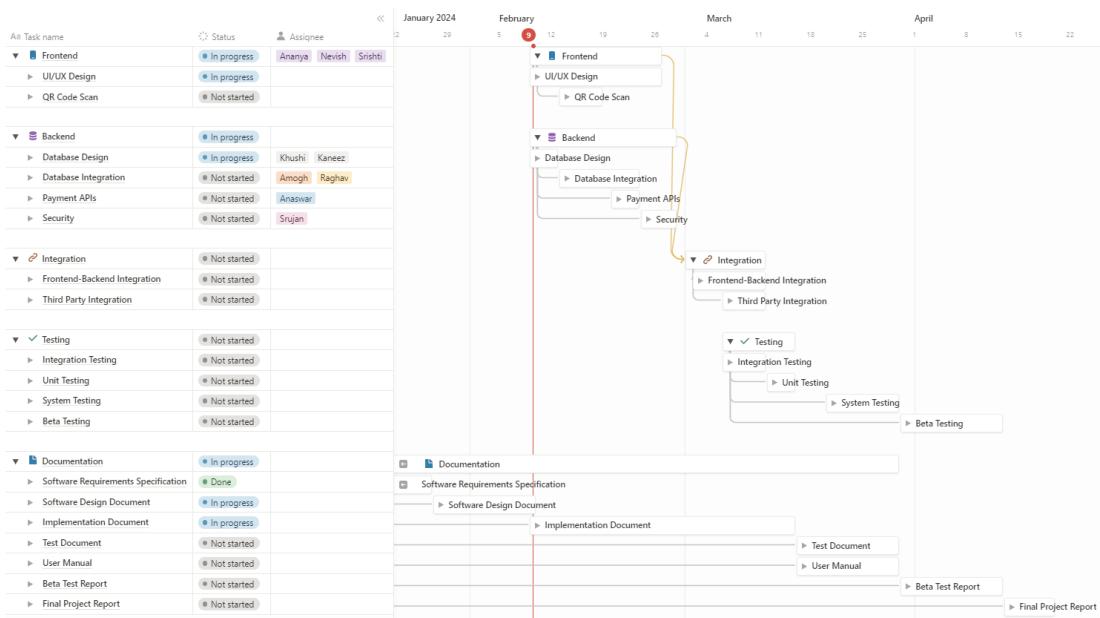
4.4.5 Account Closing



5. Project Plan

5.1 Project Planning

We are using Notion to track the progress and deadlines of the project. It also allows us to assign tasks to team members. Every team member has internally taken the responsibility to focus majorly on either frontend or the backend. The current Gantt chart looks like this:



5.2 Code Collaboration

We are using GitHub to collaborate on the code. An organization is created for the purposes of this project and all team members are added to this organization. Separate repositories are maintained for the backend and frontend of our app.

Pedal-Pal-CS253

View organization

Browse organization's repositories

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Khushi07g pushed to Khushi07g/pedal-pal-backend - 1 hour ago

3 commits to main

3c89b4a django project
f35281b backend
1 more commit »

ANNNNA106 forked ANNNNA106/pedal-pal-backend from Pedal-Pal-CS253/pedal-pal-backend - 1 hour ago

Pedal-Pal-CS253/pedal-pal-backend

Repository containing implementation of the back-end of PedalPal

Updated Feb 9

Khushi07g forked Khushi07g/pedal-pal-backend from Pedal-Pal-CS253/pedal-pal-backend - 2 hours ago

Pedal-Pal-CS253/pedal-pal-backend

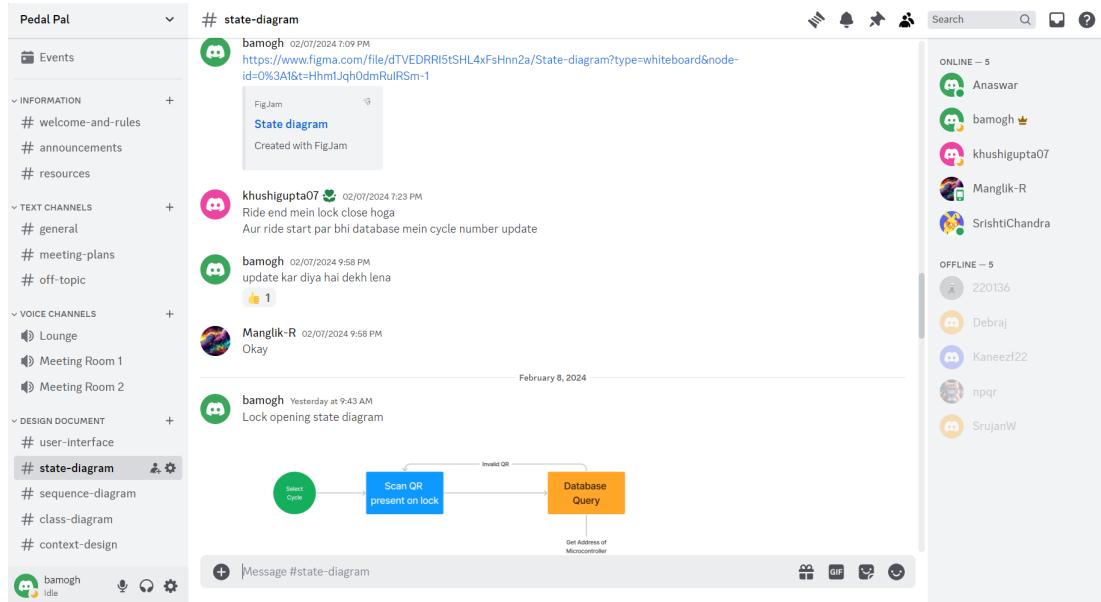
Repository containing implementation of the back-end of PedalPal

Updated Feb 9

AmoghBhagwat created a branch in Pedal-Pal-CS253/pedal-pal - 5 hours ago

5.3 Communication

We are using Discord for communication. It allows us to create separate channels for different topics, and keep data organised. It also allows us to have voice calls and video calls, which is useful for team meetings.



5.4 Work Division

Team Member	Responsibility
Amogh Bhagwat	Backend, Integration of frontend and backend
Ananya Singh Baghel	Frontend, Integration of frontend and backend
Anaswar K B	Backend, Testing
Debraj Karmakar	Backend, Hardware
Kaneez Fatima	Frontend, Testing
Khushi Gupta	Backend, Database Management
Pathe Nevish Ashok	Frontend, UI/UX Design
Raghav Manglik	Backend, Database Management
Srishti Chandra	Frontend, UI/UX Design
Wadkar Srujan Nitin	Backend, Testing

A. Group Log

S.No	Date	Timings	Venue	Description
1	31/01/2024	21:30 to 23:00	Discord	Studied deliverables for the design document. Divided work among teammates
2	03/02/2024	11:00 to 14:00	RM Building	Designed user interface Discussed project details like frameworks and other APIs
3	07/02/2024	21:30 to 22:00	Google Meet	Meet with TA to discuss details about the document.
4	08/02/2024	14:30 to 17:00	RM Building	Finalized all aspects of the design document Decided future plans