Software Requirements Specification

for

Fleet

Version 1.0

Prepared by

Group #: 11 **Group Name: Null Pointers**

Habeeb Ramith Kumar	230432	hramith23@iitk.ac.in
Reddi Pallavi	230850	rpallavi23@iitk.ac.in
Pasala Bosu Akil Teja	230742	pbosuateja23@iitk.ac.in
Shashi Bhidodiya	230956	shashib23@iitk.ac.in
Manam Amara Gayathri	230624	mamara23@iitk.ac.in
Saiprabhav	230060	addulasr23@iitk.ac.in
Koneti Karthik	230568	konetik23@iitk.ac.in
Poorvie Sadagopan	230759	poorvies23@iitk.ac.in
Pittala Sruthi	230751	sruthip23@iitk.ac.in
Jyothika Seru	230946	serujy23@iitk.ac.in

Course: CS253

Mentor TA: Souvik Mukherjee

Date: 24/01/2025

CONTENTS ..... **II**

REVISIONS ..... **II**

1 INTRODUCTION **1**

1.1 PRODUCT SCOPE 1
1.2 INTENDED AUDIENCE AND DOCUMENT OVERVIEW 1
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 2
1.4 DOCUMENT CONVENTIONS 2
1.5 REFERENCES AND ACKNOWLEDGMENTS 2

2 OVERALL DESCRIPTION **3**

2.1 PRODUCT OVERVIEW 3
2.2 PRODUCT FUNCTIONALITY 4
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS 4
2.4 ASSUMPTIONS AND DEPENDENCIES 4

3 SPECIFIC REQUIREMENTS **5**

3.1 EXTERNAL INTERFACE REQUIREMENTS 5
3.2 FUNCTIONAL REQUIREMENTS 8
3.3 USE CASE MODEL 8

4 OTHER NON-FUNCTIONAL REQUIREMENTS **19**

4.1 PERFORMANCE REQUIREMENTS 19
4.2 SAFETY AND SECURITY REQUIREMENTS 19
4.3 SOFTWARE QUALITY ATTRIBUTES 19

5 OTHER REQUIREMENTS **21**

APPENDIX A – DATA DICTIONARY **22**

APPENDIX B - GROUP LOG **24**

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Habeeb Ramith Kumar Reddi Pallavi Pasala Bosu Akil Teja Shashi Bhidodiya Manam Amara Gayathri Saiprabhav Koneti Karthik Poorvie Sadagopan Pittala Sruti Jyothika Seru	First version of the requirement document	24/01/25

1 Introduction

1.1 Product Scope

Renting a vehicle can often be a cumbersome process, involving lengthy searches, manual bookings, and unclear payment systems. Motivated by the need for a modern solution, we decided to build Fleet, a platform designed to simplify vehicle rentals by connecting vehicle owners with renters.

With Fleet, users can effortlessly browse through available vehicles, make secure bookings, and process payments, all from a single, user-friendly interface. Administrators can efficiently manage vehicle inventories, monitor transactions, and address customer needs, making the system both comprehensive and convenient.

One of the platform's key features is its ability to cater to diverse customer preferences, whether they need a car with or without a driver. It also includes feedback systems to ensure quality service and customer satisfaction.

We believe Fleet will not only streamline vehicle rentals but also redefine the rental experience for both customers and administrators.

1.2 Intended Audience and Document Overview

This document is intended for developers, project managers, marketing staff, testers and end-users. Developers will focus on functional and non-functional requirements. Project managers will use it to understand the scope, deliverables, and resources. Marketing staff will assess alignment with business goals. Testers will verify requirements, and end-users will explore features and benefits.

1.2.1 Document Overview*:*

- **Introduction***: Overview of the project goals and the system's purpose and scope.*
- **System Overview***: High-level description of the system and its design constraints.*
- **External Interface Requirements***: Interaction details with external services like payments and notifications.*
- **Functional Requirements*:*** *Detailed behaviors for customers and administrators.*
- **Non-Functional Requirements*:*** *Performance, security, scalability, usability, and reliability expectations.*

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

- **Vehicle Rental System -** *A software application that allows users to search, reserve, rent, and return vehicles, including cars, motorcycles, and other types of vehicles, based on their preferences and availability.*
- **User -** *Any individual that interacts with the Vehicle Rental System, including customers, administrators, and system operators.*
- **Customer -** *A user who rents a vehicle through the system. Customers can search for vehicles, view availability, make bookings, and manage their rentals.*
- **Reservation -** *A booking made by a customer to secure a vehicle for a specific period. A reservation includes details such as the vehicle selected, rental duration, and pickup and drop-off locations.*

1.3.2 Acronyms

- **VRS** – *Vehicle rental system*
- **UI** – *User interface*
- **SRS –** *Software requirements specification*
- **SMS –** *Short Message Service*
- **OTP –** *One-Time Password*

1.3.3 Abbreviations

- ***Admin**. - Administrator*
- ***Req**. - Requirement*
- ***Res**. - Reservation*
- ***Acc**. - Account*

1.4 Document Conventions

General Text: *Font – Arial Size – 12 Style – Italic*

Heading: *Font – Arial Size – 14 Style – Bold*

Sub Heading: *Font – Arial Size – 12 Style – Bold*

Margin-*1”*

1.5 References and Acknowledgments

https://www.discovercars.com

Home - Canva

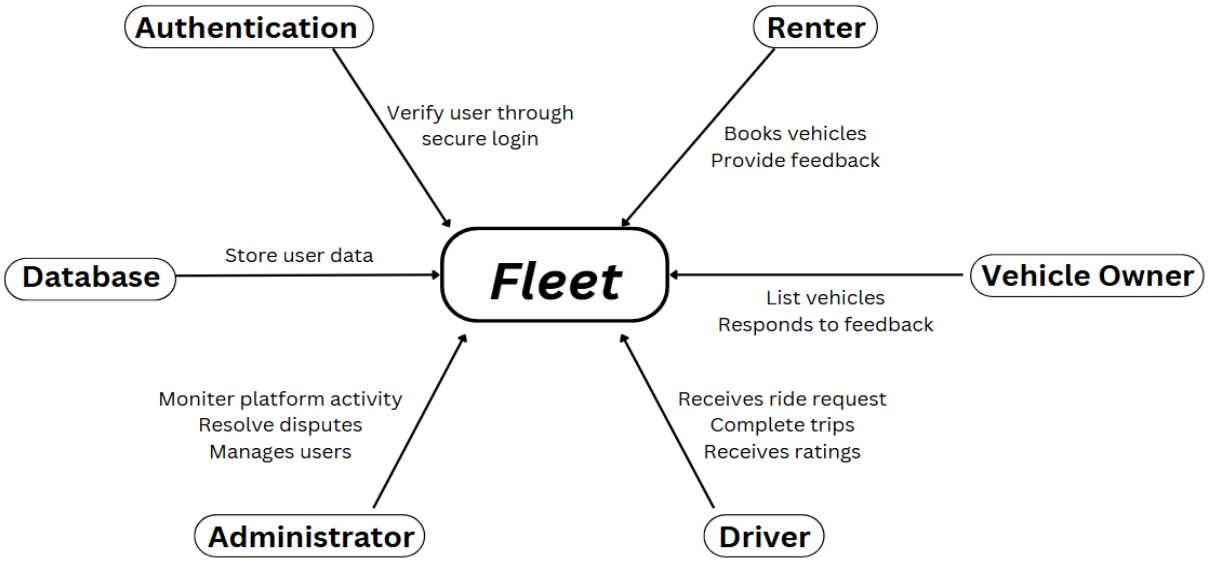
2 Overall Description

2.1 Product Overview

The Fleet platform is a new, standalone web application designed to make vehicle rentals easier by connecting vehicle owners with renters. It offers features like vehicle listings, secure bookings, payment processing, and feedback, all within an easy-to-use interface. Built from scratch, it meets the current needs of vehicle rental services and is not a part of an existing product family.

The Fleet platform is a new, self-contained web-based application developed to simplify the process of renting vehicles by connecting vehicle renters with customers. It is designed to offer a comprehensive range of features, including vehicle listings, secure booking, payment processing, and feedback systems, all integrated into a user-friendly interface. The platform is not a follow-on member of a product family or a replacement for existing systems but is developed from the ground up to address the modern needs of vehicle rental services.

Fleet operates independently but can connect with other services to provide features like instant booking confirmations and secure payment processing. The platform's backend processes all the data and business logic, ensuring seamless interaction between different components such as user authentication, vehicle management, and booking systems.



2.2 Product Functionality

- *User Registration and Login*
- *User-Friendly Interface*
- *Admin. Login*
- *Vehicle Listing*
- *Vehicle Booking*
- *Driver Management*
- *Secure Booking and Payment*
- *Feedback System*

2.3 Design and Implementation Constraints

- *Limited Server Capacity: The platform's server has a maximum limit on how many users can register, log in, and book vehicles at the same time.*
- *Language Limitation: As of now, users will only be able to interact with the platform in English.*
- *Search Limitation: The system will limit the length of any search term to (30) characters and only allow standard English characters.*
- *Booking and Payment System: As of now, there is a restriction in the methods users can use to pay for their rentals.*
- *Vehicle and Driver Listings: If there are not enough drivers registered, users may not be able to rent a vehicle with a driver.*
- *Driver Registration and Employment: There may be a limit on how many drivers can be registered at a time. Once this limit is reached, no new drivers can sign up until there is space available.*

2.4 Assumptions and Dependencies

- *Each store will have a limited number of vehicles, and customers may not always get their preferred vehicle. If unavailable, they can select alternative vehicles or check nearby stores for options.*

- *If the rental includes a driver, the system assumes there is a pool of drivers available to meet customer demand. Shortages in driver availability may cause delays or cancellations.*

- *The system assumes that the platform follows all regulations for renting vehicles and hiring drivers, including licensing, insurance, and safety requirements.*
- *The platform assumes users will access it through modern web browsers on desktops or mobile devices, with reliable internet access.*

- *The platform relies on cloud hosting services for consistent uptime and secure data storage.*

- *The platform relies on customer support tools to assist users with booking, payment, or account management issues.*

- *The system assumes that all registered drivers are highly skilled and capable of ensuring safe and reliable driving.*

3 Specific Requirements

3.1 External Interface Requirements

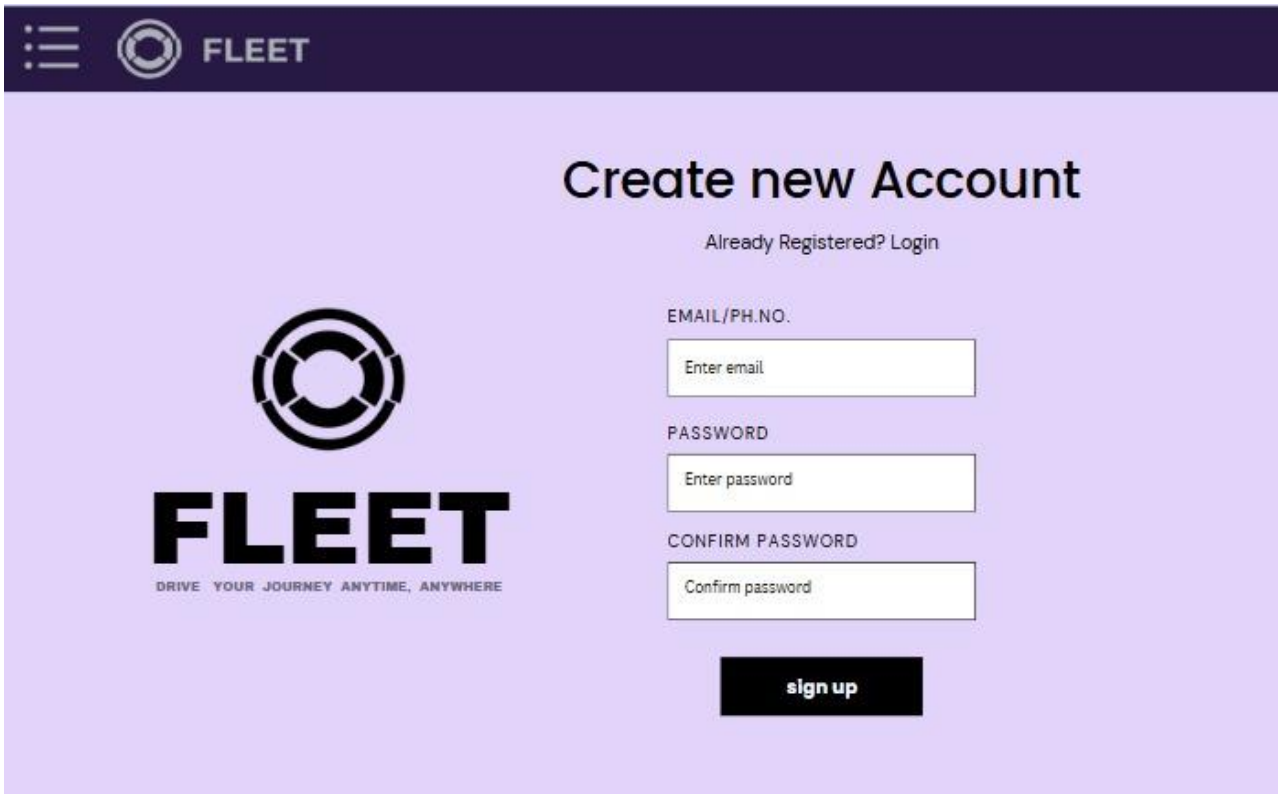
3.1.1 User Interfaces

A user only needs familiarity with browser navigation to understand all the functions provided by our system. The first image presented is a registration page, where new users can create an account by entering their email address or phone number and setting a password. The second image shown is the login page, enabling existing users to sign in to their accounts.

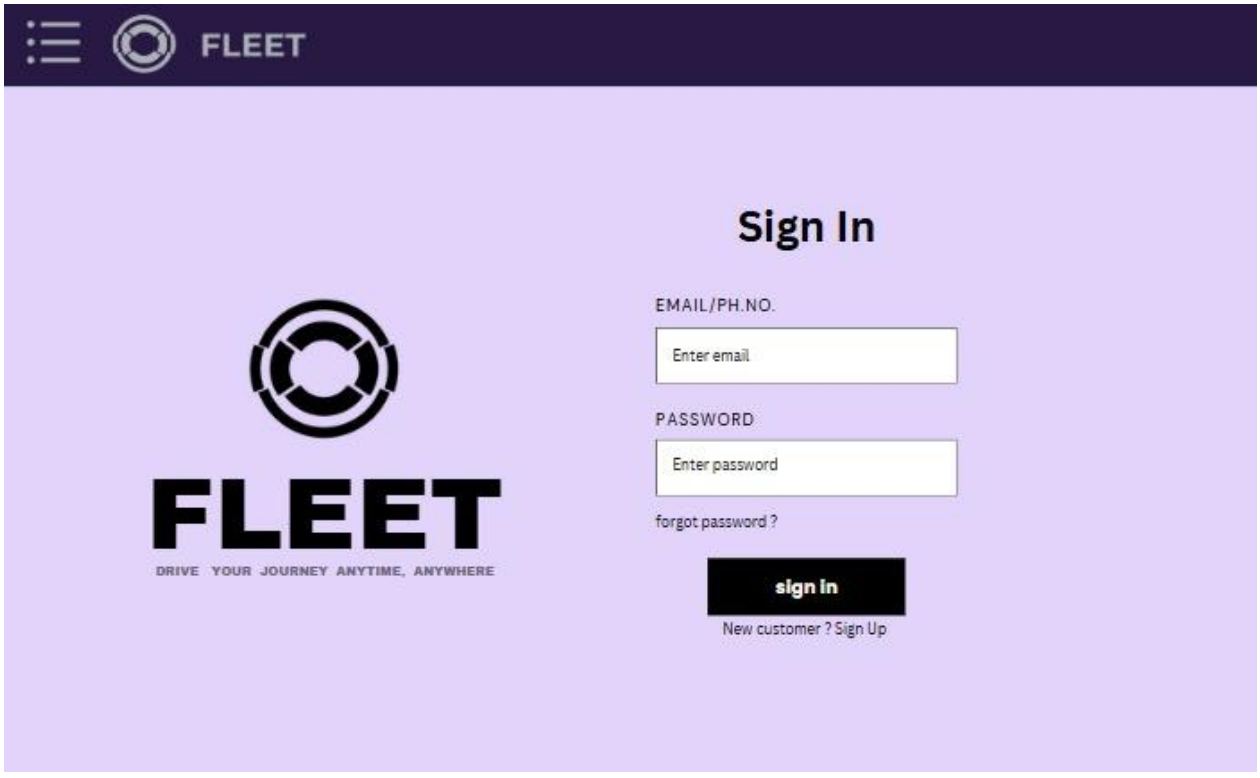
The third image displayed is the home page for users who want to rent a car with a driver. Alternatively, there is an option to rent a car without a driver, which the user can drive themselves. The home page includes a feature to select the pickup location based on the user's location on a map, allowing the driver to arrive at the specified spot. Users can also choose the pickup date, pickup time, return date, and return time to specify the duration for which they wish to rent the car.

After clicking the search button, the user is directed to the fourth page. On this page, they can select the car and driver they prefer at the listed price and proceed to book the car. Additionally, the system provides a tracking option, allowing users to monitor the car's location by clicking a provided link. This ensures transparency and convenience throughout the rental process. Users can also view their past transactions on this page, ensuring they have access to their rental history for reference or future planning.

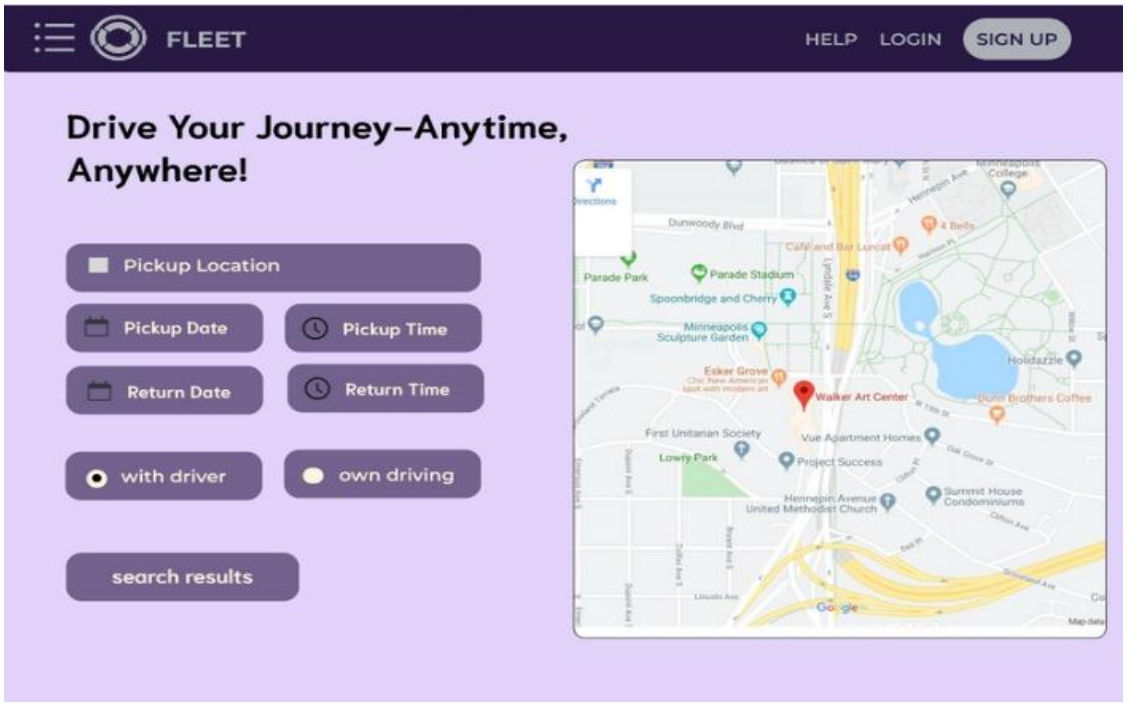
- **Create a new account page:**


FLEET
Create new Account
Already Registered? Login
FLEET
DRIVE YOUR JOURNEY ANYTIME, ANYWHERE
EMAIL/PH.NO.
Enter email
PASSWORD
Enter password
CONFIRM PASSWORD
Confirm password
sign up

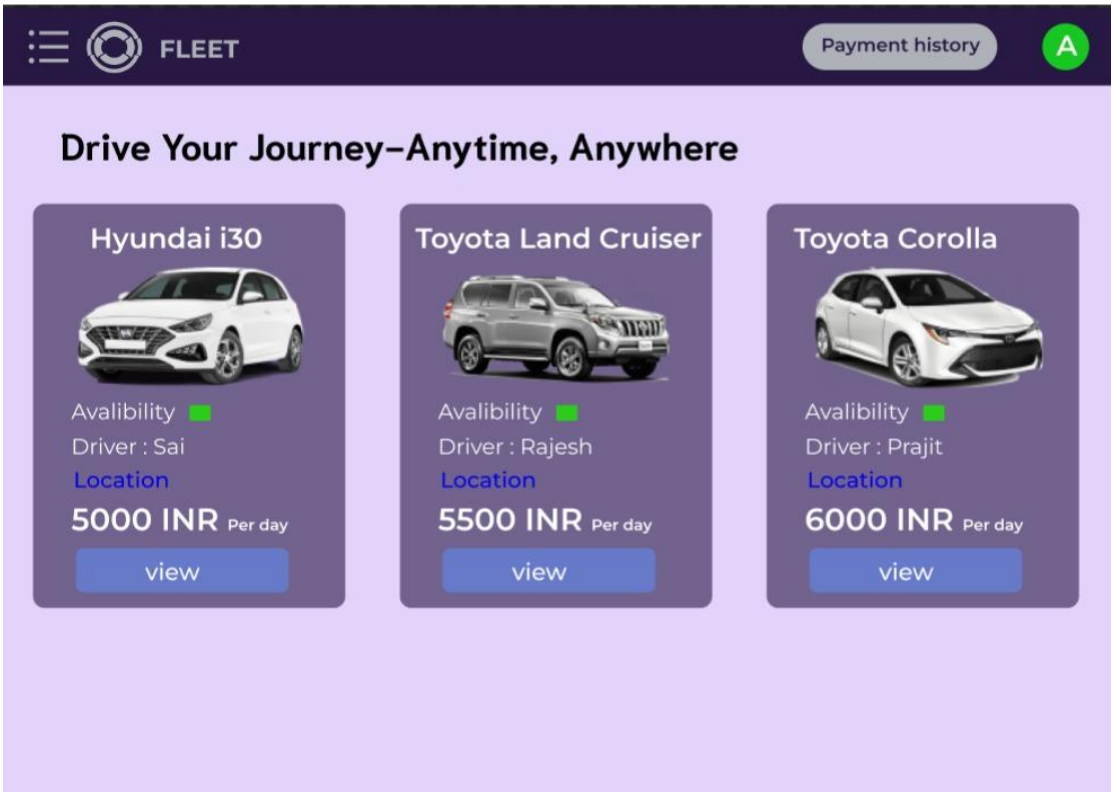
- **Sign In page:**


FLEET
Sign In
FLEET
DRIVE YOUR JOURNEY ANYTIME, ANYWHERE
EMAIL/PH.NO.
Enter email
PASSWORD
Enter password
forgot password ?
sign in
New customer ? Sign Up

- **Home Page:**



- **Page after clicking search results:**



3.1.2 Hardware Interfaces

- **User Devices**: *Devices such as desktops, laptops, tablets, and smartphones*
- **Servers**: *The platform relies on cloud-based servers for data storage, processing*

3.1.3 Software Interfaces

- **Frontend Interfaces:**

The frontend is the user-facing layer of the platform that ensures ease of interaction, responsiveness and user-friendly interface for renters, owners, and drivers. It provides search options, vehicle listings, booking management, feedback systems, and user profiles. The interface also includes dynamic filters for vehicle type, price, and availability, real-time updates on booking status.

- **Backend Interfaces:**

The backend acts as the backbone of the Fleet platform, managing logic, and communication between components. It handles all user requests, processes data, and communicates with the database. The backend host APIs that enable seamless data exchange between the frontend and external software system.

- **Database:**

Stores and manages all data, including user profiles, vehicle listings, booking history, payment records, and feedback. It handles CRUD operations to ensure data is properly added, updated, or retrieved when needed. It also stores encrypted user credentials to ensure data privacy and security.

3.2 Functional Requirements

3.2.1 User Registration and Login

- *The system allows users to register with their email.*
- *The system allows users to change their passwords.*

3.2.2 User-Friendly Interface

- *The system provides a simple and easy-to-use interface that allows users to easily navigate and access the features they need.*
- *The system allows users to search for vehicles using filters such as vehicle type, price, and availability based on their preferred time slot.*

3.2.3 Admin. Login

- *The system enables administrators to access the admin interface using their credentials.*

3.2.4 Vehicle Listing

- *The system shall display detailed descriptions, photos, and availability for all vehicles.*
- *The system shall allow vehicle owners to update vehicle details and availability in real time.*

3.2.5 Owner and Renter Portals

- *The system shall provide vehicle owners with a portal to manage and update their vehicle listings.*
- *The system shall allow renters to view their booking history and manage ongoing rentals.*

3.2.6 Driver Booking

- *The system shall provide renters the option to rent vehicles with or without drivers.*
- *The system shall keep a list of available drivers and help assign them to bookings.*

3.2.7 Secure Booking and Payment

- *The system shall allow users to make real-time bookings for vehicles.*
- *The system shall support multiple secure payment options and provide instant booking confirmations.*

3.2.8 Driver Management

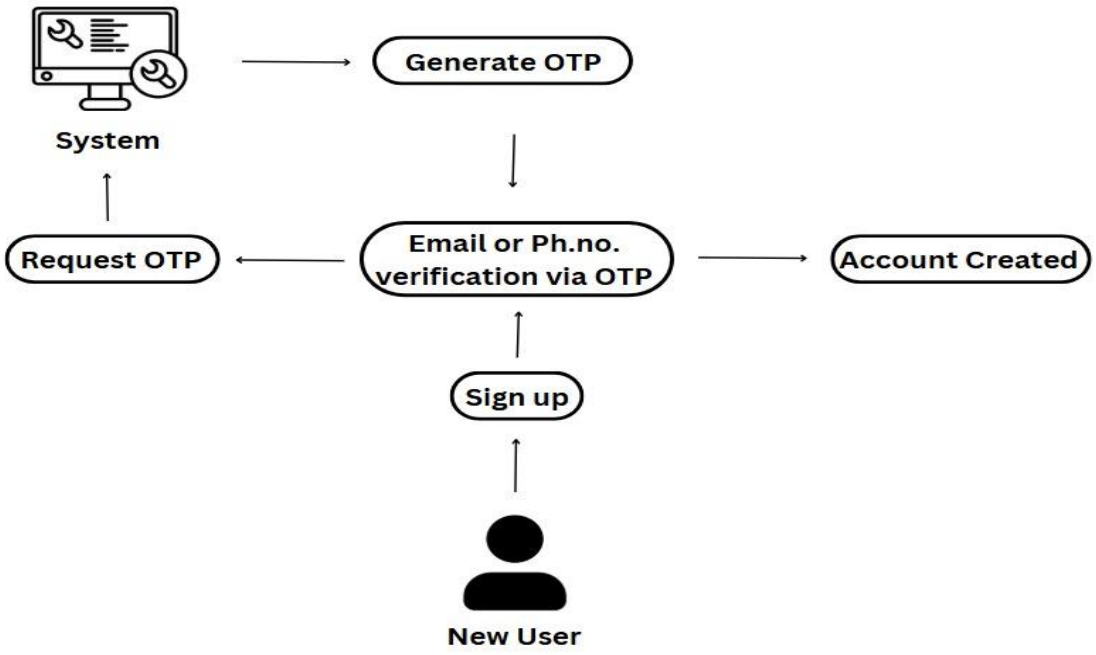
- *The system shall store driver information, including name, contact details, and past rental history.*
- *The system shall allow administrator to assign a driver to a particular vehicle.*

3.2.9 Feedback System

- *The system shall enable users to leave feedback and ratings for drivers and vehicles.*
- *The system shall display feedback and ratings on vehicle and driver profiles for future users.*

3.3 Use Case Model.

3.3.1 Use Case #1 (User Sign Up-UC1)



TO DO*: Account Creation via OTP Verification process*

Author – *Jyothika, Pallavi*

Purpose - *To enable new users to create an account by verifying their identity through an OTP (One-Time Password) sent to their email or phone number. This ensures that only valid users can register.*

Requirements Traceability – *Sign up Interface*

Priority - High - *This use case is critical as it ensures secure onboarding of users and prevents unauthorized access.*

Pre-conditions - *The user must have a valid email address or phone number.*

Post-conditions - *The account is successfully created for the user and will be able to explore the website.*

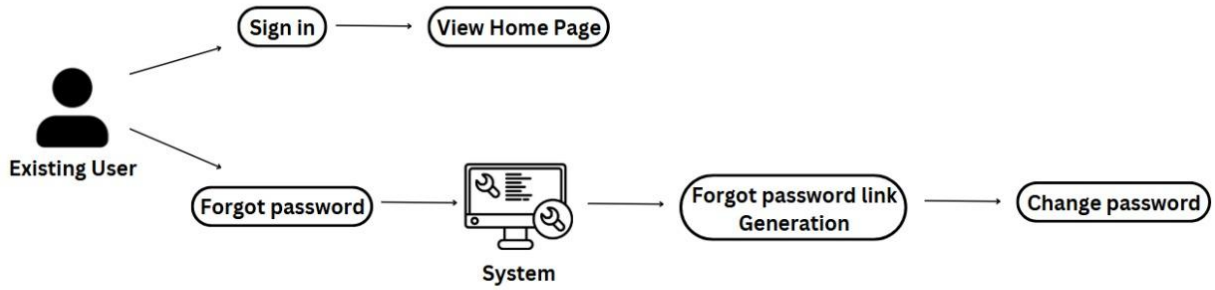
Actors – *New User, System*

Exceptions - *The system allows the user to request a new OTP if they do not receive the initial one.*

The system prompts the user to re-enter the OTP if it is incorrect.

The OTP expires after a set duration (e.g., 5 minutes). The system requires the user to request a new OTP.

3.3.2 Use Case #2(User Sign In-UC2)



TO DO*: Sign In and New Password generation*

Author – *Gayathri, Poorvie*

Purpose - *To allow existing users to access their accounts securely by providing valid login credentials and to provide a fallback option for users to recover access by resetting their passwords in case of forgotten credentials.*

Requirements Traceability – *Sign in Interface, Reset Password Interface.*

Priority - *High - Essential for granting users access to the system and enabling account recovery.*

Pre-conditions - *The user must have an active account.*

Post-conditions - *The user successfully resets their password and can log in with their new credentials.*

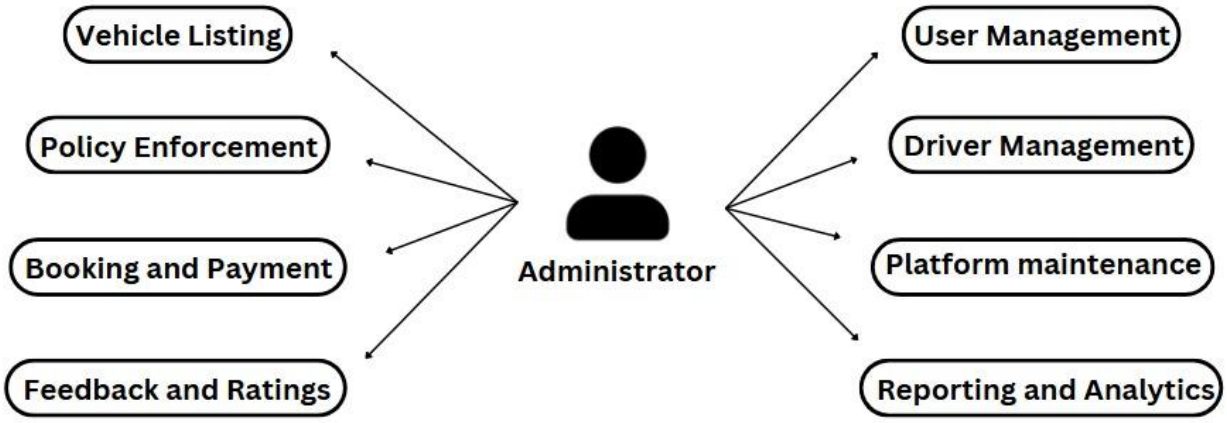
Actors – *Existing User, System*

Exceptions - *The system allows the user to request a new OTP if they do not receive the initial one.*

The system notifies the user if the provided email is not associated with any registered account.

Includes - *UC1*

3.3.3 Use Case #3(Administrator-UC3)



TO DO*: Administration*

Author – *Sruthi, Akil*

Purpose - *To enable administrators to manage and maintain the vehicle inventory, ensuring accurate and updated vehicle availability for customers, driver interactions, user payments and feedbacks.*

Requirements Traceability – *Admin Interface.*

Priority - *High.*

Pre-conditions - *Admins must be authenticated and identified by other admins.*

Post-conditions - *Admin will get the administrative powers.*

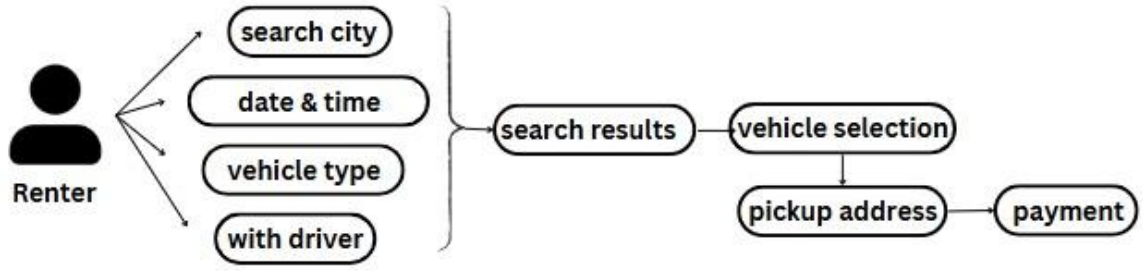
Vehicle listing is added, updated, or removed.

The changes are reflected on the customer interface.

Actors - *Administrator*

Exceptions - *Admins notifies about Invalid vehicle data input, policy enforcements*

3.3.4 Use Case #4(Renter with driver-UC4)



TO DO*: Vehicle selection with driver*

Authors – *Ramith*

Purpose – *To enable user to choose their location, time interval in which they want to rent the vehicle, vehicle type (e.g.: Car, Bike, Van etc.) and filter the options based on their choice. This is the case when a user wants a driver to drive the vehicle.*

Requirements Traceability – *The system shall allow renters to search for vehicles based on their preferences.*

The system shall display real-time vehicle availability along with the driver in the search results.

The system shall enable secure payment processing.

The system shall generate a unique PIN upon successful booking confirmation.

Priority - *High*

Pre-conditions - *The system shall require the renter to create an account.*

Post-conditions - *A booking confirmation is generated, and a unique PIN is shared with the renter. The renter provides this PIN to the driver when they arrive for pickup as confirmation.*

The vehicle along with the driver are marked as "reserved" in the system.

Actors – *User, System*

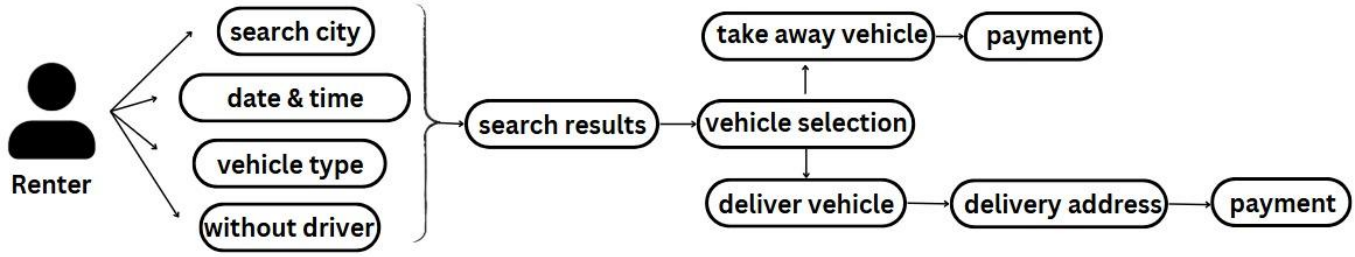
Exceptions - *If no vehicles match the renter's search criteria (e.g., location, time, or vehicle type), the system displays a message:*
"No vehicles available for the selected criteria. Please modify your search."

If the payment fails due to insufficient funds, incorrect payment details, or technical errors, the system notifies the renter:
"Payment failed. Please try again or use a different payment method."

Includes - *UC1, UC2*

Notes/Issues - *Ensure vehicle availability and pricing are updated in real-time. Search for large fleets and databases.*

3.3.5 Use Case #5(Renter without driver-UC5)



TO DO: *Vehicle selection without driver*

Authors – *Shashi Bhidodiya*

Purpose – *To enable user to choose their location, time interval in which they want to rent the vehicle, vehicle type (e.g.: Car, Bike, Van etc.) and filter the options based on their choice. This is the case when a user drives the vehicle by himself.*

Requirements Traceability –*The system shall allow renters to search for vehicles based on their preferences.*

The system shall display real-time vehicle availability in the search results.

The system shall enable secure payment processing.

The system shall enable the user to choose whether they want the vehicle to be delivered or will take away.

Priority - *High*

Pre-conditions - *The system shall require the renter to create an account.*

Post-conditions -*A booking confirmation is generated.*

The vehicle is marked as "reserved" in the system.

Actors – *User, System*

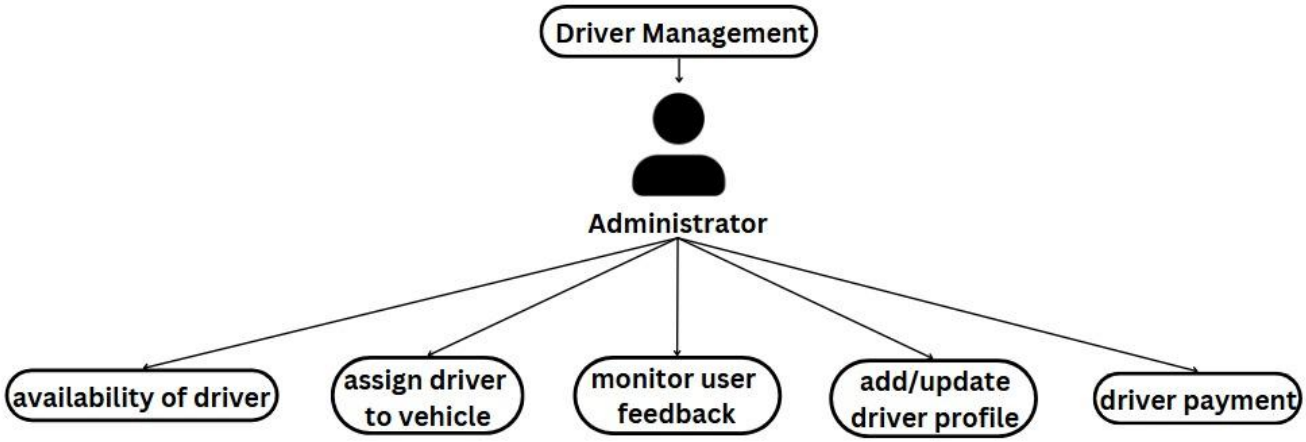
Exceptions - *If no vehicles match the renter's search criteria (e.g., location, time, or vehicle type), the system displays a message:*
"No vehicles available for the selected criteria. Please modify your search."

If the payment fails due to insufficient funds, incorrect payment details, or technical errors, the system notifies the renter:
"Payment failed. Please try again or use a different payment method."

Includes - *UC1, UC2*

Notes/Issues - *Ensure vehicle availability and pricing are updated in real-time. Optimize search for large fleets and databases.*

3.3.6 User Case #6(Administrator-driver management-UC6)



TO DO: *Driver Management by administrator*

Author – *Karthik, Saiprabhav*

Purpose - *To enable the administrator to efficiently manage all aspects of driver operations, including availability, assignment, feedback, profiles, and payment, ensuring smooth operations of the fleet management system.*

Requirements Traceability –*The system shall allow the administrator to assign a driver to a specific vehicle.*

The system shall provide functionality to monitor and respond to user feedback about drivers.

The system shall allow the administrator to add or update driver profiles.

The system shall process driver payments based on completed trips or hours worked.

Priority – *High - Driver management is critical to the functioning of the fleet system as it ensures the availability of drivers, trip completion, and customer satisfaction.*

Pre-conditions - *The administrator must be logged into the system with valid credentials. Driver profiles must exist in the system.*

The fleet database must contain updated vehicle and trip information.

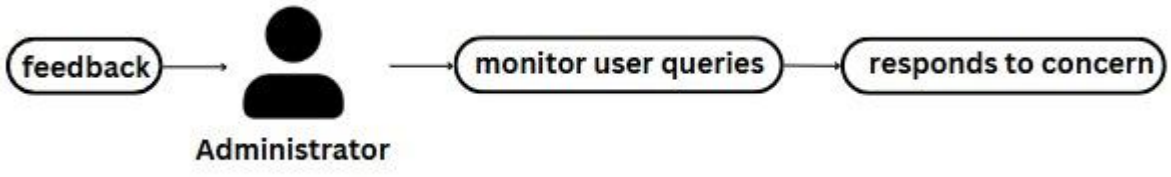
Post-conditions - *Driver availability status is updated after assigning or modifying a driver.*

Payments are successfully processed, and records are maintained.

User feedback is recorded, and appropriate actions (if needed) are taken.

Actors – *Administrator, System, Database*

3.3.7 User Case #7(Feedback-UC7)



TO DO: *Administrator's response to feedback from users*

Author – *Pallavi*

Purpose - *The objective of this use case is to allow the administrator to monitor user feedback and queries, address concerns, and respond to them appropriately. This helps improve user satisfaction and ensures smooth platform operation by resolving issues promptly.*

Requirements Traceability - *The system should allow users to submit feedback and queries.*

The system should provide a dashboard for administrators to monitor user queries.

The administrator should be able to review and respond to user concerns.

The system should store feedback for future analysis and improvements.

Priority - *High*

Pre-conditions - *The administrator must be logged into the system.*

Users must have submitted queries or feedback through the platform.

Post-conditions - *Users receive responses to their concerns or queries.*

Resolved issues contribute to a better user experience.

Actors – *Administrator, User*

Exceptions - *No Feedback or Queries Available – If there are no user queries, the administrator has no actions to take.*

Delayed Responses – If the administrator does not respond on time, users may experience dissatisfaction.

Unclear or Incomplete Queries – If users submit vague or incomplete concerns, the administrator may be unable to provide an appropriate response.

System Failure – If the system crashes or fails to save user queries, important concerns may go unresolved.

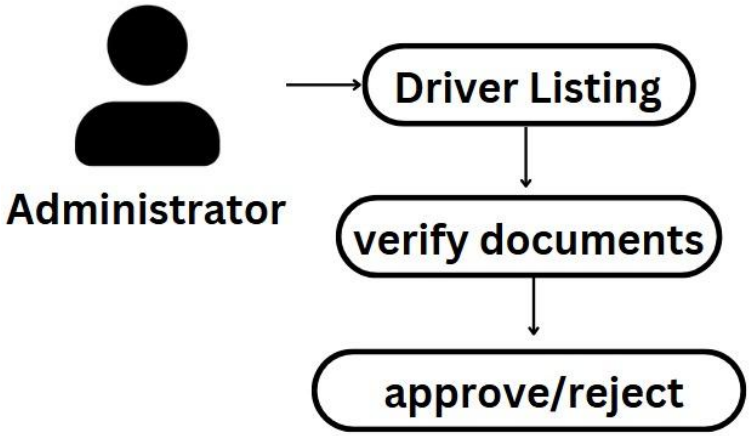
Includes- *UC3*

Notes/Issues - *Define the expected response time for administrators to address user concerns.*

Decide whether responses should be automated for common queries or manually handled by administrators.

Implement a priority system for handling urgent queries first.

3.3.8 User Case #8(Administrator – Driver listing-UC8)



TO DO*: Registering a driver*

Author- *Jyothika*

Purpose - *The objective of this use case is to allow the administrator to list drivers, verify their documents, and approve or reject them for service.*

This ensures that only verified drivers are available for customers.

Requirements Traceability - *The system should allow administrators to list new drivers.*

The system should allow administrators to approve or reject a driver's registration.

Priority - *High*

Pre-conditions - *The administrator must be logged into the system.*

Post-conditions - *If approved, the driver is added to the active driver list and available for vehicle rentals.*

Actors - *Administrator*

Exceptions - *If a driver's license or other mandatory documents have expired, the administrator will not approve the driver.*

If the system fails to add the driver after the verification process due to a technical error, the process may need to be repeated.

Includes- *UC- 3*

Notes/Issues - *Need to define the specific documents required for verification.*

4. Other Non-functional Requirements

4.1 Performance Requirements

- *The platform should handle up to 10,000 simultaneous users without performance degradation.*
- *Response time for critical operations (e.g., booking a vehicle) should not exceed 3 seconds under normal load conditions.*
- *The system must process and confirm bookings within 5 seconds of initiation.*
- *The system should handle at least 500 concurrent booking requests without impacting performance.*
- *Queries for vehicle listings, user accounts, or feedback must execute within 500 milliseconds under normal load conditions.*

4.2 Safety and Security Requirements

- *The platform should allow authentication for the user accounts.*
- *Password will be saved and encrypted in the database to ensure user's privacy.*
- *The email and phone number provided by the user should be verified.*
- *The platform must ensure secure processing of payments.*

4.3 Software Quality Attributes

4.3.1 Usability:

The interface should be intuitive, with user-friendly navigation and consistent design across all pages.

4.3.2 Maintainability:

After the system is completed, project members can implement necessary changes or add new features at any time as needed.
The software code should be thoroughly commented to ensure that future updates or modifications can be made easily and efficiently.

4.3.3 Portability:

Since the platform is a web application, the user should be able to operate it seamlessly on any device with browser and internet support.

4.3.4 Availability:

The software should available all time and ready to perform its tasks whenever the user needs it.

The software will be deployed on the device provided by the Computer Center. As the Computer Center operates 24/7, we expect our software to ensure uninterrupted availability under normal circumstances.

4.3.5 Reliability:

If the main server hosting the platform crashes, the failover system should transfer traffic to a backup server seamlessly, ensuring users can continue to access the platform without interruption.

4.3.6 Interoperability:

The platform should be able to integrate seamlessly with third-party systems, such as payment gateways, email services, SMS providers for OTP verification, and mapping services for location-based features, ensuring smooth interaction and data exchange between different systems.

4.3.7 Robustness:

The software must handle unexpected situations gracefully, such as invalid user inputs, network interruptions, or system errors, by implementing error handling, fallback mechanisms, and clear notifications to ensure a seamless user experience even during failures.

4.3.8 Reusability:

The key components of the software should be designed as modular units with proper commenting so that they can be extended for additional features and other similar projects in the future.

5 Other Requirements

- **Administrator Account:**

The administrator must own a renting organization and provide the necessary documentation to qualify for an account on the platform. The platform owner is responsible for creating the administrator's account and providing them with their login credentials.

- **Data Privacy Compliance*:***

The system must comply with data protection laws such as GDPR, CCPA, or other applicable regional regulations to ensure the protection of users' personal and financial data.

- **Terms and Conditions Agreement:**

Users must agree to the platform's terms and conditions, which outline their responsibilities, acceptable usage, and liability before accessing the services.

- **Licensing and Permits:**

All vehicle owners using the platform must upload proof of valid licensing, permits, and vehicle registration as required by local laws.

- **Insurance Requirements:**

Vehicle owners must provide valid insurance documentation for each listed vehicle. The platform should also provide guidance on insurance coverage for renters and drivers.

- **Driver Eligibility Verification:**

The platform must verify driver eligibility, including checking valid driver's licenses and conducting background checks, as per applicable laws.

- **Age Restrictions:**

Users must meet the minimum legal age to rent or drive vehicles as per jurisdiction-specific regulations.

Appendix A – Data Dictionary

Name	*String*	*50 characters max*	*Name of the user.*	*Alphabetic (e.g., John Doe)*
Email	*String*	*Valid email format*	*Email address of the user.*	*E.g., example@example.com*
Phone Number	*String*	*10-15 digits*	*User's contact number.*	*Numeric (e.g., +123456789)*
Password	*String*	*Encrypted*	*Encrypted password for user authentication.*	*Encrypted values (e.g., hashed password)*
Vehicle Type	*String*	*Variable*	*Type of vehicle available for rent (e.g., car, bike, van, truck).*	*Predefined values (e.g., Car, Bike, Truck)*
Vehicle Details	*String*	*Variable*	*Detailed description of the vehicle, including model, make, and features.*	*Text (e.g., Honda Civic, 2020, GPS)*
Availability	*Boolean/Date Range*	*Variable*	*Indicates whether the vehicle is available for rent or specifies available date ranges.*	*True/False or specific date ranges*
Price	*Float*	*Currency format*	*Rental price of the vehicle per unit time (e.g., per hour, per day).*	*E.g., 50.00 (USD/day)*
Driver Required	*Boolean*	*1-bit*	*Specifies whether a driver is required by the renter.*	*True (with driver), False (without driver)*
Booking ID	*String/Integer*	*Variable*	*Unique identifier for each booking transaction.*	*Alphanumeric (e.g., B1001)*
Booking Status	*String*	*Variable*	*Status of the booking (e.g., pending, confirmed, completed, canceled).*	*Predefined values (Pending, Completed, etc.)*
Payment ID	*String/Integer*	*Variable*	*Unique identifier for each payment transaction.*	*Alphanumeric (e.g., P12345)*
Payment Status	*String*	*Variable*	*Status of the payment (e.g., successful, failed, pending).*	*Predefined values (Successful, Failed, etc.)*
Rating	*Float*	*0.0 - 5.0*	*Feedback rating for a vehicle or driver, given*	*Range: 0.0 to 5.0*

			by the user.	
Feedback	*String*	*500 characters max*	*User-provided feedback about the rental experience, vehicle, or driver.*	*Text (e.g., Great experience!)*
Driver ID	*String/Integer*	*Variable*	*Unique identifier for the driver, if applicable.*	*Alphanumeric (e.g., D1001)*
Driver Details	*String*	*Variable*	*Information about the driver, including name, license number, and contact information.*	*Text (e.g., John Smith, License XYZ123)*
Location	*String*	*Variable*	*Pickup or drop-off location for the vehicle rental.*	*Text/Coordinates (e.g., City Center, 28.6, 77.2)*

Appendix B - Group Log

MEET DATE	TOPIC DISCUSSED	DURATI0N
08/01/2025	We discussed the potential topics for the project and the things we need to learn in the process.	60 min
10/01/2025	We finalized our project and discussed additional features to add.	90 min
10/01/2025	We had a meeting with the professor to clarify some doubts.	45 min
12/01/2025	We decided on the title of the project and finalized our rough UI and features to add.	90 min
19/01/2025	We divided the team for different parts of the SRS and held several meetings within the team.	90 min
22/01/2025	It was our first meeting with our TA, so we asked some doubts about the SRS, and he gave us valuable tips.	30 min
23/01/2025	We combined work of different teams and finalized the SRS document	120 min
24/01/2025	Submission of the SRS document	