

Darshan University

A Project Report on

**Taxi management System**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – V

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name:Jotaniya Meet | Enrollment No.:23010101116 |
| Academic Year  (2023-2024) | |
| Internal Guide Mr. Rajkumar B Gondaliya Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2101CS405)** for entitled **Taxi management System** submitted in partial fulfilment for the Semester-4 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of Prof. Rajkumar B Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Jotaniya Meet

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **Taxi management System** has been satisfactorily prepared by Jotaniya Meet(23010101116) under my guidance in the fulfillment of the course **Software Engineering (2101CS405)** work during the academic year 2023-2024.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. Rajkumar B Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. Rajkumar B Gondaliya and all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Jotaniya Meet**

**ABSTRACT**

The **Taxi Management System** (TMS) is designed to simplify and improve taxi services using a computerized system. This system makes it easy for passengers to book rides, view ride details, and track their taxi in real-time. Passengers can also manage their accounts and see their ride history.

For drivers, TMS provides tools to manage their work, accept or reject ride requests, and see their earnings. The system helps dispatch taxis efficiently, reducing waiting time and finding the best routes.

Administrators have a special panel to manage everything, including user accounts and system performance. They can also generate reports on ride details, driver performance, and earnings.

The main goal of TMS is to make the process of booking and managing taxi rides easier for everyone, reducing the need for manual work and improving overall efficiency.

**Table of Contents**

[List of Figures I](#_Toc185872477)

[List of Tables II](#_Toc185872478)

[1 Introduction 1](#_Toc185872479)

[1.1 Product perspective 1](#_Toc185872480)

[1.2 Product features 1](#_Toc185872481)

[1.2.1 There are four different users who will be using this product: 1](#_Toc185872482)

[1.2.2 The features that are required for the Admin are: 1](#_Toc185872483)

[1.2.3 The features that are required for the Manager are: 1](#_Toc185872484)

[1.2.4 The features that are required for the Driver are: 1](#_Toc185872485)

[1.2.5 The features that are required for the Customer are: 2](#_Toc185872486)

[1.3 Functional Requirement 2](#_Toc185872487)

[1.3.1 **Admin** 2](#_Toc185872488)

[**1.3.2** **Manager** 2](#_Toc185872489)

[1.4 Non-Functional Requirement 3](#_Toc185872490)

[1.4.1 **Usability**: 3](#_Toc185872491)

[1.4.2 **Accuracy**: 3](#_Toc185872492)

[1.4.3 **Availability**: 3](#_Toc185872493)

[1.4.4 **Maintainability**: 3](#_Toc185872494)

[1.4.5 **Response Time**: 3](#_Toc185872495)

[The system should respond to user queries and actions (e.g., booking a taxi, viewing available taxis) within 2 seconds under normal load conditions. 3](#_Toc185872496)

[1.4.6 **Data Protection**: 3](#_Toc185872497)

[All user data, including payment information and personal details, must be encrypted in transit (using TLS) and at rest (using AES-256 encryption). 3](#_Toc185872498)

[2 Design and Implementation Constraints 4](#_Toc185872499)

[2.1 Use case diagram 4](#_Toc185872500)

[2.2 Activity diagram and Swimlane diagram 6](#_Toc185872501)

[2.3 Sequence diagram 10](#_Toc185872502)

[2.4 State diagram 12](#_Toc185872503)

# List of Figures

[Figure 2.1‑1 Use case diagram for Taxi management System](#_Toc146706848) 4

[Figure 2.1‑2 Use case diagram for Taxi management System](#_Toc146706848) 5

[Figure 2.2‑1 Activity diagram for taxi booking](#_Toc146706849) 6

[Figure 2.2‑2 Activity diagram for profile edit](#_Toc146706849) 7

[Figure 2.2‑2 Swimlane diagram for taxi booking](#_Toc146706850) 8

[Figure 2.2‑2 Swimlane diagram for profile edit](#_Toc146706850) 9

[Figure 2.3‑1 Sequence diagram for taxi bookin](#_Toc146706851) 10

[Figure 2.3‑1 Sequence diagram for profile edit](#_Toc146706851) 11

[Figure 2.4‑1 State diagram of Ride](#_Toc146706852) 12

[Figure 2.4‑2 State diagram for Driver 1](#_Toc146706853)2

# Introduction

## Product perspective

This project updates traditional taxi services into an internet-based system, making it easier for both drivers and passengers to use. It allows passengers to book taxis, track their rides, and manage their accounts online. For drivers, it provides tools to manage their schedules, accept ride requests, and track their earnings.

## Product features

### There are four different users who will be using this product:

* **Admin**
* **Manager**
* **Driver**
* **Customer**

### The features that are required for the **Admin** are:

* **User Management**: Add, edit, and delete users; assign roles (admin, manager, driver, customer).
* **System Configuration**: Set up system preferences including pricing, service areas, and vehicle options.
* **Financial Oversight**: Process payments, handle refunds, and monitor financial transactions.
* **Monitoring and Reporting**: Generate reports, track vehicle status, and oversee the system dashboard.
* **Complaint Resolution**: Address and resolve customer complaints and issues.
* **Promotions**: Create and manage promotional offers and discounts.
* **Compliance**: Ensure adherence to local regulations and data protection rules.
* **Data Management**: Perform system backups and restore the system when needed

### The features that are required for the **Manager** are:

* **Driver Scheduling**: Assign shifts and plan work schedules for drivers.
* **Trip Monitoring**: Track the status of ongoing trips and assist with issues.
* **Routing**: Provide drivers with optimized routes considering traffic conditions.
* **Performance Tracking:** Monitor and evaluate driver performance based on trip completion and ratings.
* **Customer Assistance**: Help customers with issues during their trips.
* **Vehicle Dispatch**: Assign vehicles to drivers as per requirements.
* **Revenue Analysis**: Review revenue data and assess profitability.
* **Feedback Review:** Look at customer feedback for service quality improvements.

### The features that are required for the **Driver** are:

* **Trip Management**: Accept or decline ride requests and navigate using maps.
* **Status Updates:** Change availability status (available, on trip, offline).
* **Payment Collection:** Collect payments through various methods from customers.
* **Customer Communication**: Contact customers for trip details and updates.
* **Vehicle Inspection**: Check the vehicle's condition before starting trips.
* **Emergency Handling**: Access emergency contacts and instructions.
* **Performance Tracking:** View personal performance metrics and customer ratings.

### The features that are required for the Customer are:

* **Ride Booking**: Request and book taxis via app, website, or phone.
* **Tracking**: Track the taxi's location and estimated arrival time.
* **Fare and Payment**: Get fare estimates and pay securely using various methods.
* **Profile Management**: Update personal details and preferences in their profile.
* **Trip History**: View details of past trips and payments.
* **Feedback and Ratings**: Rate drivers and provide feedback on trips.
* **Discounts and Offers:** Apply promo codes and discounts for rides.

## Functional Requirement

### **Admin**

* **Add Users**: Add new users like drivers, managers, and customers.
* **Edit Users**: Edit details of existing users.
* **Delete Users**: Remove users from the system.
* **Set Roles**: Assign roles (admin, manager, driver, customer) to users.
* **View Dashboard**: Access a main dashboard with key metrics and system status.
* **Configure Settings**: Set up system preferences like pricing, service areas, and vehicle options.
* **Generate Reports:** Create reports on bookings, revenue, and user activity.
* **Track Vehicles**: Monitor vehicle status, including availability and condition.
* **Handle Payments:** Oversee financial transactions, including processing payments and refunds.
* **Respond to Complaints**: Address and resolve customer complaints and issues.
* **Create Promotions**: Develop and activate promotional offers and discounts.
* **Ensure Compliance**: Verify adherence to local regulations and data protection rules.
* **Backup Data**: Perform regular system backups to prevent data loss.
* **Restore System**: Use backups to restore system functionality if needed.

### **Manager**

* **Assign Shifts**: Schedule drivers for specific shifts.
* **Monitor Trips**: Keep an eye on ongoing trips and their statuses.
* **Provide Routes**: Give drivers optimized routes considering traffic conditions.
* **Track Performance**: Check driver performance, like how well they complete trips and their ratings.
* **Help Customers**: Assist customers with issues during their trips.
* **Dispatch Vehicles**: Assign vehicles to drivers based on needs.
* **Plan Schedules**: Create and adjust driver work schedules.
* **View Feedback**: Look at customer feedback and ratings for quality improvements.
* **Resolve Issues**: Address complaints from customers regarding service or billing.
* **Check Revenues:** Review revenue data and profitability for different areas.

1.3.3 **Driver**

* **Accept Trips**: Accept or decline ride requests.
* **Use Navigation**: Access maps and directions for efficient travel.
* **Update Status**: Change status to show availability (available, on trip, offline).
* **Collect Payments**: Take payments from customers through various methods.
* **Communicate with Customers**: Contact customers for trip details and updates.
* **Inspect Vehicle**: Check vehicle before starting trips for safety.
* **Set Availability**: Update availability for taking rides.
* **Handle Emergencies:** Access emergency contacts and instructions.
* **View Feedback**: See customer ratings and comments.
* **Track Own Performance**: View personal performance data like trip completion and ratings.

1.3.4 **Customer**

* **Request Rides:** Book a taxi using the app, website, or phone.
* **Track Taxi:** See the taxi’s location and estimated arrival time.
* **Get Fare Estimate**: See an estimated fare before booking.
* **Pay for Rides:** Pay securely using various methods like cards or mobile wallets.
* **Cancel Rides**: Cancel a booking within a given time frame without charges.
* **Rate Drivers**: Provide ratings and feedback on drivers and trips.
* **Edit Profile**: Update personal details and preferences in their profile.
* **Receive Notifications:** Get alerts for booking confirmation, taxi arrival, and trip updates.
* **Use Accessibility Features:** Access options for special needs like wheelchair access.
* **View Trip History**: Check details of past trips and payments.
* **Apply Discounts:** Use promo codes and discounts when booking rides.
* **Schedule Rides:** Schedule rides for future dates and times.
* **Contact Driver:** Communicate with the driver directly through the app.
* **Choose Vehicle Type**: Select preferred vehicle type if available.
* **Save Locations:** Save frequently used locations like home or work.
* **View Driver Details:** See driver’s name, photo, and vehicle information before the ride.

### 

## Non-Functional Requirement

### **Usability**:

* The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

### **Accuracy**:

* The data stored about the books and the fines calculated should be correct, consistent, and reliable.

### **Availability**:

* The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### **Maintainability**:

* The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

### **Response Time**:

### The system should respond to user queries and actions (e.g., booking a taxi, viewing available taxis) within 2 seconds under normal load conditions.

### **Data Protection**:

### All user data, including payment information and personal details, must be encrypted in transit (using TLS) and at rest (using AES-256 encryption).

# Design and Implementation Constraints

## Use case diagram

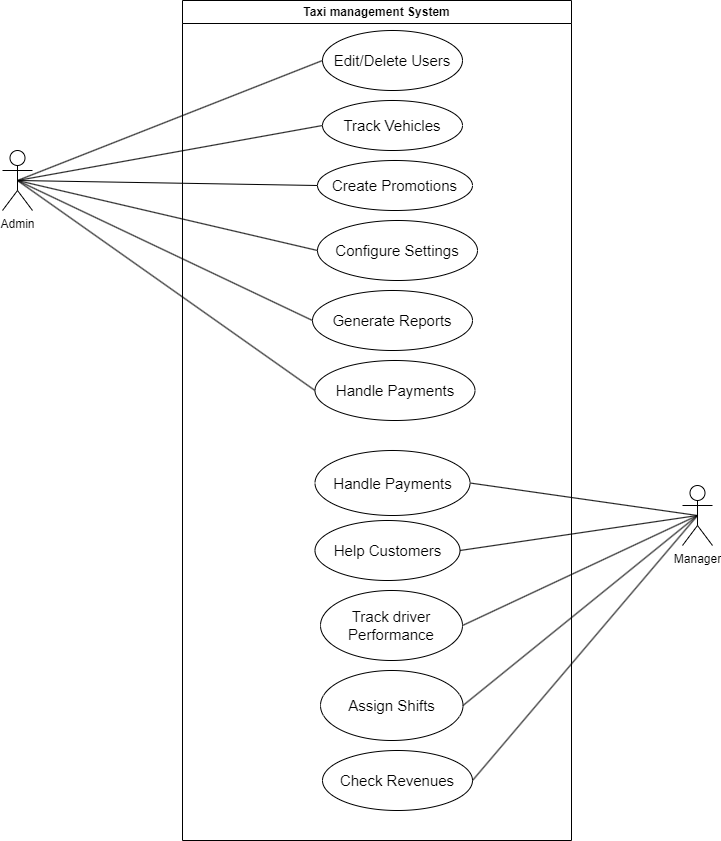


Figure ‑ Use case diagram for Taxi management System

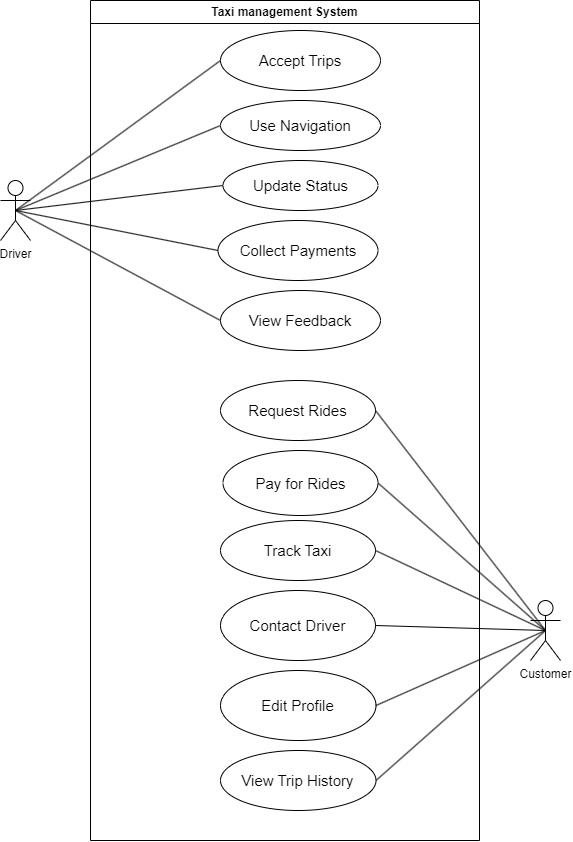


Figure ‑ Use case diagram for Taxi management System

## Activity diagram and Swimlane diagram

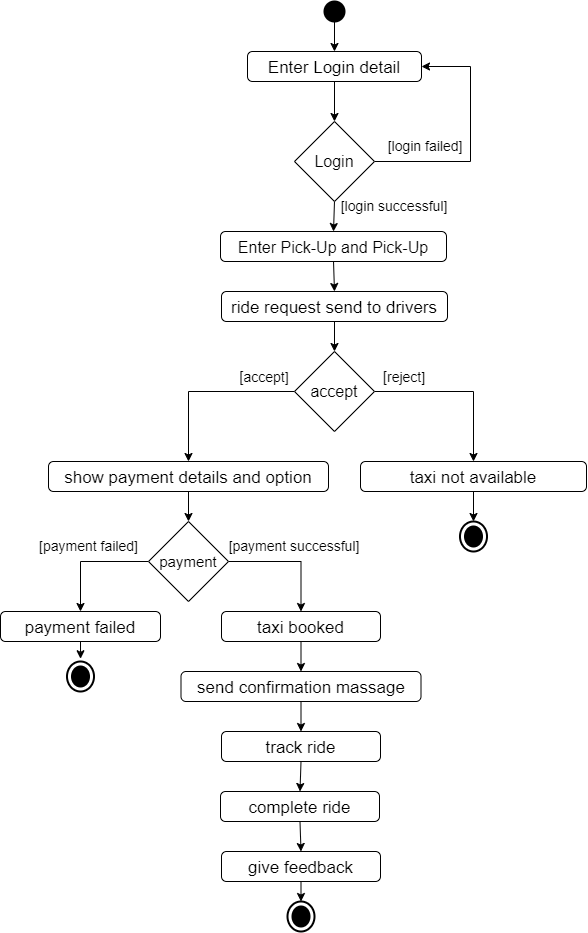


Figure ‑ Activity diagram for taxi booking

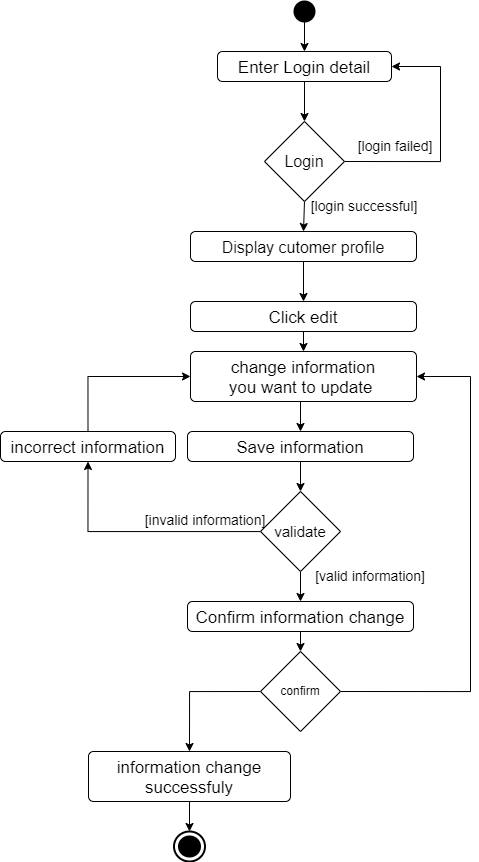


Figure ‑ Activity diagram for profile edit

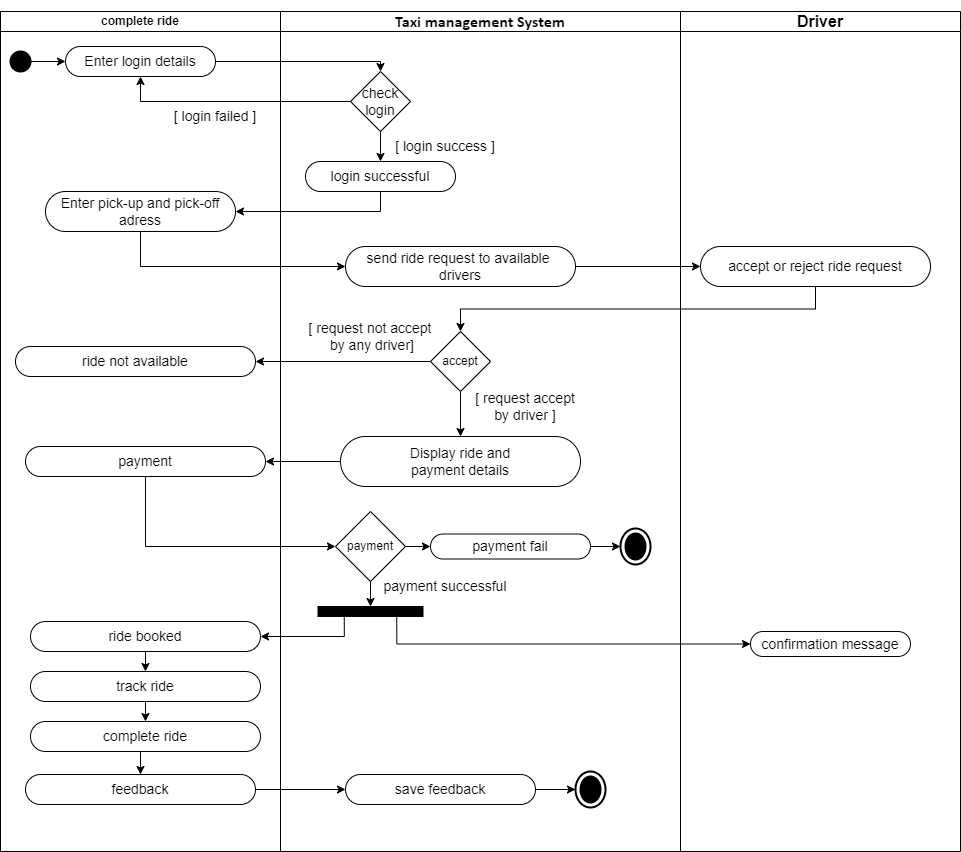


Figure ‑3 Swimlane diagram for taxi booking

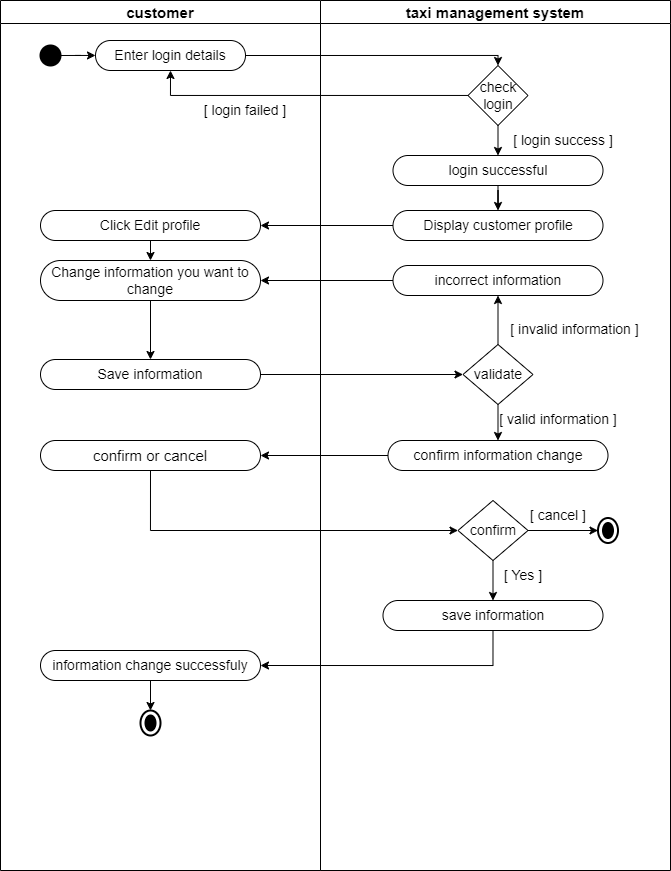


Figure ‑4 Swimlane diagram for profile edit

## Sequence diagram

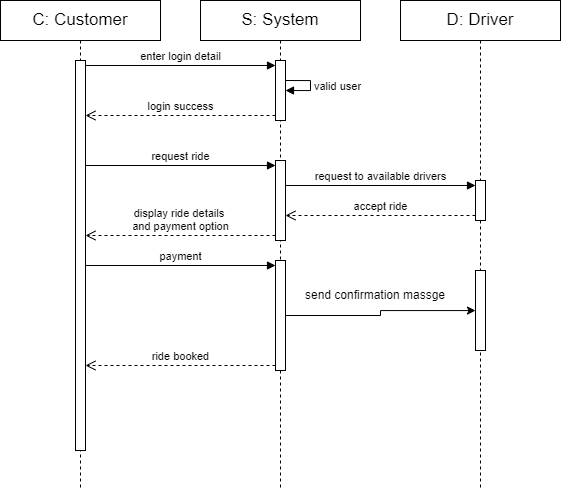
Untitled Diagram.drawio (6)Untitled Diagram.drawio (6)Untitled Diagram.drawio (6)

Figure ‑ Sequence diagram for taxi booking

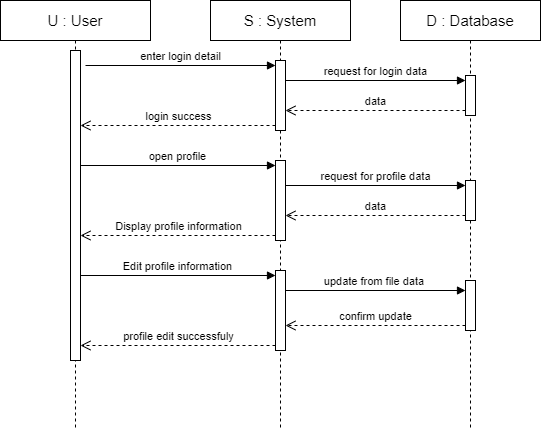
Untitled Diagram.drawio (6)Untitled Diagram.drawio (6)Untitled Diagram.drawio (6)

Figure ‑2 Sequence diagram for profile edit

## State diagram

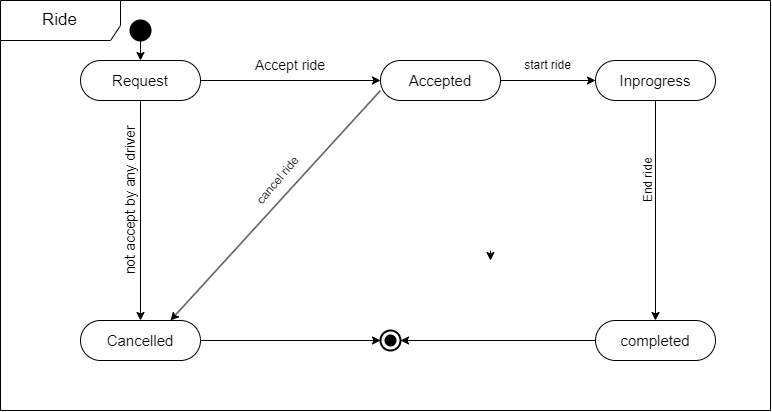


Figure ‑ State diagram for ride

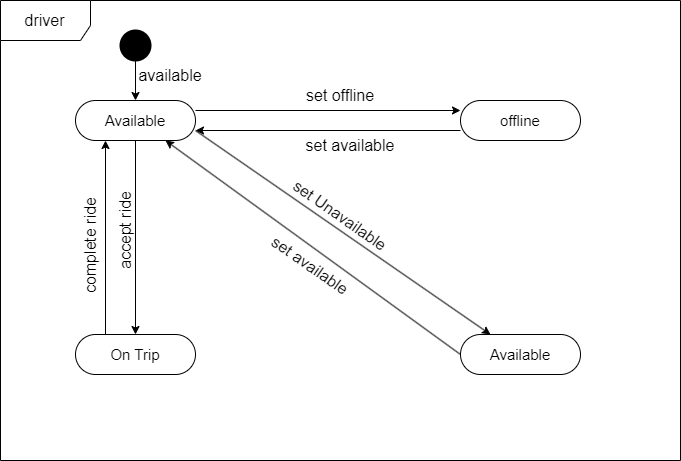


Figure ‑2 State diagram for driver

**2.5 Class Diagram**

## Data Flow Diagram

### DFD Level-0

### DFD Level-1

# External Interface Requirement (Screens)

## Screen-1: Restore System

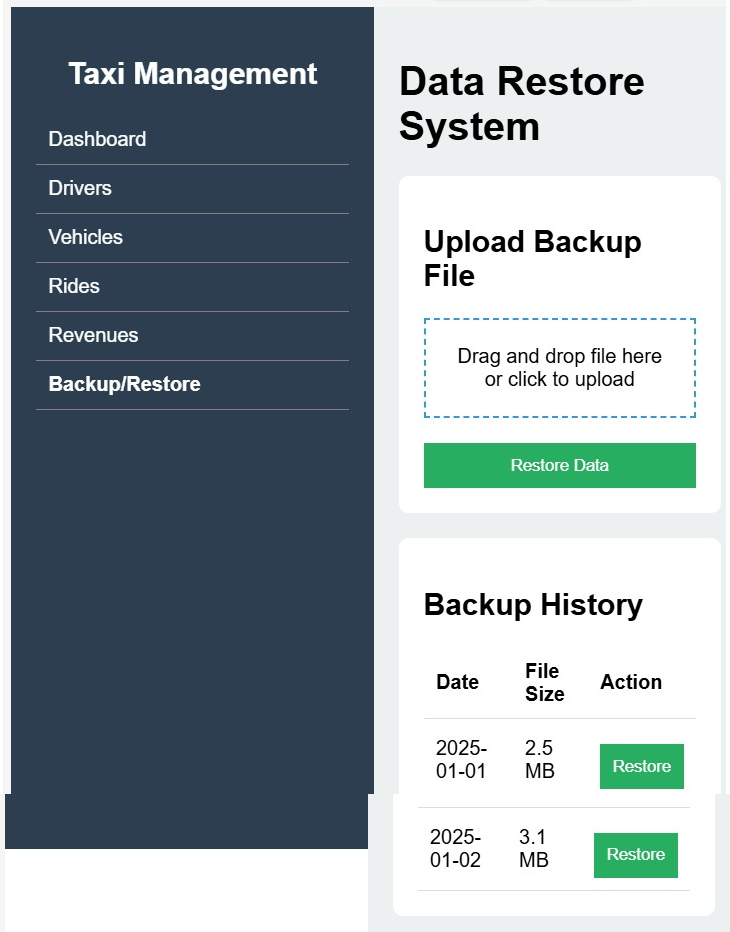


Figure ‑ Screen-1: Restore System

**Purpose:** This screen is part of a Taxi Management System that allows users to back up and restore data. It provides options to upload a backup file for data restoration and view previous backups with the ability to restore them as needed.

Table ‑ Screen element of Restore System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Dashboard Navigation | |  | | --- | |  |  |  | | --- | | Link | | M | 1 | Sidebar menu for navigating to different sections like Dashboard, Drivers, Vehicles, Rides, Revenues, and Backup/Restore. |
| 2 | Backup/Restore Menu | Link | M | 1 | A menu option to access the Data Restore System screen. |
| 3 | Upload Backup File | File Upload | M | 1 | Drag-and-drop or click-to-upload area for selecting a backup file |
| 4 | Restore Data Button | Button | M | 1 | Triggers the restoration process using the uploaded backup file. |
| 5 | Backup History Table | Table | M | 1 | Displays a list of previous backup files with date, file size, and restore action. |
| 6 | Backup Date | Text | M | 1 | Column in the Backup History table showing the backup date. |
| 7 | File Size | Text | M | 1 | Column in the Backup History table displaying the size of each backup file. |
| 8 | Restore Button | Button | M | N | Allows users to restore a specific backup file from the history list. |

## Screen-2: Check Revenues

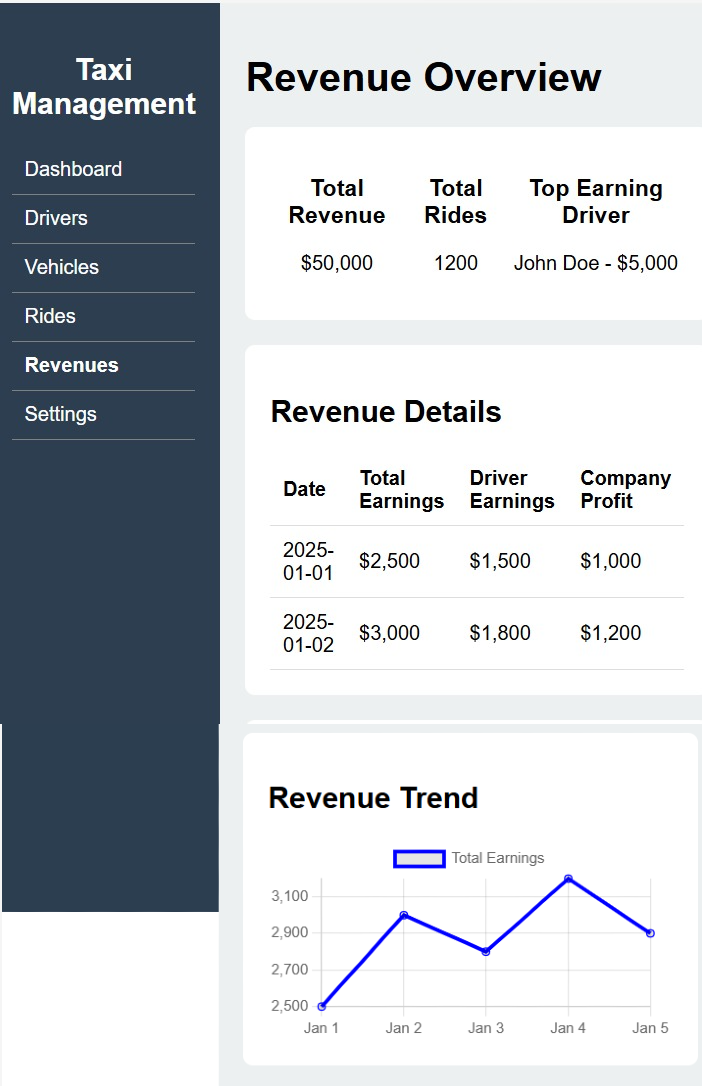


Figure ‑ Screen-2: Check Revenues

**Purpose:** This screen provides an overview of **revenue analytics** for a **Taxi Management System**. It displays key financial metrics, including total revenue, total rides, and the top-earning driver. Additionally, it includes a **detailed revenue breakdown** by date and a **trend chart** to visualize earnings over time.

Table ‑ Screen element of Check Revenues

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | **Dashboard Navigation** | Link | M | 1 | Sidebar menu for navigating to different sections like Dashboard, Drivers, Vehicles, Rides, Revenues, and Settings. |
| 2 | **Revenues Menu** | Link | M | 1 | A menu option to access the Revenue Overview screen. |
| 3 | **Total Revenue** | Text | M | 1 | Displays the total revenue earned by the taxi service. |
| 4 | Total Rides | Text | M | 1 | Displays the total number of rides completed. |
| 5 | Top Earning Driver | Text | M | 1 | Displays the name and earnings of the highest-earning driver. |
| 6 | Revenue Details Table | Table | M | 1 | Shows a breakdown of total earnings, driver earnings, and company profit by date. |
| 7 | Date Column | Text | M | 1 | Displays the date for each revenue entry. |
| 8 | Total Earnings Column | Text | M | 1 | Displays the total earnings for the given date. |
| 9 | Driver Earnings Column | Text | M | 1 | Displays the earnings allocated to drivers for the given date. |
| 10 | Company Profit Column | Text | M | 1 | Displays the company's profit for the given date. |
| 11 | Revenue Trend Chart | Chart | M | 1 | Graphical representation of total earnings over a period of time. |

## Screen-3: Track Own Performance(For Driver)

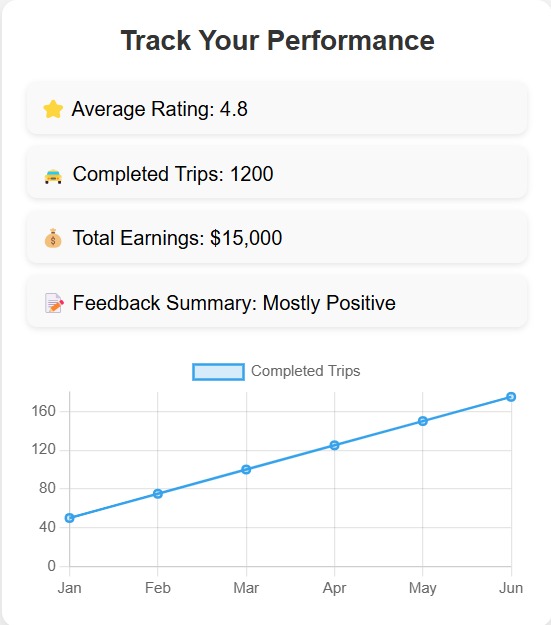


Figure 3.3‑ Screen-3: Track Own Performance

**Purpose:** This screen provides a **performance tracking dashboard** for taxi drivers, displaying key metrics such as **average rating, total completed trips, total earnings, and feedback summary**. It also includes a **trend chart** visualizing the number of completed trips over time to help drivers monitor their progress.

Table ‑ Screen element of Track Own Performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Average Rating | Text | M | 1 | Displays the driver's overall rating out of 5. |
| 2 | Completed Trips | Text | M | 1 | Shows the total number of trips completed by the driver. |
| 3 | Total Earnings | Text | M | 1 | Displays the total earnings made by the driver. |
| 4 | Feedback Summary | Text | M | 1 | Provides an overall summary of customer feedback. |
| 5 | Completed Trips Trend Chart | Chart | M | 1 | Graph showing the trend of completed trips over months. |

## Screen-4: View Driver Details

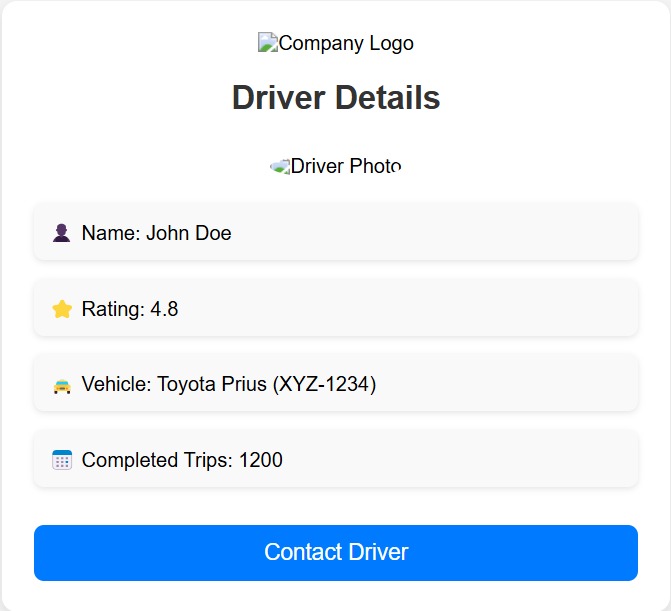


Figure 3.4‑ Screen-4: View Driver Details

**Purpose:** This screen provides a detailed profile of a driver, including their name, rating, vehicle details, completed trips, and a contact option. It is likely used in a ride-sharing or logistics app to view and connect with the driver.

Table 3.4‑ Screen element of View Driver Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Company Logo | Image | M | 1 | Displays the company's logo. It is a mandatory, non-editable visual element. |
| 2 | Driver Photo | Image | M | 1 | Displays the driver's profile photo. It is a mandatory, non-editable visual element. |
| 3 | Name | Text | M | 1 | Displays the driver's full name. It is non-editable and a required element. |
| 4 | Rating | |  | | --- | |  |  |  | | --- | | Icon + Text | | M | 1 | Displays the driver's rating with a star icon. It is non-editable and a required element. |
| 5 | Vehicle Details | Text | M | 1 | Displays the vehicle make, model, and license plate. It is non-editable and mandatory. |
| 6 | Completed Trips | Text | M | 1 | Displays the total number of trips completed by the driver. It is non-editable and mandatory. |
| 7 | Contact Driver | Button | M | 1 | Button to initiate contact with the driver (e.g., via call or message). Mandatory for action. |

## Screen-5: Choose Vehicle Type

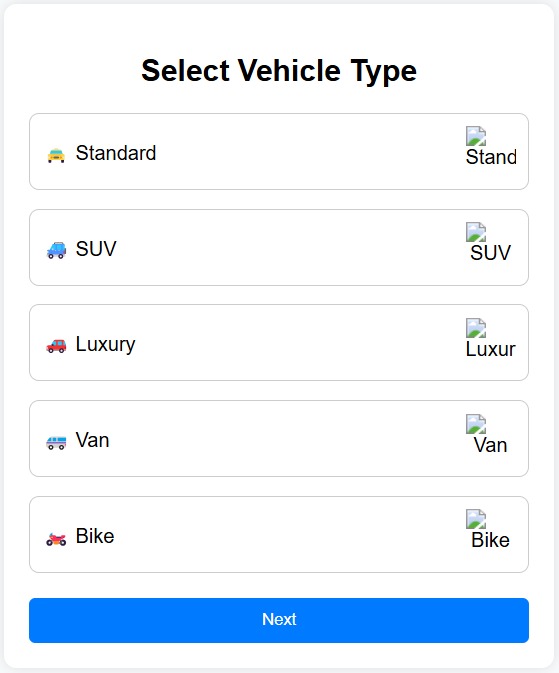


Figure 3.5‑ Screen-5: Choose Vehicle Type

**Purpose:** This screen is designed to allow the user to select a vehicle type for a specific service, such as ride-sharing or delivery. The user can choose among various vehicle options like Standard, SUV, Luxury, Van, or Bike before proceeding to the next step.

Table 3.5‑ Screen element of Choose Vehicle Type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Select Vehicle Type | Title | M | 1 | Indicates the purpose of the screen. Non-editable and mandatory for user guidance. |
| 2 | Standard | |  | | --- | | Radio Button |  |  | | --- | |  | | M | N | Option for selecting a standard vehicle. The user must choose one vehicle type to proceed. |
| 3 | SUV | |  | | --- | | Radio Button |  |  | | --- | |  | | M | N | Option for selecting an SUV. |
| 4 | Luxury | |  | | --- | | Radio Button |  |  | | --- | |  | | M | N | Option for selecting a luxury vehicle. |
| 5 | Van | |  | | --- | | Radio Button |  |  | | --- | |  | | M | N | Option for selecting a van. |
| 6 | Bike | |  | | --- | | Radio Button |  |  | | --- | |  | | M | N | Option for selecting a bike. |
| 7 | Next | |  | | --- | | Button |  |  | | --- | |  | | M | 1 | Button to confirm the selected vehicle type and proceed to the next screen or process. |

# Database design

## List of Tables

* Users
* Vehicle
* Payments
* Trips
* Feedback

Table 4.1‑1 Table: Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| user\_id | int | Not Null | PK (Auto Increment) |  |
| first\_name | varchar(100) | Not Null |  |  |
| last\_name | varchar(100) | Not Null |  |  |
| email | varchar(100) | Not Null |  |  |
| phone | varchar(50) | Not Null |  |  |
| password | varchar(100) |  |  |  |
| role\_id | int |  | FK |  |
| status | ENUM |  |  |  |
| profile\_picture | BLOB |  |  |  |

Table 4.1‑2 Table: Vehicle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| vehicle\_id | int | Not Null | PK (Auto Increment) |  |
| driver\_id | int | Not Null | FK | Reference of Driver Table |
| vehicle\_type | Varchar(50) | Not Null |  |  |
| number\_plate | Varchar(50) | Not Null |  |  |
| status | ENUM | Not Null |  |  |
| location | Varchar(50) | Not Null | FK |  |

Table 4.1‑3 Table: Payments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| payment\_id | int | Not Null | PK (Auto Increment) |  |
| trip\_id | int | Not Null | FK | Reference of Trip Table |
| amount | Decimal(10,2) | Not Null |  |  |
| payment\_method | ENUM | Not Null |  |  |
| status | ENUM | Not Null |  |  |
| created\_at | datetime | Not Null |  |  |

Table 4.1‑4 Table: Trips

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| trip\_id | int | Not Null | PK (Auto Increment) |  |
| customer\_id | int | Not Null | FK | Reference of customer Table |
| driver\_id | int | Not Null | FK | Reference of driver Table |
| pickup\_location | Varchar(100) | Not Null |  |  |
| dropoff\_location | Varchar(100) | Not Null |  |  |
| distance | decimal | Not Null |  |  |
| duration | int | Not Null |  |  |
| fare | decimal | Not Null |  |  |
| status | ENUM | Not Null |  |  |
| created\_at | datetime | Not Null |  |  |

Table 4.1‑5 Table: Feedback

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| feedback\_id | int | Not Null | PK (Auto Increment) |  |
| user\_id | int | Not Null | FK | Reference of user Table |
| trip\_id | int | Not Null | FK | Reference of trip Table |
| rating | Int | Not Null |  |  |
| comments | text | Not Null |  |  |
| created\_at | datetime | Not Null |  |  |

# Stories and Scenario

## Story-1: Add New Taxi

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Admin,  I want to add a new taxi to the system So that everyone can easily book a taxi. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | The addition of a new taxi is crucial for ensuring that the fleet is up-to-date and accessible to customers. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Taxi with Valid Information |
| Prerequisite | **:** | Admin is logged into the Taxi Management System. |
| Acceptance Criteria | **:** | **Given**: The Admin is navigated to the taxi management page. Valid taxi information, including license plate number, model, capacity, and other relevant details, is added.  **When**:  The Admin selects the "Add New Taxi" option  And the Admin enters valid taxi details  The Admin clicks the "Save" button to add the taxi to the fleet.  **Then**: The system successfully adds the taxi to the fleet, and the Admin receives a confirmation message with the taxi's identification number |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Taxi with Invalid Information |
| Prerequisite | **:** | Admin is logged into the Taxi Management System. |
| Acceptance Criteria | **:** | **Given**: The Admin is on the taxi management page **When**: The Admin selects the "Add New Taxi" option and enters incomplete or incorrect taxi details, and the Admin clicks the "Save" button to add the taxi to the fleet. **Then**: The system displays error messages for the incorrect or missing information, and the taxi is not added to the fleet. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Attempting to Add a Duplicate Taxi |
| Prerequisite | **:** | Admin is logged into the Taxi Management System and is on the taxi management page. |
| Acceptance Criteria | **:** | **Given**: The taxi information, including license plate number, model, capacity, and other relevant details, is available, and the taxi with the same license plate number is already in the fleet. **When**: The Admin clicks on the “Add Taxi” button and enters duplicate taxi details. **Then**: The system alerts the Admin about the duplicate taxi and does not add it to the fleet. |

## Story-2: Booking Taxi

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Customer,  I want to book a taxi,  So that I can travel to my destination easily. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Booking a taxi is the core functionality of the system and is crucial for user satisfaction. |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Booking a Taxi with Valid Information |
| Prerequisite | **:** | Customer is logged into the Taxi Management System. |
| Acceptance Criteria | **:** | **Given**: The Customer is on the taxi booking page.  **When**:  The Customer selects pickup and drop-off locations  The Customer chooses a vehicle type and enters valid booking details  The Customer confirms the booking.  **Then**: The system confirms the booking, provides the estimated fare, and displays the driver's details and the estimated arrival time. |

### 5.2.2 Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.2* | : | Booking a Taxi with Invalid Information |
| Prerequisite | **:** | Customer is logged into the Taxi Management System. |
| Acceptance Criteria | **:** | **Given**: The Customer is on the taxi booking page. **When**: The Customer enters incomplete or incorrect booking information and confirms the booking. **Then**: The system displays error messages for missing or invalid details, and the booking is not confirmed. |