# **BUSINESS RQUIREMENT DOCUMENT (BRD)**

**Project Name: Easy Car Rental**

**Document History:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Prepare By** | **Prepare Date** | **Review By** | **Review Date** | **Change Log** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## **INDEX**

1. **Problem Statement Index**

1.1. Problem Statement & Description

2. **Project Scope (Goal of Project)**

3. **As-Is (Existing) Business Process**

3.1. UML Diagram on High Level Flowchart)

3.2. Description

3.3. UML Diagram on High Level (Activity)

3.4. Description

4. **To-Be (Solution /End Product) Business Process**

4.1. UML Diagram on High Level (Flowchart)

4.2. Description

4.1. UML Diagram on High Level (Activity)

4.2. Description

5. **Functional Requirements**

5.1. User Interface Screen 1 – Registration & Description

5.2. User Interface Screen 2 – Login & Description

5.3. User Interface Screen 3 – Enter Trip Details & Description

5.4. User Interface Screen 4 – Select Car & Description

5.5. User Interface Screen 5 – Poc & Pax Details & Description

5.6. User Interface Screen 6 – Review Trip Details & Description

5.7. User Interface Screen 7 – Payment Details & Description

5.8. User Interface Screen 8 – Confirmation Page & Description

6. **Non-Functional Requirements**

6.1. UI/UX Requirement

6.2. Static Data

6.3. Hardware Configuration Analysis

6.4. Software & Database license

6.5. Network Requirement

6.6. Hosting Requirement

6.7. Transition Requirement

6.8. Security Requirement

6.9. Performance Requirement

6.10. Third Party Integrations

7. **Assumptions & Constraint**

7.1. Assumption

7.2. Constrained

### **Problem Statement:**

#### **1.1 Problem Statement**

Our organization’s current manual processes are inefficient, error-prone, and time-consuming. These issues lead to delays, inconsistencies, and higher operational costs, hindering productivity and scalability. To overcome these challenges, we aim to develop software that automates these processes, enhancing efficiency, accuracy, and overall business performance.

**Description:**

Our project involves designing and developing a software solution to automate the existing manual processes within our organization. The key objectives of this project are to streamline workflows, reduce human errors, and improve operational efficiency. By automating repetitive tasks, we will free up valuable time for our employees to focus on more strategic and value-added activities.

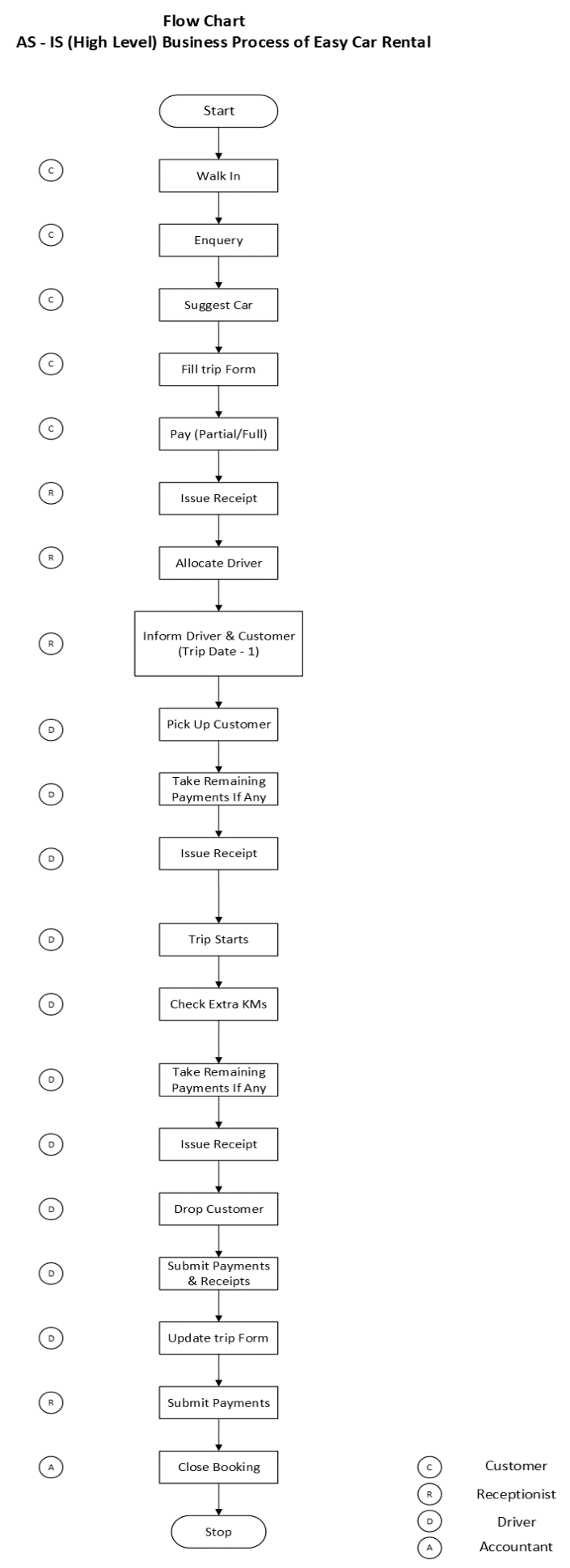
* Workflow Automation: Streamline repetitive tasks and ensure consistency.
* Data Management: Centralize data storage with validation and error-checking.
* User-Friendly Interface: Provide an intuitive interface with customizable dashboards.
* Integration Capabilities: Ensure seamless integration with existing systems.
* Scalability and Flexibility: Design for growth and adaptability.
* Enhanced Security: Implement robust data protection measures.

### **Project Scope (Goal Of Project)**

The goal of this project is to develop a comprehensive software solution to automate our organization's manual processes, thereby increasing operational efficiency, accuracy, and overall productivity.

### **As-Is (Existing) Business Process**

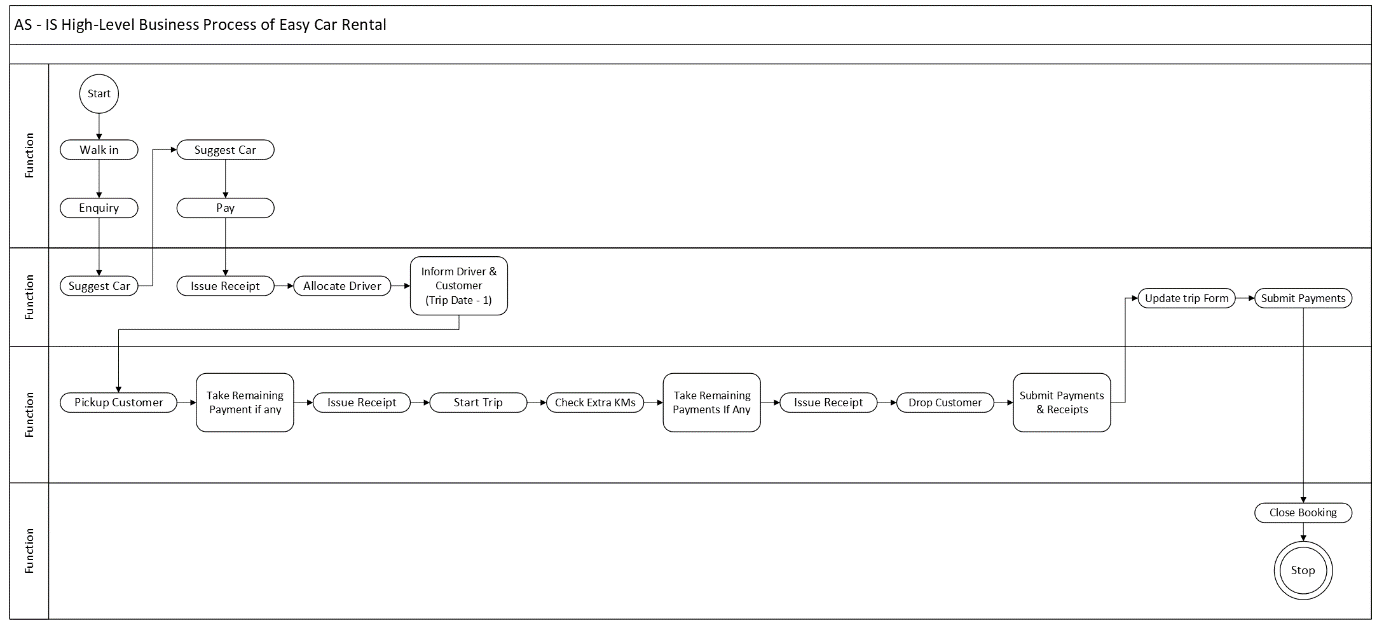
#### **3.1 UML Diagram on High Level (Flowchart)**



#### **3.2 Description**

In the current manual car rental process, customers visit the office to register or log in and discuss trip details with the staff. The staff manually searches for available cars and records passenger and driver information. Payment is completed in person, and a booking confirmation is issued. On the rental day, customers pick up the car after verifying their identity. During the rental, support is provided via phone or in-person visits. At the end of the rental, the car is returned, inspected for damages, and any additional charges are processed. This process is labor-intensive and prone to errors due to its reliance on manual tasks and paperwork

#### **3.3 UML Diagram on High Level (Activity Diagram)**

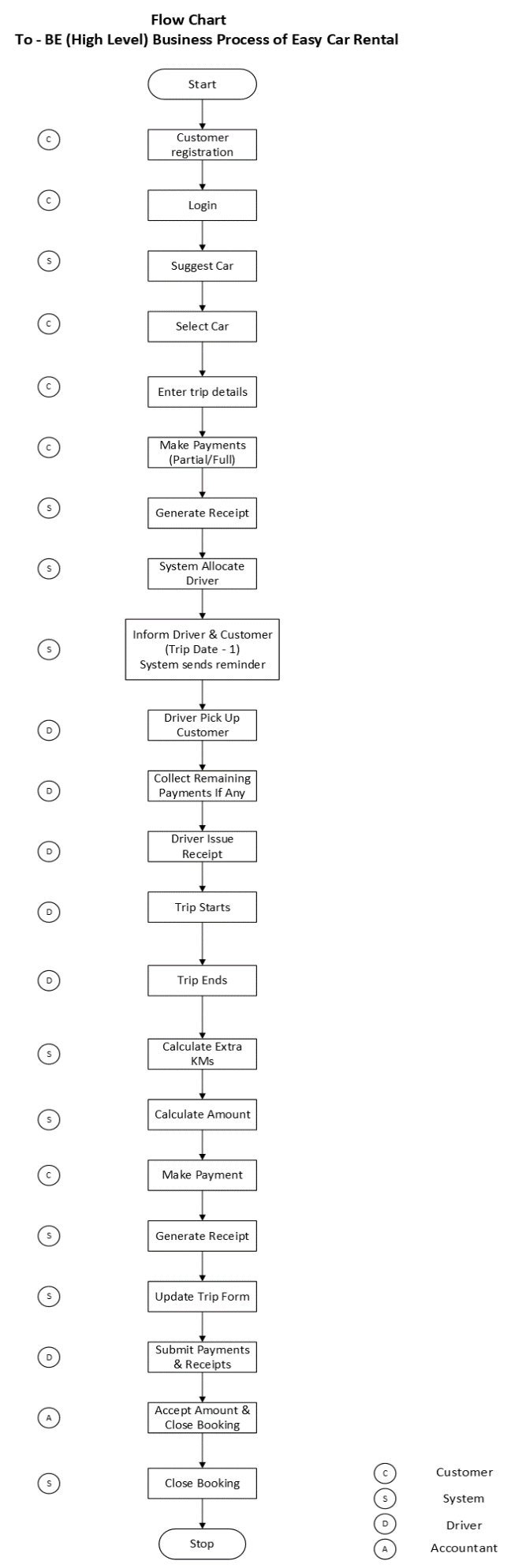


#### **3.4 Description**

In the current manual car rental process, the high-level activity diagram outlines the key steps involved. It begins with the customer visiting the car rental office to register or log in. Following this, the customer discusses their trip details with the staff, including pick-up and drop-off locations, dates, and times. The staff manually searches for available cars based on these details and presents the options to the customer, who then selects a car. Passenger and driver information is recorded manually. The customer proceeds to select a payment method and completes the transaction in person, with the staff generating a booking confirmation and receipt. On the day of the rental, the customer picks up the car from the designated location after verifying their identity. Throughout the rental period, support is available via phone or in-person visits. Upon returning the car, the staff inspects it for damages, processes any additional charges, and collects feedback from the customer. This manual process is characterized by multiple human interactions and paper-based steps, leading to inefficiencies and a higher likelihood of errors.

### **To-Be (Solution /End Product) Business Process**

#### **4.1 UML Diagram on High Level (Flowchart)**

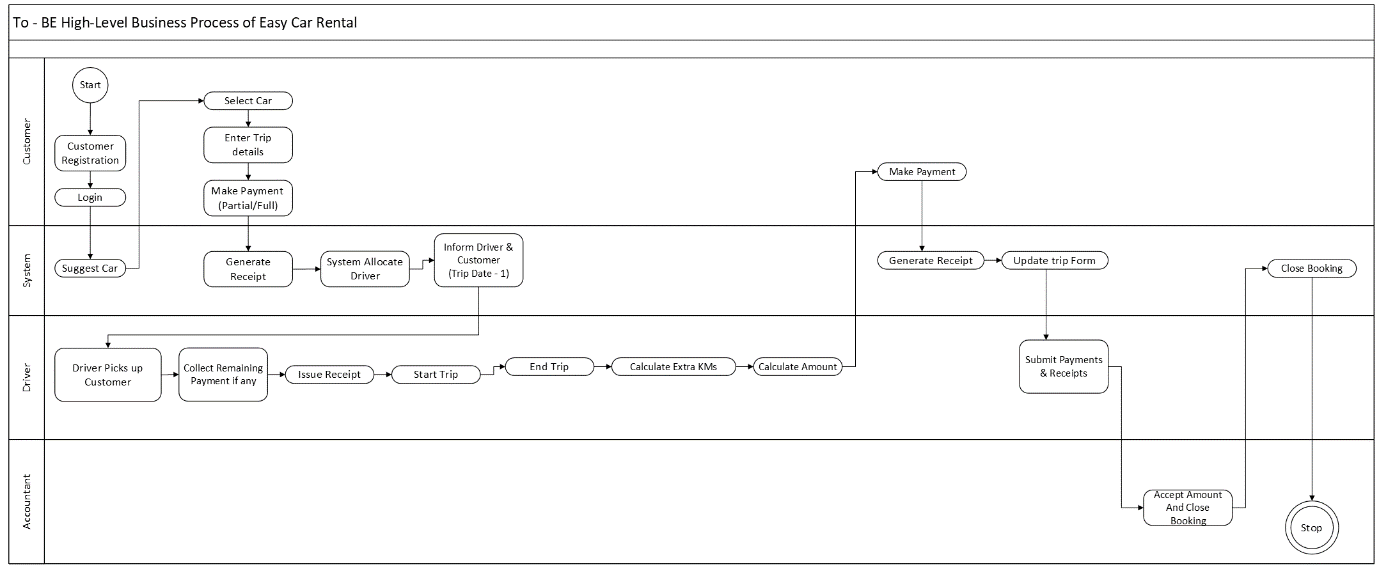


#### **4.2 Description**

In the automated car rental process, customers begin by registering or logging into the app, where they input essential trip details such as pick-up and drop-off locations, dates, and times. The app processes these details to search for available cars, allowing users to view and filter options based on their preferences, such as transmission type and car type. After selecting a car, users enter required passenger and driver information directly into the app. Payment is completed securely through integrated gateways, with options including credit card, debit card, UPI, or QR code scanning. If the payment is not successfully completed, the system prompts the user to reattempt payment. The process will only proceed once payment is confirmed.

Upon successful payment, the system generates an electronic booking confirmation and receipt, which are sent to the user via email and made available for download. The system also sends a reminder to the customer one day before the travel date. On the travel day, the customer starts the trip by picking up the car and ends it by returning the car to the designated location. Any extra charges are managed through the system, and user feedback is collected through automated surveys. This streamlined, automated approach reduces manual intervention, speeds up the process, and minimizes errors, ensuring a more efficient and user-friendly rental experience.

#### **4.3 UML Diagram on High Level (Activity Diagram)**

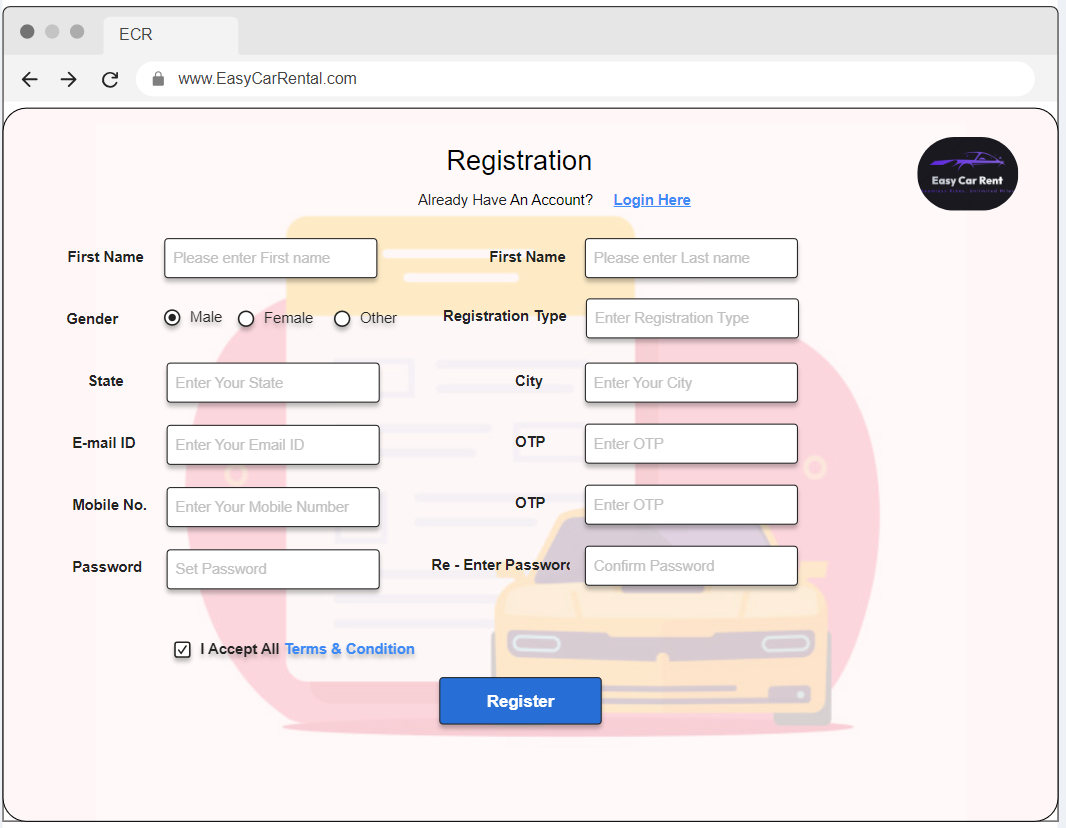


#### **4.4 Description**

In the automated car rental process, the activity diagram begins with the customer registering or logging into the app. They then input trip details, such as pick-up and drop-off locations, dates, and times. The app searches for available cars based on these details, and the customer selects a car and provides passenger and driver information. Payment is processed through secure gateways; if payment fails, the system prompts for reattempt until successful. Once payment is confirmed, an electronic booking confirmation and receipt are generated and sent to the customer. The system also sends a reminder one day before the travel date. On the day of the trip, the customer picks up the car, uses it, and returns it to the designated location. Any extra charges are handled through the app, and user feedback is collected through automated surveys.

### **05. Functional Requirements**

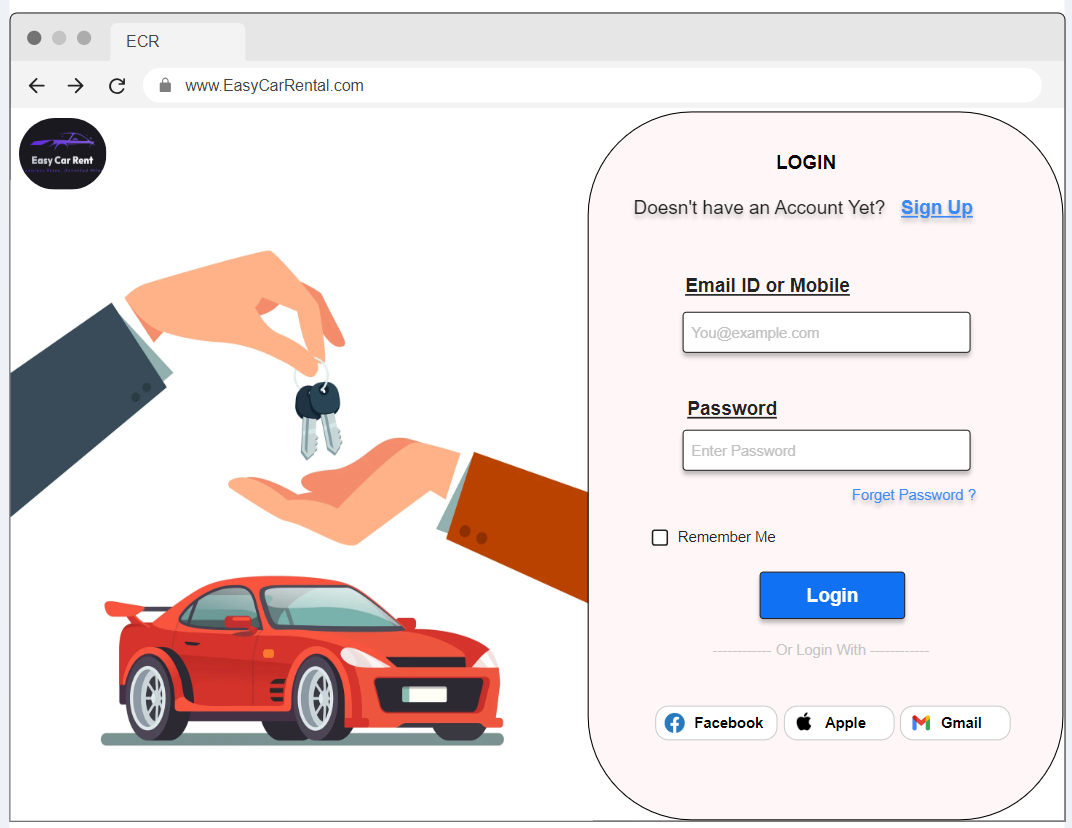
#### **5.1. User Interface Screen 1(Registration) & Description**



Description:

The ECR Registration page features a clean and user-friendly design with a right-aligned logo and a login link for returning users. The main form is organized into a single column, beginning with fields for First Name, Gender (Male, Female, Other), State, Email ID (with OTP verification), Mobile Number (with OTP verification), and Password. It continues with Last Name, Registration Type, City, OTP fields for email and mobile verification, and a Confirm Password field. This streamlined layout ensures all necessary information is collected efficiently. A checkbox for accepting terms and a prominent "**Register**" button is included. The layout is responsive and visually appealing, with clear labels and placeholders to guide users.

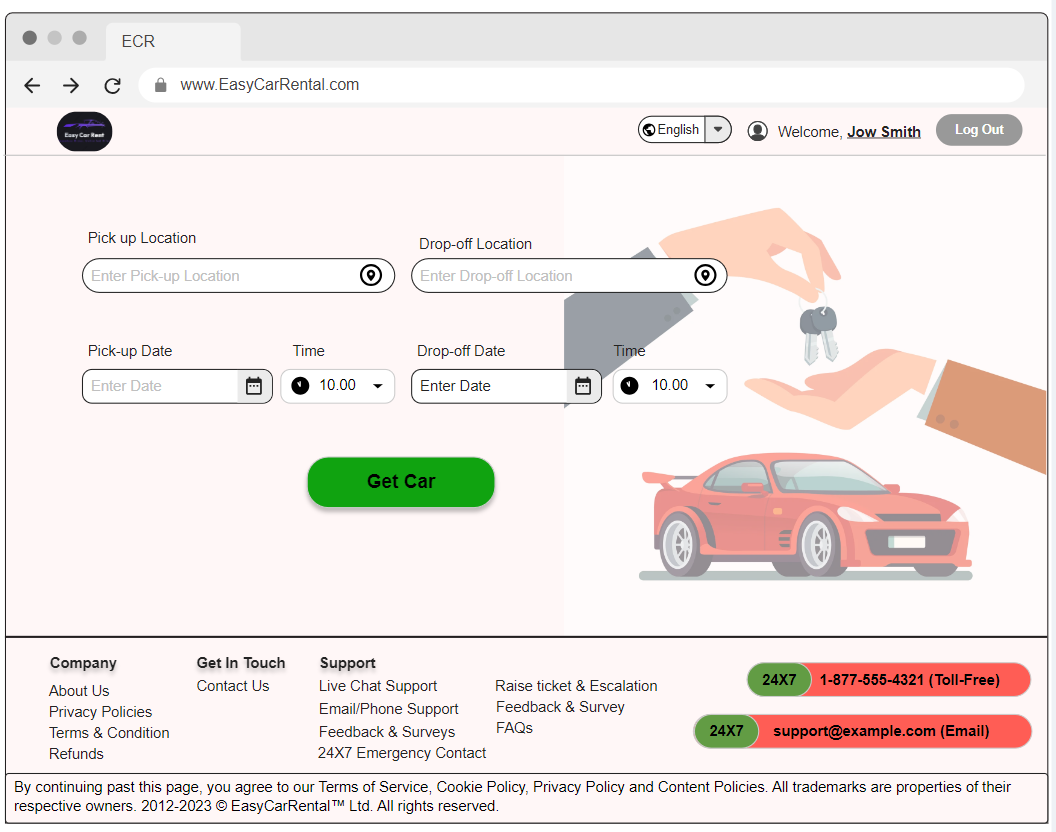
#### **5.2. User Interface Screen 2(Login) & Description**



Description:

The ECR Registration page features a clean and user-friendly design with a left-aligned logo and a link for returning users to log in. The login page maintains a simple and clean layout, centered around a single-column form for Email ID and Password. Below these fields, users can click the prominent "Login" button to submit their credentials. Additional sign-in options include Apple, Gmail, and Facebook, providing alternative ways to log in. There is also a "Forgot Password?" link for users who need to reset their password. The design ensures a straightforward and efficient login process, with a responsive layout suitable for desktops, tablets, and smartphones.

#### **5.3. User Interface Screen 3(Trip Details) & Description**



Description:

The ECR Trip Details page is designed to streamline the process of booking a car rental with a simple and intuitive user interface. At the top of the page, the header includes the ECR logo on the left, ensuring brand recognition, and a user panel on the right displaying the logged-in user's name ("Jow Smith"), a language selection dropdown, and a "Log Out" button for easy access to account management.

The main content area is dedicated to the trip booking form, which is structured to efficiently gather essential rental information. The form includes fields for:

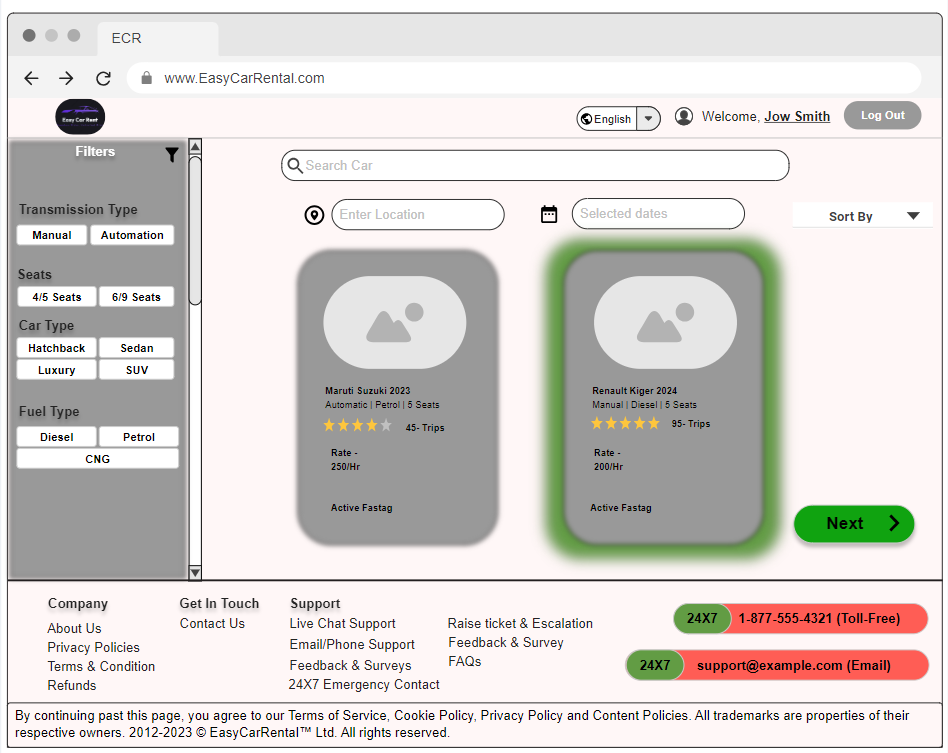
* Pick-up Location: A text input field with a dropdown for selecting the location where the user will pick up the car.
* Drop-off Location: A text input field with a dropdown for selecting the location where the user will drop off the car.
* Pick-up Date and Time: A date picker for selecting the pick-up date, and a time dropdown for selecting the pick-up time.
* Drop-off Date and Time: A date picker for selecting the drop-off date, and a time dropdown for selecting the drop-off time.

Below the form fields, a prominent "Get Car" button allows users to submit their booking details.

The page footer provides additional resources and support information divided into three sections:

* Company: Links to About Us, Privacy Policies, Terms & Conditions, and Refunds.
* Get In Touch: Contact Us link.
* Support: Links to Live Chat Support, Email/Phone Support, Feedback & Surveys, and a 24x7 Emergency Contact section with a toll-free number and email address for immediate assistance.

#### **5.4 User Interface Screen 4(Select CAR) & Description**



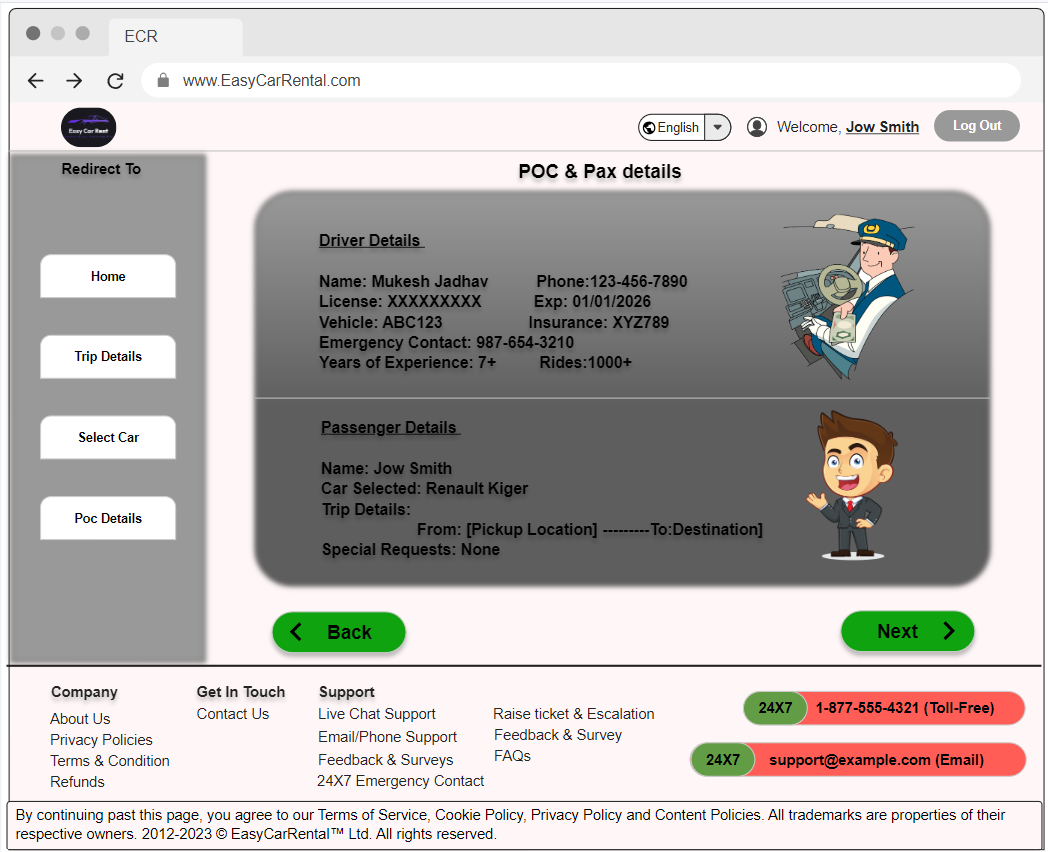
Description:

The ECR Select Car page provides an intuitive interface for choosing a rental vehicle. The header features the ECR logo, language selection, user account options, and a log-out button. The main content area includes a Filters Sidebar for refining search by transmission type, seating capacity, car type, and fuel type. A search and sort bar allows users to search specific car models, input pick-up location, select rental dates, and sort listings.

Car listings display detailed information, including model and year, transmission type, fuel type, seating capacity, rating, hourly rate, and Active Fastag status. Each listing has an image placeholder and a "Next" button for selection.

The footer provides links to About Us, Privacy Policies, Terms & Conditions, Refunds, Contact Us, and various support options, including Live Chat, Email/Phone Support, Feedback & Surveys, and 24x7 Emergency Contact. This design ensures a seamless car selection process and easy access to support and company information.

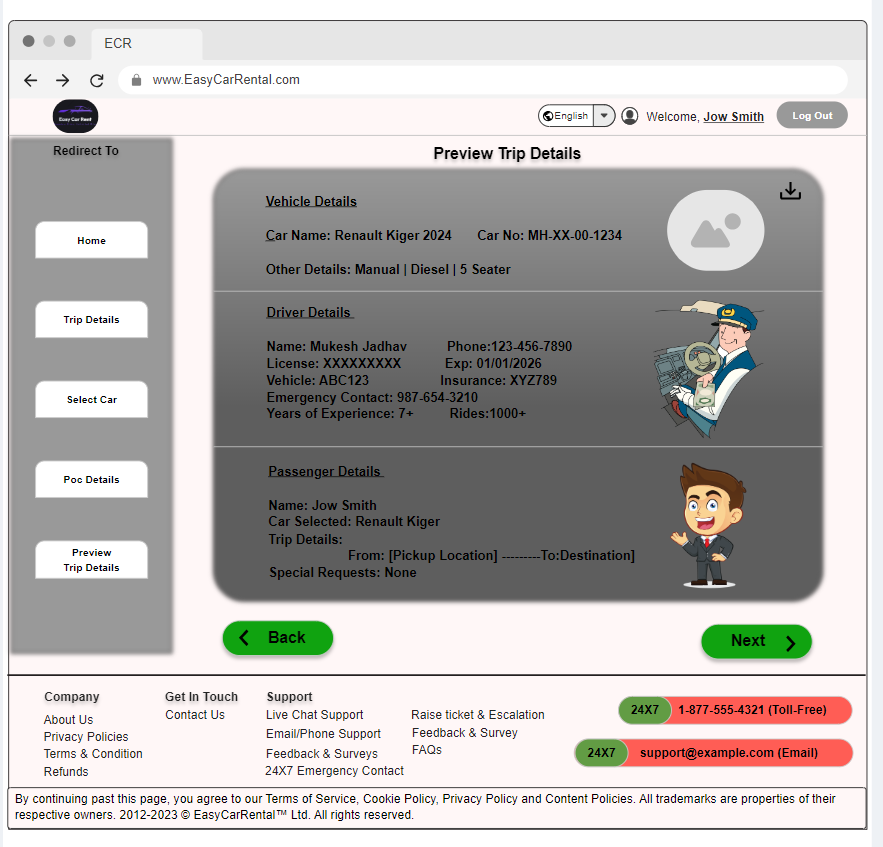
#### **5.5. User Interface Screen 5(POC & PAX Details) & Description**



Description:

The user interface screen for Customer and Passenger Details, presented after car selection, features non-editable driver information and pre-filled customer details retrieved during registration. Users have the option to add or edit passenger information, with fields for entering names, ages, contact numbers, and email addresses. The screen includes a Back button to return to the previous step and a Next button to move forward in the booking process. This design ensures that while core information remains fixed, users can still customize passenger details and navigate seamlessly through the booking stages.

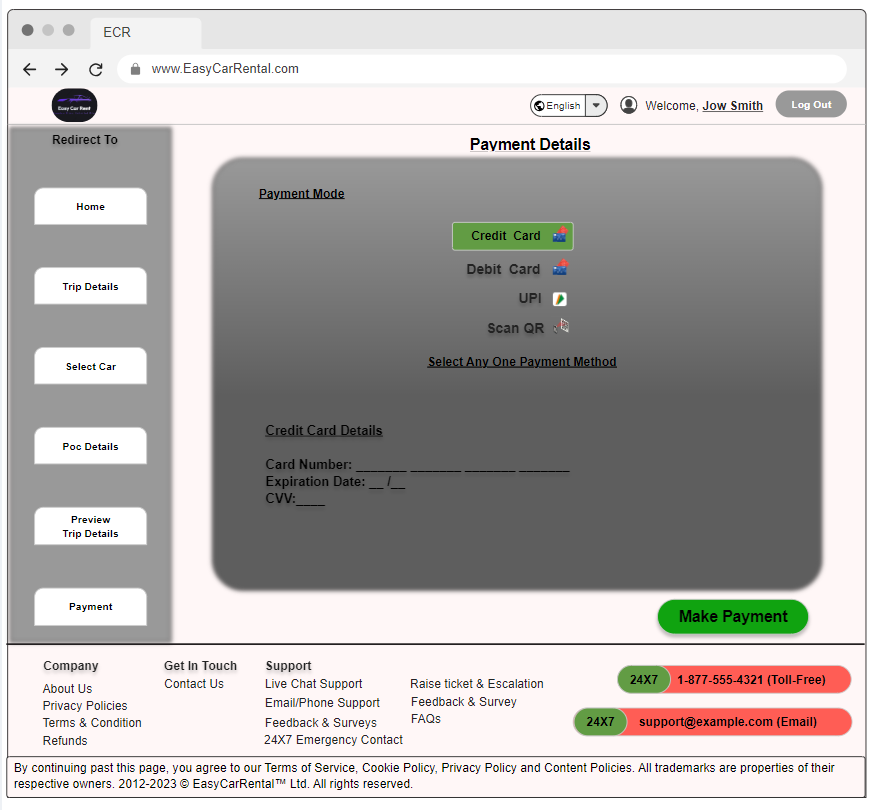
#### **5.6. User Interface Screen 6(Review Trip) & Description**



Description

The Preview Trip Details screen provides a comprehensive review of the booking information prior to final confirmation. It displays editable sections for car information and passenger details, allowing users to make any necessary changes before proceeding. Driver details are displayed but remain non-editable, reflecting the information retrieved earlier in the process. Users can download the trip details in PDF format for their records. The screen includes a Back button to return to the previous step and a Next button to move forward in the booking process. Additionally, there are multiple redirect buttons for navigating to different sections of the app, such as Home, Trip Details, Select Car, and Passenger Details, facilitating easy access to various features and information.

#### **5.7. User Interface Screen 7 (Payment Details) & Description**

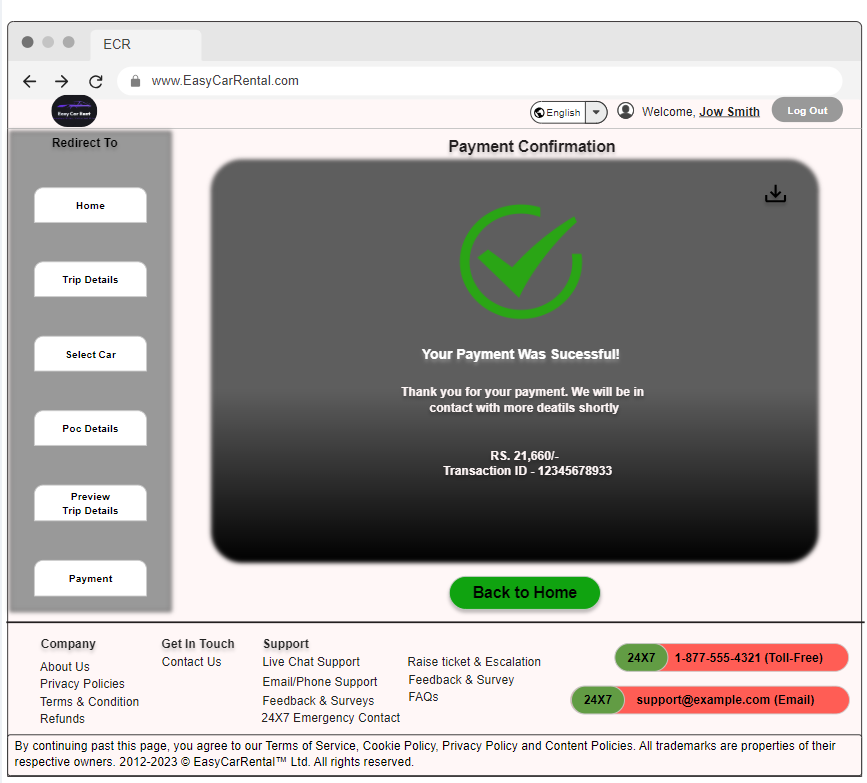


Description

The Payment Details UI screen enables users to complete their payment process with multiple options. The screen features a set of radio buttons for selecting the payment mode, including Credit Card, Debit Card, UPI, and QR Code. Each payment method provides its own input fields, which become visible based on the selected option. Users can enter their payment details directly into the respective fields. When a payment option is selected, it is highlighted in a green box for clear visibility.

The screen ensures that the Make Payment button is enabled only when a payment option is selected and all required payment details are filled. Users can navigate to other sections of the app through buttons like Company, About Us, Privacy Policies, Terms & Conditions, Refunds, and Contact Us. These buttons allow users to access additional information or support.

#### **5.8. User Interface Screen 8(confirmation) & Description**



Description:

The Payment Confirmation UI screen provides users with a summary of their payment transaction. Upon successfully completing a payment, this screen displays key information, including the payment amount, the selected payment method, and the transaction status.

Users can review a detailed breakdown of the payment, including any applicable taxes and fees. A confirmation message reassures users that their payment has been processed successfully. The screen also features a unique transaction reference number for users to keep for their records.

Additionally, users have the option to download or print the payment confirmation for future reference. Navigation buttons allow users to return to the home page, view their booking details, or proceed to other sections of the app, such as customer support or feedback forms. The design ensures that users can easily access all necessary information and perform any follow-up actions as needed.

### **06. Non-Functional Requirements**

#### **6.1. UI/UX Requirement**

User Interface Consistency: The system must have a consistent look and feel across all pages, adhering to a standard design language.

Ease of Use: Users should be able to complete a rental transaction in no more than three steps.

Accessibility: The interface must comply with WCAG 2.1 standards to ensure accessibility for users with disabilities.

#### **6.2. Static Data**

Data Storage: Static data such as car models, rental locations, and pricing tiers must be stored in a way that is easily retrievable and manageable.

Data Updates: Static data must be easily updatable by authorized personnel without requiring system downtime.

#### **6.3. Hardware Configurations**

Server Specifications: The servers should have a minimum of 16GB RAM, 8-core CPU, and SSD storage to handle high traffic and ensure fast response times.

Redundancy: Hardware configurations must include redundancy to prevent single points of failure and ensure high availability.

#### **6.4. Software & Database License**

Licensing Compliance: Ensure all software and database components are properly licensed and compliant with legal requirements.

Scalability: Licensed software and databases should support scalability to handle increased loads without performance degradation.

#### **6.5. Network Requirement**

Bandwidth: The network should have sufficient bandwidth to support peak user loads without delays.

Security: Implement network security measures such as firewalls and VPNs to protect against unauthorized access.

#### **6.6. Hosting Requirement**

Hosting Environment: The system should be hosted in a cloud environment with high availability and disaster recovery capabilities.

Data Center Compliance: Ensure the hosting provider complies with industry standards such as ISO/IEC 27001 for data security.

#### **6.7. Transition Requirement**

Data Migration: Plan and execute a seamless data migration from any existing system to the new car rental system without data loss.

Training: Provide training sessions for staff to ensure they are familiar with the new system before full deployment.

#### **6.8. Security Requirement**

Data Encryption: Encrypt all sensitive data both in transit and at rest.

Access Control: Implement role-based access control (RBAC) to restrict access to sensitive information and functionalities.

#### **6.9. Performance Requirement**

Response Time: The system should respond to user actions within 2 seconds.

Load Handling: The system should handle at least 10,000 concurrent users without performance degradation.

#### **6.10. Third Party Integrations**

Payment Gateway Integration: Integrate with multiple payment gateways to offer customers various payment options.

CRM Integration: Ensure the system integrates seamlessly with customer relationship management (CRM) software for customer data management.

### **07. Assumption and Constraints**

#### **7.1 Assumptions**

Stable Internet Connection: It is assumed that users and administrators will have a stable internet connection to access the system.

User Technical Proficiency: Users have basic computer literacy and can navigate a web-based interface.

Car Availability: It is assumed that the fleet of cars will be sufficient to meet the demand of the users.

Third-Party Services: Payment gateways, CRM systems, and other third-party services will be reliable and have minimal downtime.

Regulatory Compliance: The project will comply with local, state, and federal regulations regarding car rental services.

Data Integrity: Data imported from any existing systems will be clean, accurate, and complete.

Training Provided: Adequate training will be provided to all staff members to ensure they can effectively use the system.

Support Availability: Technical support will be available during business hours to resolve any issues that arise.

#### **7.2 Constraints**

Budget Limitations: The project must be completed within the allocated budget, restricting the extent of customizations and features.

Time Frame: The project must be delivered within a specified time frame, typically determined by business needs or market conditions.

Compliance Requirements: The system must comply with industry standards and regulations such as GDPR, PCI-DSS, and local laws.

System Integration: Integration with existing systems (e.g., CRM, payment gateways) must be seamless and cannot disrupt current operations.

Data Security: Strict security protocols must be followed to protect sensitive customer and business data.

Scalability: The system must be designed to scale with the business growth, handling increased user loads and data volume.

User Accessibility: The system must be accessible to users with disabilities, complying with WCAG 2.1 standards.

Technical Limitations: The project must work within the technical constraints of the chosen platform and technologies.

Resource Availability: The availability of project resources (developers, testers, etc.) may be limited, affecting project timelines and deliverables.

Hosting Environment: The system must be compatible with the chosen hosting environment and adhere to its limitations and requirements.