

Car Rental Requirement Specification

Version 2.0

April 03, 2025

Klaudja Bega

Adrian Malka

Franceska Keci

Kristi Hila

Kristaq Mëhilli

Gregori Haxhia

Helio Myrteza

Table of Contents

1. EXECUTIVE SUMMARY

- 1.1 Project Overview
- 1.2 Purpose and Scope of this Specification

2. PRODUCT/SERVICE DESCRIPTION

- 2.1 Product Context
- 2.2 User Characteristics
- 2.3 Assumptions
- 2.4 Constraints
- 2.5 Dependencies

3. REQUIREMENTS

- 3.1 Functional Requirements
- 3.2 Non-Functional Requirements
 - 3.2.1 Product Requirements
 - 3.2.1.1 User interface
 - 3.2.1.2 Usability
 - 3.2.1.3 Efficiency
 - 3.2.1.3.1 *Performance Requirements*
 - 3.2.1.3.2 *Space Requirements*
 - 3.2.1.4 Dependability
 - 3.2.1.5 Security
 - 3.2.2 Organizational Requirements
 - 3.2.2.1 Environmental Requirements
 - 3.2.2.2 Operational Requirements
 - 3.2.2.3 Development Requirements
 - 3.2.3 External Requirements
 - 3.2.3.1 Regulatory Requirements
 - 3.2.3.2 Ethical Requirements
 - 3.2.3.3 Legislative Requirements
 - 3.2.3.3.1 *Accounting Requirements*
 - 3.2.3.3.2 *Security Requirements*
- 3.3 Domain Requirements

4. SOFTWARE DESIGN/ DIAGRAMS

- 4.1 Requirements Analysis
 - 4.1.1 User scenarios
 - 4.1.1.1 User scenarios List
 - 4.1.1.2 User Scenarios Extended
 - 4.1.2 User cases
- 4.2 Behavioural Diagrams
 - 4.2.1 Use Case Diagrams
 - 4.2.2 Activity Diagrams

1. Executive Summary

1.1 Project Overview

The **Car Rental System** is a web-based platform designed to facilitate the seamless rental of vehicles for customers while providing an efficient management system for car rental businesses. This system allows users to **browse available cars, book rentals, make payments, and manage their reservations**. Additionally, the system provides administrators with tools to **manage users, cars, bookings, and financial transactions**.

The platform aims to **streamline the car rental process**, ensuring a **user-friendly experience**, **automated payment processing**, and **real-time availability updates** to enhance operational efficiency.

Intended Audience

- **Customers (Renters):** Individuals who need to rent a car for personal or business use. They can create an account, search for cars, make bookings, and complete payments.
- **Administrators:** Rental business owners or employees responsible for managing vehicles, managing customer requests, and overseeing payments.
- **Car Rental Companies:** Businesses looking to digitize and automate their car rental operations to improve service quality and efficiency.
- **Payment System Providers:** Third-party payment gateways integrated for secure transactions.

1.2 Purpose and Scope of this Specification

Purpose

The purpose of this specification document is to outline the functional and non-functional requirements, design considerations, and operational features of the Car Rental System. This document serves as a guideline for developers, business analysts, testers, and stakeholders involved in the project's development and implementation.

It ensures that all project objectives, constraints, and system functionalities are clearly defined to facilitate a smooth development process while meeting the needs of both customers and administrators.

In Scope

This specification covers all essential requirements and functionalities for the **Car Rental System**, including:

1. **User Authentication & Management** – Registration, login, profile management, and access control for customers and admins.
2. **Car Inventory Management** – Adding, updating, and removing cars from the system.
3. **Booking & Reservation System** – Customers can view available cars, make bookings, and modify/cancel reservations.
4. **Payment Processing** – Integration with secure payment gateways to facilitate transactions.
5. **Admin Dashboard** – Management of cars, users, bookings, and financial records.

6. **Notifications & Alerts** – Email/SMS notifications for booking confirmations, reminders, and payment receipts.
7. **Review & Rating System** – Customers can leave feedback on their rental experience.
8. **Reports & Analytics** – Data insights for admins on bookings, revenue, and car usage trends.
9. **Security & Compliance** – Secure handling of user data, compliance with data protection regulations.

Out of Scope

The following features are outside the scope of this specification and may be considered in future phases:

1. **Self-Driving Car Rentals** – Autonomous vehicle support is not included in the initial phase.
2. **Peer-to-Peer Car Sharing** – The system does not support private car owners renting out their vehicles.
3. **Advanced AI-Based Pricing** – Dynamic pricing based on real-time market trends is not included in the first phase.
4. **Integration with Ride-Sharing Services** – The system is focused on car rentals, not ride-hailing services like Uber/Lyft.
5. **International Rentals** – The system is designed for use within a specific region/country in the initial release.

2. Product/Service Description

2.1 Product Context

- System Overview:

The **Car Rental System** is a **self-contained** web application designed to facilitate vehicle rentals while integrating with various external services to enhance its functionality. The system allows **customers** to rent cars, **administrators** to manage bookings and vehicles, and **payment gateways** to process transactions securely.

While the system operates **independently**, it interacts with **external services** such as:

1. **Payment Gateways (e.g., PayPal, Stripe)** – To securely process online payments.
2. **Identity Verification Systems** – To verify user identity for fraud prevention.
3. **Third-Party Notification Services (e.g., Twilio, Email APIs)** – To send booking confirmations and reminders.
4. **External Vehicle Maintenance Systems** – To track vehicle maintenance schedules and status.

- System Components & Interconnections

The diagram below illustrates the **Car Rental System's major components** and how they interact with external systems:

Internal System Components:

- **Customer Module** – Allows users to search, book, and manage rentals.
- **Admin Dashboard** – Enables business owners to manage vehicles, users, and bookings.

- **Car Inventory Module** – Handles the addition, removal, and availability of cars.
- **Payment Processing** – Manages transactions and invoicing.
- **Notification System** – Sends real-time booking confirmations and updates.

External Interfaces:

- **Payment Gateways** – Secure transaction processing.
- **Notification APIs** – SMS/Email alerts for booking reminders.
- **Vehicle Maintenance System** – Ensures cars are regularly serviced.
- **Identity Verification API** – Prevents fraudulent bookings.

2.2 User Characteristics

The **Car Rental System** will serve multiple user types, each with distinct needs, technical skills, and usage behaviors. Below are the primary user profiles:

1. Customers (General Public, Students, Faculty, Staff, etc.)

- Type: General users renting vehicles for personal or professional use.
- Experience: Varies from first-time renters to frequent users.
- Technical Expertise: Basic to moderate (familiar with websites and online bookings).
- Other Characteristics:
 - Expect a user-friendly interface with minimal learning curve.
 - Require quick and seamless booking processes.
 - Prefer mobile-friendly design for easy access.
 - Need real-time notifications (email/SMS reminders).
 - Security-conscious; value secure payment options and identity verification.

2. Administrators (Car Rental Business Owners/Managers)

- **Type:** Business owners, rental managers, and staff handling bookings, payments, and fleet management.
- **Experience:** Experienced in car rental operations and customer service.
- **Technical Expertise:** Moderate to high (familiar with dashboards, reports, and management tools).
- **Other Characteristics:**
 - o Need a **centralized dashboard** for managing bookings, payments, and car inventory.
 - o Require **reporting and analytics** for decision-making.
 - o Need **role-based access control** for staff permissions.
 - o Expect **maintenance tracking** for vehicles.
 - o Require **fraud detection** and **identity verification** to prevent fake bookings.

3. System Administration (IT Support & Developers)

- **Type:** Technical support personnel responsible for system maintenance, updates, and security.
- **Experience:** Experienced in IT infrastructure, database management, and cybersecurity.

- **Technical Expertise:** Advanced (familiar with system administration, databases, and troubleshooting).

- **Other Characteristics:**

- Need **access to system logs** and **error reports**.
- Require **database management tools** for performance optimization.
- Expect **secure role-based permissions** to prevent unauthorized access.
- Manage **system security and compliance** with data protection regulations.
- Integrated **external services (payments, notifications, identity verification, etc.)**.

2.3 Assumptions

- Technical Assumptions

✓ **Internet Access** – Users (customers, admins, and system operators) will have a stable internet connection to access the system.

✓ **Device Compatibility** – The system will be accessed via desktop and mobile devices, assuming modern web browsers (Chrome, Firefox, Safari, Edge).

✓ **Server Availability** – The system will be hosted on a **secure and scalable** cloud server.

✓ **Payment Integration** – External payment gateways (e.g., PayPal, Stripe) will be **available** for transactions.

✓ **Operating System Support** – The system is assumed to be OS-independent and will function on **Windows, macOS, Linux**, and mobile platforms.

-User Related Assumptions

✓ **Basic User Technical Knowledge** – Customers will have **basic** web browsing skills to search for cars and complete bookings.

✓ **Admin Expertise** – Admin users will have moderate experience in managing bookings, handling payments, and overseeing fleet operations.

✓ **System Admin Knowledge** – IT personnel will be responsible for **server maintenance, security updates, and troubleshooting**.

-Business Assumptions

✓ **Car Availability** – The rental company will maintain an **adequate fleet** to meet customer demand.

✓ **Legal & Insurance Compliance** – Users will be required to provide **valid driver's licenses** and agree to terms & conditions for rentals.

✓ **User Verification** – The system assumes that identity verification services (e.g., government ID checks) will be available and used when required.

✓ **Booking Validity** – Customers must provide **accurate information** when making a reservation, and fraudulent bookings will be flagged.

-Security & Privacy Assumptions

✓ **Data Protection** – The system will comply with **GDPR** or other applicable data protection laws.

✓ **Secure Transactions** – Payment processing will follow **industry security standards** (PCI-DSS compliance).

✓ **Access Control** – Role-based access control will prevent unauthorized use of administrative features.

-External System Assumptions

✓ **Notification Services** – Email/SMS notification services (e.g., Twilio, SendGrid) will be available.

✓ **Maintenance Tracking** – Vehicle maintenance tracking services will be available or managed

manually by the admin.

✓ **API Stability** – Third-party APIs (payment, notifications, identity verification) will be **functional and stable**.

2.4 Constraints

1- SYSTEM INTEGRATION & PARALLEL OPERATION

+ **No Legacy System Integration** – The system will be developed as a **standalone** application, meaning it does not need to support an old legacy system.

+ **Third-Party API Dependencies** – Integration with external services (e.g., payment gateways, SMS notifications) may **introduce limitations** due to API restrictions.

2- SECURITY & ACCESS CONSTRAINTS

+ **Role-Based Access Control (RBAC)** – Different user roles (customer, admin, system operator) will have restricted access based on their permissions.

+ **Audit Functions** – The system must maintain:

- **Audit logs** to track all system activities (user logins, transactions, and modifications).
- **Error logs** for debugging and troubleshooting issues.
- + **Data Privacy & Protection** – The system must comply with **GDPR, PCI-DSS, or other regional data protection laws** to ensure secure data handling.

3- CRITICALITY OF THE APPLICATION

+ **High Reliability Required** – Since this system handles **financial transactions and customer data**, uptime and security are **critical**.

+ **Real-Time Processing** – Bookings must be processed **in real time** to avoid overbooking or conflicts.

+ **Failure Recovery** – The system must have **data backup and recovery mechanisms** to prevent data loss in case of failures.

4- SYSTEM RESOURCE CONSTRAINTS

+ **Server Performance & Storage**

- The system should be **optimized** to run on cloud-based hosting (e.g., AWS, Azure, or Google Cloud).
- Database storage must be **scalable** to handle customer records, car listings, and booking histories.
- + **Bandwidth Considerations** – The system should be optimized for **fast loading times** even with high traffic.

5- DESIGN & DEVELOPMENT CONSTRAINTS

+ **Programming Language & Frameworks**

- **Frontend:** HTML, CSS, JavaScript (React, Vue, or Angular).
- **Backend:** C# (ASP.NET Core) or Java (Spring Boot).
- **Database:** MySQL or PostgreSQL.
- + **Cross-Platform Compatibility** – The system must be responsive and work on **desktop, tablet, and mobile**.
- + **Scalability** – The architecture must allow future **feature expansions** without major redesigns.

6- COMPLIANCE & LEGAL CONSTRAINTS

- + **User Verification** – Customers must provide valid driver's licenses before booking a vehicle.
- + **Payment Security** – All transactions must comply with PCI-DSS standards for secure payments.
- + **Legal Agreement** – Users must accept terms & conditions before renting a vehicle.

2.5 Dependencies

Several dependencies impact the **Car Rental System** and must be addressed for smooth development and functionality. These dependencies include external services, system modules, and data integrations.

1-EXTERNAL SYSTEM DEPENDENCIES

- + **Payment Gateway Integration** – The system relies on third-party payment providers (e.g., **PayPal, Stripe, or credit card processors**) for secure transactions. Any downtime or API changes may affect booking payments.
- + **Identity Verification System** – A third-party identity verification service (e.g., **government ID checks, driver's license validation**) may be required before a customer can book a car.
- + **SMS & Email Notifications** – The system depends on external services like **Twilio, SendGrid, or AWS SES** for sending booking confirmations, reminders, and alerts.

2- MODULE DEPENDENCIES

- + **User Authentication Module** – Before users can book cars, the **registration & login system** must be fully functional.
- + **Car Inventory Management Module** – The booking system requires an **accurate and updated database** of available cars, their status, and pricing.
- + **Admin Dashboard** – Admins need an **operational backend dashboard** to manage bookings, users, and transactions before the system can go live.
- + **Booking & Payment Processing** – Customers must be able to reserve cars, but this module depends on:
 - **Car availability status** (from inventory).
 - **User identity verification** (from authentication).
 - **Successful payment processing** (from payment gateways).
 - + **Reports & Logs** – Audit logs and reports will depend on **user activity tracking and booking data**.

3- DATA DEPENDENCIES

- + **Real-Time Car Availability** – The system requires continuous **updates on vehicle availability** to prevent double bookings.
- + **Daily Data Backup** – The database must be **backed up daily** to prevent loss of booking and user data.
- + **Historical Data Access** – The system may require access to past rental data for **business insights and fraud prevention**.

4- HARDWARE & INFRASTRUCTURE DEPENDENCIES

- + **Hosting & Server Deployment** – The system must be hosted on a **cloud-based or on-premises server** with **sufficient processing power and storage**.
- + **API Rate Limits** – Third-party services (payment, SMS, identity verification) may have rate limits, affecting the number of requests per second.

5- LEGAL & COMPLIANCE DEPENDENCIES

- + **License & Insurance Requirements** – Customers must provide **valid driver's licenses** and meet insurance requirements before renting a vehicle.
- + **Data Privacy Regulations** – The system must comply with **GDPR, PCI-DSS, or local laws** for storing and processing customer data.

These dependencies **must be managed** carefully to ensure the system functions properly. Any issues or delays in these areas can **affect system performance and availability**.

3.Requirements

A Car Rental System is a comprehensive software solution designed to facilitate the seamless management of vehicle rentals for customers, administrators, and service providers. This system allows users to browse available cars, check their specifications, and make reservations based on their preferences and availability. Customers can register an account, provide personal details, upload necessary documents such as a driver's license, and process payments securely through integrated gateways. The system includes an advanced booking mechanism, enabling users to reserve cars for specific dates, extend rental periods, and cancel reservations if needed. Administrators and employees can manage vehicle listings, update availability status, track maintenance schedules, and handle customer inquiries efficiently. The system also features real-time analytics, allowing business owners to monitor revenue, customer preferences, and fleet performance. Security measures such as role-based access control, encrypted payment transactions, and secure data storage ensure user confidentiality and system integrity. Integration with external systems, such as GPS tracking for real-time vehicle location monitoring and automated notifications for customers regarding their bookings, enhances operational efficiency. Furthermore, audit functions, including logging user activities and generating financial reports, help in maintaining regulatory compliance and business transparency. The system is designed to be scalable, allowing businesses to expand their operations and integrate additional functionalities in the future. To ensure a smooth user experience, the platform offers an intuitive interface, multilingual support, and cross-device accessibility. The technical implementation follows industry best practices, leveraging cloud-based storage, high-performance databases, and a robust backend architecture to support concurrent users without performance degradation. The system also adheres to legal regulations regarding rental agreements, insurance policies, and customer identity verification. With its structured yet flexible design, the Car Rental System ensures efficient vehicle allocation, optimized fleet utilization, and an enhanced customer experience, ultimately contributing to business growth and customer satisfaction.

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
FR_1	The system shall allow users to register and login	This ensures that users can create accounts and access personalized services securely.	1	27/03/2025	Klaudja Bega
FR_2	The system shall allow users to reset their password	This improves user experience by allowing account recovery in case of forgotten credentials.	1	27/03/2025	Klaudja Bega
FR_3	The system shall allow users to update their profile information	User can modify their personal details such as name, contact info, and driving license.	2	27/03/2025	Klaudja Bega

CAR RENTAL Requirement Specification

FR_4	The system shall allow admins to manage user accounts (create, update, delete)	Admins have full control over user accounts to ensure proper account management and security	1	27/03/2025	Klaudja Bega
FR_5	The system shall allow admins to view user activity logs	This feature enables tracking of user actions, aiding in security monitoring and audits.	2	27/03/2025	Klaudja Bega
FR_6	The system shall allow admins to add, update, and delete cars.	Ensures that the fleet is up to date and accurately reflected in the system.	1	27/03/2025	Gregori Haxhia
FR_7	The system shall allow admins to update the availability status of cars	Prevents users from booking unavailable vehicles, reducing confusion and errors.	1	27/03/2025	Gregori Haxhia
FR_8	The system shall allow users to view available cars for rent	Users need visibility into available cars to make informed rental decisions.	1	27/03/2025	Gregori Haxhia
FR_9	The system shall allow users to filter and search for cars by brand, price, fuel type, and availability	Improves user experience by making it easier to find the desired vehicle quickly.	2	27/03/2025	Gregori Haxhia
FR_10	The system shall allow users to book a car for a specific period	A fundamental feature that enables customers to reserve cars for their needs.	1	27/03/2025	Adrian Malka
FR_11	The system shall allow users to cancel a booking before the start date	Provides flexibility to users in case their plans change.	2	27/03/2025	Adrian Malka
FR_12	The system shall prevent double booking of the same car for overlapping time periods	Ensures that no two users can reserve the same car simultaneously, avoiding conflicts.	1	27/03/2025	Adrian Malka
FR_13	The system shall allow admins to approve or reject bookings	Admins can verify user eligibility and control car allocation efficiently.	1	27/03/2025	Adrian Malka
FR_14	The system shall allow users to view their booking history	Helps user keep track of past and upcoming reservation.	2	27/03/2025	Adrian Malka
FR_15	The system shall allow users to make payments online using credit card, PayPal, or cash on pickup	Offers multiple payment methods for convenience and accessibility.	1	27/03/2025	Kristaq Mehilli
FR_16	The system shall generate an invoice for each completed booking	Provides users with a formal record of their transaction for reference.	2	27/03/2025	Kristaq Mehilli

CAR RENTAL Requirement Specification

FR_17	The system shall send payment confirmation via email	Confirms successful transactions, improving user trust and communication.	2	27/03/2025	Kristaq Mehilli
FR_18	The system shall support discount codes for promotional offers	Encourages customer engagement and incentivizes bookings through discounts.	3	27/03/2025	Kristaq Mehilli
FR_19	The system shall allow users to submit reviews and ratings for rented cars	Enhances transparency and helps future users make informed decisions.	2	27/03/2025	Franceska Keci
FR_20	The system shall allow users to view car reviews and ratings before booking	Builds trust by providing feedback from previous renters.	2	27/03/2025	Franceska Keci
FR_21	The system shall allow admins to remove inappropriate reviews	Prevents misuse of the review system by removing false or offensive content.	2	27/03/2025	Franceska Keci
FR_22	The system shall allow admins to schedule car maintenance	Ensures that vehicles remain in good condition and safe for customers.	1	27/03/2025	Helio Myrteza
FR_23	The system shall allow admins to update the maintenance status of a car	Prevents customers from booking cars that are under repair or servicing.	1	27/03/2025	Helio Myrteza
FR_24	The system shall prevent cars under maintenance from being booked	Avoids operational issues by ensuring unavailable cars do not appear in search results.	1	27/03/2025	Helio Myrteza
FR_25	The system shall log user activity for security and auditing purposes	Helps track suspicious activities and maintain a history of transaction for compliance.	2	27/03/2025	Kristi Hila
FR_26	The system shall restrict access to admin functionalities based on user roles	Ensures that only authorized personnel can perform administrative actions, enhancing security.	1	27/03/2025	Kristi Hila

3.2 Non-Functional Requirements

3.2.1 Product Requirements

3.2.1.1 User interface Requirements

The user interface for the web application should be compatible to any browser in order for the user to access it from Desktop or Mobile.

The system should provide a visually intuitive interface that allows customers to easily navigate between car listing, bookings, and payment options.

Performance & reliability: the interface must be responsive within 2 seconds, ensuring smooth user interactions.

Usability: The UI should be optimized for both mobile and desktop devices, with a clean layout and simple navigation menus, to enhance user engagement.

Security: The UI should display clear security indicators

+ Login Interface

Allows users (customers, admins, managers) to securely login to the system

- **Username/Email & Password Input Fields** – Users must enter credentials.
- **Remember Me Checkbox** – Allows users to stay logged in on trusted devices.
- **Forgot Password Link** – Redirects to a **password reset page**.
- **Login Button** – Validates user input and logs in if credentials are correct.
- **Error Messages** – Displays if the username or password is incorrect.
- **Multi-Factor Authentication (Optional)** – Sends a verification code for enhanced security.

+ User Dashboard

Provides a personalized interface where users can browse cars, make bookings, and manage their accounts.

- **Welcome Message & Profile Overview** – Displays user's name and profile picture.
- **Search & Filter Cars** – Users can search cars based on:
 - **Brand, Price, Fuel Type, Transmission (Automatic/Manual), Location, Availability.**
- **View Available Cars** – A **grid or list view** of cars with images, descriptions, rental price, and availability status.
- **Book a Car** – Select a car, choose **pickup & return dates**, confirm the booking.
- **View Booking History** – Displays past and current reservations.
- **Cancel Booking** – Option to cancel a booking before the start date.
- **Payment Section** – Allows **credit card, PayPal, or cash on pickup** payments.
- **Leave a Review** – Users can rate their rented car and provide feedback.

+ Admin Dashboard

Allows administrators to manage users, cars, bookings, and monitor system activity.

- **User Management**
 - View, create, update, and delete user accounts.
 - Reset passwords for users if needed.
 - Assign roles (Admin, Manager, Customer).
 - View **user activity logs** for security monitoring.
- **Car Management**

- Add, update, and remove cars from the system.
- Change car availability status (e.g., **Available, Booked, Under Maintenance**).
- Upload car images and descriptions.

📌 Booking Management

- View all car bookings with details (customer name, car rented, duration).
- Approve or reject booking requests.
- Cancel fraudulent or problematic bookings.

📌 System Reports & Logs

- View system performance analytics.
- Monitor booking trends and popular rental choices.
- Generate **revenue reports, user activity logs, and security logs**.

+ Manager Dashboard

Rental managers oversee bookings, car maintenance schedules, and payments.

📌 **Monitor Booking Requests** – Approve or decline bookings based on availability.

📌 **Car Maintenance Scheduling** – Mark cars as **unavailable** if they need servicing.

📌 **Payment Verification** – Ensure that online payments are completed.

📌 **Discount & Promotion Management** – Apply promo codes or seasonal discounts.

+ Car listing & Booking Interface

Allows users to browse, filter and select cars for rental

📌 **Car Gallery** – High-quality images of available cars.

📌 **Detailed Car Information** – Includes **brand, model, year, fuel type, price per day, and transmission type**.

📌 **Booking Form** – Users select:

- **Pickup Date & Location**
- **Return Date & Location**
- **Preferred Payment Method**

📌 **Instant Booking Confirmation** – If available, booking is confirmed immediately.

📌 **Price Calculation** – Calculates total price based on rental duration and any discounts applied.

+ Payment Interface

Allows users to complete securely for their bookings.

- **Choose Payment Method:**
 - **Credit/Debit Card**
 - **PayPal**
 - **Cash on Pickup**

- **Secure Payment Processing** – Encrypted transactions ensuring compliance with **PCI DSS standards**.
- **Invoice Generation** – Automatically generates an invoice for each completed booking.
- **Payment Confirmation Email** – Sent after a successful transaction.

+**Review & Rating Interface**

Allows users to rate and review the cars they have rented.

- **Star Rating System** – Users rate cars from **1 to 5 stars**.
- **Text Review Box** – Users can write detailed feedback.
- **Review Moderation** – Admins can **remove inappropriate reviews**.

3.2.1.2 Usability

Users should be able to perform core actions (registration, car booking, payment) within 3-5 steps.

Help tooltips and input validation will assist users throughout the system.

The system should be user-friendly even for users with minimal technical knowledge.

3.2.1.3 Efficiency

3.2.1.3.1 Performance Requirements

Search and filter responses must be returned within 2 seconds

The system must support up to 500 concurrent users without performance degradation

Page load time should not exceed 3 seconds under normal network conditions.

3.2.1.3.2 Space Requirements

Each user profile and booking should not exceed 500KB in the database

The database should efficiently store car data, images, invoices, and logs without exceeding hosting limits.

3.2.1.4 Dependability

The system should maintain 99.5% uptime

Automatic data backups must be performed daily

In case of failures, recovery mechanisms must restore the system within 10 minutes

3.2.1.5 Security

Passwords shall be hashed using modern encryption algorithms

Session must expire after 15 minutes of inactivity

The system must validate all inputs to prevent SQL injection and XSS attacks

Role-based access control will ensure only authorized users access specific functionalities.

3.2.2 Organizational Requirements

3.2.2.1 Environmental Requirements

The system will be hosted on a secure cloud infrastructure ensuring environmental stability and scalability

3.2.2.2 Operational Requirements

The system should be operable on Windows, macOS, Android, and iOS through modern browsers

Admins should receive email alerts for system downtime, booking conflicts, or suspicious activity

3.2.2.3 Development Requirements

The system should be operable on Windows, macOS, Android, and iOS through modern browsers

Admins should receive email alerts for system downtime, booking conflicts, or suspicious activity.

3.2.3 External Requirements

3.2.3.1 Regulatory Requirements

The system must comply with data protection laws such as GDPR

User consent must be obtained before collecting or storing personal data.

3.2.3.2 Ethical Requirements

User data shall not be shared with third parties without consent

The system should promote transparency by informing users of all terms and conditions clearly

3.2.3.3 Legislative requirements

3.2.3.3.1 Accounting Requirements

The system must automatically generate valid invoices containing tax, booking details, and total amount

All transactions must be logged and stored for audit purposes

3.2.3.3.2 Security Requirements

All communications between client and server must use HTTPS

Authentication must use secure login flows.

3.3 Domain requirements

The system must allow both regular users and admins to interact based on role-specific permissions

Cars must be categorized by availability, brand, fuel type, and pricing

Bookings must respect business rules.

Cancellations must comply with predefined policies

Integration with payment providers is required.

4. Software Design/ Diagrams

4.1 Requirements Analysis

4.1.1 User Scenarios**4.1.1.1 User Scenarios List**

<u>Scenario Code</u>	<u>User Scenario</u>	<u>Description</u>
US_1	User Registration	A new user registers an account by providing their name, email, phone number, and password. A confirmation email is sent to activate the account
US_2	User Login	A registered user logs into the system using email password. If credentials are incorrect, an error message is shown.
US_3	Password Reset	A user who forgets their password clicks "Forgot Password," enters their email, and receives a reset link.
US_4	Profile Update	A logged-in user updates their phone number, address, and profile picture through the settings page.
US_5	Searching for a car	A user filters cars based on brand, price, fuel type, and availability to find the best option.
US_6	Viewing Car Details	A user clicks on a car listing to view images, specifications, pricing, and user reviews.
US_7	Booking a Car	A user selects a car, enters pickup and return dates, chooses a payment method, and confirms the booking. If the car is available, the system reserves it.
US_8	Cancelling a booking	A user cancels a booking before the start date. The system verifies the cancellation policy and updates availability.
US_9	Admin Managing Users	An admin views the list of users, edits details, deactivates accounts, or deletes users if necessary.
US_10	Admin Updating Car Availability	An admin updates a car's status to available, booked, or under maintenance, ensuring the system prevents double booking.
US_11	Admin Adding Cars	An admin adds a new car by entering its brand, model, year, price per day, fuel type, and images. The car becomes available for booking.
US_12	Approving/Rejecting Bookings	An admin reviews booking requests and either approves or rejects them based on car availability and user history.
US_13	Making a Payment	A user completes a booking by selecting credit card, PayPal, or cash on pickup. The system processes the payment and generates an invoice.
US_14	Receiving Booking Confirmation	A user receives an email confirmation after successfully booking a car, including rental details and invoice.
US_15	Using Discount Codes	A user enters a promo code during payment, and the system validates it before applying a discount.
US_16	Submitting a Car Review	After returning a car, a user submits a star rating and text review. The system updates the car's review section.
US_17	Viewing Car Reviews	A user views previous customer reviews for a car before making a booking decision.
US_18	Admin Removing Reviews	If a review is flagged as inappropriate, an admin removes it from the system.
US_19	Scheduling Car Maintenance	An admin marks a car as under maintenance, preventing further bookings until it is available again.
US_20	Updating Maintenance Status	Once maintenance is completed, the admin updates the car's status back to available.
US_21	Logging User Activity	The system logs every user action (login, booking, payments, cancellations, reviews, etc.) for security and auditing.
US_22	Role-Based Access Control	Users can only access features based on their role (customers can book, admins can manage data, etc.).
US_23	Generating Invoices	After a successful booking and payment, the system automatically generates an invoice for the user.

US_24	Sending Notifications	The system sends email/SMS notifications for booking confirmations, cancellations, payment receipts, and reminders.
US_25	Tracking Rental History	A user can view their past bookings, payment history, and reviews from their dashboard.
US_26	Logging Out	A user securely logs out of the system, ending their session.
US_27	Admin Viewing User Activity Logs	Admins can check user activity logs to monitor login attempts, bookings, and suspicious behaviour.
US_28	Future Mobile App Integration	The system is designed to support future integration with a mobile application for a seamless user experience.

4.1.1.2 User scenarios extended

US_1: User registration

- The user navigates to the registration page
- The system displays a registration form
- The user fills in the required details
- The user submits the registration form
- The system validates the provided details
- The system creates a new user account
- The system sends a confirmation email to the user
- The user confirms their email by clicking the link
- The system activates the user account

US_2: User Login

- The user navigates to the login page.
- The system displays the login form.
- The user enters their email and password.
- The system verifies the credentials.
- The system grants access if credentials are correct.
- The user is redirected to the dashboard.

US_3: Password Reset:

1. The user navigates to the login page and selects the "Forgot Password" option.
2. The system prompts the user to enter their registered email.
3. The user enters their email and submits the request.
4. The system verifies if the email exists in the database.
5. If the email exists, the system generates a password reset link and sends it via email.
6. The user receives the email and clicks on the password reset link.
7. The system redirects the user to the password reset page.
8. The user enters a new password and confirms it.
9. The system validates the new password and updates it in the database.
10. The system notifies the user that the password reset was successful.

US_4: Profile Update

1. The user navigates to the profile settings page.
2. The system displays the user's current profile information.
3. The user selects an option to edit their profile details.
4. The user updates the desired information (e.g., name, phone number, address, profile picture).
5. The system validates the new information.

6. The system saves the updated profile details in the database.
7. The system notifies the user that the profile update was successful.

US_5: Searching for a Car

1. The user navigates to the car search page.
2. The system displays the available filters for brand, price, fuel type, and availability.
3. The user selects the desired filters (e.g., brand, price range, fuel type, availability).
4. The user clicks the search button to apply the filters.
5. The system processes the filters and displays the available cars matching the user's criteria.
6. The user reviews the results and selects a car for more details.

US_6: Viewing Car Details:

1. The user clicks on a car listing from the search results or car listing page.
2. The system displays the detailed view of the selected car.
3. The system shows the car's images, specifications (e.g., model, year, fuel type, mileage), pricing information, and user reviews.
4. The user reviews the details and may choose to make a reservation or return to the search results. The system activates the user account

US_7: Booking a Car

1. The user selects a car from the search results or car details page.
2. The system prompts the user to enter pickup and return dates.
3. The user selects the dates and proceeds to the payment method section.
4. The user selects a preferred payment method (e.g., credit card, PayPal).
5. The user confirms the booking details and submits the request.
6. The system checks the car's availability for the selected dates.
7. If the car is available, the system reserves it for the user and processes the payment.
8. The system sends a booking confirmation to the user with the details (pickup and return dates, car details, and payment receipt).

US_8: Cancelling a Booking

1. The user navigates to their booking details page.
2. The user selects the option to cancel the booking.
3. The system verifies that the booking is eligible for cancellation (before the start date).
4. The system checks the cancellation policy (e.g., refund eligibility, cancellation fees).
5. If the cancellation is successful, the system updates the car's availability for the reserved dates.
6. The system processes any refund if applicable based on the cancellation policy.
7. The system sends a cancellation confirmation to the user with the details of the cancellation and any refund information.

US_9: Admin Managing Users:

1. The admin navigates to the user management page.
2. The system displays a list of all registered users.
3. The admin selects a user to edit, deactivate, or delete.
4. The system updates the user information accordingly

UC_10: Admin Updating Car Availability

1. The admin navigates to the car management page.
2. The system displays a list of cars.
3. The admin selects a car and updates its status (available, booked, under maintenance).
4. The system updates the car's status accordingly.

UC_11: Admin Adding Cars

1. The admin navigates to the "Add Car" page.
2. The system displays a form for entering car details.
3. The admin enters details (brand, model, year, price per day, fuel type, images).
4. The admin submits the form.
5. The system validates and saves the new car.
6. The new car is marked as "available" in the system.

UC_12: Approving/Rejecting Bookings

1. The admin navigates to the "Booking Requests" page.
2. The system displays a list of pending bookings.
3. The admin selects a request and reviews the details.
4. The admin either approves or rejects the request.
5. The system updates the booking status accordingly.
6. The system notifies the user of the decision

UC_13: Making a payment

1. The customer selects a payment method (Credit Card, PayPal, or Cash on Pickup).
2. The system processes the payment.
3. If successful, the system generates an invoice.
4. The system updates the booking status as paid (except for cash on pickup).

UC_14: Receiving Booking Confirmation

1. The customer completes the booking process.
2. The system generates a booking confirmation.
3. The system emails the user with booking details and an invoice.

UC_15: Using Discount Codes

1. The customer enters a promo code during the payment process.
2. The system checks if the code is valid and not expired.
3. If valid, the discount is applied to the total amount.
4. The user proceeds with the discounted payment.

UC_16: Submitting a Car Review

1. The customer navigates to the car review section.
2. The customer selects a star rating and optionally writes a review.
3. The system validates the review.
4. The system updates the car's review section.

UC_17: Viewing Car Reviews

1. The user navigates to the car details page.
2. The system displays customer reviews for the selected car.
3. The user reads the reviews before making a booking decision.

UC_18: Admin Removing Reviews

1. The admin logs into the system.
2. The admin navigates to the flagged reviews section.
3. The admin selects an inappropriate review for removal.
4. The admin confirms the removal action.
5. The system deletes the review from the database

UC_19: Scheduling Car Maintenance

1. The admin logs into the system.
2. The admin navigates to the car management section.
3. The admin selects a car for maintenance.
4. The admin marks the car as "Under Maintenance."
5. The system updates the car status and prevents new bookings.

UC_20: Updating Maintenance Status

1. The admin logs into the system.
2. The admin navigates to the car management section.
3. The admin selects the car that completed maintenance.
4. The admin marks the car as "Available."
5. The system updates the car status and allows new bookings.

UC_21: Logging User Activity

1. The customer selects the payment option.
2. The system displays available payment methods.
3. The customer enters payment details and confirms.
4. The system processes the payment.
5. The system updates the rental record.
6. The system notifies the customer and admin.

UC_22: Role Based Access Control

1. The admin selects the report generation option.
2. The system displays report criteria options.
3. The admin selects the criteria and confirms.
4. The system generates the report.
5. The system displays or exports the report.

UC_23: Generating Invoices

1. The customer navigates to the review section.
2. The system displays a review submission form.
3. The customer enters and submits a review.
4. The system stores the review.
5. The admin can view, approve, or delete reviews.

UC_24: Sending Notifications

1. The customer accesses the dispute submission form.
2. The system displays the dispute submission interface.
3. The customer provides dispute details and submits them.
4. The system records the dispute.
5. The admin reviews the dispute and takes action.
6. The system notifies the customer of the resolution.

UC_25: Tracking Rental History

1. The customer logs into their account.
2. The system displays the user dashboard.
3. The customer clicks on "Rental History" section.
4. The system fetches the user's past booking data, payment history, and reviews.
5. The system displays the information in distinct sections (Bookings, Payments, Reviews).
6. The customer can scroll, filter, or sort the data as needed.

UC_26: Logging Out:

1. The user clicks on the "Log Out" button from the dashboard or the account settings menu.
2. The system prompts the user with a confirmation message: "Are you sure you want to log out?"
3. The user confirms the logout action.
4. The system securely logs the user out by invalidating the session and clearing any session data.
5. The user is redirected to the login screen or the homepage.

UC_27: Admin Viewing User Activity Logs

1. The admin navigates to the "Activity Logs" section of the admin dashboard.
2. The system displays a list of user activity logs.
3. The admin selects filters (e.g., user, date, activity type) to narrow down the log data.
4. The system displays the filtered logs, including timestamps, user actions, and statuses.
5. The admin clicks on specific entries to view detailed information (e.g., failed login attempts, suspicious bookings).

UC_28: Future Mobile App Integration

1. The user interacts with the web version of the system, creating a user account or logging in.
2. The system provides APIs to ensure mobile app data synchronization (e.g., booking history, reviews).
3. The user installs the mobile app (once developed) and logs in using the same credentials.
4. The mobile app communicates with the system's API to fetch the user's data in real-time.
5. The user has a consistent experience between the web and mobile platform.

▪ User Cases

UC Name	UC_1: User Registration
Summary	This use case describes the process of a new user registering on the car rental
Dependency	None
Actors	User, Admin
Preconditions	+ The user must have access to the registration page + The user must provide a valid email and password
Description of the main sequence	<ol style="list-style-type: none"> 1. The user navigates to the registration page 2. The system displays a registration form 3. The user fills in the required details 4. The user submits the registration form 5. The system validates the provided details 6. The system creates a new user account 7. The system sends a confirmation email to the user 8. The user confirms their email by clicking the link 9. The system activates the user account
Description of the Alternative Sequence	<ul style="list-style-type: none"> • If the user provides an already registered email, the system notifies the user and prompts them to log in or reset their password • If the user does not confirm the email within a certain period, the account remains the same
Non functional requirements	<ul style="list-style-type: none"> • The system must ensure data privacy by encrypting user passwords • The registration process must be completed within 5 seconds • The system should prevent duplicate accounts using the same email
Postconditions	<ul style="list-style-type: none"> • The user account is created and activated • The user can log in to the system.

UC Name	UC_2 : User Login
Summary	This use case describes the process of a registered user logging into the system.
Dependency	UC_1: User Registration
Actors	User, System
Preconditions	<ul style="list-style-type: none"> • The user must be registered in the system. • The user must have an active account.
Description of the main sequence	<ol style="list-style-type: none"> 1. The user navigates to the login page. 2. The system displays the login form. 3. The user enters their email and password. 4. The system verifies the credentials. 5. The system grants access if credentials are correct. 6. The user is redirected to the dashboard.
Description of the Alternative Sequence	<ul style="list-style-type: none"> • If the user enters incorrect credentials, the system displays an error message and allows retrying. • If the user forgets their password, they can reset it using the "Forgot Password" feature.
Non functional requirements	<ul style="list-style-type: none"> • The system must use encryption for password storage. • The login attempt should be processed within 3 seconds. • The system should lock an account after 5 failed login attempts.
Postconditions	<ul style="list-style-type: none"> • The user is logged into the system. • The session is securely maintained.

UC Name	UC 3: Password Reset
Summary	This use case describes how users can reset their passwords if they forget it
Dependency	UC 2: User Login
Actors	User, System
Preconditions	<ul style="list-style-type: none"> The user must have a registered account in the system The user must have access to their registered email
Description of the main sequence	<ol style="list-style-type: none"> The user navigates to the login page and selects the "Forgot Password" option. The system prompts the user to enter their registered email. The user enters their email and submits the request. The system verifies if the email exists in the database. If the email exists, the system generates a password reset link and sends it via email. The user receives the email and clicks on the password reset link. The system redirects the user to the password reset page. The user enters a new password and confirms it. The system validates the new password and updates it in the database. The system notifies the user that the password reset was successful.
Description of the Alternative Sequence	<ul style="list-style-type: none"> If the email does not exist, the system notifies the user. If the reset link expires, the system notifies the user and prompts them to request a new one. If the new password does not meet security requirements, the system prompts the user to enter a stronger password.
Non functional requirements	<ul style="list-style-type: none"> The reset process should complete within 1 minute The reset link should be valid for 15 minutes The system should use secure encryption to store password
Postconditions	<ul style="list-style-type: none"> The user's password is successfully updated The user can login using the new password

UC Name	UC 4: Profile Update
Summary	This use case describes how users can update their profile information
Dependency	UC 2: User Login
Actors	User, System
Preconditions	<ul style="list-style-type: none"> The user must be logged into the system The user must have an existing profile in the picture
Description of the main sequence	<ol style="list-style-type: none"> The user navigates to the profile settings page. The system displays the user's current profile information. The user selects an option to edit their profile details. The user updates the desired information (e.g., name, phone number, address, profile picture). The system validates the new information. The system saves the updated profile details in the database. The system notifies the user that the profile update was successful.
Description of the Alternative Sequence	<ul style="list-style-type: none"> If the user enters invalid data (e.g., incorrect email format), the system prompts them to correct it. If the update process fails due to a system error, the system notifies the user and suggests trying again later. If the user attempts to update restricted fields (e.g., username or email without verification), the system prevents the update and notifies the user.
Non functional requirements	<ul style="list-style-type: none"> The profile update should be reflected immediately.

CAR RENTAL Requirement Specification

	<ul style="list-style-type: none"> The system should store and retrieve data efficiently, with updates completing within 3 seconds. The system should provide real-time validation to prevent incorrect inputs.
Postconditions	<ul style="list-style-type: none"> The user's profile information is successfully updated The user can see the updated details in their profile

UC Name	UC_5: Searching for a car
Summary	This use case describes the process of a user searching for a car to rent using various filters to narrow down the best options
Dependency	None
Actors	User, Admin
Preconditions	+ The user must be logged into the system.
Description of the main sequence	<ol style="list-style-type: none"> The user navigates to the car search page. The system displays the available filters for brand, price, fuel type, and availability. The user selects the desired filters (e.g., brand, price range, fuel type, availability). The user clicks the search button to apply the filters. The system processes the filters and displays the available cars matching the user's criteria. The user reviews the results and selects a car for more details.
Description of the Alternative Sequence	<ul style="list-style-type: none"> If no cars match the user's filters, the system notifies the user and suggests adjusting the search criteria. If the user removes certain filters, the system updates the results accordingly.
Non functional requirements	<ul style="list-style-type: none"> The search process must be completed within 3 seconds for optimal performance. The system must ensure the accuracy of the car information displayed. The system should allow for easy navigation between search results.
Postconditions	<ul style="list-style-type: none"> The user sees a list of cars that match the selected filters. The user can proceed to view more details or make a reservation for a selected car.

UC Name	UC_6: Viewing car details
Summary	This use case describes the process of a user viewing the details of a specific car listing, including images, specifications, pricing, and user reviews.
Dependency	The car listing must exist and be available in the search results.
Actors	User, System
Preconditions	<ul style="list-style-type: none"> + The user must have access to the car search results or a specific car listing. + The user must be logged into the system (if required for viewing reviews or pricing).
Description of the main sequence	<ul style="list-style-type: none"> The user clicks on a car listing from the search results or car listing page. The system displays the detailed view of the selected car. The system shows the car's images, specifications (e.g., model, year, fuel type, mileage), pricing information, and user reviews. The user reviews the details and may choose to make a reservation or return to the search results. The system activates the user account
Description of the Alternative Sequence	<ul style="list-style-type: none"> If the car's details are not available (e.g., missing images or specifications), the system displays a notification informing the user of missing information.

CAR RENTAL Requirement Specification

	<ul style="list-style-type: none"> If no user reviews are available, the system shows a message indicating that no reviews have been submitted for this car.
Non functional requirements	<ul style="list-style-type: none"> The car details page must load within 3 seconds for a smooth user experience. The system must display high-quality images that are optimized for quick loading. The system should allow users to easily navigate back to the search results or make a reservation.
Postconditions	<ul style="list-style-type: none"> The user views the detailed car listing, including images, specifications, pricing, and reviews. The user can proceed with booking the car or returning to the search results.

UC Name	UC_7: Booking a car
Summary	<p>This use case describes the process of a user booking a car by selecting a car, entering pickup and return dates, choosing a payment method, and confirming the booking. If the car is available, the system reserves it.</p>
Dependency	The car must be available for the selected dates.
Actors	User, System, Payment Processor
Preconditions	<ul style="list-style-type: none"> + The user must be logged into the system. + The user must have access to the car details and booking page. + The car must be available for the selected dates.
Description of the main sequence	<ul style="list-style-type: none"> The user selects a car from the search results or car details page. The system prompts the user to enter pickup and return dates. The user selects the dates and proceeds to the payment method section. The user selects a preferred payment method (e.g., credit card, PayPal). The user confirms the booking details and submits the request. The system checks the car's availability for the selected dates. If the car is available, the system reserves it for the user and processes the payment. The system sends a booking confirmation to the user with the details (pickup and return dates, car details, and payment receipt).
Description of the Alternative Sequence	<ul style="list-style-type: none"> If the car is not available for the selected dates, the system notifies the user and suggests alternative cars or dates. If the payment fails, the system notifies the user and prompts them to select another payment method.
Non functional requirements	<ul style="list-style-type: none"> The booking process must be completed within 5 seconds after payment confirmation. The system should securely process payment information using encryption. The system must prevent double bookings for the same car and dates.
Postconditions	<ul style="list-style-type: none"> The car is successfully reserved for the selected pickup and return dates. The user receives a confirmation email with the booking details.

UC Name	UC 8: Cancelling a booking
Summary	<p>This use case describes the process of a user cancelling a booking before the start date. The system verifies the cancellation policy and updates car availability accordingly.</p>
Dependency	<p>The car must be reserved, and the cancellation request must be made before the start date of the booking.</p>
Actors	User, System
Preconditions	<ul style="list-style-type: none"> + The user must be logged into the system. + The user must have an active booking that is yet to start.
Description of the main sequence	<ul style="list-style-type: none"> • The user navigates to their booking details page. • The user selects the option to cancel the booking. • The system verifies that the booking is eligible for cancellation (before the start date). • The system checks the cancellation policy (e.g., refund eligibility, cancellation fees). • If the cancellation is successful, the system updates the car's availability for the reserved dates. • The system processes any refund if applicable based on the cancellation policy. • The system sends a cancellation confirmation to the user with the details of the cancellation and any refund information.
Description of the Alternative Sequence	<ul style="list-style-type: none"> • If the booking is not eligible for cancellation (e.g., it's after the cancellation window), the system notifies the user and provides information on the non-refundable status. • If the user cancels too late, the system may apply a cancellation fee or deny the cancellation based on the policy.
Non functional requirements	<ul style="list-style-type: none"> • The cancellation process must be completed within 3 seconds after submitting the request. • The system must securely process any refunds according to the cancellation policy. • The system should ensure the updated availability of the car is reflected immediately in the booking system.
Postconditions	<ul style="list-style-type: none"> • The booking is cancelled, and the car's availability is updated.

CAR RENTAL Requirement Specification

	<ul style="list-style-type: none">The user receives a cancellation confirmation email with booking details and any refund information.
--	--

UC Name	UC_9: Admin Managing Users
Summary	This use case describes how an admin can manage users by viewing, editing, deactivating, or deleting accounts
Dependency	None
Actors	Admin
Preconditions	<ul style="list-style-type: none">The admin must be logged into the system.The admin must have the necessary permissions to manage users
Description of the Main Sequence	<ol style="list-style-type: none">The admin navigates to the user management page.The system displays a list of all registered users.The admin selects a user to edit, deactivate, or delete.The system updates the user information accordingly
Description of the Alternative Sequence	<ul style="list-style-type: none">If the admin attempts to delete an already deactivated user, the system notifies them.If the admin tries to edit a non-existent user, the system shows an error message
Non-functional Requirements	<ul style="list-style-type: none">The system must update changes in real-time.User data should remain secure, with proper authorization checks.
Postconditions	<ul style="list-style-type: none">The user's account details are updated, deactivated, or deleted successfully.

UC Name	UC_10: Admin Updating Car Availability
Summary	This use case describes how an admin can update the availability status of cars
Dependency	None
Actors	Admin
Preconditions	<ul style="list-style-type: none">The admin must be logged in.The car must already exist in the system
Description of the Main Sequence	<ol style="list-style-type: none">The admin navigates to the car management page.The system displays a list of cars.The admin selects a car and updates its status (available, booked, under maintenance).The system updates the car's status accordingly.
Description of the Alternative Sequence	<ul style="list-style-type: none">If the car is already booked, the system prevents changing its status to "available" until the booking is completed

CAR RENTAL Requirement Specification

Non-functional Requirements	<ul style="list-style-type: none"> - The system must prevent double booking of cars. - The update should be reflected in real-time.
Postconditions	- The car's status is updated successfully

UC Name	UC_11: Admin Adding Cars
Summary	This use case describes how an admin can add a new car to the system.
Dependency	None
Actors	Admin
Preconditions	- The admin must be logged in.
Description of the Main Sequence	<ol style="list-style-type: none"> 1. The admin navigates to the "Add Car" page. 2. The system displays a form for entering car details. 3. The admin enters details (brand, model, year, price per day, fuel type, images). 4. The admin submits the form. 5. The system validates and saves the new car. 6. The new car is marked as "available" in the system.
Description of the Alternative Sequence	- If required fields are missing, the system displays an error message
Non-functional Requirements	<ul style="list-style-type: none"> - The system should prevent duplicate entries of the same car. - Image uploads should be optimized for performance
Postconditions	<ul style="list-style-type: none"> - The system should prevent duplicate entries of the same car. - Image uploads should be optimized for performance

UC Name	UC_12: Approving/Rejecting Bookings
Summary	This use case describes how an admin can approve or reject booking requests.
Dependency	None

CAR RENTAL Requirement Specification

Actors	Admin
Preconditions	<ul style="list-style-type: none"> - The admin must be logged in. - There must be pending booking requests
Description of the Main Sequence	<ol style="list-style-type: none"> 1. The admin navigates to the "Booking Requests" page. 2. The system displays a list of pending bookings. 3. The admin selects a request and reviews the details. 4. The admin either approves or rejects the request. 5. The system updates the booking status accordingly. 6. The system notifies the user of the decision
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If the car is no longer available, the system automatically rejects the booking.
Non-functional Requirements	<ul style="list-style-type: none"> - Booking approvals or rejections must be logged for tracking. - The system should ensure that bookings are processed within a few seconds
Postconditions	<ul style="list-style-type: none"> - The booking request is either approved or rejected. - The user is notified of the decision

UC Name	UC_13: Making a Payment
Summary	A user completes a booking by selecting credit card, PayPal, or cash on pickup. The system processes the payment and generates an invoice.
Dependency	UC_7: Booking a Car
Actors	User, Payment System
Preconditions	<ul style="list-style-type: none"> • The user must have an active booking. • The user must have a valid payment method.
Description of the main sequence	<ol style="list-style-type: none"> 5. The user selects a payment method (Credit Card, PayPal, or Cash on Pickup). 6. The system processes the payment. 7. If successful, the system generates an invoice. 8. The system updates the booking status as paid (except for cash on pickup).
Description of the Alternative Sequence	<ul style="list-style-type: none"> • If the payment fails, the system notifies the user and prompts them to retry or select another method. • If paying by cash, the system marks the booking as pending payment.

CAR RENTAL Requirement Specification

Non functional requirements	<ul style="list-style-type: none"> Payment processing should be secure and encrypted. The transaction should complete within 5 seconds.
Postconditions	<ul style="list-style-type: none"> The payment is processed successfully. The user receives a confirmation of payment.

UC Name	UC_14: Receiving Booking Confirmation
Summary	A user receives an email confirmation after successfully booking a car, including rental details and an invoice.
Dependency	UC_7: Booking a Car, UC_13: Making a Payment
Actors	User, System
Preconditions	<ul style="list-style-type: none"> The user must have successfully booked a car. A valid email must be provided.
Description of the main sequence	<ol style="list-style-type: none"> The user completes the booking process. The system generates a booking confirmation. The system emails the user with booking details and an invoice.
Description of the Alternative Sequence	<ul style="list-style-type: none"> If the email fails to send, the system retries or allows the user to request a resend.
Non functional requirements	<ul style="list-style-type: none"> The confirmation email should be sent within 1 minute of booking. The system must ensure email delivery reliability.
Postconditions	<ul style="list-style-type: none"> The user receives a booking confirmation. The booking details are stored in the system.

UC Name	UC_15: Using Discount Codes
Summary	A user enters a promo code during payment, and the system validates it before applying a discount.
Dependency	UC_13: Making a Payment
Actors	User, System
Preconditions	<ul style="list-style-type: none"> A valid promo code must exist in the system. The user must enter the code before finalizing the payment.
Description of the main sequence	<ol style="list-style-type: none"> The user enters a promo code during the payment process. The system checks if the code is valid and not expired. If valid, the discount is applied to the total amount. The user proceeds with the discounted payment.
Description of the Alternative Sequence	<ul style="list-style-type: none"> If the code is invalid or expired, the system notifies the user and does not apply a discount.
Non functional requirements	<ul style="list-style-type: none"> Promo code validation should occur instantly. The system must prevent the reuse of single-use promo codes.
Postconditions	<ul style="list-style-type: none"> The discount is applied successfully. The user completes the payment with the adjusted price.

UC Name	UC_16: Submitting a Car Review
Summary	After returning a car, a user submits a star rating and text review. The system updates the car's review section.
Dependency	UC_7: Booking Car
Actors	User, System
Preconditions	<ul style="list-style-type: none"> The user must have completed a car rental. The review must meet content guidelines.

CAR RENTAL Requirement Specification

Description of the main sequence	<ol style="list-style-type: none"> 5. The user navigates to the car review section. 6. The user selects a star rating and optionally writes a review. 7. The system validates the review. 8. The system updates the car's review section.
Description of the Alternative Sequence	<ul style="list-style-type: none"> • If the review contains prohibited content, the system rejects it and notifies the user.
Non functional requirements	<ul style="list-style-type: none"> • The review submission should be processed within 3 seconds. • The system must filter inappropriate content.
Postconditions	<ul style="list-style-type: none"> • The review is successfully submitted. • The car's review section is updated.

UC Name	UC_17: Viewing Car Reviews
Summary	This use case describes the process of a user viewing previous customer reviews for a car before making a booking decision.
Dependency	None
Actors	User
Preconditions	<ul style="list-style-type: none"> - The user must have access to the car rental platform. - There must be existing customer reviews for the car.
Description of the main sequence	<ol style="list-style-type: none"> 1. The user navigates to the car details page. 2. The system displays customer reviews for the selected car. 3. The user reads the reviews before making a booking decision.
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If no reviews are available for the selected car, the system displays a message indicating the absence of reviews.
Non-functional requirements	<ul style="list-style-type: none"> - The system should display reviews within 2 seconds. - Reviews should be sorted based on relevance or most recent.
Postconditions	<ul style="list-style-type: none"> - The user has viewed customer reviews for the car.

UC Name	UC_18: Admin Removing Reviews
Summary	This use case describes the process of an admin removing flagged inappropriate reviews from the system.
Dependency	None
Actors	Admin
Preconditions	<ul style="list-style-type: none"> - A review must be flagged as inappropriate. - The admin must have appropriate access permissions.
Description of the main sequence	<ol style="list-style-type: none"> 1. The admin logs into the system. 2. The admin navigates to the flagged reviews section. 3. The admin selects an inappropriate review for removal. 4. The admin confirms the removal action. 5. The system deletes the review from the database.
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If the admin does not confirm removal, the review remains in the system. - If the review is not flagged, it does not appear in the flagged reviews section.
Non-functional requirements	<ul style="list-style-type: none"> - The system should log all review removals for auditing purposes. - The review removal process should be completed within 3 seconds.
Postconditions	<ul style="list-style-type: none"> - The inappropriate review is removed from the system.

UC Name	UC_19: Scheduling Car Maintenance
Summary	This use case describes how an admin marks a car as under maintenance, preventing further bookings.
Dependency	None

CAR RENTAL Requirement Specification

Actors	Admin
Preconditions	<ul style="list-style-type: none"> - The admin must have access to the system. - The car must be listed in the system.
Description of the main sequence	<ol style="list-style-type: none"> 1. The admin logs into the system. 2. The admin navigates to the car management section. 3. The admin selects a car for maintenance. 4. The admin marks the car as "Under Maintenance." 5. The system updates the car status and prevents new bookings.
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If the car is already booked, the system prevents the status change until the booking is completed.
Non-functional requirements	<ul style="list-style-type: none"> - The system should reflect status changes in real-time. - The maintenance scheduling process should be completed within 3 seconds.
Postconditions	<ul style="list-style-type: none"> - The car status is updated to "Under Maintenance." - The car is unavailable for booking.

UC Name	UC_20: Updating Maintenance Status
Summary	This use case describes how an admin updates a car's status back to available after maintenance.
Dependency	None
Actors	Admin
Preconditions	<ul style="list-style-type: none"> - The car must already be marked as "Under Maintenance." - The admin must have access to the system.
Description of the main sequence	<ol style="list-style-type: none"> 1. The admin logs into the system. 2. The admin navigates to the car management section. 3. The admin selects the car that completed maintenance. 4. The admin marks the car as "Available." 5. The system updates the car status and allows new bookings.
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If the car is still undergoing maintenance, the system does not allow the status change.
Non-functional requirements	<ul style="list-style-type: none"> - The system should ensure real-time status updates. - The status update process should be completed within 3 seconds.

UC Name	UC 21: Manage Rental Payments
Summary	This use case allows customers to make rental payments, and the system processes and records them. The admin can view payment records.
Dependency	None
Actors	Customer, System, Admin
Preconditions	+ The costumer must have a valid rental agreement.
Description of the main sequence	<ul style="list-style-type: none"> • The customer selects the payment option. • The system displays available payment methods. • The customer enters payment details and confirms. • The system processes the payment. • The system updates the rental record. • The system notifies the customer and admin.
Postconditions	<ul style="list-style-type: none"> • The payment is recorded, and the rental status is updated.

CAR RENTAL Requirement Specification

UC Name	UC_22: Generate Rental Reports
Summary	This use case allows the admin to generate reports about rental transactions and revenue.
Dependency	None
Actors	Admin
Preconditions	+ The system must have stored rental data.
Description of the main sequence	<ul style="list-style-type: none"> • The admin selects the report generation option. • The system displays report criteria options. • The admin selects the criteria and confirms. • The system generates the report. • The system displays or exports the report.
Postconditions	<ul style="list-style-type: none"> • The report is generated and available for review.

UC Name	UC_23: Manage User Reviews
Summary	Customers can submit reviews about their rental experience, and the admin can manage these reviews.
Dependency	None
Actors	Customer, Admin
Preconditions	+ The customer must have completed at least one rental.
Description of the main sequence	<ul style="list-style-type: none"> • The customer navigates to the review section. • The system displays a review submission form. • The customer enters and submits a review. • The system stores the review. • The admin can view, approve, or delete reviews.
Postconditions	<ul style="list-style-type: none"> • The review is stored and available for users to view if approved.

UC Name	UC_24: Handle Rental Disputes
Summary	Customers can report rental-related disputes, and the admin can review and resolve them.
Dependency	None
Actors	Customer, Admin
Preconditions	+ The customer must have an active or completed rental agreement
Description of the main sequence	<ul style="list-style-type: none"> • The customer accesses the dispute submission form. • The system displays the dispute submission interface. • The customer provides dispute details and submits them. • The system records the dispute. • The admin reviews the dispute and takes action. • The system notifies the customer of the resolution.
Postconditions	<ul style="list-style-type: none"> • The dispute is reviewed and either resolved or escalated.

CAR RENTAL Requirement Specification

UC Name	UC_25: Tracking Rental History
Summary	This use case describes the process of a user viewing their past bookings, payment history, and reviews from their dashboard.
Dependency	<ul style="list-style-type: none"> - UC_2: User Login (User must be logged in to access their rental history.) - UC_7: Booking a Car (User must have made bookings in the past to view rental history.) - UC_13: Making a Payment (User needs payment history linked to their bookings.) - UC_16: Submitting a Car Review (User needs reviews to be part of their rental history.)
Actors	User, System
Preconditions	The user must be logged in. The system must have booking, payment, and review data stored for the user.
Description of the main sequence	<ol style="list-style-type: none"> 1. The user logs into their account. 2. The system displays the user dashboard. 3. The user clicks on "Rental History" section. 4. The system fetches the user's past booking data, payment history, and reviews. 5. The system displays the information in distinct sections (Bookings, Payments, Reviews). 6. The user can scroll, filter, or sort the data as needed.
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If no history is available, the system displays a message: "No history available." - If there is a delay in loading data, the system shows a loading spinner until the data is retrieved.
Non-functional requirements	<ul style="list-style-type: none"> - Data should be displayed in less than 2 seconds after request. - The system should support real-time updates (e.g., new booking data is visible immediately after confirmation).
Postconditions	The user has accessed and reviewed their rental history.

UC Name	UC_26: Logging Out
Summary	This use case describes the process of a user securely logging out of the system, ending their session.
Dependency	<ul style="list-style-type: none"> - UC_2: User Login (User must be logged in before logging out.) - UC_22: Role-Based Access Control (Role-based access ensures only authorized users can log out.)
Actors	User, System
Preconditions	The user must be logged into the system.
Description of the main sequence	<ol style="list-style-type: none"> 1. The user clicks on the "Log Out" button from the dashboard or the account settings menu. 2. The system prompts the user with a confirmation message: "Are you sure you want to log out?" 3. The user confirms the logout action. 4. The system securely logs the user out by invalidating the session and clearing any session data. 5. The user is redirected to the login screen or the homepage.
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If the user cancels the logout, the system returns them to the dashboard without logging out. - If the session has expired (e.g., after inactivity), the system automatically logs the user out and redirects them to the login page.
Non-functional requirements	<ul style="list-style-type: none"> - Logout should occur within 3 seconds. - Session data should be securely cleared to prevent unauthorized access.

CAR RENTAL Requirement Specification

Postconditions	The user is logged out of the system. The session is terminated, and the user is redirected to the login page.
-----------------------	--

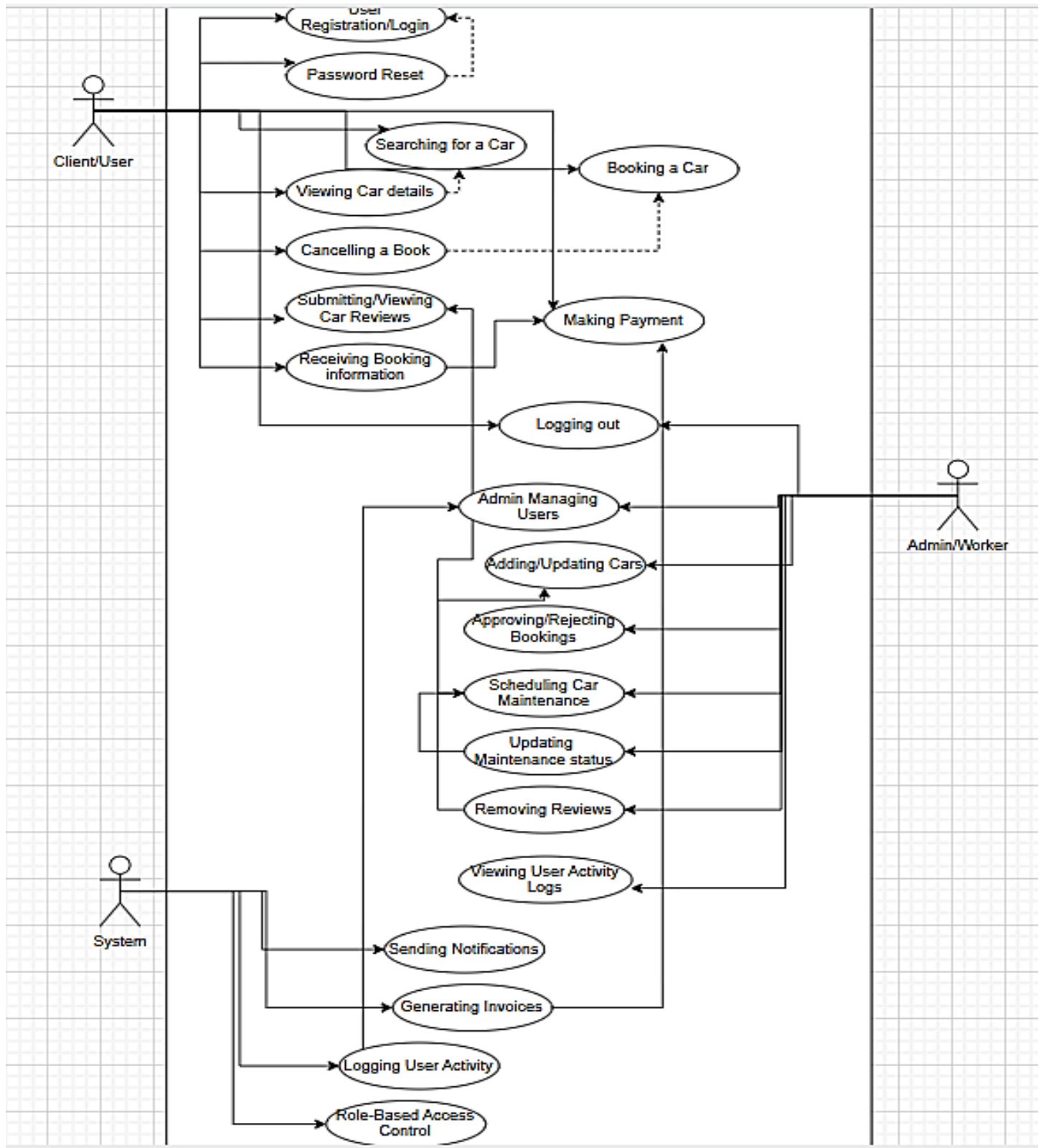
UC Name	UC_27: Admin Viewing User Activity Logs
Summary	This use case describes the process of an admin viewing user activity logs to monitor login attempts, bookings, and suspicious behavior.
Dependency	<ul style="list-style-type: none"> - UC_9: Admin Managing Users (Admin needs access to user information to monitor their activity.) - UC_21: Logging User Activity (User activities need to be logged for admins to view them.) - UC_22: Role-Based Access Control (Admins need proper permissions to view activity logs.)
Actors	Admin, System
Preconditions	The admin must be logged in and have the necessary permissions to view activity logs.
Description of the main sequence	<ol style="list-style-type: none"> 1. The admin navigates to the "Activity Logs" section of the admin dashboard. 2. The system displays a list of user activity logs. 3. The admin selects filters (e.g., user, date, activity type) to narrow down the log data. 4. The system displays the filtered logs, including timestamps, user actions, and statuses. 5. The admin clicks on specific entries to view detailed information (e.g., failed login attempts, suspicious bookings).
Description of the Alternative Sequence	<ul style="list-style-type: none"> - If no logs match the filter criteria, the system displays a message: "No logs found." - If an admin attempts to access logs without proper permissions, the system displays an error message.
Non-functional requirements	<ul style="list-style-type: none"> - Log retrieval should be completed in under 5 seconds. - The system should allow for the management of large log datasets (e.g., pagination or infinite scroll).
Postconditions	The admin has successfully reviewed user activity logs.

UC Name	UC_28: Future Mobile App Integration
Summary	This use case describes the design of the system to support future mobile app integration, ensuring a seamless user experience across platforms.
Dependency	<ul style="list-style-type: none"> - UC_1: User Registration (Mobile app will need to interact with the registration process.) - UC_2: User Login (The mobile app will need to support login and session management.) - UC_5: Searching for a Car (The mobile app will need to offer car search functionality.) - UC_7: Booking a Car (Mobile app will need booking functionality.) - UC_13: Making a Payment (Mobile app will need payment processing.) - UC_24: Sending Notifications (The mobile app will need to receive notifications.)
Actors	User, System
Preconditions	The system should be built with an API-first approach that supports mobile interactions.
Description of the main sequence	<ol style="list-style-type: none"> 1. The user interacts with the web version of the system, creating a user account or logging in. 2. The system provides APIs to ensure mobile app data synchronization (e.g., booking history, reviews).

CAR RENTAL Requirement Specification

	<ul style="list-style-type: none">3. The user installs the mobile app (once developed) and logs in using the same credentials.4. The mobile app communicates with the system's API to fetch the user's data in real-time.5. The user has a consistent experience between the web and mobile platform.
Description of the Alternative Sequence	<ul style="list-style-type: none">- If the mobile app is not yet available, the system displays an informational message indicating that mobile integration is coming soon.
Non-functional requirements	<ul style="list-style-type: none">- API response times should be under 3 seconds.- The system must be able to support simultaneous access from both web and mobile apps without conflicts.
Postconditions	The system supports seamless mobile app integration, providing users with a consistent experience across web and mobile platforms.

4.2 User Case Diagrams



CAR RENTAL Requirement Specification

