Software Requirements Specification (SRS) Document for Car Rental System

Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the Car Rental System. It outlines the system's functional and non-functional requirements, design constraints, and other essential details for development and implementation. The primary goal is to establish a clear understanding between the stakeholders and the development team, ensuring that the final product meets the intended requirements and expectations.

1.2 Scope

This SRS covers the Car Rental System intended to manage the rental process of vehicles for a car rental company. The system includes functionalities for vehicle search, booking, payment processing, user management, and administrative tasks. Additionally, the system will manage rental contracts, vehicle maintenance, expenses, and customer interactions.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- GUI: Graphical User Interface
- API: Application Programming Interface
- GDPR: General Data Protection Regulation

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Project Charter for Car Rental System

1.5 Problem Statement

The purpose of the assignment is to develop a requirements analysis document for a car rental management system. The manual car rental system was fraught with difficulties, including database management, which was a manual process. The goal is to develop a system that keeps detailed records of both the cars and the customers, the duration they rent the car, as well as the type of car they rent. The system will be mainly designed for a small company that renders car rental services to customers. The system will have the ability to generate and print invoices for each successful transaction.

The Vehicle Rental Management System (VRMS) aims to streamline and modernize the entire rental process, from vehicle availability and booking to vehicle return and billing. This system will not only improve the agency's operational efficiency but also enhance the customer experience by providing a user-friendly online platform. Key issues that the VRMS seeks to address include:

- **Inefficient Reservation Process**: The current process for making vehicle reservations is cumbersome, involving paper forms and manual record-keeping. This leads to errors and delays in confirming reservations.
- **Limited Vehicle Visibility**: Customers struggle to access real-time information about vehicle availability, types, and pricing. This lack of transparency leads to customer frustration and lost business opportunities.
- **Inaccurate Billing**: The manual billing process often results in errors, overcharges, or undercharges. This not only affects revenue but also damages the agency's reputation.
- **Manual Inventory Management**: Managing a large fleet of vehicles manually is prone to errors, making it difficult to track vehicle maintenance, service schedules, and retirements.
- **Customer Engagement**: Customer interactions and feedback collection are inefficient, making it challenging to address customer concerns and improve services.

1.6 Proper Template Usage

To use the waterfall model in a vehicle rental management system, the following steps can be taken:

- **Requirements Gathering**: Gather all the requirements for the system from stakeholders, including the features and functionalities that the system should have.
- **System Design**: Design the system architecture, including the database schema, user interface, and system modules.
- **Implementation**: Develop the system modules and integrate them into the system.
- **Testing**: Test the system to ensure that it meets the requirements and is free of bugs and errors.
- **Deployment**: Deploy the system to the production environment.

- **Maintenance**: Provide ongoing maintenance and support for the system.

The waterfall model is a linear and strict set of stages during the development process. It follows a particular process model and can be used to develop a car rental system. The iterative waterfall model is another development methodology that can be used in the development of a car rental management system. The system architecture can be designed using the SDLC framework, which describes all activities and processes in a software development project. The proposed system can be an online system that is fully integrated and automates manual procedures.

1.7 Overview

The document is structured to provide a comprehensive understanding of the system requirements, including overall description, specific requirements, external interfaces, and other non-functional requirements.

2. Overall Description

2.1 Product Perspective

The Car Rental System is a web-based application designed to automate the process of renting vehicles. It is intended to integrate with existing systems for payment processing and customer support.

2.2 Product Functions

- User registration and login
- Vehicle search and availability check
- Booking and reservation management
- Payment processing
- User profile management
- Administrative vehicle management
- Reporting and analytics

2.3 User Classes and Characteristics

- **Customers**: Individuals who wish to rent vehicles.
- **Administrators**: Staff who manage vehicle listings and reservations.
- **Support Staff**: Personnel responsible for customer support and feedback.

2.4 Operating Environment

The system will operate in a web environment accessible via modern web browsers. The backend will be hosted on cloud infrastructure ensuring scalability and reliability.

2.5 Design and Implementation Constraints

- Compliance with data protection regulations such as GDPR.
- Integration with third-party payment gateways.
- Support for multi-language and multi-currency.

2.6 Assumptions and Dependencies

- Users will have access to the internet.
- Integration with existing customer databases is feasible.
- The cloud service provider will meet uptime requirements.

3. Specific Requirements

1. User Registration and Authentication: Customers and administrators should be able to register with valid information. Users must authenticate their identities through secure login mechanisms.
2. Vehicle Management: Administrators should be able to add new vehicles, including details such as make, model, year, rental rate, and availability status. Administrators should be able to update and remove vehicles from the system. Vehicles should be categorized by type (e.g., cars, trucks) for easy search and selection.
3. Reservation System: Customers should be able to search for available vehicles based on criteria such as location, date, time, and vehicle type. Customers should be able to make reservations by specifying pickup and return dates and times. Customers should have the option to view, modify, or cancel their reservations. Administrators should be able to view and manage reservations, including approving or rejecting requests.
4. Rental and Billing: Customers should be able to rent vehicles based on approved reservations. The system should track rental periods accurately, calculate rental fees, and apply for discounts or promotions as applicable. Customers should be able to make payments securely through various methods (credit cards, online wallets, cash). Invoices should be generated, and customers should receive digital receipts.
5. Customer Support: Customers should have access to customer support features for inquiries, assistance, and issue resolution. Administrators should be able to respond to customer inquiries and resolve issues efficiently.
6. Reporting and Analytics: The system should generate reports on revenue, vehicle utilization, customer activity, and other relevant metrics. Analytics features should allow administrators to make data-driven decisions regarding fleet management and pricing strategies.

3.1 Functional Requirements

3.1.1 User Registration and Authentication

- **Requirement**: The system shall allow users to register with their email and create a password.
- **Requirement**: The system shall send a verification email to new users.
- **Requirement**: The system shall allow users to log in using their email and password.

3.1.2 Vehicle Search and Availability

- **Requirement**: The system shall allow users to search for available vehicles based on date, location, and vehicle type.

- **Requirement**: The system shall display vehicle details including make, model, and rental price.

3.1.3 Booking and Reservations

- **Requirement**: The system shall allow users to book a vehicle by selecting an available date and time.
- **Requirement**: The system shall generate a reservation confirmation and send it to the user's email.

3.1.4 Payment Processing

- **Requirement**: The system shall integrate with a payment gateway to process credit card payments.
- **Requirement**: The system shall provide a secure payment interface that complies with PCI-DSS standards.

3.1.5 User Profile Management

- **Requirement**: The system shall allow users to update their personal information.
- **Requirement**: The system shall display the user's booking history.

3.1.6 Vehicle Management

- **Requirement**: The system shall allow administrators to add, update, and remove vehicle listings.
- **Requirement**: The system shall track vehicle maintenance schedules and status.

3.1.7 Rental Agreement Management

- **Requirement**: The system shall generate rental agreements for each booking.
- **Requirement**: The system shall store signed rental agreements for future reference.

3.1.8 Reporting and Analytics

- **Requirement**: The system shall generate reports on rental activity, revenue, and customer demographics.
- **Requirement**: The system shall provide data visualization tools for trend analysis.

3.1.9 Notifications and Alerts

- **Requirement**: The system shall send email and SMS notifications for booking confirmations and reminders.
- **Requirement**: The system shall alert administrators of pending vehicle maintenance.

3.1.10 Customer Support and Feedback

- **Requirement**: The system shall provide a form for users to submit feedback and support requests.
- **Requirement**: The system shall track and manage customer support tickets.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- **Requirement**: The system shall handle up to 10,000 concurrent users.
- **Requirement**: The system shall respond to user actions within 2 seconds.

3.2.2 Security Requirements

- **Requirement**: The system shall encrypt all sensitive user data.
- **Requirement**: The system shall implement role-based access control for administrative functions.

3.2.3 Usability Requirements

- **Requirement**: The system shall provide an intuitive and user-friendly interface.
- **Requirement**: The system shall be accessible on both desktop and mobile devices.

3.2.4 Reliability Requirements

- **Requirement**: The system shall have an uptime of 99.9%.
- **Requirement**: The system shall provide data backup and recovery mechanisms.

3.2.5 Maintainability Requirements

- **Requirement**: The system shall use modular architecture to facilitate updates and maintenance.
- **Requirement**: The system shall provide comprehensive documentation for developers and users.

3.2.6 Portability Requirements

- **Requirement**: The system shall be deployable on various cloud platforms.
- **Requirement**: The system shall support multiple languages and currencies.

3.3 Interface Requirements

3.3.1 User Interfaces

- **Requirement**: The system shall provide a web-based graphical user interface for customers and administrators.

- **Requirement**: The system shall ensure the interface is responsive and compatible with various screen sizes.

3.3.2 Hardware Interfaces

- **Requirement**: The system shall support integration with barcode scanners for vehicle check-in and check-out.
- **Requirement**: The system shall integrate with GPS devices for real-time vehicle tracking.

3.3.3 Software Interfaces

- **Requirement**: The system shall provide APIs for integration with third-party services like payment gateways and customer support systems.
- **Requirement**: The system shall support integration with existing ERP systems used by the car rental company.

3.3.4 Communication Interfaces

- **Requirement**: The system shall use HTTPS for secure communication between the client and server.
- **Requirement**: The system shall support email and SMS gateways for notifications.

3.4 System Features

3.4.1 Feature 1: Vehicle Search

- **Description**: Users can search for available vehicles based on various criteria such as date, location, and type.
- **Priority**: High
- **Stimulus/Response Sequences**: User inputs search criteria -> System displays matching vehicles.

3.4.2 Feature 2: Booking Management

- **Description**: Users can book a vehicle, view their booking history, and manage their reservations.
- **Priority**: High
- **Stimulus/Response Sequences**: User selects vehicle and date -> System processes booking -> System sends confirmation email.

3.4.3 Feature 3: Payment Gateway Integration

- **Description**: The system integrates with payment gateways to process transactions securely.
- **Priority**: High

- **Stimulus/Response Sequences**: User enters payment details -> System processes payment -> System confirms payment.

3.4.4 Feature 4: Customer Feedback System

- **Description**: Users can submit feedback and support requests, and administrators can manage these requests.
- **Priority**: Medium
- **Stimulus/Response Sequences**: User submits feedback -> System logs feedback -> Administrator responds to feedback.

4. External Interface Requirements

4.1 User Interfaces

- **Requirement**: The system shall provide a dashboard for administrators to manage vehicles and bookings.
- **Requirement**: The system shall provide a user-friendly interface for customers to search and book vehicles.

4.2 Hardware Interfaces

- **Requirement**: The system shall integrate with barcode scanners for efficient vehicle check-in and check-out processes.
- **Requirement**: The system shall interface with GPS devices to provide real-time tracking of rental vehicles.

4.3 Software Interfaces

- **Requirement**: The system shall provide RESTful APIs for integration with third-party payment processors.
- **Requirement**: The system shall integrate with customer relationship management (CRM) software to manage customer data.

4.4 Communication Interfaces

- **Requirement**: The system shall use secure communication protocols (HTTPS) to protect data in transit.
- **Requirement**: The system shall support email and SMS notifications to keep users informed about their bookings and payments.

5. System Features

5.1 Detailed Description of Each System Feature

For each system feature, provide a detailed description, priority, dependencies, and related use cases.

- **Feature 1: Vehicle Search**
    - **Description**: Allows users to search for available vehicles.
    - **Priority**: High
    - **Dependencies**: Requires a comprehensive vehicle database.
    - **Use Cases**: UC-01: Search for vehicles, UC-02: Filter search results.
- **Feature 2: Booking Management**
    - **Description**: Manages user bookings and reservations.
    - **Priority**: High
    - **Dependencies**: Requires integration with user authentication and payment systems.
    - **Use Cases**: UC-03: Make a booking, UC-04: View booking history.
- **Feature 3: Payment Gateway Integration**
    - **Description**: Handles secure payment processing.
    - **Priority**: High
    - **Dependencies**: Requires integration with payment gateways.
    - **Use Cases**: UC-05: Process payment, UC-06: Refund payment.
- **Feature 4: Customer Feedback System**
    - **Description**: Collects and manages customer feedback.
    - **Priority**: Medium
    - **Dependencies**: Requires a feedback management module.
    - **Use Cases**: UC-07: Submit feedback, UC-08: Respond to feedback.

6. Other Nonfunctional Requirements

1. Performance: The system should be capable of handling a minimum of 500 concurrent users without significant performance degradation. Response times for common tasks (e.g., searching for vehicles, and making reservations) should be under 2 seconds.
2. Security: User data, including personal information and payment details, must be securely stored and transmitted using encryption. Access control mechanisms must restrict access to sensitive features and data based on user roles (e.g., customer, administrator). The system should be protected against common security threats, including SQL injection, cross-site scripting (XSS), and data breaches.
3. Reliability: The system should aim for a minimum uptime of 99.9%, with scheduled maintenance windows communicated to users in advance. Backup and disaster recovery procedures should be in place to ensure data integrity and system availability.
4. Scalability: The system should be designed to scale horizontally or vertically to accommodate an expanding fleet of vehicles and increasing user load during peak periods.
5. Usability: The user interface should be intuitive, responsive, and accessible on various devices (desktop and mobile). Error messages should be clear and user-friendly to assist users in resolving issues.

6.1 Performance Requirements

- **Requirement**: The system shall support up to 10,000 concurrent users with a response time of less than 2 seconds for most interactions.

6.2 Safety Requirements

- **Requirement**: The system shall ensure all personal and payment data is encrypted and stored securely to protect against data breaches.

6.3 Security Requirements

- **Requirement**: The system shall implement robust authentication mechanisms to prevent unauthorized access.

6.4 Software Quality Attributes

- **Requirement**: The system shall be designed to be scalable, maintainable, and reliable, with an emphasis on usability and performance.

6.5 Business Rules

- **Requirement**: The system shall adhere to the business rules defined by the car rental company, such as pricing models, rental periods, and late fees.

7. Other Requirements

7.1 Legal and Regulatory Requirements

- **Requirement**: The system shall comply with GDPR and other relevant data protection regulations.

7.2 Compliance Requirements

- **Requirement**: The system shall adhere to PCI-DSS standards for payment processing.

Appendices

A. Glossary

- **Term**: Definition
- **Customer**: A person who uses the car rental service.
- **Administrator**: A person who manages the system's backend functions.

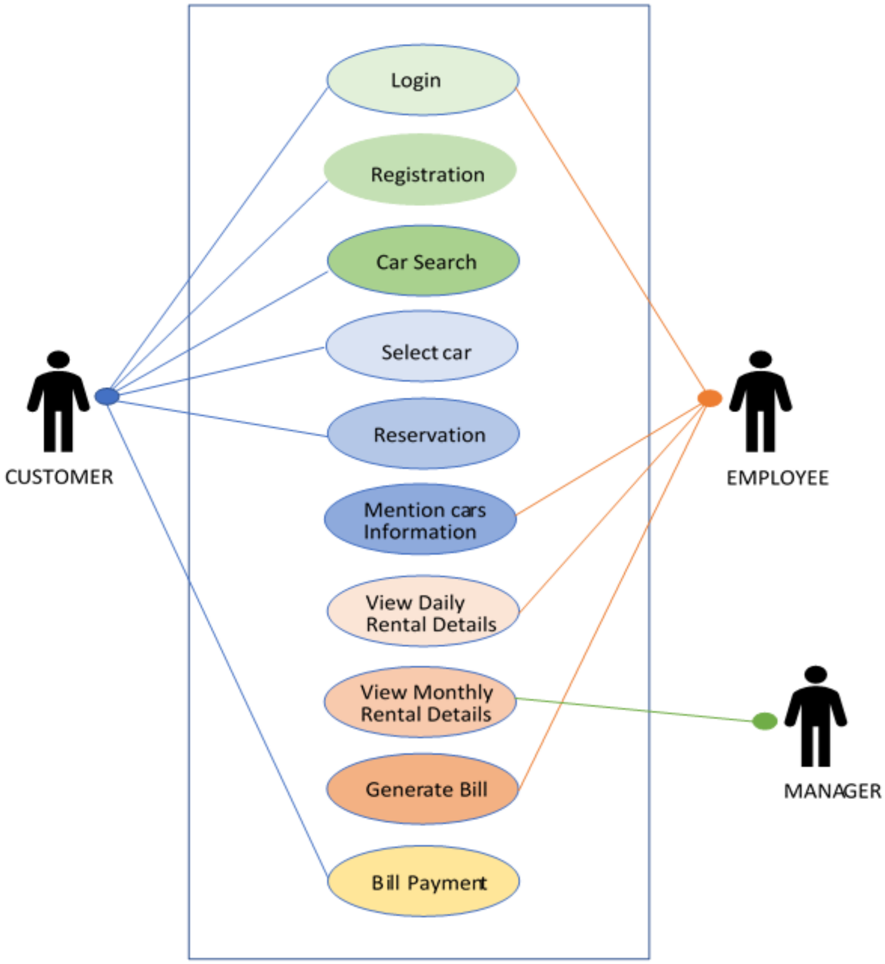
B. Analysis Models

- **Data Flow Diagrams**: Illustrate the flow of data within the system.
- **Use Case Diagrams**: Represent the interactions between users and the system.

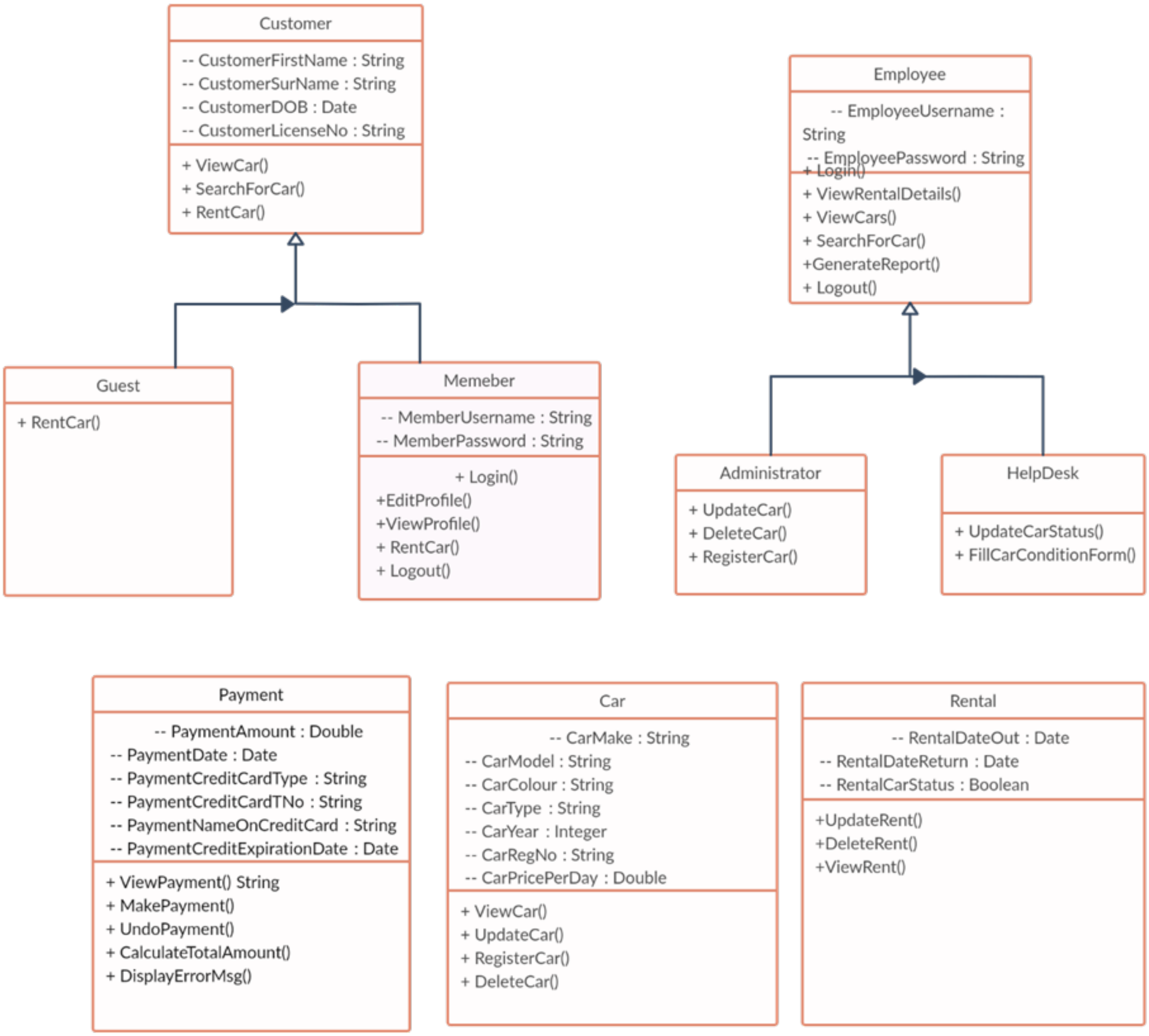
C. Issues List

- **Issue**: Description, Status, Resolution
- **Data Integration**: Challenges with integrating existing databases, In Progress, Expected completion by next release.

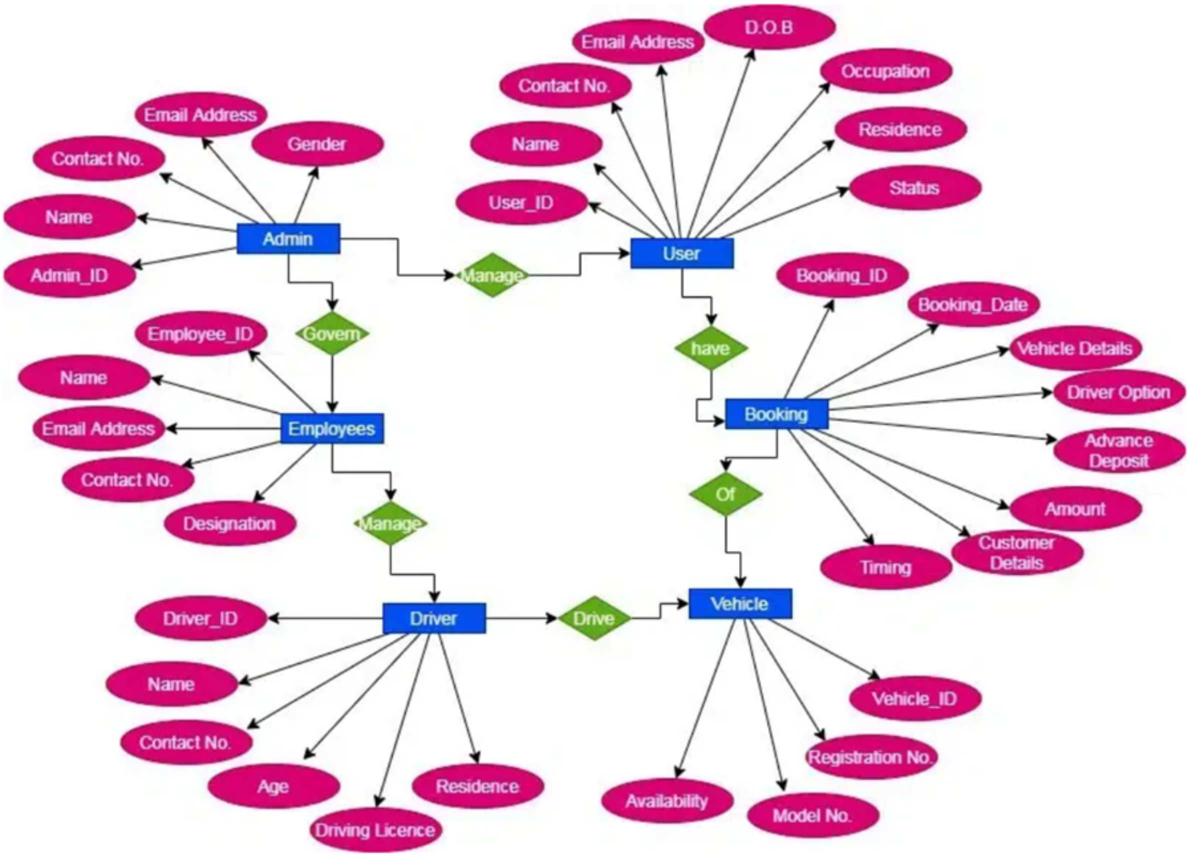
Use case diagram



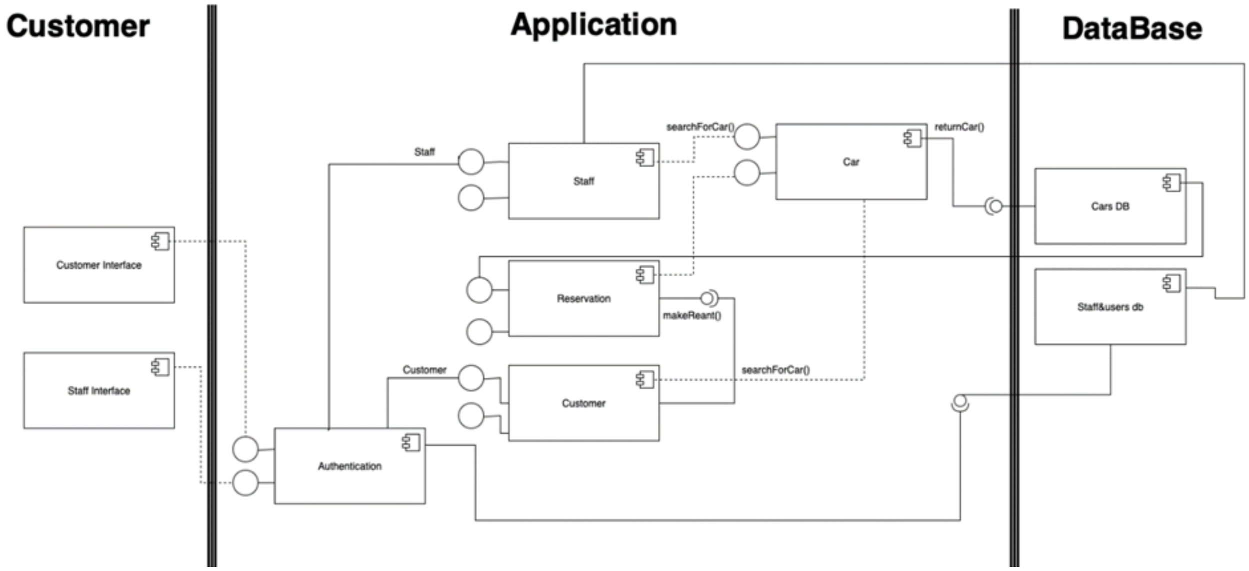
Class Diagram



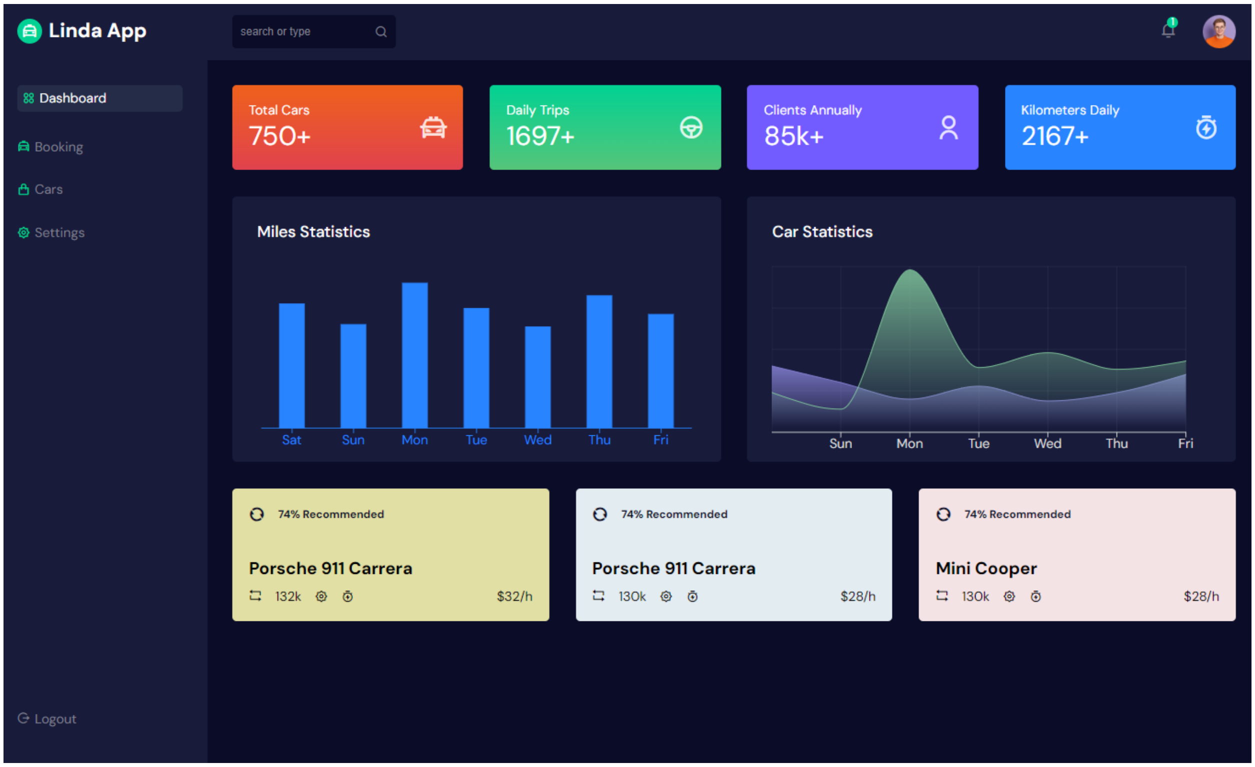
ER Diagram



Deployment Diagram



App Dashboard and UI



Conclusion

The Car Rental System outlined in this SRS document is designed to provide a comprehensive solution for managing car rentals, rental contracts, vehicle maintenance, expenses, and customer relationships. The system aims to improve operational efficiency, enhance customer satisfaction, and ensure seamless integration with existing systems and processes.

Key Features

- **RentContracts**: Efficient management of all rental contracts, with detailed reporting on active, pending, and completed contracts. The system also includes specialized reports such as Car Reports, Renter Reports, Debtor/Creditor Reports, and Traffic Violations management.
- **Cars**: Robust management of the car fleet, including scheduling and tracking maintenance activities.
- **Renters**: Comprehensive renter management, including maintaining a blacklist for problematic renters.
- **Expenses**: Detailed tracking and reporting of all expenses related to car rentals, garage services, traders, and beneficiaries.
- **Traders**: Effective management of traders and suppliers, ensuring timely procurement and service delivery.
- **Tracking**: Real-time tracking of rented vehicles to enhance security and operational efficiency.
- **CompaniesMessages**: Streamlined communication with associated companies and stakeholders to ensure smooth operations and address any issues promptly.

Future Enhancements

As the Car Rental System evolves, future enhancements may include the integration of advanced analytics for predictive maintenance, enhanced customer relationship management features, and expanded capabilities for mobile access and remote management. Continuous feedback from users and stakeholders will drive these improvements, ensuring the system remains aligned with the evolving needs of the car rental industry.

This SRS document serves as the foundation for the development and implementation of the Car Rental System. By adhering to the outlined requirements and specifications, the development team can ensure the delivery of a robust, reliable, and user-friendly solution that meets the needs of all stakeholders.