**Software Requirement Specification (SRS) of Royal Rent**

**1. Introduction:**

The car rental management system is a software solution designed to streamline the process of renting vehicles to customers. It typically includes features such as vehicle inventory management, customer management, reservation handling, billing, and reporting. SQL would be used to create and manage the database schema.

**2. Objectives:**

The objectives of a car rental management system include:

**1. User Management**: Implementing user authentication and authorization to control access to the system.

**2. Vehicle Management**: Tracking details of available vehicles, including make, model, year and mileage.

**3. Reservation Management**: Allowing customers to book vehicles for specific dates and times, with features like availability checking.

**4. Billing and Payment**: Generating invoices for rentals, calculating charges based on rental duration.

**3. Functional Requirements:**

**1. User Authentication and Authorization**: The system shall allow users to register and log in with unique credentials. Unauthorized users shall not be able to access any functionalities of the system.

**2. Vehicle Management:** Admin users shall be able to add, update, and delete vehicle information, including make, model, year and registration number.

**3. Reservation Management:** Customers shall be able to search for available vehicles based on criteria such as vehicle type. Customers shall be able to make reservations by selecting a vehicle and providing pickup and drop-off dates.

**4. Customer Management:** The system shall maintain customer profiles, including personal information. Admin and staff users shall be able to view and update customer information.

**5. Billing and Invoicing:** The system shall generate invoices for each reservation, including details such as rental fees and taxes. Customers shall be able to view invoices from their accounts.

**6. Reporting:** The system shall provide reporting functionalities to generate various types of reports, including reservation and inventory status.

**4. Non-functional Requirements:**

**1. Security:** User authentication and authorization mechanisms shall be implemented securely to prevent unauthorized access.

**2. Scalability:** The system architecture shall be designed to scale horizontally to accommodate future growth in the number of users and vehicles.

**5. Interfaces:**

The Royal Rent Car Rental Management System is a comprehensive solution designed to streamline the operations of Royal Rent Company. Developed using C# for the frontend and MySQL for the backend, this system offers an intuitive user interface and database management capabilities.

**1. Login Interface:** Allow users to log in with their credentials securely.

**2. Manage Cars Interface**: Enable adding, editing, and deleting cars, with details like make, model and year.

**3. Manage Customers Interface:** Allow management of customer information including adding, editing, and deleting customer profiles.

**4. Reservation Interface**: Allow users to make new reservations, view existing reservations, and manage reservation details.

**5. Rental Interface**: Displaying terms and conditions, rental duration and charges.

**6. Payment Interface:** Handle payment processing for reservations, including capturing payment details.

**7. Reports Interface**: Generate reports such as revenue reports and booking summaries.

**6. Assumptions and Constraints:**

**Assumptions:**

* The system will handle rental pricing, including daily rates.
* The system will generate invoices and receipts for rentals.
* The system will provide reporting and analytics for business insights.
* There will be authentication and authorization mechanisms to ensure data security and privacy.

**Constraints:**

* Scalability to handle a growing number of users.
* User interface constraints to ensure ease of use for customers.
* Documentation constraints to onboard users effectively.

**7. Appendices:**

**1. Database Schema**: Provide a detailed schema of the MySQL database, including tables, columns, data types, and relationships.

**2. Sample Reports**: Examples of reports generated by the system, such as rental transaction reports and vehicle availability reports.

**3. Deployment Instructions**: Step-by-step instructions for deploying the application, including setting up the database, configuring the application settings, and running the system.

**4. Glossary:** Definitions of technical terms and acronyms used throughout the documentation to help users understand the system better.