Software Requirement Specification

SOFTWARE DEVELOPMENT LIFE CYCLE

Car Rental System

Prepared By :

Deepasree Meena Padmanabhan, Pujan Bhuva,

Simran & OgheneRukevwe Esegba

Algoma UNIVERSITY

Department of Computer Science

Software engineering (COSC3506 001XE)

Table of Contents

Revision History

Version	Date	Author	Description
0.0a	1/25/2025	Deepasree Meena Padmanabhan	Initial draft creation.
0.1b	1/26/2025	Pujan Bhuva	Added functional specifications and core system features.
0.1c	1/28/2025	Simran	Included non-functional specifications such as security, performance, and usability.
0.1d	1/30/2025	OgheneRukevwe Esegba	Integrated UI enhancements and improved user experience elements.
0.1e	2/5/2025	Deepasree Meena Padmanabhan	Added system diagrams illustrating architecture, workflows, and relationships.
0.1f	2/6/2025	Simran	Documented revision history for tracking version updates and modifications.
0.1g	2/7/2025	Pujan Bhuva	Finalized and signed off the document for approval and completion.

Software Requirement Specification Sign-Off

Role	Name	Date
Business Analyst (BSA)	Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba	2/6/2025
IT Project Manager	Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba	2/6/2025
BU Project Manager	Amandeep S Patti	2/6/2025
Solution Architect	Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba	2/6/2025
Developer Lead	Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba	2/6/2025
Quality Engineering Lead	Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba	2/6/2025

Introduction

The Car Rental System will ease the leasing of cars and is more effective and user-friendly. It would grant car-renting agencies opportunities to manage cars, bookings, and customer data by automating these operations. There are three major kinds of users that can be included within the system: Administrator, Staff, and Customer. The administrators do the inventory management, user management, and pricing; staff handle rentals and maintenance; and customers can browse, book, and reserve cars. This system reduces paperwork, saves time, and makes the experience of renting quite smooth.

Purpose

The main aim of this project is to design an automated, user-friendly Car Rental System that will help the rental agencies in managing their fleets and customer bookings effectively. It will reduce the manual workload while improving operational efficiency by incorporating current object-oriented programming and database-driven architecture.

This SRS targets developers, project managers, QA testers, and other stakeholders who will participate in designing, implementing, and testing the Car Rental System. It will also be useful to maintenance teams that may update or enhance the system in the future.

The major focus areas for this project include:

- Vehicle inventory management
- Automating booking and return processing
- Role-based user access and authentication
- Keeping track of vehicle conditions and schedules for maintenance
- Comprehensive reporting and analytics for data-driven decision-making.

Scope

The Car Rental System aims to enable rental agencies to conduct their business and customers to rent cars without problems. It involves the following:

- Role-based authentication and access control for Administrators, Staff, and Customers.
- Inventory of cars, tracking, and the availability of vehicles.
- Car rental reservation; also process the payment based on pickup.
- Vehicle maintenance scheduling and condition tracking.
- Usage, revenue, and customer data reporting and analytics.

Definitions, Acronyms and Abbreviations

Term / Acronym	**Definition**
CRUD(Create, Read, Update, Delete)	Basic database operations.
UML (Unified Modeling Language)	A visual representation for specifying, constructing, and documenting software system components.
UI (User Interface)	The part of the system that users interact with directly.
MySQL	Relational database management system employed in the storing and manipulation of data.
JavaFX	This is a framework used to build user interfaces in java
3-Tier Architecture	A software architecture that divides the system into three layers: Presentation (UI), Business Logic, and Data Layer.
RBAC (Role-Based Access Control)	A security mechanism granting access through users' roles and privileges.
Payment Gateway	Third-party service to handle and authorize the payment transactions, which currently is excluded.

Technologies to Be Used

- **Backend Development:** Java classes for all core components with business logic
- **Frontend Development:** Java FX
- **Database Management:** MySQL for data storage
- **Architecture:** 3-Tier architecture (Presentation Layer, Business Logic Layer, Data Layer)
- **Development Tools:** Eclipse (IDE), GitHub (Version Control), and Trello (Project Management).

Overall Description

System Perspective

The Car Rental System is designed as a desktop application for small to medium-sized car rental businesses. It will employ local databases primarily meant to store users, vehicle inventories, and rental histories. For payment, the system communicates with a payment gateway when making reservations and charging cars. The system context diagram illustrates the most significant interactions between users, databases, and the systems utilized in payment processing.

System Functions

- **User Management**: Admin, Staff, and Customer account creation, updating, and removal, and assigning role-based privileges.
- **Car Management:** Manage car inventory by adding, updating, or removing vehicle records while tracking availability based on bookings and returns.
- **Booking Management**: Enabling customers to search and book cars. Confirm availability and issue confirmations.
- Payment Processing: Record payments at the time of car pickup.
- **Maintenance Management**: Monitor and schedule maintenance on car return.
- **Reporting and Analytics**: Create reports on rentals, revenue, and vehicle utilization.

User Classes and Characteristics

Administrators:

a) Manage users, cars, pricing, and business reports.

b) Assign cars to rental requests and supervise system-wide operations.

Staff:

a) Process rental request and return; check vehicle conditions.

b) Assist customers in making reservations.

Customers:

a) Search for the available automobiles, book rentals and make payments.

b) View rental history, extend bookings and update personal details.

Constraints

- **Hardware Constraints**: The system will only be supported on desktop machines running JavaFX and MySQL.
- **Performance Constraints**: The system should manage up to 100 simultaneous bookings without deterioration of performance.
- **Security Constraints**: The system will require role-based access control and sensitive data such as customer information will be encrypted.
- **Data Constraints**: The reservation and rental information are stored locally in a file; hence any lost data can be recovered easily.

Assumptions and Dependencies

Assumptions*:*

1. **Role-based User Access:**

This system assumes all users, including Admin Staff and Customers, act according to pre-defined roles with defined access privileges. To ensure security and control of functions the system assumes that there is no change in the policy to ensure security and control of functions.

2. **User Training:**

   Admins and staff will be given proper training on how to use the system’s features effectively, including user management, booking processes, and maintenance scheduling.

3. **Accurate Data Entry:**

   Users are expected to provide valid and accurate information when registering, booking rentals, and updating vehicle conditions.

4. **Limited Scope (Educational Project):**

   As this is an academic project, some real-world complexities, such as advanced payment integration and external system APIs, are simplified or excluded.

Dependencies:

1. **Database Dependency:**

   The system depends on a local MySQL database for storing and retrieving data, including user profiles, vehicle listings, booking details, and reports. Database availability is critical for system operations.

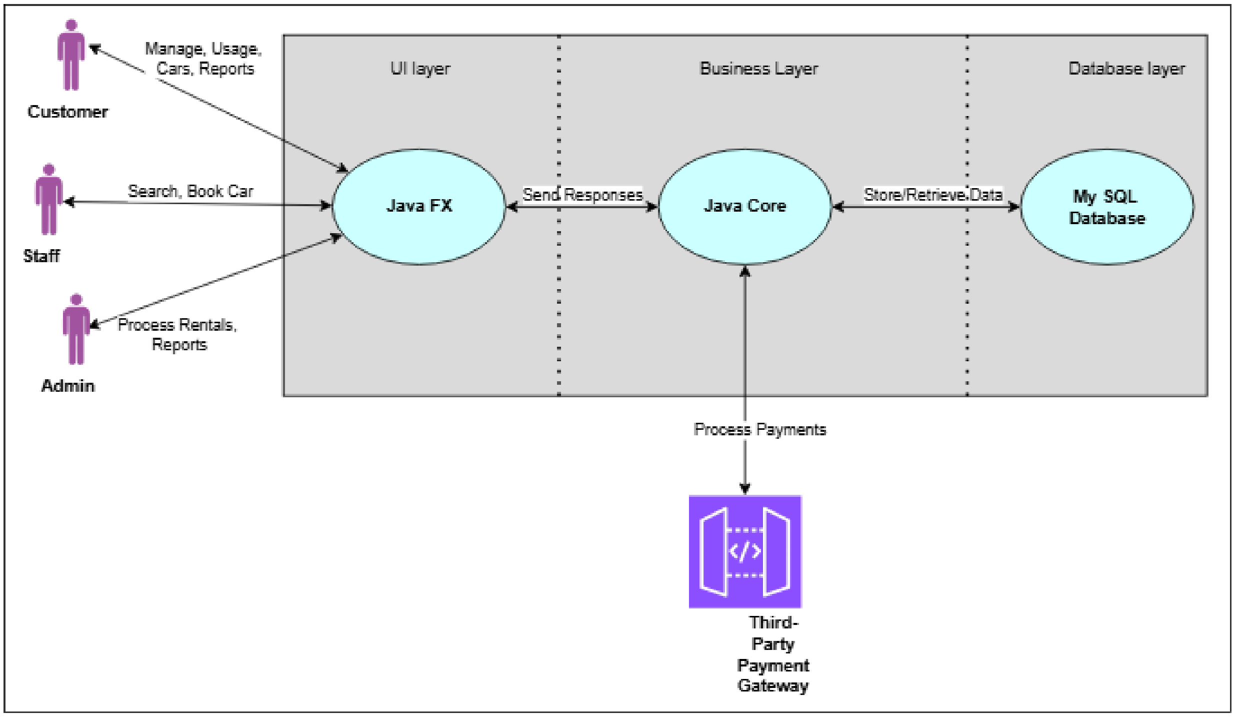
2. **Desktop Environment:**

   The system is designed to run on desktops equipped with the Java runtime environment and JavaFX for rendering UI. Users must install and configure these components properly..

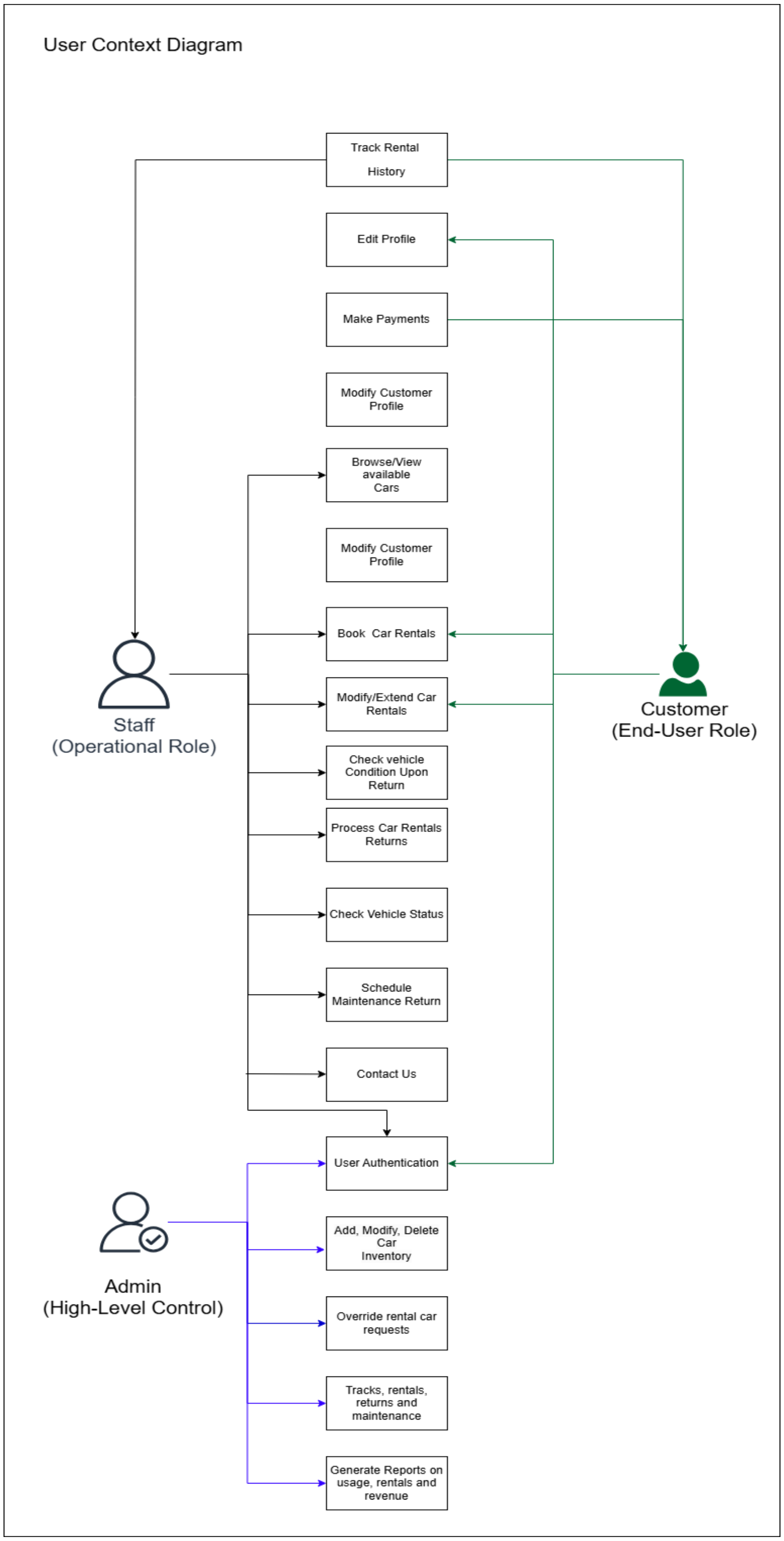
3. **External Libraries:**

   The system may rely on Java libraries for encryption, reporting, and UI components, ensuring secure and efficient system performance.

System Context Diagram



Use Case Diagram


User Context Diagram
Track Rental History
Edit Profile
Make Payments
Modify Customer Profile
Browse/View available Cars
Modify Customer Profile
Book Car Rentals
Modify/Extend Car Rentals
Check vehicle Condition Upon Return
Process Car Rentals Returns
Check Vehicle Status
Schedule Maintenance Return
Contact Us
User Authentication
Add, Modify, Delete Car Inventory
Override rental car requests
Tracks, rentals, returns and maintenance
Generate Reports on usage, rentals and revenue
Staff
(Operational Role)
Customer
(End-User Role)
Admin
(High-Level Control)

Functional Specifications

Feature 1: User Authentication

- **Description:**
  Car Rental system allow users to create an account, log in, and access the application functionalities based on specific roles (Admin, Staff, or Customer).
- **Inputs:**
  - Username will be Freeform text with character limit.
  - Password (contains at least one upper case one lower case one special character)
- **Processes:**
  - Validate user credentials matches with backend record which can be base 64 encoded, so they are not human readable.
  - Assign roles and privileges based on the user's role after successful authentication.
- **Outputs:**
  - Client will be redirected to home page or user-specific dashboard upon successful login.
  - Display Invalid user if authentication fails.

Feature 2: Car Inventory Management

- **Description:**
  Users with admins roles could add, modify, or delete vehicle in the system. They can set pricing, car availability, and track vehicle statuses for rentals and maintenance.

- **Inputs:**
  - Car details (size, colour, model, registration number, availability status, etc.)
  - Pricing information
  - Image of the car
- **Processes:**
  - Create new cars to the inventory.
  - Update existing vehicle details (e.g., availability after maintenance or returns).
  - Delete old or retired vehicles that are not supported by government.
- **Outputs:**
  - Success message for when an inventory is updated or deleted.
  - Updates to car availability and pricing.

Feature 3: Car Rental Booking

- **Description:**
  Customers will use the system to browse available cars, select a vehicle, and complete the booking process based on rental duration and preferences.
- **Inputs:**
  - Search criteria (car size, car model, rental dates, location)
  - Customer details such as age, and identification information
- **Processes:**
  - System Check vehicle available for selected dates.
  - Create a booking confirmation if the car is available.
- **Outputs:**
  - Confirmation number with rental details
  - User friendly message if unable to book the rental transactions.

Feature 4: Rental Return and Vehicle Condition Verification

- **Description:**
  Staff members will validate and inspect vehicles upon customer return, A check list will be added to validate the condition, and schedule maintenance if necessary.
- **Inputs:**
  - Vehicle details during return procedure (rental ID, customer name, return date and fuel level)
  - Check list with Condition status (damage reports, mileage)
- **Processes:**
  - Validate the Questioner Inspect the vehicle for damages and update its condition.
  - Determine if additional charges need to be acquired if maintenance is required and then schedule it accordingly.
- **Outputs:**
  - Updated vehicle status (e.g., "Available" or "Under Maintenance").
  - Maintenance schedule entry (if required).
  - Damage report (if applicable).

Feature 5: Reporting and Analytics

- **Description:**
  Admins shall be able to generate specific reports to maintain car usage, rental history, revenue, and maintenance schedules.
- **Inputs:**
  - A drop-down field to get report type selection.
  - Time frame or filter criteria (e.g., car model or customer)

- **Processes:**
  - Generate relevant data from the database.
  - Download/display reports in tabular formats.
- **Outputs:**
  - Reports displayed on the admin users.
  - Exportable formats (e.g., PDF or CSV).

Interface Requirements

User Interfaces

The system will have a simple, user-friendly interface tailored for each user role (Admin, Staff, and Customer).

- **Login Page:**
    - Users will log in using a username and password. The page will include a "Forgot Password" link and a submit button to authenticate users.
- **Admin Dashboard:**
    - Admins will see a dashboard with options to manage users, cars, bookings, and view reports. The interface will have clear buttons and navigation tabs for switching between tasks like adding cars or viewing rental data.
- **Staff Dashboard:**
    - Staff members will have access to tasks like processing rentals, checking car conditions, and scheduling maintenance. The interface will display pending rentals and car status updates.
- **Customer Interface:**
    - Customers will browse available cars, book rentals, view their rental history, and update personal details.
    - Key pages:
        - **Car Search Page:** Customers can search cars by model, rental duration, or availability using filters.
        - **Booking Page:** Displays car details, rental duration, and a "Confirm Booking" button.

Hardware Interfaces

The Car Rental System will run on desktop or laptop computers with no complex hardware requirements.

- **Desktop or Laptop Computers:**
  - The system is designed to work on desktop systems where Java and MySQL are installed.
  - No specialized hardware is required for this project.

Software Interfaces

The system will interact with a few key software components to manage and store data.

- **MySQL Database:**
  - The system will connect to a local MySQL database to store information about users, vehicles, rentals, and maintenance.
- **Internal System Components:**
  - The UI will interact with the backend services written in Java to process user requests like booking rentals or managing car inventory.
- **Payment Handling:**
  - Payments will be recorded locally during pickup since external payment gateways are not required for this educational project.

Communication Interfaces

The system will use simple, reliable communication methods to manage data flow internally.

- **Internal Communication:**
    - The system's internal components, such as the UI and backend logic, will communicate using Java function calls and database queries.
- **Data Format:**
    - Data will be exchanged between components in a structured format using SQL queries and local variables within the application.

Non-Functional Requirements

Performance Requirements

- The system will ensure a response time of under 30 seconds for user interactions such as browsing available cars and booking rentals under normal operating conditions.
- Database queries, such as checking car availability or retrieving rental history, should execute within **5 second** for 95% of queries.

Other Requirements

- The system shall be designed with a **three-tier architecture**, allowing individual components (e.g., UI, business logic, and database) to be updated independently without affecting the rest of the system.
- The user interface shall be intuitive and easy to navigate, requiring minimal training for Admins, Staff, and Customers.
- Error messages shall be user-friendly, providing clear instructions on how to resolve issues.

Technical References

- Project Guidelines - (Provided by professor on Brightspace)
- Why use Waterfall - (Provided by professor on Brightspace)
- Brightspace Requirements - (Provided by professor on Brightspace)
- Guidelines for SRS - (Provided by professor on Brightspace)