****

# **SOFTWARE DEVELOPMENT LIFE CYCLE**

Software Requirement Specification(SRS)

Car Rental System

**Document Version: 0.01(2025-02-02)**

**Prepared By**

**Department: Computer Science**

**Group: COSC3506-001XE**

**Members:** Deepasree Meena Padmanabhan,

Pujan Bhuva, Simran, OgheneRukevwe Esegba

Table of Contents

[**SOFTWARE DEVELOPMENT LIFE CYCLE** 1](#_Toc189413297)

[**Revision History** 3](#_Toc189413298)

[**Reference Documents** 4](#_Toc189413299)

[**Software Requirement Specification Sign-Off** 5](#_Toc189413300)

[**Introduction** 6](#_Toc189413301)

[Purpose 6](#_Toc189413302)

[Scope 7](#_Toc189413303)

[Technology to Be Used 7](#_Toc189413304)

[**Detailed Description** 8](#_Toc189413305)

[Product Perspective 8](#_Toc189413306)

[Product Features 8](#_Toc189413307)

[User Classes and Characteristics 8](#_Toc189413308)

[**Assumptions** 10](#_Toc189413309)

[**User Analysis** 12](#_Toc189413310)

[Admin 12](#_Toc189413311)

[Staff 12](#_Toc189413312)

[Customers 13](#_Toc189413313)

[**Functional Specification** 14](#_Toc189413314)

[**Functional Specification Table** 15](#_Toc189413315)

[**Non-Functional Specification** 17](#_Toc189413316)

[**Interface Specification** 18](#_Toc189413317)

[Wire/UI Diagram 18](#_Toc189413318)

[UI Content Table with WCAG Standards 18](#_Toc189413319)

[**Outstanding Items** 19](#_Toc189413320)

## **Revision History**

| **Version** | **Date** | **Author** | **Description** |
| --- | --- | --- | --- |
| 0.0a | 2025-01-25 | Deepasree Meena Padmanabhan | Initial Draft |
| 0.1a | 2025-01-26 | Pujan Bhuva | Added Functional Specification |
| 0.1b | 2025-01-28 | Simran | Listed Non-functional specification |
| 0.1c | 2025-01-30 | OgheneRukevwe Esegba | Added UI Tooltips |

## **Reference Documents**

* Project Guidelines - <https://github.com/Deepasree-MP/COSC3506-001XE/blob/main/Resources/BrightSpace/General%20Project%20Guidelines.pdf>
* Why use Waterfall - <https://github.com/Deepasree-MP/COSC3506-001XE/blob/main/Resources/BrightSpace/Why%20use%20Waterfall%20Model.pdf>
* Brightspace Requirements - <https://github.com/Deepasree-MP/COSC3506-001XE/blob/main/Resources/BrightSpace/Car%20Rental%20System.pdf>
* Guidelines for SRS - <https://github.com/Deepasree-MP/COSC3506-001XE/blob/main/Resources/BrightSpace/General%20Guidelines%20for%20Preparing%20SRS.pdf>

## **Software Requirement Specification Sign-Off**

| **Role** | **Name** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Business Analyst (BSA) | Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba |  |  |
| IT Project Manager | Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba |  |  |
| BU Project Manager | Amandeep S Patti |  |  |
| Solution Architect | Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba |  |  |
| Developer Lead | Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba |  |  |
| Quality Engineering Lead | Deepasree Meena Padmanabhan, Pujan Bhuva, Simran, OgheneRukevwe Esegba |  |  |

## **Introduction**

The **Car Rental System** is designed to make renting cars easier and more efficient. It helps rental agencies by automating the process of managing cars, bookings, and customer details. The system has three main users: **Administrators, Staff, and Customers**. Administrators manage inventory, users and pricing, staff handle rentals and maintenance, and customers can browse, book, and pay for cars. This system reduces paperwork, saves time, and improves the overall rental experience

### Purpose

The main goal of this project is to create an **automated and user-friendly Car Rental System** that helps rental agencies efficiently manage their vehicles and customer bookings. By implementing modern **object-oriented programming** and **database-driven architecture**, the system will minimize manual workload and enhance operational efficiency.

This project focuses on:

* **Vehicle inventory management**
* **Automated booking and return processing**
* **Role-based user access and authentication**
* **Tracking vehicle conditions and maintenance schedules**
* **Providing detailed reporting for business insights**

~~Additionally, the system will support~~ **~~multiple locations~~**~~, offer real-time availability. It follows~~ **~~a structured and scalable design~~**~~, ensuring long-term usability.~~

### Scope

The **Car Rental System** aims to help rental agencies manage their business operations efficiently while allowing customers to rent cars with ease. The system includes:

* **Role-based authentication** for different users (Admins, Staff, and Customers).
* **Car inventory management** with vehicle tracking and availability checks.
* **Rental booking system** with pickup-based payment processing.
* **Automated maintenance scheduling** and vehicle condition tracking.
* **Reporting and analytics** to monitor usage, revenue, and customer data.

### Technology to Be Used

* **Backend Development:** Java class for all core components with business logic
* **Frontend Development:** Java FX
* **Database Management:** MySQL for persistent data storage
* **Architecture:** 3-Tier architecture (Presentation Layer, Business Logic Layer, Data Layer)

## **Detailed Description**

### Product Perspective

The Car Rental System is designed as an independent application that can be integrated with application and fleet management systems. The system will work only on the desktop.

### Product Features

* **User Registration & Authentication:** Admins can create and manage accounts securely.
* **Car Inventory Management:** Admins can add, modify, and delete vehicle listings with real-time availability.
* **Booking System:** Customers can check available cars, choose rental durations, and confirm bookings.
* **Payment Management:** Payments will be collected during pickup, as this is an educational project.
* **Rental History Tracking:** Customer can view previous and current rental records.
* **Maintenance Scheduling:** Staff & Admin can monitor vehicle conditions and schedule maintenance.
* **Reports & Analytics:** Admins can generate data-driven reports for business analysis.

### User Classes and Characteristics

* **Administrators**:
  1. Manage users, cars, pricing, and business reports.
  2. Assign cars to rental requests and oversee system-wide operations.
* **Staff**:
  1. Process rental requests and returns, verify vehicle conditions.
  2. Assist customers with booking modifications.
* **Customers**:
  1. Search for available cars, book rentals, and make payments.
  2. View rental history, extend bookings, and update personal details.

## **Assumptions**

|  |  |  |
| --- | --- | --- |
| **No** | **Assumption** | **Description** |
| 1 | Role-based Access | The system will have different access levels for Admins, Staff, and Customers. Each user type will have specific permissions, ensuring security and controlled access to system features. |
| 2 | Admin Control | Admins will have full system control, allowing them to manage users, vehicles, and rental policies. This ensures smooth operations and an efficient booking process. |
| 3 | Staff Permissions | Staff members will have limited control. They can manage rentals, check car conditions, and schedule maintenance but cannot change system settings or delete user accounts. |
| 4 | Customer Booking | Customers can search for cars, make rental bookings, and modify their reservations within system limits. However, they cannot delete cars or change car availability settings. |
| 5 | Audit Log | The system will track and store all activities, including bookings and user actions. This ensures security, accountability, and prevents unauthorized activities. |
| 6 | Vehicle Condition Tracking | Staff members will have access to car condition records. They can check and update vehicle status before and after rentals to track damages. |
| 7 | System Maintenance Schedule | The system will be updated during non-busy hours to prevent disruptions. Users will be informed before any maintenance. |

## **User Context Analysis**

A diagram of a customer service

AI-generated content may be incorrect.

### Admin

Admins have the highest level of access and manage the entire system. Their main tasks include:

* **Managing User Accounts**: Admins can create, update, or remove customer and staff accounts. If a staff member leaves, the admin can deactivate their account.
* **Managing Cars and Pricing**: Admins can add new cars, update rental prices, and apply discounts. For example, during a holiday season, they can increase prices for high-demand vehicles.
* **Tracking Rentals and Maintenance**: Admins can check which cars are rented, due for return, or need maintenance. If a car has been rented for a long time, they can schedule it for servicing.
* **Generating Reports**: Admins can access reports on business performance, including rental trends and revenue. If certain cars are rarely rented, they can decide to replace them.
* **Approving Rental Requests**: Admins assign cars to customers based on availability and user preferences.

### Staff

Staff members assist customers and maintain cars. Their responsibilities include:

* **Handling Rentals and Returns**: Staff members approve bookings and check cars when they are returned. If a customer returns a car late, they ensure any applicable late fees are charged.
* **Inspecting Car Condition**: Staff check the condition of cars upon return. If any damage is found, they report it to the admin for necessary repairs.
* **Scheduling Maintenance**: If a car has issues, staff schedule it for repairs. For example, if a customer reports faulty brakes, staff ensure it is serviced before renting it out again.
* **Helping Customers**: Staff members answer customer queries, such as how to extend a booking or upgrade to a better vehicle.
* **Updating Work Schedule**: Staff can update their availability, ensuring there is always someone available to assist customers.

### Customers

Customers are the end-users who rent cars for personal or business use. Their primary activities include:

* **Searching and Booking Cars**: Customers can browse available cars, choose one that suits their needs, and book it. For example, a customer looking for a family trip might book an SUV.
* **Managing Bookings**: Customers can check their rental history and modify bookings if needed. If they need the car for an extra day, they can request an extension.
* **Making Payments**: Since this is an educational project, customers will pay at the time of pickup, but they can still add payment details for record-keeping.
* **Updating Personal Details**: Customers can change their contact information, such as phone numbers or addresses, ensuring they receive updates.
* **Receiving Notifications**: Customers will get alerts for booking confirmations, due date reminders, and rental status updates via email or SMS.

## **Functional Specification**

**UML/Entity diagram**

To be Determined once SRS is signed off

## **Functional Specification Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Spec ID** | **Specification Description** | **Responsibility** | **Mapped Requirement** |
| FS001 | Admin can create accounts | Admin | Manage users |
| FS009 | Dynamic pricing based on demand | Admin | Pricing |
| FS037 | Subscription-based rental plans | Admin | Pricing |
| FS004 | Generate monthly reports | Admin | Reports |
| FS022 | Data export for financial analysis | Admin | Reports |
| FS023 | Graphical analytics dashboard | Admin | Reports |
| FS029 | The system enforces a minimum and maximum rental duration defined by company policies. | Customer | Booking |
| FS008 | Customer rental history management | Customer | Booking History |
| FS002 | Customers can browse cars | Customer | Browse cars |
| FS027 | Advanced vehicle search filters | Customer | Browse cars |
| FS007 | Vehicle maintenance tracking | Staff | Maintenance |
| FS016 | Vehicle damage report logging | Staff | Maintenance |
| FS003 | Staff can check car status | Staff | Vehicle condition |
| FS005 | Real-time car availability check | System | Availability |
| FS006 | Automated booking confirmation | System | Booking |
| FS028 | Integration with Google Maps for navigation | System | Navigation |
| FS014 | Refunds will require admin approval and depend on cancellation timing. | System | Payments |
| FS015 | Late return penalty calculation | System | Payments |
| FS044 | Performance tracking for system optimization | System | Reports |
| FS010 | User authentication system | System | Security |
| FS036 | Audit logs for user activities | System | Security |
| FS011 | Role-based dashboard access | System | User Access |

## **Non-Functional Specification**

|  |  |  |
| --- | --- | --- |
| **Type** | **Non-Functional Number** | **Description** |
| Audit | NF001 | All transactions must be logged. |
| Privacy | NF004 | User data encrypted at rest and transit. |
| Performance | NF007 | The system must support concurrent users. |
| Portability | NF008 | Support Desktop |

## **Interface Specification**

### Wire/UI Diagram

## **Outstanding Items**

List all clarification during our meeting setup.