Software Requirements Specification

For

Car Rental System

Version 1.0

Prepared by

Group Name: **Car Rental System**

Atluri Charitha Sri	**Student 1**	**atluri_b220213cs@nitc.ac.in**
B Venkata Durga Sahithi	**Student 2**	**bandapalli_b220753cs@nitc.ac.in**
Bharath Nayak Bhukya	**Student 3**	**bharath_b220224cs@nitc.ac.in**
Bhukya Sai Lokesh	**Student 4**	**bhukya_b220227cs@nitc.ac.in**

Instructor:	***Dr. Abdul Nazeer KA, Dr. Chandramani Chaudhary***
Course:	**Data Base and Management System**
Date:	**26-10-2024**

Contents

CONTENTS .... **II**

1 INTRODUCTION **1**

1.1 DOCUMENT PURPOSE 1
1.2 PRODUCT SCOPE 1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW 1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 2
1.5 DOCUMENT CONVENTIONS 2
1.6 REFERENCES AND ACKNOWLEDGMENTS 2

2 OVERALL DESCRIPTION **3**

2.1 PRODUCT OVERVIEW 3
2.2 PRODUCT FUNCTIONALITY 4
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS 4
2.4 ASSUMPTIONS AND DEPENDENCIES 4

3 SPECIFIC REQUIREMENTS **4**

3.1 EXTERNAL INTERFACE REQUIREMENTS 4
3.2 FUNCTIONAL REQUIREMENTS 5
3.3 USE CASE MODEL 6

4 OTHER NON-FUNCTIONAL REQUIREMENTS **7**

4.1 PERFORMANCE REQUIREMENTS 7
4.2 SAFETY AND SECURITY REQUIREMENTS 7
4.3 SOFTWARE QUALITY ATTRIBUTES 8

APPENDIX A – DATA DICTIONARY **9**

APPENDIX B - GROUP LOG **11**

1 Introduction

1.1 Document Purpose

This document outlines the software requirements for the Car Rental System, which will support efficient and secure management of car rental operations, including customer reservations, vehicle tracking, and return processing. This SRS covers the functional and non-functional requirements of the database management subsystem, which is central to the Car Rental System's operations.

The purpose of this Software Requirements Specification (SRS) document is to define and detail the software requirements for the Car Rental System, specifically targeting the database management subsystem. This system is designed to streamline and automate various rental operations, ensuring smooth, secure, and efficient handling of customer reservations, vehicle tracking, rental agreements, and return processing. By providing a comprehensive outline of the functional and non-functional requirements, this SRS serves as a foundational reference for developers, stakeholders, and end-users involved in implementing and using the Car Rental System.

1.2 Product Scope

The Car Rental System Database Management System (DBMS) is designed to optimize and automate the core processes involved in vehicle rentals, such as customer management, vehicle availability tracking, rental agreement processing, and return recording. This system aims to improve the accuracy and consistency of record-keeping, providing real-time status updates on vehicle availability and minimizing the need for manual paperwork. Additionally, by centralizing and securing data, the system enhances data protection, reducing the risk of data loss or unauthorized access.

This DBMS is a powerful tool for car rental agencies, offering a streamlined platform to manage day-to-day operations more efficiently. Key benefits include reduced operational costs, faster and more accurate transaction processing, and an improved customer experience, as staff can access up-to-date information quickly and easily. Ultimately, the system's objective is to provide a reliable, scalable solution that meets the demands of a modern car rental business, supporting both immediate operational needs and long-term growth.

1.3 Intended Audience and Document Overview

This Software Requirements Specification (SRS) document is intended for various audiences involved in the Car Rental System project, including the development team, project managers, the client, and the instructor. It provides a detailed overview of the system's requirements and is structured to guide each reader through the project's objectives, functional expectations, design constraints, and performance criteria. Each section addresses different aspects of the project, ensuring that all stakeholders have a clear understanding of the scope and expectations for system functionality.

The document is organized to facilitate a logical reading flow for all users. It is recommended that readers begin with the ***Introduction*** *to gain an understanding of the project background and objectives, followed by the* ***Overall Description*** *for a comprehensive view of the system architecture and operational context. The* ***Specific Requirements*** *section provides precise details* *on system functionality, while the* ***Non-functional Requirements*** *outline expected performance, reliability, and usability standards. Additional information, technical references, and supporting documentation are included in the* ***Appendices****. This sequence ensures each reader can focus on sections most relevant to their role and understanding of the system.*

1.4 Definitions, Acronyms and Abbreviations

- ***CRUD****: Create, Read, Update, Delete (database operations)*
- ***DBMS****: Database Management System*
- ***GUI****: Graphical User Interface*
- ***IEEE****: Institute of Electrical and Electronics Engineers.*
- ***SRS****: Software Requirements Specification*

1.5 Document Conventions

This document adheres to IEEE standards for SRS documentation, using Arial font size 11 throughout the document for text, italics for comments, single-spaced, with 1" margins. Section titles are bold and numbered.

1.6 References and Acknowledgments

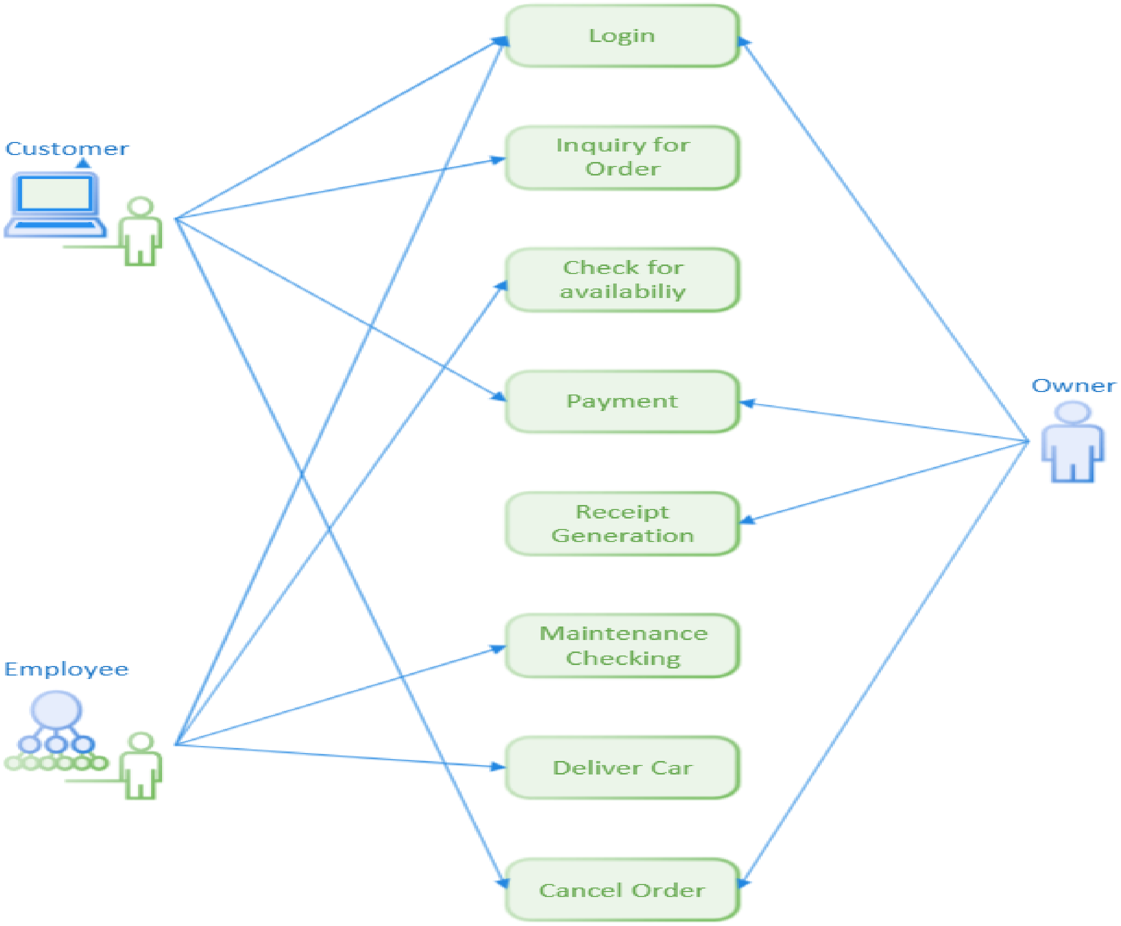
This SRS refers to IEEE documentation standards, DBMS course materials, and software architecture guides. Additional resources include database best practices and user interface design principles.

2 Overall Description

2.1 Product Overview

The Car Rental System is a standalone application designed to provide car rental agencies with a centralized database for managing essential rental operations. This system enables efficient Create, Read, Update, and Delete (CRUD) operations related to customers, vehicles, rental agreements, and return transactions. Serving as an all-in-one platform, it enhances the agency's ability to track vehicle availability, manage customer records, and monitor rental history, ultimately improving workflow efficiency and data accuracy.

Built as a new, self-contained product, the Car Rental System is designed to operate independently, though it may interface with other subsystems such as payment processing or customer relationship management tools if extended in the future. The database is the core component of the system, supporting real-time updates that allow agency staff to make timely, informed decisions. Below is a high-level illustration of the system's main components and their interactions, showing how the database connects with user-facing modules and external interfaces.



2.2 Product Functionality

- ***Customer Management****: Add, update, and delete customer records.*
- ***Vehicle Management****: Track availability and details for each vehicle in inventory.*
- ***Reservation Processing****: Capture and store rental details, including start and end dates.*
- ***Return Processing****: Update vehicle availability upon return and calculate rental fees.*

2.3 Design and Implementation Constraints

The Car Rental System will be developed with several constraints and technical requirements in mind to ensure compatibility, security, and efficient operation. As a database-driven application, it will utilize MySQL as the primary relational database management system (RDBMS), following standard DBMS constraints and supporting SQL for all data manipulation operations. The choice of MySQL allows for robust data storage, fast query execution, and easy scalability, making it ideal for handling rental agency operations that require real-time access to customer and vehicle information. Additionally, the system must incorporate role-based access control (RBAC) to manage permissions and restrict data access according to user roles, ensuring sensitive information is protected.

2.4 Assumptions and Dependencies

- *The database will be hosted on a secure server accessible to the agency.*
- *The system requires internet access for cloud-based data synchronization.*
- *User roles include Admin and Clerk, each with restricted permissions to access and modify data.*

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The Car Rental System will feature an intuitive graphical user interface (GUI) designed specifically for agency staff, enabling efficient management of rental operations and database records. This touchscreen interface will include straightforward menu options to perform CRUD (Create, Read, Update, Delete) tasks on customer, vehicle, and rental data. Users will interact through easily navigable screens and interactive buttons, allowing quick access to core functionalities such as adding new rentals, updating vehicle availability, processing returns, and reviewing customer records. The system's GUI is designed to streamline workflow, making essential database operations accessible and user-friendly for all staff members.

3.1.2 Hardware Interfaces

The Car Rental System application will be compatible with desktop and laptop devices commonly used by agency staff, allowing seamless access to the system's database and interface. The *interface will be optimized for standard computer displays and accessible via basic input devices such as keyboards and mice. Additionally, the system will be designed to read data from connected devices or peripherals as needed, supporting reliable, real-time data entry and retrieval for managing rental transactions and inventory.*

- Processor: 1GHz
- RAM: 512 MB
- Storage: 1 GB or higher
- Internet connectivity for communication interfaces

3.1.3 Software Interfaces

The Car Rental System will integrate with a mobile app, allowing users to send commands directly to the main system for operations like booking and return notifications. The system's core database, powered by MySQL, connects with a front-end GUI built in PHP, providing a seamless interface for agency staff to manage records while also allowing for synchronization with the mobile app. This connection enables real-time updates and secure data exchanges, ensuring that bookings and returns initiated from the app are immediately reflected in the main system, streamlining both user and agency interactions.

The system will interact with the following software components:

***Database Management System (DBMS):** Utilizes MySQL for efficient storage and retrieval of donor, inventory, and request data.*
***Web Server:** Runs on Apache to host the web application and handle client-server interactions.*
***Operating System:** Compatible with Windows, macOS, and Linux environments.*
***Programming Languages:** Developed using HTML, CSS, JavaScript for the front-end, and PHP for server-side processing..*

3.2 Functional Requirements

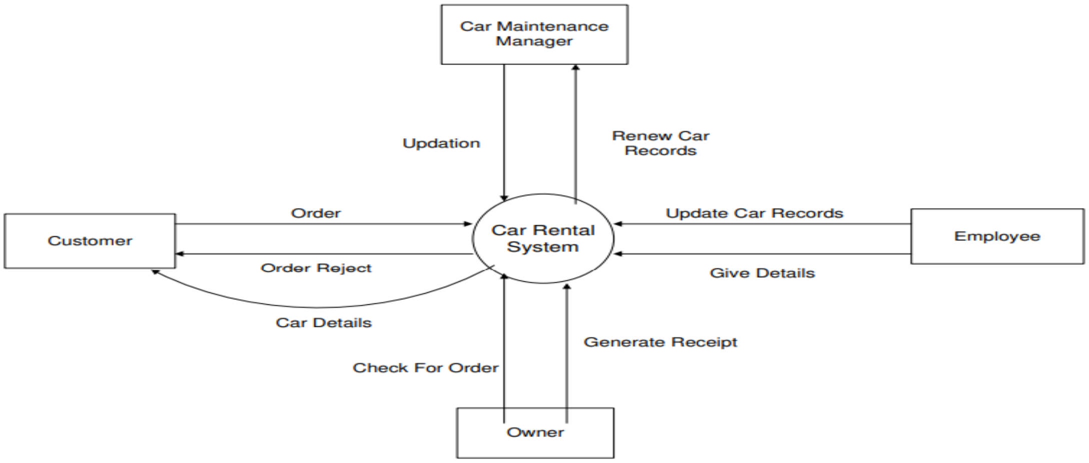
3.2.1 F1: Customer Record Management
- The system shall enable staff to add, update, and delete customer records, including personal information and contact details.

3.2.2 F2: Vehicle Management
- The system shall maintain a comprehensive record of all vehicles, including their availability status, model, type, and other relevant specifications.

3.2.3 F3: Reservation Processing
- The system shall process reservation requests, capturing necessary details and updating the vehicle status to reflect the reservation.

3.2.4 F4: Return and Availability Update
- The system shall update the availability of vehicles upon return and calculate the total rental fees based on the rental duration.

3.3 Use Case Model



3.3.1 Use Case #1: Add New Customer

*- **Author**: ###*
*- **Purpose**: To capture and store new customer details in the system for future reference and transactions.*
*- **Requirements Traceability**: Linked to F1: Customer Record Management.*
*- **Priority**: High **–** Essential for system deployment, as customer details are fundamental for rental operations.*
*- **Preconditions**: The user must be logged in with sufficient permissions to access and modify customer records.*
*- **Postconditions**: The new customer information is stored in the database and can be retrieved by other system components as needed.*
*- **Actors**: Clerk, Admin*
*- **Extends**: None.*
*- **Flow of Events**:*
*1. **Basic Flow**:*
- The user selects "Add New Customer."
- The system prompts the user to input customer details.
- Upon submission, the system saves the new customer information.
*2. **Alternative Flow**: If certain information fields are missing, the system prompts the user to complete them before proceeding.*
*3. **Exceptions**:*
- Database connection issues prevent saving details.
- User session timeout during the process.
*- **Includes**: None.*
*- **Notes/Issues**: None.*

3.3.2 Use Case #2: Process Rental

*- **Author**: ###*
*- **Purpose**: To reserve a vehicle for a customer, assigning it to the customer and marking it as rented.*

*- **Requirements Traceability**: Linked to F3: Reservation Processing and F2: Vehicle Management.*
*- **Priority**: High – Crucial for the core rental operations of the system.*
*- **Preconditions**: Customer information must exist in the system, and the selected vehicle must be available.*
*- **Postconditions**: The vehicle's status is updated to "Rented," and a reservation record is created.*
*- **Actors**: Clerk, Customer*
*- **Extends**: None.*
*- **Flow of Events**:*
*1. **Basic Flow**:*
- The user selects the "Process Rental" option.
- The system checks customer and vehicle availability.
- Upon confirmation, the reservation is created, and the vehicle status is updated.
*2. **Alternative Flow**: If the vehicle is unavailable, the user is notified, and an alternative may be selected.*
*3. **Exceptions**:*
- Insufficient permissions.
- Database error while updating the vehicle status.
*- I**ncludes**: May include customer record retrieval if the customer information needs to be verified.*
*- **Notes/Issues**: Ensure vehicle availability is refreshed in real-time to avoid double bookings.*

4 Other Non-functional Requirements

4.1 Performance Requirements

Performance Requirements

*- **P1**. The system shall process standard database queries, such as retrieving or updating customer and vehicle records, within two seconds to ensure minimal wait time for agency staff and maintain efficient workflow.*

*- **P2**. For reservation processing, the system shall confirm vehicle availability and update the rental status within three seconds of the request to avoid delays during customer interactions and ensure accurate availability tracking.*

*- **P3**. Daily and monthly report generation should complete within five seconds, enabling admin users to quickly access and review performance data without significant downtime, particularly during peak operational hours.*

Each of these performance requirements is designed to facilitate a smooth, responsive experience, reducing potential frustration or delays for users and maintaining the operational efficiency of the car rental service.

4.2 Safety and Security Requirements

Safety and Security Requirements

*- **S1**. The system must enforce user authentication protocols, requiring a unique username and password for all users to access the database and perform any data management actions. User*

sessions shall automatically time out after a 15-minute period of inactivity to prevent unauthorized access.

*- **S2**. Sensitive customer data, including personally identifiable information (PII) and payment details, must be encrypted both in transit and at rest using industry-standard encryption (AES-256) to prevent data breaches. All network communications, including those between the system and any mobile connections, must use SSL/TLS encryption to secure data transfer.*

*- **S3**. Role-based access control (RBAC) will be implemented, restricting data access and system functions based on user roles. For instance, only admin users will be authorized to generate and view financial reports, while clerks will have limited access to customer and vehicle management functionalities.*

*- **S4**. The system shall comply with relevant data protection regulations, such as GDPR or CCPA, ensuring the lawful handling and storage of customer data. Compliance audits will be conducted annually to maintain adherence to these standards.*

*- **S5**. Backup and recovery protocols shall be established, with full system backups occurring every 24 hours and incremental backups every hour. In case of system failure, the system must be capable of full data restoration within one hour to minimize service interruptions and data loss.*

These requirements ensure a secure, resilient system that protects sensitive data and meets industry compliance standards for data security and privacy.

4.3 Software Quality Attributes

4.3.1 Reliability

To ensure the Car Rental System is dependable for everyday operations, the system will implement a comprehensive data backup and error-handling framework. Data backups will occur automatically every 24 hours, storing records in a secure, redundant location to prevent data loss and allow full recovery in case of system failure. Additionally, the system will log error events in real-time to capture issues such as failed data updates or network disruptions, enabling quick diagnostics and recovery. To prevent downtime, a robust data validation process will handle unexpected inputs, and a continuous database health check will monitor connectivity and transaction processing.

4.3.2 Usability

The system’s GUI will prioritize user-friendly design to streamline workflow for agency staff and admin users. A simplified interface layout will feature clearly labeled buttons for core functions, such as "Add Customer" and "Process Rental," and quick-access buttons will facilitate common tasks. Tooltips will be available on all critical fields and buttons to guide new users and reduce training requirements, while dropdowns and autocomplete fields will minimize data entry errors. The system will include a role-based menu structure, tailoring visible options based on user roles (e.g., admin, clerk) to reduce clutter and allow staff to efficiently complete daily tasks without unnecessary complexity.

4.3.3 Adaptability

To support potential future changes, such as integrating additional vehicle types or features like loyalty programs, the system will follow a modular architecture. By organizing features into distinct modules (e.g., Customer Management, Vehicle Management, Reporting), new functionalities can be developed and integrated with minimal disruption to the overall system. The system will also be designed to interface with different payment gateways or extend to additional user interfaces, such as a mobile application, using APIs to enhance system flexibility. This approach allows for *adaptability without requiring significant redesign, supporting long-term maintenance and scalability.*

4.3.4 Maintainability

The Car Rental System will be designed with maintainability in mind, with clearly documented code, standardized naming conventions, and thorough inline comments to aid future developers. The system will employ a layered structure separating the database, business logic, and presentation layers, enabling changes to one layer without impacting others. Regular updates and patches will follow a version-controlled deployment, and comprehensive testing scripts will allow for quick validation after updates. The use of SQL-based query structures also ensures that database interactions can be modified easily, supporting efficient adaptation to new reporting or data handling requirements.

Appendix A – Data Dictionary

Item	**Type**	**Description**	**Possible States/Values**	**Operations/Requirements**
Customer ID	Constant	Unique identifier for each customer in the database.	Numeric (e.g., 1, 2, 3, ...)	F1: Customer Record Management
Customer Name	Variable	Full name of the customer.	String	F1: Customer Record Management
Contact Details	Variable	Customer's phone number and email address.	String	F1: Customer Record Management
Vehicle ID	Constant	Unique identifier for each vehicle in the inventory.	Numeric (e.g., 1001, 1002, ...)	F2: Vehicle Management
Vehicle Model	Variable	Model name of the vehicle.	String	F2: Vehicle Management
Vehicle Type	Variable	Type/category of the vehicle (e.g., sedan, SUV, etc.).	String (e.g., "sedan", "SUV")	F2: Vehicle Management
Availability Status	State Variable	Current status of the vehicle (available or rented).	"Available", "Rented"	F2: Vehicle Management, F3: Reservation Processing

Rental Agreement ID	Constant	Unique identifier for each rental agreement.	Numeric (e.g., 5001, 5002, ...)	F3: Reservation Processing
Start Date	Variable	Start date of the rental period.	Date	F3: Reservation Processing
End Date	Variable	End date of the rental period.	Date	F3: Reservation Processing
Total Rental Fee	Variable	Total cost incurred for the rental period.	Currency (e.g., $0.00)	F4: Return and Availability Update
Payment Details	Variable	Information related to payment method (e.g., credit card, cash).	String	S2: Safety and Security Requirements
Role	Constant	User role within the system (Admin or Clerk).	"Admin", "Clerk"	S3: Role-based Access Control
Session Timeout	Constant	Duration of user session inactivity before auto logout.	Time (15 minutes)	S1: Safety and Security Requirements
Backup Frequency	Constant	Frequency of system data backups.	"Daily", "Hourly"	S5: Backup and Recovery Protocols
Encryption Standard	Constant	Type of encryption used for sensitive data.	"AES-256"	S2: Safety and Security Requirements
Report Generation Time	Variable	Time taken to generate daily/monthly reports.	Time (e.g., under 5 seconds)	P3: Performance Requirements
Error Log	Variable	Logs errors encountered during system operation.	String (error messages)	Reliability and Usability Requirements
GUI Button Labels	Constant	Text displayed on buttons in the GUI.	String (e.g., "Add Customer", "Process Rental")	User Interface Requirements
Mobile App Status	State Variable	Status of synchronization with the mobile app.	"Synchronized", "Pending"	F3: Reservation Processing

Data Compliance Standard	Constant	Regulatory compliance for data handling (e.g., GDPR, CCPA).	"GDPR", "CCPA"	S4: Safety and Security Requirements

Appendix B - Group Log

Minutes

- *OCT 26TH – SRS document preparation meeting, Tasks were divided among the 4 of us and each one had taken a subtopic to be prepared, Also including the main format of the project and outline.*
- *OCT 27TH – 4 of us had updates on what was to be precisely written in the SRS document and had an understanding of the IEEE format of the SRS.*
- *OCT 28TH – Most of us had completed the preparation of their individual parts and started working on formatting and combining all the separate topics.*
- *OCT 29TH – Final discussion about SRS document and error corrections and submission.*

Work

- *Atluri Charitha Sri - Introduction*
- *Bharath Nayak Bhukya - Overall Description*
- *B Venkata Durga Sahithi - Specific Requirements*
- *Bhukya Sai Lokesh - Other Non-functional Requirements*