SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Car Rental System**

**Prepared by:-**

*Team Technocrats*

*Tabaswini P S*

*SriHarinni J E*

*Poornima J K*

# Introduction

## Purpose

The purpose of the Car Rental Management System is to provide a comprehensive and user-friendly application for managing car rentals. It aims to streamline the process of vehicle reservation, enhance billing and payment procedures, and improve the overall experience for both passengers and service providers. The system incorporates real-time tracking, navigation, and safety features to ensure a secure and efficient car rental service. Additionally, it includes innovative features such as fare splitting, pink user preference, and tourist spot recommendations to cater to diverse customer needs.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

The Car Rental Management System is a transformative solution, bringing efficiency to traditional car rental procedures. This application, simplifies vehicle listing and booking. It provides users with transparent information on rental durations and pricing structures, empowering them to make informed decisions. The system streamlines billing processes, incorporating secure payment options for a seamless and convenient experience.

Built for adaptability, the project's modular design allows for the integration of new features, ensuring flexibility and reusability in various scenarios. Developed in Java, the system capitalizes on the language's performance, cross-platform compatibility, and cost-effectiveness. This innovative solution not only modernizes car rental operations but also enhances overall user satisfaction, promising an efficient approach to managing reservations, billing, and customer interactions within the dynamic car rental industry.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

# Overall Descriptions

## Product Perspective

Use Case Diagram of Library Management System

This is a high-level diagram of the car rental booking system, offering a fundamental overview. The users may fall into two categories: service providers or customers. The system features a search function to simplify the exploration of available vehicles, utilizing criteria such as car model. Users of the system, whether service providers or customers, can initiate requests for booking, renewal, or return of cars, adhering to specific criteria. The system streamlines the process, ensuring a user-friendly experience for both the stakeholders.

## Product Function

Entity Relationship Diagram of Car Rental System

The Car Rental Management System provides real-time details on vehicle availability and reservations, aiming to streamline operations and minimize manual efforts. This software efficiently handles vehicle reservations, returns, fee calculations, and customized report generation. Car rental staff, designated as service providers, have control over user accounts and vehicle management. The system's database maintains reservation and return statuses for users, enabling service providers to access user details as needed. Valid users can conveniently view their account information, book cars, ensuring a user-friendly experience in managing car bookings. Importantly, the system safeguards security by restricting vehicle drivers from accessing and modifying the database.

## Assumptions and Dependencies

Assumptions:

* + - The coding for the car rental system should be free of errors.
    - The system must prioritize user-friendliness, ensuring ease of use for all users.
    - Information about users, vehicles, and reservations must be stored in an accessible database.
    - The system should offer ample storage capacity and swift database access.
    - Efficient search functionality and support for quick transactions are integral to the system.
    - The Car Rental System operates 24/7 to cater to users' needs.
    - Users can access the system from any computer with internet browsing capabilities.
    - Access to online accounts and performing actions requires correct usernames and passwords.

Dependencies:

* + - Specific software configurations are essential for the proper functioning of the car rental system.
    - Development and execution of the project are based on specified requirements.
    - Administrators must have a comprehensive understanding of the car rental system.
    - The system relies on accurate data storage and retrieval for general reports.
    - User information must be stored in a database accessible by the Car Rental System.
    - Any updates or modifications to vehicle information are recorded in the database, emphasizing the importance of accurate data entry.

## Data Requirement

In the Car Rental Booking System, input involves queries to the database, and output comprises solutions for these queries. Additionally, the output encompasses users receiving details related to their car rental activities. Input queries from users may include actions such as creating an account, selecting specific vehicles, and adding them to their account. Subsequently, the output becomes visible when users request the server to retrieve details about their car rental account, presenting information such as the time, date, and the current list of booked vehicles associated with their account. This ensures a dynamic and responsive interaction between users and the car rental system.

# External Interface Requirement

## GUI

The software offers an intuitive graphical interface for users and administrators, facilitating tasks like booking, updating, and viewing vehicle details.

* Quick Reports: Users can easily generate reports on rented/returned vehicles within specific time frames.
* Stock Verification and Search: Enables users to verify stock and search for vehicles based on various criteria.
* Customizable Interface: Administrators can customize the user interface to meet specific needs.
* Module Integration: All software modules seamlessly integrate into the graphical user interface, adhering to defined standards.
* Simple Design: The interface is designed for simplicity, maintaining a standard template across different modules.

Login Interface:

Users can register and create an account. Logging in requires a correct username and password, displaying an error message for incorrect entries.

Search:

Members or administrators can search for specific vehicles by entering the vehicle type and model title.

Categories View:

Displays available vehicle categories, allowing administrators to add, edit, or delete categories.

Administrator's Control Panel:

Enables administrators to manage users, add/remove vehicles, and control lending options.

# System Features

The users of the system should be provided the surety that their account is secure. Ensuring user account security is a top priority in the Car Rental Booking System, achieved through:

* Members are authenticated using unique IDs, ensuring secure access to their accounts.
* Administrators monitor and update account statuses, issue alerts if booking limits are exceeded, and apply fines for late returns.
* Members can only access and manage their own accounts.
* Administrators have exclusive access to view and handle all member accounts, enhancing overall system security.

# Other Non-functional Requirements

## Performance Requirement

The proposed Car Rental Booking System aims to be the primary performance solution for users seeking rental cars, ensuring effective interactions for both customers and administrators.

* + The system must exhibit fast response times, ensuring that users experience minimal delays when making reservations, viewing vehicle details, or processing transactions.
  + The system should handle a high transaction throughput, allowing multiple users to concurrently make bookings, returns, or inquiries without degradation in performance.
  + Efficient database operations are crucial for quick access to vehicle information, user details, and reservation records. The system must optimize database queries to maintain responsiveness even as the volume of data increases.

## Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## Security Requirement

* + - System will use secured database
    - Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
    - System will have different types of users and every user has access constraints
    - Proper user authentication should be provided
    - No one should be able to hack users’ password
    - There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

## Requirement attributes

* + - There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
    - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

In the Car Rental Booking System, users are categorized as Customers, Service Providers and Administrators. Customers and Service Providers, assumed to have basic computer and internet skills, interact with a user-friendly interface. To ensure smooth system usage, comprehensive user documentation, including a user manual, online help, and installation guides, should be provided to educate users on system operations, promoting a trouble-free experience.

Administrator facilities for users include:

* Backup and Recovery
* Forgot Password
* Data Migration i.e. storing user data on the server during initial registration.
* Data Replication i.e. safeguarding against data loss by maintaining duplicates on the server.
* Auto Recovery i.e. regularly auto-saving information to prevent data loss.
* File Organization i. e .efficiently managing and organizing system files.
* Server Maintenance i.e. regular updates and maintenance to ensure system efficiency and security.

# Other Requirements

## Data and Category Requirement

In the Car Rental Booking System, distinct user categories include Administrative staff, Rental Staff, and Customers. Access rights vary based on user roles; administrators have privileges to modify, delete, and append data. All other users, excluding rental staff, possess retrieval rights only. Vehicle categories are also established, ensuring that relevant data is displayed according to these classifications. The system is designed to code and categorize vehicle data appropriately for each distinct category. This approach ensures efficient and secure access control, tailoring the user experience based on their specific roles and responsibilities within the car rental system.

## Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - Administrator: A login id representing a user with user administration privileges to the software
    - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘Librarian’, ‘Member’ and ‘Books’ are the most important classes which are related to other classes