

**SOFTWARE**

**REQUIREMENTS**

**SPECIFICATION**

**For**

**Car Rental System**

**Prepared by:**

Priyadharshini M

Shubigsha G

Sindhuja G

**1.Introduction:**

**1.1 Purpose**

The main objective of this document is to illustrate the requirements of the project Car Rental System. The document gives the detailed description of the both functional and non-functional requirements. The purpose of this project is to provide a friendly environment for users and agencies to rent cars. This project describes the hardware and software interface requirements using ER diagrams.

**1.2 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

**1.3 Scope of Development Project**

Car Rental System is basically renting vehicles through online booking so that users and car agencies can efficiently work which makes the process faster. This win-win empowers users with convenient car access, transparent pricing, and secure transactions, while agencies benefit from maximized resource utilization and reduced paperwork.

This project is specifically designed for the use of renting cars. Car Rental System can be used by a car renting agency to rent its cars and monitor the duration of the cars which are rented. The agencies can also use the system to update the details from their end. The users can use it to rent cars which fits their budget and find the availability of cars for the required duration easily.

The language used for developing the project is Java, and it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

**1.4 Definitions, Acronyms and Abbreviations**

JAVA -> platform independence

SQL-> Structured query Language

ER-> Entity Relationship

IDE-> Integrated Development Environment

SRS-> Software Requirement Specification

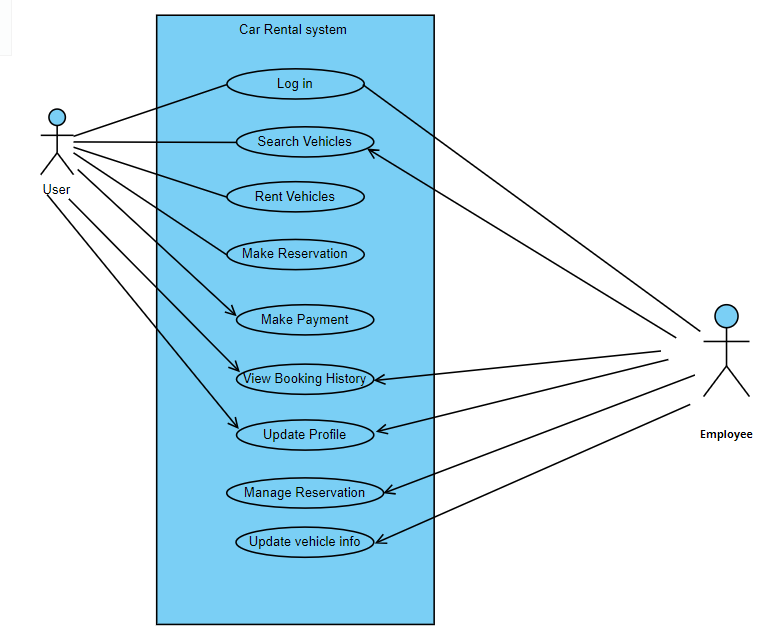
**1.5 References**

* Books
* Car Rental Management System: by NjohPrince
* Car Rental System: by Afrozchakure
* Reports
* Car Rental System Database Design Project: by Scribd
* Car Rental Management System Project Report: by Studocu
* Websites
* <https://instanteduhelp.com/car-rental-management-system/>
* <https://www.hitech.fr/web-services-car-rental/>
* <https://www.scribd.com/document/379538921/Car-Rental-System>

**2. Overall Descriptions**

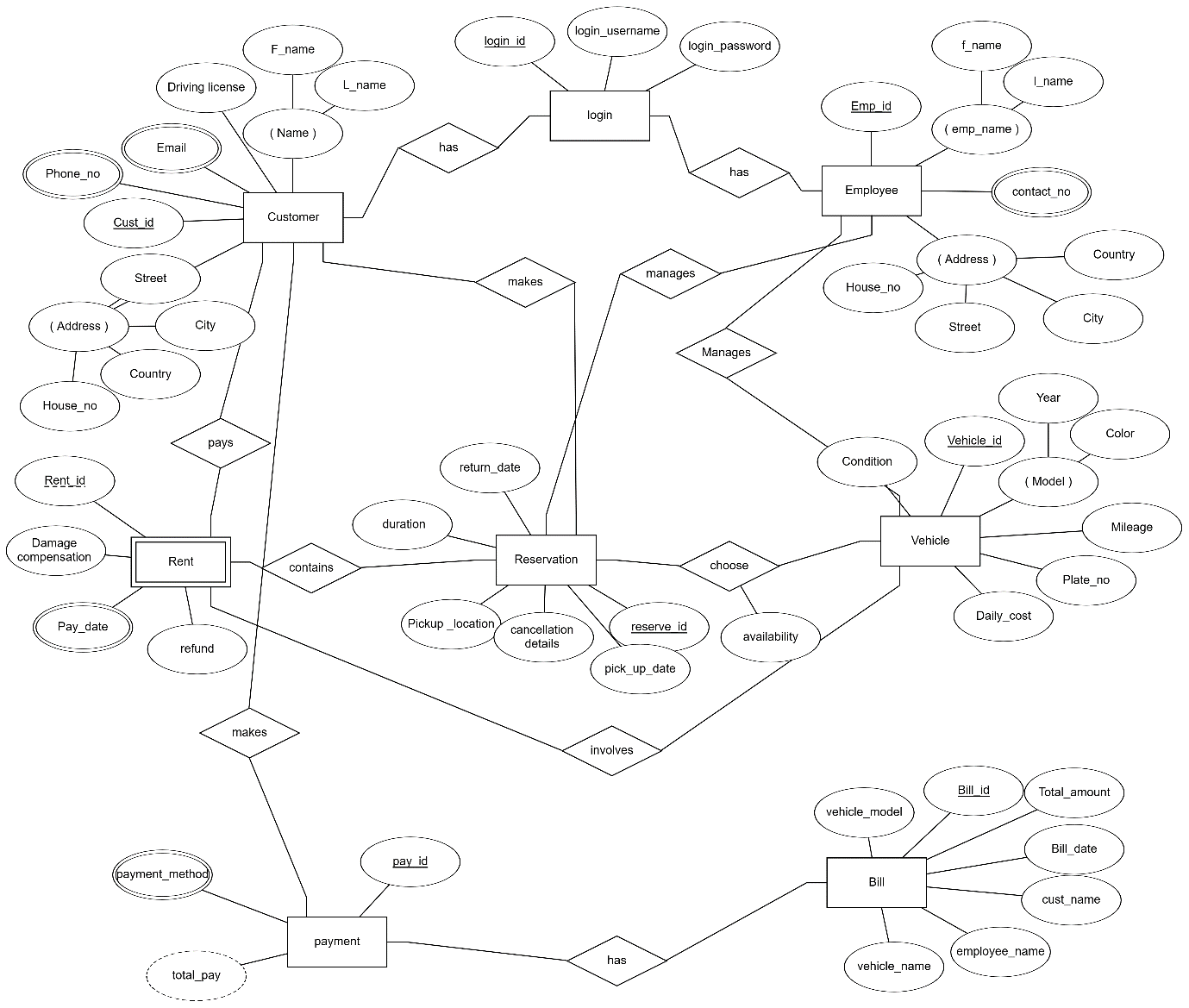
**2.1 Product Perspective**

The users are members who are in need of renting a car. This system will provide a search functionality to facilitate the search of cars. This search will be based on various categories viz. price or the car model. Further each car agency is given permission to update the agency profile i.e, by adding or deleting or updating cars. The users of the system can rent the car of their choice based on the availability and process the payment.



**2.2 Product Function**

The Car Rental System provides information about the cars available for the specified duration. The users need to register themselves and can search the availability of cars based on the pick up and drop up date and time. Later, they can process the payment. The system will send the reservation notification. The worker of the agency can modify the data like adding new cars or deleting a car, etc. The user will be notified about overdue notification. The reservation can be cancelled by both the users and the admins in case of car unavailability. The user will be notified of reservation cancellation. Only valid members are allowed to modify the details displayed.

****

**2.3 User Classes and Characteristics**

The system provides different types of services based on the type of users [User/Admin/Worker]. The user will be acting as the customer and he will only be able to register, book and cancel car reservations, as well as update his profile along with viewing car booking history. The admin will be able to create, update and delete accounts. They can cancel reservation in case of car unavailability.

The features that are available to the Renter: -

* Search for vehicles based on type, size, features, and budget
* Make online reservations with confirmation emails and manage bookings
* Create profiles
* Manage payment methods
* Securely pay for rentals online using various payment methods
* Choose pickup and drop-off locations
* Request changes to bookings, such as extending rental periods or adding extras
* Can cancel or modify the reservations
* Can view the history of cars rented by them previously

The features that are available to car rental agency: -

* Set pricing based on seasons, demand, and vehicle type
* View renter details and rental history
* Can add, edit, or delete cars and their information to the database
* Can view the list of cars available in each category and type
* Can confirm, cancel, or modify the reservations
* Can view the payments made by the customers and their mode
* Can check the report of the rented cars

**2.4 Operating Environment**

The car rental system will operate seamlessly within the Windows environment. Accessible through all major web browsers, including Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox, the system prioritizes compatibility even with older versions like IE 6.0. While most features will function flawlessly across all browsers, a few advanced functionalities might require Mozilla Firefox or Opera versions 7.0 and above. Just an active internet connection is all you need to unlock the full potential of this online platform.

As for hardware specifications, a 40 GB hard drive, a standard 15-inch colour monitor, and a 122-key keyboard will suffice. Basic input devices like a keyboard and mouse, along with output devices such as a monitor and printer, are all that's required to get started. In essence, the car rental system is designed for user-friendliness and accessibility, ensuring a smooth experience regardless of your technical setup.

**2.5 Assumptions and Dependencies**

The assumptions are: -

* Flawless code: The car rental system's code should be meticulously written and thoroughly tested to ensure smooth operation and zero errors.
* User-friendliness: From booking to checkout, the system should be intuitive and easy to navigate for users of all technical backgrounds.
* Robust database: All user, vehicle, rental, and financial data must be securely stored in a central, accessible database.
* Internet accessibility: Users should be able to access the system from any internet-connected device, regardless of operating system or browser.
* 24/7 availability: The car rental platform should be operational 24/7, allowing users to book vehicles and manage rentals at any time.
* Secure login: Usernames and passwords must be robust and protected to ensure individual account security and prevent unauthorized access.
* Performance: The system should handle high traffic volumes and complex queries efficiently, offering fast response times and seamless transactions.

The dependencies are: -

* Hardware and software: The system's functionality relies on specific hardware (servers, network infrastructure) and software (operating system, database, web server) configurations.
* Requirements and specifications: Development and implementation are dictated by the defined list of requirements and specifications.
* Admin expertise: End users, particularly administrators, should possess a basic understanding of the system's functionalities and procedures.
* Reporting system: The car rental system should generate and store reports on rentals, finances, and user activity for informed decision-making.
* Real-time updates: Any changes in vehicle availability, bookings, or user information must be promptly reflected in the database to maintain data consistency.
* Data integrity: Accurate and consistent data entry is crucial for maintaining a reliable database and generating accurate reports.

**2.6 Requirement**

Software Configuration: -

* This software package is developed using java as front end which is supported by sun microsystem. Microsoft SQL Server as the back end to store the database.
* Operating System: Windows NT, windows 98, Windows XP
* Language: Java Runtime Environment, Net beans 7.0.1 (front end)
* Database: MS SQL Server (back end)

Hardware Configuration: -

* Processor: Pentium(R)Dual-core CPU
* Hard Disk: 40GB
* RAM: 256 MB or more

**2.7 Data Requirement**

In the car rental system project, the inputs are comprised of queries submitted by users, each serving a specific purpose in the system. Users can interact with the system by issuing queries such as creating an account, selecting specific cars, and updating their account with rented vehicles. The system is designed to efficiently handle these queries and generate appropriate outputs. For instance, when a user requests to create an account, the system processes the input query and executes the necessary operations to establish an account for that user. Similarly, when users choose to select cars for rental or add vehicles to their existing account, the system interprets these queries and updates the database accordingly.

The primary output of the system comes into play when users inquire about the details of their account. When a user requests information about their account, the system responds by presenting relevant details such as the date and time of their transactions and a comprehensive list of the cars currently associated with their account. This output is crucial for users to keep track of their rental history, understand the status of their account, and manage their car rental activities effectively.

**3. External Interface Requirement**

**3.1 GUI**

GUI The software provides a user-friendly graphical interface for the customer and the administrator to operate on the system, performing the required tasks such as booking, updating, viewing the details of the car.

* It allows the customer to view the availability and rates of different cars based on their location and date of rental.
* It provides a search facility based on various criteria such as car type, model, brand, etc.
* All the modules provided with the software must fit into this graphical user interface and adhere to the standard defined.
* The design should be simple and consistent across all the different interfaces.

Login Interface:

In case the customer is not yet registered, they can enter their details and register to create an account. Once their account is created, they can ‘Login’ which asks the customer to type their username and password. If the customer entered either their username or password incorrectly, then an error message appears.

Search: -

The customer or administrator can enter the type of car they are looking for and the location they are interested in, then they can search for the available cars by entering the car name or other filters.

Categories View: -

Category’s view shows the categories of cars available and provides the ability to the administrator to add/edit or delete a category from the list.

**4. System Features**

The customers of the system should be provided the assurance that their account is secure. This is possible by providing: -

* User authentication and validation of customers using their unique customer ID.
* Proper monitoring by the administrator which includes updating account status, showing a popup if the customer attempts to book a number of cars that exceed the limit provided by the car rental policy, charging fees to customers who return the cars late or damaged.
* Proper accountability which includes not allowing a customer to see other customer’s account. Only administrator will see and manage all customer accounts.

**5. Other Non-functional Requirements**

**5.1 Performance Requirement**

A car rental system is expected to perform functionally all the requirements that are specified by the car rental company. The performance of the system should be fast and accurate. The system shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus, it should have inbuilt error testing to identify invalid username/password. The system should be able to handle a large amount of data. Thus, it should accommodate a high number of cars and users without any fault.

**5.2 Safety Requirement**

The database may be corrupted or lost at any time due to virus, hacking, or hardware failure. Therefore, it is required to take the database backup regularly and store it in a secure location. Proper security measures should be implemented to prevent unauthorized access or modification of the data. The system should also have a backup power supply in case of power outage.

**5.3 Security Requirement**

* System will use encrypted database.
* Normal customers can only view and book the available cars but they cannot edit or modify anything except their personal and payment 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 administrator and customers such that no customer can access the database and only administrator has the rights to update the database.

**5.4 Requirement Attributes**

* There may be multiple administrators managing the system, all of them will have the right to make changes to the system. But the customers or other users cannot make changes.
* The system 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 system.
* The user should be able to easily download and install the system.

**5.5 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 car rental 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.

**5.6 User Requirement**

The users of the system are customers and administrator of the car rental company who manage the system. The customers are assumed to have basic knowledge of the computers and internet browsing. The administrator of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to network issues, security breaches, and other mishaps to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems. The administrator provides certain facilities to the users in the form of: -

* Backup and Recovery
* Forgot Password
* Data migration i.e., whenever a customer registers for the first time then the data is stored in the server
* Data replication i.e., if the data is lost in one branch, it is still stored with the server
* Auto Recovery i.e., frequently auto saving the information
* Maintaining files i.e., File Organization
* The server must be maintained regularly and it has to be updated from time to time

**6. Other Requirements**

**6.1 Data and Category Requirement**

There are different categories of users namely customers, staff, managers, and administrators. Depending on the category of user, the access rights are decided. For example, if the user is an administrator, then they can modify, delete, append, or view any data in the database. All other users except the staff can only view the information about the available cars, their prices, and their locations. Similarly, there will be different categories of cars available, such as economy, luxury, SUV, etc. According to the categories of cars, their relevant data should be displayed, such as model, year, mileage, fuel type, etc. The categories and the data related to each category should be coded in a particular format.

**6.2 Appendix**

A: Administrator, Abbreviation, Acronym, Assumptions; B: Booking, Business rules; C: Car, Category, Customer; D: Database, Data requirement, Dependencies; F: Fuel type; G: GUI; J: JavaFX; L: Location; M: Manager, Mileage, Model; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, Staff, System features; U: User, User class and characteristics, User requirement; Y: Year**.**

**6.3 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

**6.4 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 ‘Employee’, ‘Customer’ and ‘Reservation’ are the most important classes which are related to other classes.

