Software Requirements Specification

for

CarManagerDB

Version 1.0 approved

Prepared by Nathan Mathew Verghese (IMT2022022),

Divyam Sareen (IMT2022010), Manda Kausthubh (IMT2022027)

Team ID - 28

International Institute of Information Technology Bangalore

5-11-2024

Table of Contents

Name		Date	Reason For Changes	Version
We love Manda		5-11-2024	Initial draft	1.0

1. Introduction

1.1 Purpose

The **CarManagerDB** management system is designed to streamline the management of a car dealership's vehicle inventory and sales processes. This system will provide a unified database to track the dealership's fleet, manage vehicle availability, record sales transactions, and store customer information. By automating key aspects of dealership operations, the system aims to enhance efficiency and accuracy, enabling the dealership to offer a seamless and satisfying customer experience throughout the purchase journey.

1.2 Document Conventions

IEEE SRS template

1.3 Intended Audience and Reading Suggestions

This document is intended for:

1. Dealership owners
2. Inventory managers
3. Financing accountants
4. Customer support

This document highlights how **CarManagerDB** will transform the business operations of a dealership. It provides an overview of the system’s goals to streamline inventory management, enhance sales tracking, and improve customer service.

1.4 Product Scope

The scope of the CarManagerDB is:

1. Manage car inventory for a dealership.
2. Track the purchases and details of a certain customer.
3. View the cars a customer owns to help in service.
4. Find information about a particular vehicle.

2. Overall Description

2.1 Product Perspective

The **CarManagerDB** project is a standalone, self-contained database management system designed specifically for car dealerships. Unlike traditional database solutions that require users to write complex SQL commands to manage and retrieve data, CarManagerDB provides an intuitive interface for dealership staff to use DBMS without advanced technical knowledge.

2.2 Product Functions

The **CarManagerDB** allows users to perform the following functions:

- Add/Remove a Car/Car Model/Customer
- Add a Customer's Car
- View details of a car/customer
- View Customer's Cars
- Get the Latest Car Models of a Brand
- Change Car Model's Generation and Mileage
- Change Car's Colour
- Update Customer Contact Information

2.3 User Classes and Characteristics

1. **Dealership owners**

   Dealership owners are major consumers of this product to manage overall car inventory and statistics. Using data from **CarManagerDB** they make executive decisions on managing both inventory and customers.

2. **Inventory managers**

   Inventory Managers focus on tracking and managing the dealership’s vehicle stock. They are key users who require precise and up-to-date information on vehicle availability to maintain optimal stock levels and ensure vehicles are ready for sale.

3. **Financing accountants**

   Financing accountants, with the help of **CarManagerDB**, cross-verify and follow up on purchases of each customer and take care of sales by tracking the number of cars sold.

4. **Customer support**

   Customer support staff use the system to access customer details, record service requests, and manage appointment scheduling. They require straightforward access to customer information to ensure efficient and accurate service.

2.4 Operating Environment

CarManagerDB works on any popular operating system, with minimal hardware requirements.

2.5 Design and Implementation Constraints

A MySQL license for enterprises is required to have to run **CarManagerDB**.

2.6 User Documentation

1. MySQL documentation - https://dev.mysql.com/doc/
2. Java documentation - https://docs.oracle.com/en/java/
3. JDBC documentation - https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/

2.7 Assumptions and Dependencies

The dependencies required to run **CarManagerDB** are the following:

1. OpenJDK@17 and above
2. Latest JDBC version supporting OpenJDK@17 and above
3. MySQL version 9.0.1

3. External Interface Requirements

3.1 User Interfaces

CarManagerDB works on an OS terminal or shell and will be menu-driven to access all features. All features will self-save on the DB after the feature has finished execution.

3.2 Software Interfaces

The CarManagerDB system uses Java and the MySQL JDBC connector to interface with a MySQL database, facilitating smooth data management. Java provides a flexible, platform-independent environment, while JDBC enables secure SQL communication between Java and MySQL. The MySQL database stores all the data for **CarManagerDB**.

3.3 Communications Interfaces

CarManagerDB uses TCP/IP for communication between the Java application and the MySQL database.

4. System Features

4.1 Add/Remove Cars and Customers

4.1.1 Description and Priority

This feature allows administrators to add new cars or customers to the system and remove existing ones when necessary. This functionality is essential for maintaining accurate records of resources and clients, making it a high-priority feature.

4.1.2 Functional Requirements

REQ-1: The system shall allow administrators to add (and remove) a new car by entering the car's details, including make, model, year, VIN, license plate, and any additional relevant information.

REQ-2: The system shall provide a user interface to add (and remove) a new customer, capturing customer details such as name, contact information, driver's license number, and address.

REQ-3: Upon successful addition or removal of a car or customer, the system shall update the database and refresh the user interface to reflect the changes in real-time.

4.2 Add Customer Car

4.1.1 Description and Priority

This feature allows a customer to be linked to the purchase of a certain car, allowing data to be stored for future service of the customer. It is a high-priority feature.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to add a new car under a customer's profile by entering necessary details such as car ID, sale date, transaction ID, and price.

REQ-2: The system shall automatically validate the car ID as well as the customer ID while adding a new row to the database.

4.3 View Details of a Car/Customer

4.1.1 Description and Priority

This feature allows you to view a car or a customer's details for reference. This is a high-priority feature.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to view the complete details of a car or a customer by selecting the relevant item from the list.

REQ-2: The system shall display all relevant information of the car, such as model, registration number, mileage, and service history.

REQ-3: The system shall display the customer's full name, contact information, and associated car details.

4.4 View Customer’s Cars

4.1.1 Description and Priority

This feature allows the user to view the details of the customer's cars for reference. It is a high-priority feature that enables quick access to important car information associated with a customer.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to view the complete details of the customer's cars by selecting the relevant item from the list.

REQ-2: The system shall display all relevant information of the car, such as model, registration number, mileage, and service history.

REQ-3: The system shall display the customer's full name, contact information, and associated car details.

4.5 Get the Latest Car Models of a Brand

4.1.1 Description and Priority

This feature allows the user to retrieve the latest car models available for a specific brand. It is a high-priority feature as it provides up-to-date information for potential customers or inventory management.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to select a brand and retrieve a list of the latest car models from that brand.

REQ-2: The system shall display the model name, release year, and key specifications of each latest car model.

4.6 Change the Car Model's Generation and Mileage

4.1.1 Description and Priority

This feature allows the user to change the generation and mileage information of a specific car model. This is a medium-priority feature.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to update the generation of a selected car model.

REQ-2: The system shall allow the user to update the mileage of a selected car model.

4.7 Change Car's Colour

4.1.1 Description and Priority

This feature allows the user to update the colour of a specific car. This is a medium-priority feature.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to select and update the colour of a specific car.

REQ-2: The system shall display a list of available colour options for the car.

4.8 Update Customer Contact Information

4.1.1 Description and Priority

This feature allows the user to update a customer's contact information. This is a medium-priority feature.

4.1.2 Functional Requirements

REQ-1: The system shall allow the user to update the contact information (e.g., phone number, email address) for a customer.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The **CarManagerDB** must ensure quick response times, with vehicle details loading in under 2 seconds and critical transactions processed within 3 seconds under peak load. It should handle up to 10,000 transactions per hour during high-traffic periods and scale to support expanding dealership networks seamlessly. Real-time updates, such as inventory changes, should reflect in under 2 seconds, while batch processes, like daily reports, must be completed within 30 minutes. Additionally, data replication for consistency must occur within 1 second, and real-time synchronization between locations should not exceed 5 seconds, ensuring data reliability and a smooth user experience.

5.2 Safety Requirements

The **CarManagerDB** must implement safeguards to prevent data loss or corruption, including automatic data backups every 24 hours and redundant storage systems to ensure data availability in case of hardware failure. Access to sensitive information, such as financial records and personal customer data, must be restricted through robust authentication and authorization mechanisms to prevent unauthorized access and mitigate the risk of data breaches. The system must comply with data protection regulations, such as GDPR or CCPA, to ensure customer privacy and the secure handling of information.

5.3 Security Requirements

The **CarManagerDB** must enforce strict security measures to protect data confidentiality, integrity, and availability. User identity authentication must include multi-factor authentication (MFA) for administrators and staff accessing sensitive information, with role-based access control (RBAC) to ensure users can only access the data necessary for their roles. All data, both in transit and at rest, must be encrypted using industry-standard encryption protocols. Compliance with security certifications like ISO/IEC 27001 must be achieved to guarantee that the system meets established standards for data protection and cybersecurity.

5.4 Software Quality Attributes

1. Reliability - The **CarManagerDB** must prioritize reliability, with a target uptime of 99.9% to ensure the system remains accessible during all business hours, minimizing disruptions.
2. Availability - It must be reinforced through redundant systems and automatic failover mechanisms.
3. Usability - There must be an intuitive user interface that enables dealership staff to perform tasks efficiently, prioritizing ease of use over ease of learning for new users.
4. Flexibility and Adaptability - Flexibility and adaptability are very important, allowing the system to be easily modified to accommodate changing business requirements, such as adding new dealership locations or integrating third-party tools.
5. Robustness - The DBMS must be robust, handling unexpected input gracefully without data loss or corruption.

5.5 Business Rules

1. Sales Staff:
   - Can initiate and complete vehicle sales transactions.
   - Must obtain managerial approval for discounts or price adjustments.
2. Service Department Staff:
   - Authorized to schedule and update service appointments.
   - Cannot modify sales records or access financial data.
3. Inventory Managers:
   - Authorized to add, update, or remove vehicle inventory.
   - All inventory changes must be logged for auditing purposes.
4. Customer Service Representatives:
   - Can view customer records and service history.
   - Cannot access or modify financial data.
5. Finance Personnel:
   - Have access to financial and sensitive customer information.
   - Must use multi-factor authentication for access.
6. System Administrators:
   - Responsible for managing user roles and permissions.
   - The only ones allowed to adjust system settings and configurations.

6. Other Requirements

1. Data Integrity Constraints:

   1.1 Constraints that ensure data correctness, such as ensuring email and phone number formats in the `Customer` table are valid.

   1.2 `AvgMileage` in `CarModel` is within a realistic range.

2. A list of additional constraints such as a limit on the number of cars a customer can purchase or any additional offers have to be managed manually and are not supported.

3. Additional constraints and warnings:
   If a `Car` is deleted, ensure any associated records in `CustomerCar` are removed to avoid orphaned records.

Appendix A: Glossary

1. **JDBC (Java Database Connectivity)** is an API (Application Programming Interface) in Java that allows applications to interact with databases. It enables Java programs to connect to a database, execute SQL queries, and retrieve results.

2. **CarManagerDB:** Software being developed by the team to enable a car dealership to perform the above-specified operations.

3. **SQL (Structured Query Language)** is a standard programming language used to manage and manipulate relational databases. The language used here is MySQL.