Software Requirement specification

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

Hardware Requirements

1) **QR Scanners:**
    - High-resolution document scanners with QR capabilities
- **Reliable High-speed Ethernet:**
    - For reliable database connectivity with MongoDB at all times

Software Requirements

1. Relational Database Management System (RDBMS): MySQL
Store structured data related to the relations for employees, customers, cars, department, sale, etc. maintaining relationships between entities while ensuring data integrity & consistency while retrieval & update operations on the database.

2. NoSQL Database: MongoDB
Generated ID card images for customers & employees and customer feedback will be stored the online non-relational database – MongoDB in the form of a document-oriented structure using JSON-like documents, ensuring high performance, scalability, and support for flexible querying and indexing.

3. QR Software: qrcode
The unique ID generated for each employee and customer will be embedded in the QR code present in the ID card image.

4. Web Development Framework: Flask
The main heavyweight of the Prototype, Flask (microframework in Python) will be used to create CRUD API Endpoints, handle user interactions and facilitate overall rendering of webpages and data transfer.

5. Frontend Framework: Tailwind CSS
Designed to style the user interface using utility classes for rapid development, ensuring a consistent and customizable look and feel throughout the various sections and components of the website.

Functional Requirements

1) **Employee & Customer Management using QR ID cards**
   - Customers should be able to log in with a unique username and password and track their purchase history, test drives, appointments, etc.
   - Employees will be able to see their performance, sales made for a particular car, appointments in which they must partake, etc.
   - Perform CRUD (Create, Read, Update, Delete) operations on various relations with concurrency control.

2) **Communication channels (email, SMS) integration**
   - Reminders regarding date of sale, appointments & test drives will be pushed to the customers via Meta API to alert on WhatsApp and SMTP protocol to inform on the registered email-id.

3) **Feedback System**
   - Allow customers to provide feedback on the system.
   - Admins should be able to view and analyse feedback for continuous improvement.

4) **Car Sale Records**
   - Record details of car sales transactions, including customer, employee, and car details.
   - Retrieve and display sales records based on various criteria.

5) **Web API Endpoints**
   - Implement CRUD API endpoints using Flask for all major entities (employees, customers, sales records).
   - Secure API endpoints to prevent unauthorized access.