

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.

Functional Requirements

- **Employee Management:**
 - User registration and management
 - Role-based access control
 - Performance evaluation
- **Customer Management:**
 - Customer registration with OCR integration for ID card scanning.
 - Appointment scheduling and reminder system
 - Test drive management
 - Purchase history tracking
 - Communication channels (email, SMS)
- **Car Management:**
 - Detailed car information including specifications, features, images, and videos.
 - Inventory tracking and management
 - Maintenance scheduling and service history
 - Sales analysis and reporting
- **Other Features:**
 - Dashboard for real-time data visualization
 - Reporting and analytics

Software Requirements

1. Relational Database Management System (RDBMS): MySQL

- **Purpose:**
 - Stores structured data, mainly for employee, customer, and car information.
 - Maintains relationships between different data entities.
 - Ensures data integrity, consistency, and security.
- **Features:**
 - Widely used, open-source, and community-supported.
 - Handles complex queries efficiently.
 - Supports ACID (Atomicity, Consistency, Isolation, Durability) properties for reliable transactions.

2. NoSQL Database: MongoDB

- **Purpose:**
 - Stores unstructured and semi-structured data, such as images, OCR-processed text, and customer feedback.
 - Provides flexibility in data schema and scalability for large datasets.
- **Features:**
 - Document-oriented structure for storing data in JSON-like documents.
 - High performance and scalability.
 - Supports flexible querying and indexing.

3. OCR Software: EasyOCR

- **Purpose:**
 - Extracts text from images and documents for features like employee ID card scanning and customer registration.
- **Features:**
 - Lightweight and easy to integrate with Python applications.
 - Supports multiple languages and text detection.
 - Can handle various image formats and document types.

4. Web Development Framework: Flask

- **Purpose:**
 - Creates the server-side logic and routes for the web application.
 - Handles user interactions, database connections, and API endpoints.
- **Features:**
 - Lightweight and flexible microframework.
 - Easy to learn and use for small to medium-sized projects.
 - Offers good performance and scalability.

5. Frontend Framework: Tailwind CSS

- **Purpose:**
 - Styles the user interface with utility classes for rapid development.
 - Provides a consistent and customizable look and feel.
- **Features:**
 - No pre-designed components, allowing complete control over the UI.
 - Highly customizable with a large collection of utility classes.
 - Promotes clean and maintainable code.

Hardware Requirements

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