**SOFTWARE REQUIREMENTS SPECIFICATION**

**for**

**Inventory Management System**

**PREPARED BY**

ABINAYA C (23CSR009)

ARVIND R(23CSR023)

ATHISH M (23CSR029)

KONGU ENGINEERING COLLEGE

27/02/2025

**Table of Contents**

**Table of Contents ............................................................ ii  
Revision History ............................................................ 1**

**1. Introduction**1.1 Purpose .............................................................................. 1  
1.2 Scope ................................................................................ 1  
1.3 Definitions, Acronyms, and Abbreviations .................. 1  
1.4 References ...................................................................... 1  
1.5 Overview .......................................................................... 1

**2. Overall Description**2.1 Product Perspective ....................................................... 2  
2.2 Product Functions ........................................................... 2  
2.3 User Characteristics ...................................................... 3  
2.4 Constraints .................................................................... 3

**3. Specific Requirements  
3.1 Functional Requirements .............................................. 3**

* User Authentication ......................................................... 3
* Property Listings ............................................................... 4
* Search & Filtering ............................................................ 4
* Appointment Scheduling .............................................. 4
* Messaging System ......................................................... 5
* Interactive UI/UX ............................................................ 5  
  **3.2 Non-Functional Requirements ................................... 5**
* Performance ................................................................... 5
* Security ........................................................................... 5
* Scalability ....................................................................... 6
* Usability .......................................................................... 6

**4. System Design & Architecture**4.1 Tech Stack ...................................................................... 6  
4.2 System Architecture ...................................................... 6

**5. Future Enhancements.................................................................. 7**

**1. Introduction**

**1.1 Purpose**

This document outlines the software requirements for the Inventory Management System, designed to provide small shops with an easy-to-use platform to track their inventory and sales performance. The system will include features such as product management, sales tracking, and a rewarding system for salespeople based on their performance.

This document serves as a reference for stakeholders, developers, and testers to understand the functional and non-functional requirements of the system.

**1.2 Scope**

The Inventory Management System will provide the following key features:

* **Product Management**: Add, edit, and delete products.
* **Salesperson Tracking**: Track sales performance and reward points for sales staff.
* **Sales Dashboard**: A visualization of sales data by product, region, and highest-performing salespeople.
* **Data Storage**: Use MongoDB or JSON files for product and sales data storage.

The system will be built using **Node.js** for the server-side and **ReactJS** for the frontend.

**1.3 Definitions, Acronyms, and Abbreviations**

| **Term** | **Definition** |
| --- | --- |
| CRUD | Create, Read, Update, Delete |
| API | Application Programming Interface |
| JWT | JSON Web Token (authentication standard) |
| MERN | MongoDB, Express.js, React.js, Node.js (tech stack) |
| UI/UX | User Interface / User Experience |

**1.4 References**

* MongoDB Documentation
* Node.js Documentation
* ReactJS Documentation
* OWASP Web Security Guidelines

**1.5 Overview**

This document provides the following structure:

* **Section 2**: Overview of the system, product functions, user characteristics, and constraints.
* **Section 3**: Detailed functional and non-functional requirements.
* **Section 4**: Tech stack, system architecture, and design considerations.
* **Section 5**: Future enhancements.

**2. Overall Description**

**2.1 Product Perspective**

The **Inventory Management System** is a full-stack web application designed to manage inventory, track sales, and incentivize salespeople. The components include:

* **Frontend**: Developed using **ReactJS** for a responsive, dynamic user interface.
* **Backend**: Powered by **Node.js** and **Express.js** for managing the business logic and handling API requests.
* **Database**: **MongoDB** (NoSQL) for storing product and sales data.
* **Authentication**: **JWT-based** authentication for secure user sessions.

**2.2 Product Functions**

The **Inventory Management System** will provide the following core functionalities:

1. **Product Management**:
   * Sellers can add, edit, and delete products, including price, description, and stock levels.
   * Products will be listed and displayed on the frontend with the necessary details.
2. **Sales Tracking**:
   * Sales made by salespeople will be tracked, including product sold and quantity.
   * The sales performance will be stored for calculating reward points.
3. **Reward System**:
   * Salespeople will accumulate reward points based on the sales they generate.
   * These points can be displayed and tracked by users on a dedicated dashboard.
4. **Sales Dashboard**:
   * The system will provide visualization tools, including charts and graphs, to display sales data by product, region, and salesperson.
5. **User Roles**:
   * Admins can manage users and have full access to the system, including product management and sales data.
   * Salespersons can track their sales and reward points.

**2.3 User Characteristics**

* **Admin**:
  + Full access to system features, including adding and editing products, managing users, and viewing reports.
* **Salesperson**:
  + Users who sell products and track their sales performance and reward points.
* **Shop Manager**:
  + Manages inventory, products, and user roles.

**2.4 Constraints**

* **Authentication**: Only authorized users (Admin or Salesperson) can perform certain actions such as adding products or viewing reward points.
* **Real-Time Updates**: The inventory must reflect sales in real-time, updating product stock levels accordingly.
* **Scalability**: The system must be scalable to handle an increasing number of products, sales, and users.

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Product Management**

* Admins should be able to add, edit, and delete products, including product details like name, price, and stock level.
* Product listings will be available for all users to view.

**3.1.2 Sales Tracking**

* The system will track each sale made by a salesperson, including product sold and quantity.
* The data will be stored in the backend (MongoDB).

**3.1.3 Reward System**

* Salespeople will earn reward points based on the value of the sales they make.
* Points will be accumulated and displayed on a dedicated dashboard.

**3.1.4 Sales Dashboard**

* Admins can visualize sales data, including sales per product, region, and salesperson.
* Charts and graphs will be used to display performance and trends.

**3.1.5 User Authentication**

* The system will use JWT for secure authentication.
* Admins and salespeople will have different access rights to specific features**.**

**3.2 Non-Functional Requirements**

**3.2.1 Performance**

* The system should load pages within 3 seconds.
* API response times should not exceed 500ms.

**3.2.2 Security**

* The system will implement secure JWT authentication and data encryption for

sensitive information.

**3.2.3 Scalability**

* The system must be designed to scale to handle increasing product data, sales, and

user load.

**3.2.4 Usability**

* The UI must be easy to navigate, providing clear functionality to add products, track

sales, and view reward points.

**4. System Design & Architecture**

**4.1 Tech Stack**

* -**Frontend**: React.js (with Material UI or Bootstrap for styling)
* **Backend**: Node.js with Express.js for API handling
* **Database**: MongoDB (NoSQL)
* **Authentication**: JWT for secure user sessions
* **Hosting**:
  + Frontend: Vercel/Netlify
  + Backend: AWS/DigitalOcean/Heroku

**4.2 System Architecture**

* **Frontend (React.js)**: Handles the UI and communicates with the backend using APIs.
* **Backend (Node.js + Express.js)**: Manages business logic, authentication, and API endpoints.
* **Database (MongoDB)**: Stores data such as products and sales.
* **REST API Layer**: Provides secure data exchange between frontend and backend.

**5. Future Enhancements**

* **Mobile App**: A cross-platform mobile app for iOS and Android.
* **Advanced Reporting**: AI-driven insights for better decision-making.
* **Payment Integration**: Integration with online payment systems for easy transactions.

This detailed **SRS** document lays out the functional and non-functional requirements for the **Inventory Management System**.