

iv) SRS for Stock Maintenance System.

1. Introduction:-

1.1 Purpose \Rightarrow The purpose of this document is to outline the requirements for developing a Stock maintenance System (SMS) that efficiently manages inventory & stock levels for businesses. The system automates stock tracking, order management etc. while providing detailed reports.

1.2 Scope \Rightarrow The scope of this system includes functionalities such as stock tracking, order management, supplier management, and reporting along with user management to control system access. System ensures real-time updates on inventory.

2. General Description:-

2.1 Product Perspective \Rightarrow SMS is a web-based application that interfaces with a relational database to manage stock data. It allows users to maintain inventory levels, manage suppliers and track incoming & outgoing orders.

2.2 Product Functions

- * Stock tracking
- * Order Management
- * Supplier Management
- * Reporting

2.3 User Characteristics

- * Warehouse Staff
- * Inventory Managers
- * Administrators

3. Functional Requirements

- * User Management: create, update & manage account
- * Stock Management: Add, update & remove stock items
- * Order Processing: Facilitate creation & tracking of stock orders
- * Reporting: Generate detailed reports

4. Interface Requirements

- * User Interface: Feature responsive & user-friendly design
- * Hardware Interface: compatible with standard server & client devices
- * Software Interface: System interact with a relational database

5. Performance Requirements

- * Concurrent Users: System must support a large number of simultaneous users

- * Page Load Times should be under 3 seconds.
- * Efficiently handle more than 10,000 stock items at once.

6. Design Constraints:-

- * Ensure Modular architecture to ensure scalability & easy maintenance.
- * The system will be developed using robust and widely supported programming languages.

7. Non-Functional Requirements:-

- * All user data & system interactions must be encrypted using SSL/TLS protocols.
- * User interface should be intuitive & require minimal training.
- * Must have high up-time.

8. Preliminary Schedule and Budget:-

8.1 Schedule =>

Requirement gathering: 2 weeks

System Design: 3 weeks

Development: 8 weeks

Testing: 4 weeks

Deployment: 2 weeks

Training: 1 week

Total: 20 weeks

8.2 Budget \$

Estimated Total: \$70,000 for labour, Training etc.