04) Stock Maintenance System

Problem statement: Design and implement an SRS document for a stock maintenance System.

SRS Document

1. Introduction

1.1 Purpose of Document:

This document specifies the requirements for the Stock Maintenance System (SMS). It outlines the scope, features, and constraints to help organizations track, manage, and update inventory levels efficiently.

1.2 Scope of Document:

The SMS will automate stock records, real time updates of quantities, generation of purchase/reorder alerts, and reporting of stock usage. It will minimize manual errors, prevent stockouts and ensure transparency in stock control.

1.3 Overview:

The System will:
- record stock inwards and outwards
- Track available quantities of each item.
- Generate alerts for low stock levels
- Provide sales and consumption reports
- Support user roles.

2. General Description

The SMS functions as a centralized inventory system. it enables businesses to keep accurate stock records.

Users include:
- Admin: Full control over System, user management
- Stock Manager: Add/Update stock, set reorder level, approve request
- Staff: View Stock, request items, update usage.

3. Functional Requirements

FR1: Add, Edit, delete stock records
FR2: Update stock levels upon purchase, sales of consumption
FR3: Generate automatic alerts for low/reorder stock,
FR4: Maintain supplies info
FR5: Provide daily, weekly and monthly stock reports.
FR6: provide barcode/RFID integration for tracking.

4. Interface Requirements:

- User interface: web dashboard with login and role-based view
- External interface: Barcode/RFID scanners, POS system integration
- API integration: REST APIs for integration with accounting

5) Performance Requirement:

- Handle 500 concurrent users
- Support up to 1 million stock records
- Average response time < 3 seconds for queries and updates

6) Design Constraints:

- Database must be relational
- Must run on standard window/linux server
- Should comply accounting inventory standards.

7) Non Functional Attributes

- Security : Uses authentication, verification of transaction
- Reliability : Redundant data for failovers
- Usability : Simple UI for intuitive usage
- Maintainability : Modular codebase with easy to update code and documented APIs

8) Preliminary Schedule and Budget :

Schedule:

    Requirement Analysis   1 week
    System design   - 2 weeks
    Development   6 weeks
    Testing   2 weeks
    Deployment & testing   - 1 week
    Total ≈ 12 weeks

Budget :

    Development costs   - ₹12,60,000
    Hardware /infrastructure - ₹4,00,000
    Security / compliance   - ₹2,00,000
    Testing & Maintenance   - ₹5,00,000
    Total = ₹23,00,000