

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the Stock Maintenance System (SMS). This system is designed to provide comprehensive control over inventory, tracking stock levels, managing product details, and facilitating efficient stock-related operations. This SRS serves as a foundational blueprint for developers, project managers, and all stakeholders, ensuring a shared understanding of the system's objectives and specifications.

1.2 Scope

This SRS addresses the core functionalities required for effective stock management, including item cataloging, inventory tracking, transaction recording, and reporting. The scope covers the management of a single or multiple warehouses and does not include advanced features such as supply chain management, demand forecasting, or complex vendor relationship management.

1.3 Overview

The SMS will be a secure, centralized, and web-based application accessible to warehouse staff and management. It will provide a digital solution to replace manual or disparate inventory management methods, thereby reducing errors, improving data accuracy, and providing real-time insights into stock levels.

2. GENERAL DESCRIPTION

The SMS is intended for use by two primary user classes: warehouse staff and management. warehouse staff will have access to the system to record incoming and outgoing stock, perform stock takes, and update item information. Management will have a higher level of access to view reports, monitor inventory, and configure system settings. The system will operate within a secure, network-based environment and can be scaled to support multiple users across different locations.

3. FUNCTIONAL REQUIREMENTS

The system shall perform the following core functions:

- Product catalog: Provide an interface to add, modify, and delete product items. Each product entry must include a unique SKU, description, unit of measure, and other relevant details.
- Inventory tracking: The system shall track real-time stock levels for each product and location. It must automatically adjust inventory upon recording a new transaction.
- Transaction recording: Allow staff to record all stock-related transactions, including receipts from suppliers, sales, and internal transfers. Each transaction must be timestamped and linked to a specific user.

- Stock Taking: Provide a streamlined process for performing a physical count of inventory to reconcile with the system's recorded stock. The system shall log discrepancies for further investigation.
- Alerts and Notifications: Automatically generate alerts with when a product's stock level falls below a predefined minimum threshold.
- Reporting: Generate comprehensive reports for management on inventory valuation, stock turnover rates, and transaction history. Reports should be filterable by date, product, or location.

4. INTERFACE REQUIREMENTS

- User Interface: The system shall have a responsive, web-based UI optimized for both desktop and mobile devices. The interface must be intuitive, with a clear separation between staff and management dashboards.
- Hardware Interface: The system must support integration with standard barcode scanners for quick and accurate recording of product items during stock receipts and sales.
- Software Interface: The system shall include an API for potential future integration with e-commerce platforms or enterprise resource planning (ERP) systems. All external communication will be secured.

5. PERFORMANCE REQUIREMENTS

- Response time: All stock-related queries and transaction recordings shall complete within 2 seconds.
- Concurrency: The system shall be capable of handling a minimum of 30 concurrent users without significant performance degradation.
- Availability: The SMS must maintain an uptime of 99.8% to ensure continuous operational functionality.

6. DESIGN CONSTRAINTS

- Technology stack: The system shall be developed using a robust, scalable and secure technology stack. Preference is given to a modern web framework with a reliable backend and a relational database.
- Security Compliance: The system must adhere to industry best practices for data security, particularly for protecting proprietary product and transaction data.
- Scalability: The architecture must be designed to support a rapid increase in product volume and accommodate the addition of new warehouses or locations.

7. NON-FUNCTIONAL REQUIREMENTS

- Security: The system shall implement a role-based access control system to restrict user access based on their roles. All data, especially transaction and inventory information, shall be encrypted.
- Usability: The UI shall be intuitive, with minimal training required for new warehouse staff. Core workflows should be simple and efficient.
- Reliability: The system must have a robust error handling and logging mechanism. Automated daily backups of the database are required to ensure data integrity.
- Maintainability: The codebase must be modular, well-documented, and easily adoptable to new business requirements.

8. PRELIMINARY SCHEDULE AND BUDGET

- Schedule: The estimated development timeline for the core functionalities is 5-7 months, followed by a one-month testing and deployment phase.

- Budget: The preliminary budget for development, based on a team of 3-4 developers, is estimated to be in the range of \$40,000 to \$70,000, excluding ongoing testing and maintenance costs.

7