Warehouse Management System

1. Introduction

1.1 Purpose

The Warehouse Management System (WMS) provides functionalities to manage a warehouse's inventory, suppliers, stock requests, and product history. This system enables users to add and update product and supplier information, request stock from suppliers, and view historical changes in product data.

1.2 Scope

The WMS is a console-based Java application with MySQL integration that allows warehouse managers to perform essential inventory-related tasks, including product management, supplier management, stock updates, and viewing transaction histories. This SRS outlines the requirements, functionalities, and design principles for the WMS.

1.3 Definitions, Acronyms, and Abbreviations

- **WMS**: Warehouse Management System
- **JDBC**: Java Database Connectivity
- **MySQL**: Relational database management system used for data storage

1.4 References

- Java SE 8 Documentation
- MySQL 8.0 Documentation

---

2. Overall Description

2.1 Product Perspective

The WMS is a stand-alone system that interfaces with a MySQL database to store and retrieve warehouse data. It is designed to operate as a console application, allowing users to interact with the database using a text-based menu system.

2.2 Product Functions

- **Add and view products**: Users can add new products and view details of all products in the warehouse.
- **Add and view suppliers**: Users can manage supplier information and retrieve supplier details.

- **Stock updates**: Users can update the quantity of existing products in the warehouse.
- **Stock requests**: Users can add requests for product stock from suppliers and track these requests.
- **View product history**: Users can view the change history for products, including type of change, quantity, and date of the change.

2.3 User Classes and Characteristics

- **Warehouse Manager**: Responsible for managing inventory, stock requests, and suppliers. Basic knowledge of console applications and database management is assumed.

2.4 Operating Environment

- **Java SE 8 or higher**
- **MySQL 8.0 or higher**
- **Operating System**: Any OS that supports Java and MySQL (e.g., Windows, Linux, macOS)

---

3. Functional Requirements

3.1 User Interface

- Display a menu with options to manage products, suppliers, stock, stock requests, and product history.
- Prompt for user inputs based on selected actions.

3.2 Functional Requirements

3.2.1 Product Management

- **Add Product**: The system shall allow the user to add new products to the database.
    - Inputs: Product name, type, quantity, price, and storage requirements.
    - Outputs: Success or failure message after insertion.
- **View Products**: The system shall display all product information from the database.

3.2.2 Supplier Management

- **Add Supplier**: The system shall allow users to add new suppliers to the database.
    - Inputs: Supplier name, contact name, contact number, and address.
    - Outputs: Success or failure message after insertion.
- **View Suppliers**: The system shall display all supplier information from the database.

3.2.3 Stock Update

- **Update Stock**: The system shall allow users to update the quantity of an existing product in the warehouse.
    - Inputs: Product ID and new quantity.
    - Outputs: Success or failure message after updating.

3.2.4 Stock Request

- **Add Stock Request**: The system shall allow users to add a request for product stock from a supplier.
    - Inputs: Product ID, supplier ID, request quantity, request date, and status.
    - Outputs: Success or failure message after insertion.
- **View Stock Requests**: The system shall display all stock request information from the database.

3.2.5 Product History

- **View Product History**: The system shall allow users to view the change history for a specific product.
    - Inputs: Product ID.
    - Outputs: List of change events with details such as history ID, product ID, change type, quantity change, change date, and notes.

---

4. Non-functional Requirements

4.1 Performance Requirements

- **Database Performance**: The system should retrieve and display data in less than 2 seconds for typical database queries.

4.2 Reliability

- The system shall handle database connectivity exceptions and display appropriate error messages without terminating unexpectedly.

4.3 Usability

- The console interface shall be simple and provide clear prompts to the user for inputs.
- Error messages should inform the user of incorrect inputs or failed database operations.

4.4 Portability

- The system shall be portable across any operating system that supports Java and MySQL.

4.5 Maintainability

- Code should be organized into methods within classes to facilitate future updates and maintenance.

---

5. System Models

5.1 Class Diagram

- **WarehouseManagement**: Main class that displays the menu and manages user interactions.
- **Database Operations**: Methods to add, view, and update records for products, suppliers, stock requests, and product history.

5.2 Database Schema

- **Products**: Stores product information, including ID, name, type, quantity, price, and storage requirements.
- **Suppliers**: Stores supplier information, including ID, name, contact details, and address.
- **StockRequests**: Stores stock request details, including request ID, product ID, supplier ID, quantity, date, and status.
- **ProductHistory**: Logs historical changes for products, including history ID, product ID, change type, quantity change, date, and notes.

---

6. Data Requirements

6.1 Database Connection

- **URL**: jdbc:mysql://localhost:3306/final
- **Username**: root
- **Password**: 0110

6.2 Data Validation

- Integer fields such as product ID, supplier ID, and quantity shall be validated to prevent null or invalid entries.
- Ensure request_date follows the format yyyy-mm-dd hh:mm:ss.

---

7. Assumptions and Dependencies

- The system assumes the user has basic knowledge of Java console operations.
- MySQL should be installed and configured properly on the user's machine with the necessary tables created.
- Java JDBC drivers should be available in the project library for database connectivity.

---

8. Appendix

- **Database Table Creation Scripts**: Provide the scripts to create Products, Suppliers, StockRequests, and ProductHistory tables.
- **Console Commands**: Sample commands to run the application and interact with different features.