



Software Requirements Specification (SRS) Document

Project Name: Spices Management System

Date: 2025/11/25

Student ID: 23DA2-0268

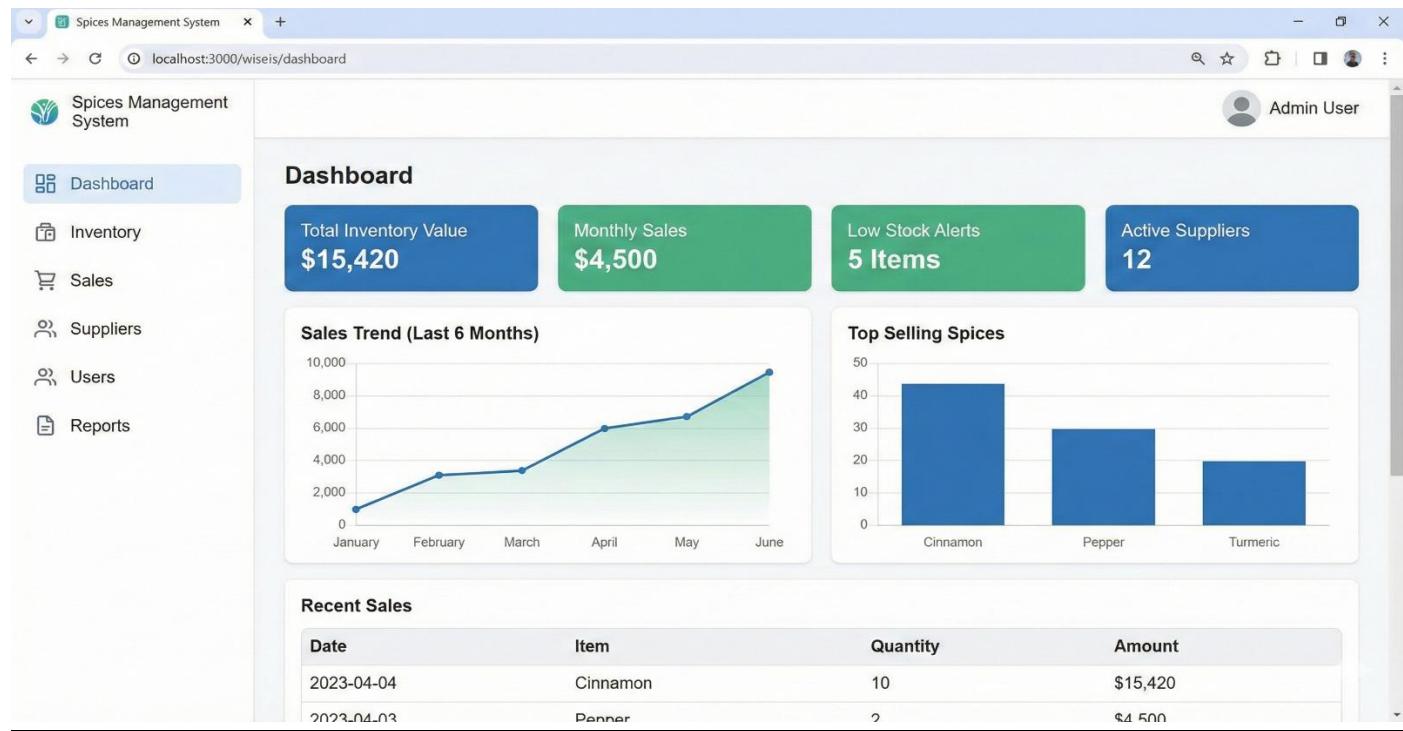
Prepared By: Kavindu Dananjaya

Contents

	Page
1. Introduction	
1.1 Purpose	03
1.2 Scope (products Scope).....	03
1.3 Definitions, Acronyms, and abbreviations	04
1.4 Reference	05
1.5 Document Overview	06
2. Overall Description	
2.1 Product Perspective	06
2.2 Product Function.....	07
2.3 User Characteristics	08
2.4 General Constraints.....	09
2.5 Assumptions and dependencies	09
3. Specific Requirements	
3.1 External Interface Requirements	10
3.1.1 User Interface (UI).....	10
3.1.2 Hardware Interfaces.....	15
3.1.3 Software Interface.....	16

3.1.4 Communication Interface	17
3.2 Functional Requirements	18
3.3 Non-Functional Requirements (NFRs)	21
3.3.1 Performance Requirements	21
3.3.2 Security Requirements	22
3.3.3 Reliability Requirements	22
3.3.4 Maintainability Requirements	22
3.3.5 Portability Requirements	23
4. Supporting Information	
4.1 Appendix A: Data Model and Class Diagram	23
4.2 Appendix B: Use Case and System Diagrams	27

Spices Management System



Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document defines the functional and non-functional requirements for the **Spices Management System (SMS)**. It details the system's intended functions, capabilities, constraints, and acceptance criteria to guide design, implementation, testing, deployment, and maintenance.

1.2 Scope (Product Scope)

Product name: Spices Management System (SMS)

Product description (high level):

SMS is a web-based application to manage spice product definitions, batch-level traceability, warehouse stock, procurement (PO) and receiving (GRN), quality control (QC) testing, sales shipments, and reporting. The system will centralize inventory data across multiple warehouses and support role-based access for operational users.

Included Features

- Product catalog and batch registration (batch attributes, barcode/QR assignment).
- Warehouse inventory management (inbound GRN, outbound shipments, transfers, stock counts).
- Purchase order lifecycle (create PO, receive goods, match GRN to PO).
- Quality Control module (QC templates, record results, attach lab reports, quarantine failed batches).
- Sales order creation, stock reservation, packing lists, and basic invoice generation.
- Traceability (view full history for a batch: supplier, PO, QC, movements, shipments).
- Dashboards and reports (stock levels, expiring batches, low-stock alerts, QC reports).
- Role-based access control and audit logs for critical actions.
- File storage for scanned certificates and lab reports (S3 or equivalent).

Excluded Features (out of scope for this phase / MVP)

- Full financial accounting / general ledger integration (may be integrated later).
- Direct B2C e-commerce storefront or public online sales channel.
- Integration with external shipping carriers (label booking, tracking APIs) in the MVP.
- Advanced forecasting or machine learning demand prediction.
- On-device label printing drivers (basic PDF/label export provided; direct printer drivers handled later).

1.3 Definitions, Acronyms, and abbreviations

This section lists all technical terms and abbreviations used throughout the SRS document.

Term / Acronym	Definition
SRS	Software Requirements Specification – a document describing system requirements.
API	Application Programming Interface – a set of protocols that allow systems to communicate.
UI	User Interface – the visual elements through which users interact with the system.
CRUD	Create, Read, Update, Delete – basic database operation types.
DB	Database – structured data storage system.
QC	Quality Control – processes used to ensure product quality standards.

GUI	Graphical User Interface – visual elements like buttons, forms, and menus.
HTTPS	Hypertext Transfer Protocol Secure – secure communication protocol for web systems.
CSV	Comma-Separated Values – a file format used to store tabular data.
ERP	Enterprise Resource Planning – business process management software.
JSON	JavaScript Object Notation – a lightweight data exchange format.
REST	Representational State Transfer – a style of web API architecture.
SKU	Stock Keeping Unit – unique ID used for product identification.
RAM	Random Access Memory – hardware memory used by systems during execution.
UX	User Experience – overall feel and interaction quality of the system.

1.4 Reference

This section lists all documents, standards, and resources referenced during the preparation of the SRS.

1. **Project Proposal Document** – Initial outline and problem definition for the Spices Management System.
2. **Vision Document** – High-level goals, scope, and stakeholders of the project.
3. **IEEE 830 Standard** – Recommended guidelines for writing Software Requirements Specifications.
4. **Industry Articles on Inventory and Supply Chain Management** – Various sources describing modern best practices.
5. **Client Requirements Notes / Meeting Summaries** – Collected during requirement-gathering sessions.
6. **Academic Resources** – Related materials used from software engineering and system analysis courses.
7. **Web Technology Standards** – HTML5, CSS3, JavaScript, REST API guidelines.

1.5 Document Overview

This document is organized into several sections that describe the full requirements of the Spices Management System:

- **Section 1 – Introduction:**
Provides the purpose, scope, definitions, and references used throughout the SRS.
- **Section 2 – Overall Description:**
Describes the high-level system environment, product perspective, main functions, user characteristics, constraints, and assumptions.
- **Section 3 – Specific Requirements:**
Contains detailed functional requirements, external interface requirements, system features, and non-functional requirements.
This section provides enough detail for designers and developers to build the system.
- **Section 4 – Appendices (if applicable):**
Additional supporting information such as diagrams, sample forms, or data models.

2.Overall Description

2.1 Product Perspective

The Spices Management System (SMS) is designed to serve as a centralized digital platform for managing all spice-related business operations, including inventory control, supplier management, sales tracking, quality monitoring, and reporting. The system replaces traditional manual processes and enhances accuracy, speed, and transparency within the company.

System Interface

- This is a **new, integrated web-based system**.
- It can also integrate with:
 - **Barcode/QR scanning devices** for inventory verification.
 - **Accounting systems** (future scope) such as QuickBooks or Tally.
 - **Email/SMS gateway** for sending alerts and notifications.

Hardware Requirements

- Standard desktop or laptop (minimum **4GB RAM**, dual-core processor).
- Mobile or tablet devices for warehouse staff (Android/iOS compatible).

Software Requirements

- Operating Systems: Windows, Linux, macOS.
- Browsers: Chrome, Firefox, Safari, Edge (latest versions).
- Backend: PHP/Laravel or Node.js.
- Frontend: React/HTML/CSS/JavaScript.
- Database: MySQL/PostgreSQL.

2.2 Product Function

The major high-level functions of the Spices Management System include:

1. User Authentication & Access Control

- Login/Logout
- Role-based permissions (Admin, Staff, Manager)

2. Inventory Management

- Add, update, delete spices
- Stock level monitoring
- Batch tracking and expiry management

3. Supplier & Customer Management

- Manage supplier details
- Track customer orders and profiles

4. Purchase & Sales Management

- Record purchase orders
- Track sales orders and invoices

5. Quality Control Monitoring

- Record quality test results
- Maintain batch quality history

6. Warehouse & Packaging Management

- Track packaging units
- Storage location mapping

7. Reporting & Analytics

- Inventory summary reports
- Monthly purchase/sales reports
- Forecasting and trend analysis

8. Alerts & Notifications

- Low stock alerts
- Expiry reminders
- Pending order alerts

9. Backup & Data Recovery

- Scheduled backups
- Restore previous data versions

2.3 User Characteristics

User Type	Technical Skill Level	Typical Tasks Performed
Admin	High	User management, system configuration, data backup, full access to reports and dashboards
Manager	Medium	Approve orders, view analytics, oversee inventory and quality reports
Staff/User	Medium	Data entry, updating inventory, recording sales/purchases, performing daily operational tasks
External Client (Optional)	Low	View available products, place basic orders, check status (if system supports client portal)

2.4 General Constraints

Regulatory/Legal Constraints

- Must comply with general data protection guidelines (similar to GDPR).
- Must follow food safety record-keeping requirements for spice production traceability.

Hardware Limitations

- System should operate smoothly on devices with at least **4GB RAM**.
- Warehouse devices may have limited processing power; therefore, UI must be lightweight.

Software Constraints

- Must be developed using **PHP/Laravel** or **Node.js + React** (based on company preference).
- Database must be relational (MySQL/PostgreSQL).
- System must support multilingual interfaces (English + local languages).

Time Constraints

- Version 1.0 must be completed and deployed within **4 months**.
- User training must be completed within **two weeks** after deployment.

Security Constraints

- Role-based access control must be enforced.
- Encrypted passwords and secure communication (HTTPS) must be used.

2.5 Assumptions and dependencies

Assumptions

- The company will provide all initial legacy data (inventory, suppliers, sales) in **CSV or Excel** format.
- Users will have basic computer knowledge.
- The organization has a stable internet connection for uninterrupted web system access.

- All business rules and workflows will be provided by the client during the requirements phase.

Dependencies

- System reporting features depend on the completion of the **database schema and data migration**.
- Real-time notifications depend on integration with **Email/SMS API services**.
- Barcode scanning features depend on **compatible hardware availability**.
- Sales forecasting module depends on the successful implementation of historical data import.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface (UI)

This section specifies the required interface screens, layout behavior, and user-experience requirements for the Spices Management System.

A. General Interface Requirements

1. Dashboard View

- Upon successful login, the system **must display a dashboard** containing:
 - Total inventory count
 - Low-stock alerts
 - Today's sales/purchase summary
 - Pending orders
 - Quick links to major modules

2. Responsive Design

- The UI **must be fully responsive**, adjusting properly to:
 - Desktop screens
 - Tablets

- Mobile devices

3. Theme Support

- The interface **must support both Light Mode and Dark Mode.**
- Users should be able to switch themes from the settings panel.

4. Navigation Menu

- A left-side collapsible navigation menu must be provided.
- Must include modules:
 - Dashboard
 - Inventory Management
 - Suppliers
 - Customers
 - Sales & Purchases
 - Quality Control
 - Reports
 - Users & Roles
 - Settings

5. Standard Layout

- Every page must include:
 - Page title/header
 - Search bar (where applicable)
 - Action buttons (Add, Edit, Delete, Export)
 - Table or form area
 - Footer with version information

B. Screen-Specific Requirements

1. Login Page

- Must include fields for:

- Username/Email
 - Password
- Must include:
 - “Forgot Password” link
 - “Login” button
- Must display error messages for invalid credentials.

2. Inventory Management Screen

- Must display a **table of all spice items**, including:
 - Spice name
 - Batch number
 - Quantity in stock
 - Unit price
 - Expiry date
 - Supplier
- Must include:
 - “Add New Item” form popup
 - Edit/Delete action buttons
 - Search and filtering options
 - Export buttons (CSV, Excel, PDF)

3. Supplier Management Screen

- Must display:
 - Supplier name
 - Contact details
 - Address
 - Category (Local/International)
- Must have forms for:

- Adding a supplier
- Editing supplier details

4. Sales & Purchase Screens

- Must allow entering:
 - Customer/Supplier name
 - Order details
 - Quantity
 - Price
 - Payment status
- Must generate an invoice or order summary.

5. Quality Control Screen

- Must allow recording:
 - Batch number
 - Moisture content
 - Color grade
 - Test results
 - QC status (Pass/Fail)
- Must display test history in table format.

6. Reporting Module

- Must provide:
 - Monthly inventory report
 - Sales summary report
 - Purchase summary report
 - Export options (PDF, Excel)
- Must provide graphical charts (bar, line, pie).

7. User Management Screen

- Must allow Admin to:
 - Add/Edit/Delete users
 - Assign roles and permissions
 - Reset passwords

C. Look-and-Feel Requirements

1. Color Scheme

- Must follow a clean, modern, professional theme.
- Suggested palette: Green (for spices), White, Black, Grey highlights.

2. Typography

- Use clear, readable fonts (e.g., Roboto, Inter, or similar).
- Minimum font size 12px for readability.

3. Tables and Forms

- Tables must support:
 - Sorting
 - Pagination
 - Column filtering
- Forms must provide:
 - Mandatory field indicators (*)
 - Inline validation messages

4. Icons and Graphics

- Use modern icons (Material Icons / FontAwesome).
- Icons must be consistent across all modules.

5. Accessibility

- UI must support:
 - Keyboard navigation
 - High contrast mode compatibility

- Screen reader friendly labels

D. Error and Notification Requirements

- System must display clear error messages:
 - Invalid input
 - Empty required field
 - Permission denied
- Notification alerts must appear for:
 - Successful operations (Saved, Updated, Deleted)
 - Low-stock warnings
 - Expired items
 - Failed operations

3.1.2 Hardware Interfaces

Client-Side Requirements (Users)

To access the system:

Hardware:

- Standard desktop or laptop computer
- Optional: Tablet/mobile device for on-site warehouse operations
- Barcode scanner (USB or wireless) optional

Supported Operating Systems:

- Windows 10/11
- Linux (Ubuntu)
- macOS

Supported Web Browsers:

- Google Chrome (recommended)
- Microsoft Edge

- Mozilla Firefox
- Safari (limited mobile compatibility only)

Network:

- Stable internet or local intranet connection
- Minimum speed requirement: 2 Mbps for standard operations

Server-Side Requirements

- **Application Server:**
Node.js / NestJS or equivalent backend framework
- **Frontend:**
React/Next.js web application
- **Database Server:**
PostgreSQL (recommended), MySQL, or SQL Server depending on deployment
- **Storage:**
Cloud storage (AWS S3 / MinIO) for storing documents and certificates
- **Deployment Environment:**
 - Cloud hosting (AWS/ECS/Lightsail), on-premise server, or Docker/Kubernetes-based setup
 - Minimum server specs (MVP):
 - 2 vCPU
 - 4 GB RAM
 - 40 GB storage

3.1.3 Software Interface

The Spices Management System (SMS) interacts with specific software components, services, libraries, and external tools to operate efficiently. These interfaces describe how the system communicates with internal modules or optional third-party APIs.

Internal Software Interfaces

- **Database Management System (DBMS)**

- The system must interact with a relational database (PostgreSQL/MySQL/SQL Server) for storing all product, batch, QC, and stock movement data.
 - The backend must communicate via standard SQL queries or ORM (e.g., TypeORM, Prisma).
- **File Storage Service**
 - The system must support integration with an object storage service (AWS S3 / MinIO) to upload and retrieve QC certificates, GRN copies, and batch documents.
- **Authentication Service**
 - SMS must support JWT-based authentication for secure user login.
 - Optional future integration with organization SSO (OAuth2/SAML).

Optional/External Software Interfaces (Future)

(These may not be implemented in the MVP but are supported by system architecture.)

- **ERP/FINAC System Integration**
 - Export or sync purchase orders, suppliers, and inventory adjustments using APIs or CSV imports.
- **Document/PDF Generator**
 - The system must use internal libraries such as **pdfkit**, **jsPDF**, or **ReportLab** to generate labels, reports, or packing slips.
- **Barcode/QR Code Generator**
 - The system must use a library (e.g., `qrcode`, `jsbarcode`) to generate batch-level barcodes/QR codes.
- **Analytics/Reporting Tools**
 - Optional integration with BI tools like Power BI through data exports.

3.1.4 Communication Interface

The communication interfaces define how data moves between the client, server, database, and external services. All communication must meet security and performance standards.

Client–Server Communication

- The system must use the **HTTPS protocol** for all client–server communication to ensure encrypted transmission of sensitive data.
- All API requests and responses must follow **RESTful standards** using JSON format.
- WebSockets may be used for optional real-time features (e.g., stock updates, live notifications).

Server–Database Communication

- The backend server communicates with the database using encrypted connections (e.g., TLS-enabled database connection).
- Database requests must use secure, parameterized queries or ORM methods to prevent SQL injection.

File Upload/Download Communication

- Files such as QC certificates and documents must be uploaded and retrieved via secure HTTPS endpoints.
- Large file uploads should use multipart/form-data format.

External Service Communication (Future Integration)

- ERP or third-party systems must be accessed through secure REST APIs or SFTP file transfers.
- Any third-party API must be accessed using API keys, OAuth2 tokens, or signed requests.

Error Handling Protocol

- All communication interfaces must return structured error messages with standardized HTTP status codes (e.g., 400, 401, 404, 500).
- The system must implement retry logic for network interruptions during API calls (optional).

3.2 Functional Requirements

ID	Function / User Story	Description	Priority	Status

FR-001	Create New User Account	As an Administrator, I want to create new user accounts by providing name, email, and assigning a role, so that new employees can access the system.	High	Approved
FR-002	View Inventory Stock	As a Standard User, I want to view a real-time list of all inventory items, so I can check availability before fulfilling an order.	High	Approved
FR-003	Generate PDF Report	As an Administrator, I want to generate a sales summary report for a specified date range in PDF format.	Medium	Approved

ID	Function / User Story	Description	Priority	Status
FR-004	Add New Spice Product	As an Admin, I want to create new spice items (SKU) with attributes like category, grade, and unit, so that the system stores accurate product data.	High	Approved
FR-005	Create Batch	As a Warehouse Officer, I want to register a new spice batch with batch number, weight, supplier, and date, so inventory can be tracked at batch level.	High	Approved
FR-006	Generate QR/Barcode	As a Warehouse Officer, I want the system to generate a QR or barcode for each batch, so scanning is easy during inbound and outbound processes.	High	Approved
FR-007	Goods Received (GRN)	As a Warehouse User, I want to record new stock arrivals with GRN details, so the system updates inventory correctly.	High	Approved
FR-008	Quality Control Entry	As a QC Officer, I want to enter moisture %, purity %, and attach lab reports, so that each batch has accurate quality data.	High	Approved
FR-009	Batch Approval / Rejection	As a QC Manager, I want to approve or reject batches after QC testing, so only approved stock moves to available inventory.	High	Approved
FR-010	Internal Stock Transfer	As a Warehouse Manager, I want to transfer stock between warehouses, so stock distribution is properly recorded.	Medium	Pending
FR-011	Create Purchase Order (PO)	As a Procurement Officer, I want to create POs with supplier details, items, quantity, and price, so procurement is tracked.	High	Approved
FR-012	Receive PO Against GRN	As a Warehouse User, I want to match an incoming GRN with a PO, so discrepancies can be flagged.	High	Approved

FR-013	Create Sales Order	As a Sales Officer, I want to create and reserve stock for a sales order, so quantities are locked until dispatch.	High	Pending
FR-014	Dispatch Order	As a Warehouse Officer, I want to outbound stock and generate delivery notes, so the shipment process is documented.	High	Approved
FR-015	View Batch Traceability	As any authorized user, I want to view full batch history (PO → GRN → QC → Stock Movements → Sales), so traceability is maintained.	High	Approved
FR-016	Role-Based Access Control	As an Admin, I want to assign user roles (Admin, Warehouse, QC, Sales), so users only access relevant modules.	High	Approved
FR-017	Generate Inventory Report	As a Manager, I want to download a real-time inventory report showing available stock per warehouse.	Medium	Approved
FR-018	Alert for Low Stock	As a Standard User, I want notifications when a product reaches a minimum stock level, so replenishment can occur.	Medium	Pending
FR-019	Export Data (CSV)	As an Admin, I want to export product, batch, or QC data as CSV for analysis.	Low	Pending
FR-020	Audit Log	As a System Admin, I want to track all critical actions (create, update, delete), so user activities are traceable.	High	Approved

3.3 Non-Functional Requirements (NFRs)

Non-functional requirements define the quality attributes and constraints of the Spices Management System.

3.3.1 Performance Requirements

- Load Time:** All system pages (Dashboard, Inventory, Orders) must load within **2 seconds** under peak load conditions of **100 concurrent users**.

- **Transaction Rate:** The system must be able to process **at least 50 inventory update transactions per minute** without performance degradation.
- **Scalability:** The system must support scaling up to **5000 total inventory items** and **unlimited user accounts** without requiring major architectural changes.
- **Response Time:** API endpoints must respond within **500ms** for standard CRUD operations.

3.3.2 Security Requirements

- The system must implement **multi-factor authentication (MFA)** for all administrative accounts.
- All sensitive data fields (passwords, user data, inventory prices) must be **encrypted using AES-256** encryption.
- All login and API data transfers must use **HTTPS (TLS 1.2 or later)**.
- Passwords must be stored using **bcrypt hashing** or stronger.
- Role-based Access Control (RBAC) must be enforced to prevent unauthorized access to admin-level pages.
- All user activities (login, inventory update, deletion) must be **logged for auditing**.

3.3.3 Reliability Requirements

- **Availability:** The system must be available **99.5% of the time**, excluding scheduled maintenance.
- **Backup:** Automatic database backup must occur **daily**, with a retention period of **30 days**.
- **Recovery:** In the event of system failure, the system must recover and restore the last saved state within **5 minutes**.
- **Fault Tolerance:** The system should continue operating even if one non-critical service becomes unavailable.

3.3.4 Maintainability Requirements

- The system codebase must follow industry standards:
 - **Backend:** PSR-12 (PHP) or appropriate framework coding standards
 - **Frontend:** ESLint/Prettier or standard React formatting rules

- The system must support **modular code structure** to allow easy updates and feature additions.
- Minor bug fixes should be deployable through **hot patches** without requiring a full restart.
- Comprehensive documentation must be provided for:
 - API Endpoints
 - Database schema
 - User/Admin manuals

3.3.5 Portability Requirements

- The application must run on **Windows, Linux, and macOS** operating systems.
- The system must support all modern browsers:
 - Google Chrome
 - Microsoft Edge
 - Mozilla Firefox
 - Safari
- The application should be deployable on cloud platforms like **AWS / Azure / DigitalOcean** without configuration issues.
- The system must be responsive and accessible on:
 - Desktop
 - Tablet
 - Mobile devices

4 Supporting Information

4.1 Appendix A: Data Model and Class Diagram

This appendix provides the high-level data structures, including the **database schema** and a description of the **UML Class Diagram** used in the Spices Management System.

1. Users Table

Field	Type	Description

user_id (PK)	INT	Unique ID for each user
name	VARCHAR	Full name of the user
email	VARCHAR	Login email (unique)
password	VARCHAR	Encrypted password
role	ENUM('Admin', 'User')	Defines access privileges
created_at	DATETIME	Timestamp of account creation

2. Inventory Table

Field	Type	Description
item_id (PK)	INT	Unique ID for each spice item
item_name	VARCHAR	Name of spice (e.g., Cinnamon Quills)
type	VARCHAR	Grade or category
quantity	INT	Current stock level
unit_price	DECIMAL	Price per unit
updated_at	DATETIME	Last modification timestamp

3. Suppliers Table

Field	Type	Description
supplier_id (PK)	INT	Unique supplier ID
supplier_name	VARCHAR	Name of supplying company
contact_number	VARCHAR	Phone number
address	TEXT	Supplier address

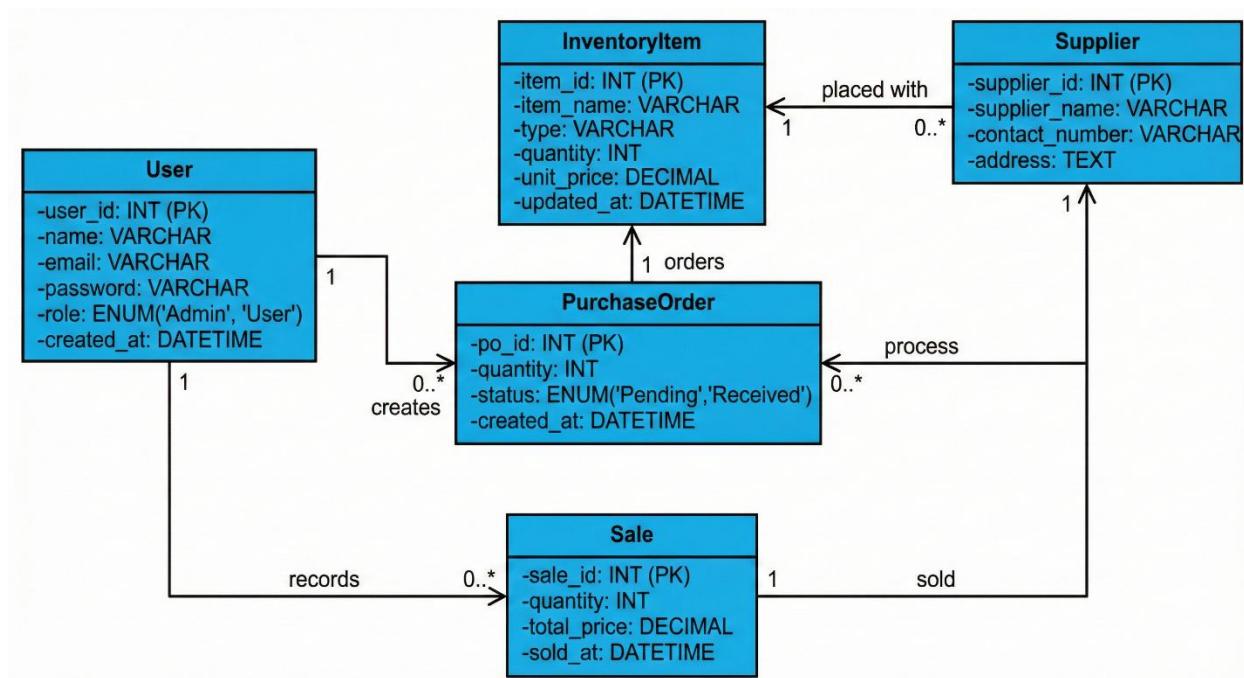
4. Purchase Orders Table

Field	Type	Description

po_id (PK)	INT	Purchase order ID
supplier_id (FK)	INT	References Suppliers table
item_id (FK)	INT	References Inventory table
quantity	INT	Quantity ordered
status	ENUM('Pending','Received')	PO status
created_at	DATETIME	Order creation date

5. Sales Table

Field	Type	Description
sale_id (PK)	INT	Unique sale record ID
item_id (FK)	INT	Spice sold
quantity	INT	Quantity sold
total_price	DECIMAL	Total amount
sold_at	DATETIME	Timestamp



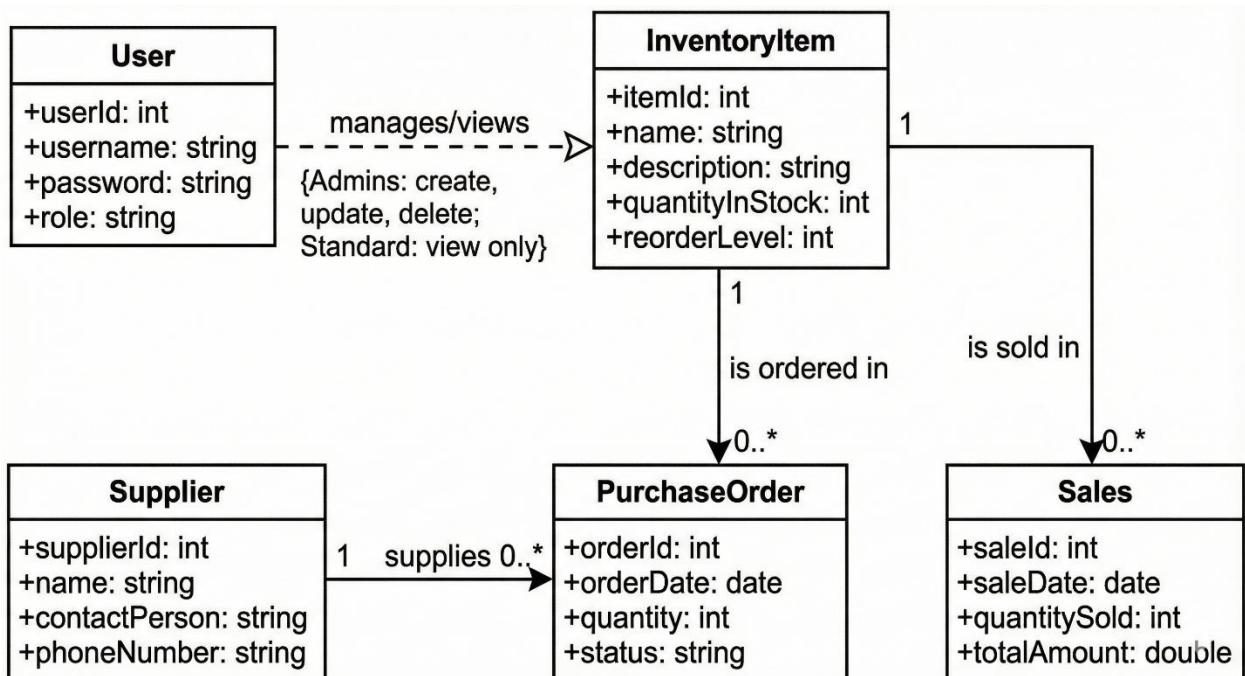
4.1.2 UML Class Diagram (Description)

The UML Class Diagram consists of the following classes:

- **User**
- **InventoryItem**
- **Supplier**
- **PurchaseOrder**
- **Sales**

Relationships:

- **User → Inventory Management:** Administrators can create, update, and delete items. Standard users can only view stock.
- **Supplier ↔ PurchaseOrder:** One supplier can have many purchase orders (*1-to-many*).
- **InventoryItem ↔ PurchaseOrder:** Each purchase order is linked to one spice item.
- **InventoryItem ↔ Sale:** Each sale record references one inventory item.



4.2 Appendix B: Use Case and System Diagrams

This appendix contains diagrams showing the interactions between users and the system, as well as internal processes.

4.2.1 System Architecture Diagram (High-Level)

The Spices Management System consists of:

1. Client Layer

- Web browser (Admin Panel / User Panel)
- Mobile responsive UI

2. Application Layer

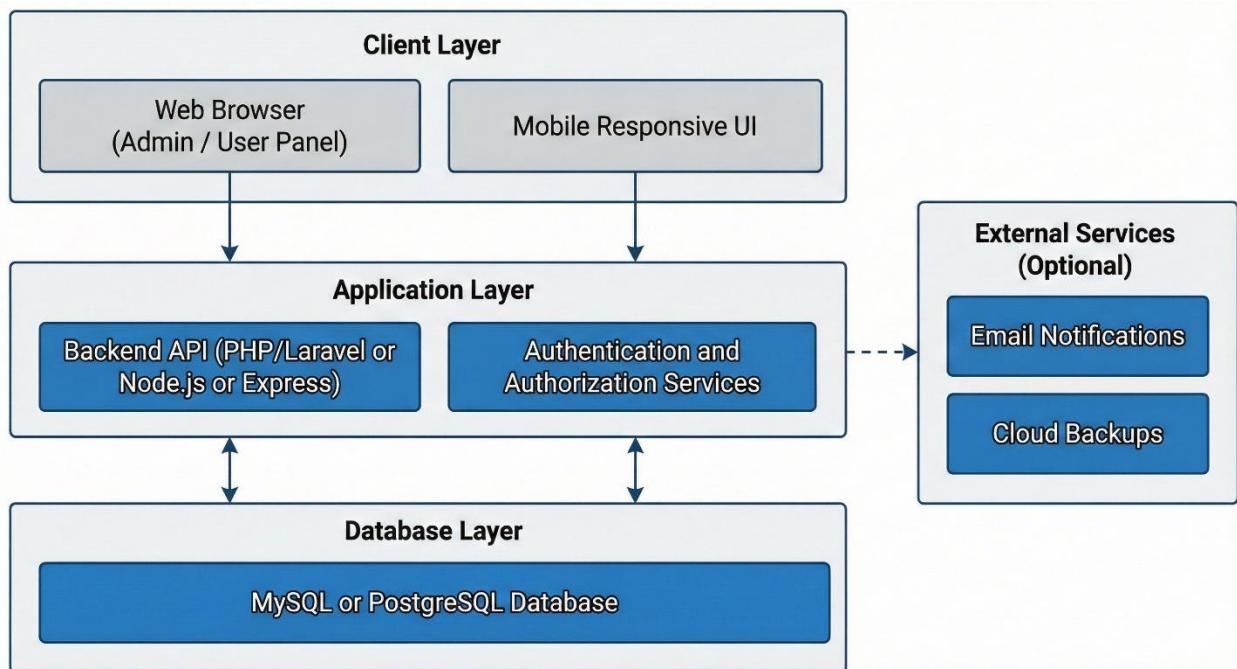
- Backend API (PHP/Laravel or Node.js or Express)
- Authentication and authorization services

3. Database Layer

- MySQL or PostgreSQL database

4. External Services (Optional)

- Email notifications
- Cloud backups



4.2.2 Use Case Diagram (Description)

Actors

- **Administrator**
- **Standard User**
- **Supplier** (optional, depending on system scope)

Use Cases

- Login
- Manage Users (Admin)
- Manage Inventory
- View Inventory Stock
- Record Sales
- Generate Reports
- Manage Suppliers
- Create Purchase Orders

- Receive Goods

4.2.3 Sequence Diagrams

1. Login Process

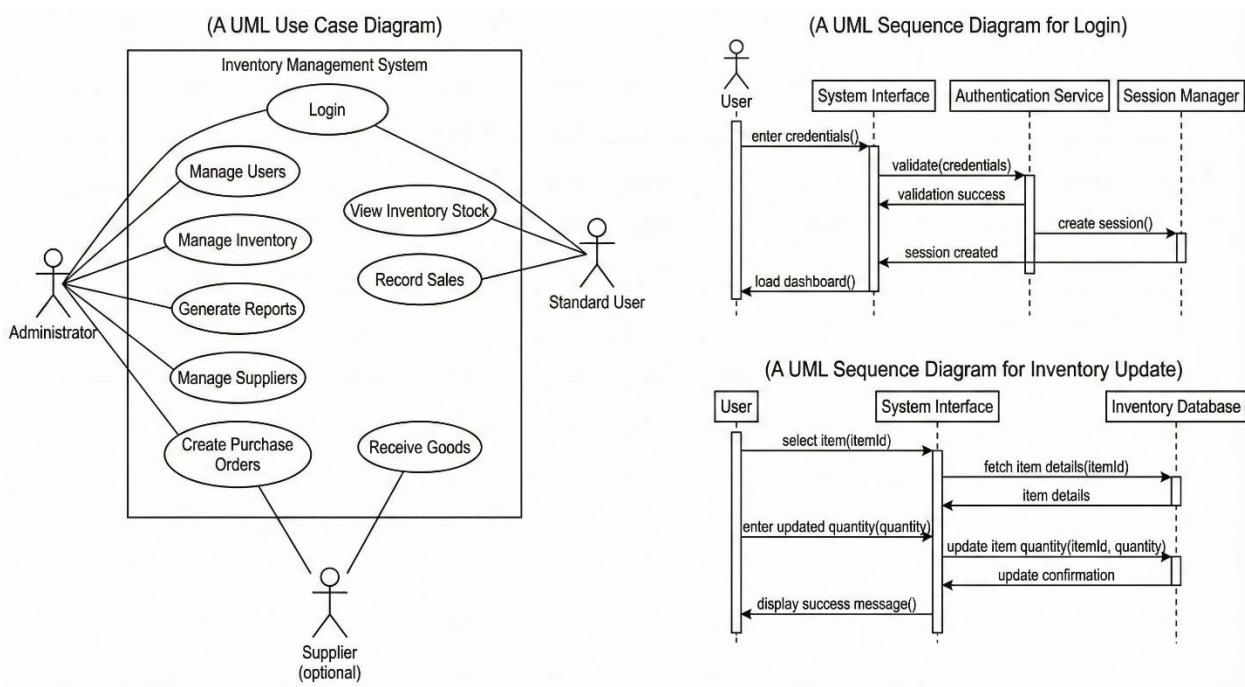
1. User enters credentials
2. System validates
3. Session created
4. Dashboard loaded

2. Inventory Update Process

1. User selects items
2. Enters updated quantity
3. System validates input
4. Database updates item quantity
5. Confirmation returned

3. Generate Report (PDF)

1. Admin enters date range
2. System retrieves sales data
3. System compiles PDF
4. Download link provided



Thank You!