Retail Shop Management System - SRS

Revision History

Name	Date	Reason For Changes	Version
Muhammad Arham	13/02/2025	Initial Draft	1.0
Alishba Zahid	23/02/2025	revised and expanded details, including the tables for better clarity	1.1

Name	Date	Reason For Changes	Version
Muhammad Moiz Ahmad	27/02/2025	Added version history, expanded security measures, role-based access, POS system details, and performance metrics	1.2
Muhammad Moiz Ahmad	28/02/2025	formatting of requirements specification, additions of assumptions.	1.3
Alishba Zahid	28/02/2025	Detail in product perspective and test cases for the functional requirements	1.4

Version History

Version	Summary of Changes
1.0	Initial draft with core system requirements
1.1	Expanded feature, added detailed functional and non-functional requirements and additional enhancement.
1.2	Added version history, enhanced security measures, role-based access, POS system details, and performance metrics
1.3	Main heading of requirements with the subheading with external interface, functional and non-functional, assumptions made for the system development.
1.4	Detail in product perspective and test cases for the functional requirements with the scenarios and functionality

1. Introduction

1.1 Purpose

This system aims to streamline the operations of a retail shop, including inventory management, sales tracking, customer management, and reporting. The RSMS aims to reduce manual work, enhance efficiency, and provide a seamless user experience for retailers and the staff.

1.2 Document Conventions

This document follows standard formatting with:

- Bold for key terms.
- Bullet points for list.
- Tables where applicable.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

Role	Responsibility
Developers	To understand system architecture and requirements.
Project Managers	To track deliverables and system functionality.
End Users(Retail Owner, Staff)	Understanding system capabilities and usage.
QA Testers	Validating system functionality against defined requirements..

1.4 Project Scope

The Retail Shop Management System is a solution designed to assist shop owners in:

- Managing inventory, including stock levels, pricing and details of the supplier.

- Generating invoices and maintaining customers and supplier detail.
- Managing different user roles with secure access.
- Providing a simple and easy to use interface.
- Creating real time reports on sale and inventory.

1.5 References

- IEEE 830-1998 Software Requirements Specification Standard.
- Relevant APIs and system documentation.

2. Overall Description

2.1 Product Perspective

A centralized solution that helps shop owner manage their daily operations smoothly. It allows multiple users to access the system with different permission levels, ensuring secure data handling. The system will be a web-based application that can be access through any modern browser. It connects with devices like barcode scanners, receipt printers and POS machine for sale transaction. It also supports online payments helping retailers to sell their product both online and in-store.

2.2 Product Features

Feature	Description
Inventory Management	Add, update and delete products, track of the stock items, and alerts on the low- stock items.
Sales and Billing	Handling of transactions, invoice printing, apply discounts and manage payment methods.
Customer Management	Save customer details, track of most frequent purchases, available discounts categorize customers and manage loyalty programs.
Supplier Management	Maintain supplier records, delivery timelines, product quality and product sourcing.

Feature	Description
Role-based Access	Assign the roles with specific access permissions.
Reports and Analytics	Generate sales, inventory and revenue reports based on real-time.
Multi-Store Support	Allow managing multiple branches under one system with centralized inventory and sales tracking.
Employee Management	Manage employee profiles, assign shifts, and track performance.
Refunds & Returns Handling	Implement policies for processing refunds, exchanges, and store credit.
Expense Management	Track store expenses such as rent, utilities, and employee salaries.
Customer Feedback System	Collect feedback from customers on their shopping experience.
AI-Based Demand Forecasting	Use machine learning models to predict stock demand based on sales history.
Loyalty & Rewards	Implement points-based reward programs to increase customer retention.

2.3 User Classes and Characteristics

- **Shop Owner/Admin**: Full access to the system, including product management and financial reports.
- **Cashiers**: Limited access for processing sales and managing customers.
- **Customers**: Can view available products and view product prices.

2.4 Operating Environment

- **Mobile Compatibility**: The system should be mobile-responsive for store owners to monitor sales remotely.
- **Cloud & On-Premise Support**: The system should offer both cloud-based and offline versions
- Web-based (accessible on Windows, macOS, Linux)
- Database: PostgreSQL for securing data storage.

- Backend: Node.js
- Frontend: React

2.5 Design and Implementation Constraints

- The system must use an SQL-based relational database.
- System must support real-time inventory updates.
- Data encryption required for user authentication.

2.6 User Documentation

- A **detailed user manual** will be provided.
- System documentation will be maintained on **Notion** and uploaded to **GitHub**.

2.7 Assumptions and Dependencies

Various assumptions and dependencies are created to ensure the product operates safely, efficiently and under optimal conditions.

2.7.1 Assumptions

- The Main assumption made for this is that the system will be used in retail environment with stable internet connectivity.
- The system assumed to be worked with popular gateway methods.
- The data stored of the customer should be accurate and correct.
- The system will operate in indoor environment with stable power supply.
- Customers will provide valid and verifiable contact details for loyalty programs and online orders.
- Any downtime in internet connectivity will be minimal and will not disrupt core operations for an extended period.

3. Requirements Specification

3.1 External Interface Requirements

3.1.1 User Interfaces

Interface Type	Description
Web Dashboard	For shop owners to manage sales, inventory and reports
Point of Sale (POS) System	Used by cashiers to process transactions.
Customer-Facing Display	Display items prices and transaction details.

3.1.2 Hardware Interfaces

- Compatible with barcode scanners and receipt printers.
- Works with standard POS hardware for retail transactions.
- **Weighing Scale Integration**: Allow integration with digital weighing scales for groceries

3.1.3 Software Interfaces

- Integration with external payment gateways.
- API-based communication with inventory suppliers.
- **WhatsApp & Email Notifications**: Send automated order confirmations and promotions.
- **E-commerce Integration**: Sync inventory with online stores (Shopify, WooCommerce).

3.1.4 Communications Interfaces

- Secure HTTPS-based communication.
- Email/SMS notifications for order updates.

3.2 Functional Requirements

3.2.1 User Authentication and Access Control

- Users must log in with a username and password.

- Role-based access control (Admin, Cashier, Customer) ensures security by limiting user permissions.

3.2.2 Inventory Management

Product Management:

- Add, update and delete products.
- Categorization of products with SKU and barcode support.

Stock Tracking:

- Track stock levels and generate low-stock alerts.
- Generate notifications for products that fall below predefined stock levels.

Batch Tracking:

- Maintain product batch details, including manufacturing and expiry dates.

Expired Products Handling:

- Automatically identified expired products and remove them from available stock.

Stock Adjustment:

- Enable users to check and correct inventory differences manually or automatically to ensure accurate stock records.

3.2.3 Sales and Billing

- Process sales through a Point of Sale (POS) system.
- Generate invoices for customers.
- Apply discounts and manage promotions.
- Enable customers to use accumulated points as discounts on purchases.

3.2.4 Customer Management

- Store customer details (name, contact, purchase history).

- Manage supplier information and product sourcing.
- Segment customers into categories such as frequent buyer and new customer.
- Track customer purchases and offer discounts based on spending pattern.

3.2.5 Supplier Management

- Monitor supplier delivery timelines and expected arrival dates.
- Rate and track supplier reliability based on order accuracy and the delivery time.
- Automate the creation and tracking of purchase orders for better supplier coordination.

3.2.6 Reports and Analytics

- Generate daily, weekly, and monthly sales reports.
- Analyze customer buying patterns and inventory trends.

3.2.7 Discount & Promotion Management

- Set up time-based promotions (e.g., Blessed Friday, Eid Sales).
- Automate seasonal discounts and track effectiveness.

3.2.8 Vendor & Purchase Order Management

- Generate purchase orders automatically based on inventory levels.
- Compare vendor prices and order history for cost-effective decisions.

3.2.2 Test Case Scenarios

3.2.2.1. Scenario: User Authentication

Description

Users must log in using their credentials to access the system.

Functional Response

The system verifies user credentials against the database. If authentication is successful, the user is granted access

3.2.2.2. Scenario: Role-Based Access Control

Description

The system restricts access to certain features based on user roles (Admin, Cashier, Customer).

Functional Response

The system grants permissions according to user roles. Admins have full access, cashiers can process transactions but cannot modify inventory, and customers can view product details. Unauthorized actions prompt an access-denied message.

3.2.2.3. Scenario: Product Management

Description

Admin modifies existing product details, such as price, name, or category.

Functional Response

The system updates the product record and reflects changes in real-time across all sales channels. Previous product history is logged for auditing purposes.

3.2.2.4. Scenario: Stock Tracking

Description

The system monitors inventory levels and alerts users when stock is low.

Functional Response

The system continuously updates stock levels based on sales and supplier orders. If stock falls below a predefined threshold, an alert is triggered for restocking.

3.2.2.5. Scenario: Batch Tracking

Description

The system records manufacturing and expiry dates for each product batch.

Functional Response

When a new batch is added, the system requires batch details, including production and expiry dates. Products nearing expiry trigger alerts for store management.

3.2.2.6. Scenario: Expired Products Handling

Description

Expired products are automatically removed from inventory.

Functional Response

The system identifies expired products daily. Expired items are removed from the available stock.

3.2.2.7 Scenario: Sales and Billing

Description

The system provides an invoice for completed transactions.

Functional Response

After processing a sale, the system generates an invoice with item details, total cost, applied discounts, and payment method. The invoice can be printed or emailed to the customer.

3.2.2.8. Scenario: Customer Management

Description

The system saves customer contact details and purchase history. The system categorizes customers based on spending patterns.

Functional Response

When a customer makes a purchase, their details can be stored in the system. Their buying patterns are analyzed for personalized offers and discounts. The system analyzes past purchases and assigns customer segments, such as frequent buyers or VIP customers. This data is used for personalized promotions. Each sale is logged under the customer’s profile, allowing shop owners to view purchase history

3.2.2.9. Scenario: Reports and Analytics

Description

The system tracks and reports customer shopping habits.

Functional Response

Based on sales history, the system identifies trends, such as peak buying hours and best-selling products, aiding business strategy.

3.2.2.10. Scenario: Vendor & Purchase Order Management

Description

The system creates a purchase order based on stock needs.

Functional Response

When stock levels are low, the system recommends purchase orders. Users can review and send orders directly to vendors.

3.3. Other Nonfunctional Requirements

3.3.1 Performance Requirements

- System should support **100+ transactions per second**.
- Inventory updates should reflect within **2 seconds**.

3.3.2 Safety Requirements

- Daily data backups with a **1-hour** recovery time.
- Error logging and alerts notifications for failures.

3.3.3 Security Requirements

- User **authentication** with encryption**.**
- **Role-based access control** to prevent unauthorized access
- **Data encryption** for sensitive information like customer details and transactions.
- **Audit logs** to track system activities and security exposure.

- **Automatic Fraud Detection**: Alert admins about suspicious transactions (e.g., excessive refunds).

3.3.4 Software Quality Attributes

- **Reusability**: Built using an object-oriented programming paradigm.
- **Maintainability**: Well-documented codebase with a simple admin panel.
- **Reliability**: Ensures **99.9% uptime** with automatic backups.
- **Scalability**: Supports multiple locations and high transaction volumes.
- Interoperability: Ensure smooth integration with third-party payment gateways.

3.3.5 Compliance & Legal Requirements

- **GDPR & Data Privacy Compliance**: Ensure customer data security as per international regulations.
- **Tax Compliance**: Automatically calculate sales tax based on region.

4. AI & Automation Features

- **Chatbot for Customer Support**: Provide automated assistance for FAQs and order status inquiries.
- **Smart Reordering**: AI-based auto-replenishment of stock based on demand forecasting

5. Other Requirements

- Compliance with relevant retail regulations.
- Localization support for multiple languages.

6. Project Management

- **Jira** will be used for task tracking.
- Development updates and version control will be maintained in **GitHub**.

7. Conclusion

This document outlines all the essential requirements for the Retail Shop Management System. The system will help retail businesses by automating inventory tracks, sales processing and reporting. It ensures security, reliability, and ease of use, making it a valuable tool for shop owners.