Software Requirements Specification

for

<Supermarket management system >

Version <1.1>

Prepared by

< rawda hesham el sayed>	< 202220105>
<rahma saber abaas>	<202220250>
<hanen Mohamed elmahdy>	<202220085>

Instructor:	< Mohamed Reda >
Course:	< software engineering>
Teaching Assistant:	< Eng *Mohamed Karam* >
Date:	<11/14/2024>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Rawda,hanen ,rahma	11/14/2024	Add requirements	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to ensure that all stakeholders—ranging from developers and testers to project managers—have a clear understanding of the system's features and functionality. It serves as a blueprint for the development, testing, and deployment of the SMS, and ensures that the system meets the operational needs of the supermarket business while aligning with technical and business goals.

The system aims to automate and streamline the operations of a supermarket, covering key functions such as inventory management, point of sale (POS), customer management, and reporting. This document specifically outlines the system's requirements for version **1.1.**

1.2 Document Conventions

This document uses the following conventions to enhance readability and ensure consistency:

- **Bold text**: Used to highlight section titles, important terms, and key concepts.
- *Italics*: Used for emphasis, citations, and external references.
- **Numbered lists**: Used for ordered steps or processes.
- **Bulleted lists**: Used for unordered items such as feature sets or descriptions.
- **Priority Indicators**: Each requirement will have a priority level to indicate its importance:
    - **P0 (Critical)**: Required for the system to function.
    - **P1 (High)**: Important features, but not essential for the first version.
    - **P2 (Medium)**: Desirable features that improve user experience but can be implemented later.

1.3 Intended Audience and Reading Suggestions

- **Project Managers**: To understand the scope, timeline, and milestones for the system's development and delivery.
- **Software Developers and System Architects**: To gain insights into the system's technical specifications and requirements.
- **Testers and Quality Assurance (QA) Engineers**: To ensure the system is thoroughly tested against the specified requirements.
- **Supermarket Staff (End Users)**: To gain an understanding of the system's functionalities and how they will interact with it daily.
- **Business Stakeholders and Decision Makers**: To understand the business objectives and how the system meets the goals of the supermarket.

Suggested reading sequence:

1. **Section 1 (Introduction)**: Provides an overview of the document, the system, and the key goals of the SRS.
2. **Section 2 (Overall Description)**: Offers context for the system, including high-level descriptions of its components and operational environment.
3. **Section 3 (System Features)**: Provides a detailed breakdown of the functional requirements and features the system must include.
4. **Section 4 (External Interface Requirements)**: Describes how the system will interact with external systems and hardware.
5. **Section 5 (Non-Functional Requirements)**: Outlines the performance, security, and usability requirements for the system.

1.4 Product Scope

The **Supermarket Management System (SMS)** is a software solution designed to automate and improve the operations of a supermarket. It will include features that facilitate the day-to-day management of supermarket processes, from inventory tracking to customer transactions and reporting. The key modules of the system are:

- **Inventory Management**: Enables supermarket staff to monitor product stock levels, update product information, manage suppliers, and handle restocking processes.
- **Point of Sale (POS)**: Provides an intuitive interface for cashiers to perform customer transactions, including scanning items, processing payments, and generating receipts.
- **Customer Management**: Tracks customer profiles, purchase history, and loyalty program details, enhancing customer experience and providing marketing insights.
- **Reporting**: Generates real-time reports on sales, inventory, financials, and other metrics to support business decision-making.

The system will also integrate with hardware such as barcode scanners, cash registers, and receipt printers, as well as external systems like payment gateways for secure transactions.

In addition to core functionality, the system is designed to be scalable, with potential future enhancements such as online ordering, mobile app integration, and advanced analytics.

This SRS focuses on the foundational features of the SMS as part of the first release, with additional features planned for subsequent phases.

1.5 References

The following documents and resources are referenced in this Software Requirements Specification:

1. **Supermarket Management System –Design and implementation**,[Yongchang Ren], [January 2015], [https://www.researchgate.net/publication/301447741_Design_and_Implementation_of_Supermarket_Management_System].

2. **Supermarket Management System –Design and implementation**,[Ajitesh27], [*Sep 8, 2021*][https://github.com/Ajitesh27/SuperMarket-Management-System]
3. **ISO/IEC 9126: Software Engineering** [https://www.geeksforgeeks.org/iso-iec-9126-in-software-engineering/]
4. **IEEE (Institute of Electrical and Electronics Engineers)** for technical documents and engineering papers,[https://doi.org/10.1109/ICAPC57304.2022.00051][08-10 September 2022]

These documents provide additional context on system specifications, interface requirements, user experience guidelines, and quality standards.

2. Overall Description

2.1 Product Perspective

- Scalability: Ensure the system can handle a growing number of products and transactions efficiently.

- Product Variants: Support *different product types, sizes, and variants.*

- Product Information: Store detailed product information including pricing, descriptions, and images.

- Product Search: Enable customers to easily search for products based on various criteria.

- Product Categorization: Organize products into categories for easy navigation.

- Product Reviews: Allow customers to leave reviews and ratings for products.

- Product Recommendations: Implement a recommendation system based on customer purchase history.

2.2 Product Functions

- Provide detailed descriptions, images, and videos for products.

- Allow customers to leave reviews and ratings.

Product Search and Navigation:-

- Enable customers to search for products by various criteria.

- Offer filtering options and personalized recommendations.

2.3 User Classes and Characteristics

1. Admin:Manages system configurations and user roles.

2. Cashiers:Process transactions at the point of sale.

,Need training on point of sale (POS)systems and payment handling.

3. Inventory Managers:Monitor stock levels and generate purchase orders.

,Regularly ensure adequate stock levels.

4. Customers: Make purchases and interact with the system for shopping needs.

, Varying technical expertise levels.

5. Suppliers:Provide products and manage orders.

, Need access to supplier portals.

6. Customer Service Representatives:Assist customers with queries and support.

, Communication skills and product knowledge.

7. System Analysts: Analyze system performance and user feedback.

, Technical expertise in system analysis.

2.4 Operating Environment

- Hardware Platform:The software is designed to run on standard hardware configurations including desktop computers and servers.

- Operating System and Versions:
  - Compatible with: Windows 10, macOS Catalina
  - Supports other modern operating systems that are prevalent in retail environments.
- Database Compatibility: The software should peacefully coexist with:
    - MySQL, PostgreSQL, MongoDB
  - Integration with the existing database infrastructure commonly used in supermarkets.

- Web Browser Support:- Compatible with major web browsers:
    - Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
  - Ensures optimal performance with the latest versions of these browsers.

- Networking: Requires network connectivity for communication between system component ,Supports both wired and wireless network configurations typically found in retail settings.

2.5 Design and Implementation Constraints

- Regulatory Compliance: Adherence to industry regulations for data security and financial transactions.

- Hardware Limitations: Operating within defined hardware constraints for performance and scalability.

- Interface Compatibility: Seamless integration with existing systems, following specific protocols.

- Technology Stack: Use of predetermined technologies and tools for compatibility and maintenance ease.

- Security Considerations: Implementation of robust security measures to protect sensitive data.

- Programming Standards: Adherence to design conventions and coding practices for maintenance and readability.

- Communication Protocols: Consistent protocols for smooth interaction between system components.

- Maintenance Responsibility: Customer organization tasked with post-implementation software maintenance.

3. External Interface Requirements

3.1 User Interfaces

The **Supermarket Management System** will have several user interfaces, including graphical user interfaces (GUIs) for staff (cashiers, managers, and inventory personnel) and web-based interfaces for administrators and remote users.

3.1.1 Point of Sale (POS) Interface

- **User**: Cashiers, Customers (during checkout)
- **Description**: A touchscreen interface for cashiers to process sales transactions. The interface will allow cashiers to scan product barcodes, apply discounts, enter quantities, calculate totals, and process payments.
- **Input**:
  - Product barcode via barcode scanner.
  - Quantity input via touchscreen or keyboard.
  - Payment method selection (cash, credit card, mobile wallet).
- **Output**:
  - Transaction total and receipt generation.
  - Payment confirmation.
  - Customer loyalty program status (if applicable).

3.1.2 Inventory Management Interface

- **User**: Inventory Managers, Store Managers
- **Description**: A GUI for managing product stock, updating inventory levels, adding new products, tracking supplier orders, and generating inventory reports.
- **Input**:
  - Product details (name, price, description, barcode, etc.).
  - Stock updates (restocking, quantity changes).
  - Supplier details for reordering stock.
- **Output**:
  - Inventory status reports (e.g., low stock alerts).
  - Supplier and restocking order information.

3.1.3 Reporting Interface

- **User**: Store Managers, Supervisors, Analysts
- **Description**: A graphical interface to generate and view various reports, such as sales reports, inventory status, and financial performance.
- **Input**:
  - Report parameters (date range, product category, specific data points).
- **Output**:
  - Sales reports, inventory reports, customer purchase reports, and financial summaries.
  - Graphical charts and exportable formats (e.g., CSV, PDF).

3.1.4 Customer Management Interface

- **User**: Customers (if supported by the system), Store Managers
- **Description**: A customer-facing interface (if applicable) where users can register for loyalty programs, update profiles, and view purchase history.
- **Input**:
  - Customer details (name, email, phone number).
  - Purchase history and loyalty program engagement.
- **Output**:
  - Loyalty points and rewards status.
  - Personalized promotions or offers.

3.2 Hardware Interfaces

The system will interact with various hardware devices commonly used in supermarkets to handle transactions and streamline daily operations.

3.2.1 Barcode Scanners

- **Description**: The SMS will interface with barcode scanners to automatically input product information during checkout or stock updates.
- **Communication**: The system will communicate with barcode scanners via USB or Bluetooth.
- **Expected Data**:
  - Product ID (barcode).
  - Quantity (if manually entered by cashier).

3.2.2 Receipt Printers

- ***Description**: The SMS will generate receipts for transactions, which will be printed using receipt printers connected to the POS system.*
- ***Communication**: Communication via USB, serial port, or network connection.*

- ***Expected Output****:*
  - *Transaction details (items purchased, prices, totals, payment method, date/time).*
  - *Customer information (if applicable), including loyalty program details.*

3.2.3 Cash Registers

- ***Description****: The POS system will interface with cash registers to process cash transactions.*
- ***Communication****: Communication via USB, serial port, or network.*
- ***Expected Output****:*
  - *Cash amount entered by cashier.*
  - *Cash change amount to be returned to customer.*

3.2.4 Payment Terminals (Credit Card Readers)

- **Description**: The SMS will interface with credit card terminals or mobile payment devices to process electronic payments (e.g., debit/credit cards, mobile wallets).
- **Communication**: Communication through standard payment protocols such as **EMV** (Europay MasterCard and Visa), NFC (Near Field Communication) for contactless payments, or **PCI-DSS** compliant APIs for mobile payments.
- **Expected Data**:
  - Payment amount.
  - Payment method (credit/debit card, mobile wallet).
  - Transaction confirmation or error response.

3.2.5 Digital Kiosks or Self-Checkout Stations (Optional)

- **Description**: The SMS may interface with self-checkout kiosks where customers can scan products, process payments, and print receipts without cashier assistance.
- **Communication**: Communication via touchscreen interface, barcode scanner, and receipt printer.
- **Expected Data**:
  - *Product scans (via barcode scanner).*
  - *Payment method (via card reader or cash).*
  - *Transaction summary.*

3.3 Software Interfaces

The **Supermarket Management System** will interact with various external software systems and APIs to enable seamless operation across different business functions.

3.3.1 Payment Gateway Integration

- ***Description**: The SMS will integrate with third-party payment gateways (e.g., PayPal, Stripe, or bank-specific solutions) to securely process customer payments.*
- ***Protocol**: Secure HTTPS (SSL/TLS) for secure data transmission between the SMS and the payment gateway.*
- ***Expected Data**:*
    - *Payment details (amount, payment method).*
    - *Transaction ID and authorization status.*
    - *Payment confirmation or failure response.*

3.3.2 External Database or Cloud Storage (Optional)

- ***Description**: The SMS may use an external database or cloud-based storage solution for backup, remote access, or to integrate with other business applications (e.g., analytics tools, CRM systems).*
- ***Protocol**: RESTful APIs or database connection via SQL/NoSQL protocols.*
- ***Expected Data**:*
    - *Inventory data (products, suppliers, orders).*
    - *Customer data (profiles, loyalty points).*
    - *Sales transaction data (for analysis or reporting).*

3.3.3 Inventory Supplier Systems (Optional)

- ***Description**: The SMS may integrate with external supplier systems to automate stock reordering and order tracking.*
- ***Protocol**: RESTful APIs or file-based integration (CSV, XML).*
- ***Expected Data**:*
    - *Product catalog information from suppliers.*
    - *Order request details and status (e.g., order confirmations, delivery tracking).*

3.3.4 Email and SMS Service Providers

- **Description**: The SMS will interface with email and SMS services to send receipts, promotions, and loyalty program updates to customers.
- **Protocol**: SMTP for email, SMS gateway API for text messaging.
- **Expected Data**:
    - Customer email addresses or phone numbers.
    - Receipt and promotional message content.

3.4 Communications Interfaces

The **Supermarket Management System** will use the following communication protocols for external integrations and data exchange:

- **HTTP/HTTPS**: For secure web-based communications, including payment gateways and cloud integrations.
- **RESTful APIs**: For interfacing with third-party services, such as payment processors, suppliers, and inventory databases.
- **TCP/IP or Serial Communication**: For connecting hardware devices like barcode scanners, printers, and cash registers.
- **SMTP**: For sending emails (e.g., receipts or customer notifications).
- **SMS Gateway Protocol**: For sending text message notifications to customers (e.g., loyalty rewards, promotions).

3.5 Security and Data Privacy Considerations

Given the sensitive nature of the data being processed, especially payment information and customer details, the system will adhere to the following security and privacy standards:

- **PCI-DSS Compliance**: For securely processing credit card payments and handling financial transactions.
- **SSL/TLS Encryption**: For secure communication between the system and external services (payment processors, cloud storage, etc.).
- **Data Privacy**: Ensure that customer data is handled in compliance with relevant privacy regulations (e.g., GDPR) and that sensitive information is stored securely.

4. System Features

The **Supermarket Management System (SMS)** is designed to streamline the operations of a supermarket by automating key tasks such as inventory management, sales transactions, customer management, and reporting. This section outlines the core features and functionality of the system.

4.1 System Feature 1

4.1.1 Feature Description:

The Inventory Management module allows supermarket staff to track product quantities, manage suppliers, and handle stock replenishments. It helps ensure that the supermarket is stocked with the right products at the right levels, minimizing out-of-stock situations while avoiding overstocking.

4.1.2 Functional Requirements:

- **Track Product Stock**: The system must keep track of stock levels for all products and update quantities in real-time as items are sold or restocked.
- **Barcode Scanning**: When receiving deliveries or updating inventory, staff should be able to scan barcodes to quickly add or update product information.
- **Low Stock Alerts**: The system will generate alerts when stock levels fall below a predefined threshold to trigger reordering from suppliers.

- **Supplier Management**: The system will store information about suppliers, including contact details, product catalogs, and order histories. Managers can generate purchase orders directly from the system to suppliers when stock needs replenishing.
- **Product Categorization**: Products can be organized into categories (e.g., dairy, beverages, bakery) for easier management and reporting.
- **Inventory Adjustment**: The system allows authorized users to adjust inventory manually in cases of loss, damage, or other errors.

4.1.3 Non-Functional Requirements:

- **Performance**: The system should be capable of managing inventory data for hundreds to thousands of unique products.
- **Security**: Access to inventory management features must be restricted to authorized personnel (e.g., inventory managers, store managers).
- **Usability**: The interface for updating stock levels should be intuitive, enabling staff to perform tasks quickly with minimal training.

4.2 System Feature 2 (and so on)

The Supermarket Management System (SMS) **is designed to streamline the operations of a supermarket by automating key tasks such as inventory management, sales transactions, customer management, and reporting. This section outlines the core features and functionality of the system.**

4.1 Inventory Management

4.1.1 Feature Description:

The Inventory Management module allows supermarket staff to track product quantities, manage suppliers, and handle stock replenishments. It helps ensure that the supermarket is stocked with the right products at the right levels, minimizing out-of-stock situations while avoiding overstocking.

4.1.2 Functional Requirements:

- ***Track Product Stock****: The system must keep track of stock levels for all products and update quantities in real-time as items are sold or restocked.*
- ***Barcode Scanning****: When receiving deliveries or updating inventory, staff should be able to scan barcodes to quickly add or update product information.*
- ***Low Stock Alerts****: The system will generate alerts when stock levels fall below a predefined threshold to trigger reordering from suppliers.*

- ***Supplier Management**: The system will store information about suppliers, including contact details, product catalogs, and order histories. Managers can generate purchase orders directly from the system to suppliers when stock needs replenishing.*
- ***Product Categorization**: Products can be organized into categories (e.g., dairy, beverages, bakery) for easier management and reporting.*
- ***Inventory Adjustment**: The system allows authorized users to adjust inventory manually in cases of loss, damage, or other errors.*

4.1.3 Non-Functional Requirements:

- ***Performance**: The system should be capable of managing inventory data for hundreds to thousands of unique products.*
- ***Security**: Access to inventory management features must be restricted to authorized personnel (e.g., inventory managers, store managers).*
- ***Usability**: The interface for updating stock levels should be intuitive, enabling staff to perform tasks quickly with minimal training.*

4.2 Point of Sale (POS) System

4.2.1 Feature Description:

The POS system is a central part of the **Supermarket Management System** that handles customer checkout. It enables cashiers to process transactions efficiently, calculate totals, accept various payment methods, and print receipts.

4.2.2 Functional Requirements:

- **Product Scanning**: The system must allow products to be scanned by barcode using a barcode scanner. If the product does not have a barcode, cashiers can enter the product ID manually.
- **Price Calculation**: The system will automatically calculate the total price based on the scanned products and applied discounts (if any).
- **Discount Management**: Cashiers can apply various discounts, including coupons, promotions, and loyalty-based discounts.
- **Payment Processing**: The system must support multiple payment methods, including cash, credit/debit cards, and mobile wallets (e.g., Apple Pay, Google Pay).
- **Receipt Generation**: After completing the transaction, the system will generate a printed receipt for the customer, with an option for digital receipts (via email/SMS).
- ***Sales Reporting**: After each transaction, the system will record details for sales analysis, including items sold, payment method, and total revenue.*

4.2.3 Non-Functional Requirements:

- **Response Time**: The POS system should process transactions in less than 3 seconds to ensure a fast checkout experience.

- **Security**: Sensitive customer and payment information must be encrypted and stored securely to meet PCI-DSS requirements.
- **Reliability**: The system must be reliable and available 24/7 with minimal downtime. It should support offline operations during network outages, allowing transactions to be stored locally and synchronized when the network is restored.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Response Time

- **POS transactions** should be completed within **3 seconds** of scanning a product or entering a payment method.
- System queries (such as retrieving product data or customer information) should return results in **less than 2 seconds** for smooth user experience, especially at checkout.
- **Inventory updates** (e.g., stock changes after a sale or restocking) should be reflected in the system **in real-time** with no noticeable delay.

5.1.2 Throughput

- The system should be able to handle up to **500 transactions per hour** at peak times, ensuring scalability as the business grows.
- **Batch operations**, such as generating sales reports or updating prices for multiple items, should be processed efficiently with no significant delays (processing time for batch operations should be under **30 minutes**).

5.1.3 Concurrent Users

- The system should support **up to 1,000 concurrent users** (e.g., cashiers, inventory managers, and administrators) without performance degradation.
- The **database** should be capable of handling **multiple concurrent read and write operations** without compromising data integrity or system speed.

5.1.4 Scalability

- The system should be **scalable** to accommodate future growth, including adding more POS terminals, employees, and locations, without a major overhaul of the infrastructure.
- The architecture should be designed to handle **up to 10,000 products** and **50,000 customer records** seamlessly.

5.2.1 System Fault Tolerance

- The system must ensure **failover mechanisms** are in place for critical components such as **payment processing** and **inventory management**. In case of failure, these components should switch to a backup system without causing a disruption to operations.
- The **POS system** should be able to continue operating **offline** during short periods of network failure and synchronize the data once the connection is restored.

5.2.2 Data Backup and Recovery

- The system should implement **automatic daily backups** to prevent data loss due to system failures or disasters. These backups should be encrypted and stored securely in an offsite location or cloud service.
- The backup strategy should allow for quick recovery within **1 hour** in case of system failure. The **Recovery Time Objective (RTO)** should be within **1 hour**, and the **Recovery Point Objective (RPO)** should not exceed **15 minutes**.

5.2.3 Disaster Recovery

- The disaster recovery plan should include detailed steps for restoring system functionality within **4 hours** after a significant failure or disaster (e.g., hardware crash, data breach).
- The system should allow for **redundant power supplies** for critical infrastructure, ensuring no downtime due to power failure.

5.2.4 Safe User Interfaces

- User interfaces should be designed with **safety in mind**, such as ensuring that employees can cancel or correct transactions easily at the **POS** without the risk of accidental data loss or fraud.
- The POS system should include a **confirmation dialog box** for critical actions (e.g., deleting items from the cart, finalizing payments).

5.3 Security Requirements

5.3.1 Data Protection and Privacy

- The system must **encrypt sensitive data** (e.g., customer information, credit card details) both **in transit** and **at rest** using industry-standard encryption protocols such as **AES-256** and **TLS**.
- **Payment card information** must be **tokenized** or **masked** according to **PCI DSS** standards, ensuring no sensitive payment information is stored in plain text.
- The system should provide customers with **the option to manage their privacy settings**, including opting in or out of promotional communications and deleting personal information.

5.3.2 Authentication and Access Control

- The system should implement **role-based access control (RBAC)** to limit user access to specific functions based on their role (e.g., cashier, manager, admin).
- **Admin-level access** should require **multi-factor authentication (MFA)**, ensuring secure access to sensitive areas of the system (such as system settings and payment configurations).
- Users should be automatically **logged out after a period of inactivity** (e.g., 15 minutes), and any session should be **encrypted** to prevent unauthorized access.

5.3.3 Audit Trails

- The system must keep **audit logs** for all user actions, especially administrative tasks such as adding/removing products, updating inventory, or modifying prices. The logs should include timestamps, user IDs, and the details of the action performed.
- The logs should be accessible to system administrators for monitoring purposes but protected from unauthorized access. Logs should be **stored securely** for a minimum of **6 months**.

5.3.4 Incident Response

- The system must include an **incident response protocol** in case of a data breach, unauthorized access, or other security threats. The protocol should involve:
  - Immediate investigation of the incident.
  - Notifying affected parties (e.g., customers) within **72 hours** if their data is compromised.
  - Implementing steps to prevent future incidents.

5.4 Software Quality Attributes

5.4.1 Reliability

- The system should be **99.9% reliable**, with minimal downtime during operational hours. If downtime occurs, it should be **brief and well-communicated** to users.
- The software should be thoroughly **tested for reliability** through load testing, fault injection, and stress testing to ensure it can handle peak loads and failures without impacting functionality.

5.4.2 Maintainability

- The system should be designed with a **modular architecture** that allows easy updates, bug fixes, and new feature additions without disrupting the whole system.
- Code should follow established coding standards and best practices, ensuring that developers can easily maintain and extend the system.

5.4.3 Usability

- The system must be intuitive, with a **user-friendly interface** designed for non-technical staff (e.g., cashiers, managers). This includes simple and clear navigation, icons, and prompts to reduce training time.
- The **POS interface** should allow transactions to be processed quickly with **minimal user input**. For example, a **barcode scanner** should automatically detect products, and discounts or promotions should be applied with one click.

5.4.4 Flexibility

- The system must be **flexible enough to adapt** to changes in business needs, such as adding new types of discounts, promotions, or reports. For example, the system should allow custom discount rules for specific customer groups (e.g., senior citizens, loyalty program members).
- The system should also allow for easy integration with **third-party systems**, such as e-commerce platforms or accounting software.

5.4.5 Portability

- The software should be portable across multiple devices (e.g., desktop, mobile, tablets) and platforms (Windows, Linux, etc.) to ensure flexibility in deployment across various supermarket locations.
- The system should support **browser-based access** for administrators, providing remote access to the system for tasks like inventory management, sales tracking, and reporting.

5.5 Business Rules

5.5.1 Pricing Rules

- The system must allow for **dynamic pricing** based on various factors such as promotions, discounts, and customer loyalty. For example:
  - **Discounts** should be applied automatically based on predefined criteria (e.g., "Buy 1, get 1 free").
  - **Seasonal pricing** should be configurable to account for fluctuations in product prices, such as for holidays or sales periods.

5.5.2 Inventory Management

- The system must ensure that **inventory levels** are updated in real-time as products are sold, returned, or restocked. If a product's stock reaches a **reorder threshold**, the system should trigger an alert to restock the item.
- **Stock levels** should be checked automatically during transactions to prevent over-selling (i.e., selling more items than are in stock).
- The system should allow for **inventory adjustments** (e.g., manual stock counting, stock reconciliation) and track these changes with an audit trail.

5.5.3 Discount and Promotion Rules

- The system should support complex **discount logic**, such as applying a discount only if specific conditions are met (e.g., a customer must buy a certain quantity of a product to receive a discount).
- **Coupon codes** should be easily configurable, with options for **one-time use**, **percentage off**, or **fixed amount off**.
- The system should ensure that **promotions** and discounts cannot be combined unless explicitly allowed (e.g., "Cannot combine this discount with any other offer").

5.5.4 Return and Refund Policies

- The system must support **returns and refunds** based on specific store policies, such as allowing returns within a defined period (e.g., 30 days).
- The system should ensure that returned items are **restocked correctly**, and the inventory is updated to reflect the return. The system should also automatically process **refunds** or generate store credit as needed.

5.5.5 Tax Calculation Rules

- The system should support **taxation rules** based on product type, geographic region, and customer status (e.g., tax exemptions for certain customers).
- The **tax rate** should be automatically applied based on the **store's location** or customer's delivery address, ensuring compliance with local tax laws

5. Other Requirements

- Support and Maintenance
- Monitoring and Reporting
- Backup and Disaster Recovery

Appendix A: Glossary

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>