

Software Requirements Specification (SRS) for Supermarket Billing Application

1. Introduction

The Supermarket Billing Application aims to streamline the billing process at supermarkets by automating billing and payment systems. It provides functionalities for managing products, customers, billing, payments, and inventory, enhancing the overall shopping experience and efficiency at supermarkets.

2. Scope

The Supermarket Billing Application will include the following key features:

User authentication and authorization for administrators and cashiers.

Product management for adding, updating, and deleting products.

Customer management for adding, updating, and deleting customer information.

Billing functionalities for generating invoices and calculating total amounts.

Payment integration for accepting payments via various modes (credit/debit cards, cash).

Inventory management to track stock levels and manage product inventory.

Reporting functionalities for viewing sales data and generating reports.

3. Objectives

The objectives of the Supermarket Billing Application are to:

Improve efficiency and accuracy in the billing process at supermarkets.

Enhance the shopping experience for customers by providing a seamless checkout process.

Enable administrators to manage products, customers, and inventory effectively.

Provide insights into sales data and inventory status for better decision-making.

4. Functional Requirements

4.1 User Management

FR-1: The system shall provide user authentication and authorization for administrators and cashiers.

FR-2: Administrators shall be able to add, update, and delete cashier accounts.

FR-3: Cashiers shall only have access to cashier-specific functionalities.

4.2 Product Management

FR-4: Administrators shall be able to add, update, and delete products in the system.

FR-5: Products shall have attributes such as name, SKU, price, and quantity.

FR-6: Products shall be searchable by name or SKU.

4.3 Customer Management

FR-7: Administrators shall be able to add, update, and delete customer information.

FR-8: Customer information shall include name, contact details, and address.

4.4 Billing Functionality

FR-9: The system shall generate invoices for each transaction.

FR-10: Invoices shall include details of purchased items, quantities, prices, and total amount.

4.5 Payment Integration

FR-11: The system shall support payment via credit/debit cards and cash.

FR-12: Cashiers shall be able to process payments and update transaction status accordingly.

4.6 Inventory Management

FR-13: The system shall track stock levels for each product.

FR-14: Administrators shall be able to view and update inventory status.

4.7 Reporting

FR-15: The system shall provide reporting functionalities for viewing sales data and inventory status.

FR-16: Reports shall include information such as total sales, top-selling products, and low-stock items.

5. Non-Functional Requirements

NFR-1: The system shall be user-friendly and intuitive, requiring minimal training for users.

NFR-2: The system shall be secure, ensuring data confidentiality and integrity.

NFR-3: The system shall be scalable to handle a large number of transactions and users.

NFR-4: The system shall have low latency and respond quickly to user requests.

6. Use Cases

UC-1: Admin Login

UC-2: Cashier Login

UC-3: Add Product

UC-4: Update Product

UC-5: Delete Product

UC-6: Add Customer

UC-7: Update Customer

UC-8: Delete Customer

UC-9: Process Payment

UC-10: Generate Invoice

UC-11: View Inventory

UC-12: View Sales Report

7. User Stories

As an administrator, I want to be able to add, update, and delete products in the system.

As a cashier, I want to process payments using various payment methods (credit/debit cards, cash).

As an administrator, I want to generate sales reports to analyze business performance.

8. Additional Information

The application shall be developed using Java with Spring Boot framework for backend development.

PostgreSQL database shall be used for data storage.

SwaggerUI shall be integrated for API documentation.