\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Software Requirements Specification

For

**Supermarket Management System**

Group Name: G04

Members:

Mohamed Osama (202201723) Mostafa\_M01084@cic-cairo.com

Mostafa bakery (202201084) Mohamed\_A01723@cic-cairo.com

Abdelrahman Haithem (202201768) Abdelrahman\_i01768@cic-cairo.com

Sheref Hesham (202201769) Sheref\_A01769@cic-cairo.com

Zeyad Ashraf (202201782) Ziad\_A01782@cic-cairo.com

Instructor: Dalia Ahmed

Course: Web Engineering

Lab Section: Web

Lecturer: Asmaa AbuBakr

Date: 12-April-2025

**Table of Contents……………………………………………………………………..……….ii   
Revision History......................................................................................……………………...ii**

**1. Introduction**............................................................................................................................1

1.1 Purpose......................................................................................................................1

1.2 DocumentConventions..............................................................................................1

1.3 Intended Audience and Reading Suggestions...........................................................1

1.4 ProductScope……………………………………………………………………….1

1.5 References…………………………………………………………….....................1

**2. Overall Description.....................................................................................................2**

2.1 Product Perspective.…..………………...…......................................................2  
2.2 Product Functions.…...…………………………..………………………………...2  
2.3 User Classes and Characteristics………………………...………………………2  
2.4 Operating Environment……………………………………...……………………..2  
2.5 Design and Implementation Constraints..........................................................2  
2.6 User Documentation…………………………………………………..…………...2

2.7 Assumptions and Dependencies……………………………………..…………..2

**3. External Interface Requirements………………………………………………………..3**

3.1 User Interfaces................................................................................................3

3.2 Hardware Interfaces .......................................................................................3

3.3 Software Interfaces ...........................................................…………………….3

3.4 Communications Interfaces…………………………………………………..,…..3

**4. System Features.........................................................................................................4**

4.1 System Feature 1 ............................................................................................4

4.2 System Feature 2 (and so on).........................................................................4

**5. Other Nonfunctional Requirements..........................................................................5**

5.1 Performance Requirements.............................................................................5

5.2 Safety Requirements.......................................................................................5

5.3 Security Requirements....................................................................................5

5.4 Software Quality Attributes..............................................................................5

5.5 Business Rules……………………………………………………………………...5

**6. Other Requirements....................................................................................................6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  | 1.0 |
|  |  |  |  |

1. **INTRODUCTION**

**1.1 Identification**

With the increase in the number of customers, better management of supermarket operations is required.   
**The Supermarket Management System** focuses on improving the management of supermarkets in a city or town.

Imagine being able to check product availability, view discounts, or even place orders through your phone application, or managing inventory and sales efficiently!

The Integrated Supermarket Management System provides the convenience of managing inventory, processing sales, and analyzing customer data through a mobile app and web platform.

This system aims to streamline supermarket operations and enhance the shopping experience by leveraging technology.

* 1. **Document Conventions**

**Font**: Times New Roman, 16pt

**Headings**: Bold and numbered (e.g., 1.1, 1.2)

**Format**: This document follows the IEEE SRS template.

* 1. **Intended Audience and Reading Suggestions**
* **Project Stakeholders:** Supermarket owners, managers, and investors who are interested in understanding the system's capabilities and benefits.
* **Development Team:** Software developers and designers responsible for implementing the system.
* **System Analysts**: Professionals analyzing requirements and ensuring the system meets business needs.
* **End Users**: Cashiers, inventory managers, and customers interacting with the system.
  1. **Product Scope**

The Supermarket Management System aims to enhance the efficiency of supermarket operations by offering features such as:

* Product availability checks
* Online ordering and payment
* Inventory tracking and automated stock alerts
* Sales reporting and data analysis
  1. **References:** [**Super market srs**](https://www.studocu.com/row/document/bahria-university/software-engineering/super-market-srs-srs/42923943)
  2. **Definitions, Acronyms, and Abbreviations**
* **SRS**: Software Requirements Specification
* **Inventory**: A complete list of products available in the supermarket.
* **UI**: User Interface
* **UX**: User Experience
* **API**: Application Programming Interface

**2. Overall Description**

**2.1 Product Perspective**

The Supermarket Management System is a standalone software designed to help supermarkets manage daily operations, tracking inventory, processing sales.   
It can connect to devices like payment terminals and barcode scanner.  
The system works for supermarkets of all sizes, from small local shops to large retail stores.

**2.2 Product Functions**

* **Employee Management**:
* Manage employee roles, permissions, and schedules.
* Track attendance, working hours, and performance.
* **Customer**:
* Create and maintain customer profiles.
* Track purchase history and offer personalized discounts.
* Support reward points.
* **Inventory Management**:
* Add, update, and delete product details.
* Check stock levels and get alerts for low stock.
* Manage product categories, brands, and suppliers.
* **Checkout**:
  + Process customer purchases quickly and easily.
  + Calculate totals, apply discounts, and generate receipts.
  + Update stock levels after each sale.
* **Reporting and Analytics**:
* Create sales reports (daily, weekly, monthly).
* Track stock movement and profits.
* **Promotions and Discounts**:
* Create and manage special offers, discounts, and seasonal sales.
* Apply discounts automatically at the checkout.
* **Security and Access Control**:
* Provide role-based access to system features.
* Encrypt sensitive data to ensure security.
* **Order Management:**
* Record customer orders, both online and in-store.
* Track order status from processing to delivery or pickup
* **Product Expiry Tracking:**
* Monitor expiry dates for perishable goods.
* Alert staff about soon-to-expire products to reduce waste.
* **Feedback and Reviews:**
* Collect customer feedback at checkout or online.
* Analyze reviews to improve services and products.

**2.3 User Classes and Characteristics**

The system will have multiple user roles:

* Administrators:  
  Role: Manage system settings, user accounts, and security.  
  Tasks: Add/remove users, generate reports, oversee system performance.
* Managers:  
  Role: Supervise inventory, sales, and employee activities.  
  Tasks: Manage product lists, view sales reports, and approve discounts.
* Cashiers:  
  Role: Handle customer transactions at the POS.  
  Tasks: Scan barcodes, process payments, and print receipts.
* Inventory Staff:  
  Role: Track stock levels and update product information.  
  Tasks: Add new products, update quantities, and check expiry dates.
* User:

Role: Interact with the supermarket system to purchase products and access services.

Tasks:

* + Browse products (through kiosks, self-checkout screens, or online portals).
  + Scan items and pay using cash, card, or digital wallets (via self-checkout if available).
  + View loyalty points, discounts, and promotional offers.
  + Receive e-receipts or printed receipts for purchases.
  + Provide feedback or request assistance through customer support features.

### **2.4 Operating Environment**

The Super Market System will operate in the following environment:

* **Hardware**: Desktop computers, barcode scanners, receipt printers.
* **Operating System**: Windows servers for backend, Windows for administrative use.
* **Database**: MySQL.
* **Web and Mobile Interface**: Web-based dashboard with a mobile-friendly interface.
* **Network**: Local area network (LAN) and cloud-based connectivity for remote branches.

* 1. **Design and Implementation Constraints**
* **Hardware Constraints:**

The system must be compatible with standard hardware configurations, including:

* Minimum 4GB RAM.
* At least 500GB of storage for database and application files.
* **Software Constraints:**

The system must be developed using platform-independent technologies (e.g., Java, Python)

The database must be implemented using a relational database management system (RDBMS) such as MySQL

The system should integrate with third-party APIs for payment gateways (e.g., Fawry, Instapay) and inventory management tools.

* **Security Constraints**

All sensitive data (e.g., payment information, employee details) must be encrypted using industry-standard encryption protocols (e.g., AES-256).

* **Development Standards:**
* Agile methodology will be used for flexibility.
* Code must follow security and maintainability best practices.

**2.6 User Documentation**

* **User Manuals**: Guides for admins, managers, and cashiers.
* **Online Help**: Built-in assistance for quick support.
* **Training Materials**: Video tutorials and step-by-step guides.
* **API Docs**: For developers integrating third-party tools.
* **FAQs**: A list of common questions and answers to address user concerns.

**2.7 Assumptions and Dependencies**

* **Assumptions**:
* Reliable internet will be available for cloud features.
* Users have basic tech knowledge.
* Third-party systems (e.g., payment gateways) will remain stable.
* **Dependencies**:
* Integration with external systems (e.g., payment gateways, accounting software).
* Availability of hardware (e.g., barcode scanners, printers).
* Software libraries and frameworks (e.g., Java, Python, React).

1. **External Interface Requirements**

3.1 **User Interfaces**

1. **Sign-up Page**

* **Purpose:** Allows users to create accounts.
* **Fields:** Username, Email, Password.
* **Validation:** Error messages for invalid email formats, password strength requirements.
* **Buttons:** Sign *up*, Already *have an account? Log in*.

1. **Login Page**

* Purpose: Allows users to access their accounts.
* Components:

Email Field: Input for the user’s registered email.

Password Field: Secure input for the user’s password

* Login Button: Authenticates credentials and redirects to the Home page.

1. **Home Page**

* **Purpose:** Displays product categories, and a search bar.
* **Components:**
  + Navigation bar (Home, Cart, Promotions,).

1. **Cart Page**

* **Purpose:** Displays selected items and lets users update their order.
* **Components:**
  + Product name, quantity selector, price, and remove button.
  + Order now button.
* **Error Handling:** Displays "Cart is empty" message when no items are added.

1. **Checkout Page**

* **Purpose:** Collects shipping and contact details.
* **Fields:** Full Name, Address, Phone Number, Email.
* **Validation:** Ensures required fields are not empty and phone/email formats are correct.
* **Buttons:** Check Out

1. **Payment/Confirmation Page**

* **Purpose:** Finalizes the purchase by processing payment and confirming the order.
* **Options:** Credit Card, Cash on Delivery, Apple Pay.
* **Confirmation Message:** Displays "Order placed successfully!" along with the order number and estimated delivery date.
* **Buttons:** Confirm

1. **OTP Verification Screen**

* A text field for entering the OTP sent via SMS or email.
* Resend OTP option (Didn't receive a code? Resend").
* Buttons: Verify.

1. **Payment Confirmation Overlay**

* **Purpose:** Confirms successful payment.
* **Content:**
  + Message: "You Have Completed Your Payment"
  + Total amount: "Your total is …"
  + Thank you note: "Thank You For Purchasing From Fresh-market Hope you get back soon!"

1. **Locations Page**

* **Purpose:** Allows users to view, addresses.
* **Components:**
  + List of saved locations (each showing Name, Address)
  + *Set as* Default button for each location.
  + Edit icons.

1. **Offers Page**

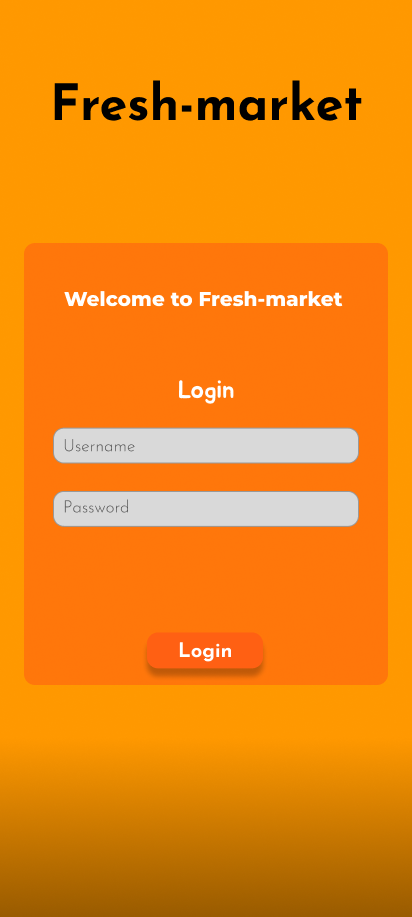
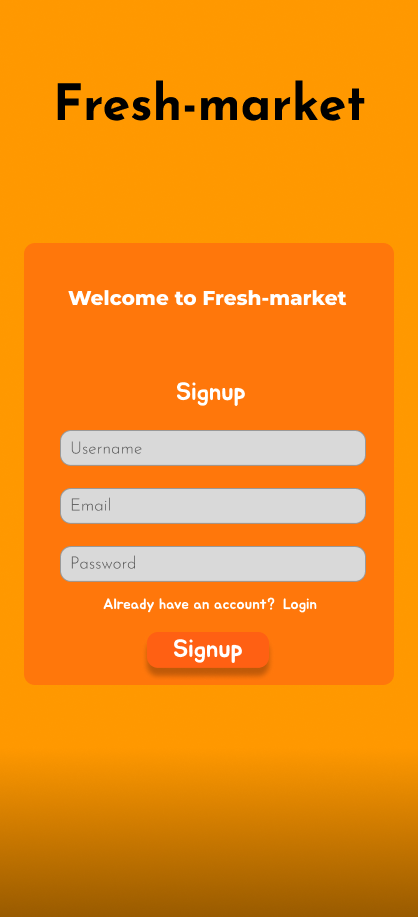
* **Purpose:** Displays products with ongoing discounts and special offers, allowing users to quickly view and purchase items at reduced prices.
* **Components:**
  + **Offer Cards:** Each card shows:
    - Product image
    - Product name
    - Original price
    - "Order Now" button to add the item to the cart

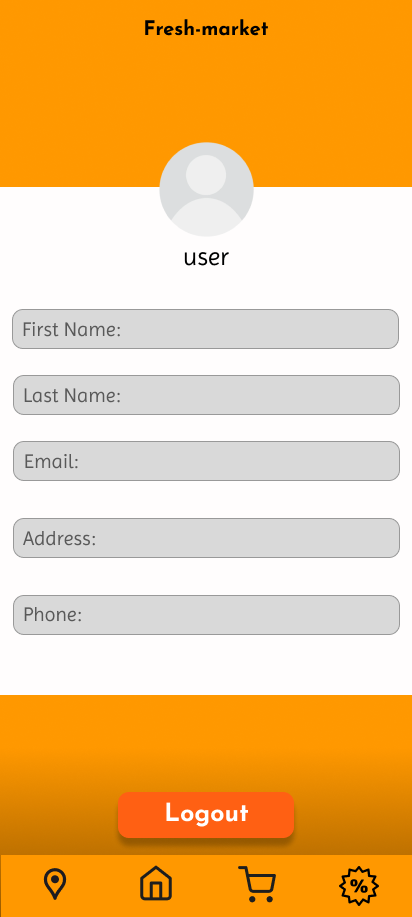
1. **Settings Page**

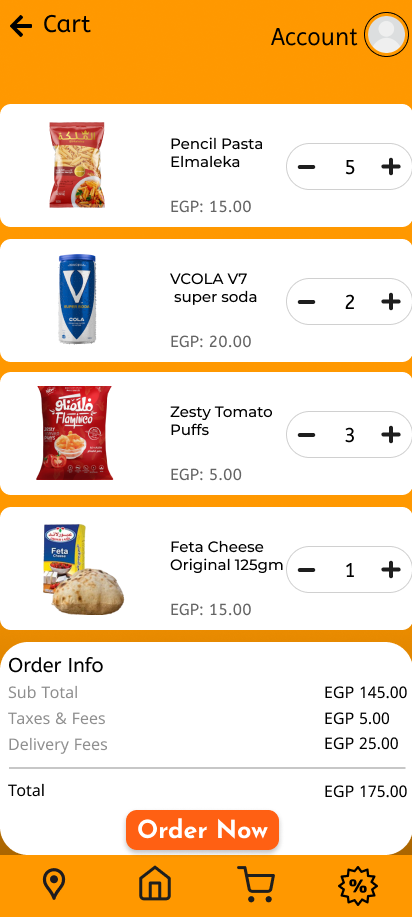
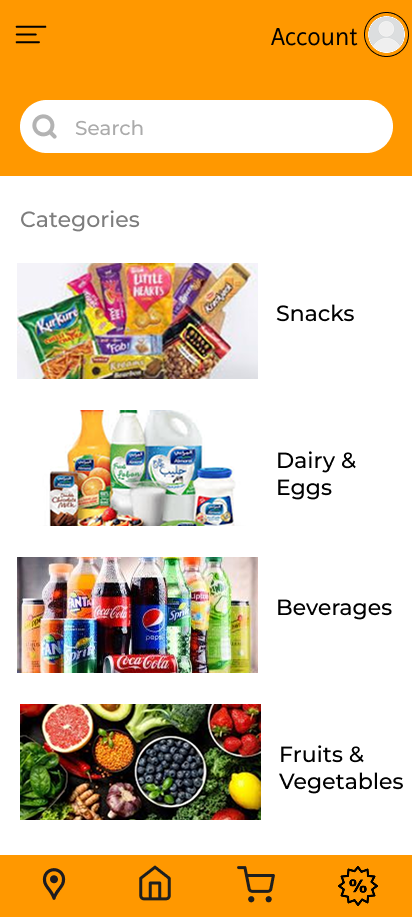
**Purpose:** Allows users to manage personal information and account settings.

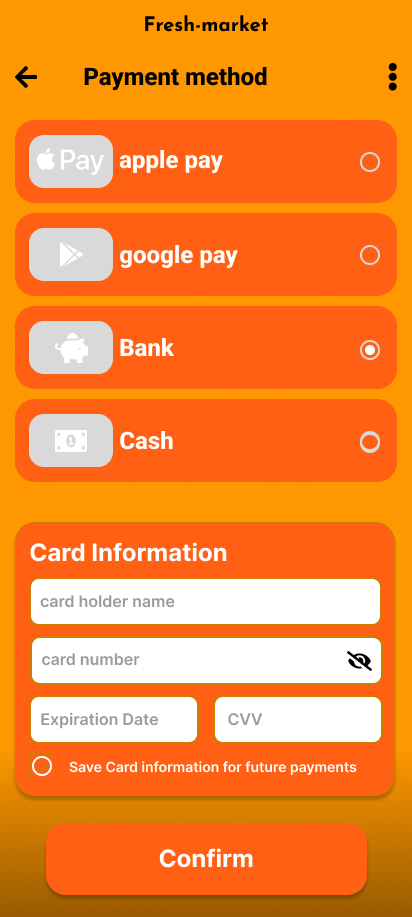
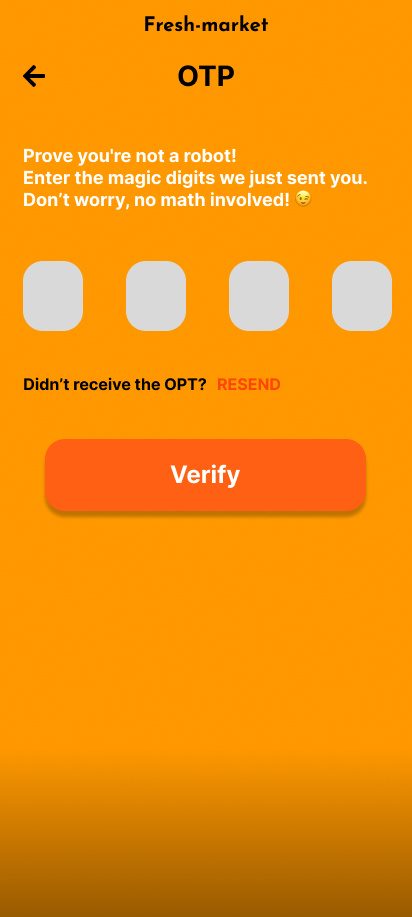
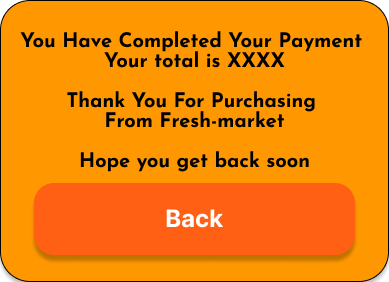
**Components**:

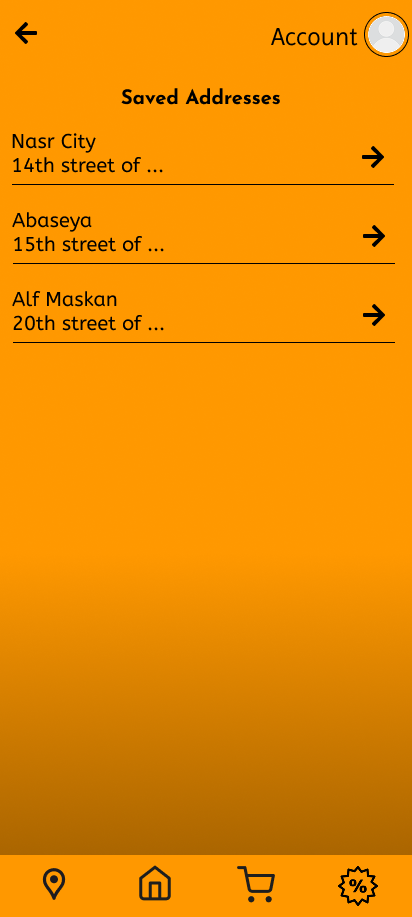
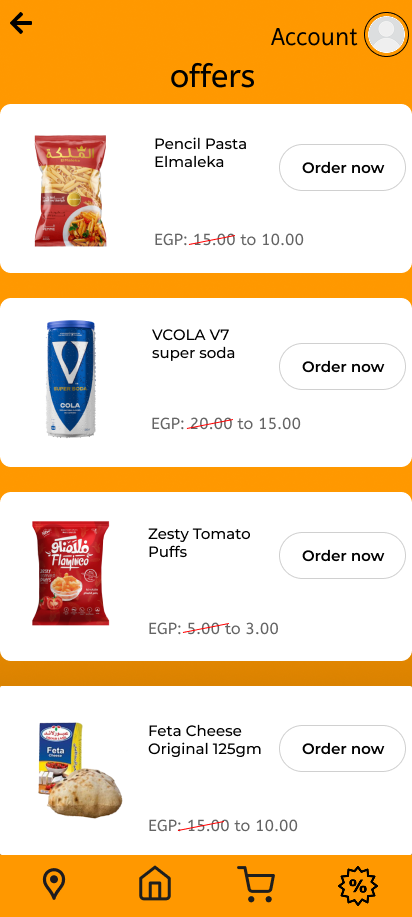
* Account Info: Displays user details such as name, email, and phone number.
* Saved Addresses: Lists previously saved delivery addresses
* Change Email: Allows users to update their registered email address.
* Change Password: Prompts users to enter their current password and set a new one.
* Log Out: Signs the user out of their account and returns them to the login/sign-up page.

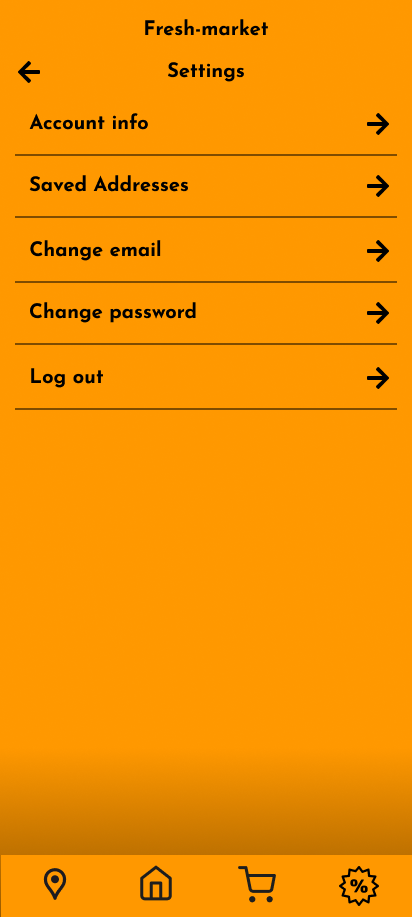




****

****

****

****

* 1. **Hardware Interfaces**

**Barcode Scanners:**

* + Used at checkout to scan product barcodes,  
    automatically fetching item details (name, price,  
    discount, etc.) from the database.

**Cash Drawers:**

* + Integrated with the Point of Sale (POS) system to open the cash drawer during cash transactions.

**Payment Terminals:**

* + Used for processing card payments (credit/debit cards).
  + Interface: Connected through USB or wireless networks (NFC or Bluetooth).

**Servers and Storage Devices:**

* + Hosts the system’s database and backend services.
  + Interface: Connected via LAN or cloud-based storage.  
    Data: Stores user data, product information, order history, and logs.

**3.3 Software Interfaces**

**Database Management System (DBMS):**

* Used to store and manage product data, user accounts, order history, and transaction records.
* Interface: SQL-based interactions for data retrieval, insertion, updating, and deletion.

**Operating System:**

* The system will run on Windows for the admin panel and POS software.
* Interface: Standard OS-level APIs for file access, network communication, and hardware control.

**Payment Gateway APIs:**

* Facilitates secure online payment processing.
* Interface: Communicates with external payment gateways (e.g., Fawry, Instapay) using HTTPS requests.
* Protocol: Encrypted data exchange using SSL/TLS.

**Payment Gateway Integration:**

* To process card and mobile payments securely.

**3.4 Communications Interfaces**

**Internet Communication:**

* Used for processing online payments, fetching live offers, and syncing data with cloud storage.
* Protocols: HTTPS for secure communication and data encryption using SSL/TLS.

**Email Notifications:**

* Used for sending account verification emails, order confirmations, and promotional offers.
* Protocols: SMTP (Simple Mail Transfer Protocol).
* Message Format: HTML emails for structured content and plain text fallback.

**Local Network Communication:**

* Connects hardware components (e.g., barcode scanners, cash drawers) to the POS system
* Protocols: TCP/IP for internal device communication.

1. **System Features**

**4.1 User Authentication and Management**

* **Description:** Enables customers and staff to register, log in, and manage their accounts.
* **Inputs:** Email, password, name, and contact information.
* **Outputs:** Access to user-specific dashboards.
* **Functional Requirements:**
  + Users can sign up or log in using valid credentials.
  + Admins can manage user roles (customer, cashier, manager).
  + Password reset functionality.

**4.2 Product Management**

* **Description:** Allows admins to add, update, or remove products from the inventory.
* **Inputs:** Product name, price, quantity, category, and barcode.
* **Outputs:** Updated product lists shown to customers.
* **Functional Requirements:**
  + Admins can add new products with all necessary details.
  + Real-time updates for stock levels.
  + Automatic notifications for low-stock items

**4.3 Shopping Cart and Checkout**

* **Description:** Facilitates adding/removing products from the cart and processing payments.
* **Inputs:** Selected products, quantities, and payment information.
* **Outputs:** Order summary and confirmation message.
* **Functional Requirements:**
* Customers can add, update, or remove items from their cart.
* Multiple payment options (credit card, cash, e-wallet).
* Secure checkout with order confirmation.

**4.4 Order and Transaction Management**

* **Description:** Tracks and processes customer orders and payments.
* **Inputs:** Order ID, product details, and customer info.
* **Outputs:** Order status (pending, completed) and receipts.
* **Functional Requirements:**
* Generate and display receipts.
* Track and update order statuses.
* Provide order history for customers.

**4.5 Inventory Management**

* **Description:** Maintains real-time tracking of product stock and updates automatically.
* **Inputs:** Product quantity changes (sales, restocking).
* **Outputs:** Updated stock levels and low-stock alerts.
* **Functional Requirements:**
* Track current stock levels per product.
* Alert admins for restocking when thresholds are reached.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The system should process customer transactions within **2 seconds**.
* Inventory updates must reflect in the database within **1 second** of a change.
* The system should support **up to 500 users at the same time** without performance slowdown.

**5.2 Security Requirements**

* User passwords must be encrypted using **SHA-256** or a similar algorithm.
* Sensitive data (payment info, personal details) should be transmitted over **HTTPS**.
* Admin-level access must be restricted through **role-based authentication**.

**5.4 Software Quality Attributes**

The Supermarket Management System will adhere to the following software quality attributes to ensure reliability, efficiency, and user satisfaction:

* **Reliability**: The system should function correctly under all conditions, ensuring 99.9% uptime and seamless recovery from unexpected failures**.**
* **Usability:** The interface will be user-friendly, intuitive, and accessible, requiring minimal effort for customers and staff to navigate.
* **Maintainability:** The code will follow modular design principles, making it easy for developers to debug, update, and extend features.
* **Performance:** All critical operations (like checkout and inventory updates) must respond within 2 seconds.
* **Portability:** The system will be accessible through web browsers (Chrome, Firefox, Safari, Edge) on both desktop and mobile devices.

**5.5 Business Rules**

* **Role-Based Access:**
  + **Customers** can register, log in, shop, and view their order history.
  + **Cashiers** can process in-store transactions and view product info but cannot edit product data.
  + **Admins** can view sales reports, and assign user roles.
  + **Managers** canmanage inventory and create offers.
* **Product and Inventory Management:**
  + Products marked **"out of stock"** remain visible but cannot be added to carts.
  + A **low-stock alert** is triggered when a product’s quantity drops below **10 units**.
* **Discounts and Offers:**
  + Discounts cannot exceed **50%** without admin approval.
  + Promotions have fixed **start and end dates** and cannot be edited once active.
* **Payments:**
  + Refunds require **manager approval** and are processed within **7 business days**.
  + Online payments must use secure gateways with **HTTPS** encryption.
* **Order Management:**
  + Orders can be canceled by customers within **1 hour** of placement.
  + Orders cannot be modified once marked as **"Shipped"**.
* **Security:**
  + Passwords must have a minimum of **8 characters**, including **1 number and 1 symbol**.
  + Admin-level accounts must enable **two-factor authentication (2FA)**.