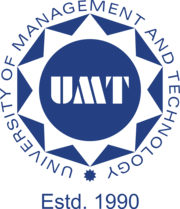
SOFTWARE ENGINEERING PROPOSAL

**MINAHIL ZUBAIR**

**F2023065367**

**W-4**



**Submitted to:** MISS KINZA SARDAR

**SUBJECT:** Software Engineering

**DEPARTMENT OF SOFTWARE ENGINEERING**

**UNIVERSITY OF MANAGEMENT**

**AND TECHNOLOGY**

**LAHORE, PAKISTAN.**

Online Shopping System

Software Requirements Specifications

1. Introduction
   1. **Purpose of this Document**

This document provides a description of the interfaces, key concepts, and overall purpose of the software project, “Online Shopping System.” This document intends to comprehend and clarify the requirements, also serving as the basis of further design.

* 1. **Scope of the Development Project**

This project automates the tasks of the buying and selling products online.

1. General Description
   1. **Actors Characteristics**

Following are the actors of this system.

1. Customer: This is the primary actor who interacts with the online shopping system to browse products, place orders, make payments, and manage their account.
2. Seller/Merchant: Sellers or merchants are individuals or businesses who list their products on the Online shopping system for sale. They interact with the system to manage their product listings, inventory, orders, and customer inquiries.
3. Delivery Personnel: Delivery personnel or courier services are responsible for delivering orders placed on Online shopping system to customers. They interact with the system to receive order details, track deliveries, and update delivery status.
4. Customer Service Representative: Customer service representatives assist customers with inquiries, complaints, returns, and other issues related to their shopping experience on the Online shopping system.
5. Administrator: Administrators manage and maintain the Online shopping system. They have access to administrative functionalities such as user account management, product categorization, system configuration, and reporting.
6. Payment Gateway: The payment gateway is an external actor responsible for processing payment transactions between customers and Online shopping system. It ensures secure and seamless payment processing for orders placed on the system.
   1. **Domain Overview**

This is an overview of the main processes the system will automate. For more details the Process Document can be consulted.

* + 1. **Online shopping process**

The process to buy and sell products is outlined below.

1. Customer logins or register on online shopping system to browser products.
2. Admin provides customers with a user-friendly interface to browse

products.

1. Merchants list their products on their respective stores, providing details and pricing.
2. Customers can search for products offered by merchants based on their interests.
3. Customers select desired products and place orders with merchants.
4. Customers provide their address and select preferred payment method during checkout.
5. Merchants view orders and prepare packaging for customer orders.
6. Admin shares pick-up details with delivery personnel.
7. Delivery personnel pick up the package, confirming the customer's address.
8. Customers track their order status and expected delivery date.
9. Admin notifies customers when the delivery personnel are en route.
10. Customers can leave reviews and initiate returns or exchanges if necessary.
11. Delivery personnel transfer payments to merchants after deducting system commission and delivery charges.
12. Specific Requirements
    1. **Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Serial No** | **Name** | **Description** |
| FR1 | Register/Login | Users should be able to register for a new account or log in to an existing one to access the platform securely. |
| FR2 | Add Merchant | Admin should be able to add new merchants to the platform to expand the seller base. |
| FR3 | Edit Merchant | Admin should have the capability to edit merchant details such as contact information or business details. |
| FR4 | Delete Merchant | Admin should be able to remove merchants from the platform if necessary. |
| FR5 | Create Store | Merchants should have the ability to create their own store within the platform to showcase their products. |
| FR6 | Delete Store | Merchants should be able to delete their store if they no longer wish to sell products on the platform. |
| FR7 | Add Products | Merchants should be able to add new products to their store, including details like descriptions and prices. |
| FR8 | Remove Products | Merchants should have the option to remove products from their store that are no longer available or relevant. |
| FR9 | Edit Products | Merchants should be able to edit product details such as descriptions, prices, or images. |
| FR10 | Search Products | Users must have the ability to search for products using keywords or filters to find items of interest quickly. |
| FR11 | View Products | Users should be able to view detailed product listings, including images, descriptions, and prices. |
| FR12 | Buy Product | Customers should be able to purchase products directly from the platform. |
| FR13 | Add to Cart Product | Customers should be able to add products to their shopping cart for later purchase. |
| FR14 | Add Customer Details | Customers should have the capability to add or edit their personal details such as shipping addresses. |
| FR15 | Edit Customer Details | Customers should be able to modify their personal information like contact details or billing address. |
| FR16 | Select Payment Method | Customers should have the option to choose their preferred payment method during checkout. |
| FR17 | Edit Payment Method | Customers should be able to edit their saved payment methods or add new ones. |
| FR18 | Dispatch Product | Admin should oversee the dispatch process to ensure timely delivery of products to customers. |
| FR19 | Track Order | Customers should be able to track the status of their orders from purchase to delivery. |
| FR20 | Give Feedback/Reviews | Users should have the ability to leave feedback and reviews for products and sellers. |
| FR21 | Customer Support | The platform should offer accessible customer support channels to address user inquiries or issues promptly. |

NON-FUNCTIONAL REQUIRMENTS

|  |  |  |
| --- | --- | --- |
| **NO** | **REQUIRMENT** | **DESCRIPTION** |
| **NFR1** | Performance | The system should respond to user actions within 2 seconds, even under peak load conditions. |
| **NFR2** | Scalability | The system should be able to handle a minimum of 10,000 simultaneous users without significant performance degradation. |
| **NFR3** | Availability | The system should be available 99.9% of the time, excluding scheduled maintenance windows. |
| **NFR4** | Security | All user data must be encrypted during transmission and storage using industry-standard encryption protocols. |
| **NFR5** | Usability | The user interface should be intuitive and easy to navigate, with clear instructions for using all features. |
| **NFR6** | Compatibility | The system should be compatible with major web browsers  (Chrome, Firefox, Safari, Edge) and devices (desktop, mobile). |

USE CASES

USE CASE #: UC – OSS – 001

**Use Case:** Register/Login

**Precondition:** The user accesses the online shopping system.

**Postcondition:** The user is logged into their account or successfully registers a new account.

**Basic Flow:**

1. The user accesses the online shopping system.
2. If the user is new, they select the "Register" option.
3. The system prompts the user to enter their registration details (e.g., username, email, password).
4. The user fills in the required information and submits the registration form.
5. The system verifies the entered information and creates a new account for the user.
6. If the user is already registered, they select the "Login" option.
7. The system prompts the user to enter their login credentials (username/email and password).
8. The user enters their credentials and submits the login form.
9. The system verifies the credentials and logs the user into their account.
10. The user gains access to their account dashboard.

**Alternate Flow:**

If the user enters incorrect credentials during login, the system displays an error message and prompts the user to re-enter their credentials.

# USE CASE #: UC – OSS – 002

**Use Case:** Add Merchant

**Precondition:** The administrator is logged into the admin panel of the online shopping system.

**Postcondition:** The new merchant is successfully added to the platform.

**Basic Flow:**

1. The administrator accesses the admin panel of the online shopping system.
2. The administrator selects the option to add a new merchant.
3. The system prompts the administrator to enter the details of the new merchant (e.g., name, contact information, business details).
4. The administrator fills in the required information and submits the form.
5. The system verifies the entered information and creates a new merchant account.
6. The new merchant receives login credentials and access to their dashboard.

**Alternate Flow:**

If the entered information is incomplete or invalid, the system displays an error message and prompts the administrator to correct the information.

# USE CASE #: UC – OSS – 003

**Use Case:** Edit Merchant

**Precondition:** The administrator is logged into the admin panel of the online shopping system.

**Postcondition:** The merchant's details are successfully edited.

**Basic Flow:** 1. The administrator accesses the admin panel of the online shopping system.

1. The administrator selects the option to edit a merchant.
2. The system displays the current details of the selected merchant.
3. The administrator modifies the necessary details (e.g., contact information, business details).
4. The administrator submits the updated information.
5. The system verifies the changes and updates the merchant's details accordingly.

**Alternate Flow:**

If the administrator encounters any issues while editing the merchant details, the system displays an error message and prompts the administrator to try again.

# USE CASE #: UC – OSS – 004

**Use Case:** Delete Merchant

**Precondition:** The administrator is logged into the admin panel of the online shopping system.

**Postcondition:** The merchant is successfully removed from the platform.

**Basic Flow:**

1. The administrator accesses the admin panel of the online shopping system.
2. The administrator selects the option to delete a merchant.
3. The system prompts the administrator to confirm the deletion.
4. The administrator confirms the deletion.
5. The system removes the merchant from the platform.

**Alternate Flow:**

If the administrator cancels the deletion process, the system retains the merchant on the platform.

# USE CASE #: UC – OSS – 005

**Use Case:** Create Store

**Precondition:** The merchant is logged into their account on the online shopping system.

**Postcondition:** The merchant's store is successfully created.

**Basic Flow:** 1. The merchant accesses their account dashboard on the online shopping system.

1. The merchant selects the option to create a new store.
2. The system prompts the merchant to enter the necessary details for the store (e.g., store name, description).
3. The merchant fills in the required information and submits the form.
4. The system verifies the information and creates the store for the merchant.

**Alternate Flow:**

If the merchant encounters any issues while creating the store, the system displays an error message and prompts the merchant to try again.

# USE CASE #: UC – OSS – 006

# 

**Use Case:** Delete Store

**Precondition:** The merchant is logged into their account on the online shopping system.

**Postcondition:** The merchant's store is successfully deleted from the platform.

**Basic Flow:** 1. The merchant accesses their account dashboard on the online shopping system.

1. The merchant selects the option to delete their store.
2. The system prompts the merchant to confirm the deletion.
3. The merchant confirms the deletion.
4. The system removes the merchant's store from the platform.

**Alternate Flow:**

If the merchant cancels the deletion process, the system retains the store on the platform.

# USE CASE #: UC – OSS – 007

**Use Case:** Add Products

**Precondition:** The merchant is logged into their account and has access to their store dashboard.

**Postcondition:** The new product is successfully added to the merchant's store.

**Basic Flow:**

1. The merchant accesses their store dashboard on the online shopping system.
2. The merchant selects the option to add a new product.
3. The system prompts the merchant to enter the details of the new product (e.g., name, description, price).
4. The merchant fills in the required information and submits the form.
5. The system verifies the information and adds the new product to the store.

**Alternate Flow:**

If the merchant encounters any issues while adding the product, the system displays an error message and prompts the merchant to try again.

# USE CASE #: UC – OSS – 008

**Use Case**: Remove Products

**Precondition:** The merchant is logged into their account and has access to their store dashboard.

**Postcondition:** The selected product is successfully removed from the merchant's store.

**Basic Flow:**

1. The merchant accesses their store dashboard on the online shopping system.
2. The merchant navigates to the product management section.
3. The merchant selects the option to remove a product.
4. The system displays the list of products available in the store.
5. The merchant selects the product they want to remove.
6. The system prompts the merchant to confirm the removal.
7. The merchant confirms the removal.
8. The system removes the selected product from the store.

**Alternate Flow:**

If the merchant cancels the removal process, the system retains the product in the store.

# USE CASE #: UC – OSS – 009

**Use Case:** Edit Products

**Precondition:** The merchant is logged into their account and has access to their store dashboard.

**Postcondition:** The selected product's details are successfully edited.

**Basic Flow:** 1. The merchant accesses their store dashboard on the online shopping system.

1. The merchant navigates to the product management section.
2. The merchant selects the option to edit a product.
3. The system displays the list of products available in the store.
4. The merchant selects the product they want to edit.
5. The system presents the current details of the selected product.
6. The merchant modifies the necessary details (e.g., description, price).
7. The merchant submits the updated information.
8. The system verifies the changes and updates the product's details accordingly.

**Alternate Flow:**

If the merchant encounters any issues while editing the product details, the system displays an error message and prompts the merchant to try again.

# USE CASE #: UC – OSS – 010

**Use Case:** Search Products

**Precondition:** The user accesses the online shopping system.

The user is presented with search results matching their query.

The user accesses the online shopping system.

The user navigates to the search bar or filters section.

1. The user enters keywords or selects filters to refine their search.
2. The system processes the search query.
3. The system retrieves products matching the search criteria.
4. The system displays the search results to the user.

**Alternate Flow:**

If no products match the search criteria, the system displays a message indicating no results found.

# USE CASE #: UC – OSS – 011

**Use Case:** View Products

**Precondition:** The user is browsing the online shopping system.

**Postcondition:** The user is presented with detailed information about the selected product.

**Basic Flow:**

1. The user accesses the online shopping system.
2. The user browses through product listings or search results.
3. The user clicks on a product to view more details.
4. The system retrieves the detailed information of the selected product.
5. The system displays the product details to the user.

**Alternate Flow:**

If the selected product is unavailable or removed, the system displays a message indicating the product is not found.

# USE CASE #: UC – OSS – 012

# 

**Use Case:** Buy Product

**Precondition:** The user is logged into their account and viewing a product they wish to purchase.

**Postcondition:** The user successfully completes the purchase transaction.

**Basic Flow:**

1. The user accesses the online shopping system.
2. The user navigates to the product they wish to buy.
3. The user clicks on the "Buy Now" or "Add to Cart" button.
4. The system prompts the user to confirm the purchase.
5. The user confirms the purchase.
6. The system processes the payment transaction.
7. The system updates the order status and notifies the user of the successful purchase.

**Alternate Flow:**

If the payment transaction fails, the system prompts the user to try again or choose an alternative payment method.

# USE CASE #: UC – OSS – 013

**Use Case:** Add to Cart Product

**Precondition:** The user is logged into their account and viewing a product they wish to add to their cart.

The selected product is successfully added to the user's shopping cart.

The user accesses the online shopping system.

The user navigates to the product they wish to add to their cart.

1. The user clicks on the "Add to Cart" button.
2. The system adds the selected product to the user's shopping cart.
3. The system displays a confirmation message indicating the product has been added to the cart.

**Alternate Flow:**

If the user encounters any issues while adding the product to the cart (e.g., connectivity issues), the system prompts the user to try again.

# USE CASE #: UC – OSS – 014

**Use Case:** Add Customer Details

**Precondition:** The user is logged into their account and accessing the checkout process.

**Postcondition:** The user's personal details are successfully added or updated for the order.

**Basic Flow:**

1. The user accesses the online shopping system.
2. The user navigates to the checkout process.
3. The system prompts the user to provide their personal details (e.g., shipping address).
4. The user fills in the required information.
5. The user confirms the details.
6. The system updates the user's personal details for the order.

**Alternate Flow:**

If the user encounters any issues while providing their personal details, the system displays an error message and prompts the user to correct the information.

# USE CASE #: UC – OSS – 015

**Use Case:** Edit Customer Details

**Precondition:** The user is logged into their account and accessing the account settings.

**Postcondition:** The user's personal details are successfully edited and updated.

**Basic Flow:**

1. The user accesses the online shopping system.
2. The user navigates to their account settings or profile section.
3. The system displays the user's current personal details.
4. The user selects the option to edit their details.
5. The user modifies the necessary information (e.g., contact details, billing address).
6. The user confirms the changes.
7. The system updates the user's personal details accordingly.

**Alternate Flow:**

If the user encounters any issues while editing their details, such as invalid input, the system displays an error message and prompts the user to correct the information.

# USE CASE #: UC – OSS – 016

**Use Case:** Select Payment Method

**Precondition:** The user is logged into their account and accessing the checkout process.

The user successfully selects their preferred payment method for the order.

The user accesses the online shopping system. The user navigates to the checkout process.

1. The system prompts the user to select a payment method.
2. The user chooses their preferred payment method (e.g., credit card, PayPal).
3. The system verifies the selected payment method.
4. The user confirms their choice.
5. The system updates the order with the selected payment method.

**Alternate Flow:**

If the selected payment method is not available or encounters an issue, the system prompts the user to choose an alternative payment method.

# USE CASE #: UC – OSS – 017

**Use Case:** Edit Payment Method

**Precondition:** The user is logged into their account and accessing the payment settings.

**Postcondition:** The user's payment methods are successfully edited and updated.

**Basic Flow:** 1. The user accesses the online shopping system.

1. The user navigates to their account settings or payment methods section.
2. The system displays the user's current payment methods.
3. The user selects the option to edit their payment methods.
4. The user modifies the necessary information (e.g., adds a new payment method, updates existing payment details).
5. The user confirms the changes.
6. The system verifies and updates the user's payment methods accordingly.

**Alternate Flow:**

If the user encounters any issues while editing their payment methods, such as invalid input or connectivity issues, the system displays an error message and prompts the user to correct the information or try again later.

# USE CASE #: UC – OSS – 018

**Use Case:** Dispatch Product

**Precondition:** The administrator is logged into the admin panel and has access to order management.

**Postcondition:** The order is successfully dispatched for delivery to the customer.

**Basic Flow:**

1. The administrator accesses the admin panel of the online shopping system.
2. The administrator navigates to the order management section.
3. The system displays the list of pending orders.
4. The administrator selects the order to dispatch.
5. The system verifies the order details and confirms the dispatch.
6. The system updates the order status to "Dispatched."
7. The system notifies the delivery personnel about the dispatched order.

**Alternate Flow:**

If there are no pending orders or the selected order cannot be dispatched for any reason, the system displays a message indicating the issue.

USE CASE #: UC – OSS – 019

**Use Case:** Track Order

**Precondition:** The customer is logged into their account and accessing their order history.

**Postcondition:** The customer can view the status and tracking information of their order.

**Basic Flow:** 1. The customer accesses the online shopping system.

1. The customer navigates to their account dashboard or order history section.
2. The system displays a list of the customer's previous orders.
3. The customer selects the order they want to track.
4. The system retrieves the status and tracking information of the selected order.
5. The system displays the order status and tracking details to the customer.

Alternate Flow:

If the order status or tracking information is unavailable for any reason, the system displays a message indicating that the information cannot be retrieved now.

# USE CASE #: UC – OSS – 020

**Use Case:** Give Feedback/Reviews

**Precondition:** The user has completed a purchase and has access to the product details page. **Postcondition:** The user's feedback or review is successfully submitted and visible to other users.

**Basic Flow:**

1. The user accesses the online shopping system.
2. The user navigates to the product details page of the purchased item.
3. The system displays the product details, including an option to leave feedback or review.
4. The user provides their feedback or review in the designated field.
5. The user submits their feedback.
6. The system verifies the feedback and updates the product's rating and reviews section accordingly.

Alternate Flow:

If the user encounters any issues while submitting their feedback, such as network errors, the system displays an error message and prompts the user to try again.

# USE CASE #: UC – OSS – 021

**Use Case:** Customer Support

**Precondition:** The user requires assistance with an issue related to the online shopping system. **Postcondition:** The user's inquiry or issue is successfully addressed by the customer support team.

**Basic Flow:** 1. The user accesses the online shopping system.

1. The user navigates to the customer support section or contacts the support team through available channels (e.g., live chat, email, phone).
2. The system connects the user to a customer support representative.
3. The user explains their inquiry or issue to the representative.
4. The representative investigates the matter and helps or resolves the issue.
5. The user confirms their satisfaction with the resolution.

**Alternate Flow:**

If the user is unable to reach a customer support representative or is dissatisfied with the help, the system escalates the issue to higher-level support or provides alternative contact options.

# Key Components

## User Management Component

* Register/Login (FR1)
* Add Customer Details (FR14)
* Edit Customer Details (FR15)
* Customer Support (FR21)

## Merchant Management Component

* Add Merchant (FR2)
* Edit Merchant (FR3)
* Delete Merchant (FR4)
* Create Store (FR5)
* Delete Store (FR6)
* Add Products (FR7)
* Remove Products (FR8)
* Edit Products (FR9)

## Product Management Component

* Search Products (FR10)
* View Products (FR11)
* Give Feedback/Reviews (FR20)
* Order Management Component
* Buy Product (FR12)
* Add to Cart Product (FR13)
* Dispatch Product (FR18)
* Track Order (FR19)

## Payment Management Component

* Select Payment Method (FR16)
* Edit Payment Method (FR17)
* Payment Gateway Integration (Actor) **Delivery Management Component**
* Delivery Personnel Interaction (Actor)

# Relationships Between Components

* User Management Component interacts with Merchant Management Component for managing merchants and stores.
* User Management Component interacts with Order Management Component for customer orders.
* Order Management Component interacts with Payment Management Component to process payments.
* Order Management Component interacts with Delivery Management Component for dispatching and tracking orders.
* Product Management Component interacts with Order Management Component for orderrelated product details.
* Product Management Component interacts with User Management Component for customer reviews and feedback.

# Interfaces

**User Management Interface:** Provides user registration, login, and profile management services.

**Merchant Management Interface:** Provides merchant and store management services.

**Product Management Interface:** Provides product search, view, and feedback services.

**Order Management Interface:** Provides order creation, cart management, and tracking services.

**Payment Management Interface:** Provides payment processing services.

**Delivery Management Interface:** Provides order dispatch and tracking services.

# System Architecture Diagram

Here is a visual representation of the system architecture diagram:

Interfaces

**U**

**ser**

**M**

**anagement**

**C**

**omponent**

-

Register/Login

-

Add

Customer Details

-

Edit Customer Details

-

Customer Support

**Order Management**

**Component**

-

Buy Product

-

Add to Cart Product

-

Dispatch Product

-

Track Order

**Merchant Management**

**Component**

-

Add/Edit/Delete Merchant

-

Create/Delete Store

-

Add/Remove/Edit

Products

**Delivery**

**Management**

**Component**

-

Delivery Personnel

Interaction

**P**

**ayment**

**M**

**anagement**

**C**

**omponent**

-

Select Payment Method

-

Edit

Payment Method

**Product**

**Management**

**Component**

-

Search Products

-

View Products

-

Give Feedback/Reviews

# Explanation

* The User Management Component handles user-related functionalities such as registration, login, and profile management.
* The Merchant Management Component is responsible for managing merchants, their stores, and products.
* The Product Management Component deals with product search, viewing details, and handling feedback.
* The Order Management Component manages the entire order lifecycle, including cart management, purchasing, and tracking.
* The Payment Management Component processes payments and interacts with the payment gateway.
* The Delivery Management Component manages the logistics of delivering products to customers.

Each component interacts with others through well-defined interfaces, ensuring low coupling and high cohesion, facilitating efficient and maintainable system architecture.

# Key Classes and Relationships

## User

* Attributes: userID, name, email, password, address, phoneNumber
* Methods: register(), login(), editProfile()

## Customer (inherits from User)

* Attributes: customerDetails
* Methods: addCustomerDetails(), editCustomerDetails()

## Merchant (inherits from User)

* Attributes: merchantID, storeList
* Methods:addStore(),editStore(), deleteStore(), addProduct(), editProduct(), removeProduct()

## Store

* Attributes: storeID, storeName, merchantID, productList
* Methods: createStore(), deleteStore()

## Product

* Attributes: productID, productName, description, price, stockQuantity, category, merchantID
* Methods: viewProduct(), addProduct(), editProduct(), removeProduct()

## Order

• Attributes: orderID, customerID, productList, orderStatus, paymentStatus, trackingNumber • Methods: createOrder(), addToCart(), checkout(), trackOrder()

## Payment

* Attributes: paymentID, orderID, paymentMethod, amount, paymentStatus
* Methods: selectPaymentMethod(), processPayment()

## Delivery

* Attributes: deliveryID, orderID, deliveryStatus, deliveryPersonID
* Methods: dispatchOrder(), trackDelivery()

## Feedback

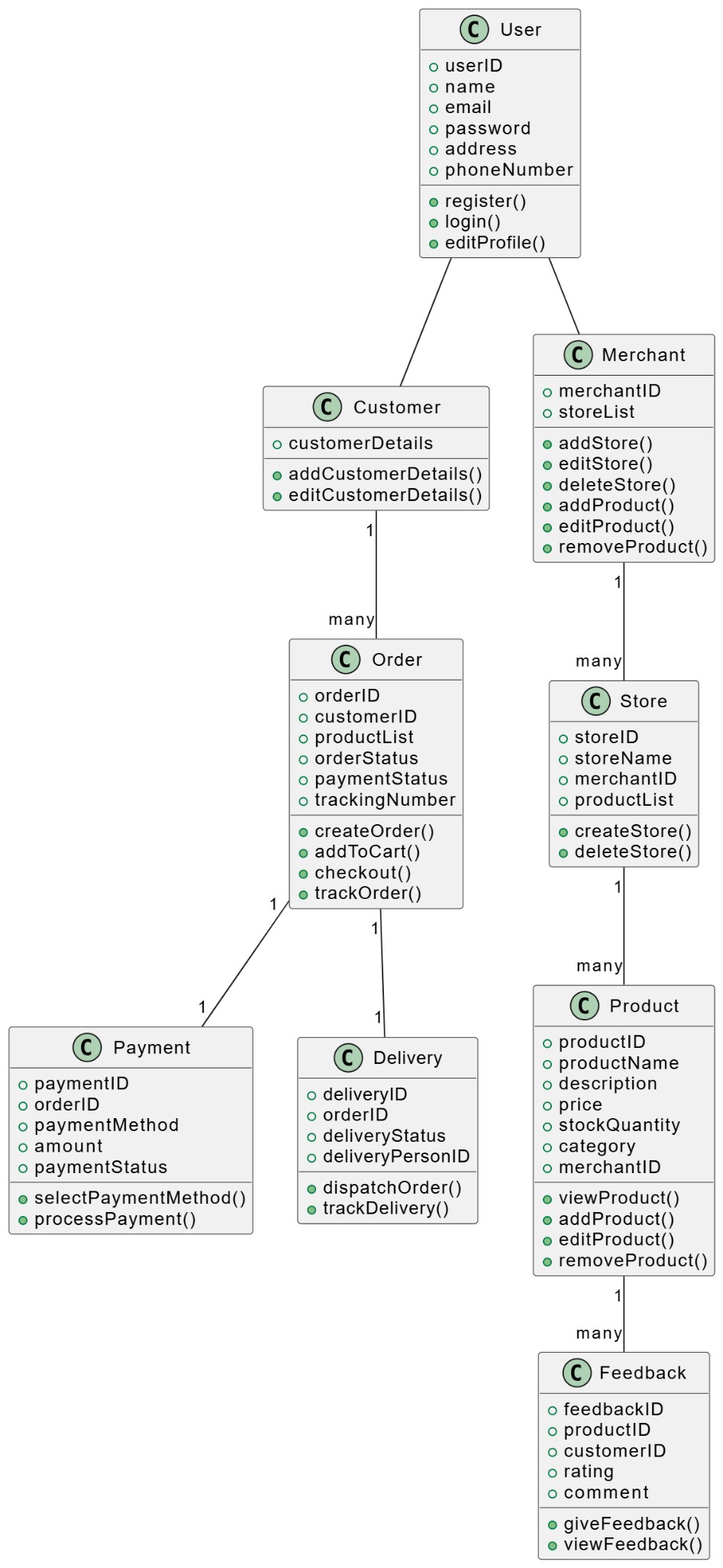
* Attributes: feedbackID, productID, customerID, rating, comment
* Methods: giveFeedback(), viewFeedback()

## Relationships

* User has a 1-to-1 relationship with Customer and Merchant.
* Merchant has a 1-to-many relationship with Store.
* Store has a 1-to-many relationship with Product.
* Customer has a 1-to-many relationship with Order.
* Order has a 1-to-1 relationship with Payment and Delivery.
* Product has a 1-to-many relationship with Feedback.

# Class Diagram

Below is the visual representation of the class diagram for the Online Shopping System:



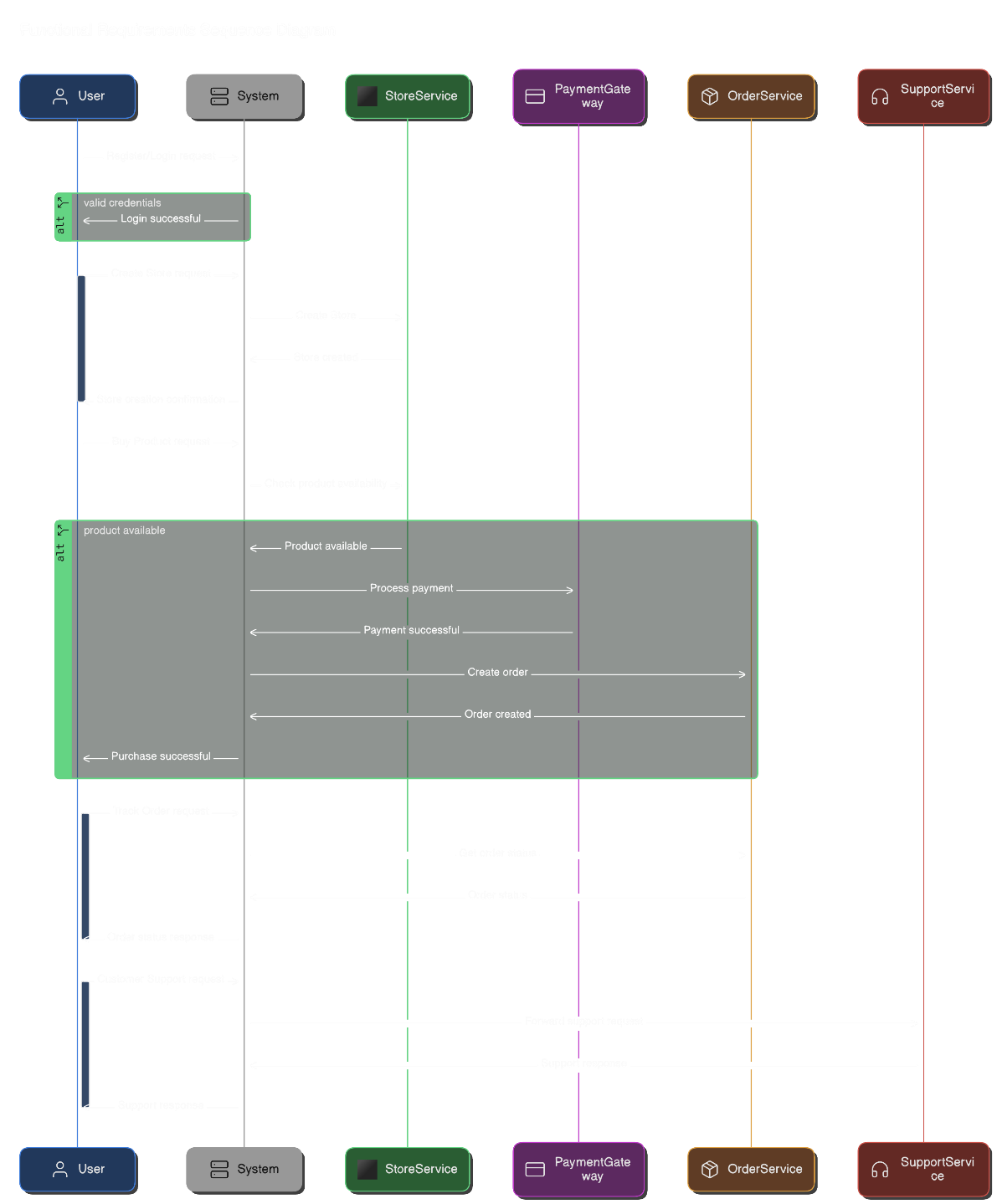
# SEQUENCE DIAGRAM

Sequence diagram for the following five important functional requirements:

* FR1: Register/Login
* FR5: Create Store
* FR12: Buy Product
* FR19: Track Order

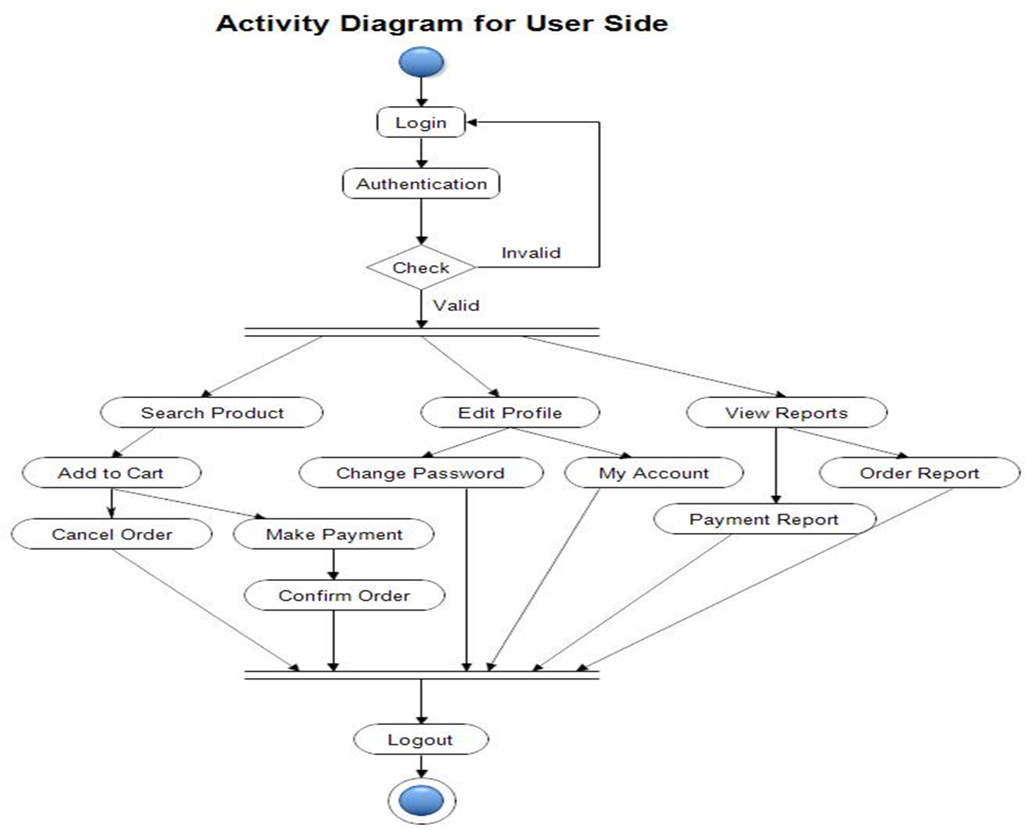
•

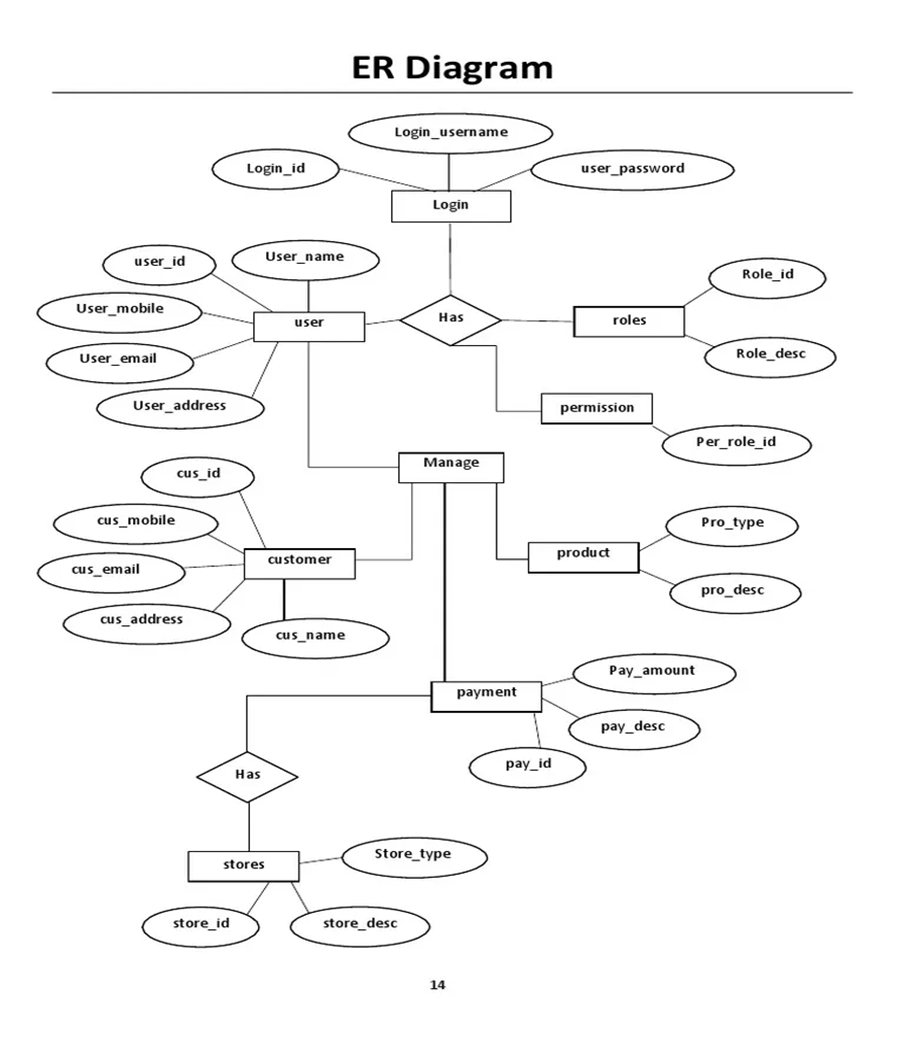
FR21: Customer Support



ACTIVITY DIAGRAM:

### **Flow of the Diagram:**

1. **Start** → Login/Register → [Decision] → Login → Proceed
2. **Admin Tasks** → Add/Edit/Delete Merchant
3. **Customer Actions** → Browse/Search/View Products → Add to Cart → Checkout
4. **Checkout** → Payment Method → Add/Edit Customer Details → Confirm Order
5. **Admin Dispatch** → Customer Track Order
6. **Feedback** → End

ERD DIAGRAM:

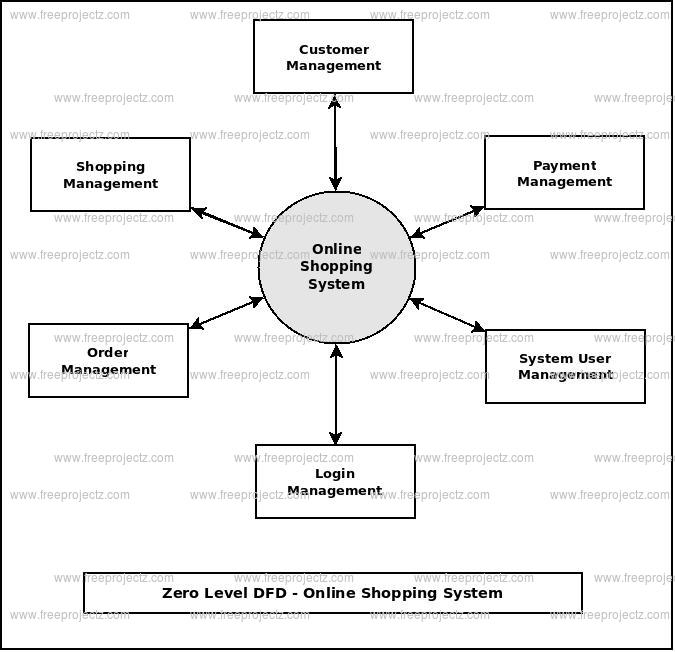
### **Relationships:**

* **User** to **Cart**: A user can have one cart (1-to-1).
* **User** to **Order**: A user can place many orders (1-to-many).
* **Order** to **Order\_Item**: An order can have multiple products (1-to-many).
* **Product** to **Order\_Item**: A product can appear in many orders (many-to-many via Order\_Item).
* **Store** to **Product**: A store can have many products (1-to-many).
* **Merchant** to **Store**: A merchant can own multiple stores (1-to-many).
* **User** to **Feedback**: A user can leave multiple feedbacks for products (1-to-many).
* **Product** to **Feedback**: A product can have multiple feedbacks (1-to-many).
* **Cart** to **Cart\_Item**: A cart can have many items (1-to-many).
* **Product** to **Cart\_Item**: A product can be added to many carts (many-to-many via Cart\_Item).

DFD DIAGRAM:

### **Relationships in the Online Shopping System:**

1. **User - Order**: One-to-Many (A user can place multiple orders).
2. **User - Cart**: One-to-One (A user has one cart).
3. **User - Feedback**: One-to-Many (A user can leave multiple feedbacks).
4. **Merchant - Store**: One-to-Many (A merchant can have multiple stores).
5. **Store - Product**: One-to-Many (A store can have multiple products).
6. **Product - Order\_Item**: One-to-Many (A product can be in multiple order items).
7. **Cart - Product**: Many-to-Many (A cart can contain multiple products, and a product can be in multiple carts).



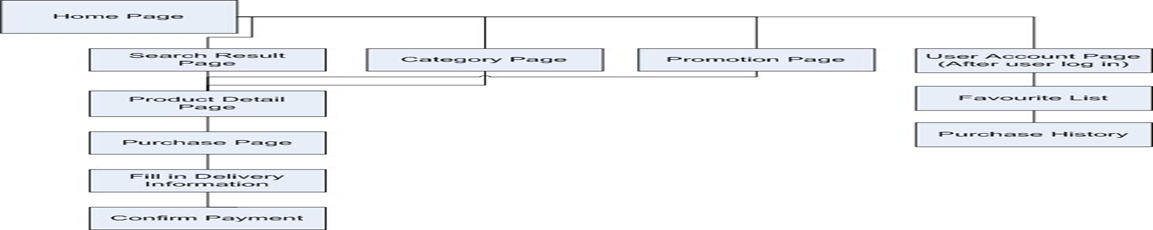
SYATEM ARCHITECTURE DIAGRAM:

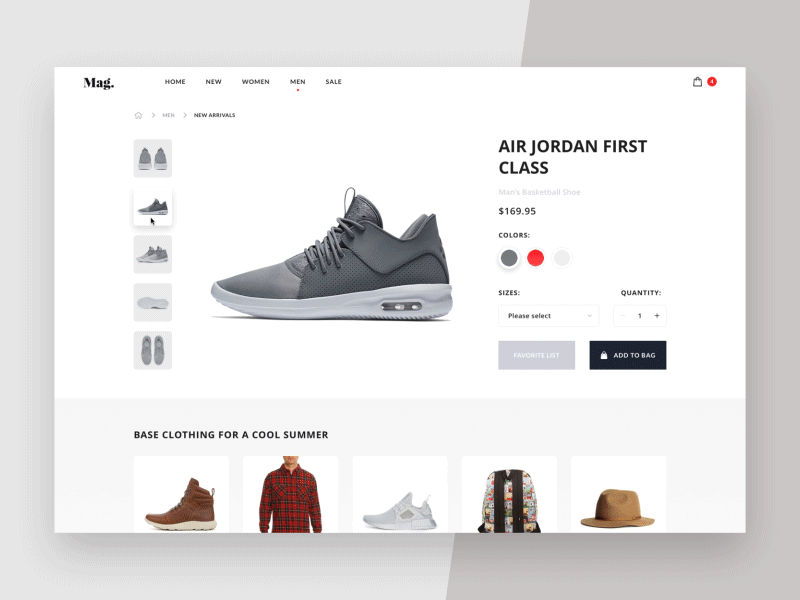
### Architecture Overview (High-Level Relationships):

* **Frontend** (User Interface) ↔ **Backend** (API Layer): User sends requests like browsing products, adding to cart, and making purchases.
* **Backend** ↔ **Database**: Stores and retrieves user data, product details, and order information.
* **Backend** ↔ **Payment Gateway**: Handles payment processing.
* **Backend** ↔ **Product Management Service**: Manages product data and inventory.
* **Backend** ↔ **Order Management**: Tracks orders, updates statuses.
* **Backend** ↔ **Shipping Service**: Coordinates delivery and shipment tracking.
* **Frontend** ↔ **Search Engine**: Provides product search functionality.
* **Admin Panel** ↔ **Backend/Database**: Admin performs CRUD operations for managing the system

### 

PROTOTYPES FOR ONLINE SHOPPING SYSTEM:





### Conclusion for Online Shopping Project

The development and implementation of an online shopping system provides a powerful platform for businesses to reach a global audience and offer customers the convenience of shopping from anywhere at any time. Throughout this project, several critical components were integrated to ensure a seamless and efficient user experience, alongside robust backend functionality.

Key takeaways from this project include:

1. **User-Centric Experience**: The system ensures a user-friendly interface, allowing customers to browse products, manage their cart, securely check out, and track their orders with ease. Features like personalized recommendations and a smooth checkout process enhance user engagement and satisfaction.
2. **Scalability & Flexibility**: The backend architecture was designed with scalability in mind, allowing for easy integration of new features and the capacity to handle growing traffic and transactions as the business expands. Modular components, such as product management, order management, and payment services, ensure the system is adaptable to future business needs.
3. **Security**: Strong emphasis was placed on security, particularly regarding user data protection and payment transactions. Technologies like encryption, secure payment gateways, and user authentication mechanisms (e.g., JWT, OAuth) help mitigate security risks, ensuring that both customer data and transactions remain safe.
4. **Real-Time Integration**: The integration of various external services—such as payment gateways (PayPal, Stripe), shipping APIs (FedEx, UPS), and search engines—ensures real-time processing and efficient delivery of services. This also helps maintain the accuracy of inventory and order status, improving the overall shopping experience.
5. **Admin Management**: The inclusion of an administrative panel allows business owners to manage products, track orders, and analyze customer data. This functionality streamlines the backend operations, enabling better decision-making and operational efficiency.
6. **Performance & Reliability**: The system architecture was designed to ensure high performance, low latency, and reliability. By using cloud services, load balancers, and caching mechanisms, the system can handle high traffic during peak times, ensuring smooth and uninterrupted service for customers.

### Future Enhancements:

While the current system is functional and effective, there is always room for improvement. Future enhancements could include:

* **AI and Machine Learning**: Incorporating AI for personalized product recommendations and customer behavior analysis.
* **Mobile Optimization**: Expanding the system's capabilities for mobile users, potentially developing native apps for iOS and Android.
* **Advanced Analytics**: Implementing more advanced data analytics tools for better insights into customer behavior, sales trends, and inventory management.
* **Multi-Language and Multi-Currency Support**: Expanding the system’s reach by supporting multiple languages and currencies to cater to a more diverse audience.

In conclusion, this online shopping system serves as a comprehensive solution for e-commerce, balancing both customer-facing features and backend functionalities. It not only enhances the shopping experience but also provides businesses with a scalable and secure platform to manage and grow their online presence.