**SYSTEM REQUIREMENT SPECIFICATION DOCUMENT FOR ONLINE SHOPPING MART**

**ABSTRACT:**

This application  facilitates virtual shopping for customers, offering them the convenience of browsing and purchasing items from the mart through online platforms. It is divided into two modules: one to allow consumers to perform purchases and another for storekeepers to manage inventory. A key position in database maintenance is played by the system administrator, who manages operations including adding, changing, and removing products. Consumers interact with the database through an easy-to-use menu system that lets them browse products, add items to their carts, and easily place orders. The database instantly refreshes after every transaction, guaranteeing precise inventory control.

Additional perks for registered customers include the ability to use a shopping cart, which helps them arrange their selections and expedite the purchasing process. On the other hand, walk-in customers are unable to use the platform to make direct purchases; instead, they can only browse the things that are available. The system places a strong emphasis on security and gives transactional integrity and client data protection top priority. The application's overall objective is to provide a safe and intuitive online shopping experience that meets the various needs of both consumers and retailers.

**FUNCTIONAL REQUIREMENT:**

Functional requirement of this application has divided into two modules. One is for User and Another one is for Administrator.

**1. User Module Functional Requirements:**

**Search Products:** Users can search for products using criteria like name, category, price range, etc. Results should be accurate, relevant, and displayed in a timely manner.

**View Product Details:** Users can view detailed information about each product. Information includes descriptions, images, prices, available variants (sizes, colors), etc. The product page should be user-friendly and informative to aid in purchasing decisions.

**User Account Management:** Users can create new accounts or log in with existing credentials. Registered users have access to their order history, saved addresses, and payment methods. Users can update profile information such as contact details and password.

**Add to Wishlist:** Users can add products to a wishlist for future reference. The wishlist allows users to save and manage desired items separately from the cart.

**Add to Cart:** Users have the ability to add desired products to their shopping cart. The cart displays selected items, quantities, and prices. Users can easily view and manage items in their cart.

**Place Order:** Users can place orders for items in their cart through a straightforward process. The system provides clear order confirmation with details such as estimated delivery times and costs. Users receive order status updates, including order confirmation, shipping details, and delivery notifications.

**2. Administrator Module Functional Requirements:**

**Manage Product Database:** Admin should be able to add new products to the system, change the price, description, and availability of already-existing products, remove products from the system as needed, and define and update product specifications like colors, sizes, materials, and other features.

**User Management:** For the goal of providing customer service, administrators should have the ability to examine and manage user accounts, as well as create new accounts, edit user profiles, change passwords, and view user activity and order histories.

**Generate Reports:** The system ought to enable administrators to produce diverse reports for the purpose of business analysis. Reports may cover popular items, inventory levels, customer demographics, and sales trends.

**Manage order and send notification:** The system should allow administrators to view, update, and track all incoming orders, including order status changes and details. Upon order placement, automated notifications should be sent to customers via email or SMS, confirming their purchase and providing estimated delivery times. These notifications should also include links for order tracking, ensuring customers stay informed throughout the shipping process. Admins should have the ability to customize and monitor these notifications for efficient order management.

**NON FUNCTIONAL REQUIREMENTS:**

Non-functional requirements of the system known as "quality attributes" ensure the overall quality of a system:

**Usability:** Pays attention to documentation clarity, accessibility, interface design, and user interaction.

**Security:** Using encryption, access restrictions, and compliance, it guards against threats, unauthorized access, and breaches.

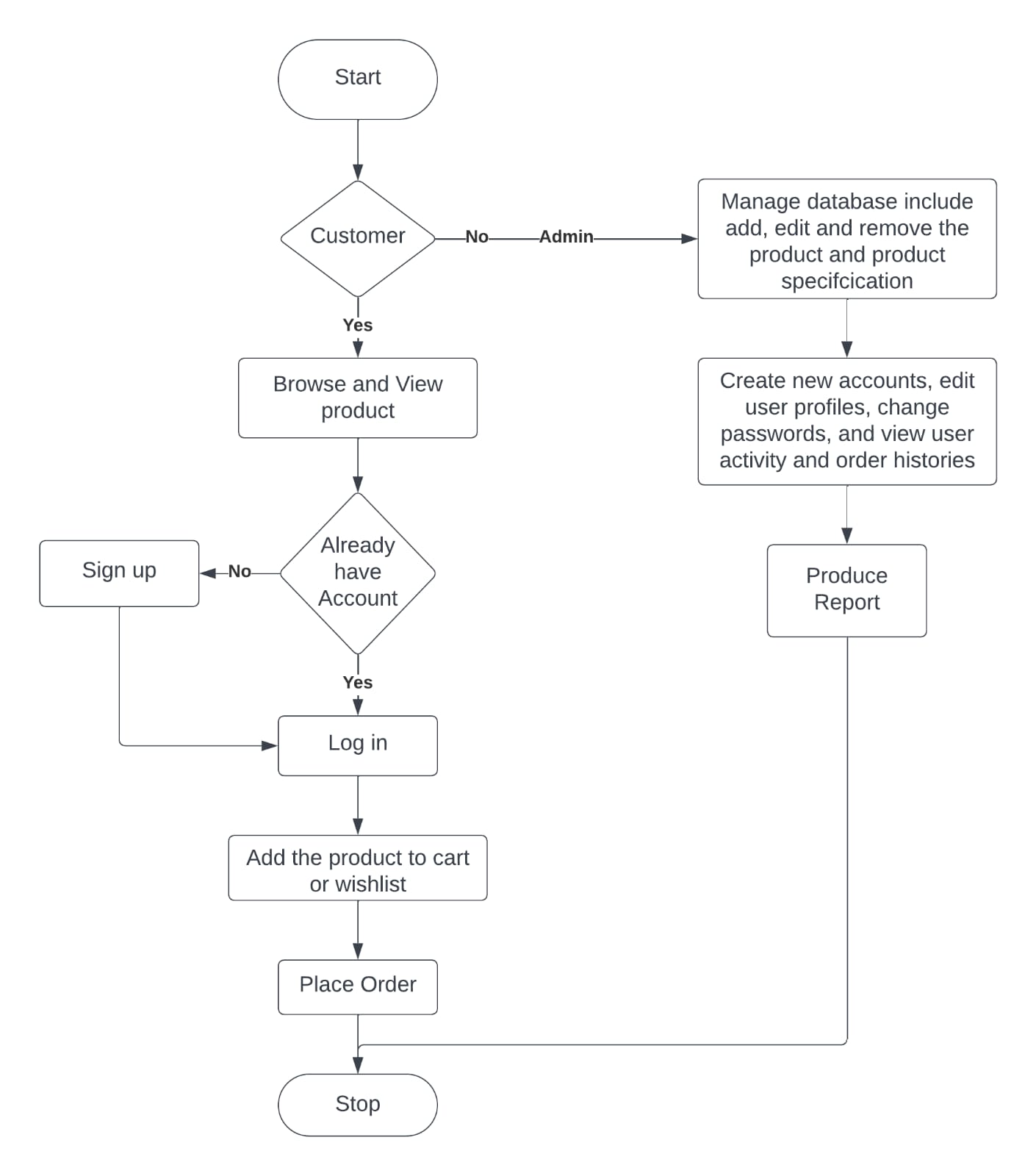
**Performance:** Ensures that, in certain circumstances, set metrics for responsiveness, throughput, and resource utilization are satisfied.

**Maintainability:** Assesses the system's lifetime accessibility for modification, updates, and repairs, as well as the readability of code and documentation.

**Scalability:** relates to the system's capacity to accommodate growing loads by allowing for the addition of resources through either horizontal or vertical scalability without compromising performance.

Together, these specifications guarantee that the system satisfies user needs, runs safely, functions well, is maintained, and may expand as needed.

**FLOW CHART:**

****

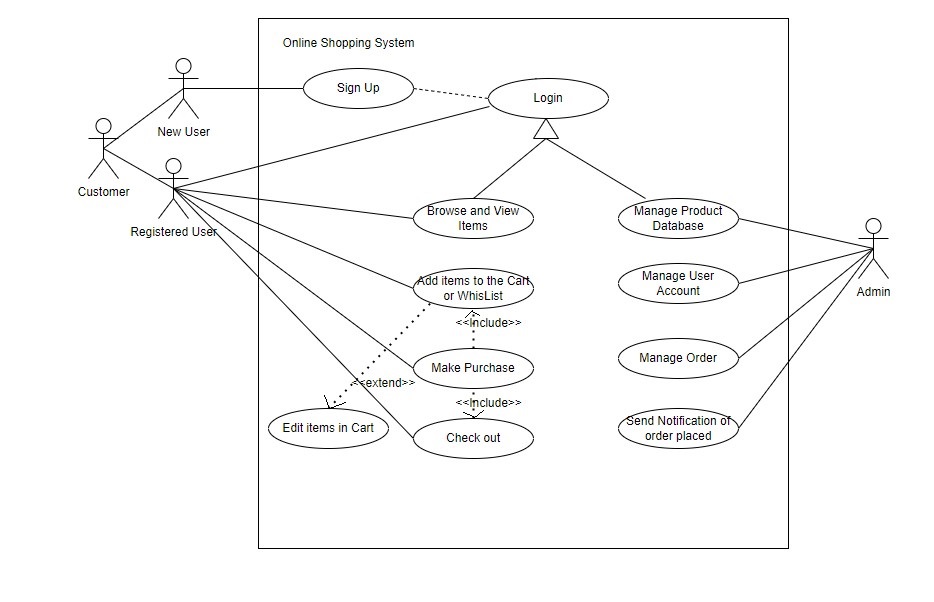
* 1. Login: The first step is login into the system based on whether the user is customer or admin.
  2. Customer Login: Customers can either log in with their registered credentials to access their accounts directly or sign up as new customers by providing their email and creating a password.
  3. Admin Login: Admin gain access to the admin portal upon login, enabling them to manage the database, user accounts, and product listings.
  4. Customer Actions: Once logged in, customers can browse products, view details, images, and prices. They can add items to their cart for future purchase, proceed to checkout, review selections, and make payments using various methods like credit cards or digital wallets.
  5. Admin Actions: Admin can add new products, update existing listings, and delete products. They create customer accounts, manage user profiles, reset passwords, and oversee user activity, including login times and order history.
  6. Order History and Activity: Customers can view past orders, check statuses, and track deliveries. Admin have a broader view, analyzing system-wide order history to track sales trends and popular items, aiding in inventory management and customer service.

**UML (Unified Modeling Language) Diagram:**

**Use case Diagram:**

Use case diagram is the type of Unified Modeling language diagram. It is a vital tool for system. It is a behavioral diagram that provides the visual representation of how users with the system. It serves as a blueprint for understanding the functionality of the system from the user’s perspective. It has three main components or part. Those are actor, use case and system boundary.

Actors are like external entities that interact with the system. These include users, other software and external hardware devices. Proper identification of actors are crucial for the proper modeling of the system. Use case are like scenes in the play. They represent the all specific function that the system does. System Boundary is a visual representation of scope or limit of the system. It defines what is inside the system and what is external to the system.

****

**Use Case Diagram Explanation:**

**Actors:**

1. Customer: The primary user of the online shopping mart.
2. Admin: The administrator who manages the system, including items, user accounts, and orders.

**Use Cases:**

1. Register if New Customer: Customer signs up with the system using their email.
2. Login if Registered User: Customer provides their credentials to access the system.
3. Browse Items: Customer can view the available items in the online store.
4. View Item Details: Customer can see detailed information about a specific item.
5. Add Item to Cart: Customer adds selected items to their shopping cart for potential purchase.
6. Add Item to Wishlist: Customer can add items to a wishlist for future reference.
7. Place Order: Customer confirms their shopping cart contents and proceeds to place an order.
8. Make Payment: Customer provides payment information to complete the purchase.
9. Manage Items (Admin): Admin can add, update, or remove items from the online store.
10. Manage User Accounts (Admin): Admin can manage customer accounts, such as approving new registrations or updating user details.
11. Manage Orders (Admin): Admin can view, update, or process customer orders.
12. Send Notification (Admin): Admin can send notifications to customers, such as order confirmation or shipment updates.

**Relationships:**

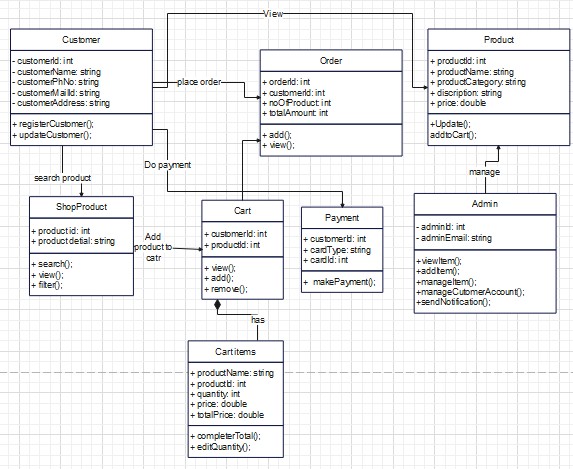
Customer:

1. Registers if the user is a New Customer to become a registered user.
2. Logs in if you are a Registered User to access the system after registration.
3. Browses Items, Views Item Details, Adds Item to Cart, and Adds Item to Wishlist, Places Order, and Makes Payment while interacting with the system.

Admin:

1. Manages Items to control the products available in the online store.
2. Manages User Accounts to oversee customer registrations and accounts.
3. Manages Orders to handle customer purchases and fulfillment.
4. Sends Notification to update customers about their orders or other relevant information.

**Class Diagram:**



**Class Diagram Description:**

1.Customer Class:

* Attributes: customerID, name, email, address, etc.
* Functions: viewProfile(), updateProfile(), placeOrder(), viewOrders(), etc.

2.ShopProduct Class:

* Attributes: productID, name, category, description, price, etc.
* Functions: searchProduct(), viewProductDetails(), filterProducts(), etc.

3.Cart Class:

* Attributes: customerID, cartItems
* Functions: viewCart(), addItem(), removeItem(), calculateTotal(), etc.

4.CartItem Class:

* Attributes: productID, quantity
* Functions: getProductDetails(), editQuantity(), calculateItemTotal(), etc.

(Note: Cart and CartItem are in a composition relationship, denoted by the diamond shape.)

5.Order Class:

* Attributes: orderID, customerID, productID, orderStatus
* Functions: addOrder(), viewOrderDetails(), updateOrderStatus(), etc.

6.Payment Class:

* Attributes: customerID, cardType, cardID, amount, etc.
* Functions: makePayment(), verifyPayment(), generateReceipt(), etc.

7.Product Class:

* Attributes: productID, name, category, description, price, etc.
* Functions: updateProductDetails(), addToCart(), viewDetails(), etc.

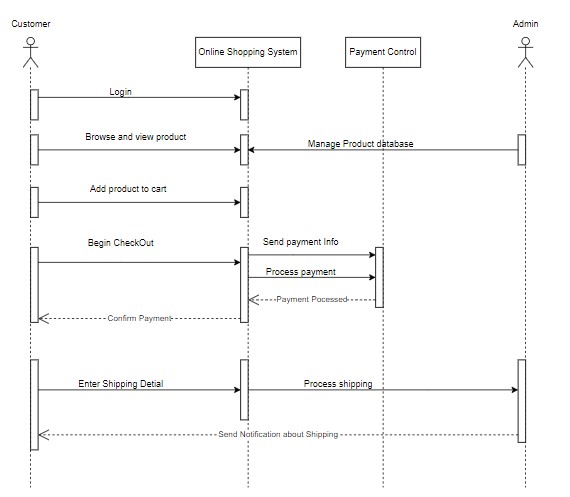
8.Admin Class:

* Attributes: adminID, email
* Functions: viewItems(), manageItems(), manageCustomerAccounts(), sendOrderNotification(), etc.

This diagram represents the structure of the classes in the system and their relationships. Each class encapsulates attributes and functions relevant to its responsibilities within the application.

**Sequence Diagram:**

A sequence diagram illustrates how objects interact in a particular sequence to achieve a specific functionality or use case within a system.



**High-Level Design:**

**Modules:**

* User Interface (UI) Module: Handles user interactions and displays information.
* Business Logic Module: Contains core application logic for products, carts, orders, and payments.
* Data Access Module: Manages interactions with the database for CRUD operations.
* Notification Module: Sends notifications to users and admins.
* Admin Interface Module: Allows admins to manage products, orders, customers, and notifications.

**Interactions:**

* Users browse products, add to cart, checkout, and make payments.
* Admins manage products, view orders, handle notifications, and manage customers.

**Low-Level Design:**

Classes:

* UI Classes: UserInterface, CartUI, OrderUI, PaymentUI, AdminUI
* Business Logic Classes: Customer, ShopProduct, Cart, CartItem, Order, Payment, Product, Admin
* Data Access Classes: CustomerDAO, ProductDAO, CartDAO, OrderDAO, PaymentDAO, AdminDAO
* Notification Classes: EmailNotification, SMSNotification

**Key Functions:**

* Customer: viewProfile, updateProfile, placeOrder, viewOrders
* ShopProduct: searchProduct, viewProductDetails, filterProducts
* Cart: viewCart, addItem, removeItem, calculateTotal
* CartItem: getProductDetails, editQuantity, calculateItemTotal
* Order: addOrder, viewOrderDetails, updateOrderStatus
* Payment: makePayment, verifyPayment, generateReceipt
* Product: updateProductDetails, addToCart, viewDetails
* Admin: viewItems, manageItems, manageCustomerAccounts, sendOrderNotification