OBJECT ORIENTED PROGRAMMING PROJECT

</>

ONLINE SHOPPING SYSTEM

IBRAHIM HILVANI

221504918

Introduction

Our online shopping system is a comprehensive platform meticulously crafted to enhance the purchasing experience for users spanning various product categories. Engineered with a focus on user accessibility and functionality, our system boasts an array of features tailored to simplify every step of the shopping journey. From dynamic discount capabilities to streamlined checkout processes and intuitive user interfaces, we strive to ensure that navigating our platform is both effortless and enjoyable. Administrators are empowered with robust tools for managing product listings, including implementing discounts, guaranteeing that the system remains agile and responsive to user demands. Through these meticulously designed functionalities, our system is dedicated to delivering a seamless and convenient online shopping experience for both customers and administrators alike.

Functional Requirements of the System:

User Authentication:

- Register new users and allow login with username and password.

Product Management:

- Add, update, and delete products in the inventory.

Product Browsing and Selection:

- Display product catalog with search and filter options.

Shopping Cart Management:

- Add, remove, and update items in the shopping cart.

Discount Management:

- Create and apply discounts to products.

Order Processing:

- Allow customers to proceed to checkout and complete purchases.

Order Tracking and History:

- Provide order tracking and history for customers.

User Account Management:

- Allow customers to update account information and manage passwords.

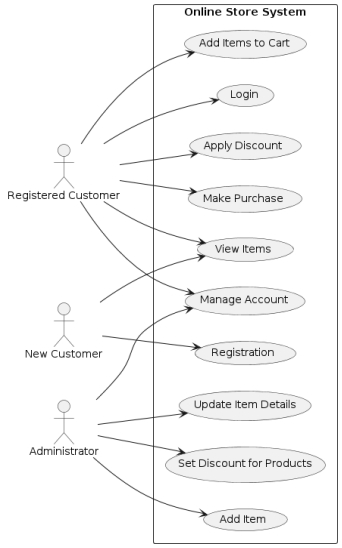
System Requirements:

- At least 4GB RAM is needed for optimal performance.
- Disk space of at least 2000 MB is required for storage purposes.

Use Case

Actors	Use Cases
Registered Customer	• Login • View items • Add items to cart • Make purchase • Manage account • Apply discount (optional)
New Customer	• Registration • View items
Administrator	• Manage account • Update item details • Add item • Set discount for products

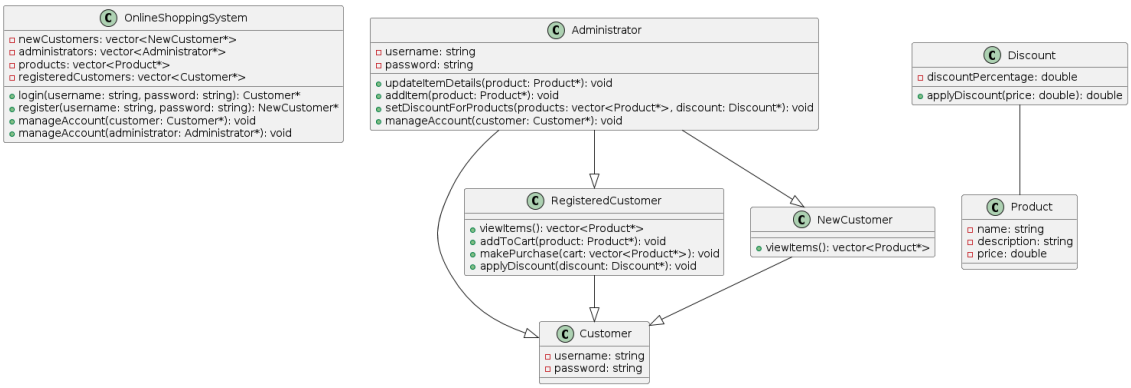
Use Case Diagram



Identifying Class

Class	Attributes	Behaviors
OnlineShoppingSystem	• newCustomers: vector<NewCustomer*> • administrators: vector<Administrator*> • products: vector<Product*> • registeredCustomers: vector<Customer*>	• login(username: string, password: string): Customer* • register(username: string, password: string): NewCustomer* • manageAccount(customer: Customer*): void • manageAccount(administrator: Administrator*): void
RegisteredCustomer	• username: string • password: string	• viewItems(): vector<Product*> • addToCart(product: Product*): void • makePurchase(cart: vector<Product*>): void • applyDiscount(discount: Discount*): void
NewCustomer	_____	• viewItems(): vector<Product*>
Administrator	• username: string • password: string	• updateItemDetails(product: Product*): void • addItem(product: Product*): void • setDiscountForProducts(products: vector<Product*>, discount: Discount*): void • manageAccount(customer: Customer*): void
Product	• name: string • description: string • price: double	_____
Discount	• discountPercentage: double	• applyDiscount(price: double): double

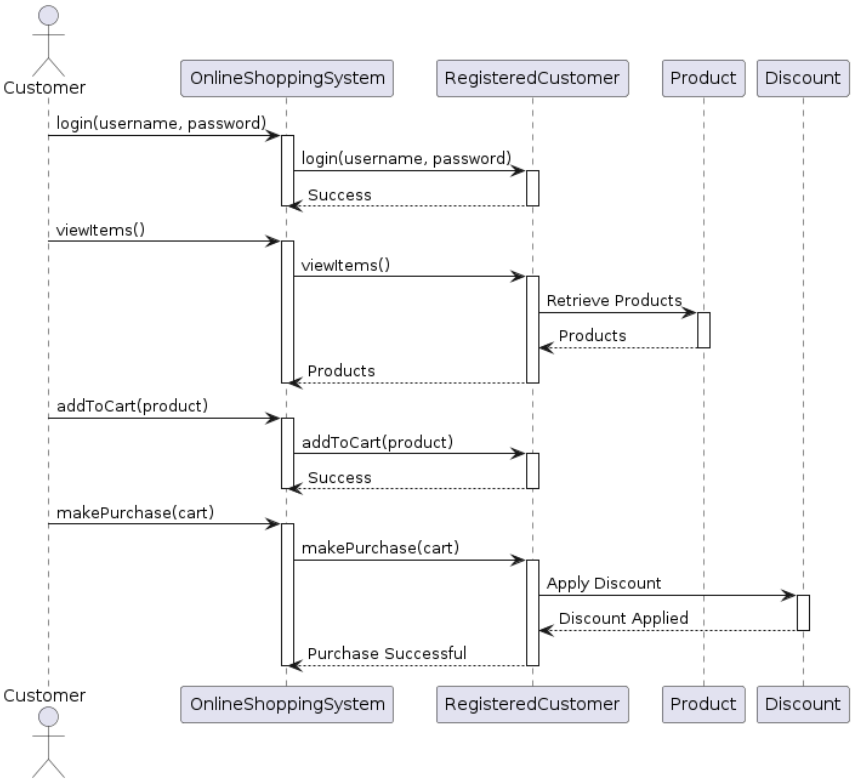
Class Diagram



Identifying Sequence

Action	System Response
login(username, password)	• OnlineShoppingSystem receives login request • OnlineShoppingSystem forwards login request to RegisteredCustomer • RegisteredCustomer processes login request • RegisteredCustomer sends success message to OnlineShoppingSystem
viewItems()	• OnlineShoppingSystem receives viewItems request • OnlineShoppingSystem forwards viewItems request to RegisteredCustomer • RegisteredCustomer retrieves products • RegisteredCustomer sends products to OnlineShoppingSystem
addToCart(product)	• OnlineShoppingSystem receives addToCart request • OnlineShoppingSystem forwards addToCart request to RegisteredCustomer • RegisteredCustomer processes addToCart request • RegisteredCustomer sends success message to OnlineShoppingSystem
makePurchase(cart)	• OnlineShoppingSystem receives makePurchase request • OnlineShoppingSystem forwards makePurchase request to RegisteredCustomer • RegisteredCustomer applies discount

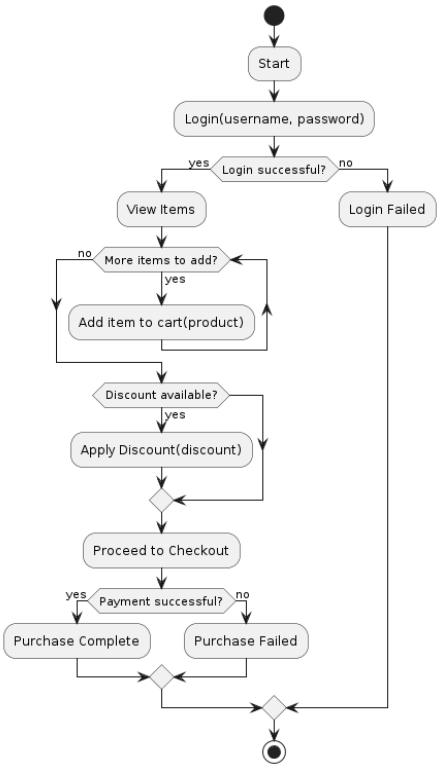
Sequence Diagram



Identifying Activity Diagram

Step	Action	Decision	Subsequent Action/Outcome
Start	Start the process		
Login	Login with username and password	Is login successful?	If successful, proceed to "View Items"; If failed, show "Login Failed" message
View Items	Browse available products		Proceed to select items to add to cart
Add item to cart	Add selected item to the shopping cart	Are there more items to add?	If yes, continue adding items; If no, proceed to "Apply Discount" (if available)
Proceed to Checkout	Move to the checkout process		Proceed to payment processing
Payment successful	Payment is processed successfully	Is payment successful?	If yes, show "Purchase Complete" message; If no, show "Purchase Failed" message
Purchase Complete	Display purchase confirmation		End of the process
Purchase Failed	Display purchase failure message		End of the process
Login Failed	Display login failure message		End of the process

Activity Diagram



In conclusion, the online shopping system project aims to provide a seamless and efficient platform for customers to browse, select, and purchase products. Through the analysis of requirements and the design of various diagrams such as use case, class, sequence, activity, and collaboration diagrams, we have established a comprehensive understanding of the system's functionality and interactions.

The system caters to different types of users, including registered customers, new customers, and administrators. Registered customers can log in, view items, add items to their cart, apply discounts, and make purchases. New customers can also register to become registered customers. Administrators have additional privileges such as managing accounts, updating item details, adding new items, and setting discounts.

The backbone of the system is the **OnlineShoppingSystem,** which coordinates the interactions between customers, products, discounts, and administrators. Throughout the system's operation, objects collaborate to fulfill user actions and ensure smooth transactions.

By employing various diagrams and models, we have provided a structured representation of the system's requirements, functionality, and interactions. This holistic approach facilitates effective communication among stakeholders, guides development efforts, and lays the groundwork for the implementation phase.

In summary, the online shopping system project demonstrates the application of software engineering principles and methodologies to design a robust and user-friendly e-commerce platform that meets the needs of customers and administrators alike.