

## e-commerce class diagram:

### ▣ Key classes and their attributes:

1. User: Represent a customer or administrator interacting with the system.

#### • Attributes:

- \* userID: Unique identifier for the user.
- \* username: User's login name.
- \* password: User's password.
- \* email: User's email address.

#### • Methods:

- \* register(): Allows a new user to create an account.
- \* login(): Authenticates a user.
- \* update Profile(): Updates user information.

2. Product: Represents an item available for purchase.

• Attributes:

\* ProductID: Main Unique identifier for the product.

\* name: Name of the product.

\* description: Detailed information about the product.

\* price: cost of the product.

\* stockQuantity: Number of item available.

• Methods:

\* updateStock(): Modify the stock quantity.

\* applyDiscount(): Applies a discount to the product price.

3. Category: Organizes products into groups.

• Attributes:

\* categoryID: Unique identifier for the category.



\* name: Name of the category.

\* description: Description of the category.

#### • Methods:

\* addProduct(): Adds a product to the category.

\* removeProduct(): Removes a product from the category.

4. ShoppingCart: Holds products a user intends to purchase.

#### • Attributes:

\* cartID: Unique identifier for the cart.

\* userID: Identifier linking the cart to a user.

#### • Methods:

\* addItem(): Adds a product to the cart.

\* removeItem(): Removes a product from the cart.

\* calculateTotal(): Compute the total cost of items in the cart.

5. Order: Records completed purchases.

• Attributes:

\* orderId: Unique identifier for the order

\* userId: Identifier linking the order to a user.

\* OrderDate: Date when the order was placed.

\* status: Current status of the order (e.g, pending, shipped).

• Methods:

\* placeOrder(): Finalizes the order.

\* cancelOrder(): Cancels the order.



6. Payment: Manages payment information for orders.

• Attributes:

\* paymentID: Unique identifier for the payment.

\* orderID: Identifier linking the payment to an order.

\* amount: Total amount paid.

\* paymentMethod: Method used for payment (e.g: credit card)

\* paymentStatus: Status of the payment (e.g: completed, pending)

• Methods:

\* ~~pro~~ processPayment(): Processes the payment for an order.

7. Shipping: Handles delivery details for orders.

• Attributes:

- \* shippingID: Unique identifier for the shipping records.
- \* orderID: Identifier linking the shipping to an order.
- \* shippingAddress: Delivery address.
- \* shippingStatus: status of the shipment.

• Methods:

- \* shipOrder(): Initiates the shipping process.

8. Admin: Manages the platform. Including the users and products.



## • Attributes:

- \* adminID: Unique identifier for the administrator.
- \* username: Admin's login name.
- \* password: Admin's password.

## • Methods:

- \* login(): Authenticates the admin.
- \* manageUser(): Manages users accounts.
- \* manageProduct(): Manages product listings.

## Relationships:

- \* User to Shopping Cart: A User has one shopping cart.
- \* User to Order: A user can place multiple orders.

\* Order to Product : An Order can include multiple Products. Products and a Product can be a part of multiple Orders (many-to-many relationship)

\* Shopping Cart to Product : A shopping cart can contain multiple products, and a product can be in multiple shopping carts (many-to-many relationship)

\* Order to Payment : An order has one payment.

\* Order to Shipping : An order has one shipping record.

\* Admin to user/Product/Order : An Admin can manage multiple Users, Products, or Orders.



## UML class diagram:

