**OBJECT MODELING FOR FASHION E-COMMERCE PLATFORM**

In my object model the classes represent these key entities

**Customer:** Represents users who browse and buy products.

**Product:** Represents the fashion items available for sale.

**Order:** Represents a customer’s order.

**Cart:** Represents the customer’s selected products for purchase.

**Payment:** Manages the payment process.

**Admin:** Manages the platform, including product and user data.

**Supplier:** Manages product inventory.

When it comes to attributes and methods,Each class has its own set of attributes(properties) and methods(behaviors).

**Customer**

* + **Attributes:** Customer ID, Name, Email, Password, Address, Payment Information
  + **Methods:** Register, Login, View Products, Add to Cart, Checkout, View Order History

**Product**

* + **Attributes:** Product ID, Name, Description, Price, Stock, Category
  + **Methods:** Get Product Details, Update Stock, Apply Discount (optional)

**Order**

* + **Attributes:** Order ID, Date, Customer, Products (List), Total Amount, Status
  + **Methods:** Create Order, Update Status, View Order Details

**Cart**

* + **Attributes:** Cart ID, Customer, Products (List), Total Price
  + **Methods:** Add Product, Remove Product, Update Quantity, Calculate Total

**Payment**

* + **Attributes:** Payment ID, Order, Amount, Payment Method
  + **Methods:** Process Payment, Verify Payment, Confirm Payment

**Admin**

* + **Attributes:** Admin ID, Name, Email
  + **Methods:** Manage Products, Manage Users, Manage Orders

**Supplier**

* + **Attributes:** Supplier ID, Name, Contact Info, Supplied Products
  + **Methods:** Manage Inventory, Restock Products, View Orders
* 1. **The Use of cases also for;**Specific interactions or functionalities that the system offers to its actors. For my fashion project, common use cases might be:
     + **Browse Fashion Products**
     + **Add to Cart**
     + **Checkout and Make Payment**
     + **View Order History**
     + **Search Products**
     + **Manage User Account**
     + **Update Product Listings (Admin)**
     + **Manage Inventory (Supplier)**
  2. **System Boundary:** Defines the scope of the system being modeled. Everything inside the boundary is within the system, and everything outside is external to it.

**Relationships:**

* 1. **Association:** Links actors to use cases they interact with.
  2. **Extend:** Shows optional behaviors that extend the base use case (e.g., "Apply Discount" could extend the "Checkout" process).
  3. **Include:** Shows that one use case includes behavior from another (e.g., "Checkout" includes "Make Payment").

**Sequence Diagrams:**

**Purpose:** Sequence diagrams illustrate the order in which interactions occur between different objects or actors over time to perform a specific use case.

**Key Elements:**

* + **Lifelines:** Represent the participants (actors or system components) involved in the interaction.
  + **Messages:** Show the flow of communication between participants (method calls, responses, or interactions).
  + **Activation Bar:** Shows the period during which a participant is actively engaged in the process.
  + **Arrows:** Represent the flow of control or data, showing the order of interactions.

**Example Sequence Diagrams for Fashion Project:**

**Browse Products Use Case:**

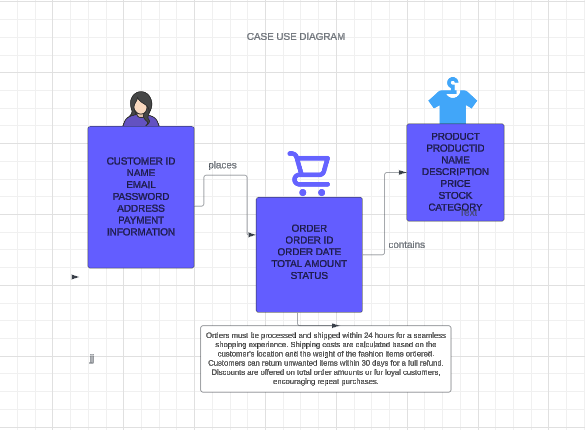
* + - **Actors:** Customer, Fashion E-commerce System.
    - **Steps:**
      1. Customer selects the "Browse Products" option.
      2. The system retrieves available products from the database.
      3. The system displays the products to the customer.

**Checkout and Make Payment Use Case:**

* + - **Actors:** Customer, Fashion E-commerce System, Payment Gateway.
    - **Steps:**

1.Customer clicks "Checkout."

2.The system confirms the cart items. 3.The customer enters payment details. 4.The system forwards payment data to the Payment Gateway. 5.The Payment Gateway confirms payment. 6.The system completes the order and displays confirmation to the customer.



NABAGGALA RACHEAL ROBINS M23B13/035 B20182