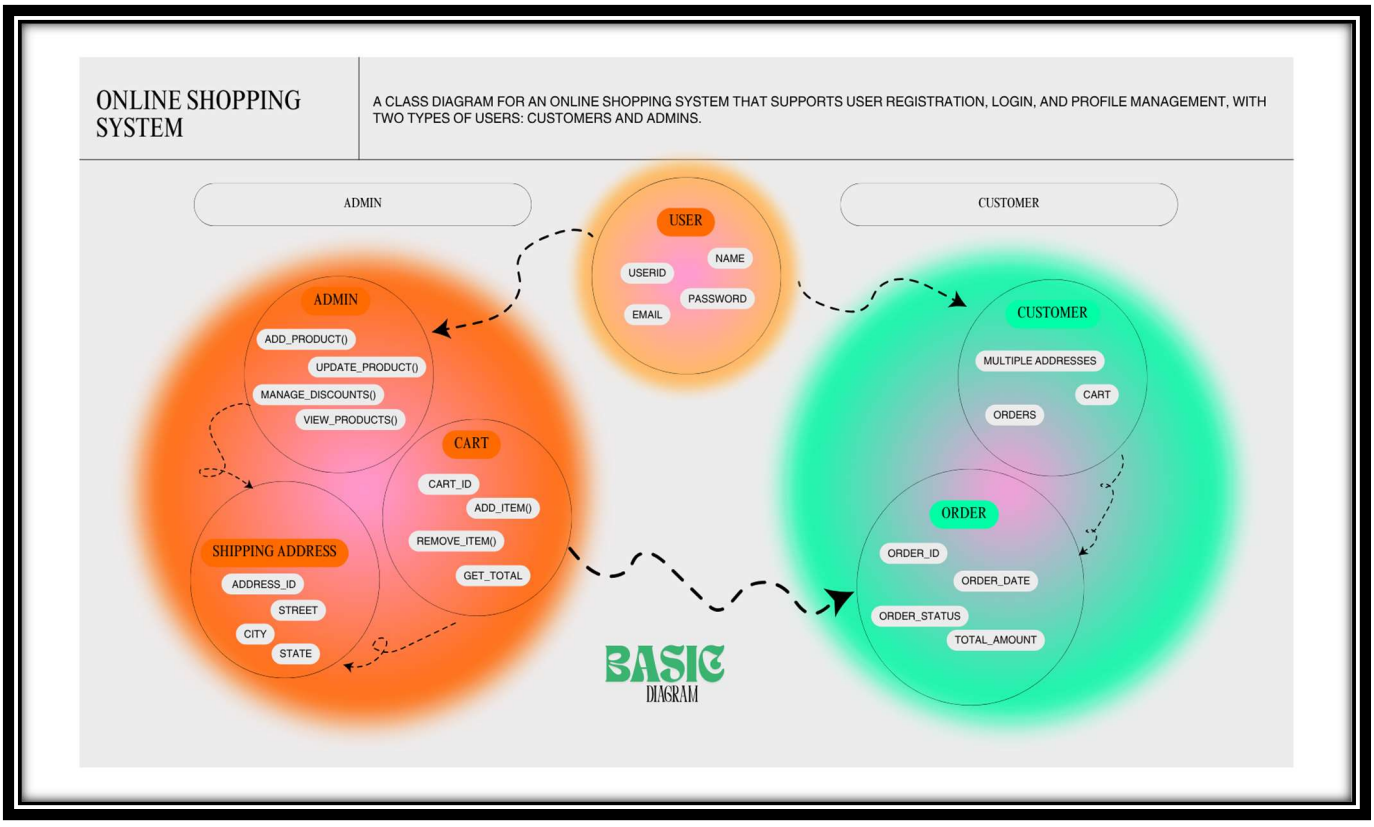
Online Shopping System:

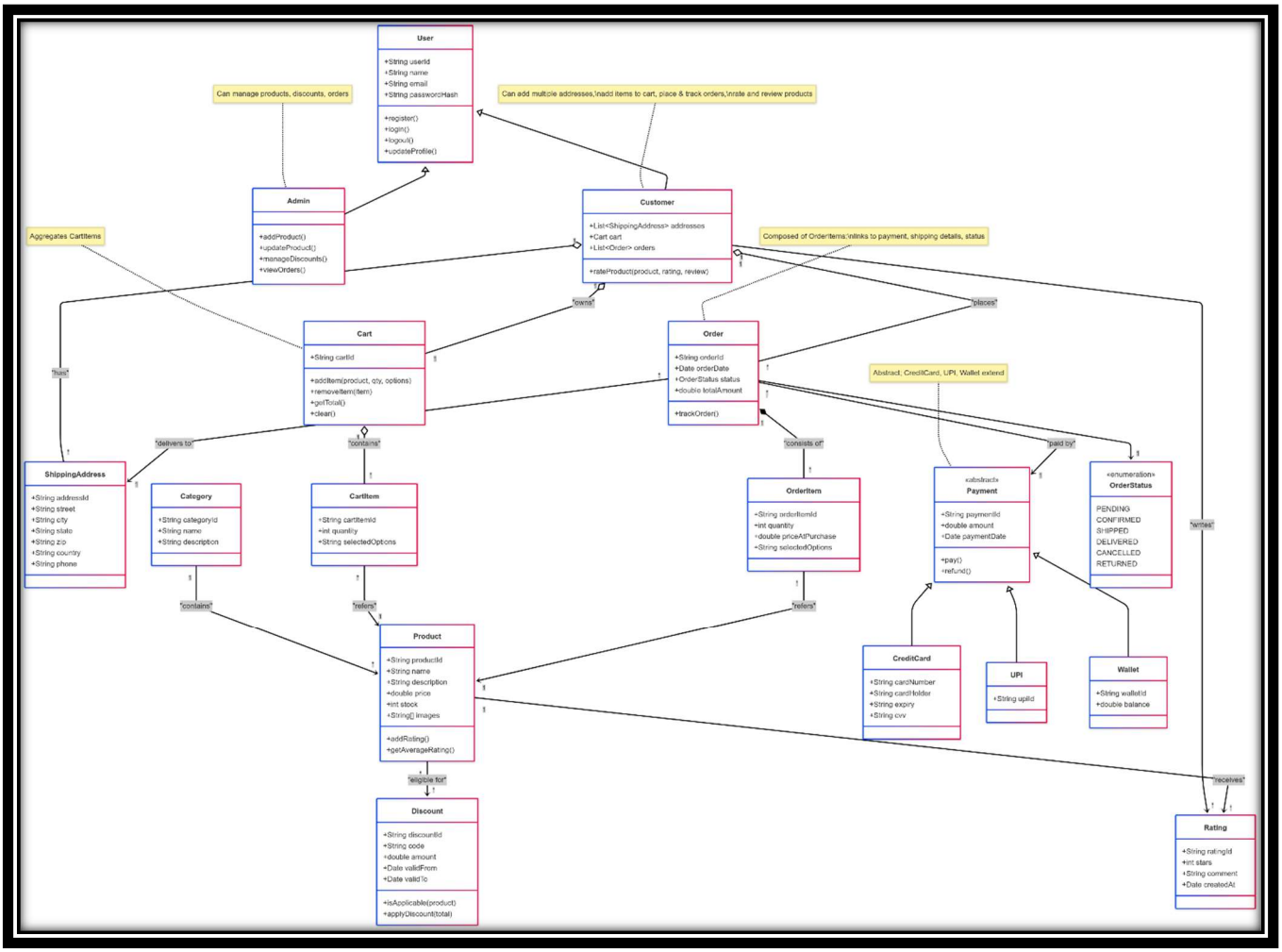
Design a detailed class diagram for an online shopping system that supports user registration, login, and profile management, with two types of users: customers and admins. Customers should be able to browse products by category, add items to a shopping cart, place orders, and make payments using multiple methods (credit card, UPI, wallet). The system must handle product inventory, multiple shipping addresses, discounts, order tracking, and allow customers to rate and review products. Use appropriate object-oriented concepts such as inheritance (for user and payment types), aggregation (cart with items), and composition (order with order items), and include multiplicities and relationships among classes.

OVERVIEW:

This class diagram represents the structure of an online shopping system using object-oriented principles such as **inheritance**, **aggregation**, and **association**. It defines the major entities (classes), their attributes, and the relationships among them. The system supports **user registration, product management, cart and order processing, discounts, shipping, and multiple payment methods**.

DIAGRAM:





1. User Hierarchy

User *(Base Class)*

- Common for both Admin and Customer.
- Attributes:
  - userId, name, email, passwordHash
- Methods:
  - register(), login(), logout(), updateProfile()

Admin *(Inherits from User)*

- Manages the platform.
- Methods:
  - addProduct(), updateProduct(), manageDiscounts(), viewOrders()

Customer *(Inherits from User)*

- System end-users who can:
  - Maintain multiple shipping addresses
  - Use a Cart

    - Place and track Orders
    - Rate and review products
- Attributes:
    - addresses: List of ShippingAddress
    - cart: An instance of Cart
    - orders: List of Order
- Method:
    - rateProduct()

2. Shopping Cart

Cart

- Belongs to a single customer.
- Aggregates CartItem objects.
- Methods:
    - addItem(), removeItem(), getTotal(), clear()

CartItem

- Represents a product added to the cart.
- Attributes:
    - cartItemId, quantity, selectedOptions

3. Products and Categories

Product

- Attributes:
    - productId, name, description, price, stock, images
- Methods:
    - addRating(), getAverageRating()
- Linked to:
    - Category
    - Discount
    - Rating

Category

- Groups similar products.
- Attributes:
    - categoryId, name, description

Discount

- Can apply to a product or subtotal.
- Attributes:
    - discountId, code, amount, validFrom, validTo
- Methods:
    - isApplicable(), applyDiscount()

4. Shipping

ShippingAddress

- Belongs to a customer.
- Attributes:
  - addressId, street, city, state, zip, country, phone

5. Orders and Order Items

Order

- Created when a customer places an order.
- Composed of multiple OrderItems.
- Attributes:
  - orderId, orderDate, status (enum), totalAmount
- Method:
  - trackOrder()

OrderItem

- Details of each product in an order.
- Attributes:
  - orderItemId, quantity, priceAtPurchase, selectedOptions

OrderStatus ***(Enumeration)***

- Possible values:
  - PENDING, CONFIRMED, SHIPPED, DELIVERED, CANCELLED, RETURNED

6. Payments

Payment ***(Abstract Class)***

- Attributes:
  - paymentId, amount, paymentDate
- Methods:
  - pay(), refund()
- Extended by:
  - CreditCard, UPI, Wallet

CreditCard

- Attributes:
  - cardNumber, cardHolder, expiry, cvv

UPI

- Attribute:
  - upiId

Wallet

- Attributes:
  - walletId, balance

7. Product Reviews

Rating

- Created by customers for a product.
- Attributes:
  - ratingId, stars, comment, createdAt

CONCLUSION:

This class diagram ensures that the online shopping platform is:

- **Modular and maintainable** through inheritance and composition.
- **User-focused**, with distinct roles for Admin and Customer.
- **Feature-rich**, supporting essential e-commerce functionalities such as cart, payment, reviews, and discounts.

It provides a solid foundation for implementing the backend of a scalable and extensible e-commerce system.