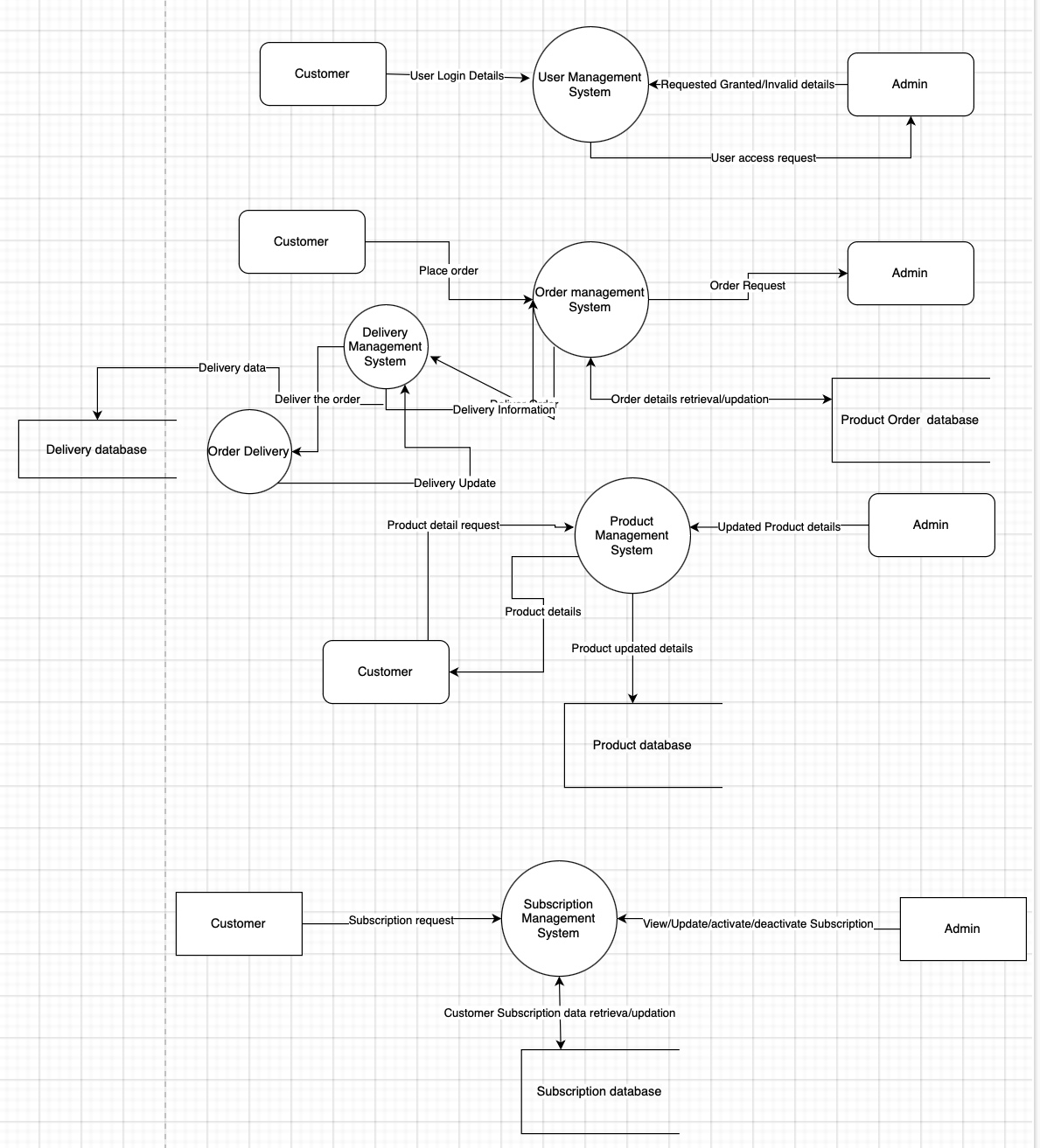
**Code Fury**

**BigCart Platform Project**

**UML Diagram Report**

**Data Flow Diagram**



The diagram represents the flow of data and actions among users, management systems, and databases, focusing on user management, order management, product management, and subscription management.

Diagram Components

1. Customer

- Description: Represents the end-users who interact with the system to request services or information.

- Interactions:

- User Login Details: Customers submit their login details to the User Management System.

- Place Order: Customers place orders through the Order Management System.

- Product Detail Request: Customers request details about products.

- Subscription Request: Customers request subscriptions via the Subscription Management System.

2. Admin

- Description: Represents the system administrators who manage and oversee various operations within the system.

- Interactions:

- Requested Granted/Invalid Details: Admins interact with the User Management System to approve or reject user access requests.

- Order Request: Admins handle order requests within the Order Management System.

- Updated Product Details: Admins update product information in the Product Management System.

- View/Update/Activate/Deactivate Subscription: Admins manage subscriptions, including activation and deactivation, within the Subscription Management System.

3. User Management System

- Description: Manages user information, including login details and access requests.

- Interactions:

- Receives User Login Details from Customers.

- Processes User Access Requests submitted by Customers, granting or denying access as needed.

4. Order Management System

- Description: Handles customer orders, processing, and management.

- Interactions:

- Receives Place Order requests from Customers.

- Manages Order Requests from Admins.

- Facilitates Order Details Retrieval/Updation through the Product Order Database.

5. Product Management System

- Description: Manages product information, including product details and updates.

- Interactions:

- Processes Product Detail Requests from Customers.

- Admins provide Updated Product Details for system updates.

- Interacts with the Product Database to retrieve and store product information.

6. Subscription Management System

- Description: Handles customer subscriptions, including activation, updates, and retrieval.

- Interactions:

- Receives Subscription Requests from Customers.

- Admins manage subscriptions through View/Update/Activate/Deactivate Subscription actions.

- Interacts with the Subscription Database for data storage and retrieval.

7. Databases

- Product Database

- Description: Stores product-related data, which is retrieved and updated by the Product Management System.

- Subscription Database

- Description: Stores subscription-related data, used by the Subscription Management System.

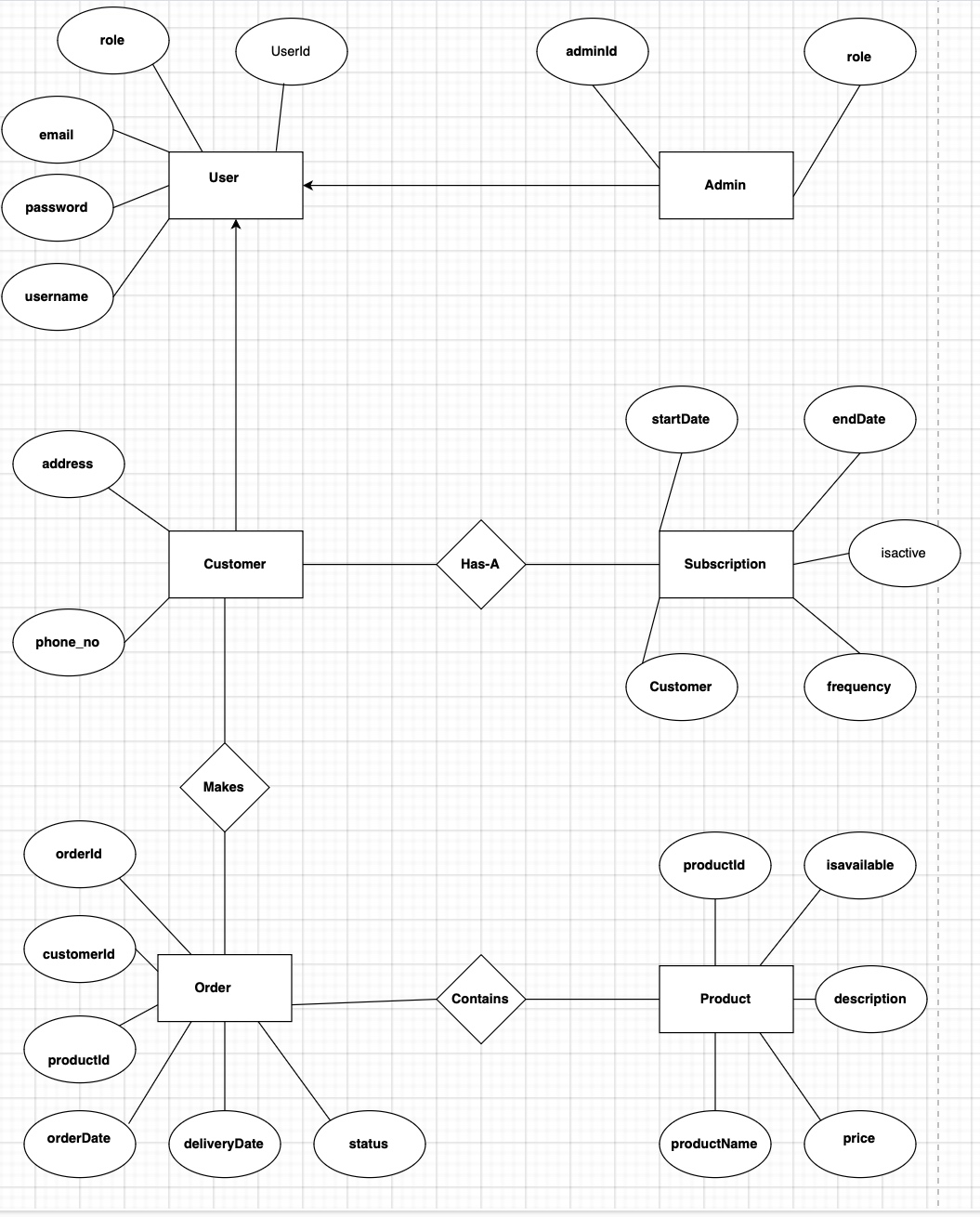
- Product Order Database

- Description: Stores data related to customer orders, interacting with the Order Management System.

Conclusion

The diagram illustrates a well-structured system where each component plays a specific role, interacting with others to manage users, orders, products, and subscriptions. This architecture supports a smooth flow of information and ensures that different system functionalities are effectively integrated.

**ER Diagram:**



Entity-Relationship Diagram

1. Introduction

Entity-Relationship (ER) diagram, visually represents the data model and relationships between various entities. ER diagrams are essential in database design, as they help illustrate the logical structure and connections between entities like Users, Products, Subscriptions, and Orders.

2. Entities and Attributes

The diagram contains the following key entities, each with its associated attributes:

- User

- Attributes: user\_id, username, password, email, address, phone, role

- Product

- Attributes: product\_id, name, description, price, is\_active

- Subscription

- Attributes: subscription\_id, subscription\_type, start\_date, end\_date, status, customer\_id, product\_id

- Order

- Attributes: order\_id, order\_date, delivery\_date, status, customer\_id, product\_id

3. Relationships

The diagram depicts the following relationships:

- User to Subscription: A User can have multiple Subscriptions (1-to-Many relationship).

- User to Order: A User can place multiple Orders (1-to-Many relationship).

- Product to Subscription: A Product can be part of multiple Subscriptions (1-to-Many relationship).

- Product to Order: A Product can be ordered in multiple Orders (1-to-Many relationship).

4. Cardinality

The diagram effectively highlights the cardinality between entities, specifying that:

- Each User can have multiple Subscriptions and Orders.

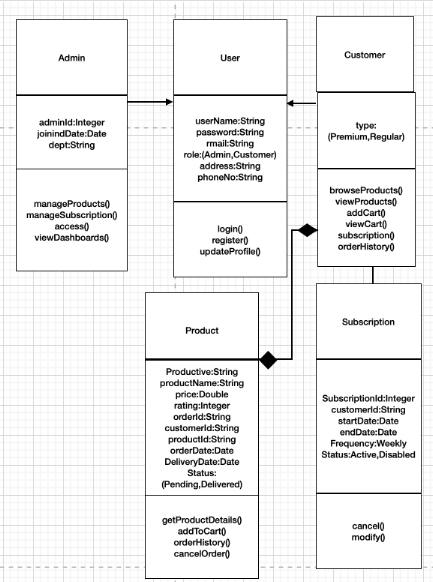
- Each Product can be associated with multiple Subscriptions and Orders.

- Subscriptions and Orders are linked to both Users and Products.

5. Conclusion

This ER diagram illustrates the core structure of the database system, focusing on the relationships between Users, Products, Subscriptions, and Orders. It provides a clear understanding of how data entities interact, which is critical for ensuring database integrity and functionality.

**Class Diagram**



1. Introduction

This report summarizes the class diagram, which visually represents the structure of a software system, focusing on its classes, attributes, methods, and relationships. Class diagrams are a key part of object-oriented design, offering insights into the system's architecture and how its components interact.

2. Classes and Attributes

The diagram identifies key classes in the system and their attributes. Common classes include:

User

Attributes: userId, username, password, email, address, phone, role

Methods: registerUser(), loginUser(), viewProfile(), updateProfile()

Admin (inherits from User)

Methods: defineSubscriptionTypes(), addProduct(), updateProduct(), deleteProduct(), deactivateSubscription(), activateSubscription()

Customer (inherits from User)

Attributes: subscriptionPlan

Methods: browseProducts(), subscribeToProduct(), cancelSubscription(), placeOrder(), orderFood()

Product

Attributes: productId, name, description, price, isActive

Methods: getProductDetails(), listAvailableProducts()

Subscription

Attributes: subscriptionId, subscriptionType, startDate, endDate, status, customerId, productId

Methods: activateSubscription(), deactivateSubscription(), replenishSubscription()

Order

Attributes: orderId, orderDate, deliveryDate, status, customerId, productId

Methods: placeOrder(), viewOrderHistory(), viewOrderDashboard()

3. Relationships

The diagram likely shows relationships between the classes, including:

Inheritance:

The Admin and Customer classes inherit from the User class, meaning they share common attributes and methods but also have unique functionality.

Associations:

The User class is associated with the Subscription and Order classes, representing that a user can have multiple subscriptions and place multiple orders.

The Product class is associated with both Subscription and Order classes, indicating that products are linked to subscriptions and orders.

4. Cardinality

The class diagram specifies cardinality, such as:

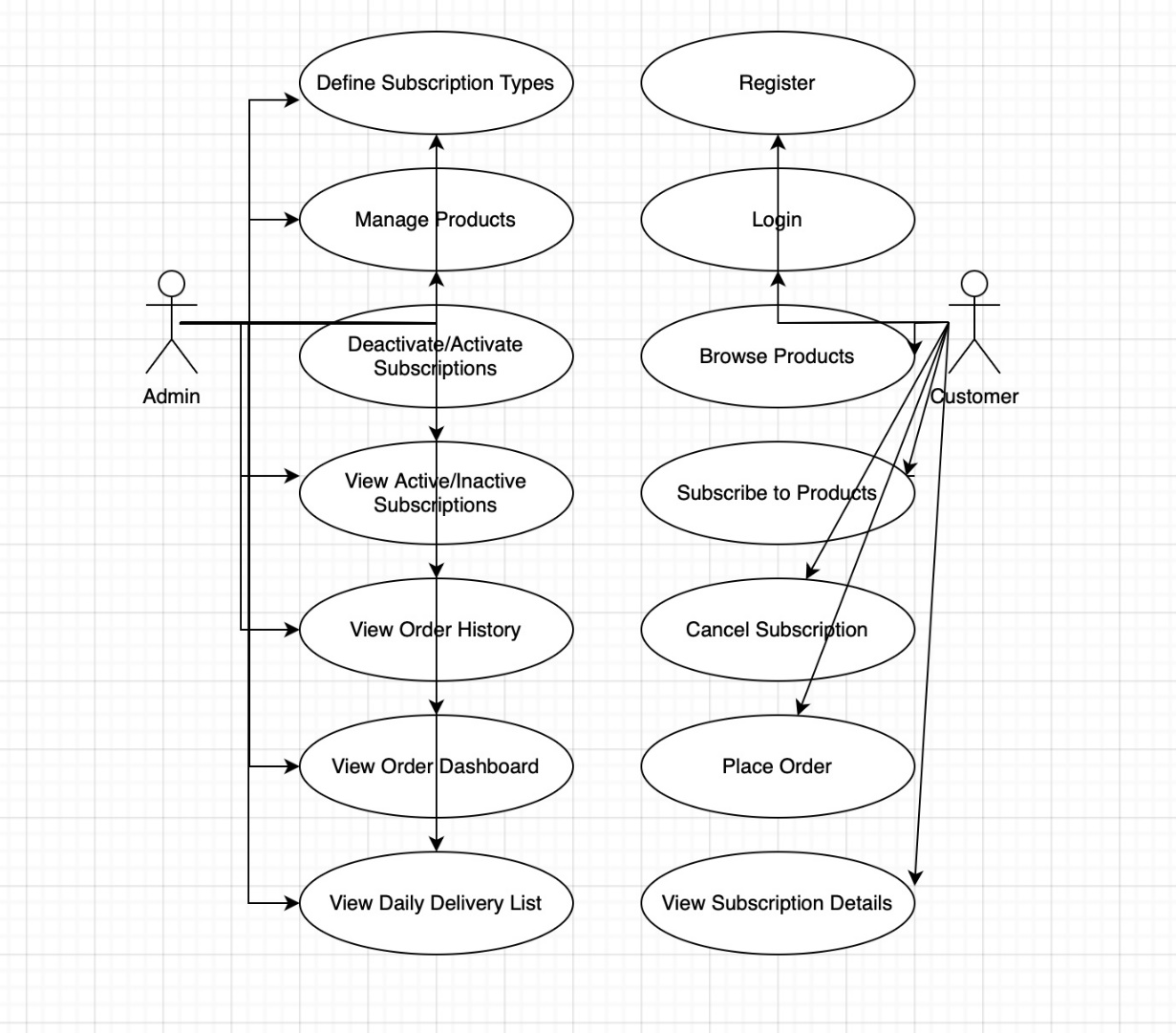
1-to-Many relationships between Users and Subscriptions, as well as Users and Orders.

1-to-Many relationships between Products and Subscriptions, and Products and Orders.

5. Conclusion

The class diagram effectively illustrates the structure and interactions of the software system's core components. It provides a clear view of the system’s architecture, showing how users interact with products, subscriptions, and orders, as well as the specific roles of Admin and Customer classes.

**Use Case Diagram**



Define Subscription Types: This appears to be a process or component related to setting up different types of subscriptions.

Manage Products: This element likely refers to a process for handling or overseeing products within the system.

Deactivate/Activate Subscriptions: This process involves managing the activation or deactivation of subscriptions.

Report on Diagram: Subscription Management Process

Introduction: This report provides an overview of the processes involved in managing subscriptions as depicted in the diagram. The diagram outlines key components and their relationships within the subscription management system, focusing on defining, managing, and modifying subscription types.

1. Define Subscription Types: The process begins with the "Define Subscription Types" component. This step involves establishing the various categories or types of subscriptions available within the system. It likely includes setting parameters such as duration, pricing, and features associated with each subscription type.

2. Manage Products: Following the definition of subscription types, the "Manage Products" process is highlighted. This step involves the administration of products linked to the subscription service. It may include adding new products, updating existing ones, or managing product availability.

3. Deactivate/Activate Subscriptions: The diagram also includes a component labeled "Deactivate/Activate Subscriptions". This process is critical for maintaining active or inactive statuses of subscriptions. It ensures that customers can start or stop their subscriptions as per their requirements.

Conclusion: The diagram provides a clear and structured representation of the core processes involved in managing subscriptions. By defining subscription types, managing related products, and controlling the activation status, the system ensures a comprehensive approach to subscription management.