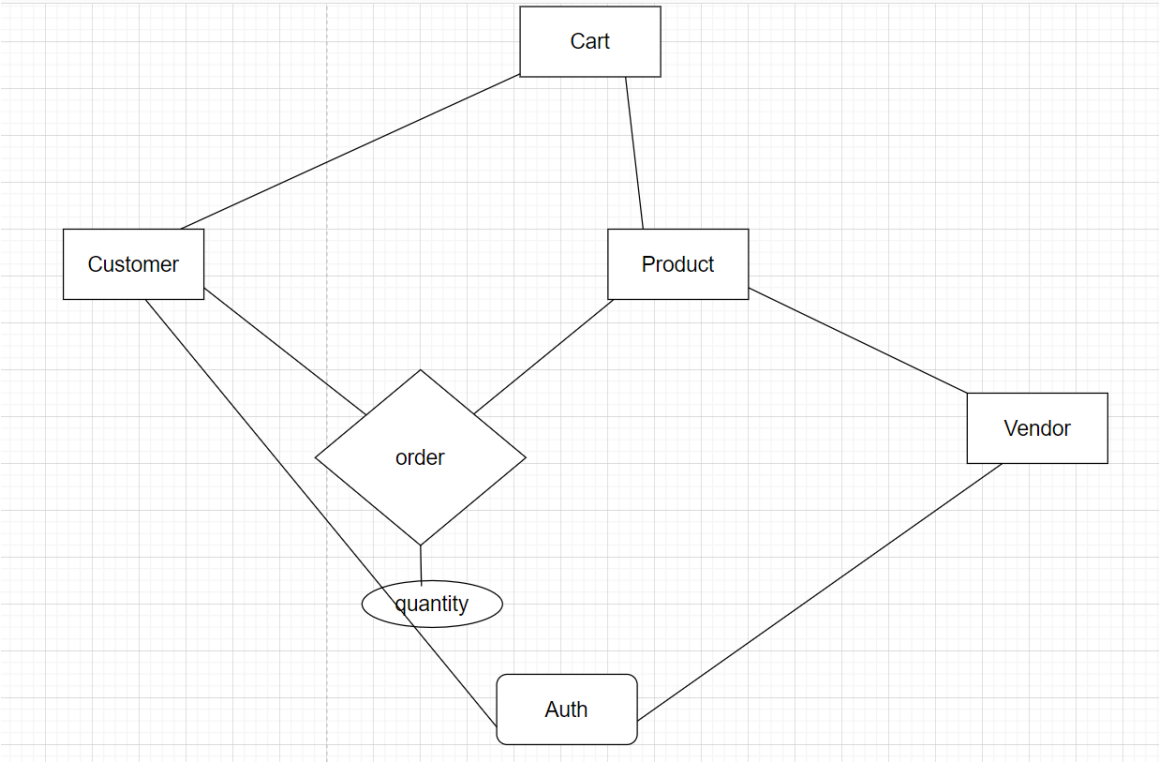
Ecom



Instructions

- Project submissions should be done through the partcipants' Github repository, and the link should be shared with trainers and Hexavarsity.
- Each section builds upon the previous one, and by the end, you will have a comprehensive **Ecommerce** implemented with a strong focus on **SQL**, **control flow statements, loops, arrays, collections, exception handling**, **database interaction** and **Unit Testing**.
- Follow **object-oriented principles** throughout the project. Use classes and objects to model real-world entities, **encapsulate data and behavior**, and **ensure code reusability**.
- Throw **user defined exceptions** from corresponding methods and handled**.**
- The following **Directory structure** is to be followed in the application.
    - **entity/model**
        - Create entity classes in this package. All entity class should not have any business logic.
    - **dao**
        - Create Service Provider interface to showcase functionalities.
        - Create the implementation class for the above interface with db interaction.
    - **exception**
        - Create user defined exceptions in this package and handle exceptions whenever needed.

- **util**
    - Create a **DBPropertyUtil** class with a static function which takes property file name as parameter and returns connection string.
    - Create a **DBConnUtil** class which holds **static method** which takes connection string as parameter file and returns **connection object(Use method defined in DBPropertyUtil class to get the connection String)**.
- **main**
    - Create a class MainModule and demonstrate the functionalities in a menu driven application.

Key Functionalities:

1. **Customer Management**
    - Add new customers, Update, and retrieve customer information and order details,
2. **Product Management:**
    - Users can view a list of available products, add, and delete products.
3. **Cart Management:**
    - Users can add and remove products to their shopping cart.
4. **Order Management:**
    - Users can place orders, which include product details, quantities, and shipping information.
    - The order total is calculated based on the cart contents.

Create following tables in SQL Schema with appropriate class and write the unit test case for the Ecommerce application.

Schema Design:

1. **customers** table:
    - customer_id (Primary Key)
    - name
    - email
    - password
2. **products** table:
    - product_id (Primary Key)
    - name
    - price
    - description
    - stockQuantity
3. **cart** table:
    - cart_id (Primary Key)
    - customer_id (Foreign Key)
    - product_id (Foreign Key)
    - quantity

4. **orders** table:
    - order_id (Primary Key)
    - customer_id (Foreign Key)
    - order_date
    - total_price
    - shipping_address
5. **order_items** table (to store order details):
    - order_item_id (Primary Key)
    - order_id (Foreign Key)
    - product_id (Foreign Key)
    - quantity