**Design Online shopping system**

**Functional requirements:**

1. When customer registers, a cart needs to be assigned to him
2. Every customer has cart entity. Customer can insert, update, delete items in cart
3. After completing all buying, he can generate the order and make payment.

**Entities:**

1. **Product**
2. **Item**
3. **Cart**
4. **Account**
5. **Customer**
6. **Order**
7. **Order status-Enum**
8. **Shipment**

Public class Product

{

private productID,

private productName,

private productDescription,

private double price

private productType type,

private int availableCount,

private Account seller;

}

Public class ProductType

{

private string name;

private string description;

}

Public class Item

{

private string productID;

private int quantity;

private double price;

public boolean updateQuantity(quantity);

}

Public class Cart

{

List<Item>items;

Public boolean AddItems(Item item);

Public boolean removeItem(Item item);

Public List<Item> getItems();

Public boolean updateItemQuantity(Item item, int quantity);

Public checkOut();

}

Public class Account

{

Private string userName;

Private string password;

Private string shippingAddress;

Private string emailId;

Private string mobile;

}

Public class Customer

{

Private Account account;

Private Cart shoppingCart;

Private Order order;

Public boolean addToCart(Item item)

Public boolean removefromcart(Item item);

Public OrderStatus placeOrder(Order order);

}

Public class Order

{

Private long orderId;

Private OrderStatus orderStatus;

Private Date orderDate;

Public String getOrderDetails();

Public boolean sendForShipment();

Public boolean makePayment();

}

Public enum OrderStatus

{

UNSHIPPED; PENDING; SHIPPED; COMPLETED; CANCELLED;

}