

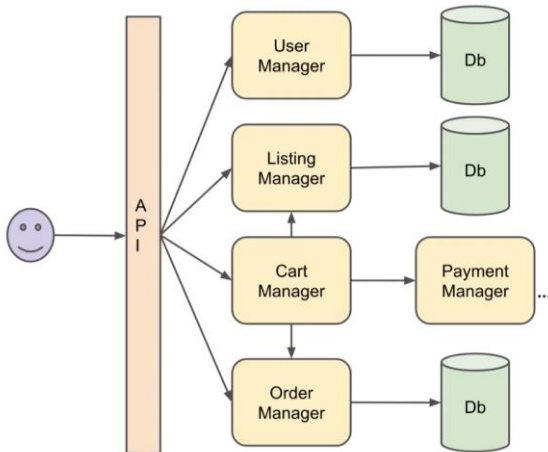
## Problem

Design eCommerce website like Amazon

- Uses Cases
- System design
- APIs
- Database model
- Concurrency

## Use cases

- Create User Profile
- List Products
- Add Products to Cart
- Purchase all Products from cart
- Track Orders



## APIs

```
UserManager {
    boolean createUser(User user);
    void updateUser(User user);
}

ListingManager {
    List<Product> getProducts();
    boolean reduceQuantity(long productId, int quantityToReduce);
}

CartManager {
    boolean addProductToUserCart(long userId, long productId);
    long checkout(long userId); // returns order id
}

OrderManager {
    void createOrder(long userId);
    List<Order> track(long userId);
    Order trackOrder(long userId, long orderId);
}

PaymentManager {
    long makePayment(long userId, long orderId); // returns payment tracking id
}
```

## Database Model

```
User {
    long id;
    Name name;
    String emailId;
    Phone phone;
    //... other details
}

Product {
    long id;
    String name;
    double price;
    int quantity;
}

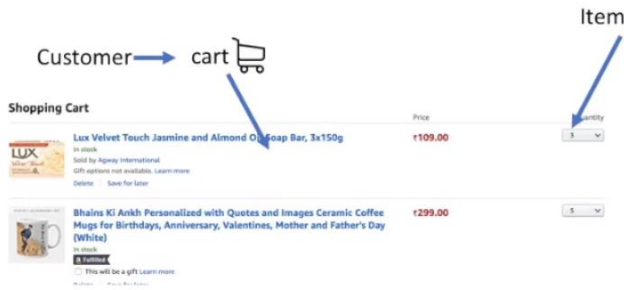
Order {
    long id;
    long userId;
    List<Product> products;
    Date placedOn;
    List<TrackingDetail> trackingDetails;
}
```

## Concurrency

- What if two users trying to order same item?
  - Take lock in ListingManager while updating order details.

## Conclusion

- Very broad problem.
- Try to list use cases.
- Create simple design first....
- More detailed discussion can happen on sub systems like Order, Listing Manager etc.



```
public class ProductCategory {
    private String name;
    private String description;
}

public class Product {
    private String productID;
    private String name;
    private String description;
    private double price;
    private ProductCategory category;
    private int availableItemCount;
    private Account seller;
}
```

cart

```
public class Cart {
    private List<Items> items;
    public boolean addItem(Item item);
    public boolean removeItem(Item item);
    public boolean updateItemQuantity(Item item, int quantity);
    public List<Item> getItems();
    public boolean checkout();
}
```

Order → Shipping

Item

```
public class Item {
    private String productID;
    private int quantity;
    private double price;
    updateQuantity(int quantity);
}
```



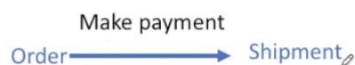
```
public class Account {
    private String userName;
    private String password;
    private AccountStatus status;
    private String name;
    private Address shippingAddress;
    private String email;
    private String phone;

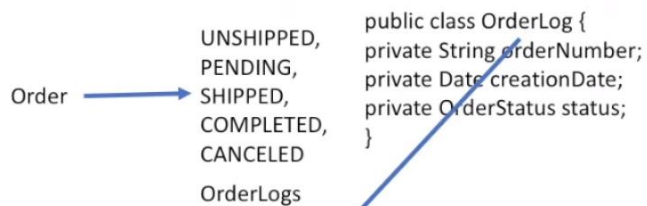
    private List<CreditCard> creditCards;
```

```
public class Customer {
    private Account account;
    private ShoppingCart cart;
    private Order order;

    public bool addItemToCart(Item item);
    public bool removeItemFromCart(Item item);
    public OrderStatus placeOrder(Order order);
}
```

```
public boolean addProduct(Product product);
addProductReview(ProductReview review);
}
```





```
public class Order {  
    private String orderNumber;  
    private OrderStatus status;  
    private Date orderDate;  
    private List<OrderLog> orderLog;  
  
    public boolean sendForShipment();  
    public boolean makePayment(Payment payment);  
    public boolean addOrderLog(OrderLog orderLog);  
}
```

```
public class Shipment {  
    private String shipmentNumber;  
    private Date shipmentDate;  
    private Date estimatedArrival;  
    private String shipmentMethod;  
    Order orderDetails;  
}
```

Notification Service    Payment Service

Account  
Customer  
ProductCategory  
Product  
Cart  
Item  
Order  
OrderLog  
Shipment