Expt No: 4,	CLASS DIAGRAM – STOCK INVENTORY MANAGEMENT

Aim:

To design a class diagram for the Stock Inventory Management System.

Definition:

A Class Diagram is a type of static structure diagram in Unified Modeling Language (UML) that describes the system's structure by showing its classes, attributes, methods (operations), and the relationships between the classes. It is a blueprint for the system's design, focusing on how data is structured and how different parts of the system interact with each other.

Purpose of a Class Diagram:

- **Visualize the System Structure:** It helps to represent the system's architecture, showing how classes and objects interact.
- **Define Class Responsibilities:** It outlines what each class does (methods) and what data it stores (attributes).
- **Facilitate Communication:** It serves as a shared reference for developers, stakeholders, and designers, ensuring everyone understands the system's structure.
- **Guide Code Implementation:** It provides a detailed design blueprint that developers can follow when implementing the code.
- **Document the System:** It serves as formal documentation for future reference, especially for system maintenance or expansion.

Components of Class Diagram:

1. Classes:

- **Definition:** A class is a blueprint for objects. It defines the attributes (data) and methods (functions) of an object.
- **Example:** In the Stock Inventory Management System, classes include Admin, Cashier, Inventory Manager, Stock, and Customer.

2. Attributes:

- **Definition:** Attributes are properties of a class. They represent the state or data of an object.
- **Example**: In the Stock class, attributes might include the name of the stock item (e.g., "Laptop"), quantity, and arrival_date.

3. Methods (Operations):

- **Definition:** Methods define the behavior or actions a class can perform.
- **Example:** In the Inventory Manager class, methods might include stock_in() and stock_out(). In the Cashier class, methods might include billing() and get_payment().

Relationships:

4. Associations:

- **Definition:** A relationship between two or more classes showing how they interact (e.g., one class uses or depends on another).
- **Example:** A Cashier interacts with a Customer when processing billing and payment.

5. Inheritance (Generalization):

- **Definition:** Represents a parent-child relationship where a subclass inherits attributes and methods from a superclass.
- **Example:** RegularCustomer and FirstTimeCustomer inherit from the Customer class, meaning they share common attributes and methods like make_payment().

6. Aggregation/Composition:

- **Definition:** Represents a "whole-part" relationship between classes. Composition is a strong form of aggregation, where parts cannot exist independently of the whole.
- **Example:** A Customer may place an order, and if the Customer is deleted, the related order (part) may be deleted too.

7. Multiplicity:

- **Definition**: Specifies how many instances of one class can be associated with instances of another class (e.g., one-to-many, many-to-many).
- **Example:** A Customer can place many orders (one-to-many), while many Cashiers can process many Customer transactions (many-to-many).

Visibility Modifiers:

- **Public (+):** Can be accessed from anywhere.
- **Private (-):** Can only be accessed within the class.
- **Protected (#):** Can be accessed within the class and its subclasses.

Stock Inventory Management:

1) Classes:

- **Admin:** This class is responsible for administrative operations such as creating new users and viewing the system's dashboard.
- **Inventory Manager:** A specialized user responsible for managing the stock, such as checking stock in and out.
- **Executive Manager:** This class manages high-level operations and has access to an overview dashboard to track key metrics.
- **Cashier:** This class handles billing and payment processing as well as stock-out procedures.
- **Customer:** This class represents customers, both regular and first-time, with functionalities like billing, making payments, and sending receipts. It is further specialized into:
    - **Regular Customer:** This subclass manages regular customers who can cancel their membership.
    - **First Time Customer:** This subclass represents new customers who can create or cancel membership.
- **Stock:** This class holds details about the stock, such as the stock name, quantity, and arrival date.

2) Attributes:

In the **Stock Inventory Management System**, attributes represent data specific to each class. For example:

- **Admin Class:**
    - name: string
    - user_name: string
    - password: string
    - registration_date: date
- **Inventory Manager Class:**
    - name: string
    - user_name: string
    - password: string
    - registration_date: date
    - salary: number
- **Cashier Class:**
    - name: string

    - user_name: string
    - password: string
    - salary: number
    - joining_date: date
- **Customer Class:**
    - name: string
    - phone_number: number
    - is_regular: boolean
    - membership_id: string

- **Stock Class:**
    - name: string
    - quantity: number
    - arrival_date: date

3) Methods:

Methods define the actions and behaviors of each class:

- **Admin Class:**
    - create_new_user(): boolean – Create new user accounts in the system.
    - view_dashboard(): void – View system statistics and overall dashboard.
- **Inventory Manager Class:**
    - login(): boolean – Login to the system.
    - stock_in(): Stock – Register stock that has been added.
    - stock_out(): Stock – Track stock that has been removed.
- **Cashier Class:**
    - login(): boolean – Login to the system.
    - stock_out(): boolean – Process a stock-out (item sale).
    - billing(): number – Generate bills for customers.
    - get_payment(): boolean – Process payments from customers.
- **Customer Class:**
    - get_bill(): void – Retrieve the bill for a purchase.
    - make_payment(): void – Process customer payment.
    - send_pdf(): void – Send the bill in PDF format.

- **Regular Customer Class:**
    - cancel_membership(): void – Cancel the membership of a regular customer.
- **First Time Customer Class:**
    - get_mobile_number(): number – Retrieve the mobile number of a first-time customer.
    - create_membership(): boolean – Create a membership for a new customer.
    - cancel_membership(): void – Cancel membership.

4) Relationships:

- **Association:**
    - The **Customer** class interacts with the **Cashier** class, as customers receive billing and payments are processed by the cashier.
    - The **Inventory Manager** is responsible for managing the stock, which creates an association between the **Inventory Manager** and **Stock** classes.
- **Inheritance:**
    - Both **Regular Customer** and **First Time Customer** inherit from the **Customer** class, meaning they share attributes and behaviors like name, phone number, and the ability to retrieve a bill.
    - The **Executive Manager** and **Admin** are both higher-level roles that inherit common functionality like login() from a hypothetical **User** class.
- **Composition:**
    - **Stock** is linked to the **Inventory Manager** and **Cashier** classes. If a stock item is deleted, its stock-in/stock-out records should also be deleted, indicating a composition relationship.

CLASS DIAGRAM

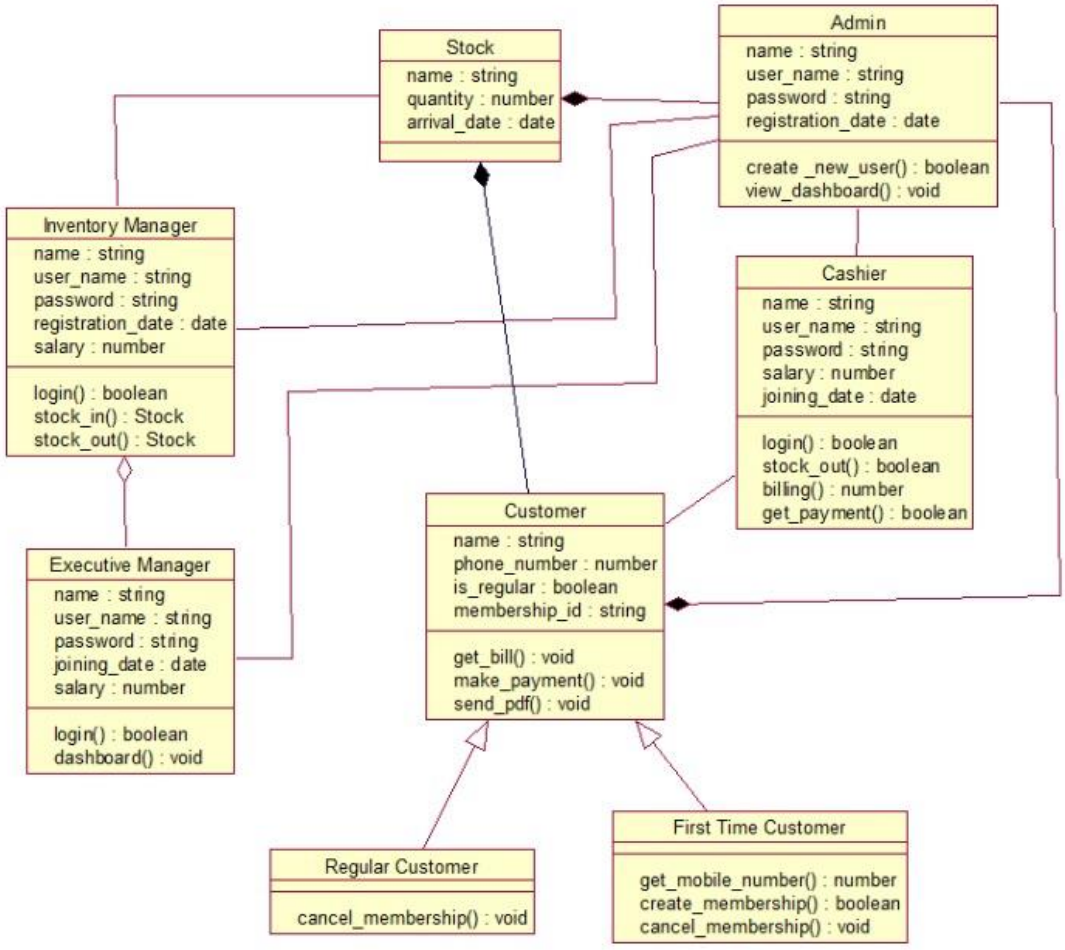

Figure 1Class Diagram for Stock Inventory Management System

RESULT:

Thus, the development of class diagram for Stock Inventory Management has been done successfully