

Name – Sharyu Charde, Hitesha Shahane
UID – 23011046, 23011037

Practical No. 5 – Class Diagram for Pharmacy Management System

Aim:

To design and understand the Class Diagram for the Pharmacy Management System, representing the structural relationships and entities involved in the system.

Introduction:

A Class Diagram is a fundamental building block in object-oriented modeling. It represents the static structure of a system by showing classes, their attributes, operations, and relationships among objects. In the context of a Pharmacy Management System, the Class Diagram helps to identify and organize the core entities and their interconnections that form the basis of the application's functionality.

Objectives:

1. To represent the structure and relationships among different classes in the Pharmacy Management System.
2. To illustrate the use of associations, inheritance, and aggregation.
3. To model the real-world entities in a conceptual framework that can be converted into actual software components.
4. To ensure that all essential data and behavior of the system are clearly defined.

Theory:

A Class Diagram defines the blueprint of an object-oriented system. It shows the different classes, their attributes, methods, and the relationships between them. These diagrams are static, meaning they describe the system structure at one particular time. Each class represents a specific part of the system with properties and behaviors. In a Pharmacy Management System, the classes generally represent entities such as medicines, sales transactions, users, and suppliers. The relationships define how these classes interact and depend on one another. The diagram uses notations like rectangles for classes, lines for associations, arrows for inheritance, and diamonds for aggregation.

Case Study:

In a Pharmacy Management System, the Class Diagram organizes the system's components into logical groups. It includes classes that define the core operations of a pharmacy such as managing stock, handling billing, and maintaining user data. This diagram helps in visualizing how each part of the system interacts. For example, an Employee class may manage the functions of adding, updating, or deleting data, while the Customer class could handle purchase requests. Such representation enables developers to understand dependencies and streamline the development process. Class Diagrams thus serve as a bridge between the real-world understanding of the pharmacy's workflow and the implementation of the system in code. It provides clarity to both developers and analysts while designing the architecture of the software.

Advantages:

1. Helps in visualizing the structure of the system before actual implementation.
2. Simplifies the communication among developers, designers, and stakeholders.
3. Enhances reusability of components through clear depiction of relationships.
4. Provides a strong foundation for converting design models into code efficiently.
5. Ensures that system functionalities are well-documented and easily maintainable.

Conclusion:

The Class Diagram for the Pharmacy Management System serves as a crucial design element in understanding the system's structure. It provides a clear overview of entities, their properties, and interconnections. Through the use of object-oriented principles, this diagram helps ensure modularity, maintainability, and scalability in the software design process. Thus, the Class Diagram acts as a blueprint for the development and implementation of the Pharmacy Management System.

References:

1. Sommerville, Ian. "Software Engineering." Pearson Education.
2. Pressman, Roger S. "Software Engineering: A Practitioner's Approach." McGraw-Hill.
3. UML Documentation – Object Management Group (OMG).