Laboratory Report

Experiment No - 05

Batch -

Date of Experiment: __________ Date of Submission: __________

Title: Identify scenarios & develop UML Use case and Class Diagram for the project

____

____

Evaluation:

1) Attendance [2] ----------------

2) Lab Performance [2] ----------------

3) Oral [1] ----------------

Overall Marks [5] ---------------

Subject In-Charge

Experiment No: - 05

TITLE: Identify scenarios & develop UML Use case and Class Diagram for the project.

PREREQUISITE:

1. Concepts of Object Oriented Programming & Methodology
2. Knowledge of developing applications with front end & back end connectivity.

HARDWARE CONFIGURATION / KIT:

Sr. No	Minimum Hardware Configuration	
1	Processor	800MHz Intel Pentium III or above versions
2	RAM	512 MB
3	HDD	1.5 GB of free disk space

SOFTWARE CONFIGURATION:

Sr. No	Minimum Software Configuration	
1	Operating System	Microsoft Windows Vista/7 or above versions
2	Editor	MS Word, Notepad

Theory: -

UML is a standardized modeling language used for software design. It provides a graphical notation for representing software designs using a variety of diagrams, such as class diagrams, activity diagrams, and sequence diagrams.

1. **Use Case:**

A use case diagram software and system engineering term that describe how a user uses a system to accomplish particular goal. A use case act as a software modeling technique that defines the features to be implemented and the resolution of any errors that may be encountered.

In software engineering, a use case diagram is a type of diagram that is used to visualize the relationships and interactions between different actors (users, systems, or external entities) and the functionalities provided by a system or application.

Use case diagrams can be used to identify and analyze the different scenarios or use cases that the system or application needs to support. They can also be used to identify the different types of actors that interact with the system, and the different functionalities that the system needs to provide to these actors.

The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

General steps to draw Use case Diagram:

1. Identify the actors: Identify the different types of users or external systems that interact with the system or application.
2. Identify the use cases: Identify the various tasks, actions, or functions that the system or application can perform.
3. Determine the relationships: Determine how the actors and use cases interact with each other. Identify the dependencies, associations, and generalizations between actors and use cases.

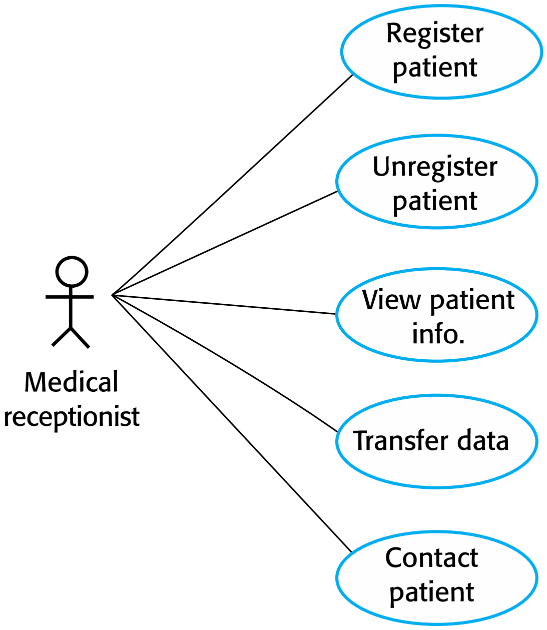

Fig: Use case Diagram for Mentcare System

2. Class Diagram:

A class diagram is a type of diagram in the Unified Modeling Language (UML) that is used to represent the structure of a system or software application. It is a graphical representation of classes, interfaces, associations, and other relationships between objects.

The basic elements of a class diagram are:

1. Class: A class represents a set of objects with similar properties, methods, and relationships. It is represented as a rectangular box with the class name written inside.
2. Interface: An interface is a collection of methods that define a set of operations that a class can implement. It is represented as a circle with the interface name written inside.
3. Attribute: An attribute is a property of a class that describes its state. It is represented as a name-value pair inside the class box.
4. Operation: An operation is a method or function that can be performed on a class. It is represented as a name followed by a parameter list inside the class box.

5. Association: An association represents a relationship between two or more classes. It is represented as a line connecting the classes, with optional arrows indicating the direction of the relationship.
6. Aggregation: Aggregation is a special form of association that represents a "whole-part" relationship between two classes. It is represented as a line with a diamond at the end that points to the whole class.
7. Inheritance: Inheritance is a mechanism that allows a new class to be based on an existing class, inheriting all its attributes and operations. It is represented as a line with an open arrowhead pointing to the base class.
8. Dependency: A dependency represents a relationship between two classes where one class depends on the other. It is represented as a dashed line with an arrowhead pointing to the class that is depended upon.

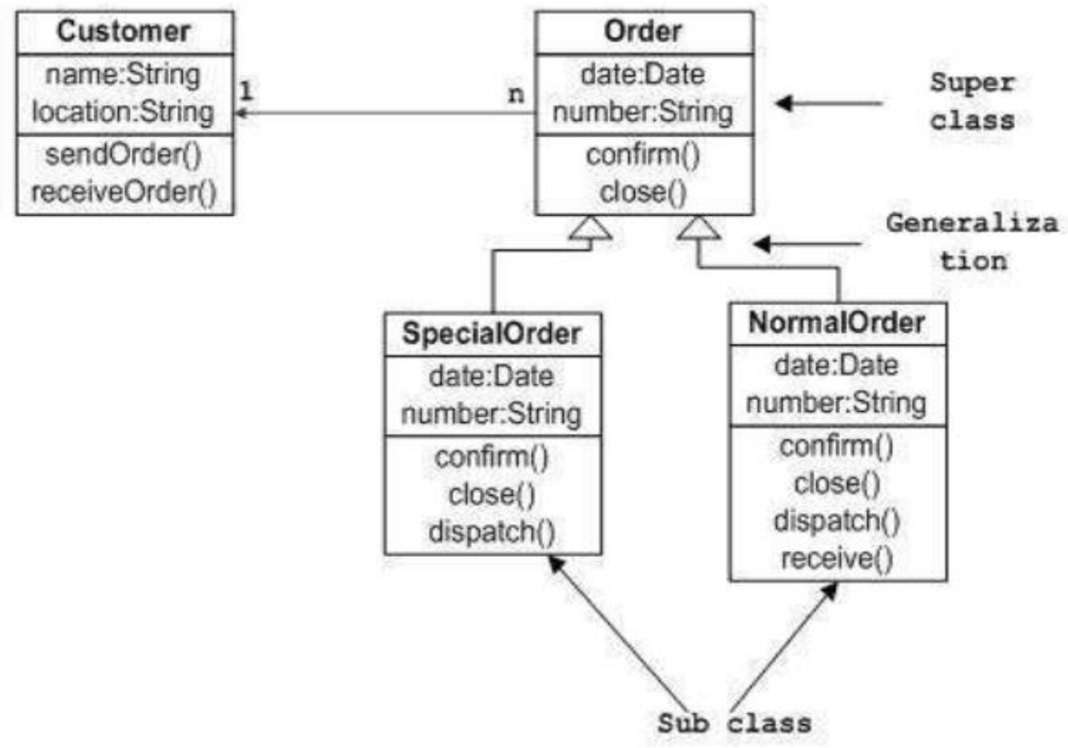

Fig: Sample Class Diagram

Exercise:

1. Identify the different user scenarios and draw Use Case diagram for the project assigned to you.
2. Draw Class diagram for the project assigned to you.