Practical no-04

Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

Modeling UML (Unified Modeling Language) Use Case Diagrams and capturing Use Case Scenarios are critical aspects of system design and software development. Here's a guide to understanding both concepts:

**1. UML Use Case Diagrams:**

A **Use Case Diagram** is a graphical representation of a system’s functionalities (use cases) from the user’s perspective (actors). It helps in understanding the interactions between users (or other systems) and the system itself.

**Key Components of a Use Case Diagram:**

* **Actors:** Represent the roles that interact with the system (e.g., users, other systems, or hardware). Actors are typically depicted as stick figures.
  + **Primary Actor**: The user or system that initiates an interaction to achieve a goal.
  + **Secondary Actor**: Provides a service to the primary actor (e.g., a database).
* **Use Cases:** These represent the functionalities or services the system provides to the actors. Use cases are usually depicted as ellipses or ovals.
* **Associations (Relationships):** Lines connecting actors to use cases.
  + **Association**: A solid line showing interaction between actors and use cases.
  + **Include Relationship**: An arrow pointing to another use case that is always included when the original use case is executed.
  + **Extend Relationship**: A dashed line with an arrow showing optional behavior that extends the base use case.
* **System Boundary:** A rectangle that defines the scope of the system. Use cases are placed inside the system boundary, while actors are placed outside it.

Steps to Create a Use Case Diagram:

1. Identify Actors: Determine the people, systems, or external devices that will interact with the system.
2. Identify Use Cases: Define the main functionalities or services that the system should provide.
3. Determine Relationships: Establish how actors interact with each use case, and specify any include or extend relationships.
4. Draw the Diagram: Use UML notation to draw actors, use cases, and relationships.

**2. Capturing Use Case Scenarios:**

A Use Case Scenario provides detailed information about a specific use case’s interaction. It describes the steps involved when a user interacts with the system to accomplish a particular goal. The scenario can represent a Basic Flow (normal sequence of events) and Alternative Flows (variations or exceptions).

Components of a Use Case Scenario:

* Use Case Name: Describes the goal or functionality.
* Actors: Lists all actors involved in the use case.
* Preconditions: Describes the state the system must be in before the use case can start.
* Postconditions: Describes the expected state of the system after the use case has completed.
* Basic Flow (Main Success Scenario): A sequence of steps that describes the normal, expected behavior.
* Alternative Flows: Describes deviations or exceptions that may occur during the execution of the use case (e.g., error handling).
* Exception Flows: Describes scenarios when the normal flow is interrupted by errors or exceptions.
* Extensions (Optional): Describes any extensions or additional actions that could occur based on certain conditions.

Steps to Write Use Case Scenarios:

* Identify the Use Case
* Define Preconditions
* Define the Basic Flow
* Consider Alternative Flows
* Define Postconditions
* Handle Exceptions