

# EXPERIMENT NO. 6

**Aim :** - Identify scenarios & Develop Use Case Diagram for the project.

**Theory :** -

Use Case Diagrams are a fundamental tool in software engineering for modeling the interactions between users (actors) and a system. They help identify and define various scenarios or user interactions within a project, providing a visual representation of system functionality from an external perspective. In this theory, we will explore the process of identifying scenarios and developing Use Case Diagrams for a project.

## Identifying Scenarios

Before creating a Use Case Diagram, it's essential to identify and understand the scenarios or use cases that describe how users interact with the system. Scenarios represent specific tasks, actions, or processes that users or external systems perform within the software.

### Steps to Identify Scenarios:

**Stakeholder Analysis:** Identify all stakeholders, including end-users, administrators, and external systems, who will interact with the software.

**User Interviews and Surveys:** Conduct interviews or surveys with users and stakeholders to gather information about their requirements, needs, and expected interactions with the system.

**Brainstorming Sessions:** Organize brainstorming sessions with the project team to generate a list of potential use cases. Encourage participants to think from different user perspectives.

**Requirements Documentation:** Review project requirements documents and specifications to identify user stories or tasks that can be converted into use cases.

**Contextual Analysis:** Analyze the system's context within the organization or industry to identify potential external interactions and scenarios.

**Functional Decomposition:** Break down the system's functionality into smaller, manageable tasks or features, each of which can become a use case.

**Boundary Identification:** Clearly define the boundaries of the system and identify what falls within the scope of the project.

### Developing Use Case Diagrams

Once scenarios are identified, Use Case Diagrams are created to represent the relationships between actors and use cases, providing a visual overview of how users interact with the system.

#### Steps to Develop a Use Case Diagram:

**Identify Actors:** Actors are entities external to the system that interact with it. They can be users, other systems, or hardware devices. List and name all actors involved in the scenarios.

**Identify Use Cases:** Each use case represents a specific scenario or functionality within the system. List and name the use cases based on the scenarios identified in the previous step.

**Establish Relationships:** Connect actors to their associated use cases using lines with arrows pointing from actors to use cases. This signifies that an actor interacts with the use case.

**Include System Boundary:** Draw a boundary around the use cases to represent the system's scope. This boundary separates the system from its external actors.

**Extend Relationships (Optional):** Use extends and includes relationships to depict how one use case can extend or include another. This helps represent complex interactions and alternative flows.

**Generalization (Inheritance):** Use generalization relationships to show how one use case inherits common behaviors from another. This is useful for modeling variations of a use case.

**Document Use Case Descriptions:** Add brief descriptions or comments to each use case to provide a clear understanding of its purpose and functionality.

**Keep It Simple:** Use Case Diagrams should be clear and concise. Avoid including excessive details or technical implementation aspects.

**Conclusion : -**

Identifying scenarios and developing Use Case Diagrams are essential steps in software engineering, particularly during the requirements analysis and design phases. These diagrams help stakeholders, including developers and project managers, visualize how users interact with the system and understand the software's expected functionality from an external perspective. Properly constructed Use Case Diagrams serve as valuable tools for communication, requirement documentation, and the foundation for subsequent stages of software development, such as system design and implementation.