Name : Kahar Ankit Shriram
Enroll: 2301030400014
Roll: CE-A2-064
Sub : Software engineering
Pr: 04

Case Study on Use-Case Diagram

What is a Use Case Diagram ?

A **use case diagram** visually shows how users (called actors) interact with a system. It highlights the main functions (use cases) the system offers and helps understand what the system needs to do from the users' perspective.

When to apply Use Case Diagram?

Use case diagrams are useful in several situations. Here's when you should consider using them:

- When you need to gather and clarify user requirements, use case diagrams help visualize how different users interact with the system.
- If you're working with diverse groups, including non-technical stakeholders, these diagrams provide a clear and simple way to convey system functionality.
- During the system design phase, use case diagrams help outline user interactions and plan features, ensuring that the design aligns with user needs.
- When defining what is included in the system versus what is external, use case diagrams help clarify these boundaries.

Use Case Diagram Notations ?

1. Actor

- **Symbol: A simple stick figure**

- **Meaning: Represents a person, group, organization, or external system that interacts with the system. Actors are typically users but can also be other systems or devices that communicate with the system.**

- **Role: Actors initiate or participate in use cases, meaning they perform actions or receive services from the system.**

- **Example: In an online shopping system, actors might be *Customer*, *Admin*, or *Payment Gateway*.**

2. Use Case 

- **Symbol: An oval (ellipse) shape**

- **Meaning: Represents a specific function, task, or service that the system performs in response to an actor's interaction.**

- **Role: Describes what the system does from the user's perspective without specifying how it is done internally.**

- **Example: "Place Order," "Login," "Search Product," "Generate Report."**

3. System Boundary

- **Symbol: A rectangle enclosing all use cases**

- **Meaning: Defines the scope of the system, distinguishing what is inside (part of the system) and what is outside (external to the system).**

- **Role: Helps clarify which use cases belong to the system under consideration and which actors are external entities interacting with it.**

- **Label: The rectangle is often labeled with the system's name, e.g., "Online Bookstore System."**

4. Association

- **Symbol: A solid line connecting an actor to a use case**
- **Meaning: Represents communication or interaction between the actor and the use case. It shows that the actor participates in or initiates that use case.**
- **Direction: Usually, associations are bidirectional (no arrow), meaning both actor and system can communicate. Sometimes arrows show the initiation of interaction.**
- **Example: A line connecting *Customer* to "Place Order" shows the customer participates in placing orders.**

5. Include Relationship

«include»

- **Symbol: A dashed arrow pointing from one use case to another, labeled with «include»**
- **Meaning: Indicates that the base use case always includes the behavior of the included use case. The included use case is mandatory and reusable across multiple use cases.**
- **Role: Helps break complex use cases into smaller, reusable parts to avoid repetition.**
- **Example: The "Checkout" use case might include "Make Payment" because every checkout involves a payment process.**

6. Extend Relationship

«extend»

- **Symbol: A dashed arrow pointing from an extension use case to a base use case, labeled with «extend»**
- **Meaning: Represents optional or conditional behavior that extends the base use case at specific points. The base use case can function independently without the extension.**
- **Role: Used to add optional features or variations without cluttering the main use case.**
- **Example: "Apply Discount" might extend "Checkout," occurring only if the customer has a discount coupon.**

7. Generalization (Inheritance)

- **Symbol: A solid line with a hollow triangular arrowhead pointing from the specialized actor or use case to the generalized one**
- **Meaning: Shows inheritance or hierarchy where one actor or use case is a specialized version of another.**
- **Role: Helps organize actors or use cases that share common behavior but differ in specifics.**
- **Example:**
    - **Actors: *Registered Customer* is a specialized type of *Customer*.**
    - **Use Cases: "Pay by Credit Card" and "Pay by PayPal" may be specialized forms of a general "Make Payment" use case.**

How to draw a Use Case diagram in UML?

Below are the main steps to draw use case diagram in UML:

● Step 1: Identify Actors: Determine who or what interacts with the system. These are your actors. They can be users, other systems, or external entities. ● Step 2: Identify Use Cases: Identify the main functionalities or actions the system must perform. These are your use cases. Each use case should represent a specific piece of functionality.

- **Step 3: Connect Actors and Use Cases: Draw lines (associations) between actors and the use cases they are involved in. This represents the interactions between actors and the system.**

● Step 4: Add System Boundary: Draw a box around the actors and use cases to represent the system boundary. This defines the scope of your system. ● Step 5: Define Relationships: If certain use cases are related or if one use case is an extension of another, you can indicate these relationships with appropriate notations.

- **Step 6: Review and Refine: Step back and review your diagram. Ensure that it accurately represents the interactions and relationships in your system. Refine as needed.**
- **Step 7: Validate: Share your use case diagram with stakeholders and gather feedback. Ensure that it aligns with their understanding of the system's functionality.**

What is the Purpose and Benefits of Use Case Diagrams? The Use Case Diagram offers numerous benefits throughout the system development process. Here are some key advantages of using Use Case Diagrams:

- **Use Case Diagrams offer a clear visual representation of a system’s functions and its interactions with external users. This representation helps stakeholders, including those without technical expertise, in grasping the system’s overall behavior.**
- **They establish a shared language for articulating system requirements, ensuring that all team members have a common understanding.**
- **Use Case Diagram illustrate the different ways users engage with the system, contributing to a thorough comprehension of its functionalities.**
- **In the design phase, Use Case Diagrams help outline how users (actors) will interact with the system. They support the planning of user interfaces and aid**

in structuring system functionalities.