

## **SOFTWARE ENGINEERING LAB**

### **EXERCISE – 3**

### **TOPIC – 2**

## **UML DIAGRAMS – USECASE**

### **Use Case Diagrams**

A Use Case Diagram is a type of diagram that helps us understand how a software system works from the user's point of view. It shows the people (called actors) or other systems that interact with the software, and the specific actions they can perform. This diagram makes it easy to see what the system can do and how it works.

### **Purpose of a Use Case Diagram**

The main reason for using a Use Case Diagram is to show the functions of a system in a simple way. This diagram is very useful when you need to understand the overall picture of what the system does, without diving into too many details. It does three main things:

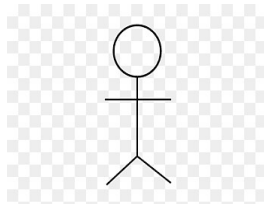
1. **Shows the Actors:** It identifies who is using the system (like a customer or another system).
2. **Describes the Actions:** It explains what actions or tasks the user can do with the system (like withdrawing money from an ATM).
3. **Defines the System's Boundaries:** It clearly shows what is **inside the system** (what the system does) and what is **outside the system** (the users or other systems).

### **Key Parts of a Use Case Diagram**

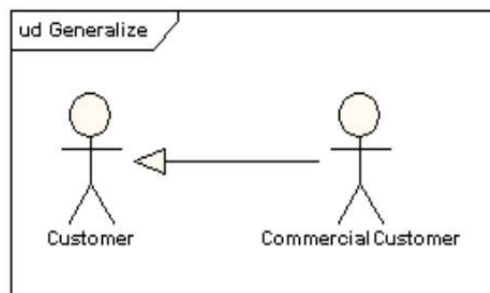
There are several important parts (symbols) in a Use Case Diagram. Each one plays a specific role in explaining how the system works.

**1. Actor:** An actor is anyone or anything that interacts with the system. This could be a person, another software system, or even a piece of hardware.

- **Example:** In an ATM system, the actor could be the "Customer" using the ATM or the "Bank" that processes the transactions.
- **Symbol:** It is represented by a **stick figure**. This makes it easy to recognize as someone or something interacting with the system.



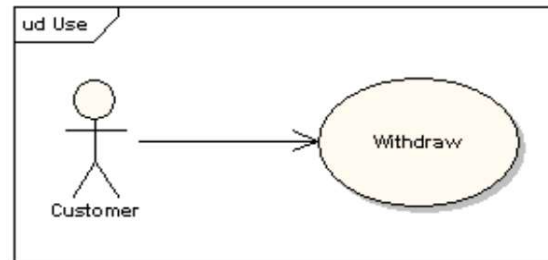
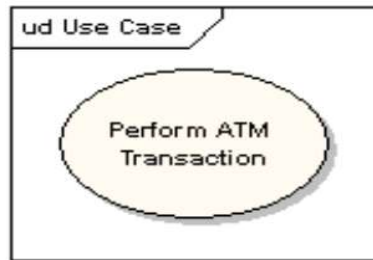
- **Generalization of Actors:** Actors can have a hierarchical relationship. One actor can "generalize" another, meaning one type of actor shares similarities with another but may also have additional roles.



- In the image, we see a Customer Actor and a Commercial Customer actor. A Commercial Customer is a type of customer, but with more specific responsibilities. The arrow between them shows this generalization, meaning all commercial customers are customers, but not all customers are commercial customers.

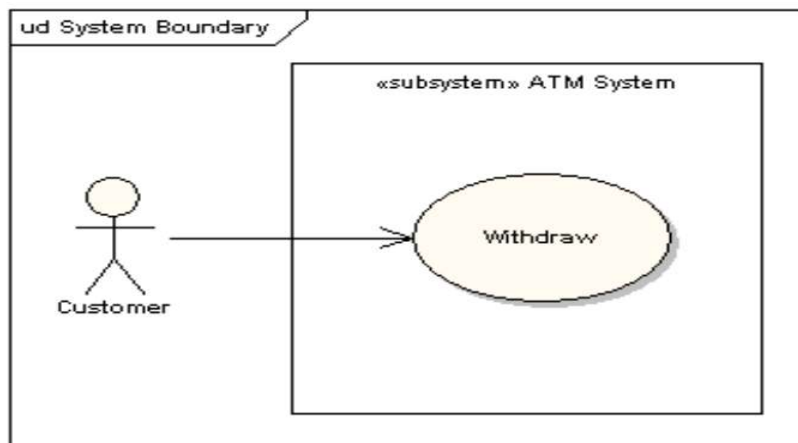
## 2. Use Case

- **Definition:** A use case is an action or task that the actor can do with the system. It is like a function or activity the system provides to the user.
- **Example:** For an ATM system, use cases could include tasks like "Withdraw Cash," "Check Balance," or "Transfer Funds."
- **Symbol:** A use case is shown as an **oval** with the name of the action written inside.

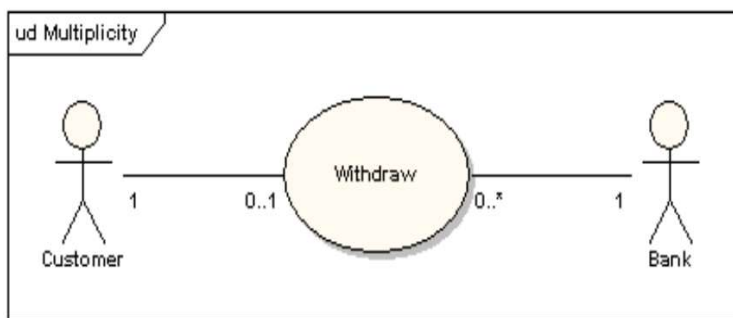


### 3. System Boundary

- **Definition:** The system boundary is a box or rectangle that surrounds the use cases. It helps to show what is part of the system and what is not. Everything inside the boundary is controlled by the system, and everything outside interacts with it.
- **Symbol:** The boundary is represented by a **rectangle** that encloses all the use cases.

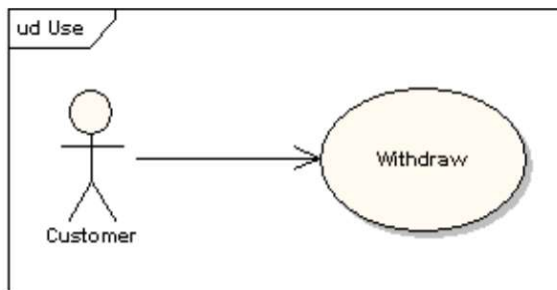


**4. Associations (Lines):** An Association is a straight line that shows a connection between an actor (a person or system) and a use case (a task or action). It means the actor is involved in the task, but it doesn't say who starts the action.



*Example:* A line between a Customer and a Login task means the customer is part of the login process, but it doesn't tell us who begins the login.

**Directed Association:** A Directed Association is like an association, but it has an arrow. The arrow shows who starts or controls the action. It points to the task, meaning the actor is the one who starts the action.



*Example:* If there's an arrow from a Customer to a Withdraw task, it means the customer starts the process of withdrawing money. Without the arrow, it just shows involvement, but we don't know who initiates the task.

**Multiplicity:** Multiplicity tells us how many times an actor can be involved in a task. Numbers next to the line show this.

*Example:* If a Customer is linked to a Purchase task with multiplicity 0..1, it means they can either make one purchase or none at all. If it's 1..\*, it means the customer can make as many purchases as they want.

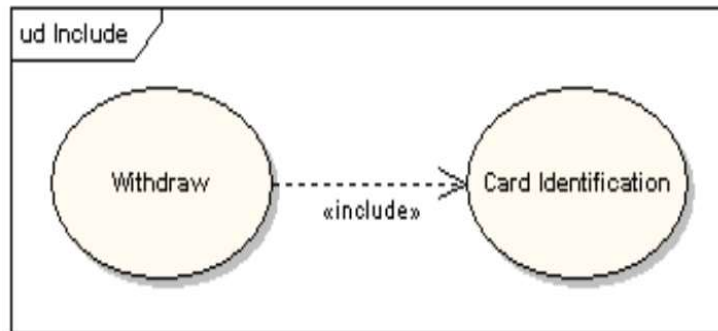
Use multiplicity when you need to explain how many times a person or system can do something. For example, a customer might log in only once, but they might make many purchases.

### **Special Relationships in a Use Case Diagram**

Sometimes, use cases can be related to each other in special ways. These relationships help make the diagram clearer.

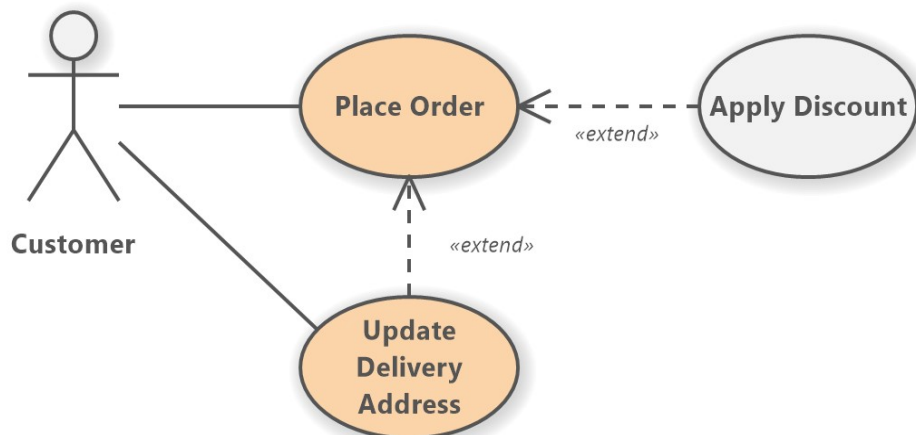
**1. Include:** This means that one use case **must always include** another use case to complete its task. It is a way of saying that a certain action is part of a larger action.

- **Example:** In an ATM system, before a customer can "Withdraw Cash," they must first "Authenticate" themselves (by entering a PIN). So, "Authenticate User" is always included in "Withdraw Cash."
- **Symbol:** This relationship is shown with a **dashed arrow** labelled <<include>>.



**2. Extend:** This means that one use case can **add extra steps** or **features** to another use case, but only if needed. It is like an optional action that only happens under certain conditions.

- **Example:** After withdrawing cash, the customer might want to check their balance. This means the "Check Balance" use case extends the "Withdraw Cash" use case.
- **Symbol:** This is shown with a **dashed arrow** labelled <<extend>>.



## Steps to Create a Use Case Diagram for an ATM System

Use Case Diagram for an ATM system.

### Step 1: Identify the Actors

- **Customer:** This is the person using the ATM to do tasks like withdrawing money or checking their account balance.
- **Bank:** This represents the system or organization that processes the customer's transactions.

### Step 2: Identify the Use Cases

- **Withdraw Cash:** The customer takes money out of their account using the ATM.
- **Check Balance:** The customer checks how much money is in their account.
- **Deposit Cash:** The customer puts money into their account.
- **Transfer Funds:** The customer moves money from one account to another.
- **Change PIN:** The customer changes the PIN number used to access the ATM.

### Step 3: Draw the Diagram

- First, draw a large rectangle to represent the system boundary and label it "ATM System."
- Inside the rectangle, draw ovals for each use case (Withdraw Cash, Check Balance, etc.).
- Outside the rectangle, draw stick figures for the actors (Customer, Bank).
- Connect the actors to the use cases using straight lines to show what actions the actors can perform.

### Step 4: Use Include and Extend

- **Include:** The "Withdraw Cash" use case always includes "Authenticate User," since the customer must be authenticated before they can withdraw money. Draw a dashed arrow from "Withdraw Cash" to "Authenticate User" and label it <<include>>.

- **Extend:** The "Check Balance" use case extends "Withdraw Cash" because, after withdrawing cash, the customer may want to check their balance. Draw a dashed arrow from "Check Balance" to "Withdraw Cash" and label it <<extend>>.

