

SE 3140

DESIGN AND MODELLING

By M. Mangong C. Fosah

mangong.clement@ictuniversity.org

maclef88@yahoo.com

(+237) 653 519 879

M. Mangong Clement

I. Use case modelling

- **Requirement modelling**
- **Use cases**
- **Actors**
- **The Include Relationship**
- **The Extend Relationship**
- **Exercises/Assignment**

I. Use case modelling

- ❑ The requirements of a system describe
 - **What the user expects from the system.**
 - **What the system will do for the users.**
- ❑ How to view the system when defining the requirements of the system
 - **View it as a black box so as to consider only the external characteristics of the system**

I. Use case modelling

- ❑ What type of requirement should you consider when defining the requirements
 - **Both functional and nonfunctional requirements**
- ❑ What makes up Requirements modeling?
 - **Requirement analysis + requirement specification**

I. Use case modelling

- ❑ What is the role of using use case modeling
 - **It's an approach for describing the functional requirements of the system.**
- ❑ How is the system's data requirement that is stored by the system describe?
 - **The static modeling approach is use**

I. Use case modelling

- ❑ How is the use case and static modeling related in terms of input and output data?
- **Input and output data from the system are describe initially in the use case models and then specified in more details during the static modeling.**

I. Use case modelling : *Requirement Modeling(1)*

- ❑ The requirements of **each group users** of the system must be understood and specified.

I. Use case modelling : Requirement Analysis(1)

- ❑ Software requirements describe the functionality that the system must provide for the users.
- ❑ Requirement analysis involves analyzing the requirements

Examples

- *By interviewing users*
- *By analyzing the existing system(s)*

I. Use case modelling : Requirement Analysis(2)

Some of the questions to ask

1. What is your role in the current system (*manual or automated*)
2. How do you use the current system?
3. What are the advantages and limitations of the current system?
4. What features should the new system provide for you?

I. Use case modelling : Requirement Analysis(3)

❑ Important point that involves analyzing an existing manual system

- Understanding and documenting the current system.
- Determining which features of the current system should be automated.
- Determining which features of the current system should remain manual.
- Discussing with users what functions could be done differently when the system is automated.

I. Use case modelling : Requirement Analysis(4)

□ Important point that involves analyzing and existing software system

- Extracting the software requirements
- Separating functional requirements from functions that result from design or implementation decisions
- Identifying nonfunctional requirements
- Deciding what functions should be done differently
- Deciding what new functions should be added

Use case modelling : Requirement Specification(1)

□ Important point that involves analyzing an existing software system

- After analysis, the requirements need to be specified
- The requirement specification is the document that needs to be agreed on by the requirement analysts and the users.
- The document must also be understood by the developers
- It supposed to have both the functional requirement and the nonfunctional requirement

Use case modelling : Requirement Specification(2)

Basic information about what the functional requirement should specify

- Describe what functionality the system needs to provide
- What information needs to be input to the system from an external environment (external users , systems and devices)
- What the system need to output to the external environment
- What stored information the system reads or updates.

Use case modelling : Requirement Specification(3)

Basic information about what the functional requirement should specify

Example

- Describe the functional requirement to view the balance of a bank account.

Use case modelling : Requirement Specification(4)

Basic information about what the nonfunctional requirement should specify

- It refers to the quality attribute
- It refers to a quality-of-service goal that the system must fulfill

Use case modelling : Requirement Specification(5)

Basic information about what the nonfunctional requirement should specify

Example

- A performance requirement specifying a system response time of 2 seconds
- An availability requirement specifying a system must be operational for 99% of the time
- A security requirement such as protection from system penetration

Use case modelling : Requirement Specification(6)

□ Quality Attributes of software Requirement specification

- Correct
- Complete
- Unambiguous
- Consistent
- Verifiable
- Understandable by non-computer specialists
- Modifiable
- Traceable

Use case modelling : Use Cases

The Use case modeling approach

Use case modelling : Use case approach

➤ *Types of Requirements*

❑ ***User Requirements*** (written for customers) :

Describe the functional and non functional requirements so that they are understandable by system users without detailed technical knowledge.

Use case modelling : Use case approach

➤ *Types of Requirements*

- ❑ ***System Requirements*** (written as a contract between client and contractor
 - ✓ Expands the version of the user requirements that are used by engineers as the starting point for the system design.
 - ✓ They add details and explain how the user requirements should be provided by the system.

Use case modelling : Use case approach

➤ *Types of Requirements*

❑ ***Software requirement specification (SRS)***

(Detailed software description to serve as the basic for a design or implementation)

- ✓ Written for developers
- ✓ A set of use cases that describe all of the interactions that the users will have in the software (use cases)
- ✓ Both functional and non functional requirements

Use case modelling : Use case approach

□ *Requirement classification*

- ✓ Functional requirements**
- ✓ Non functional requirements**
- ✓ Domain requirements**

Use case modelling : Use case approach

- An approach to developing requirements.
- The requirement that describes the services that the software must offer is called functional requirement.
- The functional requirement is defined in terms of **use cases and actors**.
- An **actor** initiates a use case.
- A **use case** defines a sequence of interaction between one or more actors and the system.

Use case modelling : Use case approach

- A use case model is a *model* of how different *types of users* interact with the system to solve a problem.

Points to Note

- ✓ *A model*
- ✓ *Types of users*
- ✓ *Interaction of the users with the system*

Use case modelling : Use case approach

- The use case model considers the ***system as a black box*** and describes the interactions between the actor(s) and the system in a narrative form consisting of ***user inputs*** and ***system responses***.

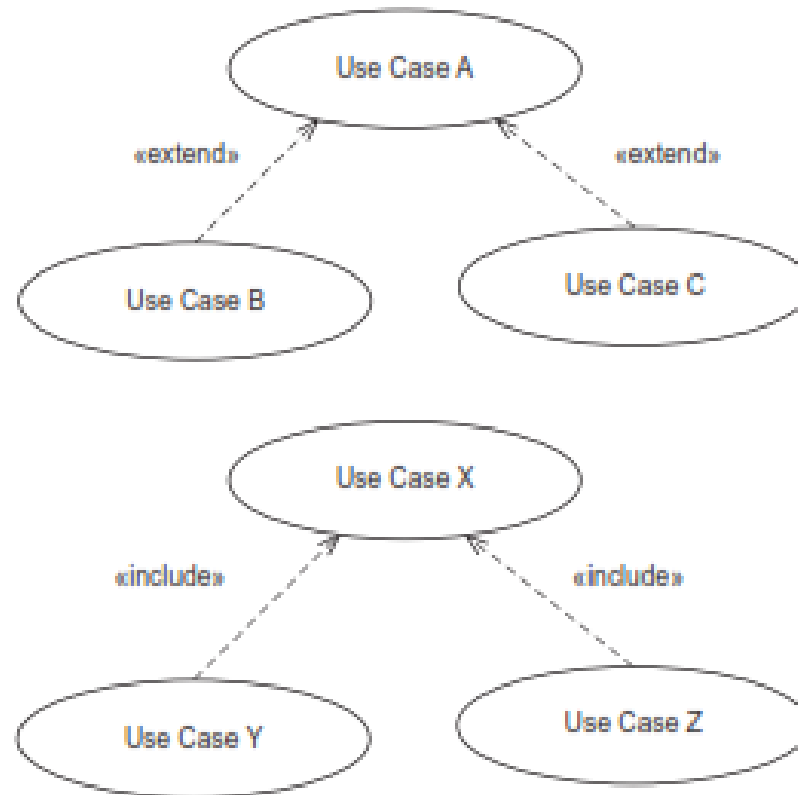
Use case modelling : Use case approach

➤ UML Notation for a use case diagram(1)




Use case modelling : Use case approach

➤ UML Notation for a use case diagram(2)



Use case modelling : Use case approach

➤ UML Notation for a use case diagram(3)

Use case : 

- A system function (process-automated or manual)
- Named by **verb + Noun**(or Noun Phrase)
- i.e Do something
- An actor must be linked to a use case, while some use cases may not be linked to actors.

Use case modelling : Use case approach

➤ UML Notation for a use case diagram(4)

Communication link :

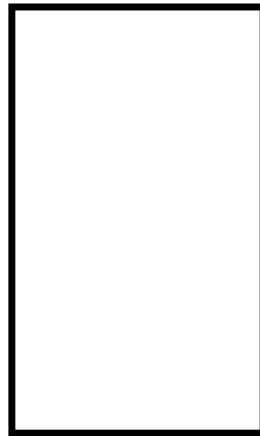
- The participation of an actor in a use case is shown by connecting an actor to a use by a solid link.
- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

Use case modelling : Use case approach

- UML Notation for a use case diagram(5)

Boundary of system

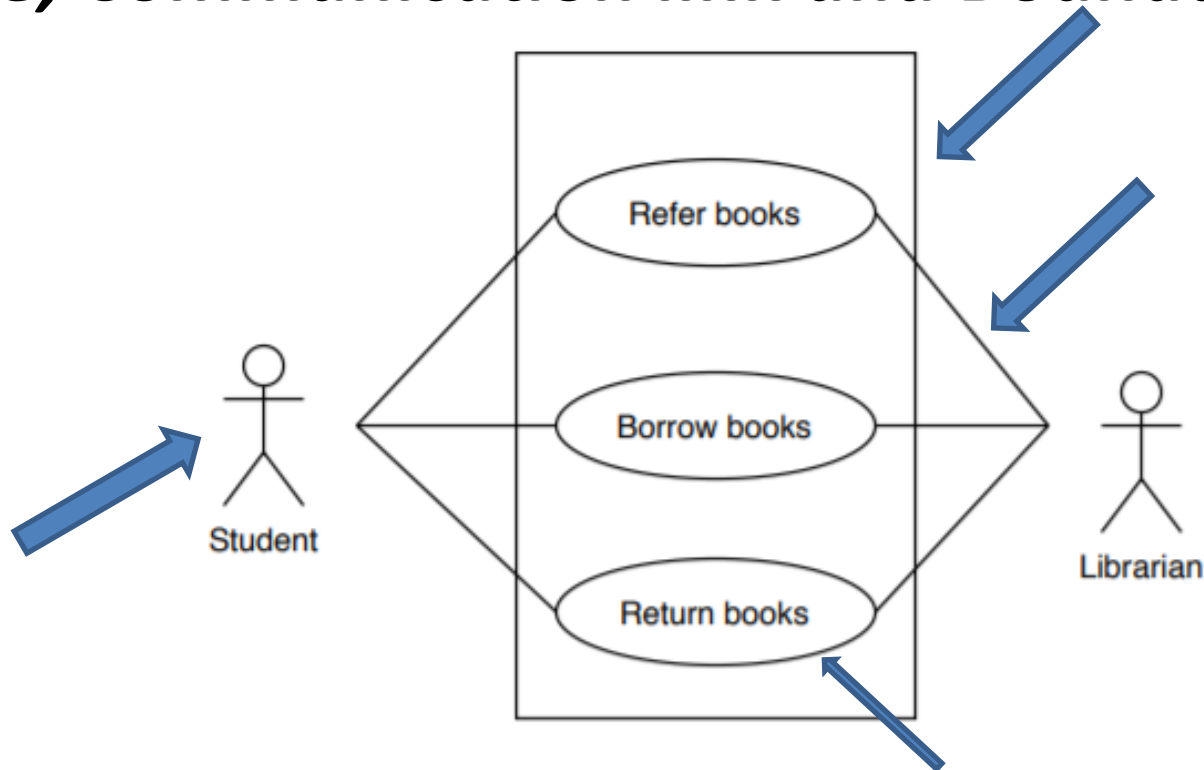
- The system boundary is potentially the entire system as defined in the requirement document



Use case modelling : Use case approach

- UML Notation for a use case diagram(6)

Use case, Communication link and Boundary of system



Use case modelling : Use case approach

- The model treats **what** the system does in response to the **actor's inputs**, not the internal of **how** it does it.

NB : Requirement is all about what? and not How?

Use case modelling : *How does a use case starts? (1)*

- It starts with *input from an actor*.
- An *actor provides input to the system* and the *system provides responses to the actor*.
- Some use case model have simple use case while others have complex use cases.

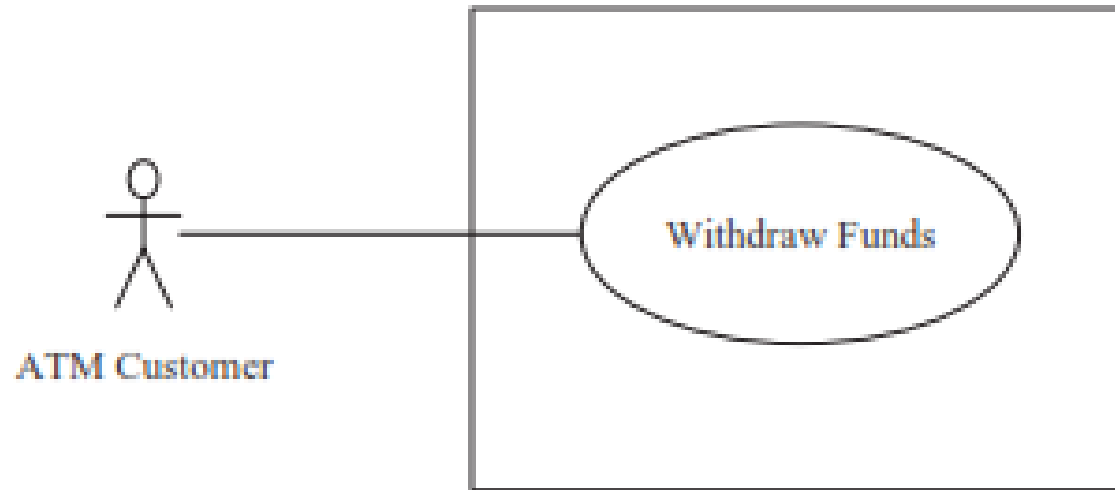
Use case modelling : *How does a use case starts?*

- A simple use case may consist of ***one interaction between an actor and the system***
- A more complex use case will consist of ***several interactions between the actor and the system*** and ***some may involve more than one actor.***

NB: Use cases are what happens when actors interact with the system.

Use case modelling : *Example(1)*

- What is the lone use case for an Automated Teller Machine (ATM) customer? (Withdraw funds)



Use case modelling : *Example(2)*

- The use case starts with when the customer inserts an ATM card into the card reader.
- The response to the system prompt for the PIN, and eventually receive the cash dispensed by the ATM machine.

Use case modelling : A simple Use Case

A use case description consist of five major points

1. The name of the use case
2. The name of the actor.
3. A one-sentence use case summary.
4. The description of the main sequence of events .
5. The description of any alternative to the main sequence.

Use case modelling : A simple Use Case - Example

Consider view alarms from the emergency Monitoring system.

- There is one actor, the monitoring operator, who can request to view the status of all alarms.
- We shall have the essentials of the use case description.

Use case modelling : A simple Use Case - Example

1. The name of the use case , **View Alarm**
2. The name of the actor, ***Monitoring Operator***
3. A one-sentence use case summary, **which provides a brief description**
4. The description of the main sequence of events. ***For this case , the first step is the operator request and the second step is the system response.***
5. Alternative description, the could be at step 2.

Use case modelling : A simple Use Case - Example

Use case Name: view Alarms

Summary: The monitoring operator views outstanding alarms

Actor: Monitoring operator

Main sequence:

The monitoring operator requests to view the outstanding alarms

The system displays the outstanding alarms.

For each alarms , the system displays the name of the alarms, alarm description, location of alarm, and severity of alarm(high, medium, low)

Alternative sequences:

Step 2: Emergency situation. System displays emergency warning messages to operator.

Use case modelling : A simple Use Case - Example

Use case Name: view Alarms

Summary: The monitoring operator views outstanding alarms

Actor: Monitoring operator

Main sequence:

The monitoring operator requests to view the outstanding alarms

The system displays the outstanding alarms.

For each alarms , the system displays the name of the alarms, alarm description, location of alarm, and severity of alarm(high, medium, low)

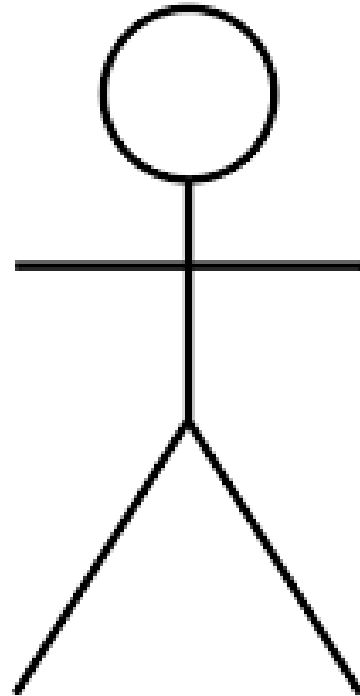
Alternative sequences:

Step 2: Emergency situation. System displays emergency warning messages to operator.

Use case modelling : Actors(1)

- An actor *represents a role of a user* that interacts with the system that you are modeling.
- It characterizes an external user (*outside the system*) that interacts with the system.
- Actors are the *only external entities that interact with the system*

Use case modelling : Actor UML Notation



Actor Role Name

Use case modelling : Actors, Roles and Users (1)

- Actor represents a role played in the application domain.
- A *user is an individual* whereas an *actor represents the role* played by all users of the same type.
- A user can be a human user, organization, machine, another external system

Use case modelling : Actors, Roles and Users (2)

- A user can be a human user, organization, machine, another external system.
- An actor can perform a single use case
- An actor can be a person, such as a customer
- An actor can be a computer such as a database system or server.

Use case modelling : Actors, Roles and Users (3)

- We may have several application users, the user(s) play one or many roles, each of the role is represented by an actor.
- The use case diagram uses the terminology of an actor (the role the user plays and not the user)

Use case modelling : Actors, Roles and Users (4)

Example:

- There are several customers in the banking system, which are all represented by the ATM customer actor.
- ATM customer actor models a user type, and individual customer are instances of the actor.

Use case modelling : Actors, Roles and Users (5)

- You can represent multiple users with a single actor and a single user can have the role of multiple actors.
- Multiple users with a single actor
- Single role of multiple actors

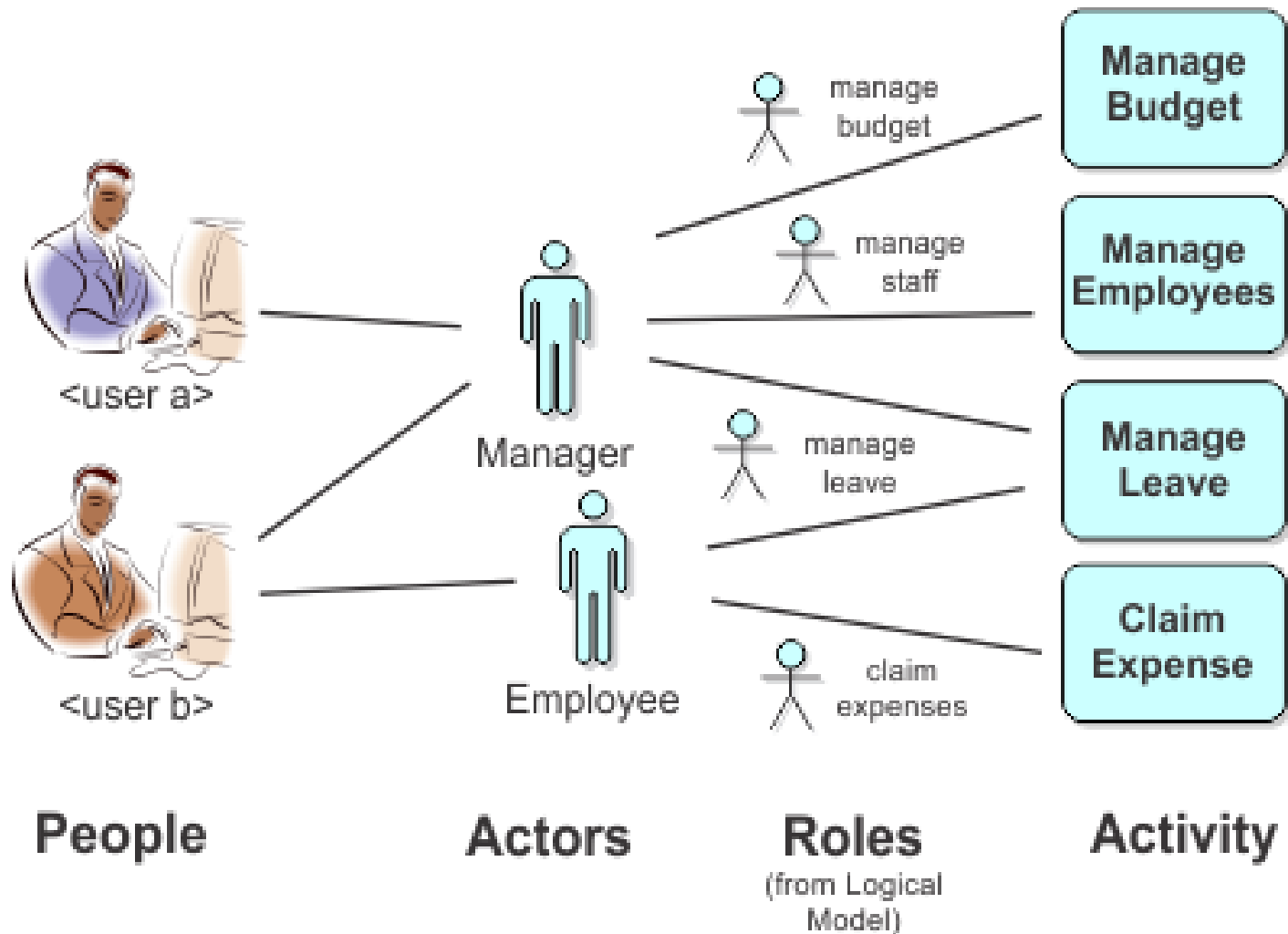
Use case modelling : Actors, Roles and Users (6)

- Actors are external to the system.
- They can initiate the behavior described in the use case or be acted upon by the use case.
- Actors can also exchange data with the system.

Use case modelling : Actors, Roles and Users (4)

- In most systems, actors are humans.
- In more complex systems, an actor can be an external I/O device or a timer.
- External I/O devices or a timer are particularly use in real-time embedded systems, in which the system interacts with the external environment through a sensors and actuators.

Use case modelling : Actors, Roles and Users (5)



Use case modelling : Primary & Secondary actors (1)

We have two major types of actors

- **Primary actor** initiate a use case while the **secondary actors** are used by the system but they do not interact with the system on their own.

Primary actor is independent while secondary actor depends on the system.

Use case modelling : Primary actors (1)

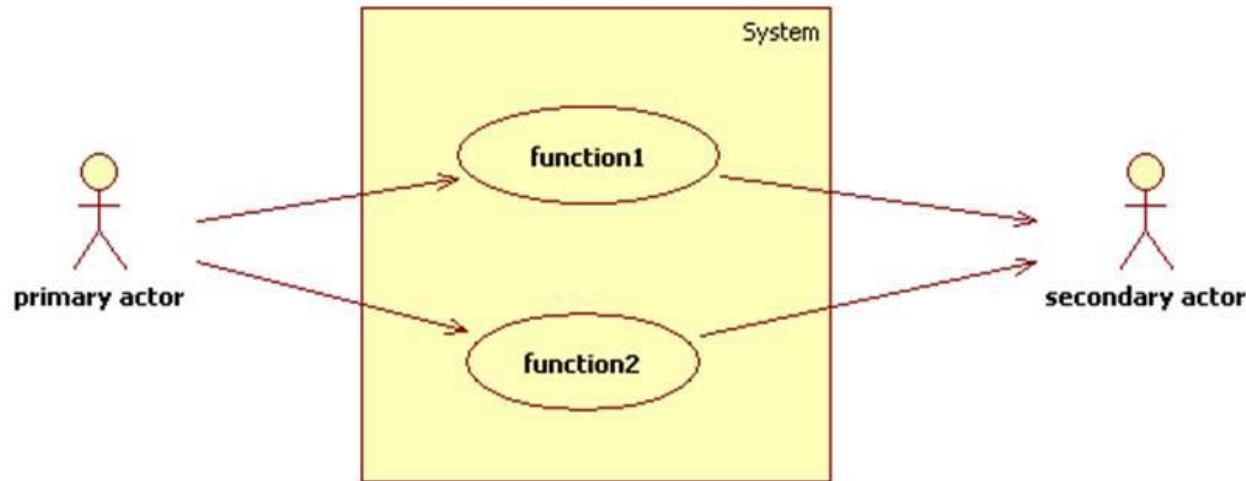
Who is a primary actor?

- Actor that calls on the system to deliver one of its services. Always place on the left side.
- The use case starts with an input from the primary actor to which the system has to respond.
- It initiates use cases and interact with the system

Use case modelling : Secondary actors (1)

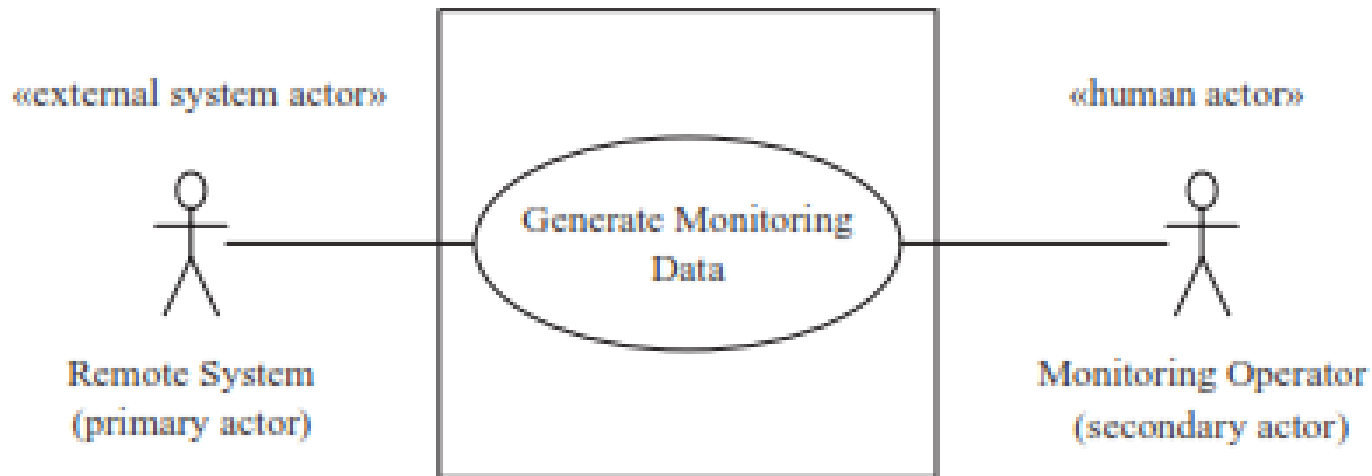
Who are secondary actors?

- Other actors do not initiate the use case, but can participate in the use case.



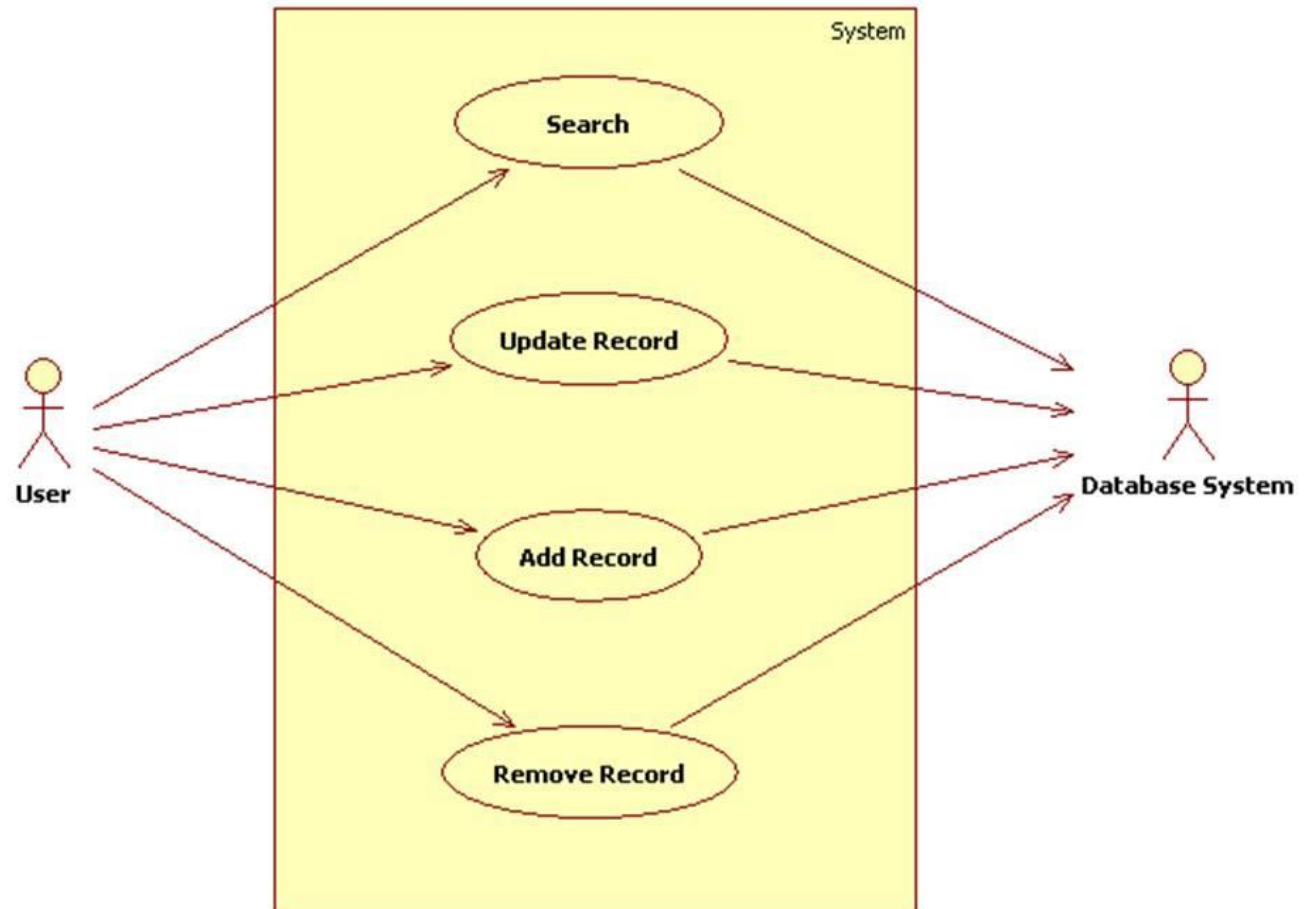
Use case modelling : Primary & Secondary actors

Example(1)



Use case modelling : Primary and Secondary actors

Example(2)



Use case modelling : Primary and Secondary actors

An actor that is both a Primary and Secondary actor

A primary actor in one use case can be a secondary actor in another use case, at least one of them gain value from the system.

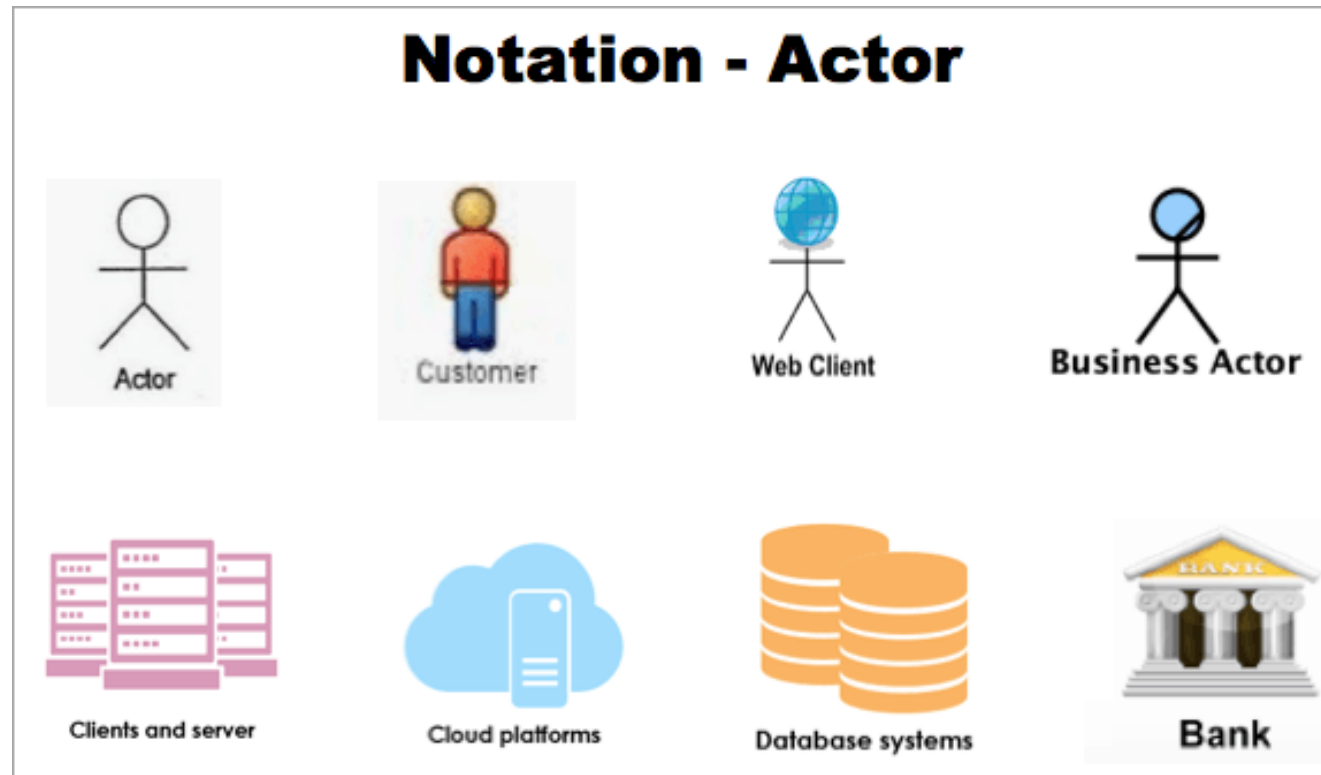
Use case modelling : Primary and Secondary actors

Example:

- The Remote system actor initiates the Generate monitoring data use case, in which the remote system sends monitoring data that is displayed to monitoring operators.
- The remote system is the primary actor that initiates the use case
- The monitoring operator is a secondary actor who receives the monitoring data and hence, gains value from the use case.

Use case modelling : Modeling Actors

We have several actors notations (tool oriented)



Use case modelling : Modeling Actors(1)

1. Human actors

- Typically uses various I/O devices to physically interact with the system.
- Usually interact with the system via standard I/O devices, such as keyboard, display, or mouse.
- It can also occur via sensors.
- This type of actor is an end-user

Use case modelling : Modeling Actors(2)

1. Human actors : Example

- The Emergency Response System has the monitoring operator as the human actor that interacts via the standard I/O devices
- The ATM customer is also a human actor who interacts with the Banking system by using several I/O devices, including a card reader, cash dispense, and receipt printer , in addition to keyboard and display.

Use case modelling : Modeling Actors(3)

2. External system actors : Example:

May be a primary or secondary actor

Use case modelling : Modeling Actors(4)

Input device actor or an input/out device actors :

- In some cases, an actor can be an input device actor or an input/output device actor.
- This can happen when there is no human involvement with the use case and the actor providing external input to the system is an input device or I/O device

Use case modelling : Modeling Actors(5)

Example:

The input device actor interacts with the system via a sensor.

The Monitoring sensor, which provides sensor input to the Generate Alarm use case.

Use case modelling : Modeling Actors(6)

3. Timer actor

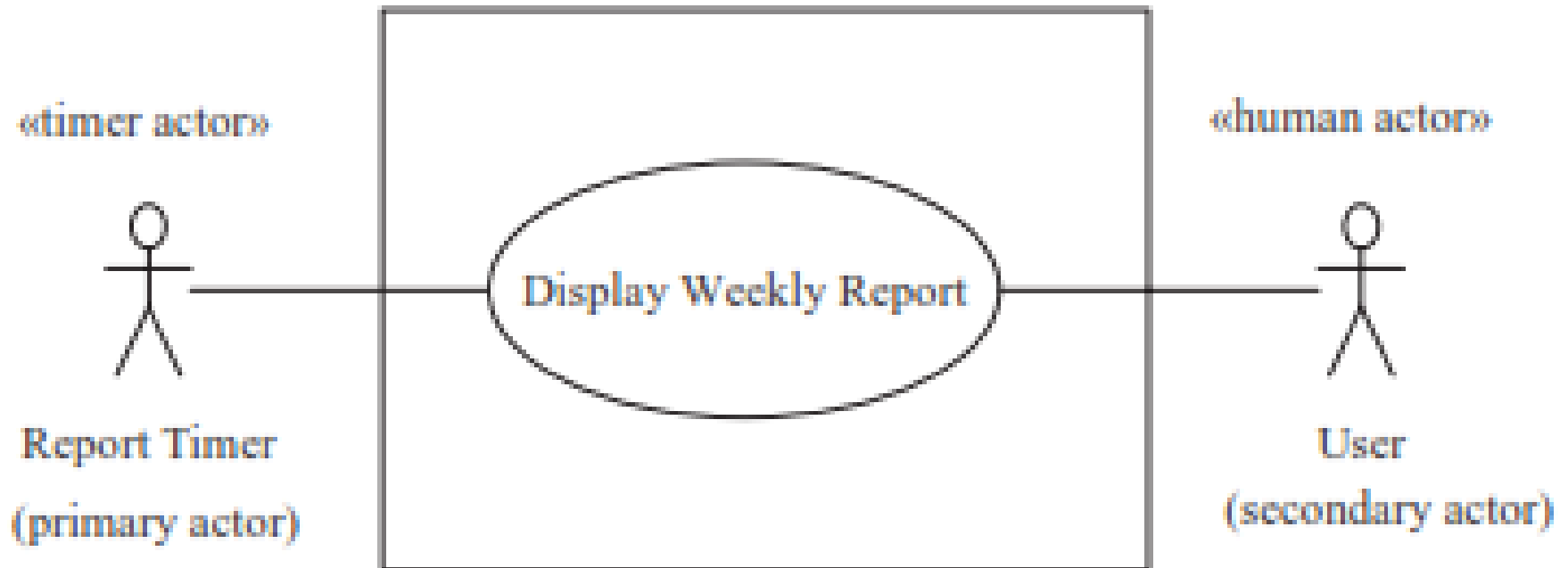
- A type of actor that periodically sends timer events to the system. Periodic use cases are needed when certain information needs to be output by the system on a regular basis.

Use case modelling : Modeling Actors(7)

3. Timer actor : Example

Display weekly Report, the primary actor is Report timer, and the secondary actor is a User, a Report timer actor initiates the Display Daily Report use case, which periodically prepares a daily report and displays it to the user.

Use case modelling : Modeling Actors(8)



Use case modelling : Generalization & specialization(1)

Generalization and specialization of actors

The possibility for a human user to play two or more independent roles, this is represented by a different actor for each role.

Use case modelling : Generalization & specialization(2)

Generalization and specialization of actors

In other cases, different actors might have some roles in common but other roles that are different.

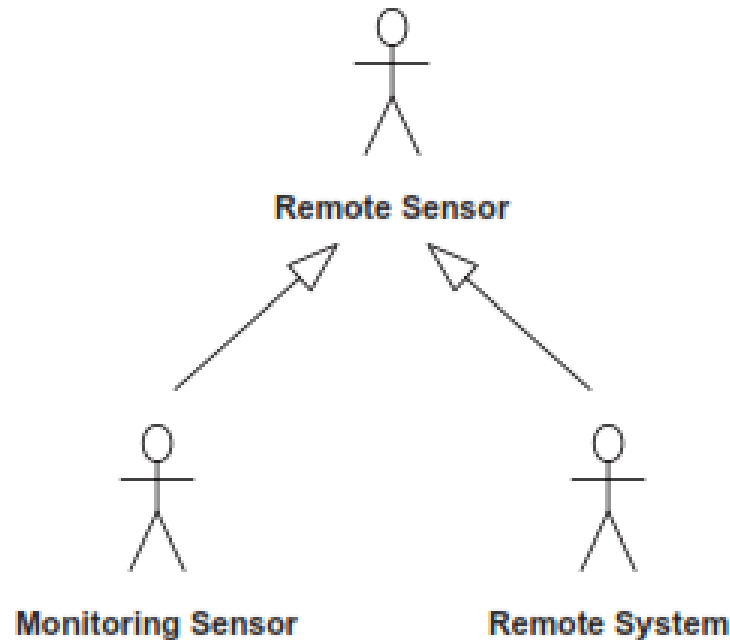
In this situation, the actors can be generalized, so that the common part of their roles is captured as a generalized actor and the different parts by specialized actors.

Use case modelling : Example

Generalization and specialization of actors

Generalization: Remote sensor

Specialization: Monitoring sensor and Remote sensor



Use case modelling : Who is an actor?

- In the case of Report Stolen card, in which a user actor phones the bank to inform them that the ATM card has been stolen, it would be obvious that the customer is the actor.
- However, if the customer in fact talks over the phone to a bank clerk, who actually enters the information into the system, then it is the clerk who is the actor.

Use case modelling : Identifying the Use cases

- Identify the actors and the interactions they have with the system.
- Each use case describes a sequence of interactions between the actor and the system.
- This gives room for the functional requirement to be described in terms of the use cases, which constitute a functional specification of a system.

Use case modelling : Identifying the Use cases

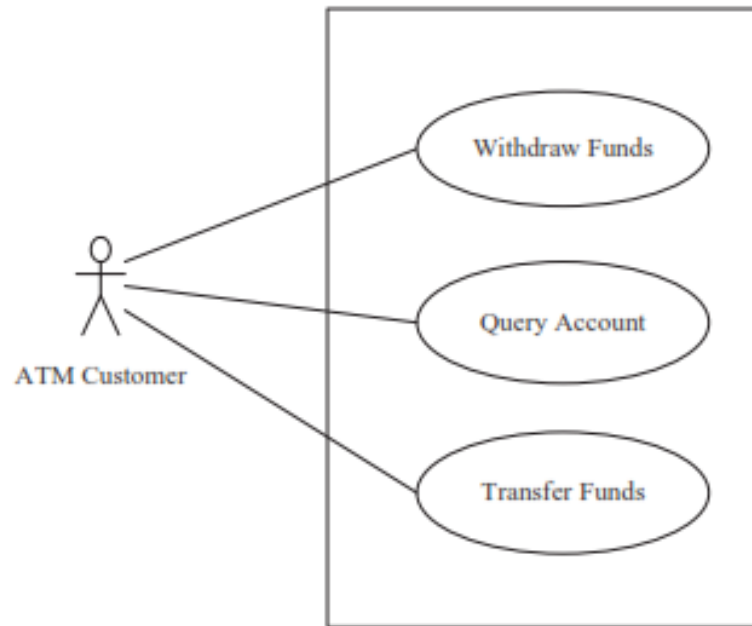
- Avoid functional decomposition when developing use cases, in which several small use cases describe small individual functions of the system rather than describe a sequence of events that provides a useful result to the actor.

Use case modelling : Identifying the Use cases

- ATM customer actor is also allowed to query an account or transfer funds between two accounts. Because these are distinct functions initiated by the customer with different useful results.
- The query and transfer functions should be modeled as separate use cases, rather than being part of the original use case.

Use case modelling : Identifying the Use cases

Thus the customer can initiate three use cases :
Withdraw funds, Query Account, Transfer Funds



Backing System actor and use cases

Use case modelling : A scenario

- ***Each sequence through the use case*** is called a scenario
- A use case usually describes several scenarios, ***one main sequence*** and a number of ***alternative sequences***.
- A scenario is a complete sequence, through the use case, so a scenario could start out executing the main sequence and then follow the alternative branch at the decision point.

Use case modelling : Documenting Use case

To completely document each use case in the use case model, we have the following

1. Use case Name
2. Summary
3. Dependency
4. Actors
5. Preconditions
6. Description of main sequence
7. Description of alternative sequence
8. Nonfunctional requirements
9. Outstanding questions

Use case modelling : Use case Relationships

- Complex use cases entitles dependencies between them defined by using the *include and extend relationships*.

What is the objective of the relationship?

- To maximize the extensibility and reuse of use cases.

Use case modelling : Use case Relationships(1)

Putting several common sequence of interaction together.

- *Inclusive use cases* are determined to identify common sequences of interactions in several uses cases.
- The use cases can be further extracted and reused.

Use case modelling : Use case Relationships(2)

The include Relationship

- A relationship in which one use case (the base use case) includes the functionality of another use case (the inclusion use case)
- It supports the reuse of functionality in a use-case model.

Use case modelling : Use case Relationships(3)

The include Relationship

- A case where common sequences of interactions between the actors and the system can be determined in spanning several use cases.
- These common sequences of interactions reflect functionality that is common to more than one use case.

Use case modelling : Use case Relationships(4)

The include Relationship

- ***A common sequence of interaction*** can be extracted from several of the original use cases and made into a **new use case** which is called an **inclusion use case.**

Use case modelling : Use case Relationships (5)

The include Relationship

- An inclusion use case is usually abstract: that is cannot be executed on its own.
- An abstract use case must be executed as part of a concrete- that is , executable –use case
- When the common functionality is separated into an inclusion use case, this use case can now be reused by other use cases.

Use case modelling : Use case Relationships(6)

The include Relationship

- It's then possible to define a more concise version of the old use case, with the common interaction sequence removed.
- This version of the old use case is referred to as a base use case or concrete use case, which includes the inclusion use case.

Use case modelling : Use case Relationships(7)

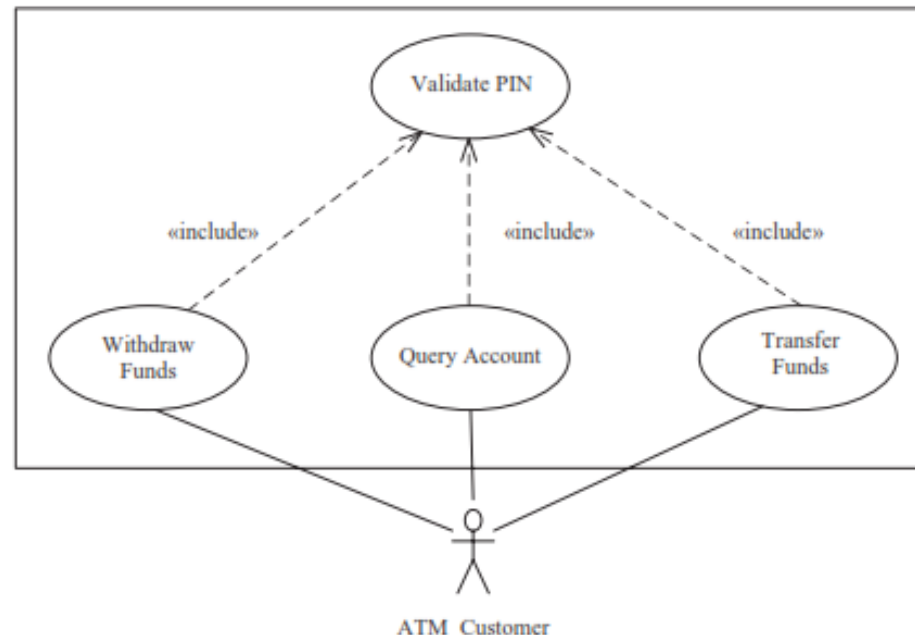
The include Relationship

- **Inclusion use cases** always reflect the functionality that is common to more than one use case.

Use case modelling : Use case Relationships(8(1))

The include Relationship

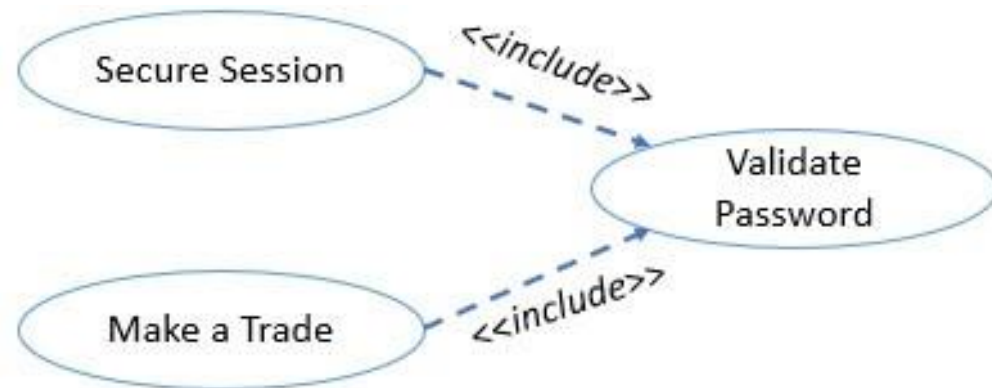
- Example of an inclusive use case and include relationships



Use case modelling : Use case Relationships(8(2))

The include Relationship

- Example of an inclusive use case and include relationships

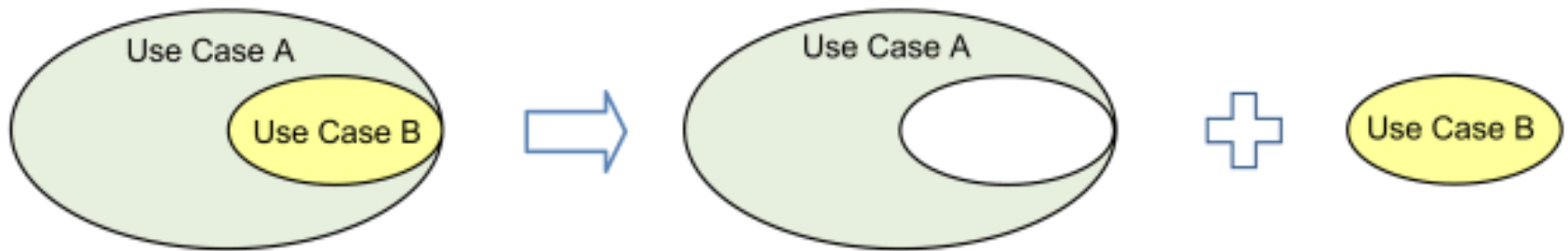


Include Use Case Relationship

Use case modelling : Use case Relationships(8(3))

The include Relationship

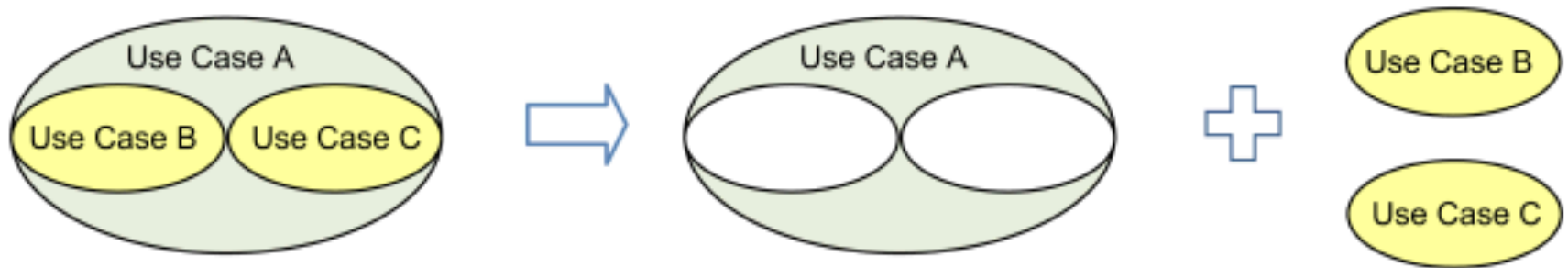
- **Example of an inclusive use case and include relationships**



Use case modelling : Use case Relationships(8(4))

The include Relationship

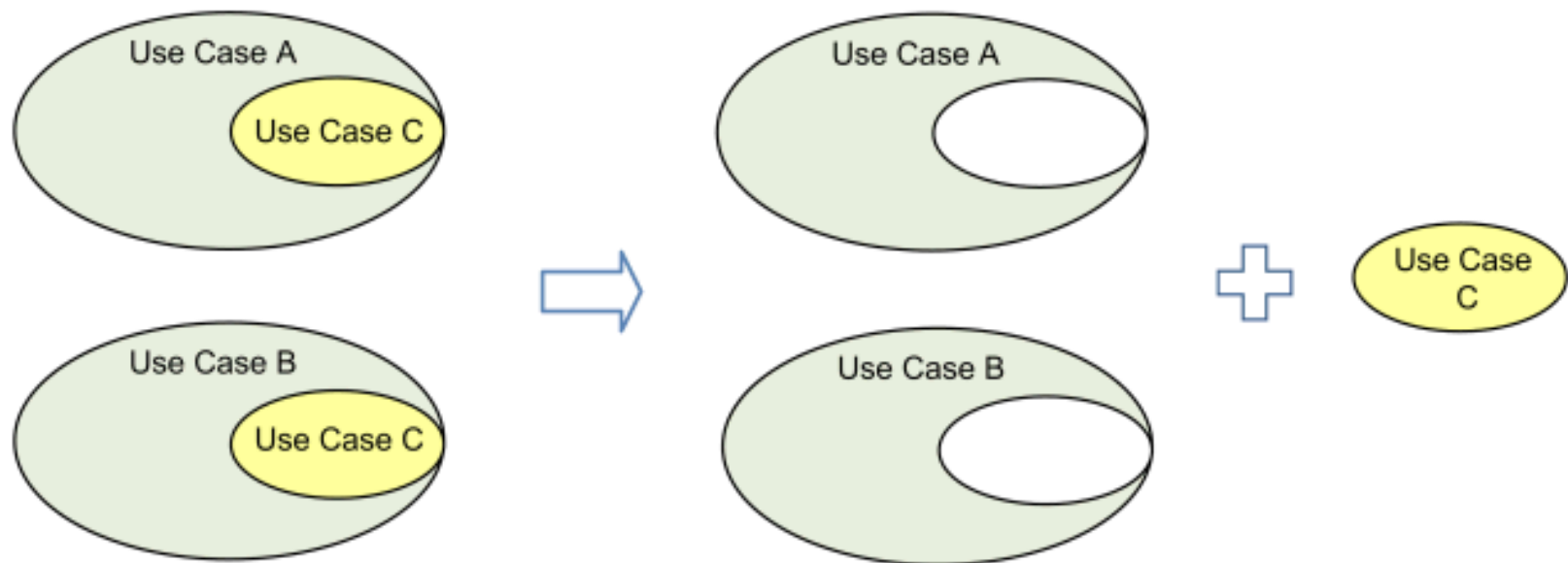
- Use case B and C are extracted from larger use case A into separate use cases.



Use case modelling : Use case Relationships(8(5))

The include Relationship

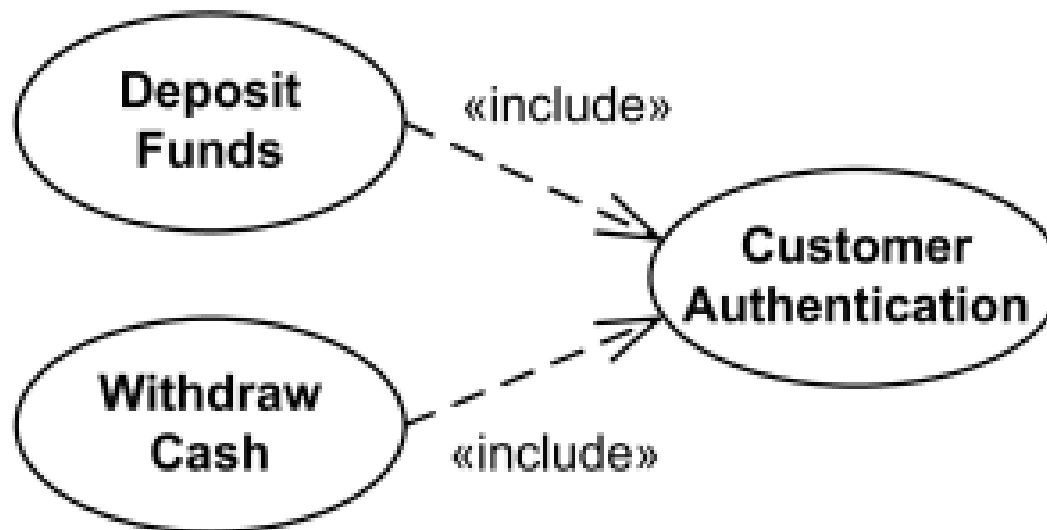
- Use case c is extracted from use cases A and B to be reused by both use cases



Use case modelling : Use case Relationships(8(6))

The include Relationship

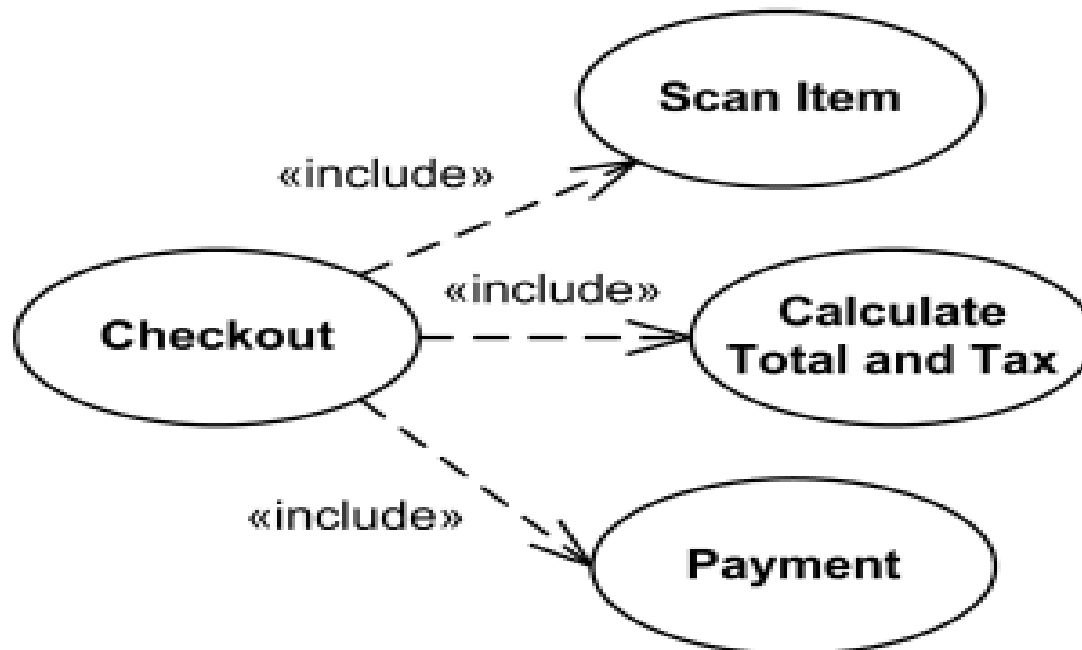
- **Deposit Funds and withdraw case use cases include Customer Authentication use case**



Use case modelling : Use case Relationships(8(7))

The include Relationship

- Checkout use case includes several use cases – Scan Item, Calculate Total and Tax , and Payment



Use case modelling : Use case Relationships(9)

The Extend Relationship

- A use case may become complex and has many alternative branches.
- The extend relationship is used to model alternative paths that a use case might take where it has many alternative, optional and exceptional sequences of interactions.

Use case modelling : Use case Relationships(10)

The Extend Relationship

- The solutions is the split off an alternative or optional sequence of interactions into a separate use cases.
- The purpose of the new use case is to extend the old use case, if the appropriate condition holds.

Use case modelling : Use case Relationships(11)

The Extend Relationship

- The use case that is extended is called the based use case and the use case that does the extending is referred to as the extensive use case.
- Under certain conditions , a base use case be extended by a description given in the extension use case.

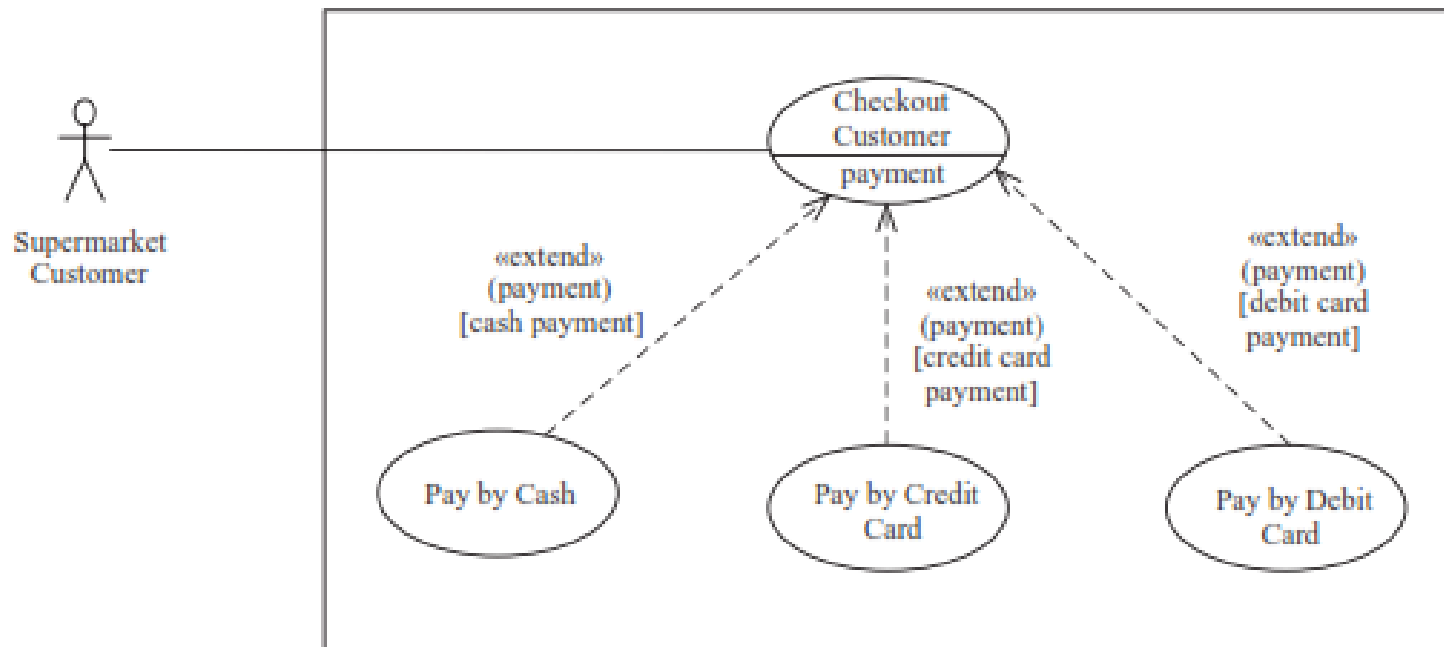
Use case modelling : Use case Relationships(12)

The Extend Relationship

- The base use case can be extended in different ways , depending on which condition is true.
- Extending is referred to as the extensive use case.

Use case modelling : Use case Relationships(13)

The Extend Relationship and extension use cases



Use case modelling : Use case Relationships(14)

The extend relationship can be used as follows:

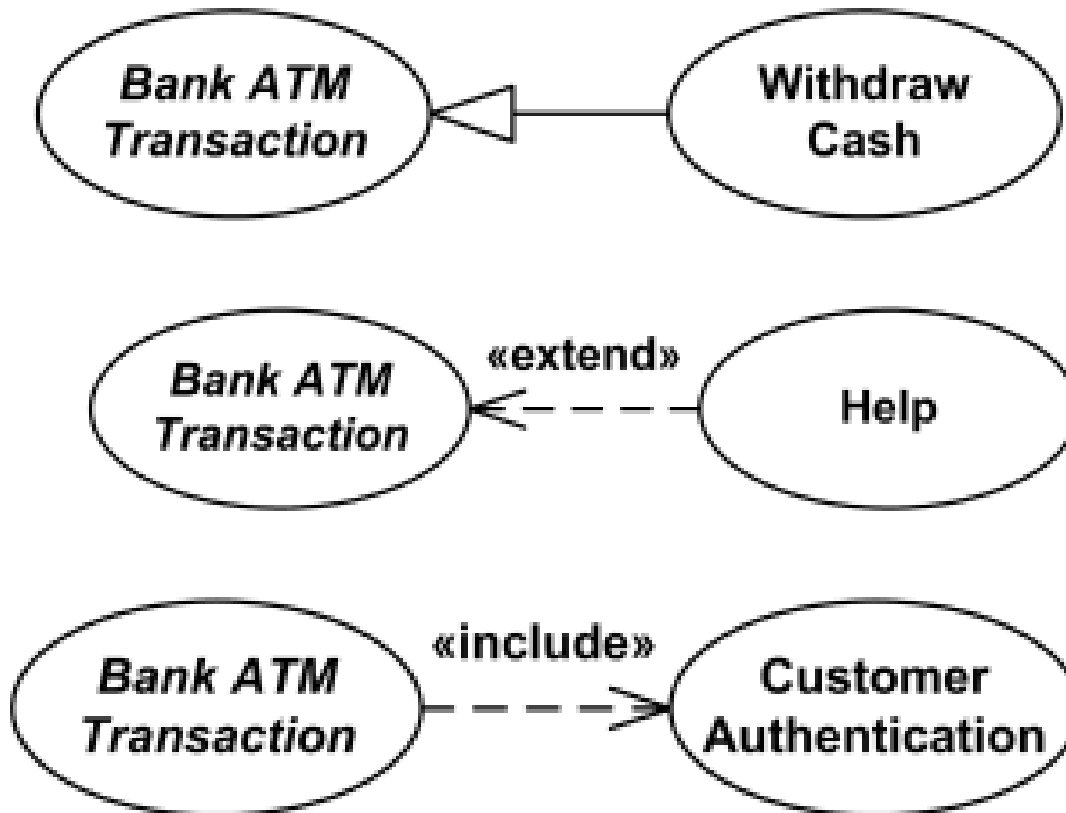
1. To show a conditional part of the base use case that is executed only under certain circumstances
2. To model complex or alternative paths

Use case modelling : Use case Relationships(15)

- The base use case does not depend on the extension use case.
- The extension use case depends on the base use case and executes only if the condition in the base use case that causes it to execute is true.
- An extension use case usually extends only one base use case; it is possible for it to extend more than one.
- A base use case can be extended by more than one extensive use case

Use case modelling : Use case Relationships(16)

Identify the relationships



Use case modelling : Extension Points

- Use to specify the precise locations in the base use case at which extension can be added
- The extension use case may extend the base use case only at these extension points
- Extension point is given a name
- The extension use case has one insertion segment for the extension point

Use case modelling : Extension Points(1)

- This segment is inserted at the location of its extension point in the base use case
- The extend relationship can be conditional , that must be true for the extension use case to be invoked
- We can have more than one extension use case satisfying a different condition
- A segment defines a behavior sequence to be executed when the extension point is reached

Use case modelling : Extension Points(2)

- When an instance of the use case is extended and reaches the extension point in the base use case, if the condition is satisfied, then execution of the use case is transferred to the corresponding segment in the extension use case
- Execution transfers back to the base use case after completion of the segment

Use case modelling : Extension Points(3)

- An extension point with multiple extension use cases can be used to model several alternatives in which each extension use case specifies a different alternative.
- The extension conditions are design in such a way that only one condition can be true and hence only one extension use case selected, for any given situation
- The values of the extension point are set during runtime execution of the use case.

Use case modelling : Extension Points(5)

- An extension point with multiple extension use cases can be used to model several alternatives in which each extension use case specifies a different alternative.
- The extension conditions are design in such a way that only one condition can be true and hence only one extension use case selected, for any given situation
- The values of the extension point are set during runtime execution of the use case.

Use case modelling : Use case structuring guidelines

- Use case relationship should be carefully used
- It should be noted that small inclusion use cases corresponding to individual functions such as dispense cash, Print receipt, Eject card should not be considered

Use case modelling : Use case structuring guidelines

- These functions are too small , and making them separate use cases would result in a functional decomposition with fragmented use cases in which use case description would be only a sentence each and not a description of a sequence of interactions.

Use case modelling : Use case structuring guidelines

SPECIFYING NONFUNCTIONAL REQUIREMENTS

- Can be specified in a separate section of the use case description.
- For example, for the validate PIN use case, there could be a security requirement that the card number and PIN must be encrypted.
- There could also be a performance requirement that the system must respond to the actor inputs within 5 seconds

Use case modelling : Use case structuring guidelines

SPECIFYING NONFUNCTIONAL REQUIREMENTS

- The nonfunctional requirements can be specified in a separate section of the use case. For the validate PIN use case, they would be describe as follows
- **Security Requirement:** System should encrypt ATM card number and PIN
- **Performance requirement:** System shall respond to actor inputs within 5 seconds

Use case modelling : USE CASE PACKAGES

A good way to handle a large number of use cases in a use case model is to introduce a use case package that groups related use cases

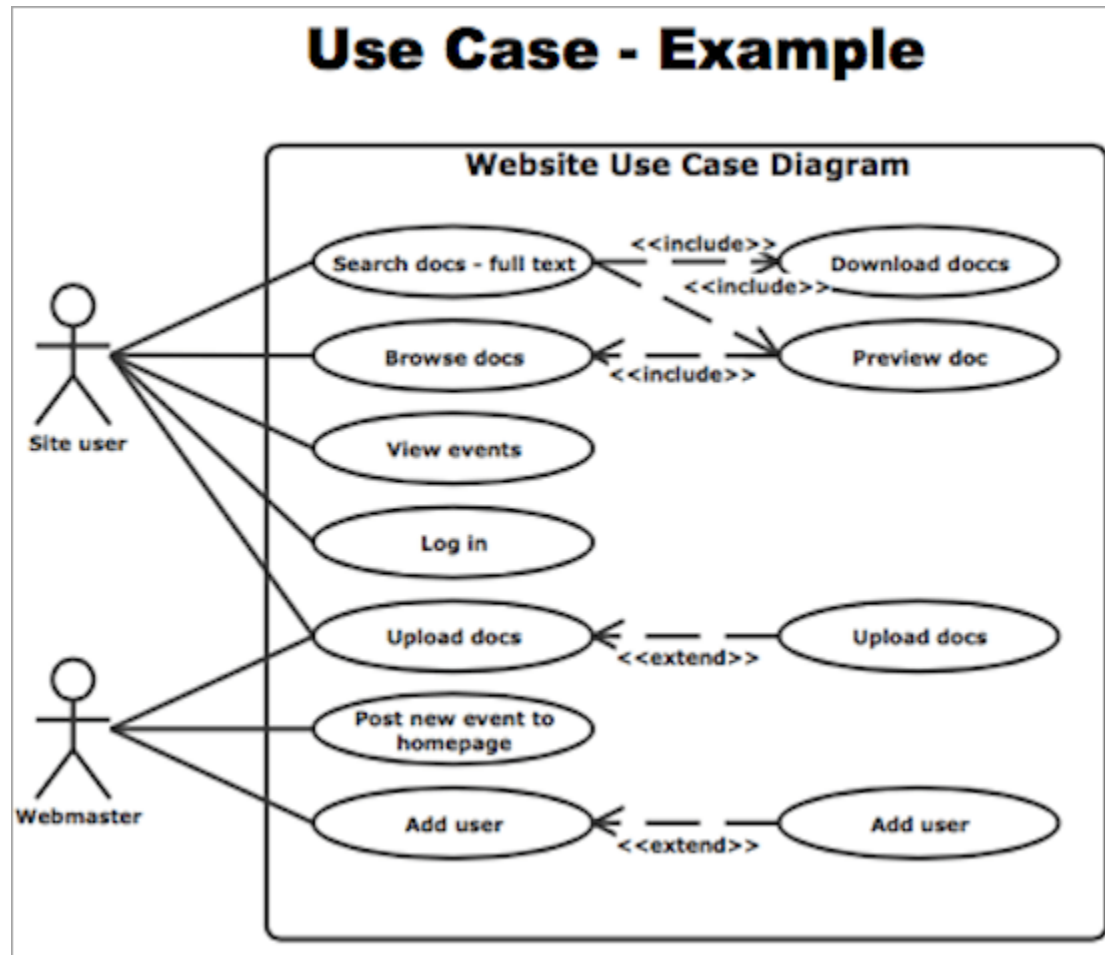
Activity Diagram

- It depicts the flow of control and sequence among activities.
- An activity diagram shows the sequence of activities , decisions nodes , loops and even concurrent activities

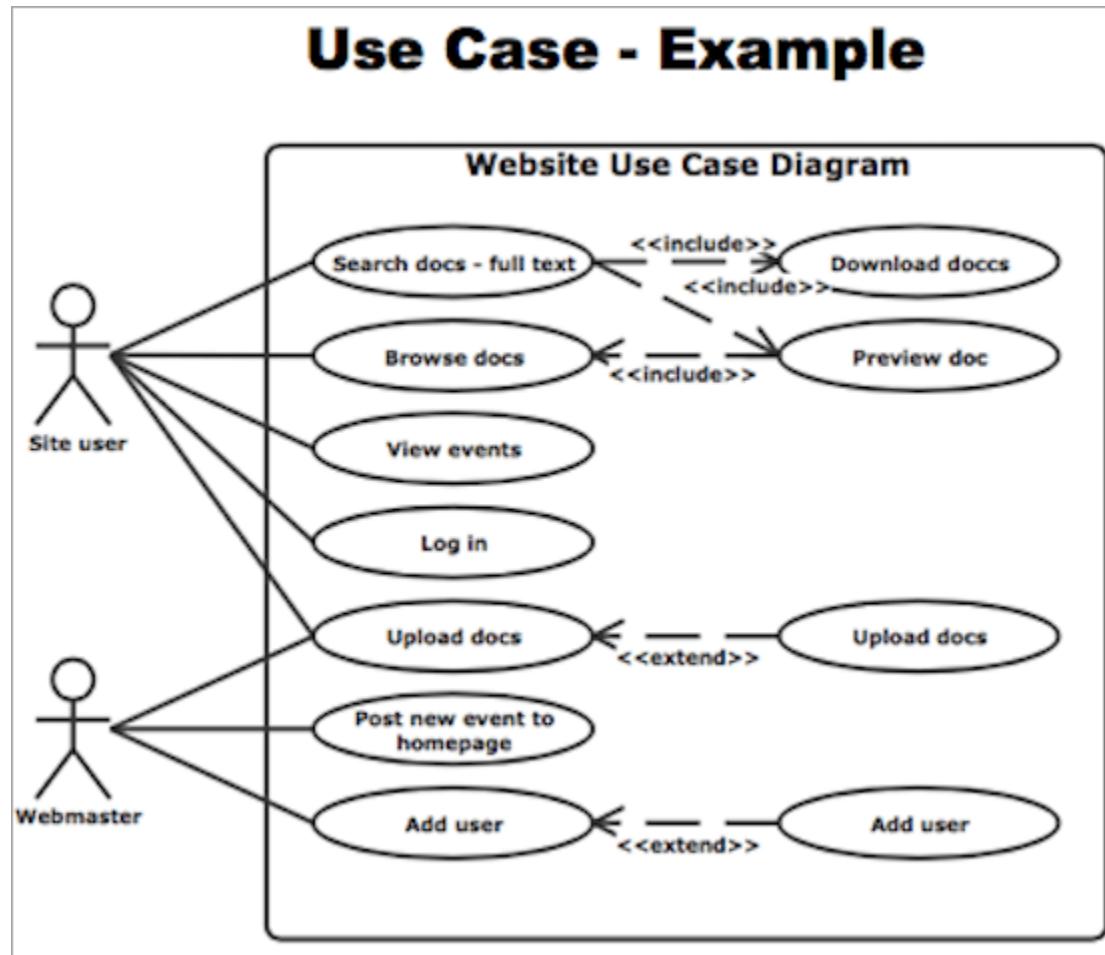
QUESTIONS



Exercise : Draw the use case diagram



Exercise : Draw the use case diagram 1



Exercise : Draw the use case diagram 2

