CS3004D Software Engineering

Usecase Diagram Contd...

Usecases -

The tasks that each actor will need to do with the system

A *use case* is a typical sequence of actions that a user performs in order to complete a given task

Usecase

- A usecase represents ***one particular way the system will be used***.
- In general, a use case should cover the ***full sequence of steps*** from the beginning of a task until the end.
- A use case should describe the *user's interaction* with the system and not the computations the system performs.
- A use case should only include actions in which the actor interacts with the computer.

From the problem definition,

- **Identify the actors**
- **Identify the usecases. Write down description**
- **Draw diagram indicating how they are related**

Eg: Library System

Actors :- Borrower, Clerk

Borrower:

- Search book
- Place book on hold
- Listing the books borrowed
- .....

Clerk

- All above usecase
- Check out an item
- Check in an item
- .....

How to describe a single use case

A. Name: Give a short, descriptive name to the use case.

B. Actors: List the actors who can perform this use case.

C. Goals: Explain what the actor or actors are trying to achieve.

D. Preconditions: State of the system before the use case.

E. Description: Give a short informal description.

F. Related use cases.

G. Steps: Describe each step using a 2-column format.

H. Postconditions: State of the system in following completion.

Check out an item for a borrower

Use case: Check out an item for a borrower

Actors: Checkout clerk (regularly), chief librarian (occasionally)

Goals: To help the borrower to borrow the item if they are allowed, and to ensure a proper record is entered of the loan.

Preconditions: The borrower must have a valid card and not owe any fines. The item must have a valid barcode and not be from the reference section.

Steps:

Actor actions	System responses
1. Scan item's barcode and barcode of the borrower's card.	
	2. Display confirmation that the loan is allowed.
3. Stamp item with the due date.	
4. Confirm that the loan is to be initiated.	
	5. Display confirmation that the loan has been recorded.

Postconditions: The system has a record of the fact that the item is borrowed, and the date it is due.

Scenario

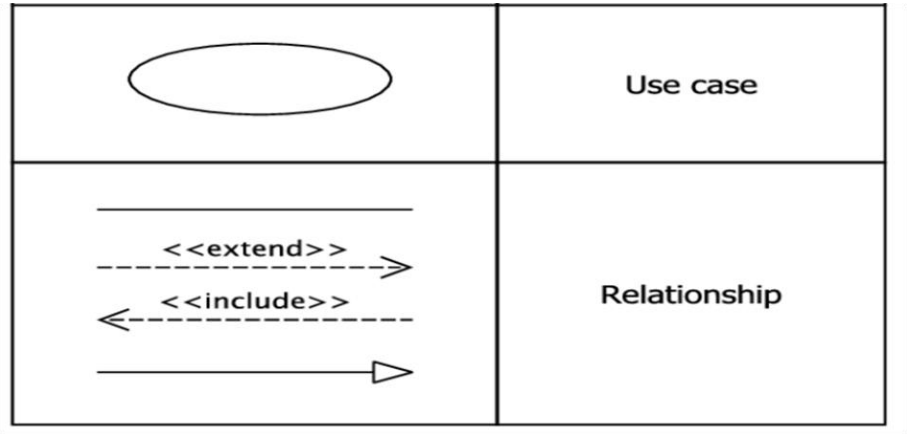
A scenario is an *instance* of a use case

- It expresses a *specific occurrence* of the use case

—a specific actor ...
—at a specific time ...
—with specific data.

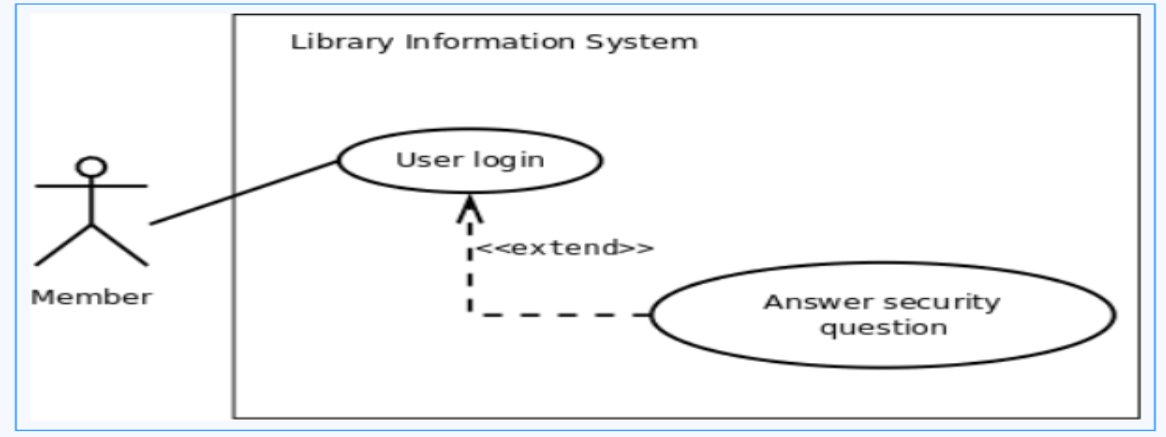
While a usecase itself might drill into a lot of detail about every possibility, a **use-case diagram can help provide a higher-level view** of the system.

Use case diagrams can be **a good communication tool** for stake holders.



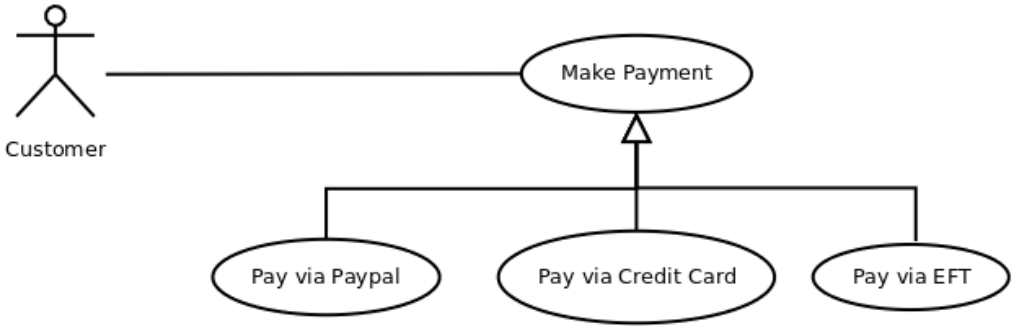
Extensions

- Used to make ***optional*** **interactions** explicit or to handle ***exceptional*** **cases.**
  - — Eg: What happens, when a user types a wrong file name.
- By creating separate use case extensions, the **description of the basic use case remains simple**.



Generalizations

- Much like superclasses in a class diagram.
- A generalized use case represents *several similar* use cases.
- One or more specializations provides details of the similar use cases.

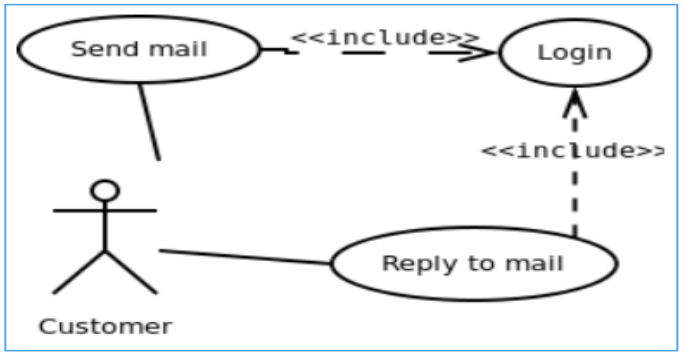


Inclusions

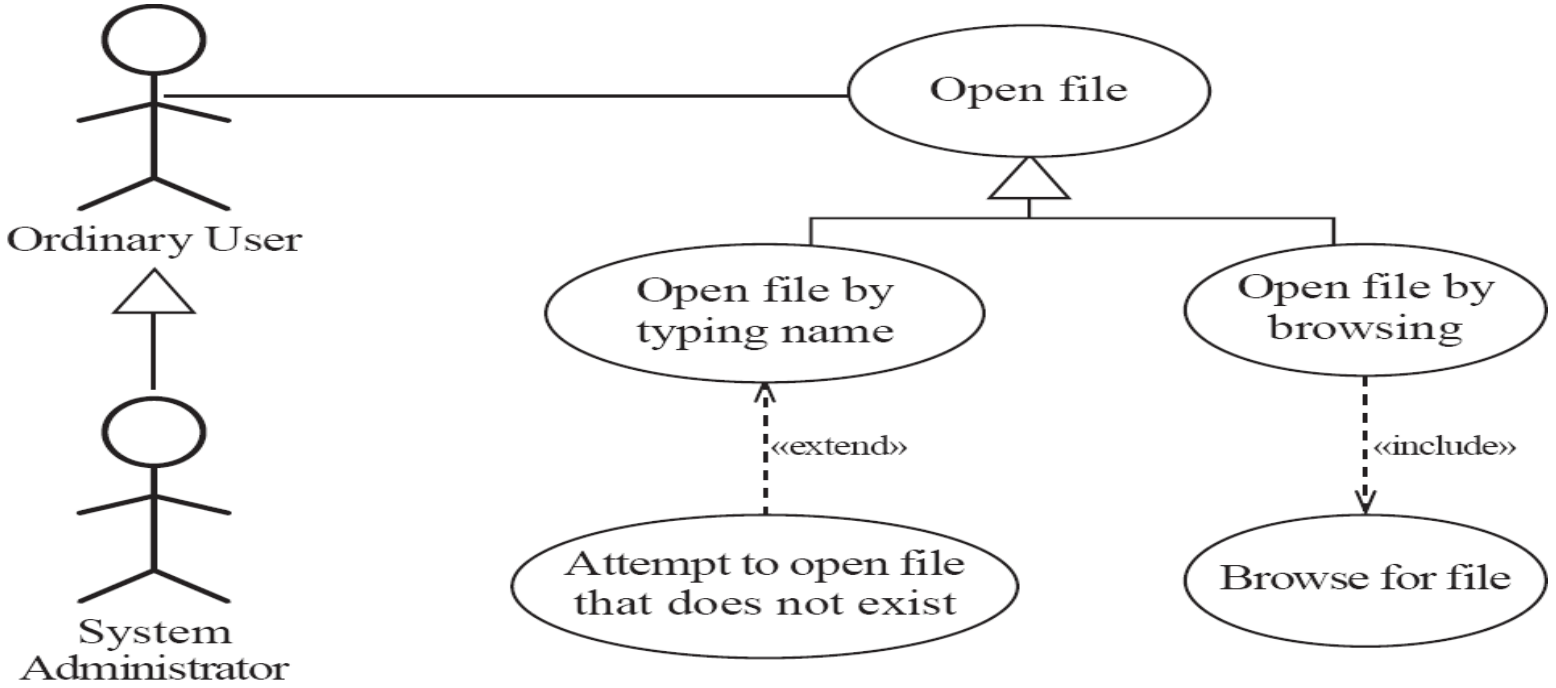
- Allow one to express *commonality* between several different use cases.

- Are included in other use cases
  - —Even very different use cases can share sequence of actions.
  - —Enable you to avoid repeating details in multiple use cases.

- Represent the performing of a *lower-level task* with a lower-level goal.



Example of generalization, extension and inclusion



Use case: Open file

Related use cases:

Generalization of:

- ❏ open file by typing name
- ❏ open file by browsing

Steps:

Actor actions	*System responses*
1. Choose ‘Open…’ command.	2. Display ‘File open’ dialog.
3. Specify filename.	
4. Confirm selection.	5. Remove dialog from display.

Use case: Open file by typing name

Related use cases:

Generalization: Open file

Steps:

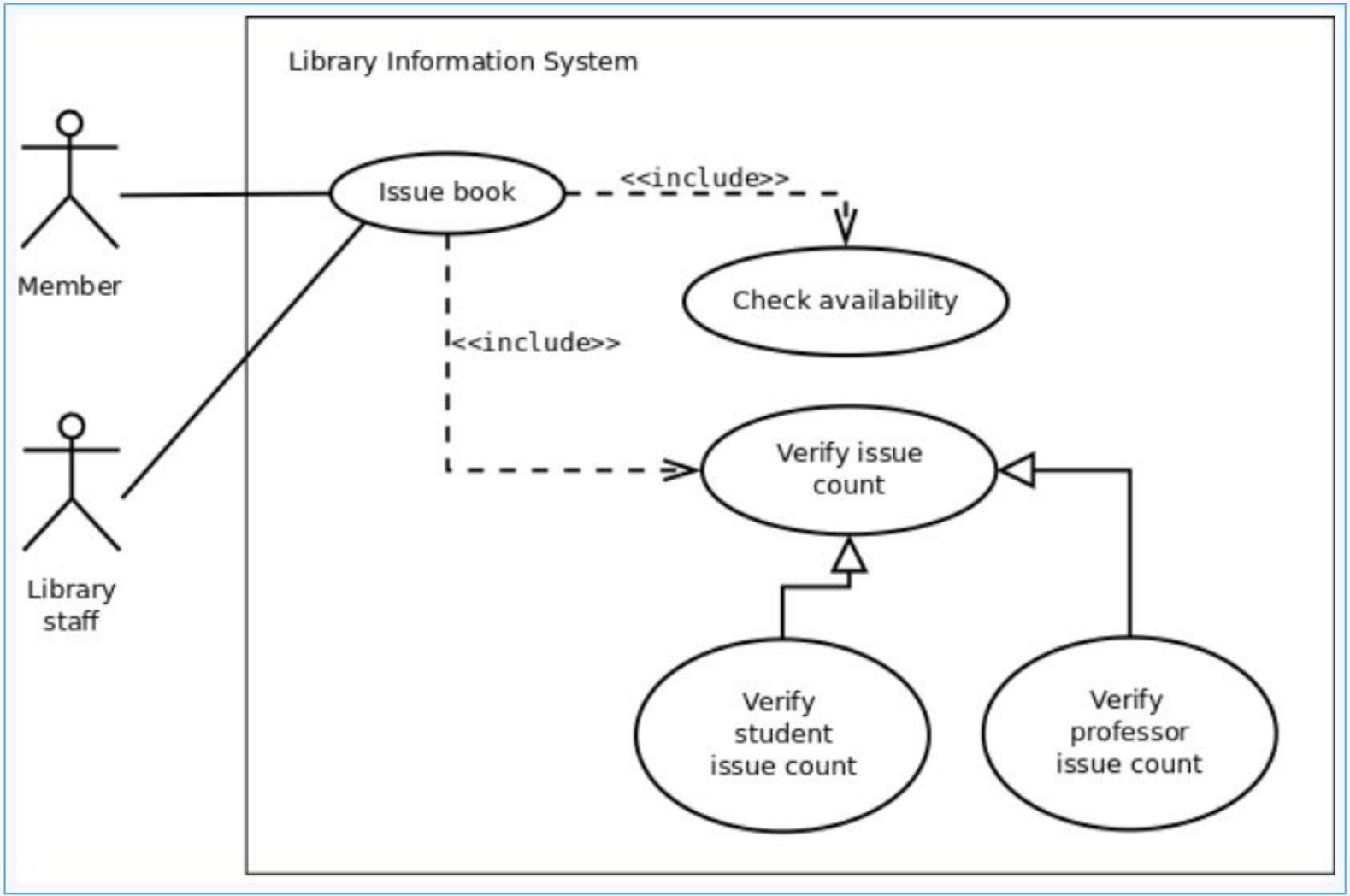
Actor actions	*System responses*
1. Choose ‘Open…’ command.	2. Display ‘File open’ dialog.
3a. Select text field.	
3b. Type file name.	
4. Click ‘Open’.	5. Remove dialog from display.

Use case: Attempt to open file that does not exist

Related use cases:

Extension of: Open file by typing name (extension point: step 4: Click 'Open')

Actor actions	*System responses*
4. Click 'Open'.	4a. Indicate that file does not exist.
4b. Correct the file name.	
4c. Click 'Open'.	5. Remove dialog from display.

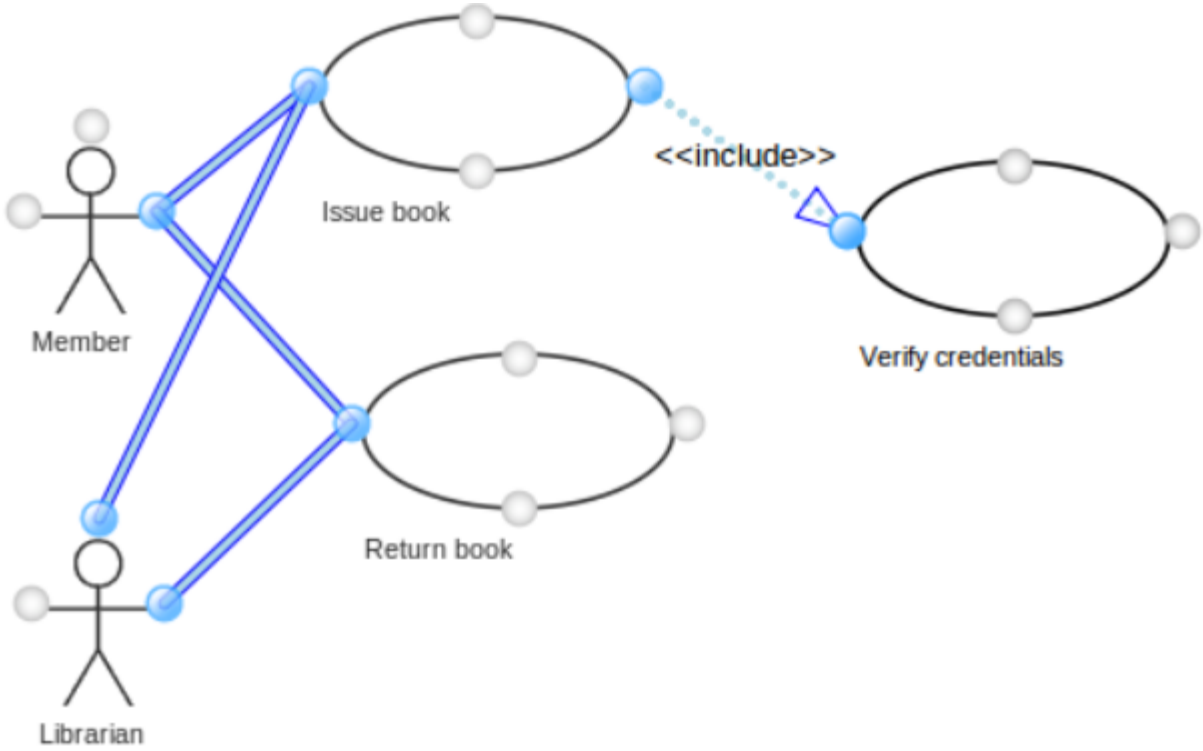

Library Information System
Member
Issue book
<<include>>
Check availability
<<include>>
Library staff
Verify issue count
Verify student issue count
Verify professor issue count

Exercise

Consider a library, where a member can perform two operations: *issue book* and *return it*. A book is issued to a member only after verifying his credentials.

Draw a use case diagram for the problem.

1. Identify the actors and use cases
2. Identify the relationships


Issue book
<<include>>
Verify credentials
Member
Return book
Librarian