SLIIT
Discover Your Future

Use Case Diagrams

Software Process Modeling

Activity from Requirements Engineering

- Visit a library where a Library System is not used. Observe their behavior.
  - Note the differences in the case study and actual system.
  - Note down the additional requirements you identified

Session Outcomes

- Introduction
- Components of a Use case diagram
  - System
  - Actors
  - Use cases
  - Relationships
- Applying use case diagrams in real world applications
- Use case scenarios

Requirements Specification

- Structured Natural Language
  - –User Stories
- Mathematical Specifications
  - –Decision Trees
  - –Decision Tables
- Graphical Notations
  - **–Use Case Diagrams and Use Case Scenarios**
  - –Activity Diagrams

Use Case Modeling

- Use cases were developed originally to support requirements elicitation and now incorporated into the UML
- Use case modeling is a mechanism for describing the functionality of a system in terms that the client can understand

What is a Use Case Diagram?

The purpose is,

- To graphically represent an overview of the functionality provided by a system.
- To demonstrate the understanding on the functional requirements to the clients.

Use Cases for Requirements Engineering

- Use Case Model captures the requirements of a system.
- Use cases act as a means of communicating with stakeholders about what the system is intended to do.
  - –It is an excellent way to communicate to management, customers, and other non-development people:
  - –WHAT a system will do when it is completed.
  - –But….it does not go into detail of HOW a system will do anything.

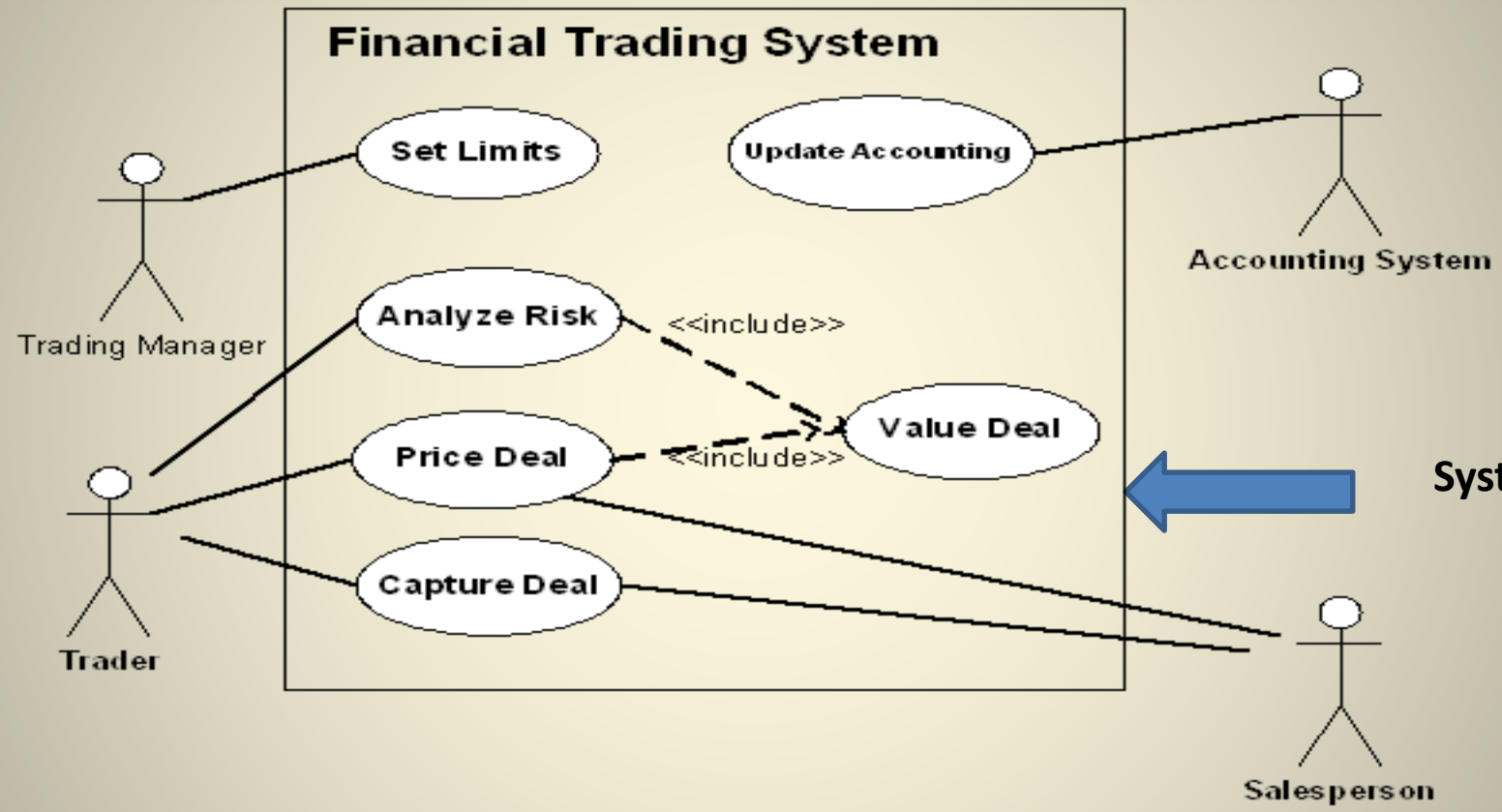
Components of a Use Case Diagram

- To construct a Use Case diagram, there are FOUR basic components.
  - **System**: something that performs function(s).
  - **Actors**: the roles adopted by those participating.
  - **Use Cases**: high level activities to be supported by the system.
  - **Relationships / Links**: which actors are involved in which use cases (dependency, generalization, and association).

1) System

- **System** is something which perform function(s).
- **System Boundary** Represents the boundary between the (physical) system and the actors who interact with the (physical) system.

System - example

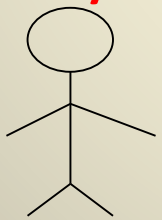


2) Actors

- A Use Case Diagram shows the interaction between the system and entities external to the system. These external entities are referred to as Actors.

- Actors represent roles which may include human users, external hardware or other systems.

- Notation →



Activity

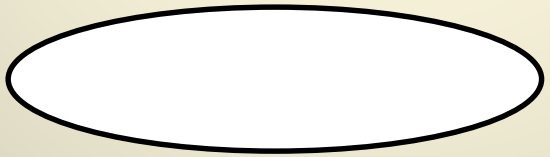
- Identify actors of the SLIIT Library system.

3) Use Case

- Use cases specify a desired behavior.

- A Use Case represents a set of sequence of events (actions) which combine to form some interaction between the software system and the outside world.

- Notation →



Use case name

How to Identify a use case?

- The best way to find use cases is to consider what each actor requires of the system.

- For each actor, human or not, ask yourself the following questions in order to figure out the relevant use cases.
  - What are the primary tasks the actor wants the system to perform?
  - Will the actor create, store, change, remove, or read data in the system?
  - Will the actor need to inform the system about sudden, external changes?
  - Does the actor need to be informed about certain occurrences in the system?
  - Will the actor perform a system start-up or shutdown?

Activity

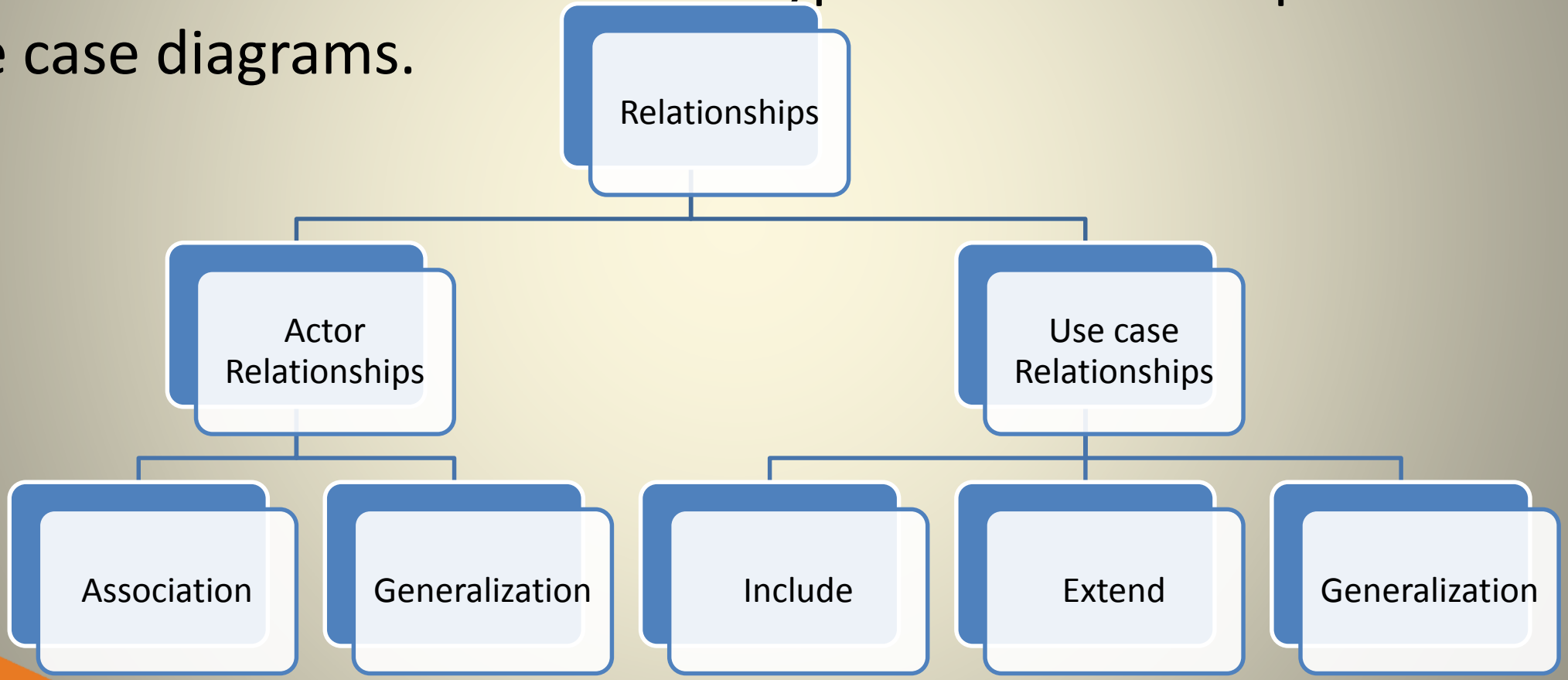
- Identify use cases for the SLIIT Library System.

Activity

- Identify the relevant actors for the use cases for the SLIIT Library System.

4) Relationships

- Below mentioned are the main types of relationships used in use case diagrams.



Actor to Use case Relationships

Association.

- indicates that an actor participates in (i.e. communicates with) the use case.

Activity

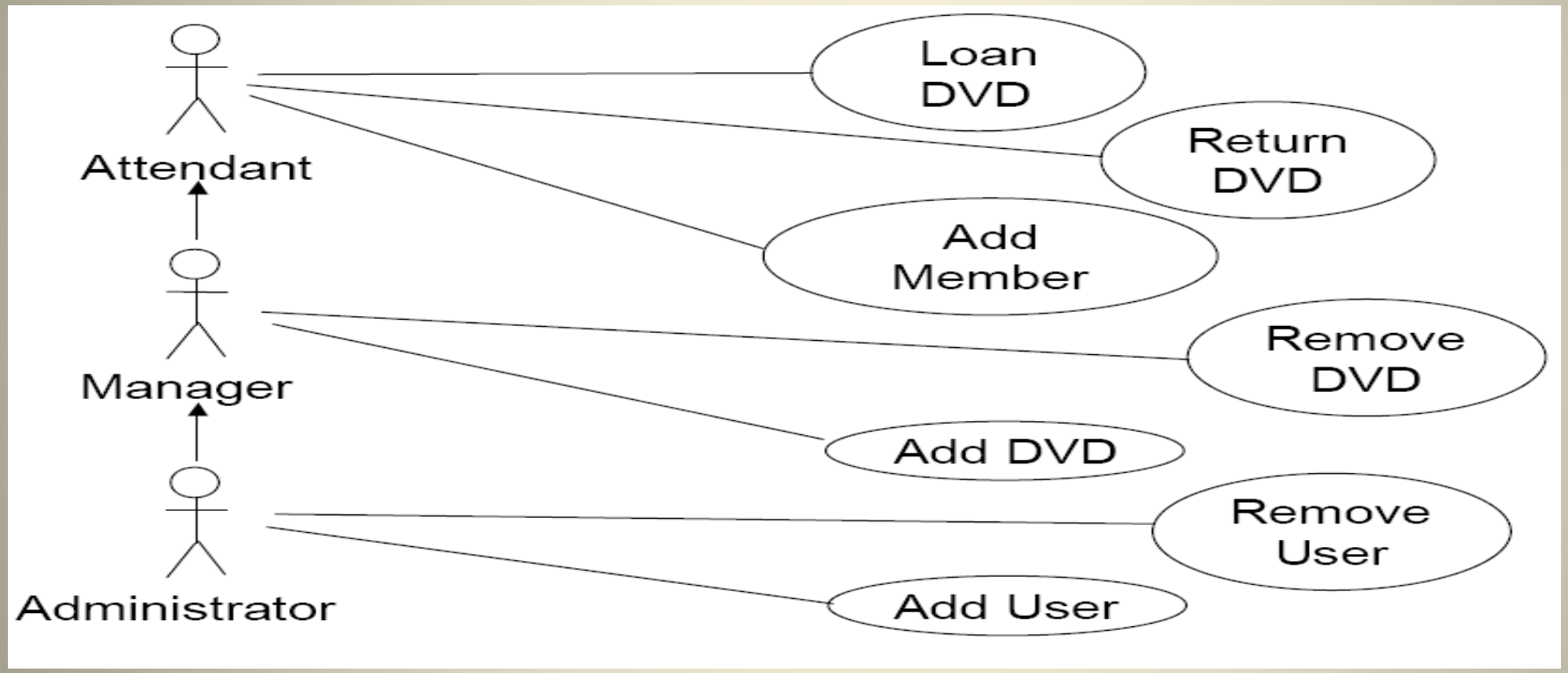
- Draw the Actors and Associations for the SLIIT Library System

Actor to Actor Relationships

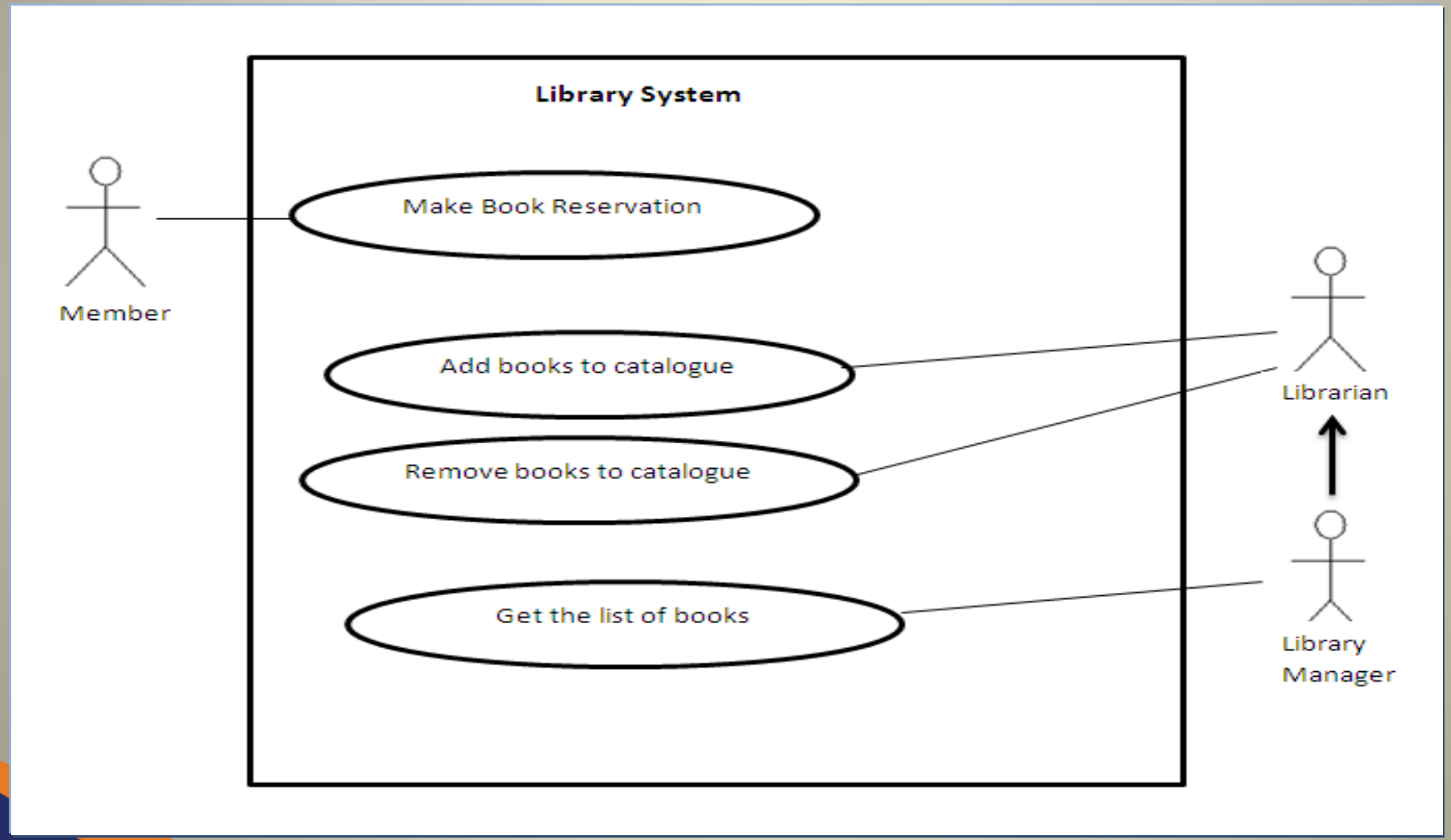
Generalization.

- Actor Generalization is drawn from the concept of inheritance in Object Oriented Programming.
- A child actor Inherits all of the characteristics and behavior of the parent actor.
- Can add , modify, or ignore any of the characteristics and behaviors of the parent actor.

Who has the most rights in the system?



Library System – Sample



Activity

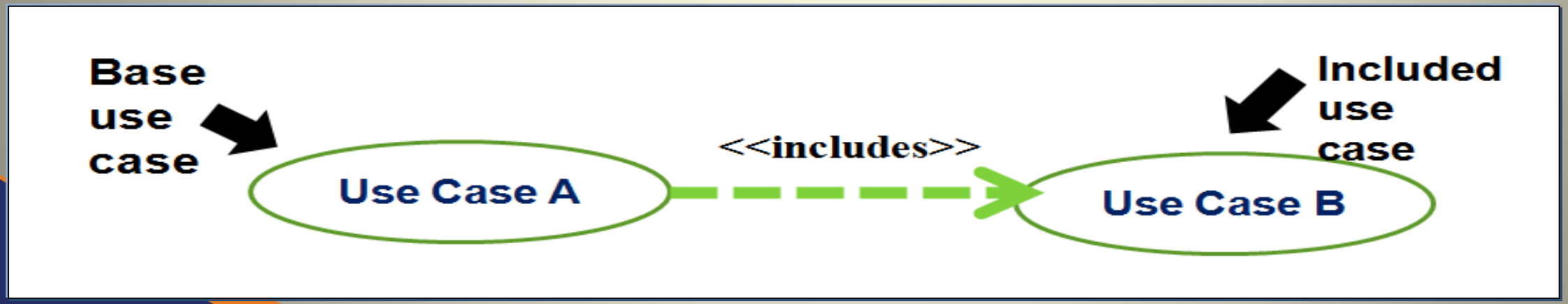
- Draw the actor to actor relationships for the SLIIT Library System
  - Actor Generalization

Use Case Relationships

1) Include

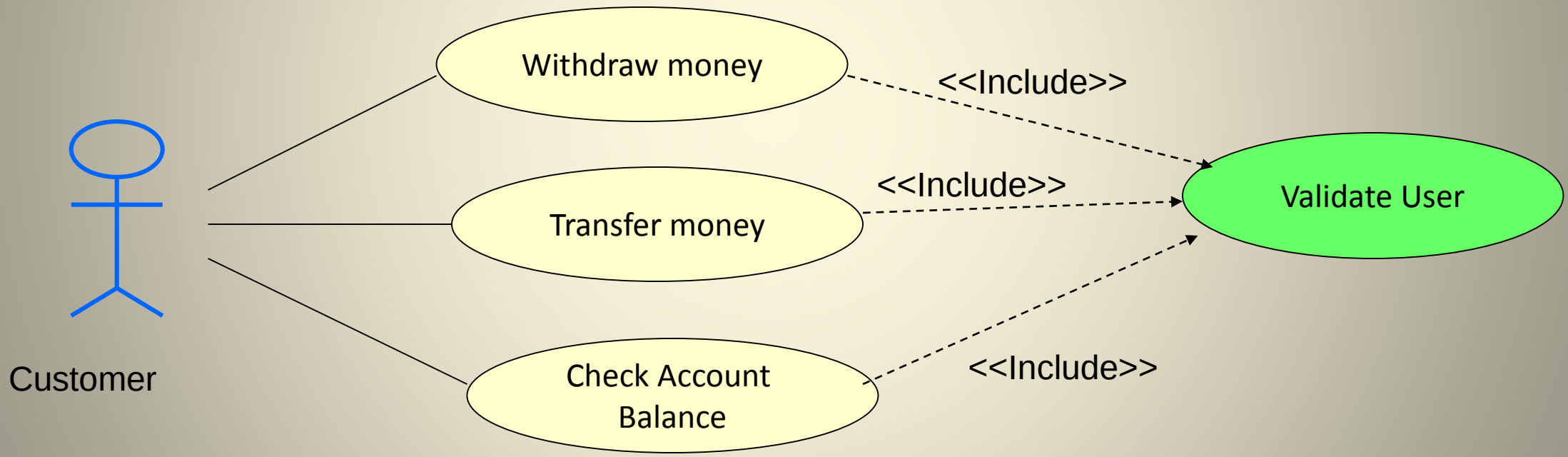
– The base use case explicitly incorporates the behavior of another use case at a location specified in the base.

– The included use case never stands alone. It only occurs as a part of some larger base that includes it.



Use Case Relationships

- Enables us to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.



Activity

Update the use case diagram of the Library System for the below given criteria.

When member is reserving the books he/she has to login to the system.

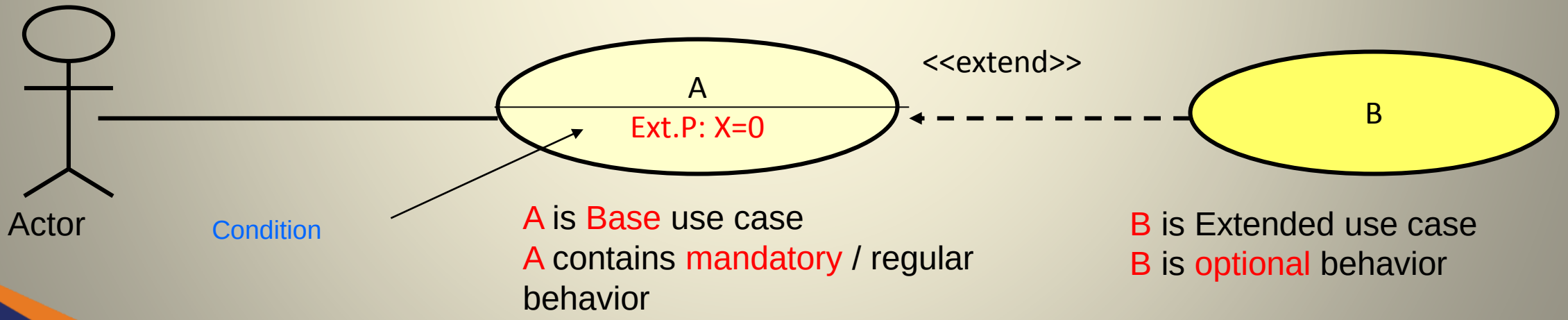
Activity

- Draw the include relationships between the use cases for the SLIIT Library System

Use Case Relationships

2) Extend

- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.



Use Case Relationships

- Eg:- When a student get enrolls in the university they perform a visa check if he/she is a foreign student.

Activity

Update the use case diagram of the Library System for the below given criteria.

Member can renew the books he/she has borrowed. When renewing if book has exceeded the loan period a fine will be calculated. For renewing purposes the member should login to the system.

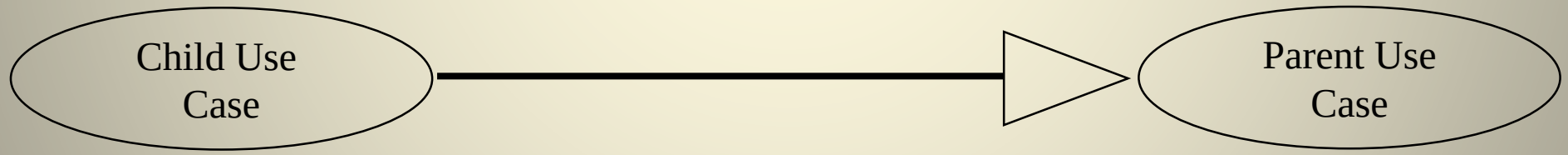
Activity

- Draw the extends relationships for the SLIIT Library System

Use Case Relationships

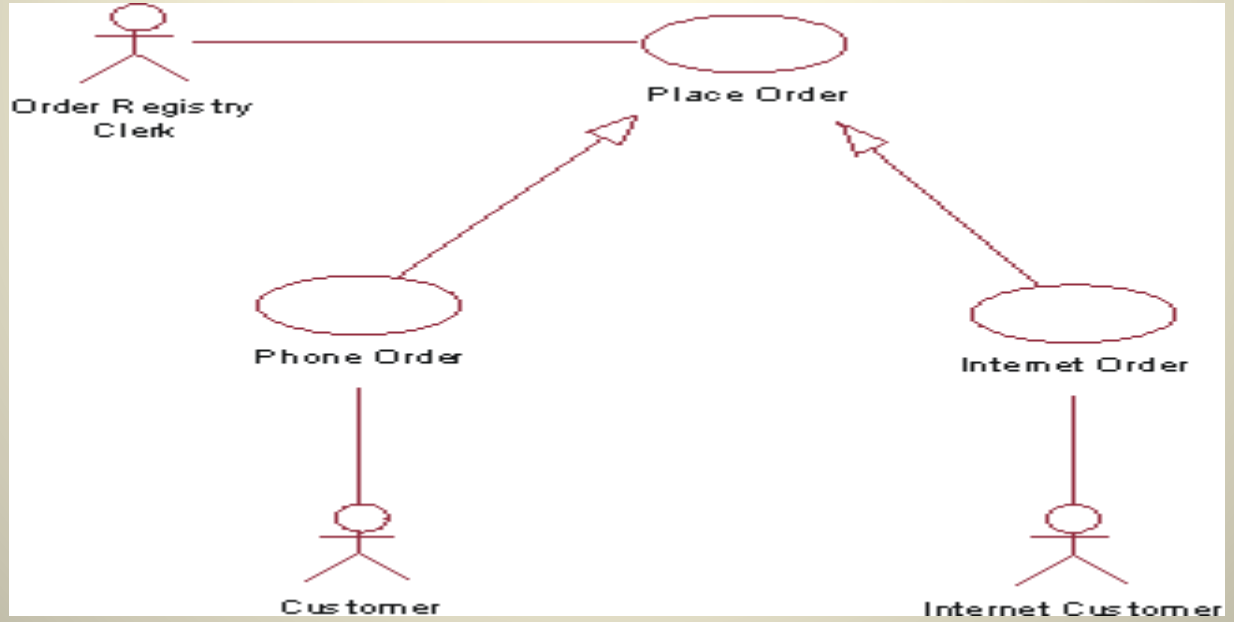
3) Generalization

- The child use case inherits the behavior and meaning of the parent use case.
- The child may add to or override the behavior of its parent.



Example

- The actor Order Registry Clerk can instantiate the general use case Place Order. Place Order can also be specialized by the use cases Phone Order or Internet Order.



Activity

Update the use case diagram of the Library System for the below given criteria.

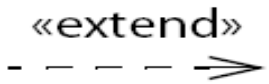
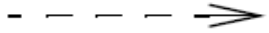
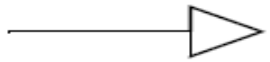
Library Manager can generate reports of the Borrowed books, Overdue books at the end of each month.

Activity

- Draw the generalization relationships for the SLIIT Library System

Relationship Summary

Table 6-1: *Kinds of Use Case Relationships*

Relationship	*Function*	*Notation*
association	The communication path between an actor and a use case that it participates in	
extend	The insertion of additional behavior into a base use case that does not know about it	
include	The insertion of additional behavior into a base use case that explicitly describes the insertion	
use case generalization	A relationship between a general use case and a more specific use case that inherits and adds features to it	

Use Case Scenarios

- A Scenario is a formal description of the flow of events that occur during the execution of a Use Case instance. It defines the specific sequence of events between the system and the external Actors.

- There is usually a Main scenario, which describes what happens when everything goes to plan. It is written under the assumption that everything is okay, no errors or problems occur, and it leads directly to the desired outcome of the use-case.

Use Case Scenarios

- Other scenarios describe what happens when variations to the Main scenario arise, often leading to different outcomes.

- So the flow of events should include:
  - How and when the use case starts and ends
  - When the use case interacts with the actors
  - What objects are exchanged
  - The basic flow and
  - Alternative flows (exceptional) of the behavior.

Use Case Sample Template

1. Use Case ID and name
2. Characteristic Information
   - » Goal in Context
   - » Scope
   - » Level
3. Pre-Conditions
4. Primary Actor
5. Main Success Scenario Steps
6. Extensions
7. Optional Information

Use Case Specification Template*

Number	*Unique use case number*	
Name	*Brief noun-verb phrase*	
Summary	*Brief summary of use case major actions*	
Priority	*1-5 (1 = lowest priority, 5 = highest priority)*	
Preconditions	*What needs to be true before use case "executes"*	
Postconditions	*What will be true after the use case successfully "executes"*	
Primary Actor(s)	*Primary actor name(s)*	
Secondary Actor(s)	*Secondary actor name(s)*	
Trigger	*The action that causes this use case to begin*	
Main Scenario	**Step**	**Action**
	Step #	*This is the "main success scenario" or "happy path."*
	…	*Description of steps in successful use case* "execution"
	…	*This should be in a "system-user-system, etc." format.*
Extensions	**Step**	**Branching Action**
	Step #	*Alternative paths that the use case may take*
Open Issues	*Issue #*	*Issues regarding the use case that need resolution*

*Adapted from A. Cockburn, "Basic Use Case Template"

Use Case Specification Template Example

Number	1
Name	Withdraw Money
Summary	User withdraws money from one of his/her accounts
Priority	5
Preconditions	User has logged into ATM
Postconditions	User has withdrawn money and received a receipt
Primary Actor(s)	Bank Customer

Continued …

Trigger	User has chosen to withdraw money	
Main Scenario	**Step**	**Action**
	1	System displays account types
	2	User chooses account type
	3	System asks for amount to withdraw
	4	User enters amount
	5	System debits user's account and dispenses money
	6	User removes money
	7	System prints and dispenses receipt
	8	User removes receipt
	9	System displays closing message and dispenses user's ATM card
	11	User removes card
	10	System displays welcome message
Extensions	**Step**	**Branching Action**
	5a	System notifies user that account funds are insufficient
	5b	System gives current account balance
	5c	System exits option
Open Issues	1	Should the system ask if the user wants to see the balance?

Activity

- Write a Use Case Scenario for “Borrowing a Book”

You could consider the process given below as the manual system procedure.

The member identifies him or herself to the librarian and indicates which books they wish to borrow.

If it is acceptable for them to borrow these books, i.e. they are not marked “for reference only”, or the number of books on loan to the customer is less than some predetermined maximum, then the books are loaned to the customer for a specified loan period.

The members loan record is updated to reflect the loaned books. The libraries card index system is updated to show who has borrowed the books.

References

- *Writing Effective Use Cases*
  – *By Dr. Alistair Cockburn*

- UML 2 Bible