



Use Case Diagram



UML SIG

Session Objectives

At the end of this lecture You will be able to :

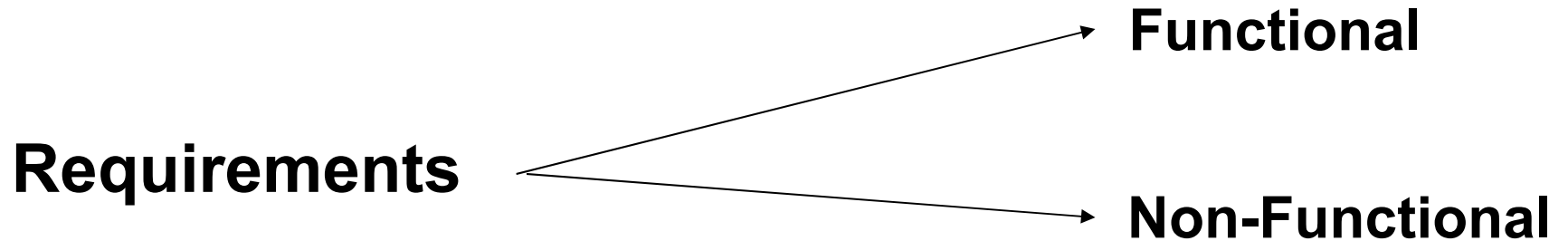
- 3. Draw a use case diagram to depict functional requirements of a system.**
- 5. Expose typical mistakes by students.**
- 6. Write use case description for a use case.**

Keywords used in this lecture

- **use case**
- **use case diagram**
- **functional requirements**
- **non-functional requirements**
- **actor**
- **include**
- **extend**
- **use case specification/definition/description**
- **flow of events**
- **alternate flow of events / alternate pathway**
- **exception flow of events / exception pathway**
- **basic flow of events / happy pathway**
- **non-functional requirements / shadow use cases**

UML SIG

Functional vs. Non-Functional



Functional requirement are user ‘visible’ features and are typically initiated by stakeholders of the system – generate report, login, etc.

Non-functional requirements are ‘non-visible’ features and but required for a effective running of an application – security, backup, etc.

Session Objectives

At the end of this lecture You will be able to :

- 3. Draw a use case diagram to depict functional requirements of a system.**
5. Expose typical mistakes by students.
6. Write use case description for a use case.

Use Case Diagram

Definition:

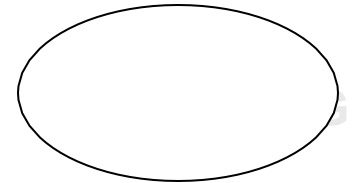
A diagram that shows a set of use cases and actors and their relationships.

Use cases represent system functionality, the requirements of the system from the user's perspective.

Notations

use case

A description of a set of sequences of actions, including variants, that system performs that yields an observable value to an actor.



actor

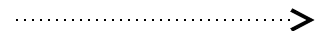
The people or systems that provide or receive information from the system; they are among the stakeholders of a system.



include

Specifies that the source use case explicitly incorporates the behaviour of another use case at a location specified by the source

<<include>>



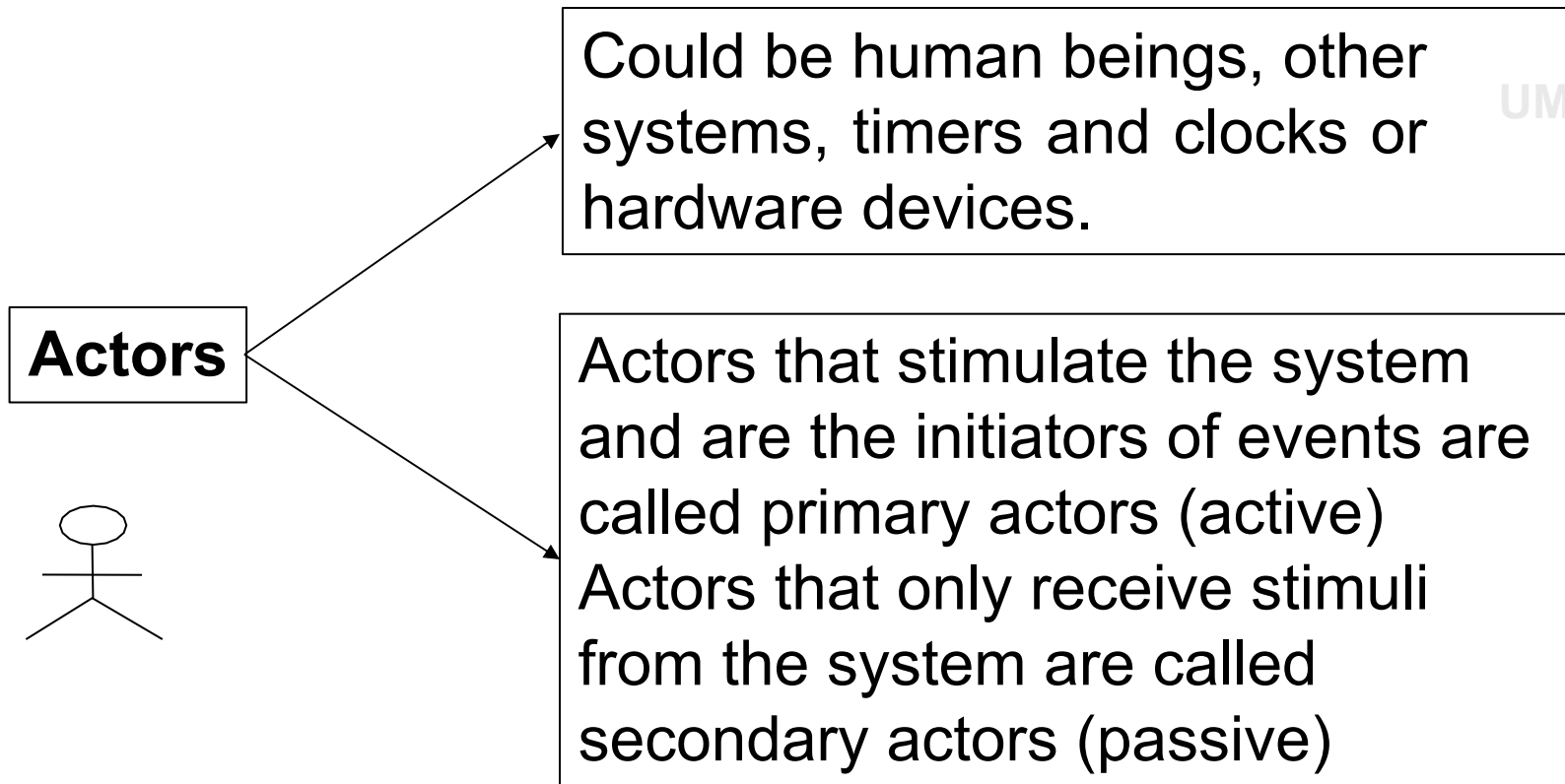
extend

Specifies that the target use case extends the behaviour of the source.

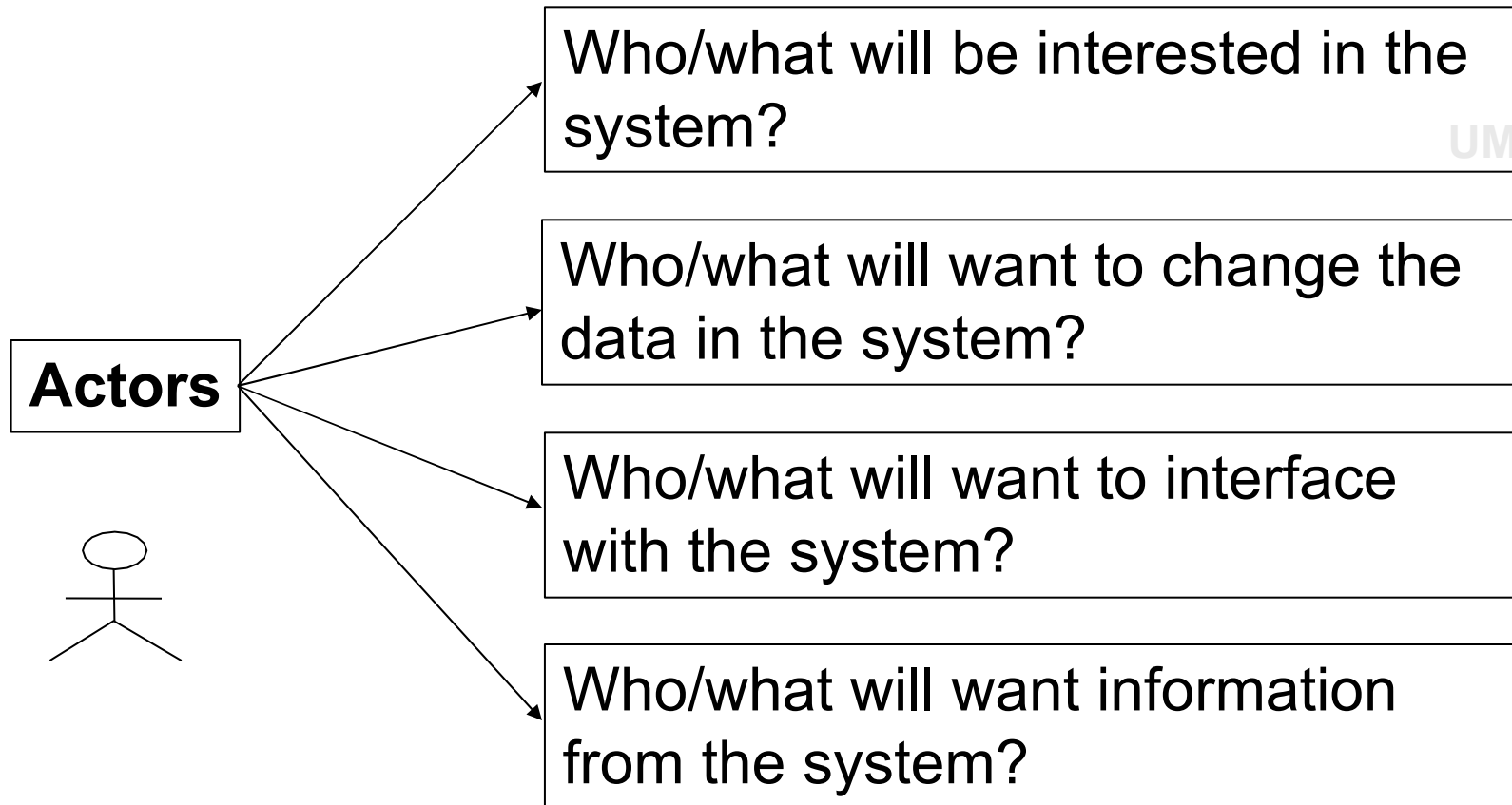
<<extend>>



Actors



Actors

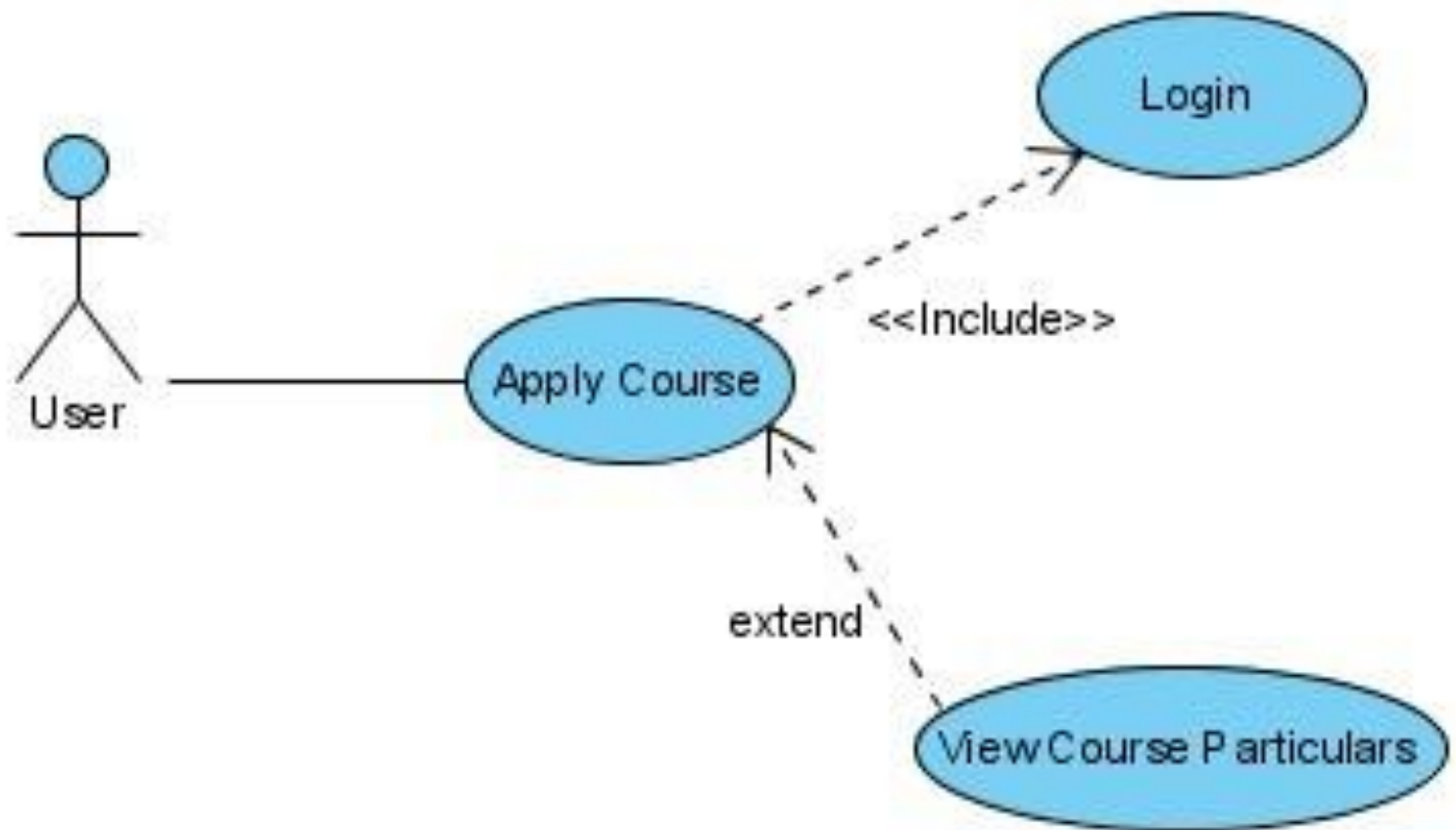


Use Case Diagram

Guidelines & Caution

1. Use cases should ideally begin with a verb – i.e generate report. Use cases should NOT be open ended – i.e Register (instead should be named as Register New User)
2. Avoid showing communication between actors.
 1. Actors should be named as singular. i.e student and NOT students. NO names should be used – i.e John, Sam, etc.
4. Do NOT show behaviour in a use case diagram; instead only depict only system functionality.
5. Use case diagram does not show sequence – unlike DFDs.

Example – Include and Extend

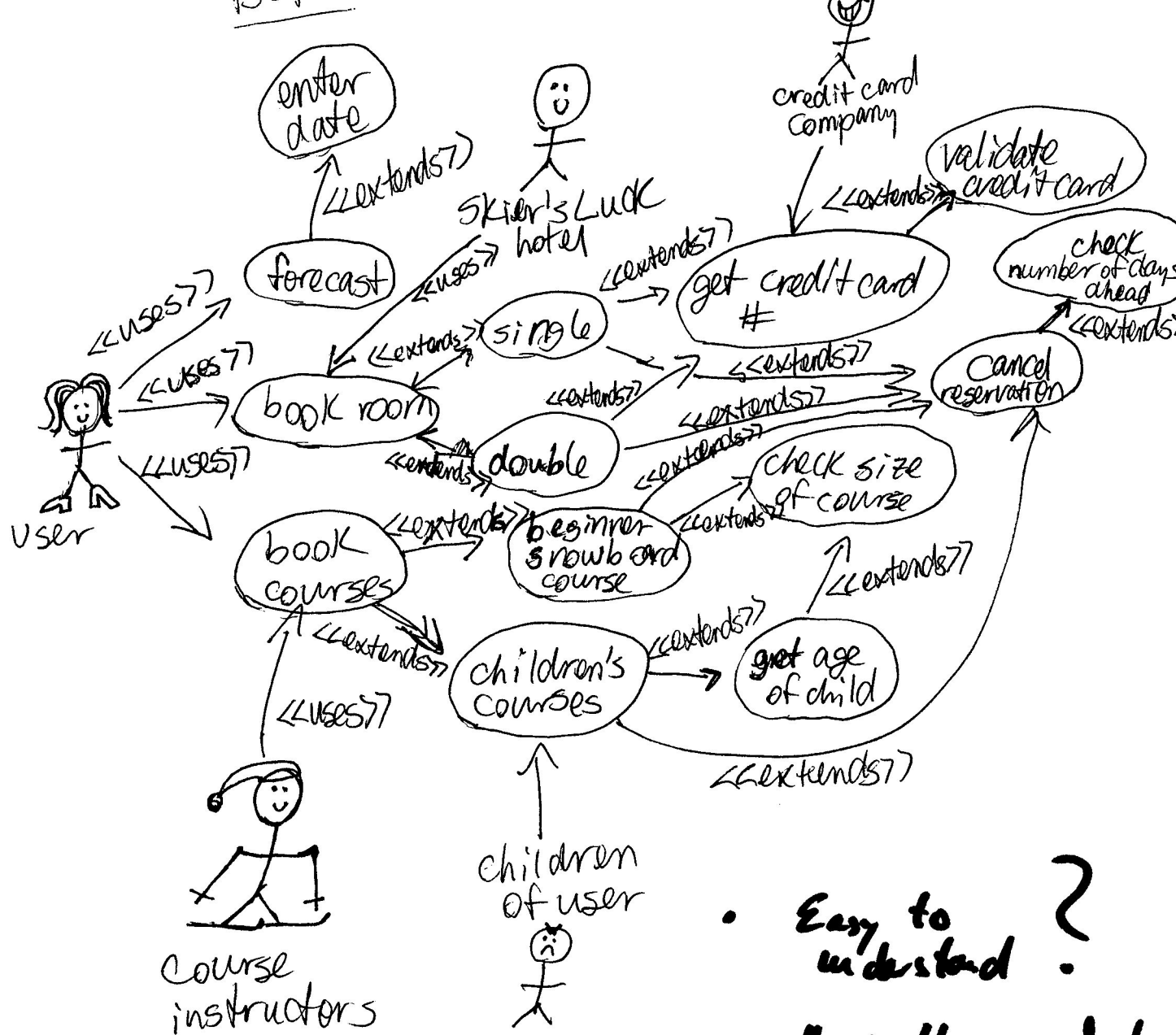


Include and Extend

The use of include and extend is discouraged simply because they add unnecessary complexity to a Use Case diagram.

Since the primary purpose of use cases is to show user centred functionality, the precedence of use cases takes little importance.

Mistake 1



- Easy to understand ?
- More than asked

Discussion Exercise

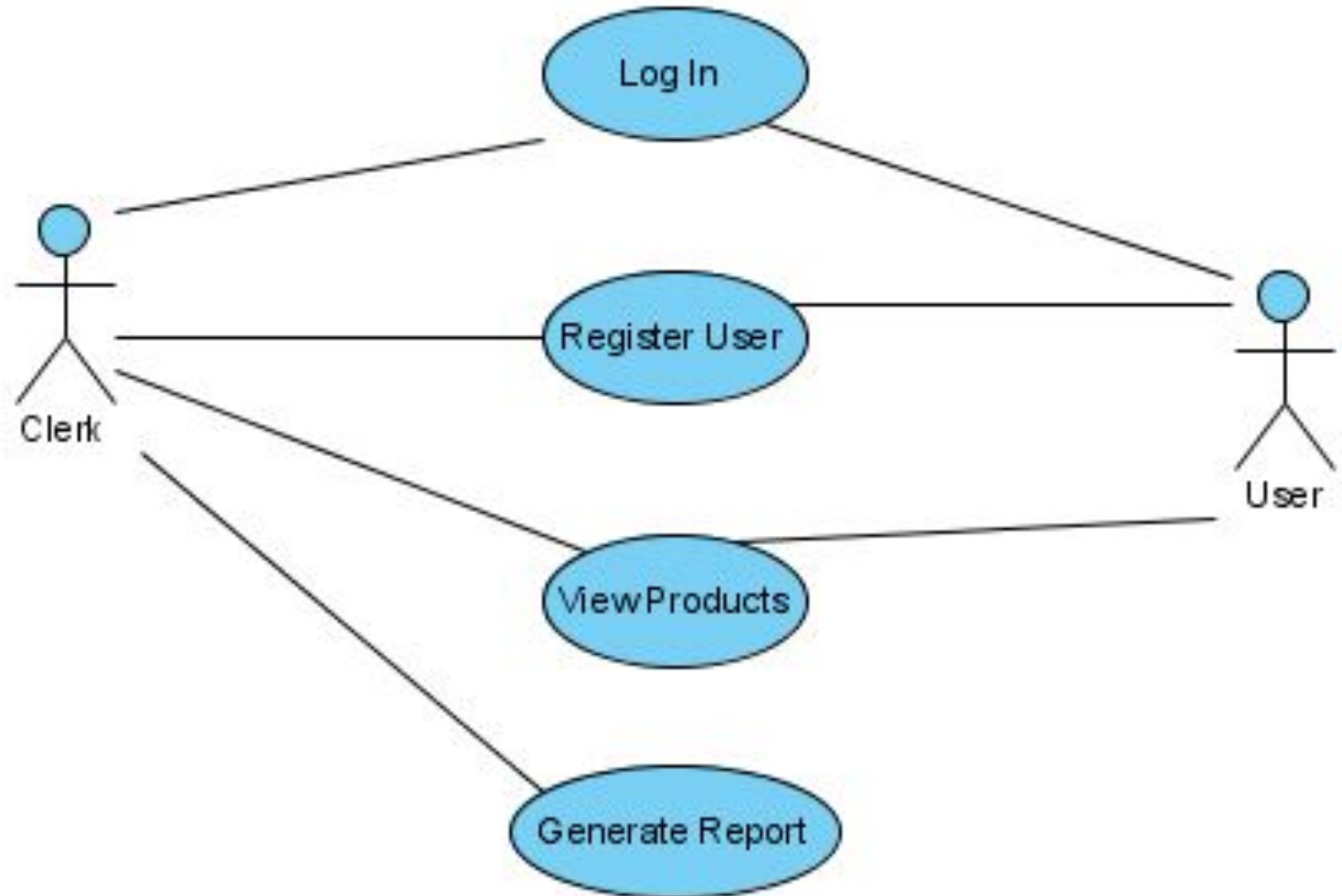
Draw a use case diagram for the scenario below:

Inventory System

UML SIG

In order to generate an invoice a clerk must log in. If a clerk is a first time user, one must have themselves registered. There should be an option for a user to register oneself within the login page. Any user can use the system to view products online. The option of login is also provided when a user views products online.

Exercise - Solution

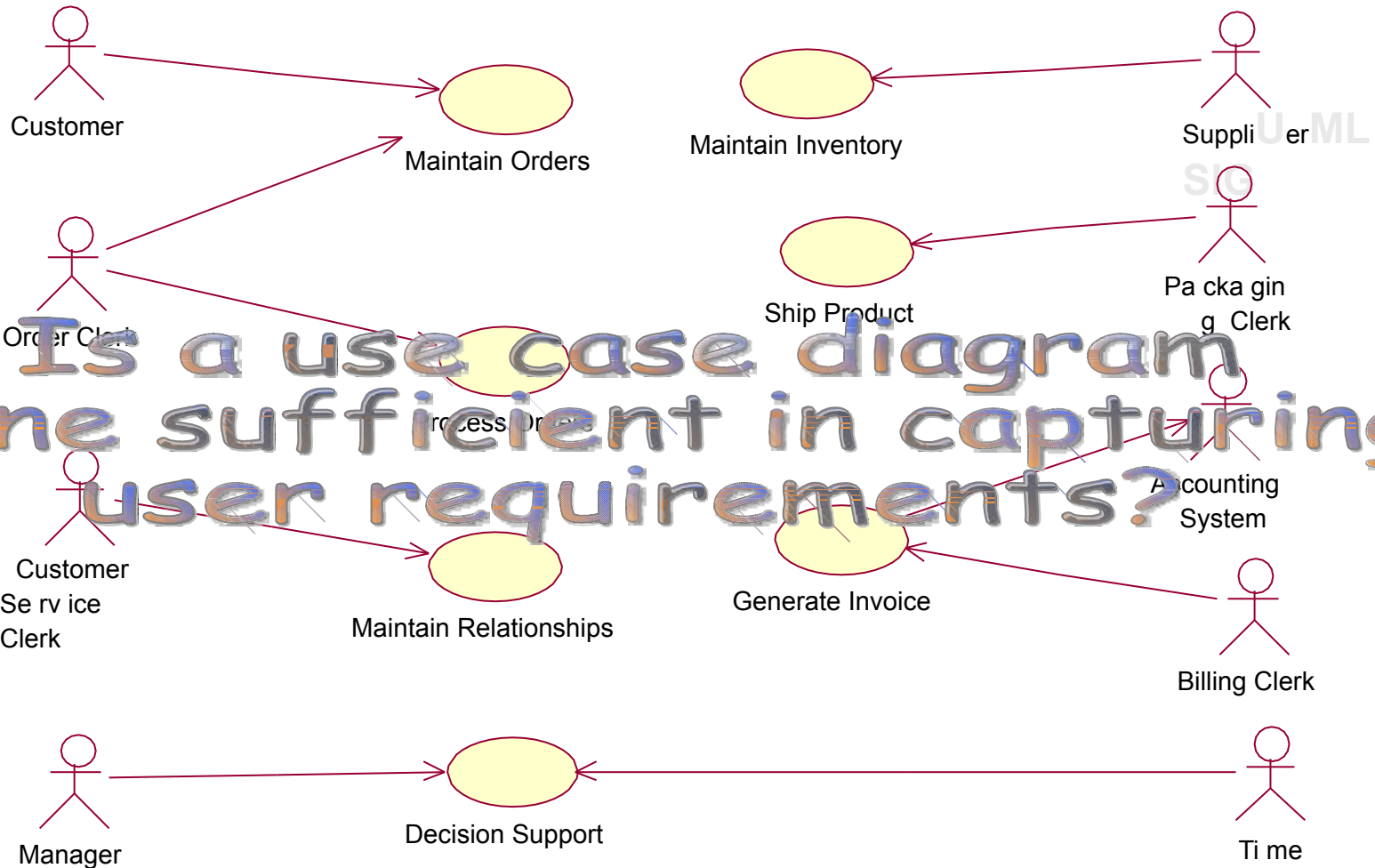


Session Objectives

At the end of this lecture You will be able to :

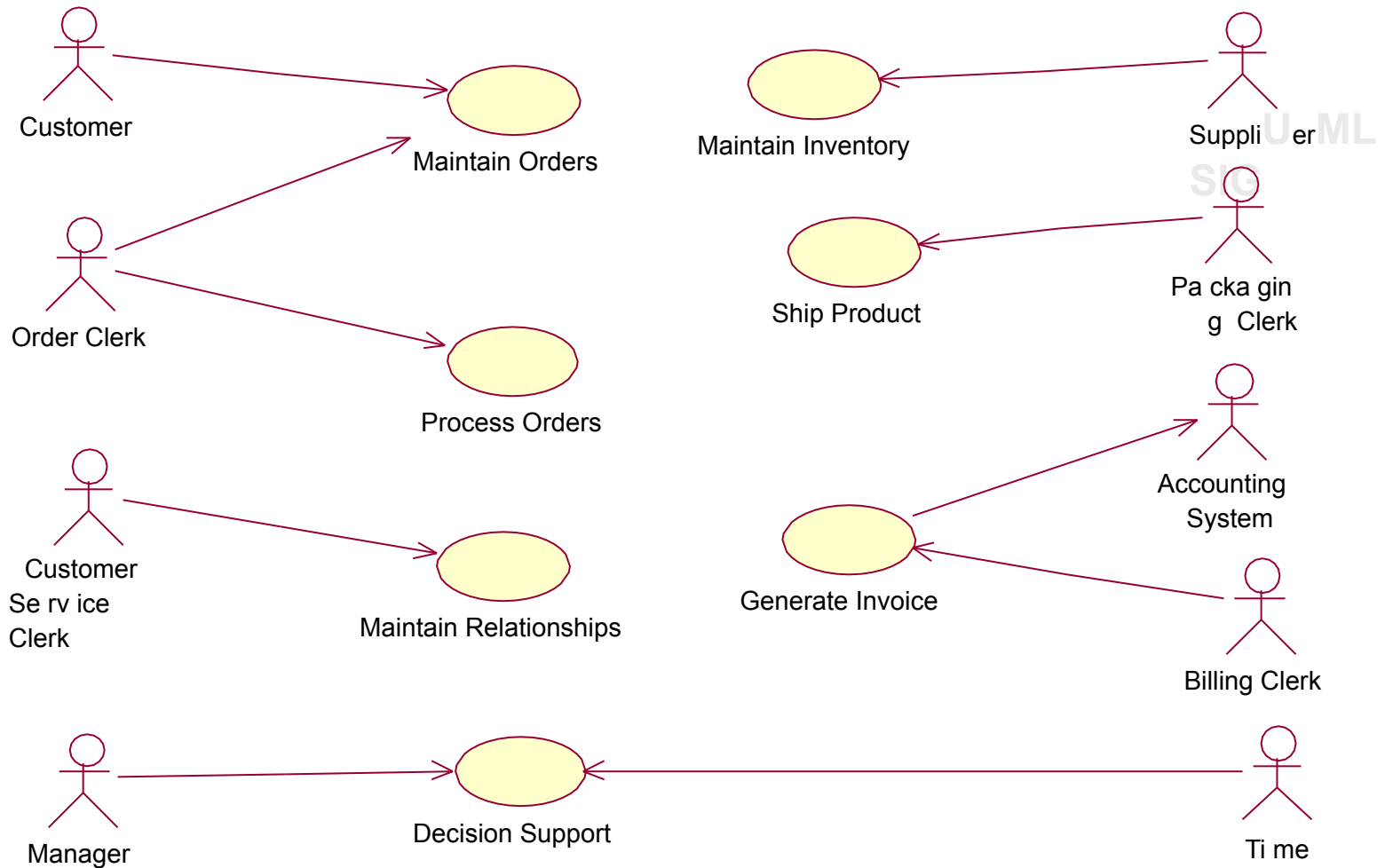
- 3. Draw a use case diagram to depict functional requirements of a system.
- Expose typical mistakes by students.
- **Write use case description for a use case.**

Use Case Specification



Is a use case diagram alone sufficient in capturing user requirements?

Use Case Specification



Use Case Specification

Use case specification is synonymous to use case description and use case definition and can be used interchangeably.

UML SIG

Use case specification defines information that pertains to a particular use case which is important in understanding the purpose behind the use case.

Use case specification is written for every use case.

A use case specification has one or more flow of events or pathways associated with it.

Flow of Events / Pathways

A flow of events or pathway is a textual description embodying sequence of events with regards to the use case and is part of the use case specification.

Flow of events is understood by the customer. A detailed description is necessary so that one can better understand the complexity that might be involved in realising the use cases.

Flow of Events / Pathways

Flow of events describes how and when the use case starts and ends, when the use case interacts with the actors, and the information exchanged between an actor and the use case.

Flow of events is derived from a what perspective, NOT how perspective. Hence, specific information like: interface details and technical specifications should NOT be included in a use case description.

Use case description serves as a 'bridge' between stakeholders of a system and the development team.

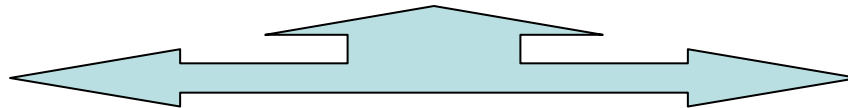
Flow of Events / Pathways

Use case description serves as a 'bridge' between stakeholders of a system and the development team.

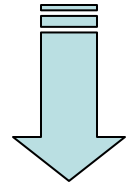
Systems analyst produce use case diagram & use case specification in consultation with end users



Use Case Diagram



Use Case Specification



Programmers look at use case specification to understand complete requirements - SRS

Flow of Events / Pathways

Flow of Events

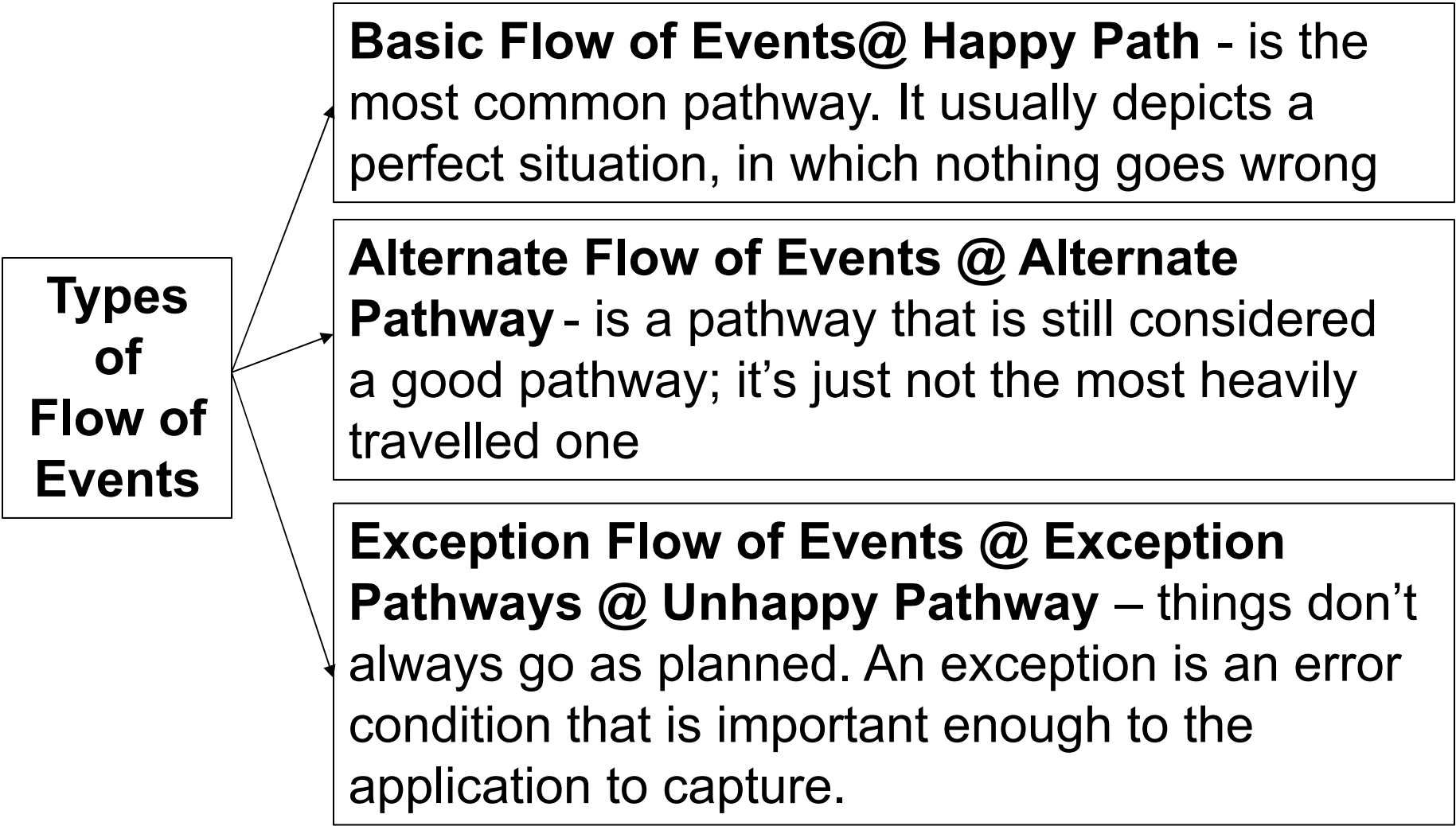
Describes how and when use case starts and ends.

Does NOT describe user interface details.

Is generally a text-based file that is included under its use case in Rational Rose/Visual Paradigm

Types Flow of Events / Pathways

Types of Flow of Events



```
graph LR; A[Types of Flow of Events] --> B[Basic Flow of Events @ Happy Path]; A --> C[Alternate Flow of Events @ Alternate Pathway]; A --> D[Exception Flow of Events @ Exception Pathways @ Unhappy Pathway];
```

Basic Flow of Events @ Happy Path - is the most common pathway. It usually depicts a perfect situation, in which nothing goes wrong

Alternate Flow of Events @ Alternate Pathway - is a pathway that is still considered a good pathway; it's just not the most heavily travelled one

Exception Flow of Events @ Exception Pathways @ Unhappy Pathway – things don't always go as planned. An exception is an error condition that is important enough to the application to capture.

Types Flow of Events / Pathways

Basic Flow of Events@ Happy Path – You get to the ATM and successfully withdraw money

UML SIG

Alternate Flow of Events @ Alternate Pathway - You get to the ATM but could not withdraw money due to insufficient funds in your account.

Exception Flow of Events @ Exception Pathways @ Unhappy Pathway – You get to the ATM machine but your valid pin number is not accepted.