

Lecture 2

Use Case Diagrams – Part 1

Objectives

At the end of this session, the student should be:

- Able to identify and use the various symbols used in use case diagrams
- Be able to draw use case diagrams, based on written scenarios

Why are Models Important?

- A model is a simplification of reality
- We build models so that we can better understand the system we are developing
 - Visualise the system as it is or will be
 - Specify the structure or behaviour of system
 - A guide for using when constructing a software system
 - Document the design decisions made

Use Case Diagrams

What is a Use Case Diagram?

- Shows the relationship between an actor and a use case



Use Case Diagrams

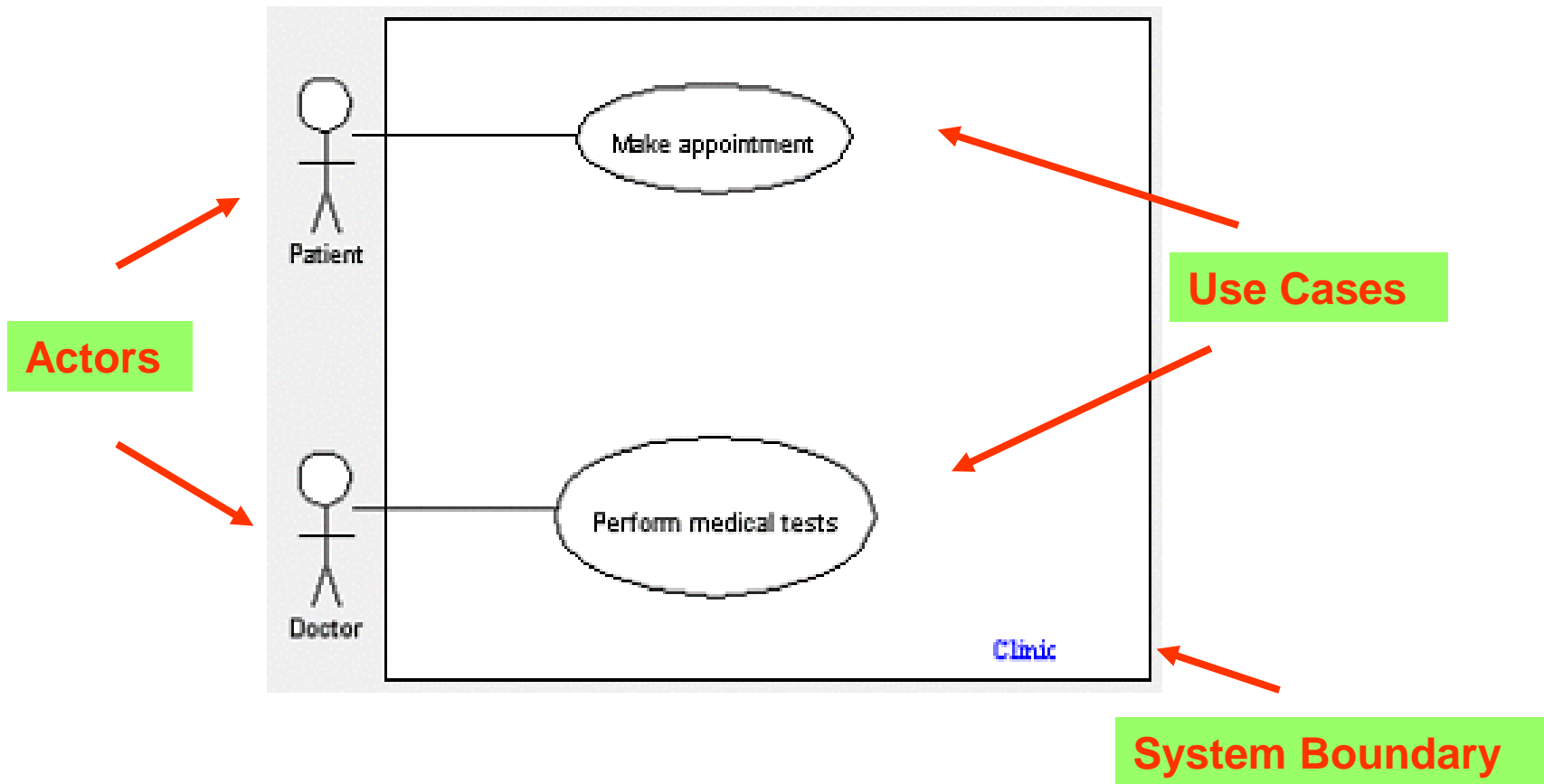
- **Actors** are represented by stick figures
- **Use Cases** are represented by ovals
- Create **scenarios**



What do they represent?

- An **actor** represents
 - a **user** or another system that will interact with a use case
- A **use case** represents
 - some **task** the system might perform with the actor
- Used to model how a system behaves

Use Case Example



Use Case Diagram for a Clinic

What do Actors represent?

- Anything outside the system that interacts with the system



- **The User**

- human --- machine --- another system

Actor 1

- User can actively interact with the system (e.g. via user interface) or
- User can passively receive information
- A single user may act as several actors

More on Use Cases

- Represent some function / task that a system does for a particular user
- Represent a complete sequence of actions / complete unit of functionality
- Must be started by an actor



Log in

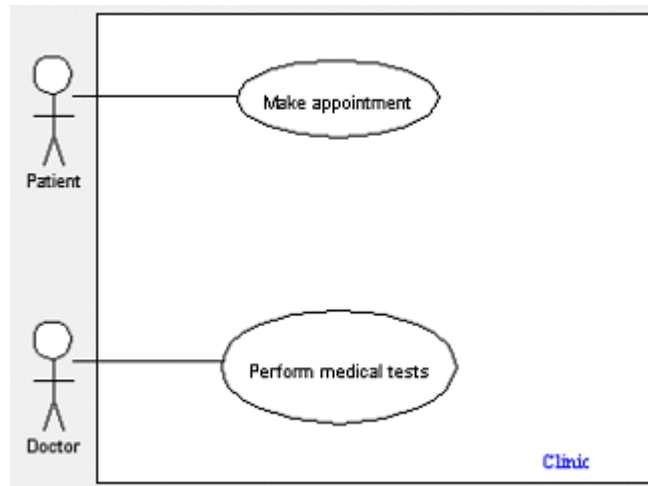
Search

Book Flight

Add to Shopping
Cart

System Boundary

- Defines the **scope / limit** of what a system will be.
- Shown as a rectangle spanning all the use cases in the system.



System boundary of a clinic application

Question Time

- What are the 3 elements in a Use Case diagram?

- -----

- -----

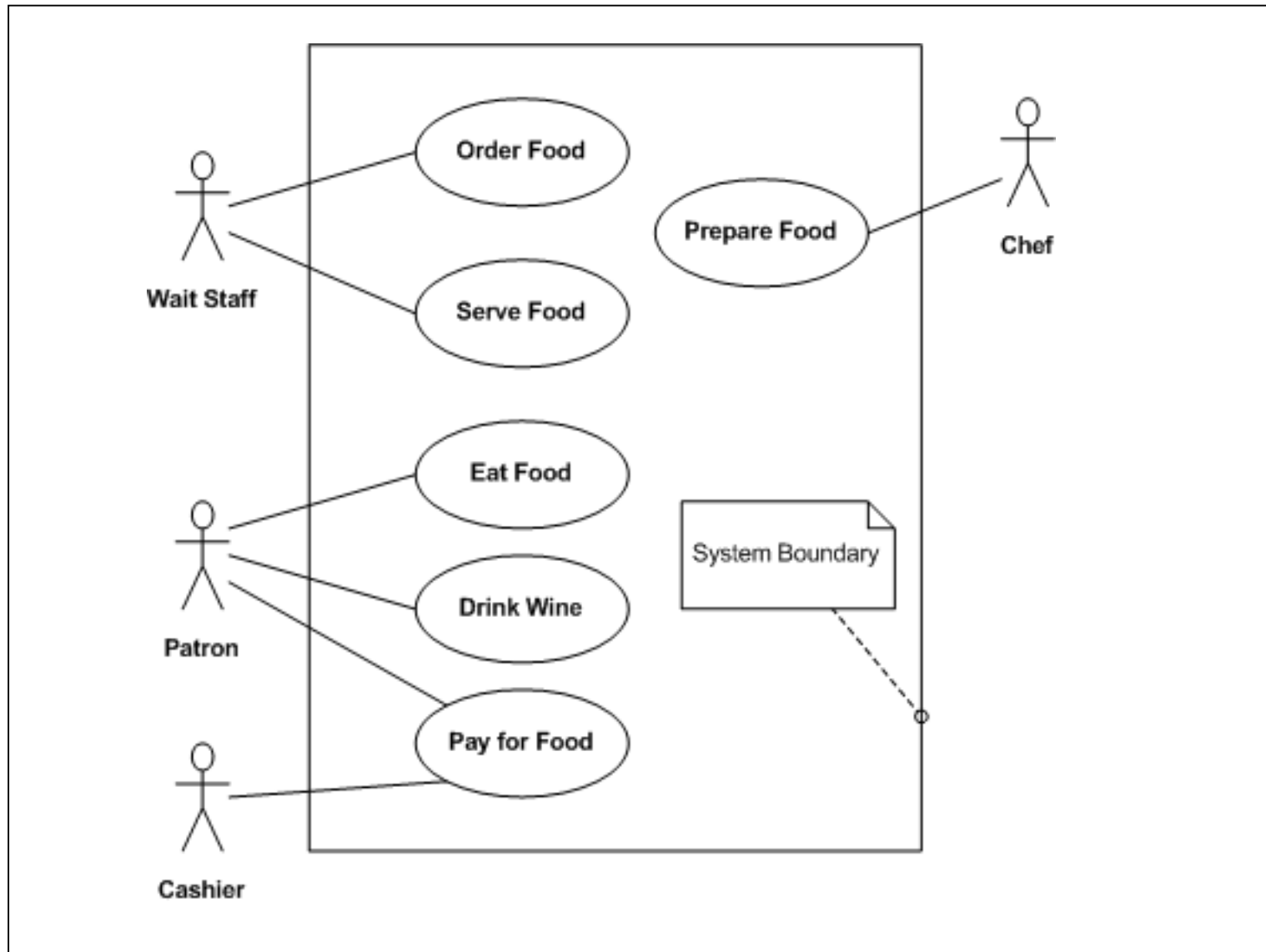
- -----



How do you find out what the system should do?

- System must meet needs of the users
- Need to analyze the system from the users point of view
- Read the system specifications
- Interview users, survey users
- Look at relevant documentation etc.
- Observe people working
- Identify:
 - Users of the system (*actor*) and other systems it interacts with
 - Tasks undertaken by the system (*use case*) with the actors

Use Case for Restaurant System



Use Case for Restaurant System

- Patron actor can Eat Food, Pay for Food, or Drink Wine.
- Chef actor can Prepare Food.
- Both the Patron and the Cashier involved in the Pay for Food use case.
- Box defines the boundaries of the Restaurant System i.e. the use cases shown are part of the system being modelled, the actors are not.
- Don't show the interaction among actors on a use case diagram!!!

Sharpen your pencil



Draw the following use case diagram.

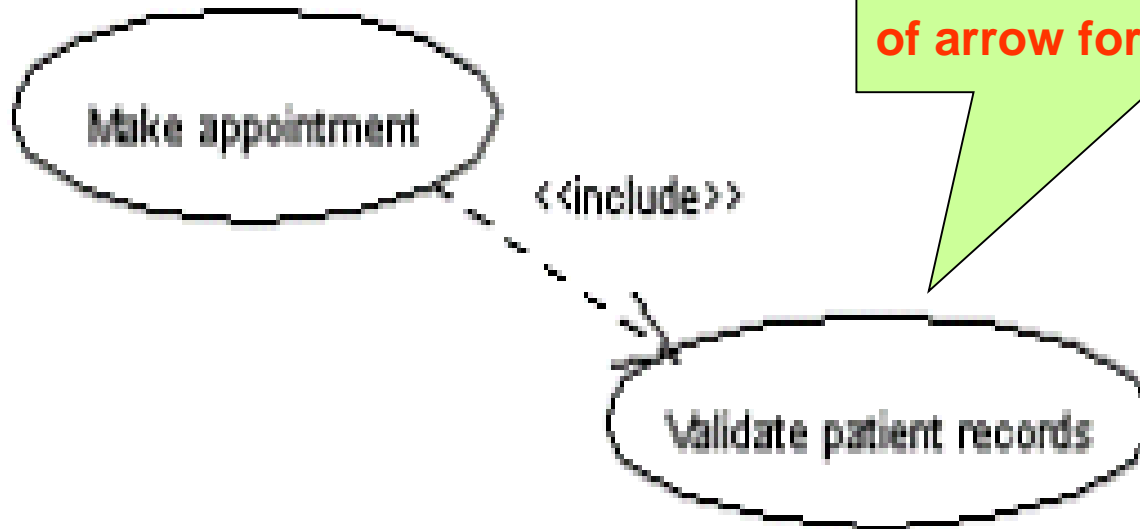
- A teacher can record grades, view grades, and distribute reports.
- A student can view grades.
- An administrator can create report cards.

Use Case Relationships

- Three relationships among use cases
 1. Include
 2. Extend
 3. Generalisation

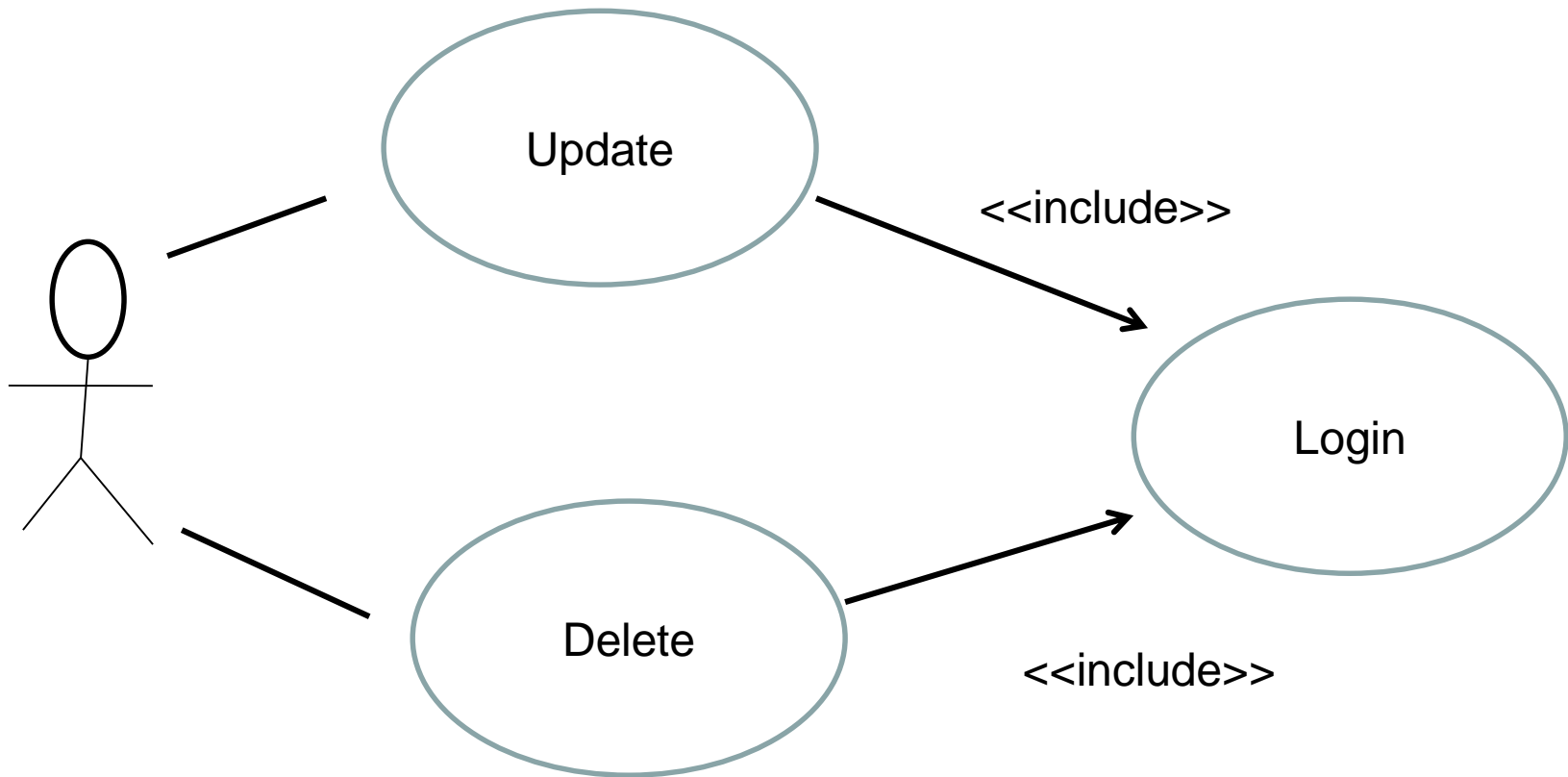
1. Include Relationship

Parent Use Case



- Whenever the "Make appointment" use case occurs, you also do the "Validate patient records" use case

1. Include Relationship



Exercise

- Draw a use case diagram for the following:
A user must login before they can access the internet.

1. Include Relationship

- When a particular use case **uses** the functionality of another use case
- Relationship between the use cases is named as an **include** relationship.
- First use case (parent) often depends on the outcome of the included use case.
- Show and include relationship with a **directed arrow having a dotted shaft**.
- The **tip of the arrowhead points to the parent use case** and the child use case is connected at the base of the arrow.
- Show the label "**<<include>>**"

Exercise

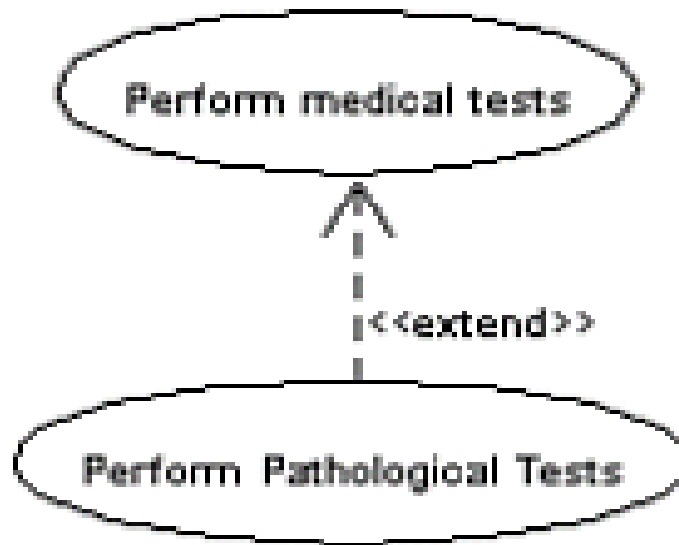


Draw a use case diagram

- Whenever a teacher records grades or updates grades, the grades are saved.

2. Extend Relationship

Parent Use Case



Note: direction & type of arrow for <<extend>>

"Perform Pathological Tests" use case adds to the functionality of the "Perform medical tests" use case i.e. Under certain circumstances, perform pathological tests.

2. Extend Relationship

- A use case may *extend* another
- Relationship indicates that the behavior of the extension use case may be inserted in the extended use case **under some conditions**
- Useful for dealing with **exceptional or special cases**
- Tip of the arrowhead points to the parent use case and the child use case is connected at the base of the arrow.
- Label "**<<extend>>**" shows the relationship as an extend relationship

Exercise

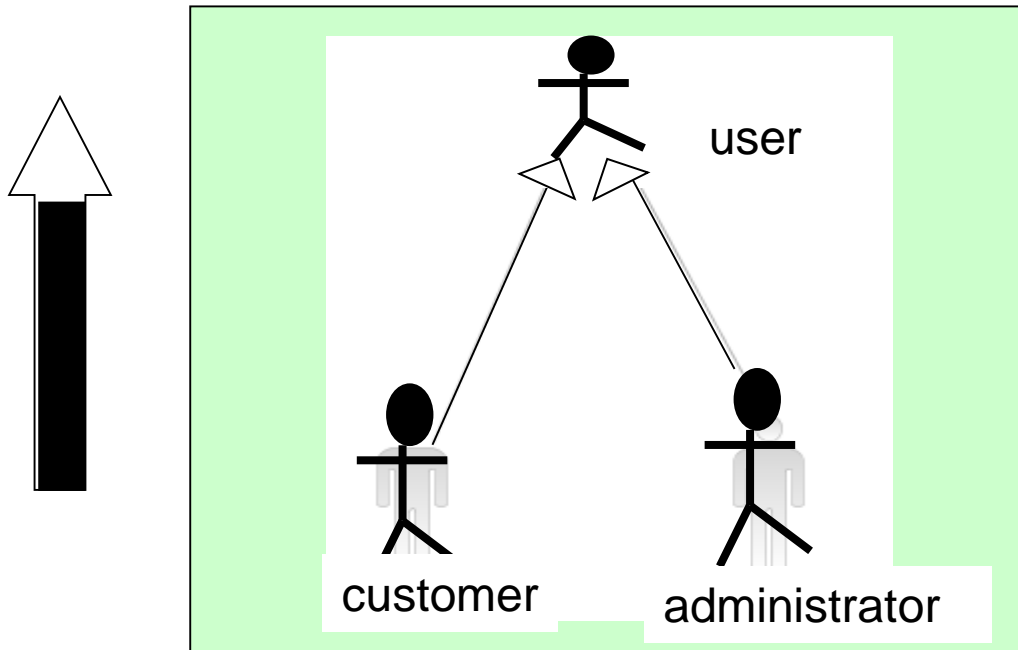


Draw a use case diagram

- Whenever a teacher records grades or updates grades, the grades are saved and in some cases, the student's guardians are contacted.

3a. Actor Generalization

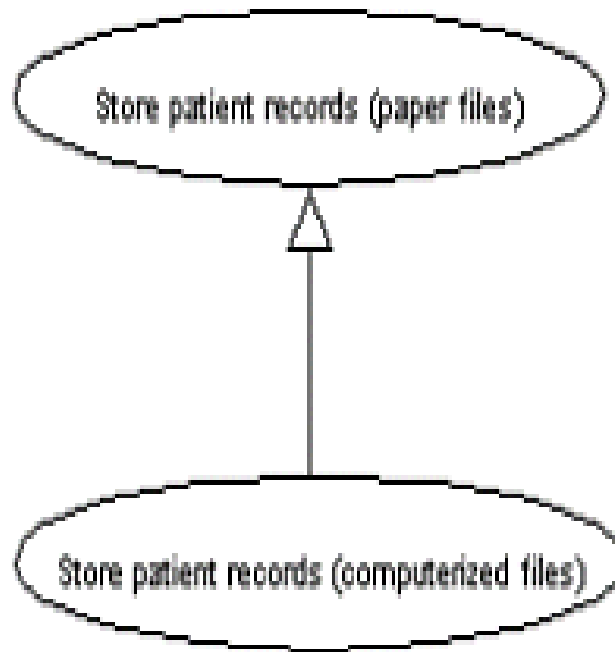
- Only relationship allowed between actors is generalization.
- Useful in defining **overlapping roles between actors**.
- Show as solid line ending in a hollow triangle drawn from the specialized to the more general actor.



3b. Use Case Generalization

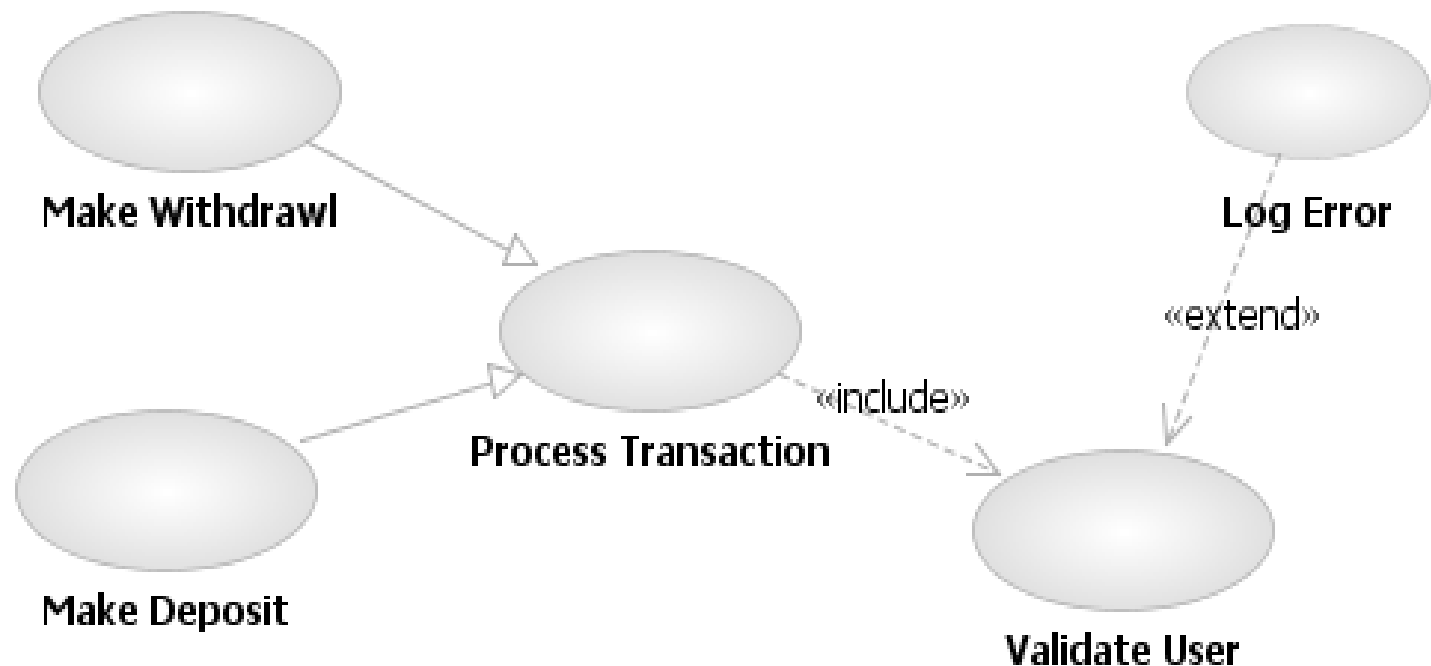
- A *generalization/specialization* relationship may exist among use cases.
- A given use case may be a specialized form of an existing use case.
- Symbol is a solid line ending in a hollow triangle drawn from the specialized to the more general use case.
- Similar to the OO concept of sub-classes

3. Use Case Generalisation



Implies that you can replace any occurrence of the "Store patient records (paper file)" use case with the "Store patient records (computerized file)" use case without affecting how the business works

Include/Extend/Generalisation



Exercise



- Write an explanation of the use case diagram on the previous slide.

Benefits of Use Case Models

- Used to **communicate** with users & domain experts – ensures early buy-in and mutual understanding
- Helps **identification** of all actors interacting with the system, system interfaces, and system functionality
- Helps **verification of requirements** (have they all been captured?)
- Helps **developer understanding of requirements**
- Helps identify suitable iterations in the development process **Decide which use case will be delivered by which iteration in developing the software**

Video Store System

Req#	Requirement
1	<i>Before a video can be rented out, the system confirms customer's identity and standing by swiping over scanner his/her Video Store membership card.</i>
2	<i>A video tape or disk can be swiped over scanner to obtain its description and price (fee) as part of customer's enquiry or rental request.</i>
3	<i>Customer pays the nominal fee before the video can be rented out. The payment may be with cash or debit/credit card.</i>
4	<i>The system verifies all conditions for renting out the video, acknowledges that the transaction can go ahead, and can print the receipt for the customer.</i>

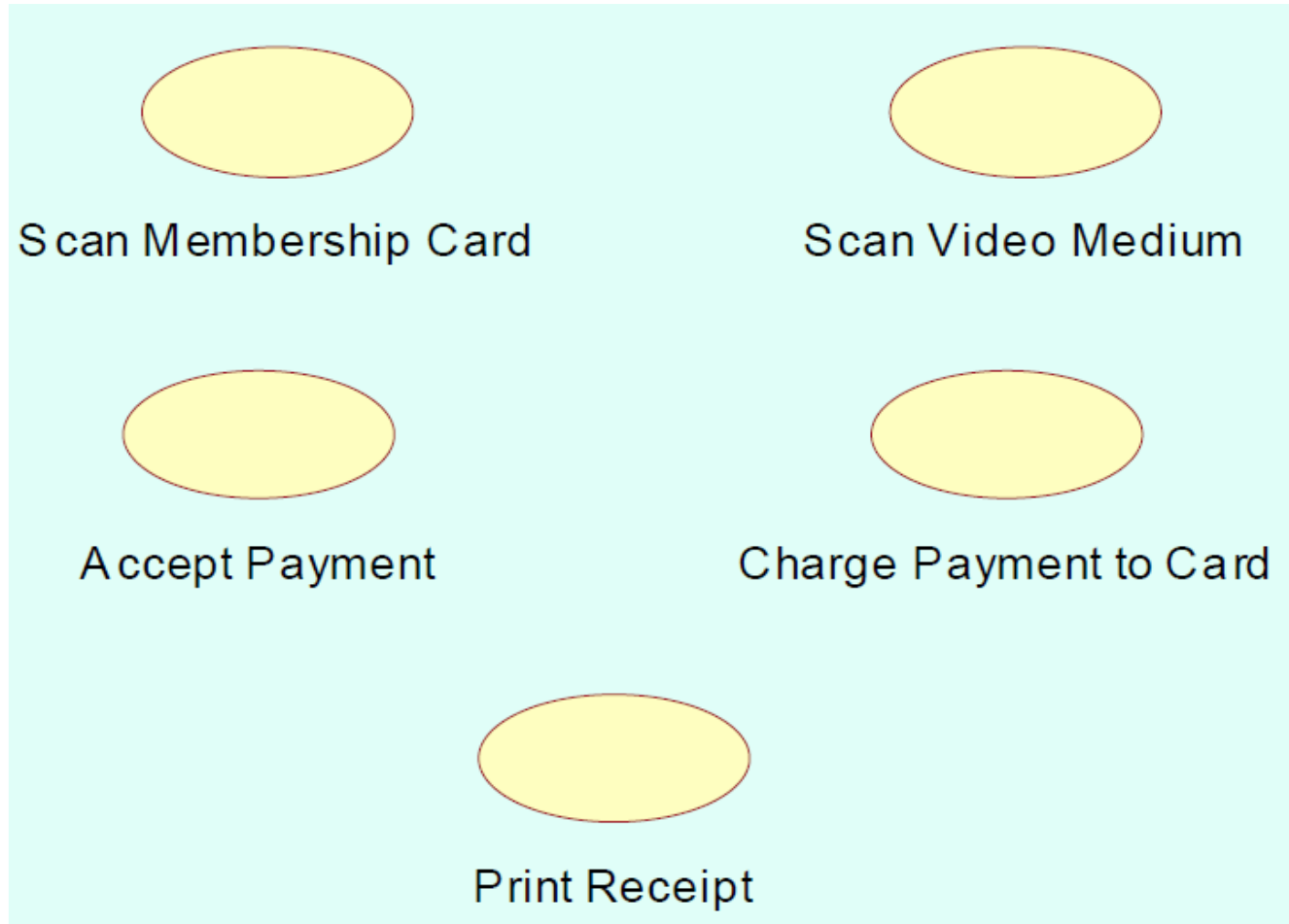
Use Cases for Video Store System

- So where do we get the use cases for the video system from???
- We can **identify the main events in the requirements description**
- Look for the **verbs** and **verb phrases** e.g.
 - **Scan membership card**
 - **Scan video**
 - **Accept Payment etc.**
- These behaviours are themselves composed of sequences of events and therefore would have to be described in greater detail in a use case description.

Use Cases for Video Store System

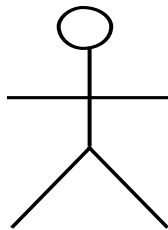
Req#	Requirement	Use case
1	<i>Before a video can be rented out, the system confirms customer's identity and standing by swiping over scanner his/her Video Store membership card.</i>	Scan Membership Card
2	<i>A video tape or disk can be swiped over scanner to obtain its description and price (fee) as part of customer's enquiry or rental request.</i>	Scan Video Medium
3	<i>Customer pays the nominal fee before the video can be rented out. The payment may be with cash or debit/credit card.</i>	Accept Payment Charge Payment to Card
4	<i>The system verifies all conditions for renting out the video, acknowledges that the transaction can go ahead, and can print the receipt for the customer.</i>	Print Receipt

Use Cases for Video Store System

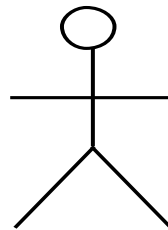


Actors for Video Store System

- Who is involved in carrying out the events....?
- Actors
- **Look at Nouns** – people / roles...
- List actors in the video system



customer



employee

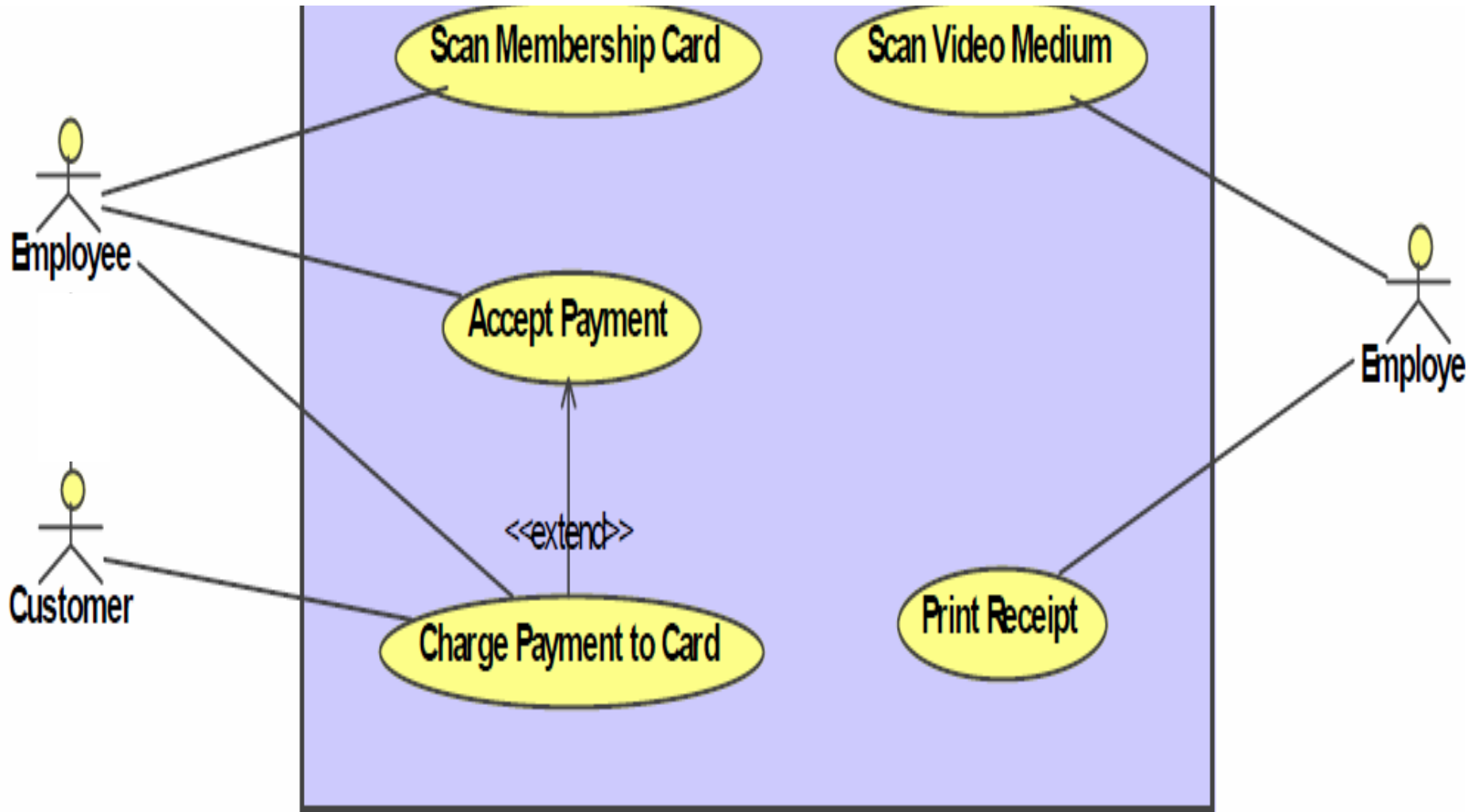
Use Case Diagram of Video Store System

Req#	Requirement	Actor
1	<i>Before a video can be rented out, the system confirms customer's identity and standing by swiping over scanner his/her Video Store membership card.</i>	<i>Customer, Employee</i>
2	<i>A video tape or disk can be swiped over scanner to obtain its description and price (fee) as part of customer's enquiry or rental request.</i>	<i>Customer, Employee</i>
3	<i>Customer pays the nominal fee before the video can be rented out. The payment may be with cash or debit/credit card.</i>	<i>Customer, Employee</i>
4	<i>The system verifies all conditions for renting out the video, acknowledges that the transaction can go ahead, and can print the receipt for the customer.</i>	<i>Employee, Customer</i>

Video Store System

Req#	Requirement	Actor	Use case
1	<i>Before a video can be rented out, the system confirms customer's identity and standing by swiping over scanner his/her Video Store membership card.</i>	<i>Customer, Employee</i>	<i>Scan Membership Card</i>
2	<i>A video tape or disk can be swiped over scanner to obtain its description and price (fee) as part of customer's enquiry or rental request.</i>	<i>Customer, Employee</i>	<i>Scan Video Medium</i>
3	<i>Customer pays the nominal fee before the video can be rented out. The payment may be with cash or debit/credit card.</i>	<i>Customer, Employee</i>	<i>Accept Payment Charge Payment to Card</i>
4	<i>The system verifies all conditions for renting out the video, acknowledges that the transaction can go ahead, and can print the receipt for the customer.</i>	<i>Employee, Customer</i>	<i>Print Receipt</i>

Video Rental System



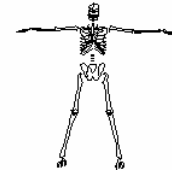
Exercise



Draw a use case diagram for a lift system.

- A user can open the door by pushing a button.
- A user can go to the required floor by pushing the Floor button.
- An engineer can stop the lift system for necessary maintenances.
- After maintenance, the engineer can start the lift again

Exercise



Draw a use case diagram of flight information/booking system.

- A customer can get information about a particular flight.
- A customer can book a flight.
- A customer can cancel a flight.

The system is connected to the main international flight information system and is updated by the main system after every ten minutes.

Any new booking or cancellation is also reported to the main system.

Summary

- **Use case diagrams** describe what a system does from the standpoint of an external observer. The emphasis is on *what* a system does rather than *how*.
- Use case diagrams are closely connected to scenarios. A **scenario** is an example of what happens when someone interacts with the system e.g. clinic system.
- *"A customer calls the clinic to make an appointment for a yearly checkup. The receptionist finds the nearest empty time slot in the appointment book and schedules the appointment for that time slot."*
- A **use case** is a summary of scenarios for a single task or goal.
- An **actor** is who or what initiates the events involved in that task. Actors are simply roles that people or another system plays.

Summary

How are these drawn?

- **Actors** are stick figures. **Use cases** are ovals. Communications are lines that link actors to use cases.
- A **use case diagram** is a collection of actors, use cases, and their communications.
- A **system boundary** rectangle separates the clinic system from the external actors.
- A use case or actor **generalization** shows that one use case or actor is simply a variation of another.
- **Include** relationships are use cases which use another use case. into additional ones.
- **Extend** relationship indicates that one use case is a variation of another or uses another use case, in particular circumstance