Software Design and Analysis
CS-324

Use case

**Lecture # 10,11,12
21, 23, 26 Sept**

Rubab Jaffar
rubab.jaffar@nu.edu.pk

Today's Outline

- Use Cases
- Use Cases Notations
- Examples
- Exercises

Use Cases
Ivar Jacobson 1994
Use Case Model

"What will the System do?"

Use Cases

- **What is a Use Case?**
  - A <u>scenario</u>-based technique in the UML
  - A formal way of representing system functionality, the requirements of the system from the <u>user's perspective</u>.
  - representing how a system interacts with its environment
- **Use Case diagram:** that shows a set of use cases and actors and their relationships.

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

3. 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.

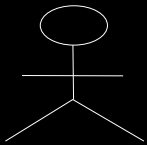5. Use case diagram does not show sequence – unlike DFDs

Components of Use Case Diagram

- Actors
- Use Case
- Relationship
- Boundary

Actors

Humans   Machines   External systems
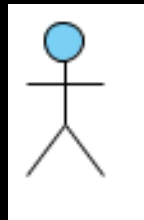
Organizational Units   Sensors
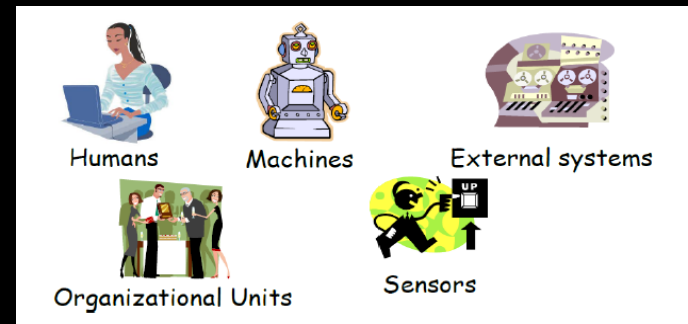
Actors

The people or systems that provide or receive information from the system;
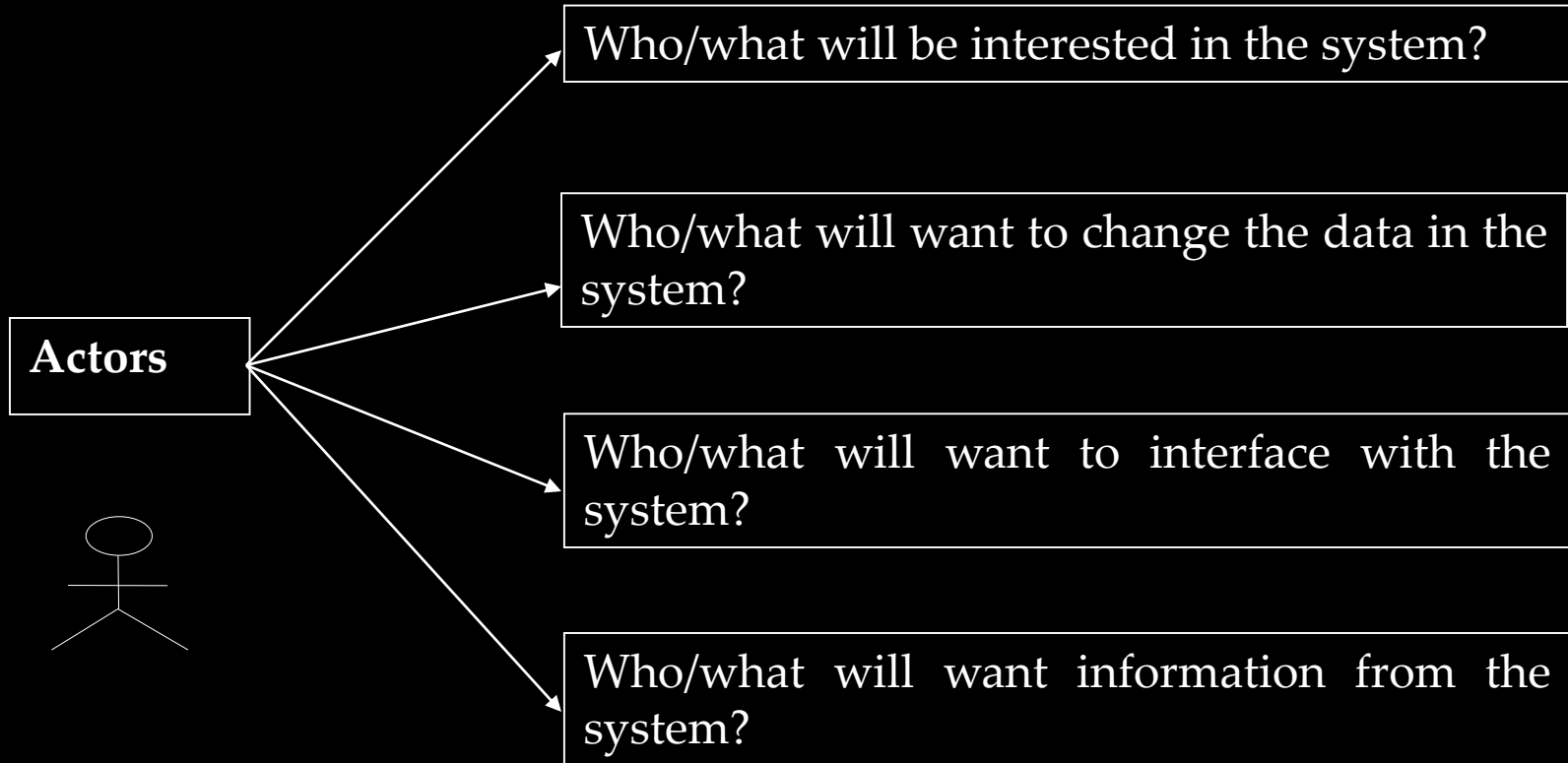Could be human beings, other systems, timers and clocks or hardware devices.

Actors that stimulate the system and are the initiators of events are called primary actors (active)
Actors that only receive stimuli from the system are called secondary actors (passive)

Notation

Actors

Actors

Who/what will be interested in the system?

Who/what will want to change the data in the system?

Who/what will want to interface with the system?

Who/what will want information from the system?

Components of Use Case Model

- Use Case
  - Define the functionality that is handled by the system.
  - Each use case specifies a complete functionality (from its initiation by an actor until it has performed the requested functionality).
  - Describes the interactions between various actors and the system.

- Notation
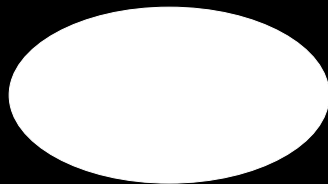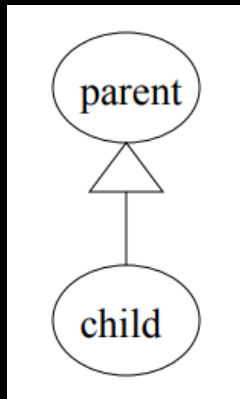
Use Case - Relationships and its Types

- Relationships
  - Represent communication between actor and use case
- 4 types of relationships
  - Association relationship
  - Generalization relationship
    - Generalization relationship between actors
    - Generalization relationship between use cases
  - Include relationship between use cases
  - Extend relationship between use cases

SDA

10

Use Case - Relationships and its Types

- Association relationship: Represent communication between actor and use case

- Often referred to as a communicate association

- use just a line to represent

- Notation _____

Use Case - Relationships and its Types

- Generalization:
- The child use case inherits the behaviour and meaning of the parent use case.
- The child may add to or override the behaviour of its parent.
- Notation:
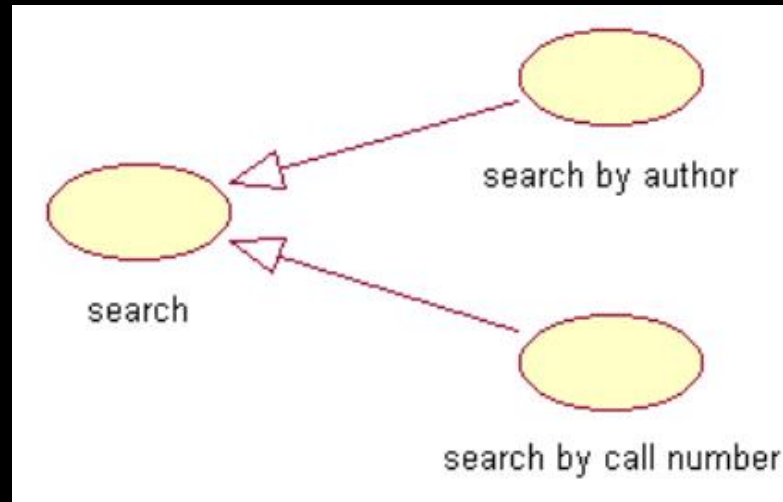
Use Case - Relationships and its Types

- Generalization relationship between actors

  o actor generalization refers to the relationship which can exist between two actors and which shows that one actor (descendant) inherits the role and properties of another actor (ancestor).

- Generalization relationship between use cases

  o use case generalization refers to the relationship which can exist between two use cases and which shows that one use case (child) inherits the structure, behavior, and relationships of another use case (parent).

Use Case - Relationships and its Types
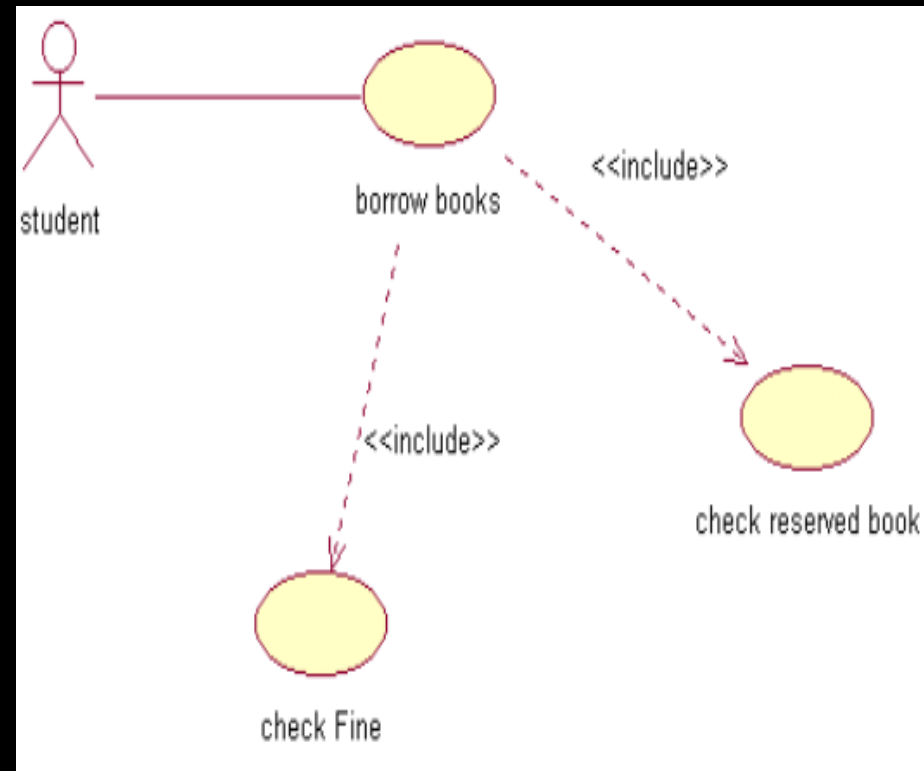
- **Include**
    - o Specifies that the source use case explicitly incorporates the behavior of another use case at a location specified by the source
    - o The include relationship adds additional functionality not specified in the base use case.
    - o <<include>> is used to include common behaviour from an included use case into a base use case
    - o Notation

<center>*<<include>>*</center>
<center>- - - - - - - - - - - - - -></center>

<<include>>

- An include relationship connects a base use case (i.e. borrow books) to an inclusion use case (i.e. check Fine).

- An include relationship specifies how behaviour in the inclusion use case is used by the base use case.

Use Case - Relationships and its Types
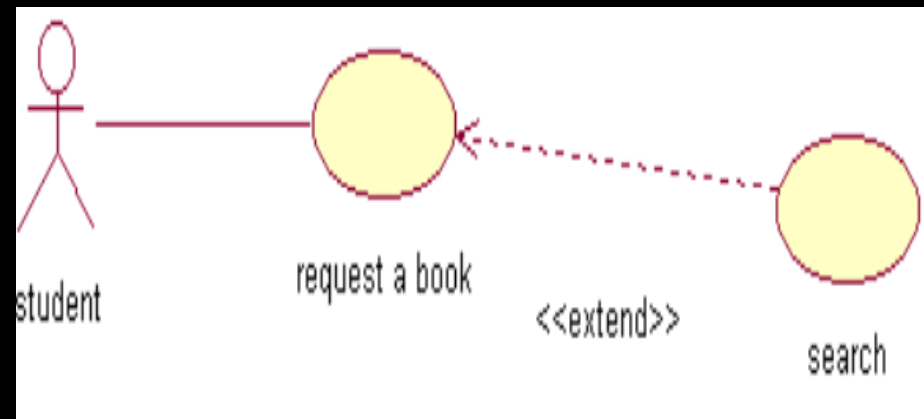
- **Extend**
  - o Specifies that the target use case extends the behavior of the source.
  - o The extend relationships shows optional functionality or system behaviour.
  - o <<extend>> is used to include optional behaviour from an extending use case in an extended use case.

  - o Notation   *<<extend>>*
                <------------------
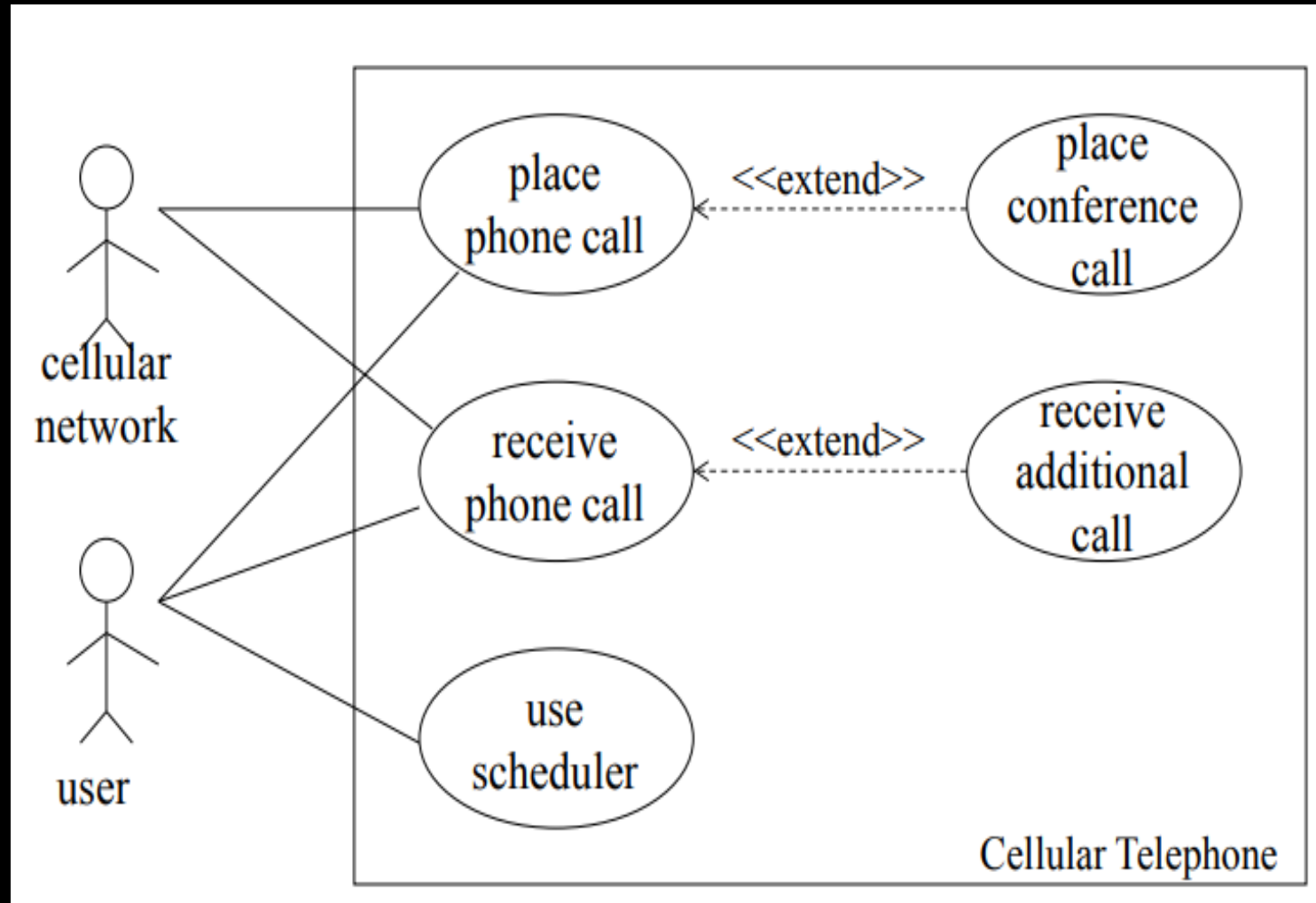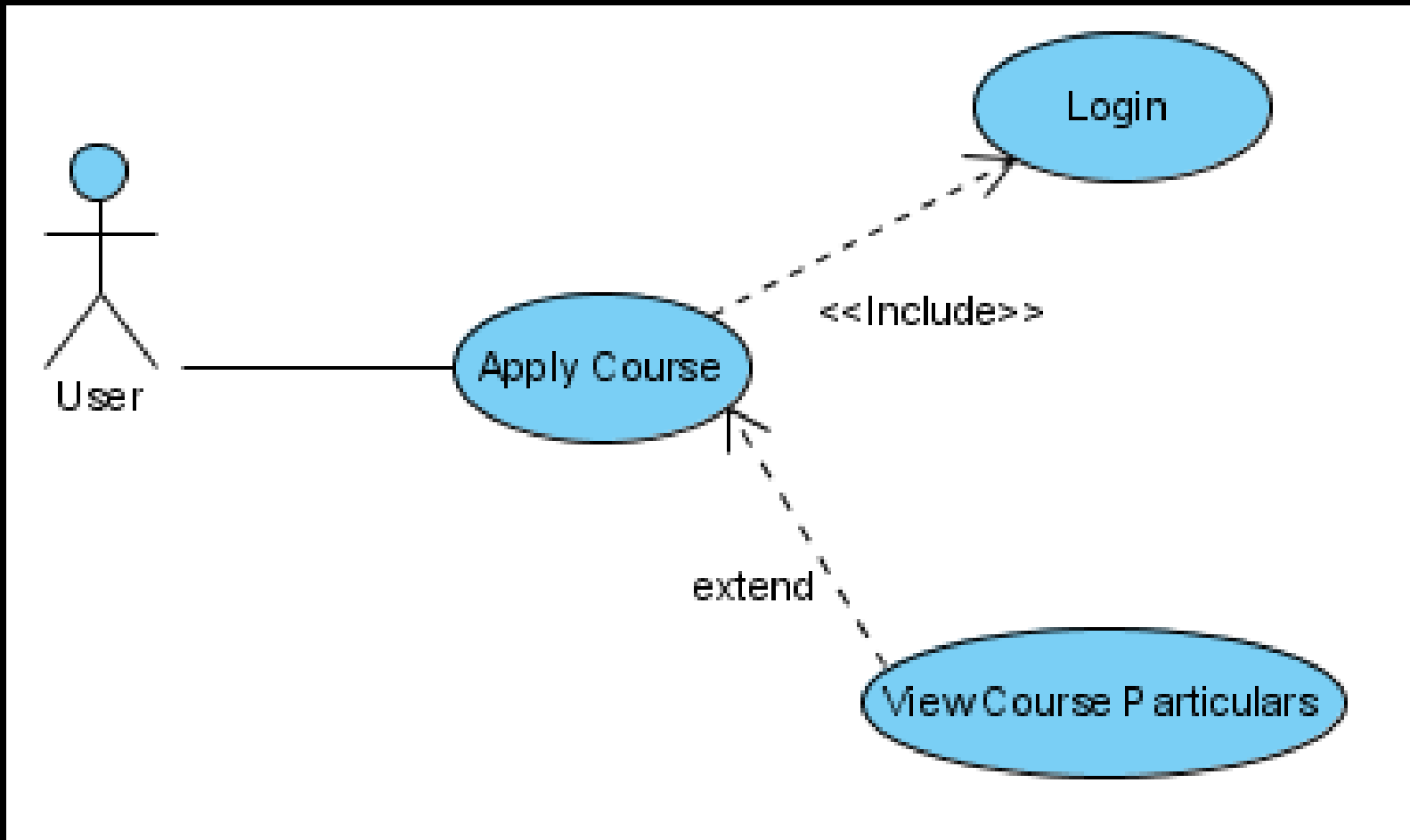
<<extend>>

- The extend relationship is in between Request a book and Search.

- If the student desires, he/she can search the book through the system.

- However, the student may only Request a book through the system without searching the book if the student knows the call number.

student   request a book   <<extend>>   search

Cellular telephone

Example – Include and Extend


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

Use Case - Boundary

- Boundary
  - o A boundary rectangle is placed around the perimeter of the system to show how the actors communicate with the system.
  - o A system boundary of a use case diagram defines the limits of the system.

Use Diagram for a Library System
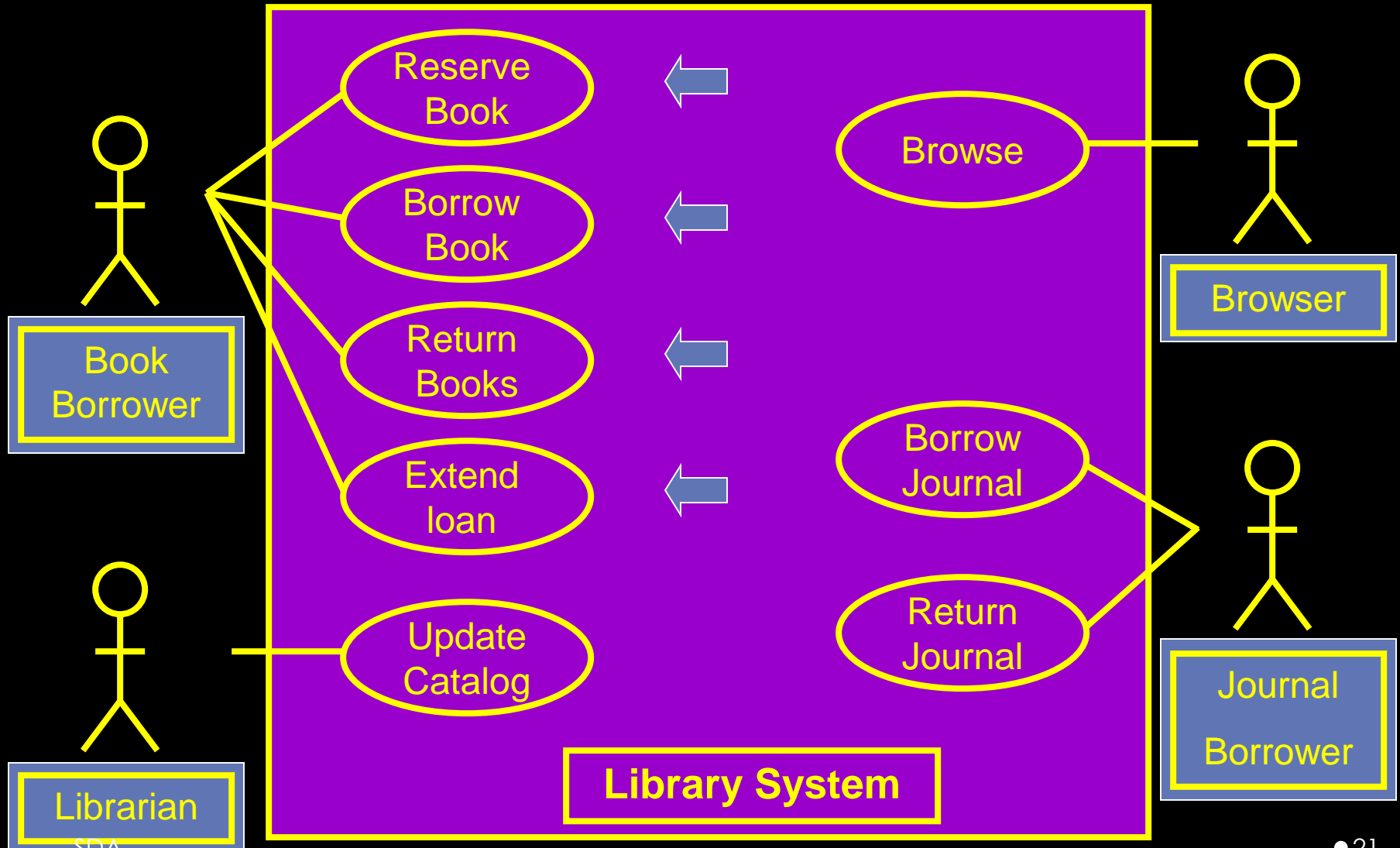
Use Cases

- Here is a scenario for a purchasing items.

- "Customer arrives at checkout with items to purchase in cash. Cashier records the items and takes cash payment. On completion, customer leaves with items. "

- We want to write a use case for this scenario.
- Remember: A **use case** is a summary of scenarios for a single task or goal.

SDA

22

Steps involved in creating use cases

1. Identify use cases.
2. Draw the system boundary.
3. Place the use cases on the diagram.
4. Identify the actors.
5. Add relationships.

<<uses>>
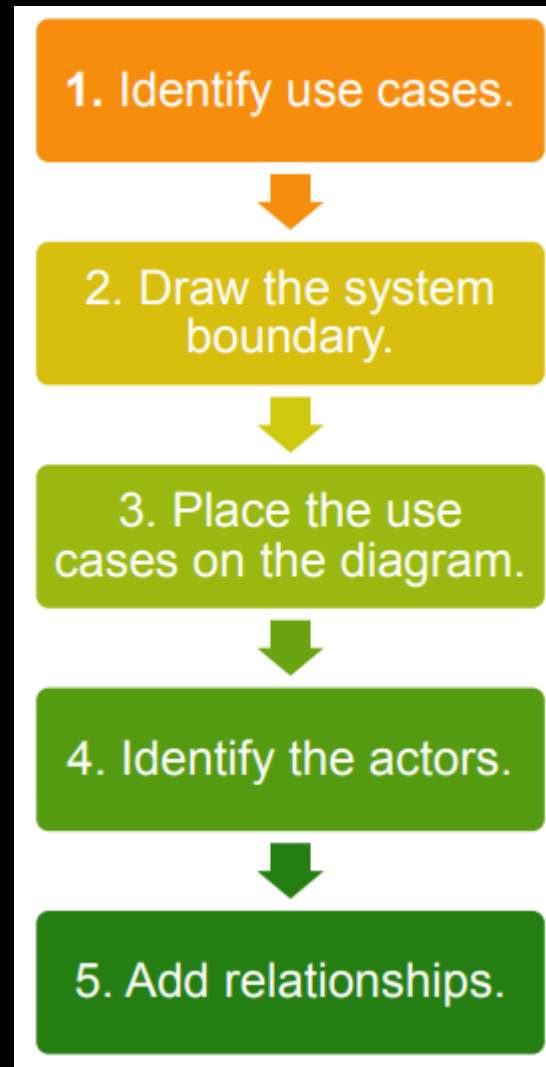
One use case invokes another (like a procedure call)
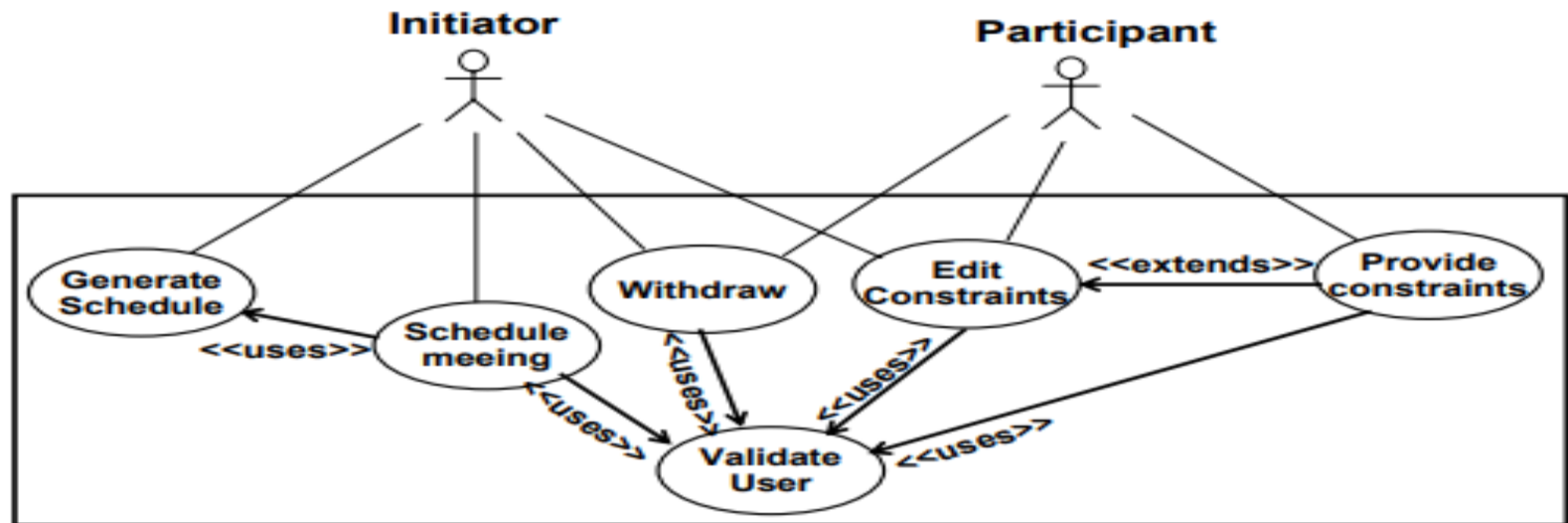- used to avoid describing the same flow of events several times
- puts the common behaviour in a use case of its own.

Example: Meeting Scheduler

CSC340   University of Toronto   10

Difference between Generalization and Extend
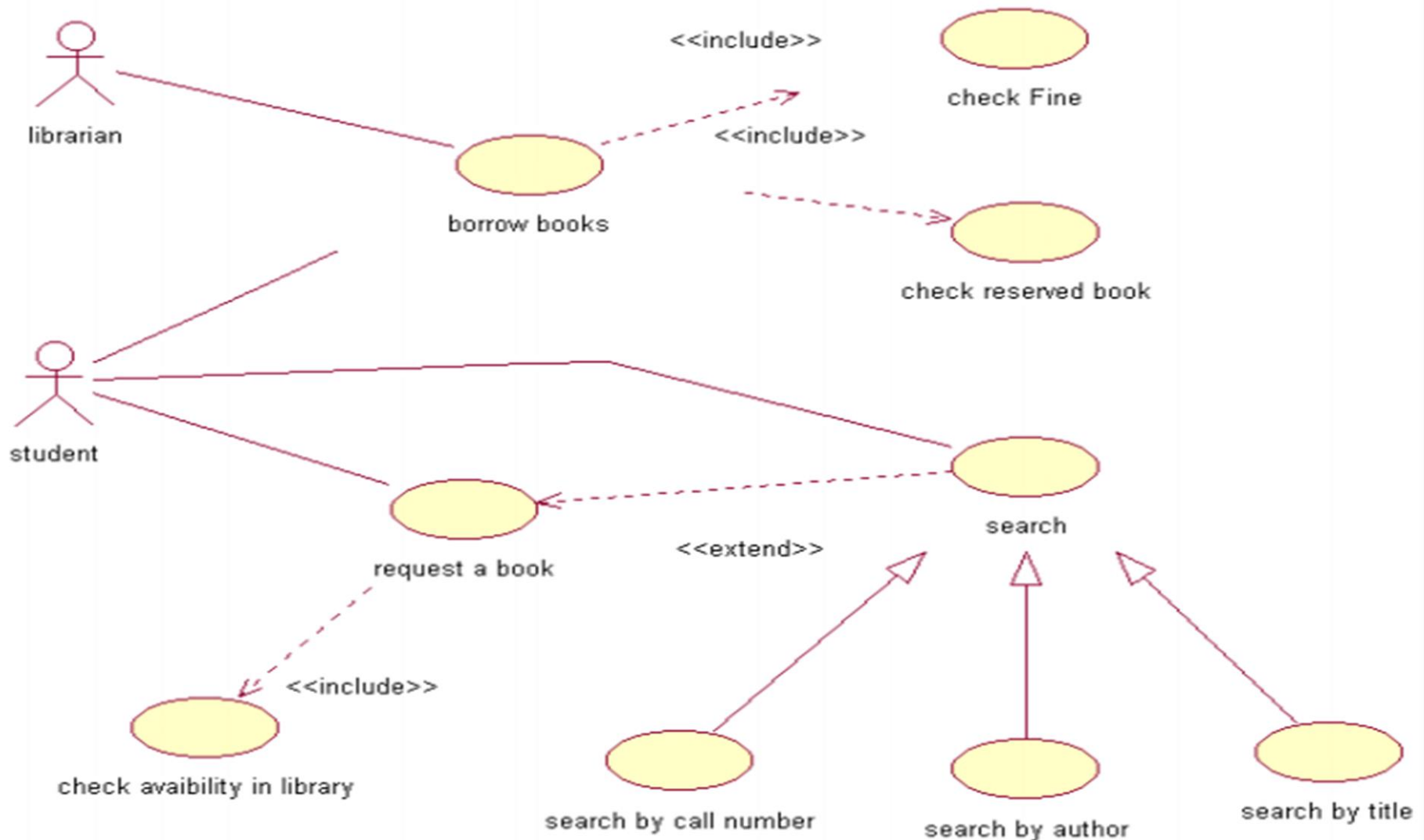
Difference between Generalization and Extend
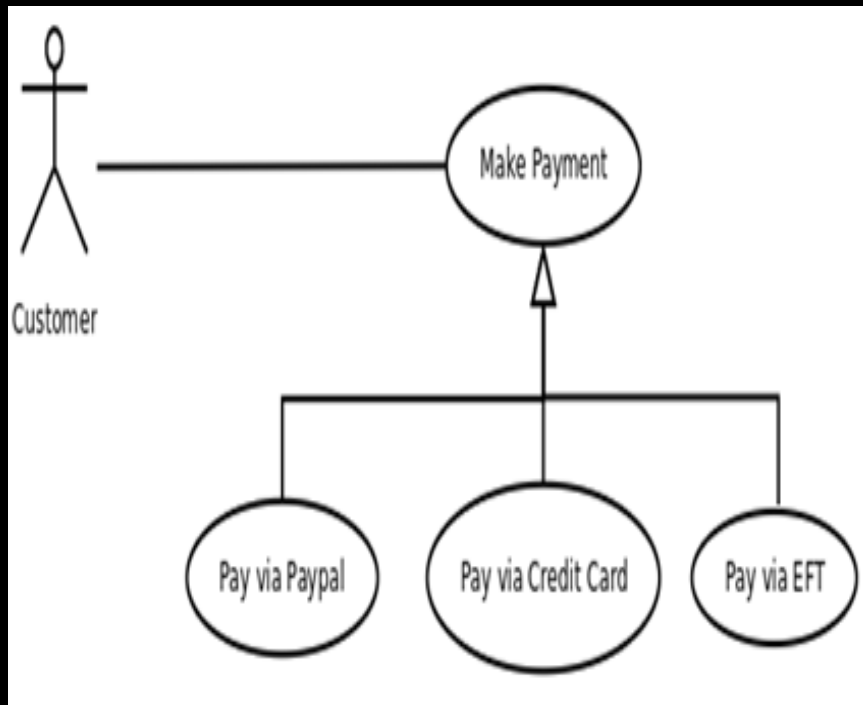
Example

- A user placing an order with a sales company might follow these steps :

- Browse catalog and select items.

- Call sales representative.

- Supply shipping information.

- Supply payment information.

- Receive conformation number from salesperson.

Altered State University (ASU) Registration System

- Professors indicate which courses they will teach on-line.
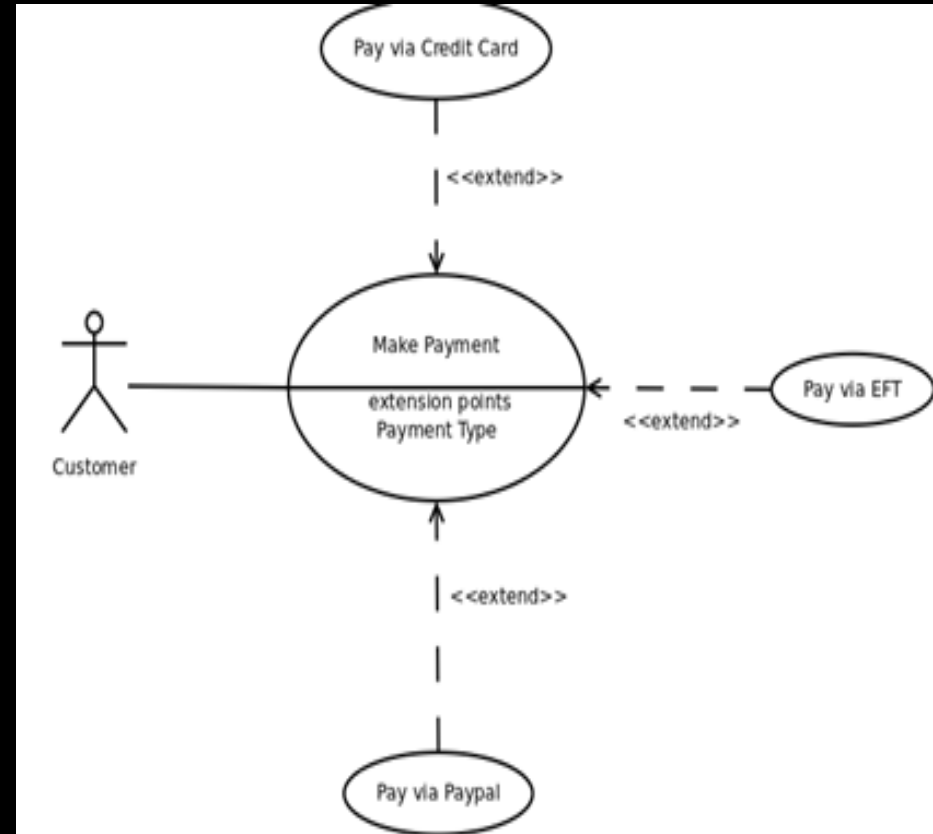
- A course catalog can be printed which is created by Registrar.

- Allow students to select on-line courses for upcoming semester.

- When the registration is completed, the system sends information to the billing system.

- Professors can obtain course rosters on-line.

- Students can add or drop classes on-line.

- Registrar maintains all the information about student, course and professor.

Exercise

- Draw a use case diagram for the hospital reception system. In this system, receptionist can schedule patient appointment and patient hospital admission after the patient registration. Both types of patients i.e. outpatient and inpatient can be admitted in the hospital. Receptionist also checks the insurance and claim forms and put them in file. Patient medical report is also filed by the receptionist.

Exercise

- Draw a use case diagram for the vehicle sales system. Customer makes offer for the vehicle. Customer can be new customer or old customer. New and old customer can make their own offers. For every customer they have to get registered. System can update the existing customer information as well. Customer make payment if his/her offer is accepted. Management has right to accept or reject the offer by managing the offer. Sales person records the sales contract of the accepted offer.

Describing a Use Case
How to write use cases

Alternative Use Case Formats

- Focus on use case analysis with three types of techniques:

- 1. Use case diagram
  - o Components: actors, relationships, use case, system boundary

- 2. Textual / Tabular use case
  - o Full dressed template

- 3. User stories

- A full-dressed use case is very thorough, detailed, and highly structured.

- The project team may decide that a more casual use case format is acceptable.

A Recommended Template

Full Use case Description

Use Case Description	
Use Case name:	
Use Case Description:	
Primary actor:	**Other actors:**
Stakeholders:	
Description:	
Relationships ▪ Includes: ▪ Extends:	
Input:	
Pre-conditions: ▪	
Flow of Events: 1. Actor does…. 3. 4.	
Alternative and exceptional flows: **4.1** ….	
Post-conditions: ▪	

How Do You Write a Use Case?

- Use cases contain the following elements:

- **Name** – A clear actor/verb/noun descriptor that communicates the scope of the use case.

- **Brief Description** – A brief paragraph of text describing the scope of the use case.

- **Actors** – A list of the types of users who can engage in the activities described in the use case.

- **Preconditions** – Anything the solution can assume to be true when the use case begins.

- **Basic Flow** – The set of steps the actors take to accomplish the goal of the use case. A clear description of what the system does in response to each user action.

- **Alternate Flows** – Capture the less common user/system interactions, such as being on a new computer and answering a security question.

- **Exception Flows** – The things that can happen that prevent the user from achieving their goal, such as providing an incorrect username and password.

- **Post Conditions** – Anything that must be true when the use case is complete.

Use Case Basics

- A use case has four mandatory elements:
  1. Name:
  2. Brief description:
  3. Actor(s)
  4. Flow of events

- Optional elements in a Use case:
  - Pre-conditions
  - Post-conditions

Brief Description of Use Case

Create new order *description*

When the customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.

Full Use Case Description

Use Case Name:	Create new order
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.
Actors:	Telephone sales clerk
Related Use Cases:	Includes: *Check item availability*
Stakeholders:	Sales department: to provide primary definition Shipping department: to verify that information content is adequate for fulfillment Marketing department: to collect customer statistics for studies of buying patterns
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.

Full Use Case Description

Flow of Events:

1. Sales clerk answers telephone and connects to a customer.

2. Clerk verifies customer information.

3. Clerk initiates the creation of a new order.

4. Customer requests an item be added to the order.

5. Clerk verifies the item (*Check item availability* use case).

6. Clerk adds item to the order.

7. Repeat steps 4, 5, and 6 until all items are added to the order.

8. Customer indicates end of order; clerk enters end of order.

9. Customer submits payment; clerk enters amount.

Exception Conditions:

2.1 If customer does not exist, then the clerk pauses this use case and invokes *Maintain customer information* use case.

2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative.

4.1 If an item is not in stock, then customer can

    a. choose not to purchase item, or

    b. request item be added as a back-ordered item.

9.1 If customer payment is rejected due to bad-credit verification, then

    a. order is canceled, or

    b. order is put on hold until check is received.

Exercise

- "Customer arrives at checkout with items to purchase in cash. Cashier records the items and takes cash payment. On completion, customer leaves with items. "

Limitations of use cases

What Use Cases Include	What Use Cases Do NOT Include
• Who is using the website • What the user want to do • The user's goal • The steps the user takes to accomplish a particular task • How the website should respond to an action	• Implementation-specific language • Details about the user interfaces or screens.

That is all