CS-3004
SOFTWARE DESIGN AND ARCHITECTURE

RUBAB JAFFAR
RUBAB.JAFFAR@NU.EDU.PK

Introduction

Use case Diagram

Lecture # 7, 8, 9

TODAY'S OUTLINE

- Use Cases
- Use Cases Notations
- Examples
- Exercises

INCEPTION

FIRST PHASE OF THE UNIFIED PROCESS

INCEPTION

- **Inception is the initial short step to establish a common vision and basic scope for the project.**

- It will include analysis of perhaps 10% of the use cases, analysis of the critical non-functional requirement, creation of a business case, and preparation of the development environment so that programming can start in the following elaboration phase.

INCEPTION

- **Inception in one sentence:**
- Envision the product scope, vision, and business case.

- **The main problem solved in one sentence:**
- Do the stakeholders have basic agreement on the vision of the project, and is it worth investing in serious investigation?

INCEPTION ARTIFACTS

Artifact[1	Comment
Vision and Business Case	Describes the high-level goals and constraints, the business case, and provides an executive summary.
Use-Case Model	Describes the functional requirements. During inception, the names of most use cases will be identified, and perhaps 10% of the use cases will be analyzed in detail.
Supplementary Specification	Describes other requirements, mostly non-functional. During inception, it is useful to have some idea of the key non-functional requirements that have will have a major impact on the architecture.
Glossary	Key domain terminology, and data dictionary.
Risk List & Risk Management Plan	Describes the risks (business, technical, resource, schedule) and ideas for their mitigation or response.
Prototypes and proof-of-concepts	To clarify the vision, and validate technical ideas.
Iteration Plan	Describes what to do in the first elaboration iteration.
Phase Plan & Software Development Plan	Low-precision guess for elaboration phase duration and effort. Tools, people, education, and other resources.
Development Case	A description of the customized UP steps and artifacts for this project. In the UP, one always customizes it for the project.

IDENTIFYING REQUIREMENTS

- A **requirement** may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification
- There are two types of requirements:
  - **User Requirements:**
    - Written for customers.
    - Often written in natural language (no technical details)
  - **System Requirements:**
    - Written for developers
    - Detailed functional and non-functional requirements discussed.
    - Include clear implementation details

SYSTEM STAKEHOLDERS

- End users
- System managers
- System owners
- External stakeholders

FUNCTIONAL REQUIREMENTS:

- Statements of services the system should provide
- How the system should react to particular inputs and in particular situations.
- May state what the system should not do

NON-FUNCTIONAL REQUIREMENTS

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Similar to Quality attributes
- Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless

USE CASES

USE CASES
IVAR JACOBSON 1994
USE CASE MODEL

"WHAT WILL THE SYSTEM DO?"

USE CASES

- What is a Use Case?
  - A scenario-based technique in the UML
  - A formal way of representing system functionality, the requirements of the system from the user's perspective.
  - 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

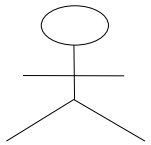
- Actors
- Use Case
- Relationship
- Boundary

ACTORS

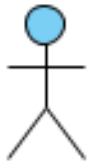
Actors

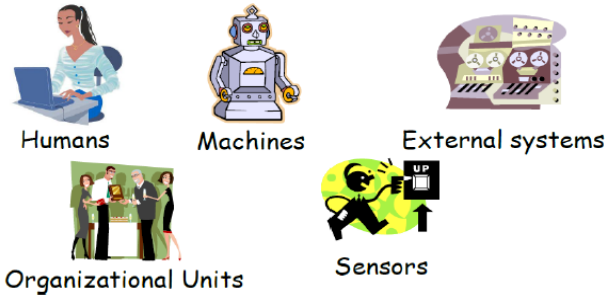
The people or systems that provide or receive information from the system;
Could be human beings, other systems, timers and clocks or hardware devices.

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)

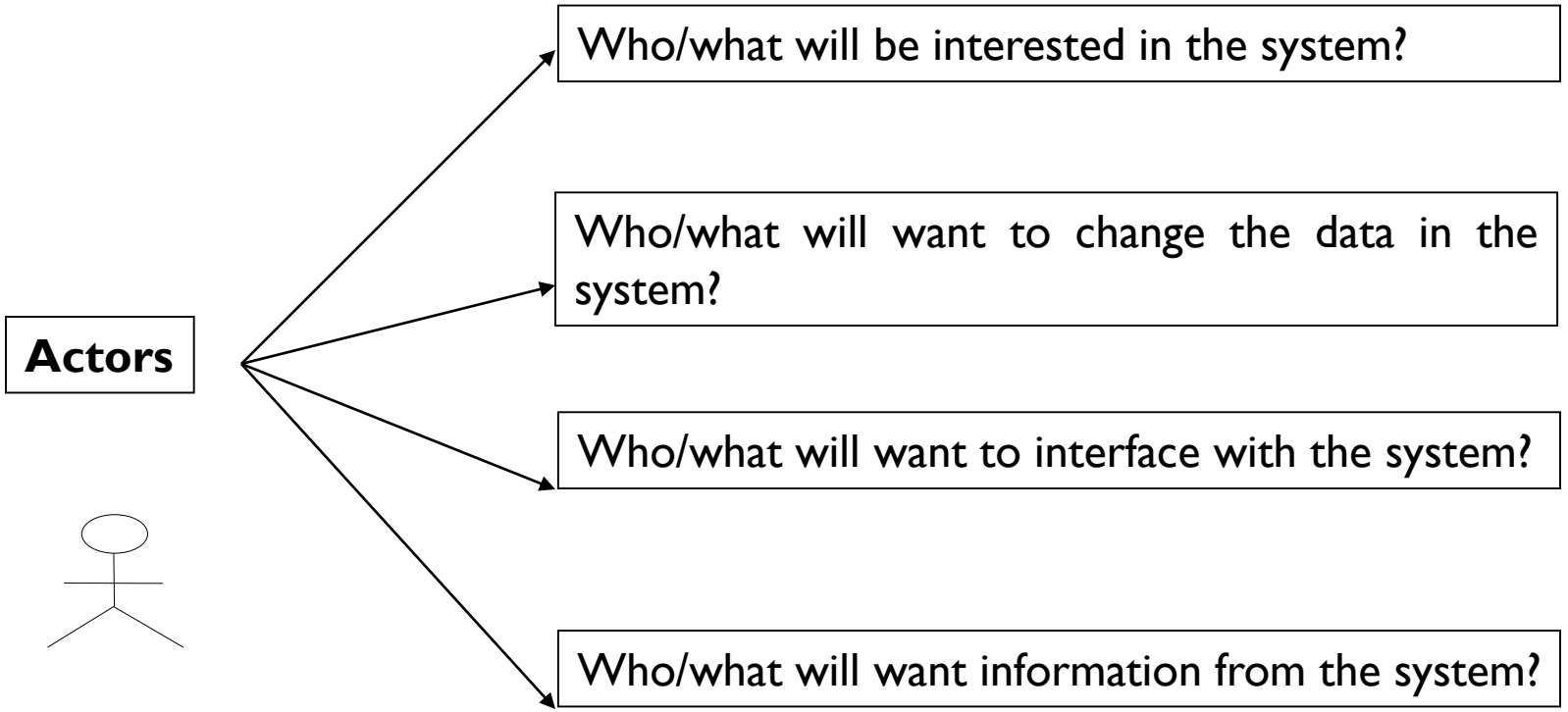


Notation





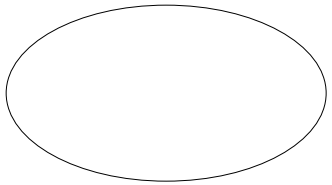
ACTORS


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



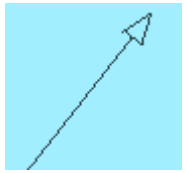
USE CASE - RELATIONSHIPS AND ITS TYPES

- 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

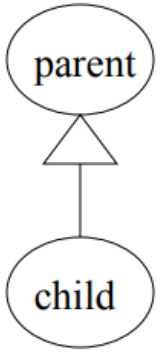
USE CASE - RELATIONSHIPS AND ITS TYPES

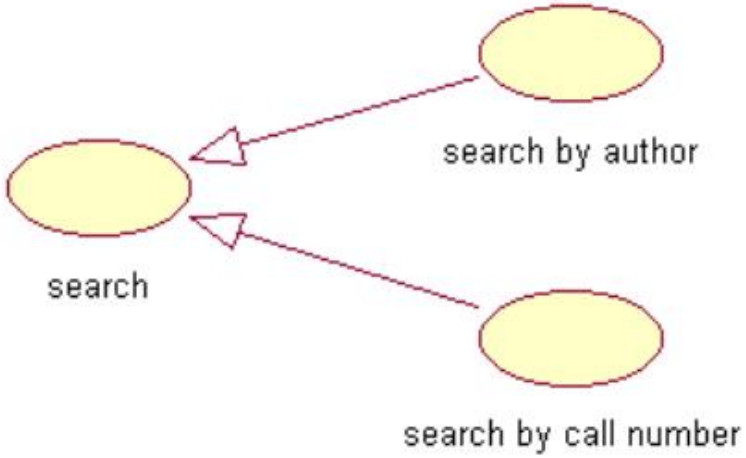
- Association relationship: Represent communication between actor and use case
- Often referred to as a communicate association
- use just a line to represent
- Notation _____

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 o
- Notation:

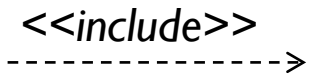




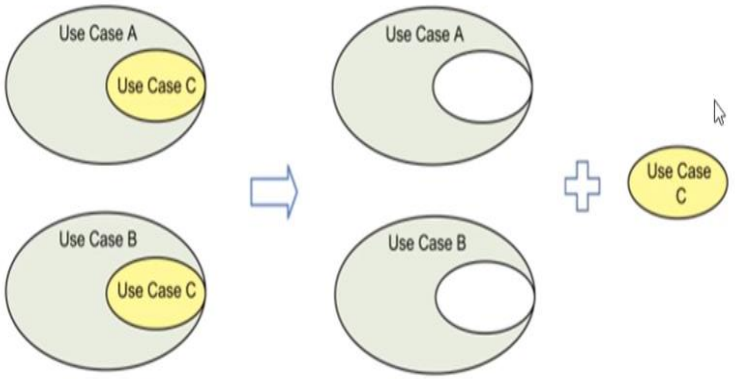
USE CASE - RELATIONSHIPS AND ITS TYPES

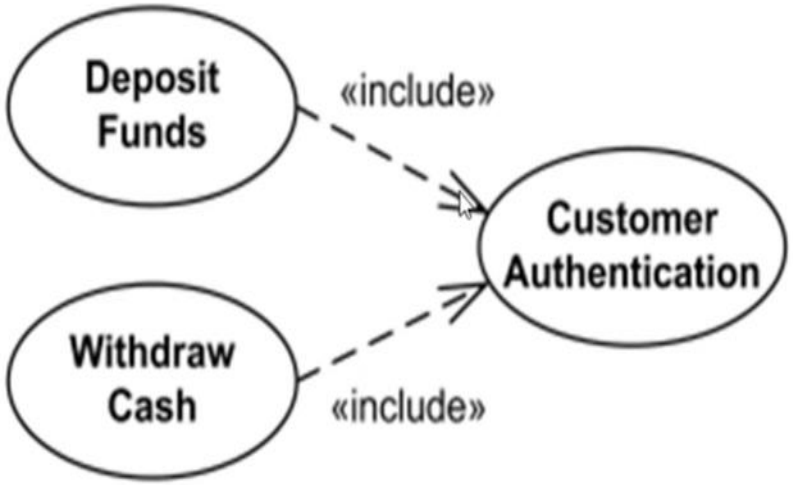
- Generalization relationship between actors
  - actor generalization refers to the relationship which can exist between two actors and which shows that one actor (descendant) inherits the role and properties of another actor (ancestor).
- Generalization relationship between use cases
  - use case generalization 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**
  - Specifies that the source use case explicitly incorporates the behavior of another use case at a location specified by the source
  - The include relationship adds additional functionality not specified in the base use case.
  - <<include>> is used to include common behaviour from an included use case into a base use case
  - Notation 

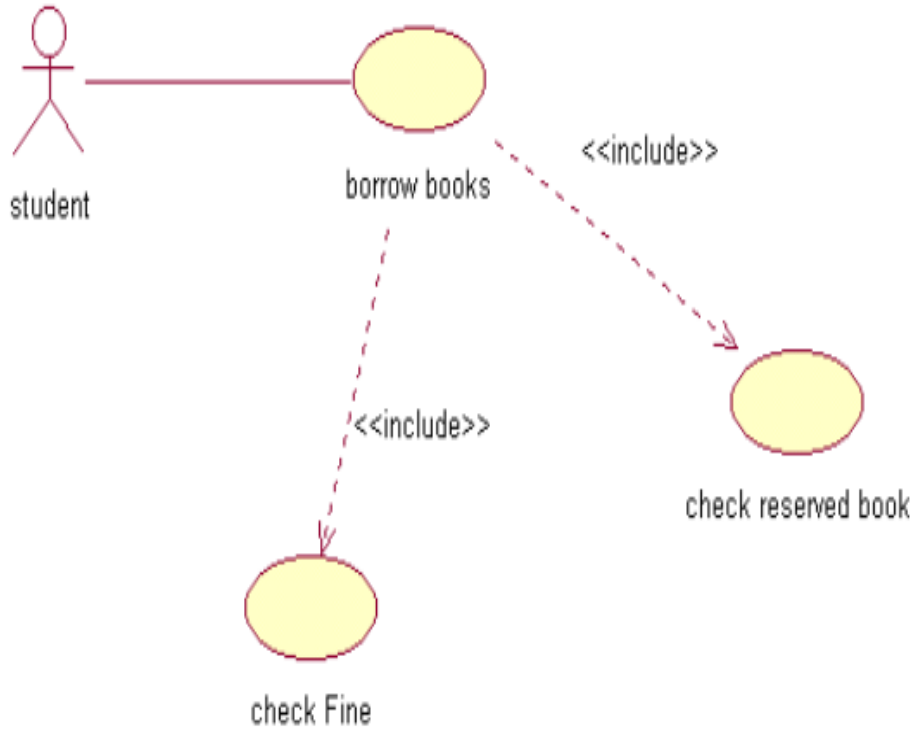
CONCEPT OF INCLUDE
PRACTICAL EXAMPLE OF INCLUDE





<<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.

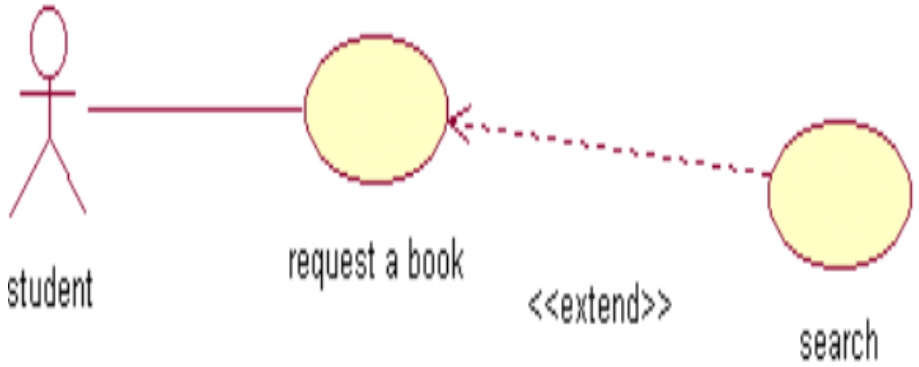


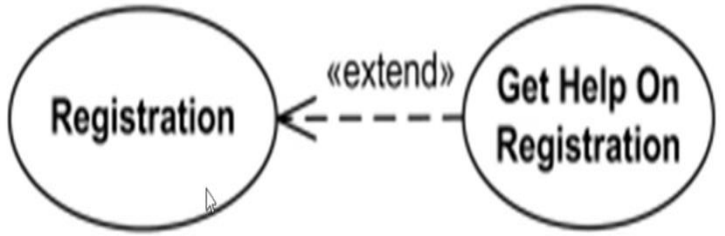
USE CASE - RELATIONSHIPS AND ITS TYPES

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

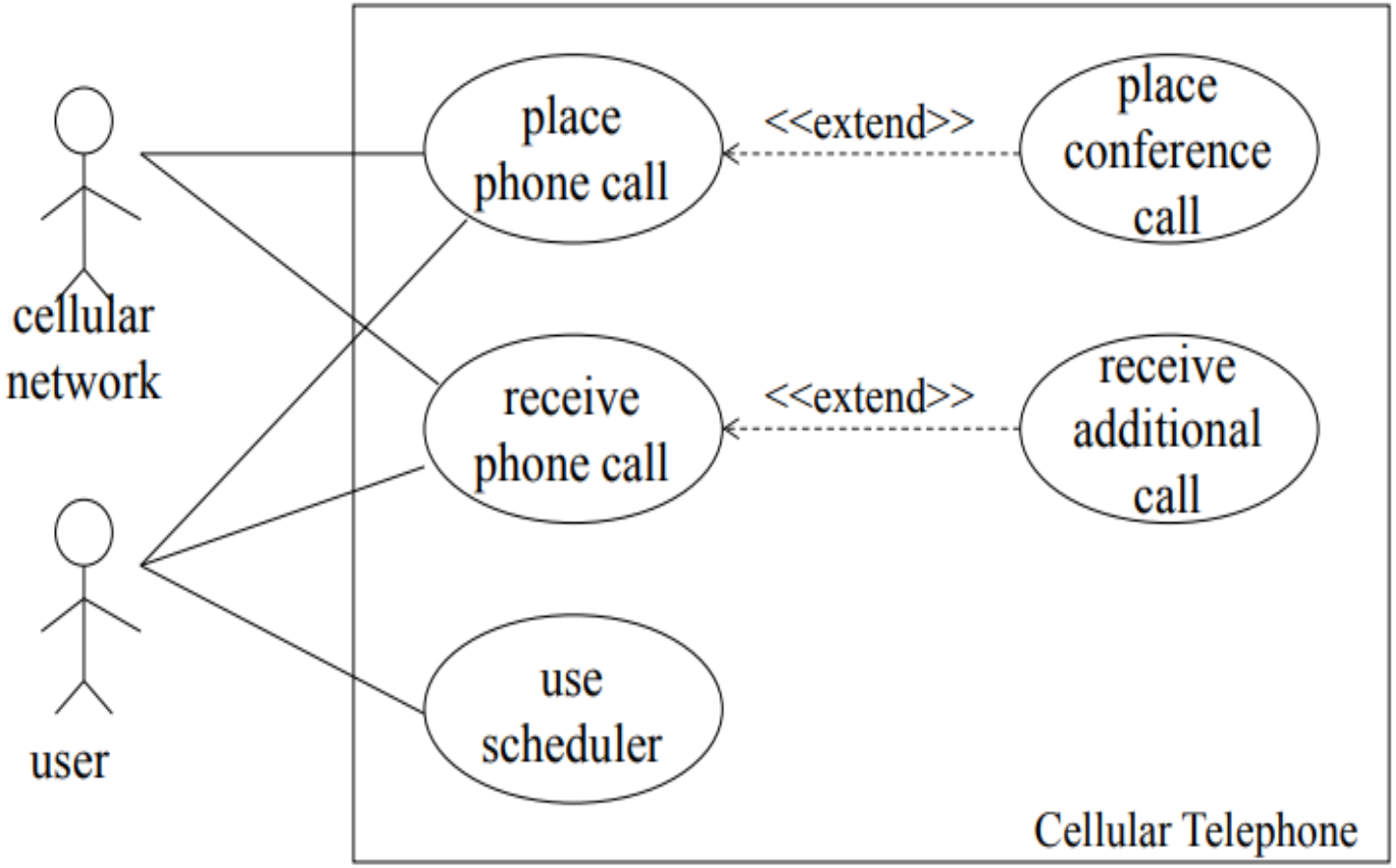
<<EXTEND>>

- 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.

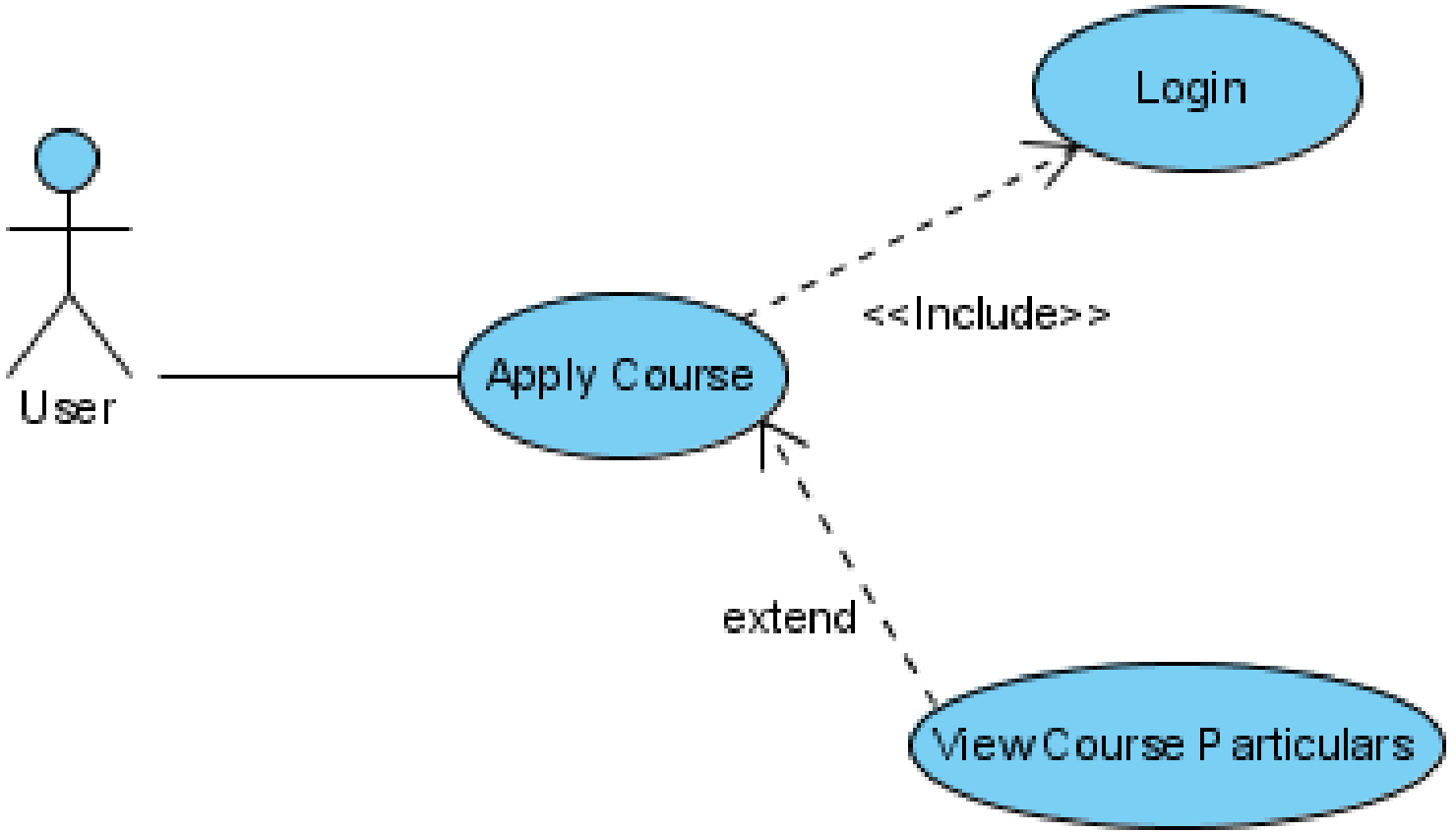




CELLULAR TELEPHONE

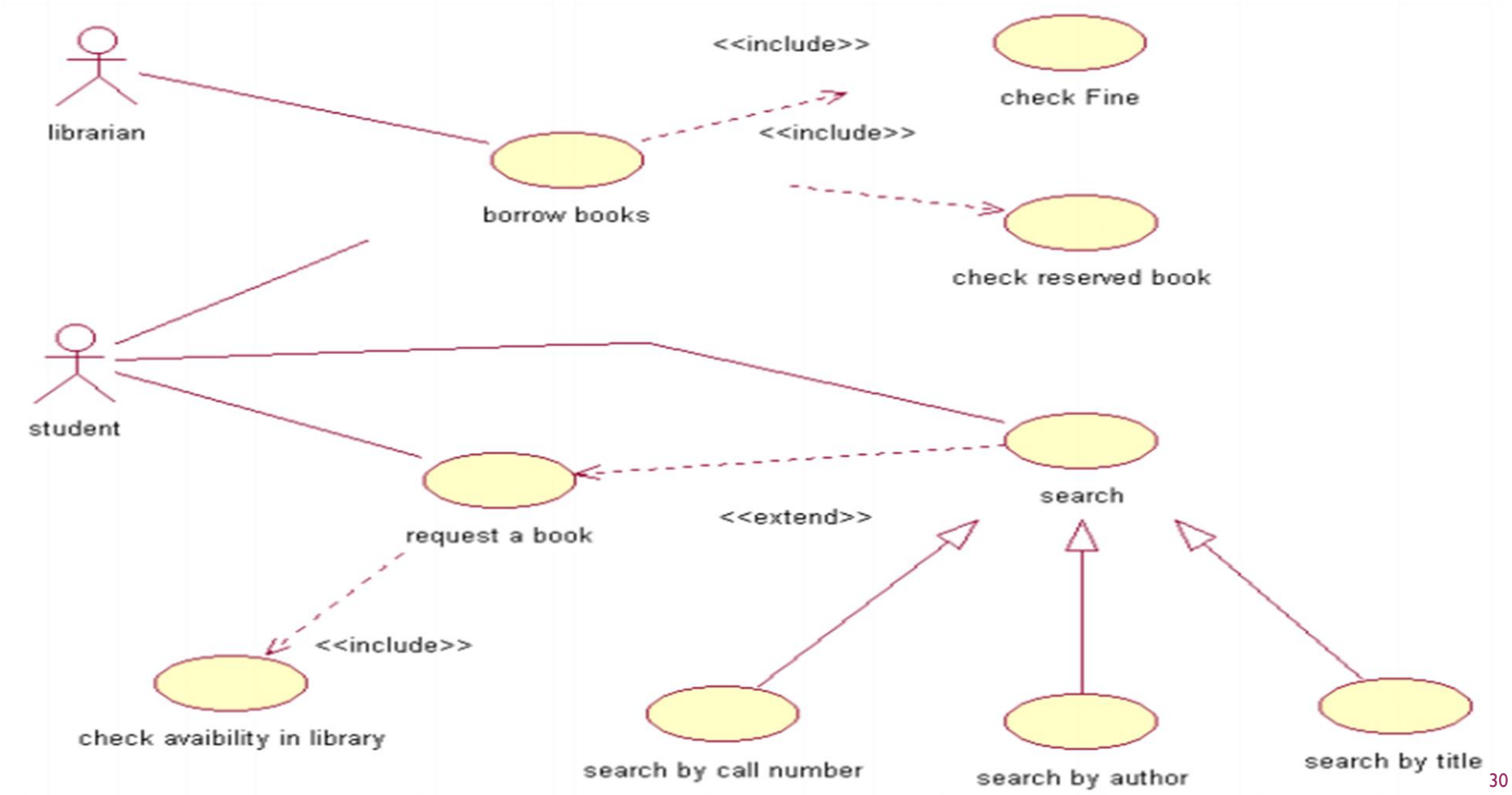


EXAMPLE – INCLUDE AND EXTEND

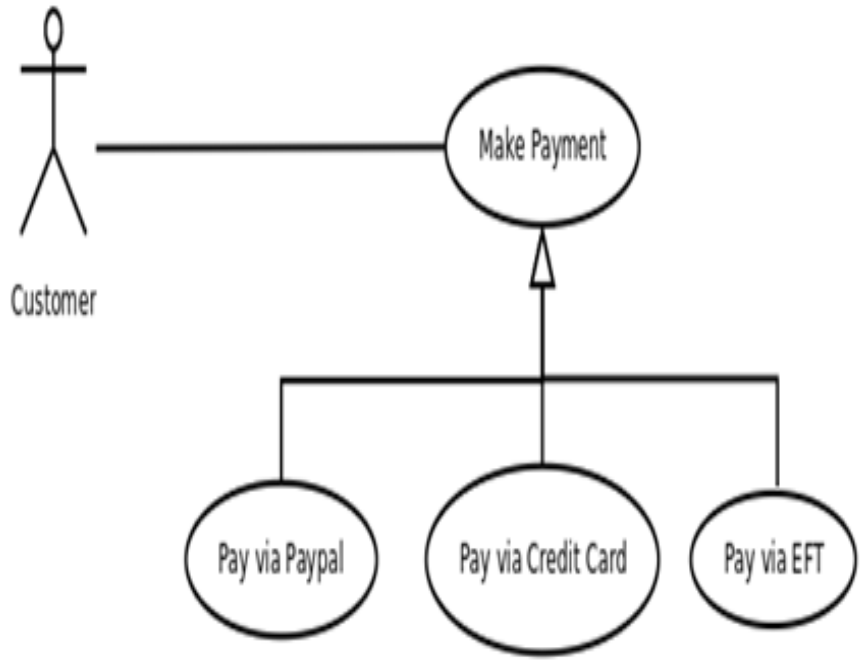


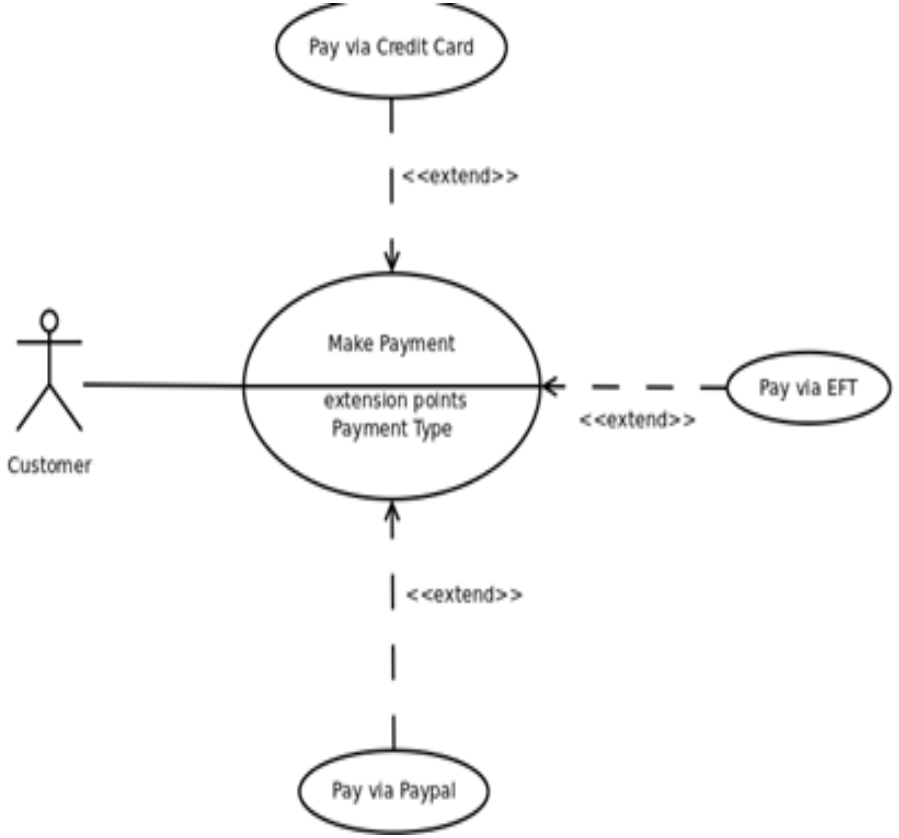
The use of include and extend is discouraged simply because they add unnecessary complexity to a Use Case diagram.

Difference between Generalization and Extend



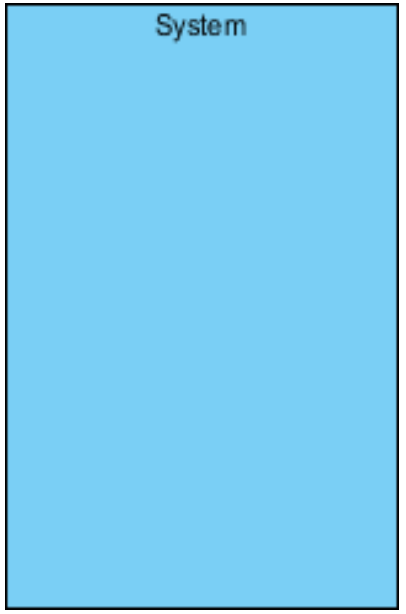
Difference between Generalization and Extend



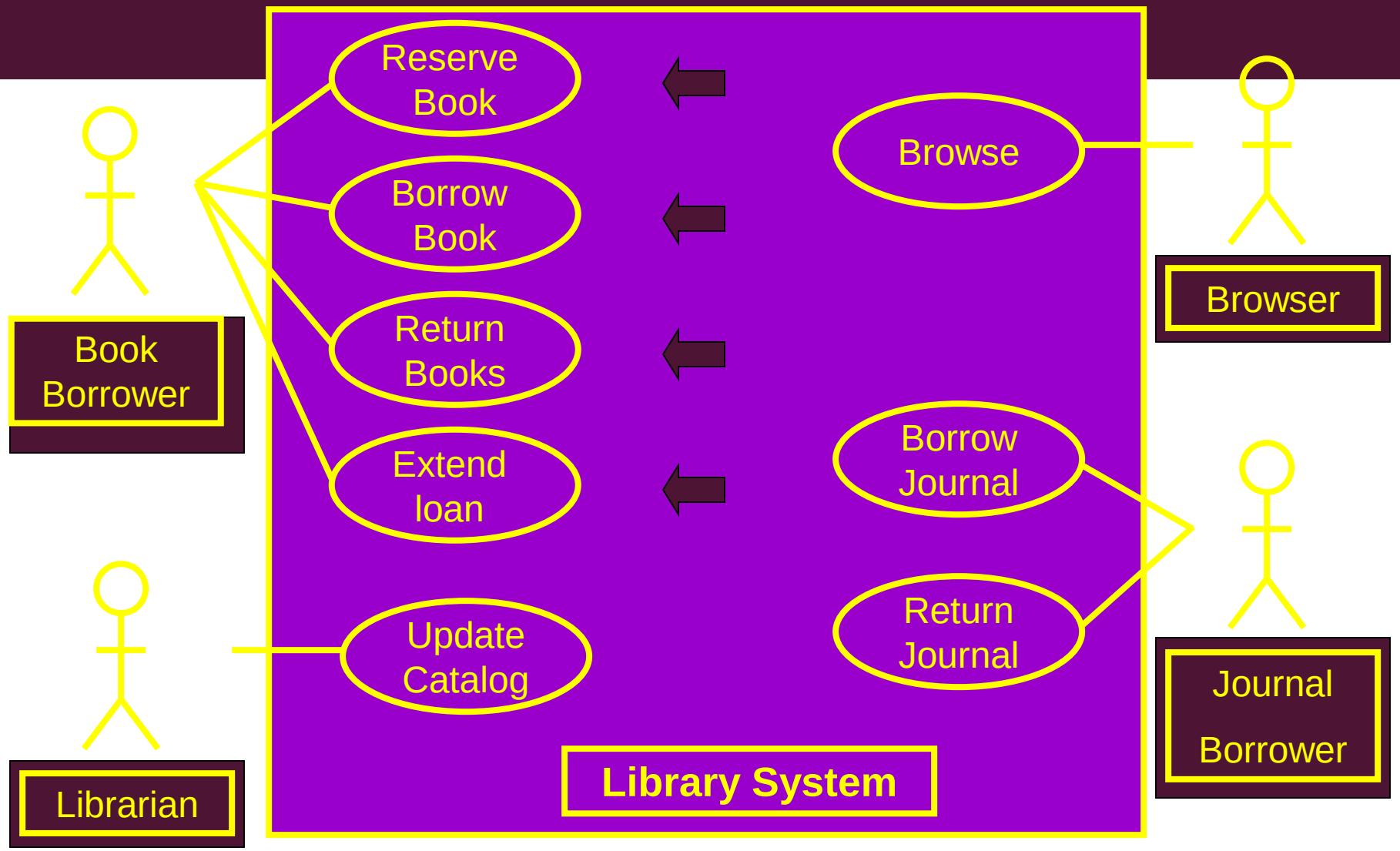


USE CASE - BOUNDARY

- Boundary
  - A boundary rectangle is placed around the perimeter of the system to show how the actors communicate with the system.
  - 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.

USE CASES

- Step 1 Identify the actors
- As we read the scenario, define those people or systems that are going to interact with the scenario.

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

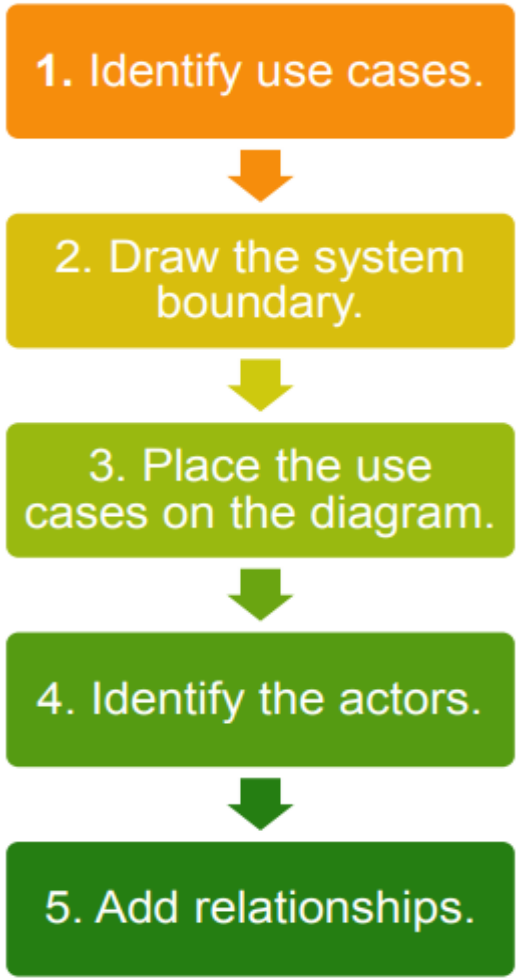
QUESTIONS FOR IDENTIFYING PEOPLE ACTORS

- Who is interested in the scenario/system?
- Where in the organization is the scenario/system be used?
- Who will benefit from the use of the scenario/system?
- Who will supply the scenario/system with this information, use this information, and remove this information?
- Does one person play several different roles?
- Do several people play the same role?

QUESTIONS FOR IDENTIFYING OTHER ACTORS

- What other entity is interested in the scenario/system?
- What other entity will supply the scenario/system with this information, use this information, and remove this information?
- Does the system use an external resource?
- Does the system interact with a legacy system?

STEPS INVOLVED IN CREATING USE CASES



ACTORS, SCENARIOS & USE CASES

- An **actor** is something with behavior, such as a person (identified by role), computer system, or organization; for example, a cashier.
- A **scenario** is a specific sequence of actions and interactions between actors and the system; it is also called a **use case instance**.
- A **use case** is a collection of related success and failure scenarios that describe an actor using a system to support a goal.

TYPES OF ACTORS

- **Primary actor** has user goals fulfilled through using services of the system. For example, the cashier.

- Why identify?
- To find user goals, which drive the use cases.

TYPES OF ACTORS

- **Secondary actor** provides a service (for example, information) to the system. The automated payment authorization service is an example. Often a computer system, but could be an organization or person.

- Why identify?
- To clarify external interfaces and protocols.

TYPES OF ACTORS

- **Offstage actor** has an interest in the behavior of the use case, but is not primary or secondary; for example, a government tax agency.

- Why identify?
- To ensure that all necessary interests are identified and satisfied. Offstage actor interests are sometimes subtle or easy to miss unless these actors are explicitly named.

EXAMPLE:

- Suppose you have to work on IncreasePaisa project that is a banking app.
- IncreasePaisa should provide a set of core functionalities that are among the most demanded tasks users want to compete on their smartphones.
- These core functionalities include basic financial transactions such as login, balance checking, digital payments, money transfers, and mobile deposits. If the customer is new to app, he/she first has to register with the app. When the customer types in the login information, the app is going to verify password before completing the login process. But if the information is incorrect, the app is going to display an error message. When the customer wants to transfer funds or make payment, the app is going to make sure that there is enough money to complete those transactions. If the customer wants to make payment, the banking app will provide them the options to either pay with checking account or with saving account.
- Customer can setup his/her profile in the IncreasePaisa app. when the customer is setting up his/her profile, they can navigate between two different options: customer can seek profilehelp if he has any confusion in setting up profile; Customer can also check his privacy information.

TYPES OF USE CASES

- **Brief:**
- Terse one-paragraph summary, usually of the main success scenario.
- Used during early requirements analysis, to get a quick sense of subject and scope. May take only a few minutes to create.
- **Example:**
- **Process Sale:** A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

TYPES OF USE CASES

- **Casual**
- Informal paragraph format. Multiple paragraphs that cover various scenarios.
- Used during early requirements analysis, to get a quick sense of subject and scope. May take only a few minutes to create.
- **Example:**
- **Handle Returns:**
- Main Success Scenario: A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item …
- Alternate Scenarios: If the customer paid by credit, and the reimbursement transaction to their credit account is rejected, inform the customer and pay them with cash. If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).

TYPES OF USE CASES

- **Fully dressed**
- All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees.

- After many use cases have been identified and written in a brief format, then during the first requirements workshop a few (such as 10%) of the architecturally significant and high-value use cases are written in detail.

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.

YOUR TURN: CASE STUDY – WRITE A USE CASE

- An executive at a chain of family restaurants called Brampton's Pizza and Pasta. They are looking to develop some software for in-house use at their restaurants. Their goal is to take the restaurant ordering process and make it more efficient. She is thinking that customers should be able to view the menu of the restaurant they're in, and once they're ready, place an order. She'd really like there to be a kids' page where you can see the kids' menu. Maybe there's a few games for the kids to play, but most importantly, it should be easy enough to use that kids can make an order themselves. Customers should also be able to specify any changes they'd like to make for their meal, and they should be able to list any dietary restrictions they may have before they submit their order to the kitchen. The kitchen should then be able to view these orders as they come in. Customers should be able to view and pay their bill within the system.

USE CASE : SOLUTION

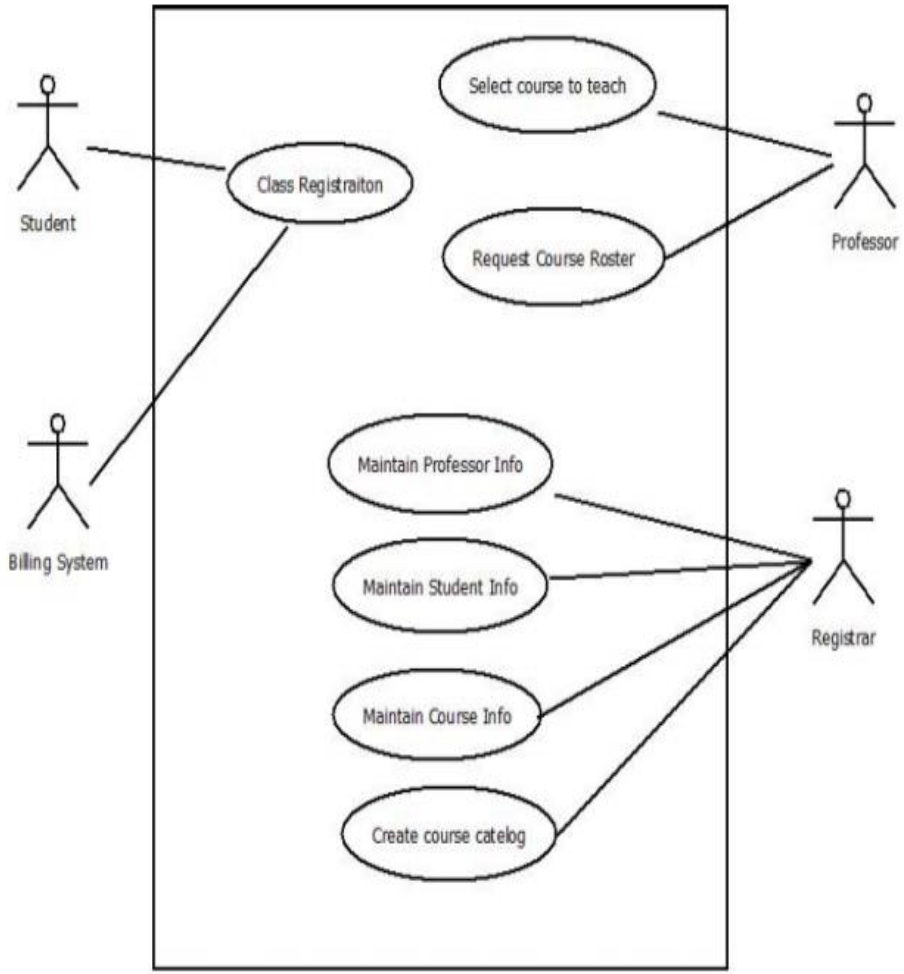
Name	View Bill
Participating Actors	Customer
Goals	View the Bill for the Order
Triggers	Request to View Bill
Pre-Condition	Menu Items on Menu, Selecting Dish, Placing Order
Post-Condition	View Bill and Pay for Bill
Basic Flow	1) User Requests to View Bill 2) User Views Bill
Alternate Flows	User Gets Wait Staff to Print and Bring Them Bill
Exceptions	2A) Display error message if no dishes were ordered
Qualities	Bill Takes Less than 10 Seconds to Load

YOUR TURN: 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.

YOUR TURN: 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.



YOUR TURN: 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.

YOUR TURN: 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.

IDENTIFY USE CASES

- **Automated Teller Machine (ATM)** is a banking system. This banking system allows customers or users to have access to financial transactions. These transactions can be done in public space without any need for a clerk, cashier, or bank teller. The user is authenticated when enters the plastic ATM card in a Bank ATM, Then enters the user name and PIN (Personal Identification Number). For every ATM transaction, a Customer Authentication is required and essential. User checks the bank balance as well as also demands the mini statement about the bank balance if they want. Then the user withdraws the money as per their need. If they want to deposit some money, they can do it. After complete action, the user closes the session. If there is any error or repair needed in Bank ATM, it is done by an ATM technician. ATM technician is responsible for the maintenance of the Bank ATM, upgrades for hardware, firmware or software, and on-site diagnosis.

CLASS ACTIVITY: IDENTIFY FUNCTIONAL REQUIREMENTS

Scenarios:

Jack, an incoming student from US, is studying computer science at TUM. He has business administration as minor subject and is already used to visit the courses in the computer science building. The business administration courses however, are located in a lecture hall in another building in the city center of Munich. He never visited the other building before, so he does not know how to find the lecture halls for his minor subject. He browses through the courses in the course catalog of the University App and finds the course "Foundations of Business Administration" with course times and the location of the lecture hall on a map. While he is attending the course, he makes contact with fellow students who also attend the course and reads their comments. He likes one comment "Great exercises" by Jenny, who is also studying informatics. From Jenny's picture, he remembers that they met a week ago at the coffee machine. He requests friendship with Jenny (she might help him to pass the final) and adds a new comment about exam questions from earlier exams. While he is browsing, Jenny is notified about the friend request and accepts it. jack, in turn, is notified that Jenny has accepted his request and now browses through all the courses that Jenny is visiting. He finds another interesting course "Cost Accounting" that he wants to visit and saves it into his course list.

That is all