

Chapter 4

Requirement Modeling

MS. Le Thanh Trong

Targets

- ❖ Understand the role of requirements modeling
- ❖ Understand the purpose and elements of a Use-case diagram as actors, use-case, communication, system boundary, relationships
- ❖ Understand and apply techniques to specify Use-case diagram elements and draw specify Use-case diagram
- ❖ Build and refine Use-case diagram

Outline

1. Review
2. Use-case Requirement Modeling
3. Use-case diagram elements
4. Refine Use-case diagrams

Outline

- 1. Review**
2. Use-case Requirement Modeling
3. Use-case diagram elements
4. Refine Use-case diagrams

Review

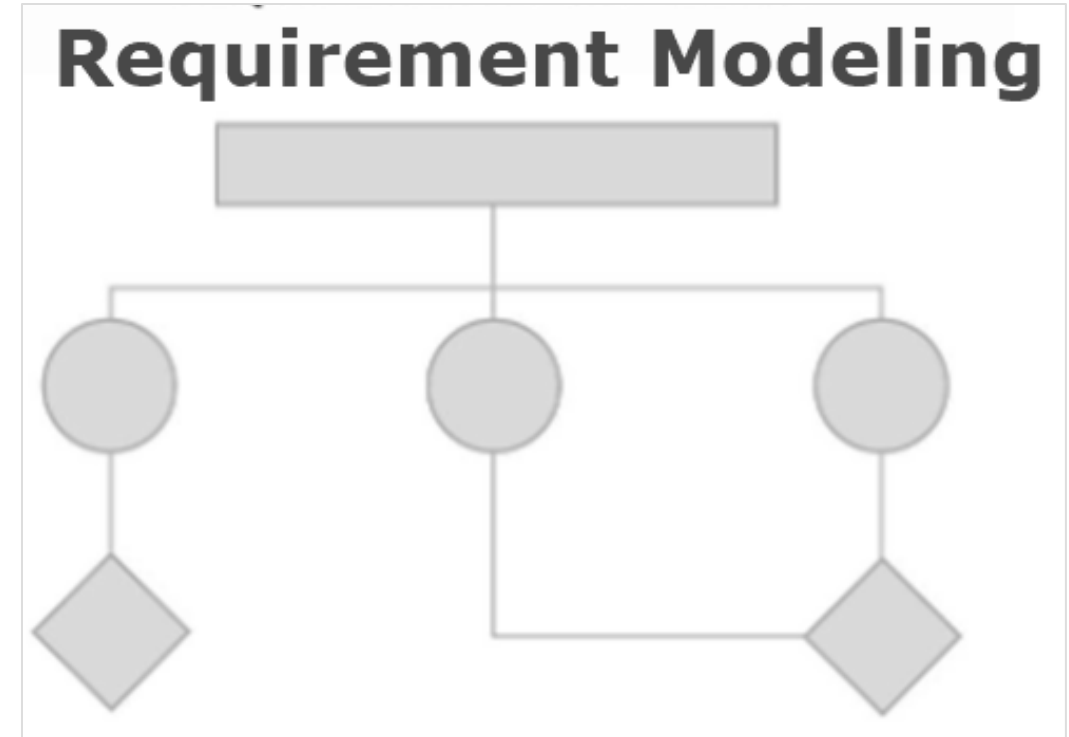
- ❖ Analysis phase can be broken into to two processes:
 - Requirement gathering
 - Modeling the requirements
- ❖ Requirement gathering is an process that involves researching and documenting the project's exact requirements from start to finish
- ❖ Requirement gathering starts at the beginning of the project
- ❖ Efficiency requirement gathering is a need for building a successful system

Review

❖ Various techniques of requirement gathering

- Interview
- Questionnaire
- Analyzing Existing Documents
- User Observation
- Prototyping
- Group Analysis Design (JAD)
- ...

❖ What is the next step ?

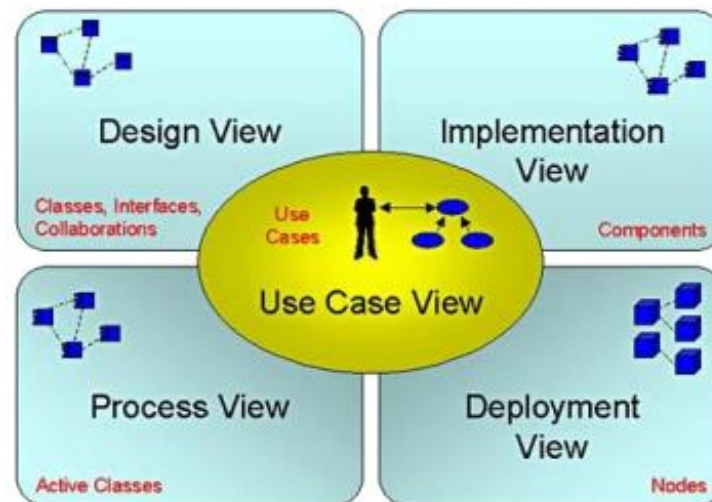


Outline

1. Review
- 2. Use-case Requirement Modeling**
3. Use-case diagram elements
4. Refine Use-case diagrams

Requirement Modeling

- ❖ Descriptions of requirements in the requirement gathering process:
 - Only describes mainly information related to the performance of real-world operations, not clearly showing the performance of operations on computers
 - Over-describing the text is confusing and unintuitive



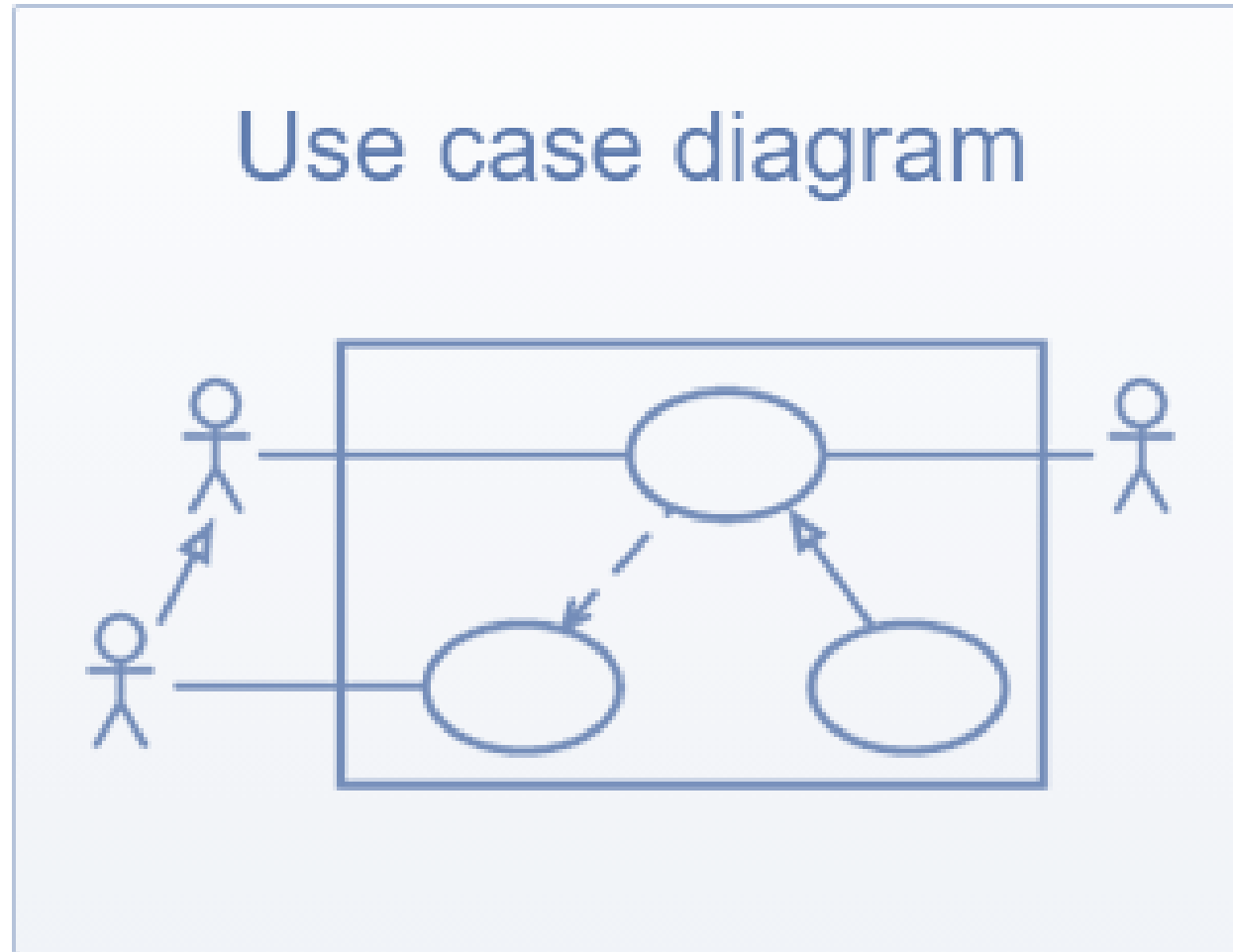
Use-case model

- ❖ Is a model of how different types of **users** interact with the **system** to solve a problem
- ❖ Consists of Use-case (UC) diagrams depicted in UML and use case descriptions. Basic elements in UC diagram:
 - Actors
 - Use-cases
 - Communication
 - System boundary
 - Relationships

Use-case Diagram

- ❖ A primary UML diagram used in business modeling
- ❖ Specify the context of a system
- ❖ Defining and organizing functional requirements in a system
- ❖ As the basis for analysis and design
- ❖ Validate a systems architecture
- ❖ As the basis of defining test cases and as the basis for user documentation
- ❖ Handle frequent changes is to maintain a list of requirements

Use-case Diagram

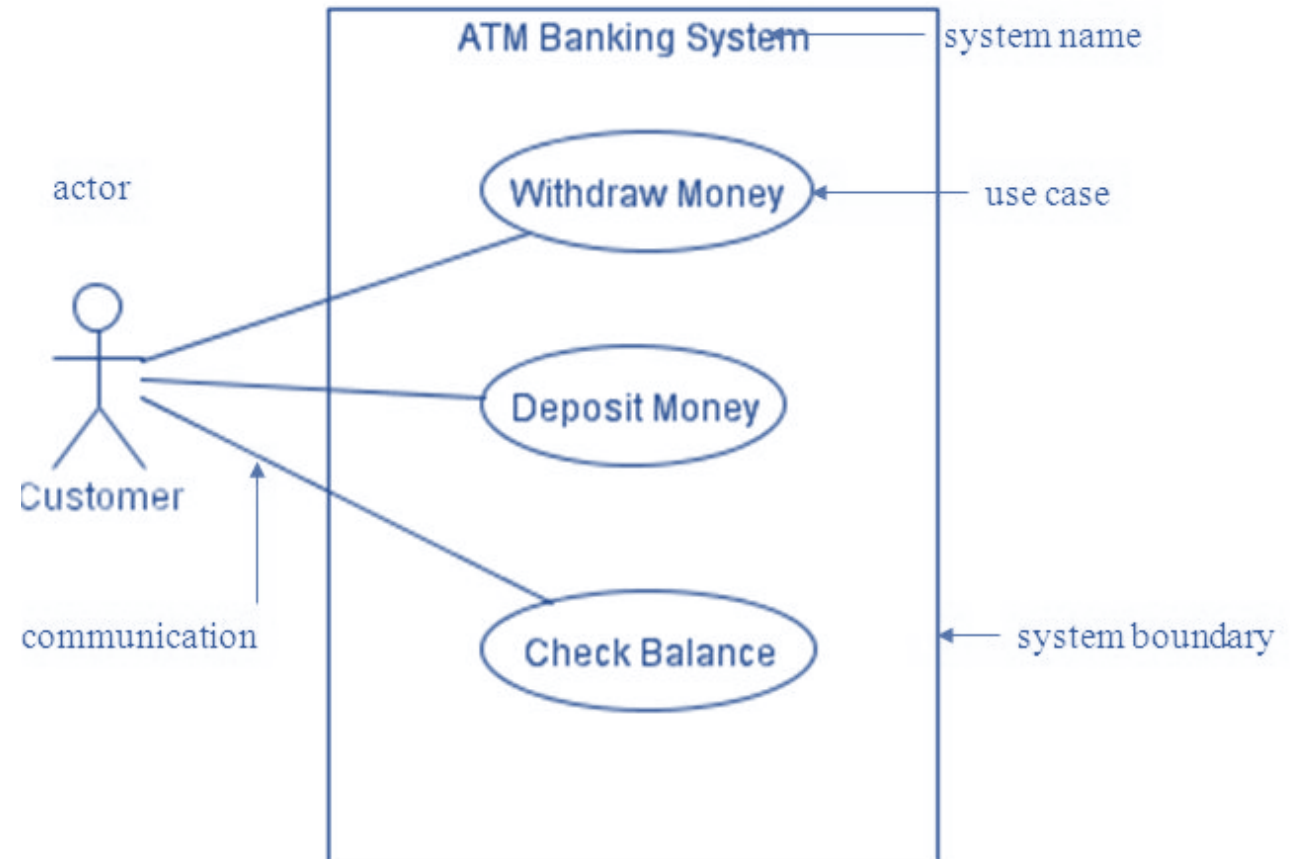


Outline

1. Review
2. Use-case Requirement Modeling
- 3. Use-case diagram elements**
4. Refine Use-case diagrams

Use-case diagram elements

- ❖ Actors
- ❖ Use-cases
- ❖ Communication
- ❖ Relationships
- ❖ System boundary



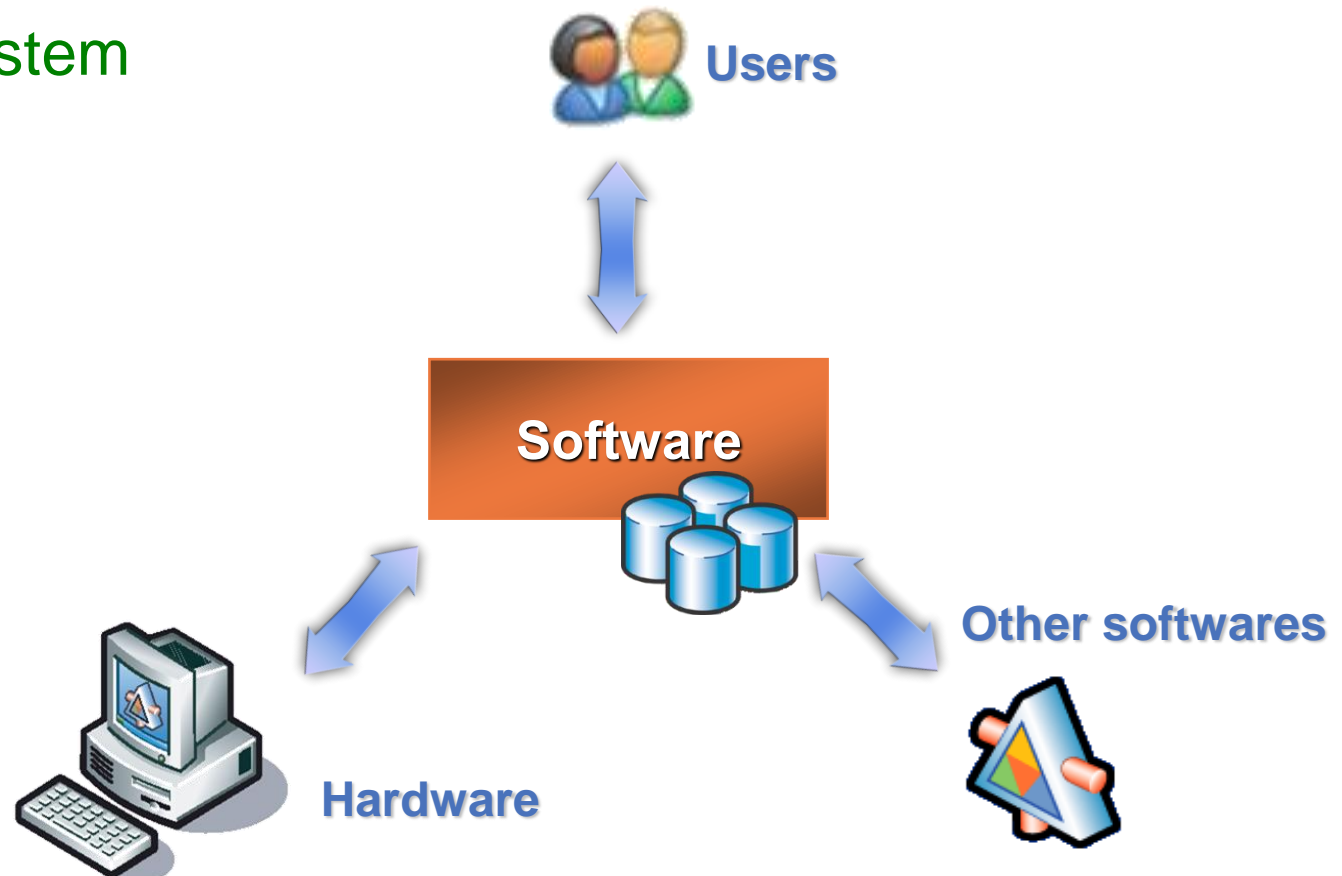
Use-case diagram elements

- ❖ **Actors**
- ❖ Use-cases
- ❖ Communication
- ❖ Relationships
- ❖ System boundary

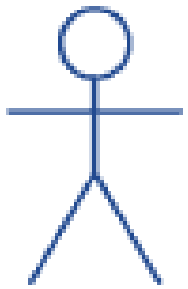
Actor

❑ **EXTERNAL**

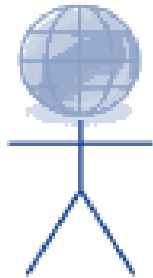
❑ **Interacts** with the system



Actor notation



Student



Web Client

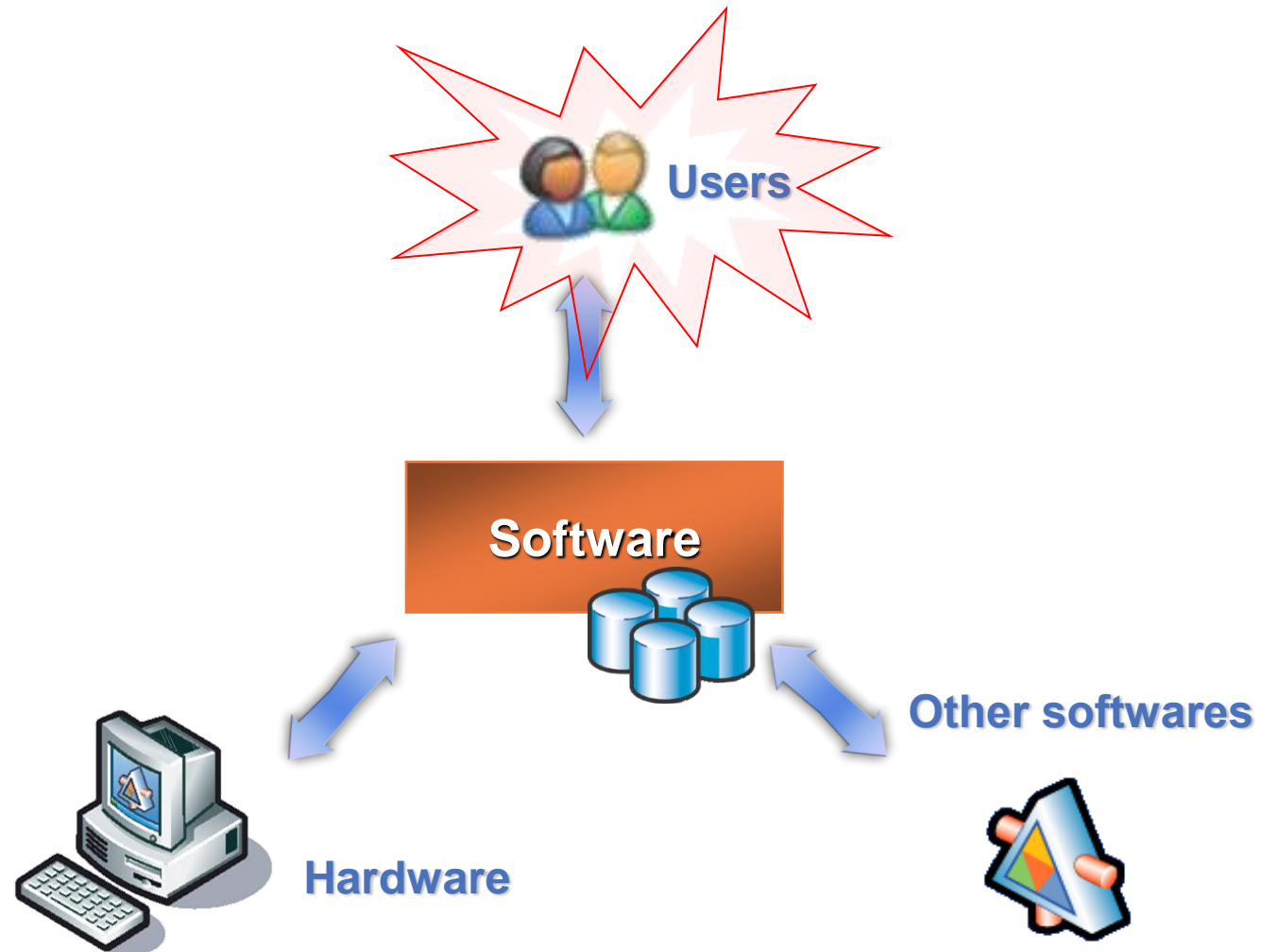


Bank

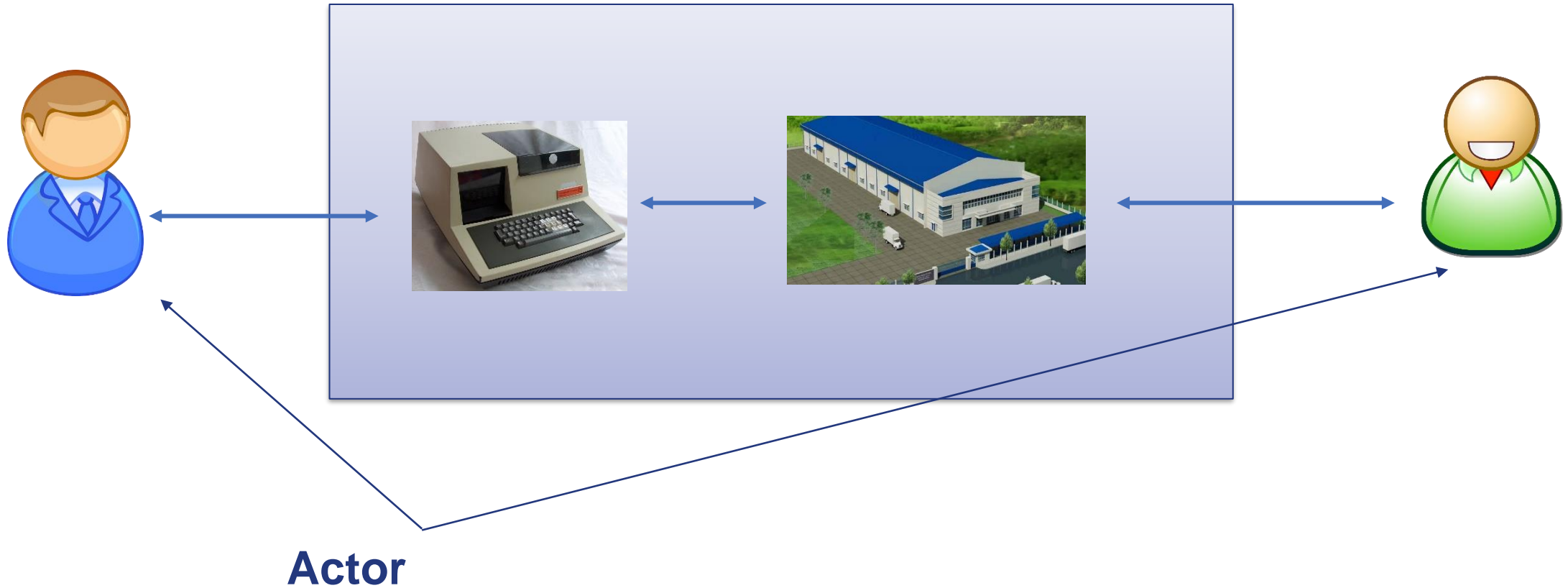
**«actor»
Customer**

+ name: Name
+ address: Address

Actor \Leftrightarrow Users



Actor \Leftrightarrow Users



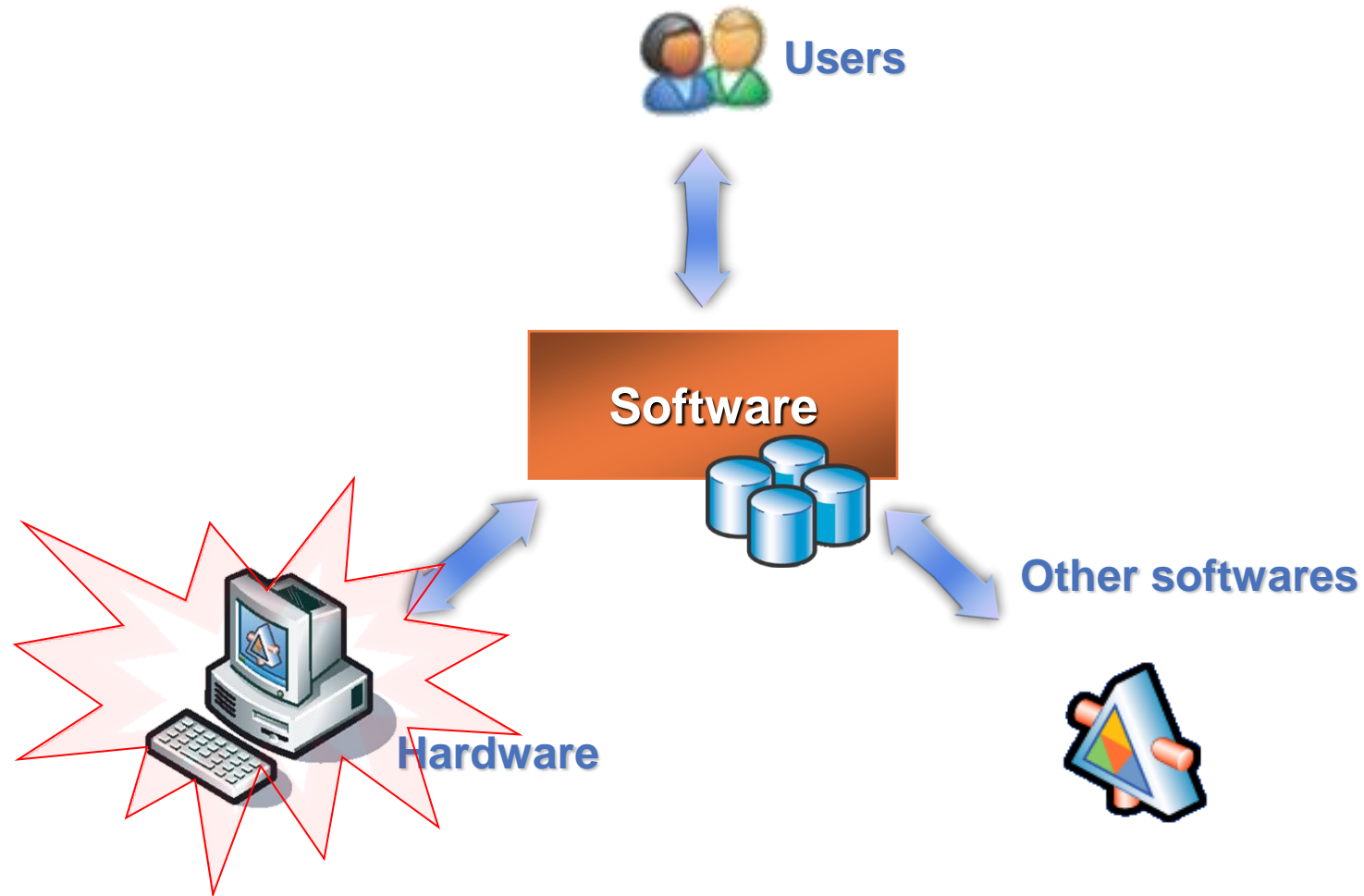
Example

No	Requirements		Users
1	Student enrolling	↔	Officer?
2	Make a class list	↔	Officer?
3	Look up students	↔	All? Parents? Student?
4	Get score sheet	↔	Teacher? Officer?
5	View summary report	↔	President?
6	Change rules	↔	President? System Administrators?

Student Management System

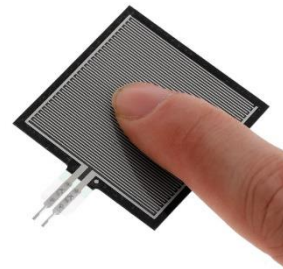


Actor \leftrightarrow Hardware

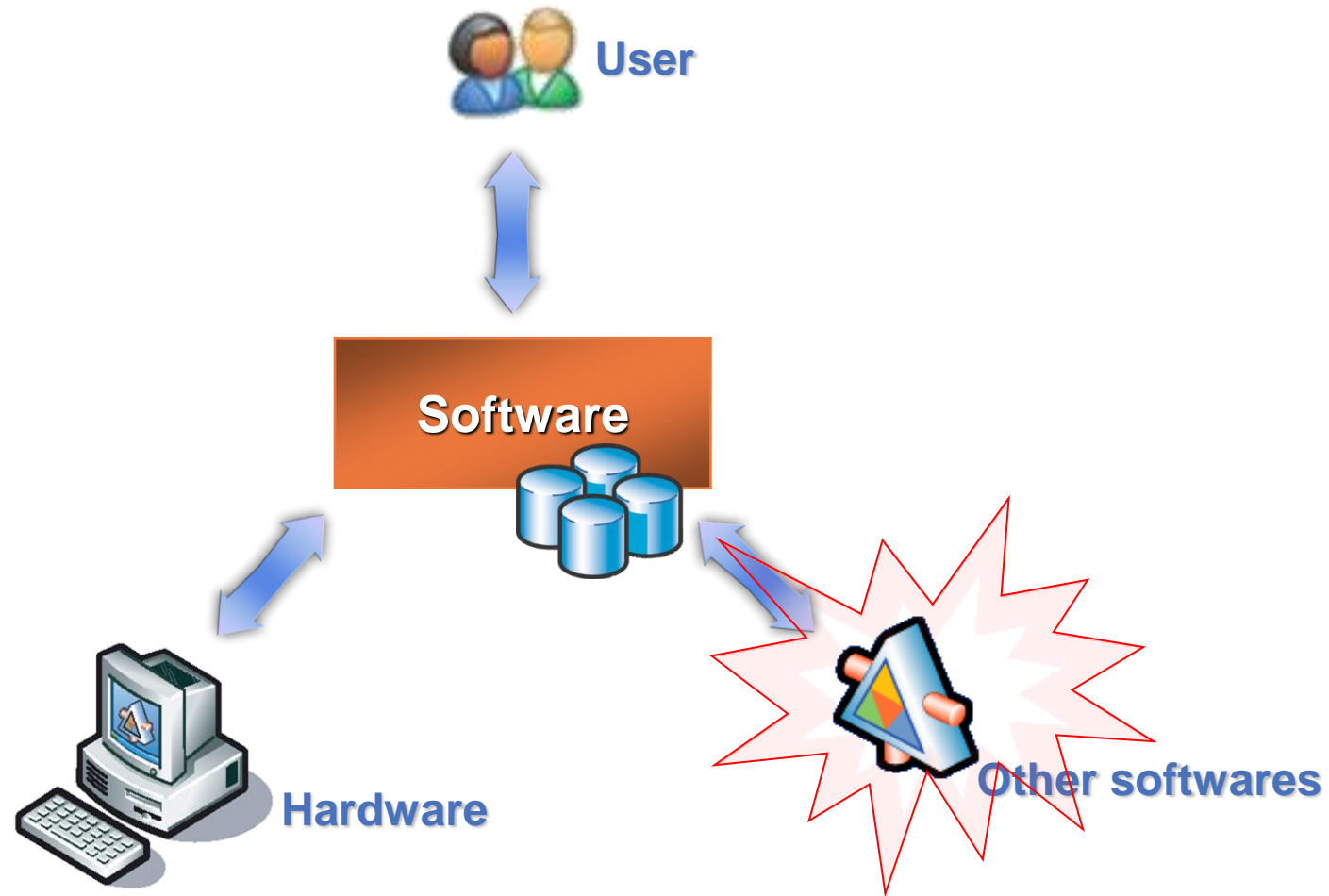


Hardware example

- ❖ Supermarket management software:
 - Read information from **barcode reader**
- ❖ Automatic door management software:
 - Read information from the **camera**
 - Control command to open the door
- ❖ Software to manage access to rooms in the office
 - Read signal from **magnetic card reader**
 - Control command to open the door
- ❖ Anti-theft software
 - Read signal from **camera, sensor**
 - Issue control commands to **speakers, lights, phones...**

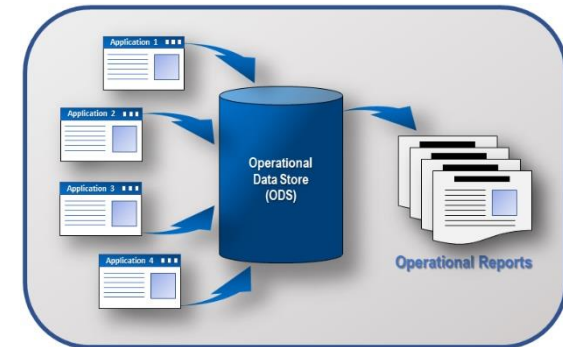


Actor ⇔ Other softwares



Softwares example

- ❖ Export/load data from Excel or database
- ❖ Export report data to email software (Microsoft Outlook, Outlook Express...)
- ❖ Cloud system
- ❖ Other services
- ❖ ...



Type of Actors

❖ Primary Actors

- Stakeholder that calls on the system to deliver one of its services
- Has a goal with respect to the system – one that can be satisfied by its operation
- The primary actor is often, but not always, the actor who triggers the use case

❖ Secondary Actors

- An external that provides a service to the system
- May not have goals that they expect to be satisfied by the use case

Type of Actors

Description

*A **user** clicks the search button on an application's user interface. The application sends an SQL query to a **database system**. The database system responds with a result set. The application formats and displays the result set to the user*

The **user** is a **primary actor** because he initiates the interaction with the system

The **database system** is a **secondary** actor because the application initiates the interaction by sending an SQL query

Actor Role Description

- ❖ Brief description that includes the actor's area of responsibility, and what the actor needs the system for
- ❖ Make list all your actors with their **role** description and their **objectives** in a tabular format

Actor / Role Name	Role Description and Objective
Customer	Customers will place food orders and may or may not order juice. When the order is served, the customer will eat his/her meal and pay the check.
Waiter	The waiter will receive the food order from the customer and confirm the order with the cook and serve food to the customer.

Identifying actors questions

- ❖ Who **uses** the system?
- ❖ Who **installs** the system?
- ❖ Who **starts up** the system?
- ❖ Who **maintains** the system?
- ❖ Who **shuts down** the system?
- ❖ What **other systems use** this system?
- ❖ Who **gets information** from this system?
- ❖ Who **provides information** to the system?
- ❖ Does anything **happen automatically** at a present time?

Identifying actors examples



Customer

ATM System



Librarian

Library Management System

The group requires the system's help

Identifying actors examples



Operating staff

ATM System

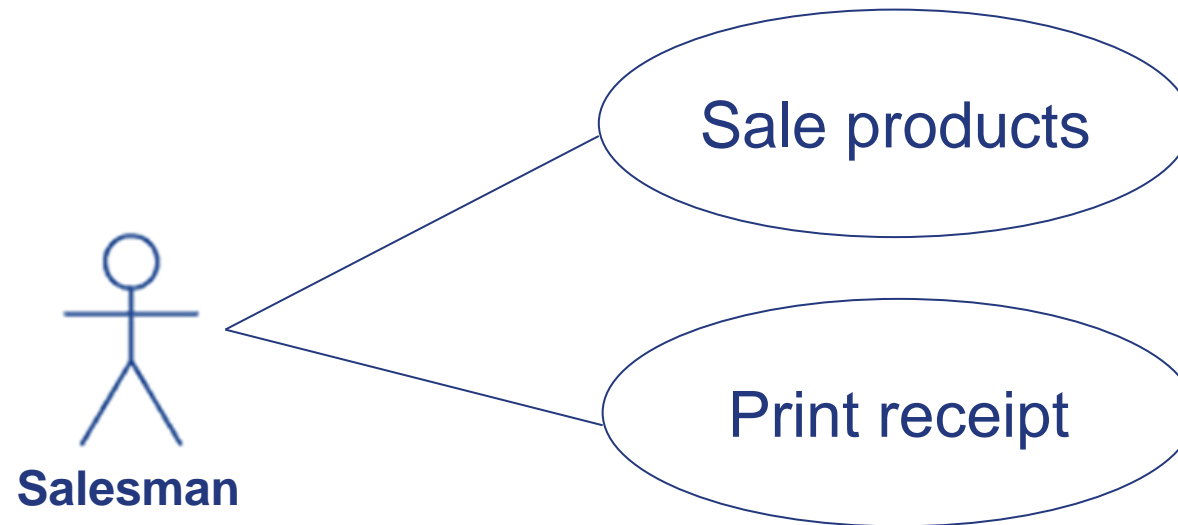


System Administrator

Library Management System

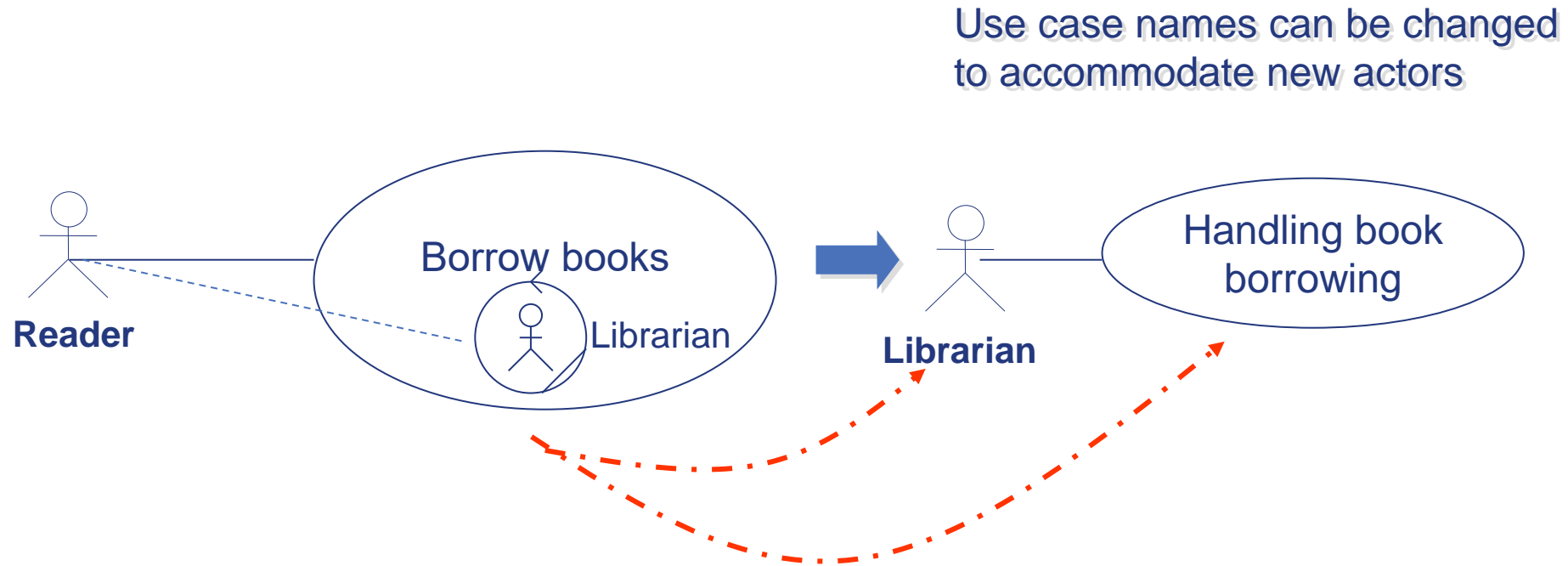
The group needed to perform the system's functions

Identifying actors examples



Group of devices or software systems related to the system

Identifying actors examples



Library Management System

Identify actors from the results of business modeling

Use-case diagram elements

- ❖ Actors
- ❖ **Use-cases**
- ❖ Communication
- ❖ Relationships
- ❖ System boundary

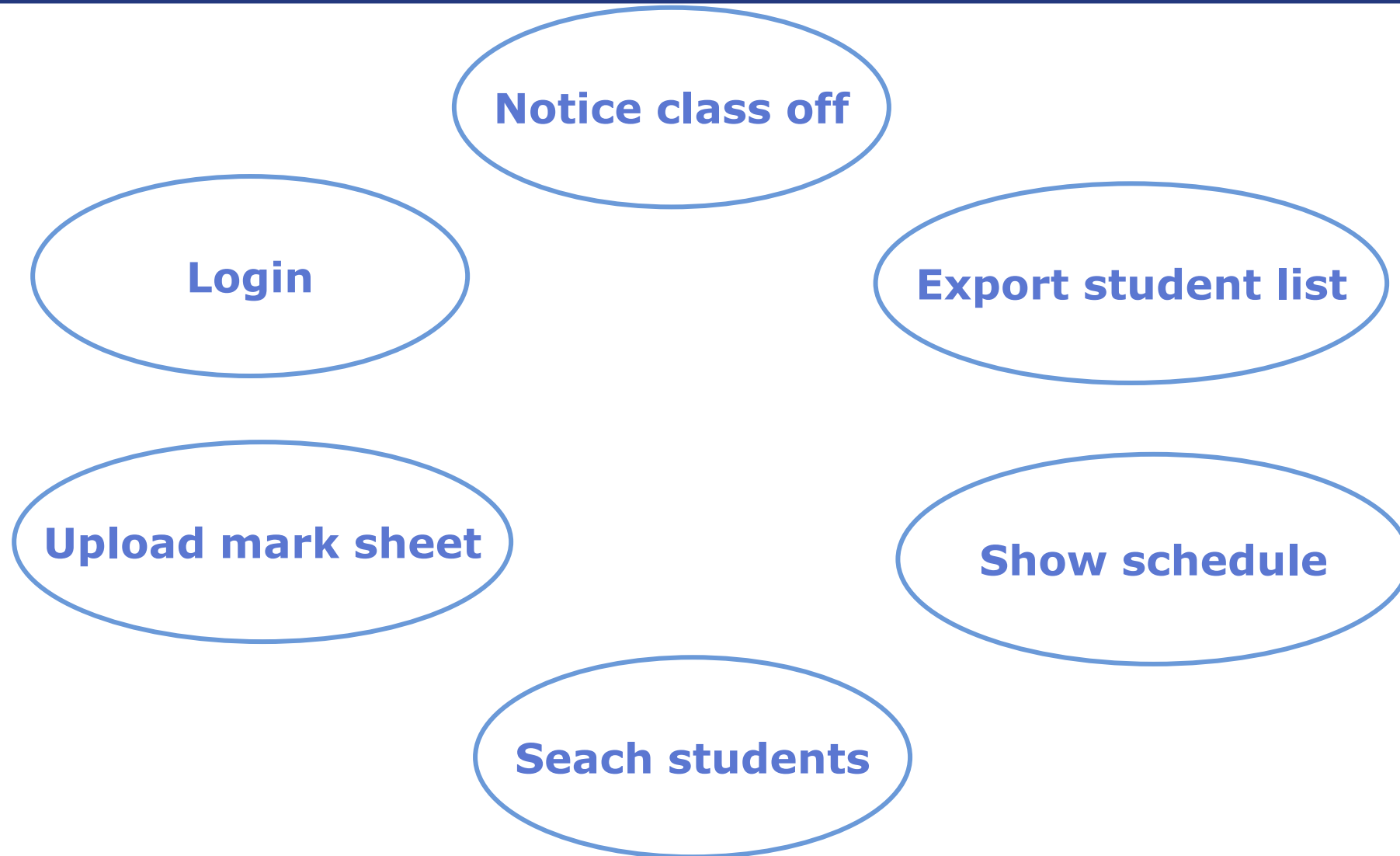
Use-case

- ❖ Is a sequence of actions a system performs
- ❖ Use-cases works as a contract between the end users and the developers
- ❖ Usually named by verb + [noun] (or noun Phrase)



[Use-case name]

Example



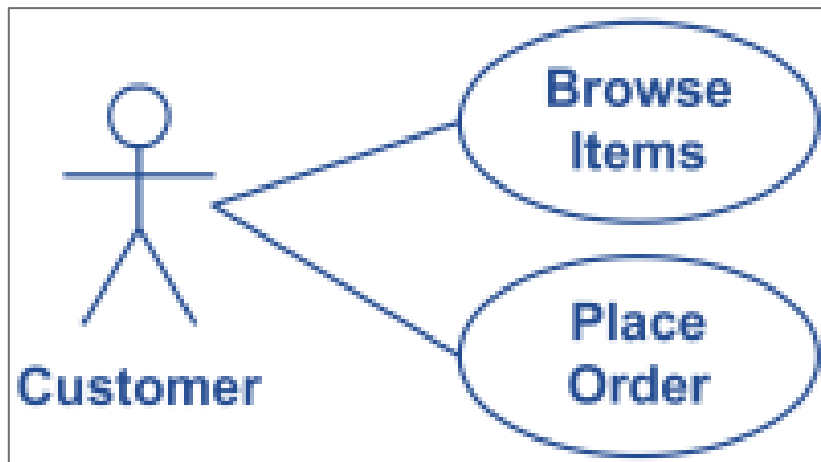
Identifying use case questions

- ❖ What **functions** will the actor want from the system?
- ❖ Does the system **store information**? What actors will create, read, update or delete this information?
- ❖ Does the system need to **notify** an actor about changes in the internal state?
- ❖ Are there any **external events** the system must know about? What actor **informs the system** of those events?

Use-case diagram elements

- ❖ Actors
- ❖ Use-cases
- ❖ **Communication**
- ❖ Relationships
- ❖ System boundary

Association between Actor and Use-case

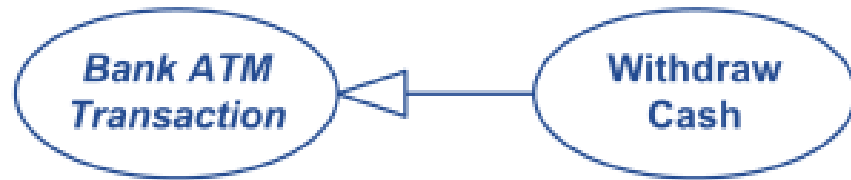


Use-case diagram elements

- ❖ Actors
- ❖ Use-cases
- ❖ Communication
- ❖ **Relationships**
- ❖ System boundary

Use-case Relationships

Generalization



Extend

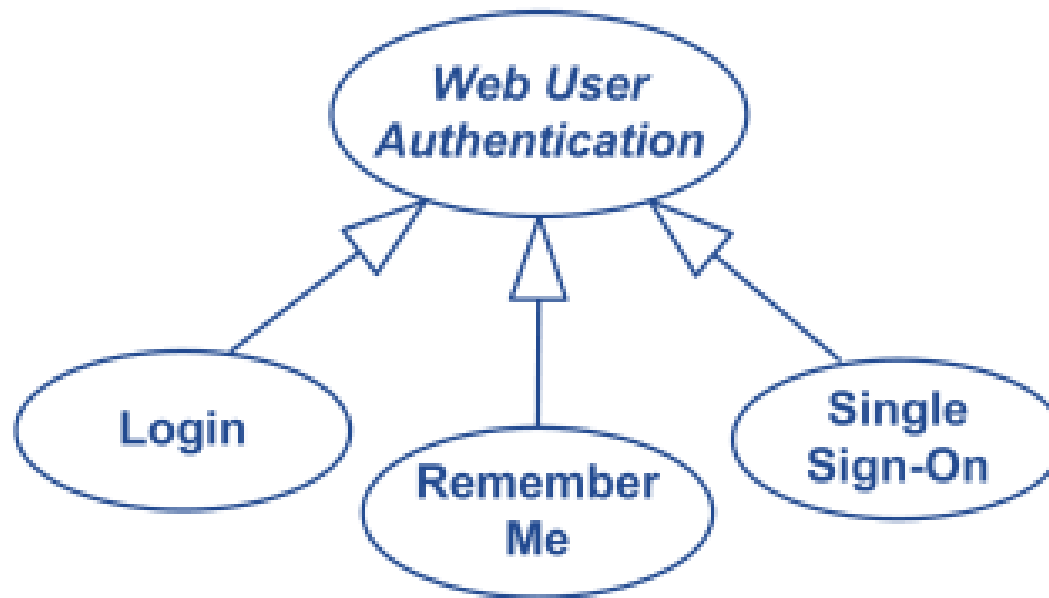


Include



Generalization between Use-cases

- ❖ Similar to generalization between classes
- ❖ Is shown as a solid directed line with a large hollow triangle arrowhead

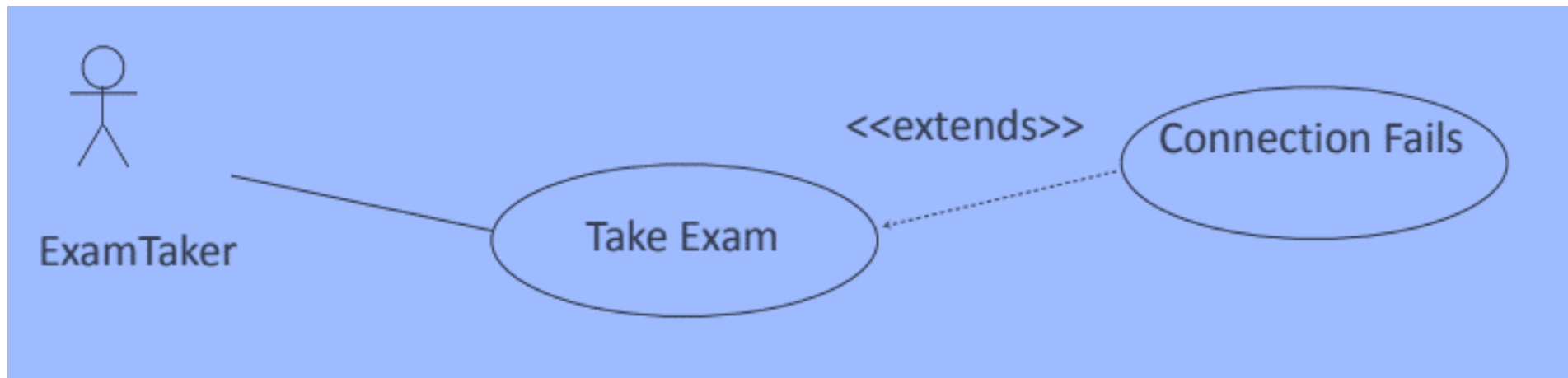


Use-case Extend

- ❖ Behavior defined in usually supplementary (optional) extending, is used for exceptional conditions
- ❖ Is shown as a dashed line with an open arrowhead directed from the extending use case to the extended (base) use case
- ❖ The arrow is labeled with the keyword **«extend»**

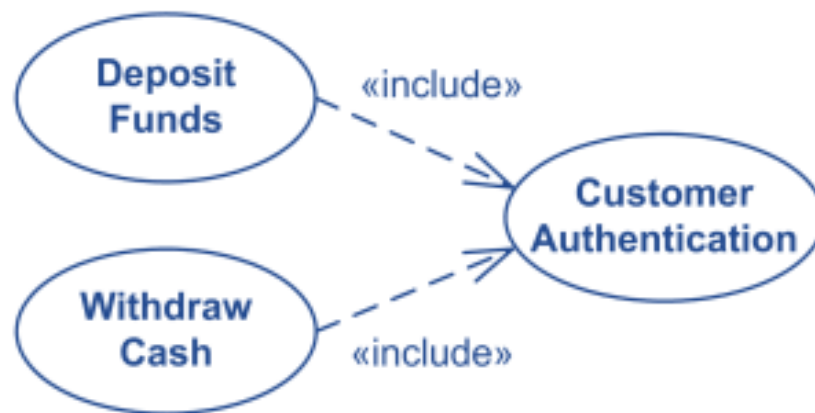


Use-case Extend



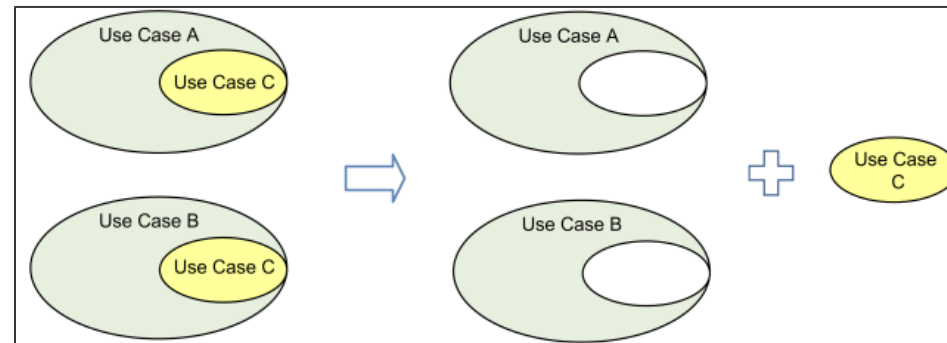
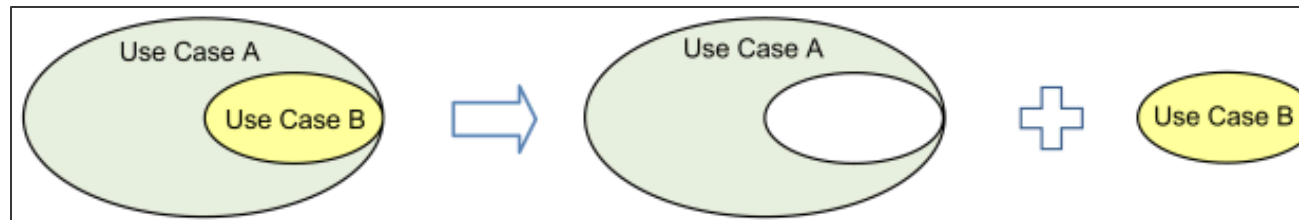
Use-case Include

- ❖ Show that behavior of the included use case (the addition) is inserted into the behavior of including (the base) use case
- ❖ Is shown by a dashed arrow with an open arrowhead from the including (base) use case to the included (common part) use case
- ❖ The arrow is labeled with the keyword **«include»**

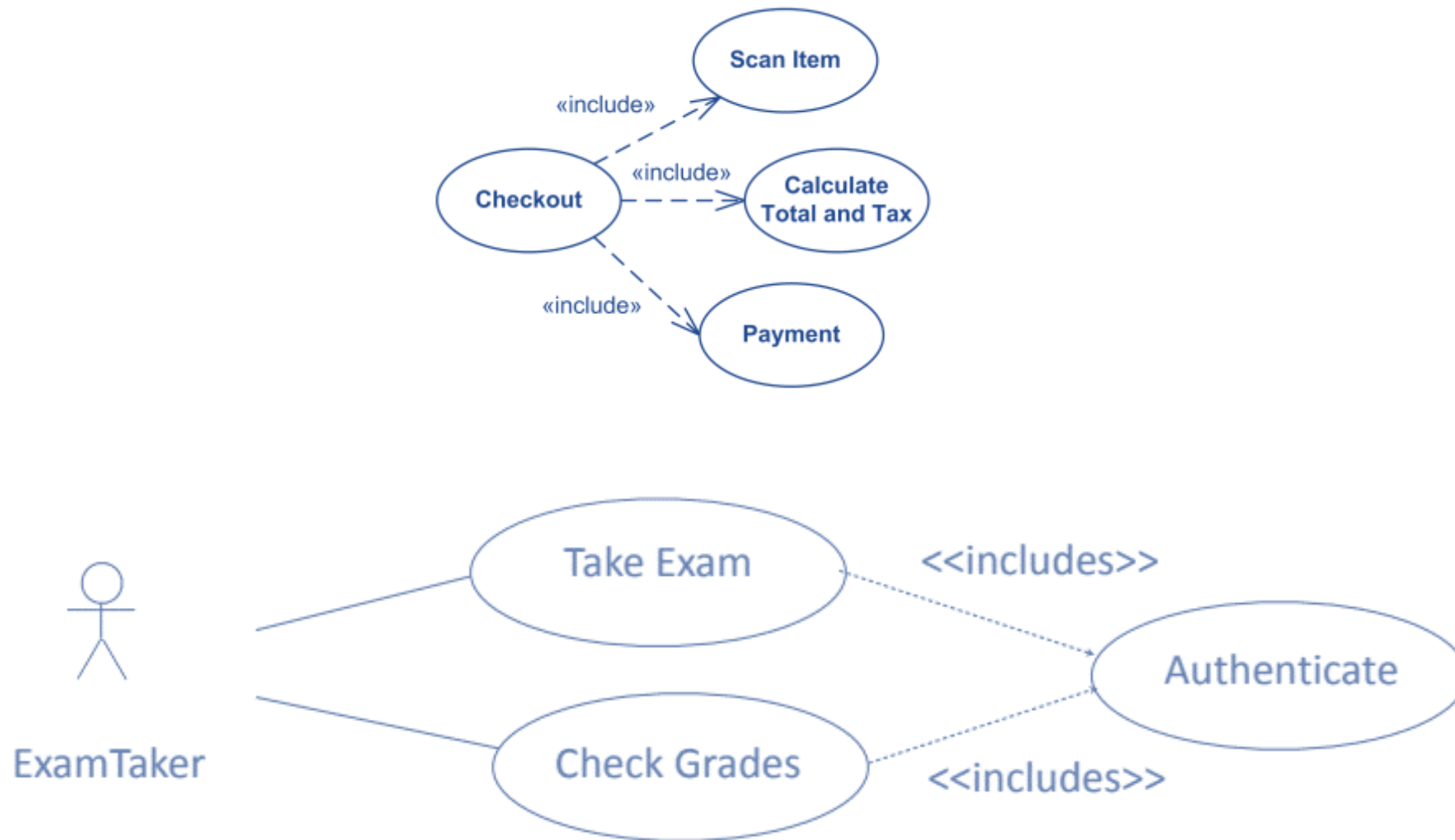


Use-case Include

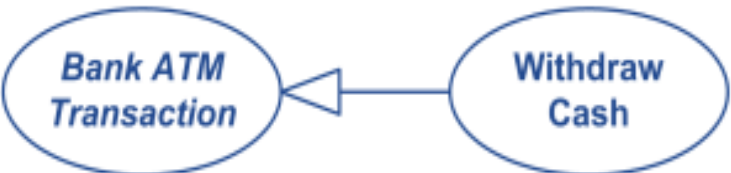
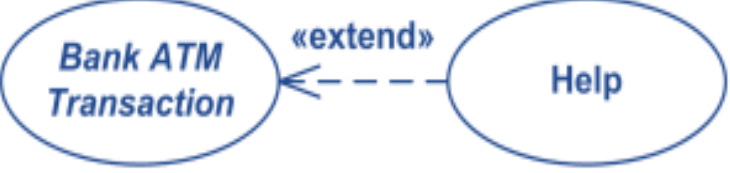

- ❖ The include relationship could be used:
 - To simplify large use case by splitting it into several use cases,
 - To extract common parts of the behaviors of two or more use cases



Use-case Include



Use-case Relationships Compared

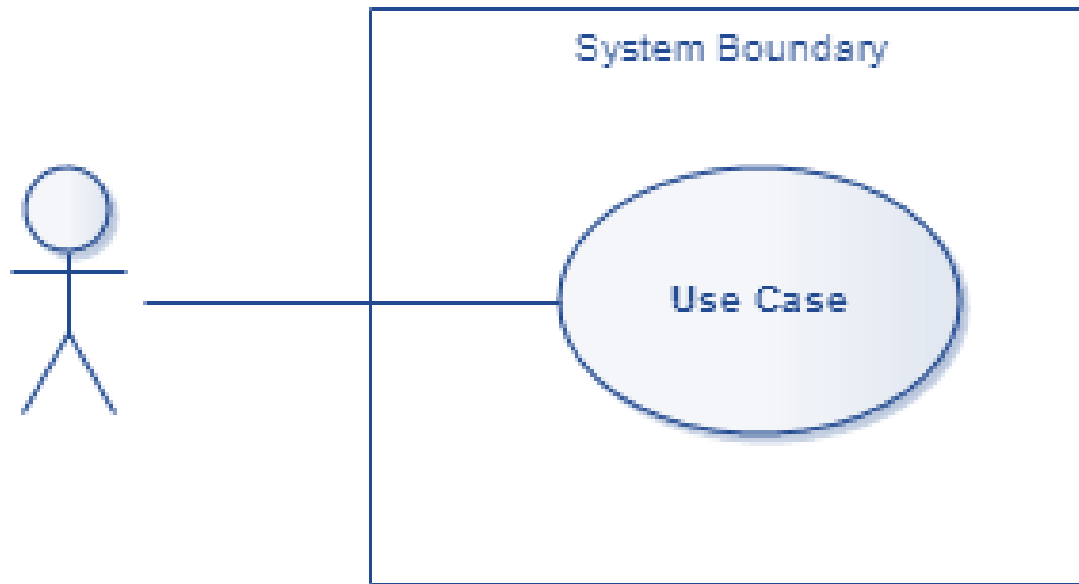
Generalization	Extend	Include
		
Base use case could be abstract use case (incomplete) or concrete (complete).	Base use case is complete (concrete) by itself, defined independently.	Base use case is incomplete (abstract use case).
Specialized use case is required, not optional, if base use case is abstract.	Extending use case is optional, supplementary.	Included use case required, not optional.
No explicit location to use specialization.	Has at least one explicit extension location.	No explicit inclusion location but is included at some location.
No explicit condition to use specialization.	Could have optional extension condition.	No explicit inclusion condition.

Use-case diagram elements

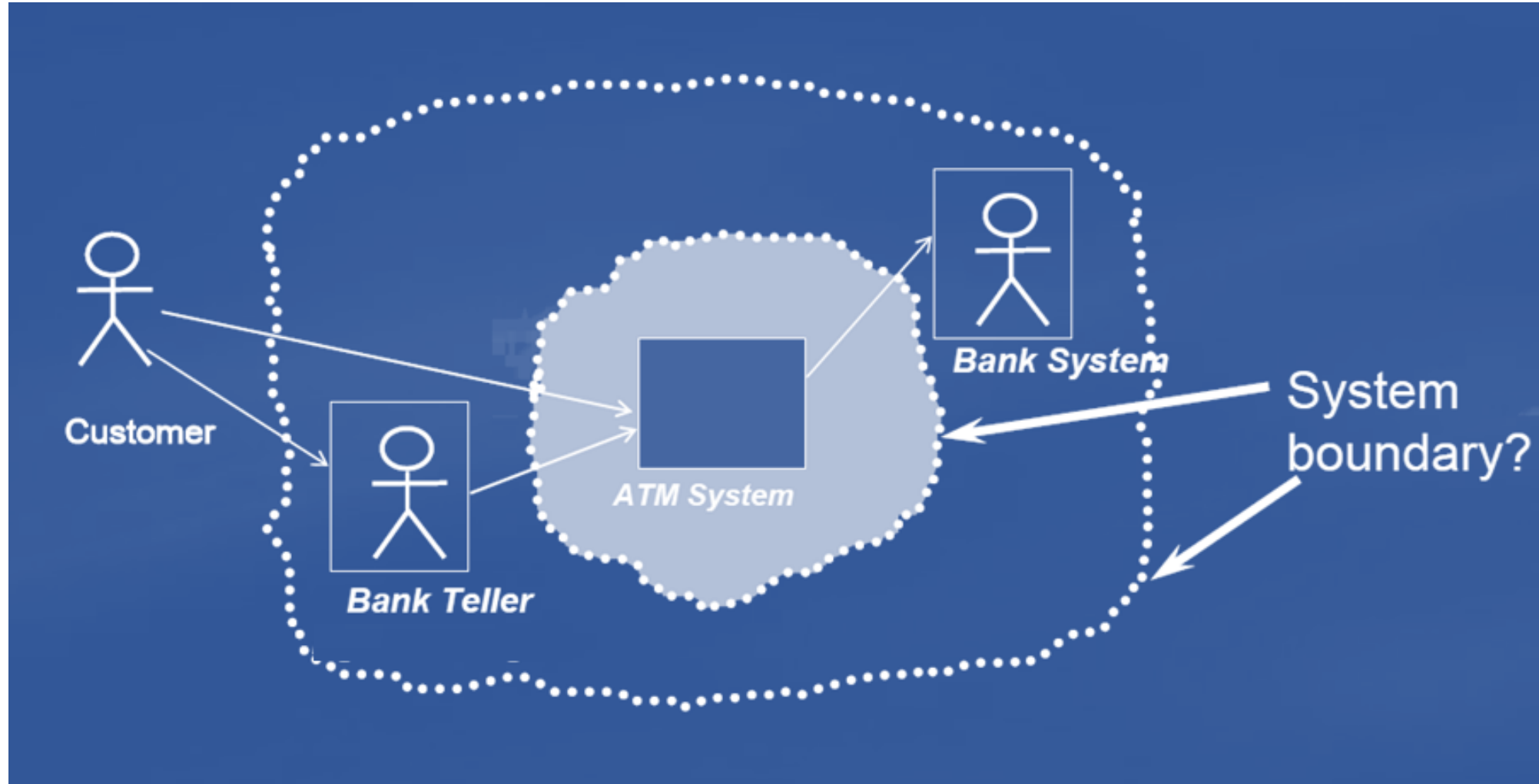
- ❖ Actors
- ❖ Use-cases
- ❖ Communication
- ❖ Relationships
- ❖ **System boundary**

System Boundary

- ❖ To indicate the application of a Use-case to another entity
- ❖ Help group logically related elements
- ❖ Shown as a rectangle with a name



System Boundary example



Outline

1. Review
2. Use-case Requirement Modeling (RM)
3. Use-case diagram elements
- 4. Refine Use-case diagrams**

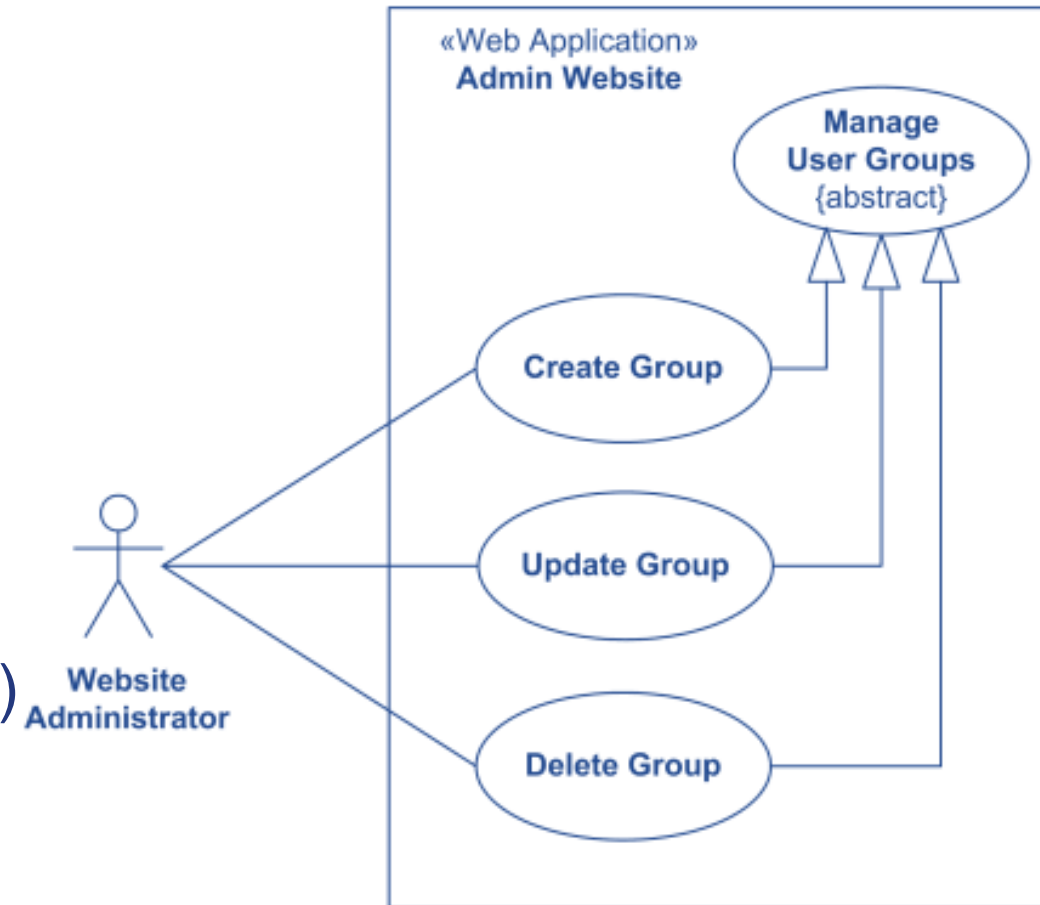
Refine Use-case diagrams

- ❖ Add use cases that describe software-specific functionality
- ❖ Develop <<extend>> relationship
- ❖ Develop <<include>> relationship

Refine Use-case diagrams

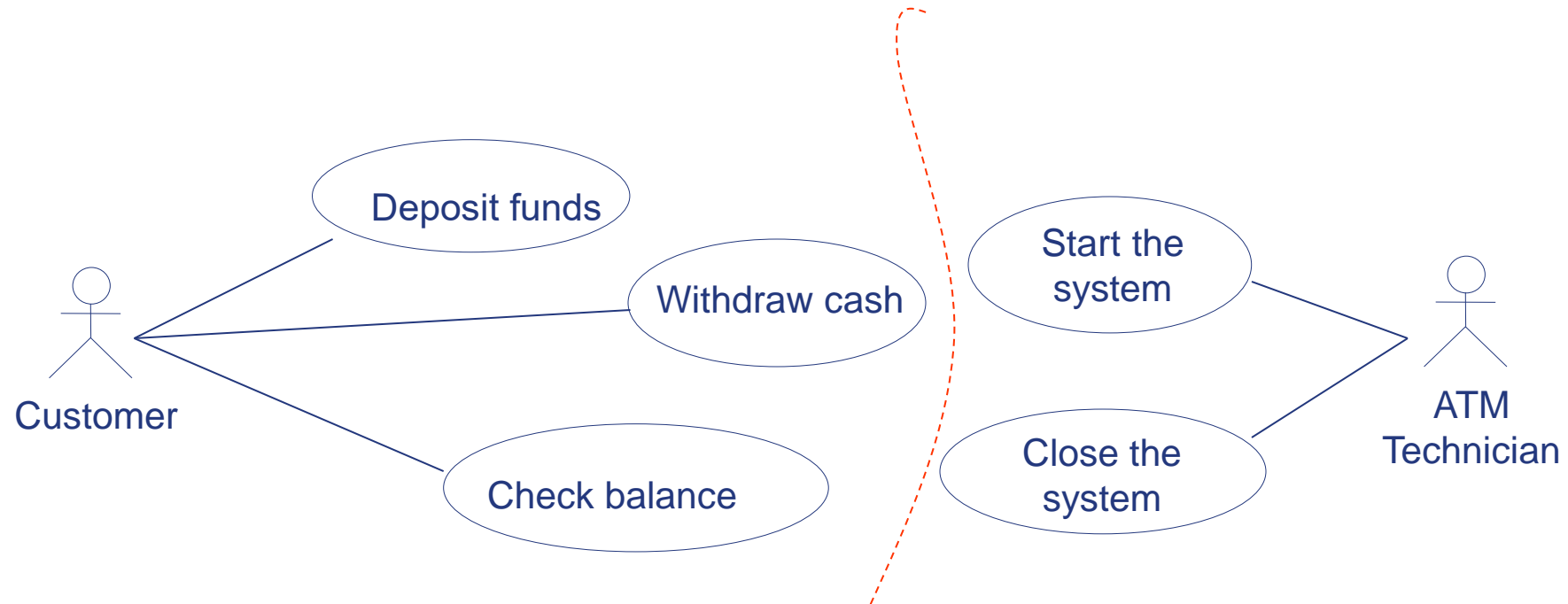
❖ Add use cases that describe software-specific functionality

- System administration use cases:
 - User Administration
 - System parameter management
- Data management use cases
 - Data catalog management
 - Secure administration (backup/restore)



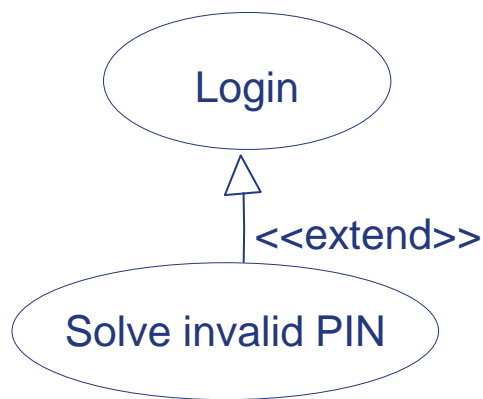
Refine Use-case diagrams

- ❖ Add use cases that describe software-specific functionality

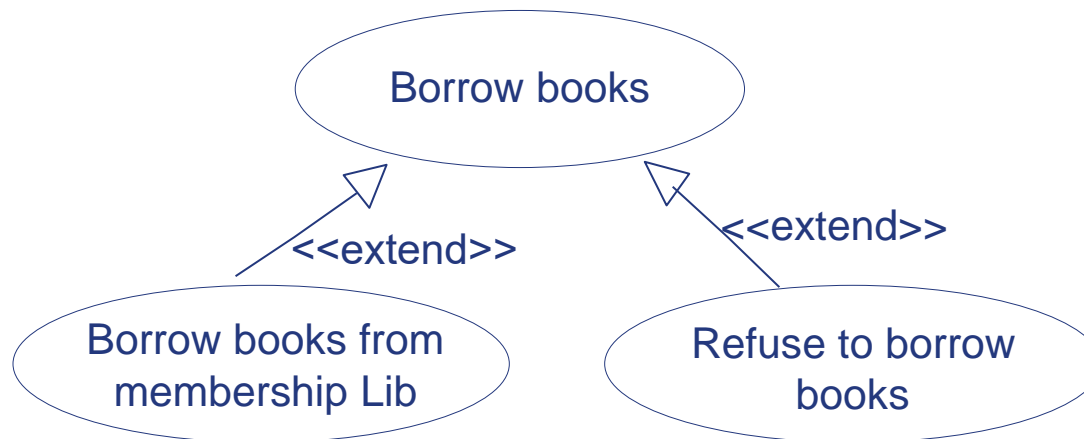


Refine Use-case diagrams

- ❖ Develop <<extend>> relationship
 - Separation of special handling lines extended use case



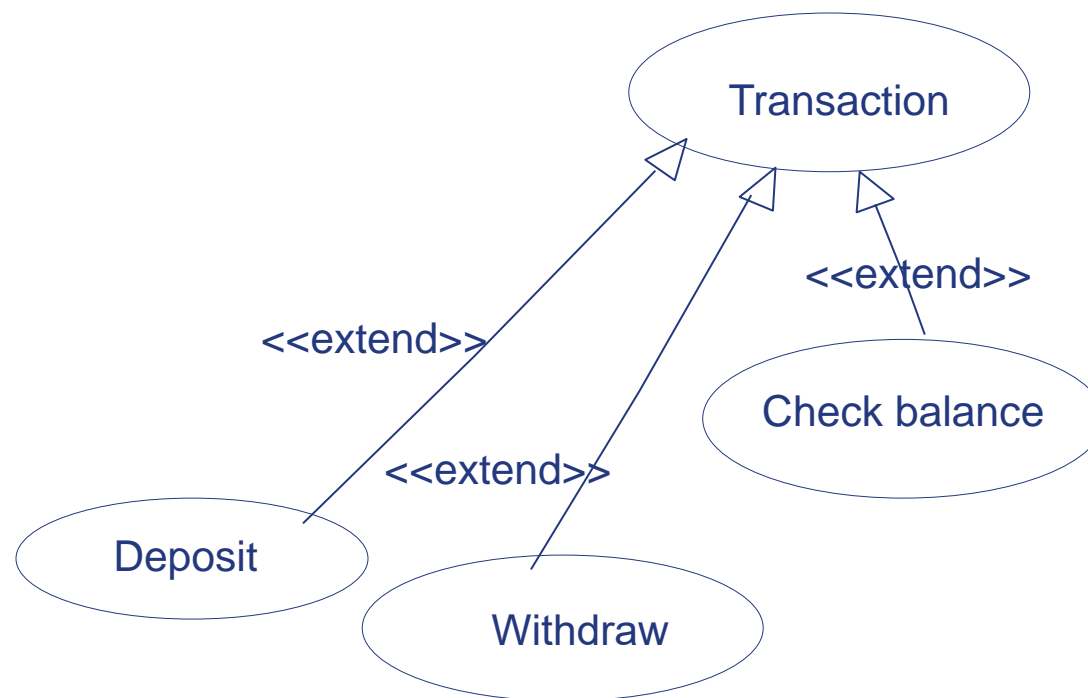
ATM System



Library System

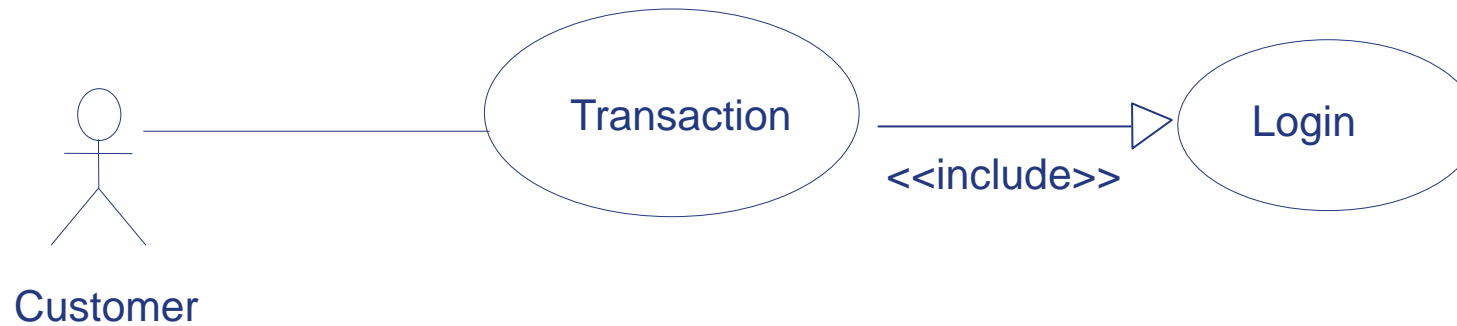
Refine Use-case diagrams

- ❖ Develop <<extend>> relationship
 - Generalize use cases with common flow to generalized use cases



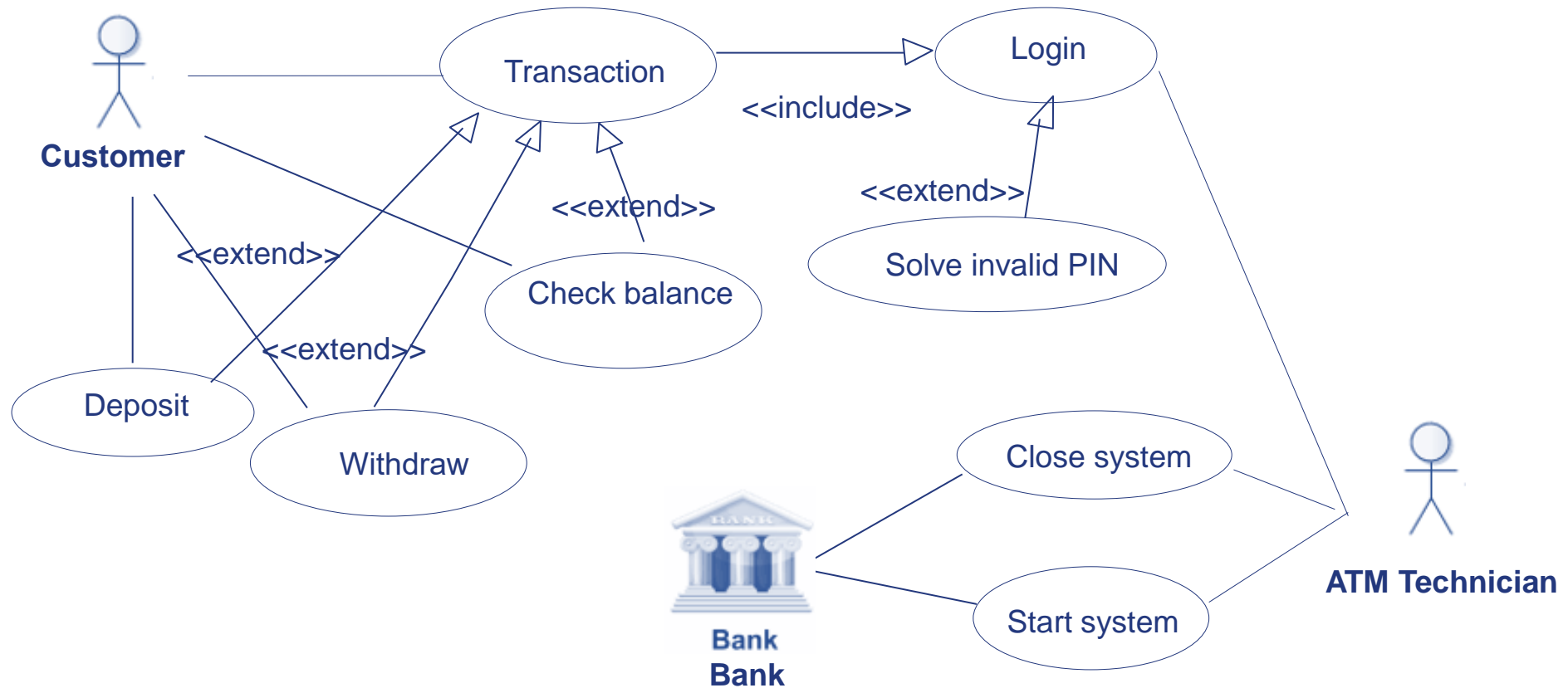
Refine Use-case diagrams

❖ Develop <<include>> relationship

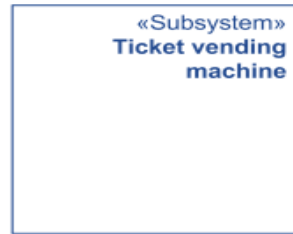


ATM System

Draw UML Use-case Diagram



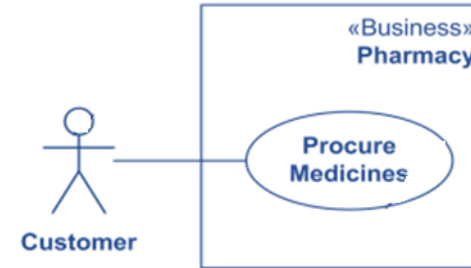
Next steps?



1. Define Subject



2. Define Actors



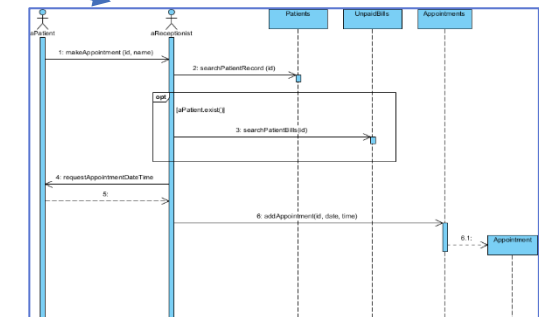
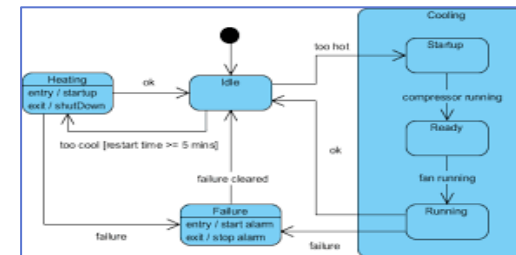
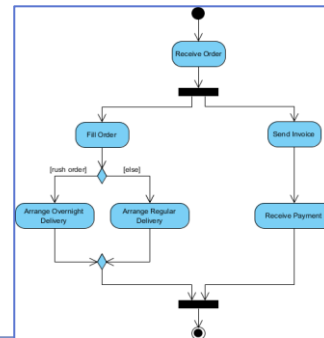
3. Define Use-cases and draw diagram

5. DESCRIBE USE CASE BEHAVIORS

4. Refine Use-case diagram

Use Case Description Example - login

Use Case ID:	ACC UC 1
Use Case Name:	login
Created By:	Joe Blogs
Date Created:	1-1-2012
Description:	This use case allows user to login into the system to access the relevant functions according to the user's role. The various user roles are staff, admin staff, system administrator, manager and department head. To login to the system, all users have to enter their unique staffid which is their NRIC number. The users have a maximum of 3 attempts to login after which their account are locked and they will have to contact the system administrator to unlock their account upon successful login the system will display the relevant user's home page.
Primary Actor:	User
Secondary Actor:	None
Preconditions:	1. User has to have a valid account
Postconditions:	1. The system displays the relevant homepage



Use-case description

Brief Description: A short description of the goals the use-case and when the use-case will take place

Basic flow: normal events and activities of the use case as expected

Alternative flow: unusual use case events and activities in addition to the main activities (unexpected)

Preconditions: describes the state of the system that must be reached before the use case starts

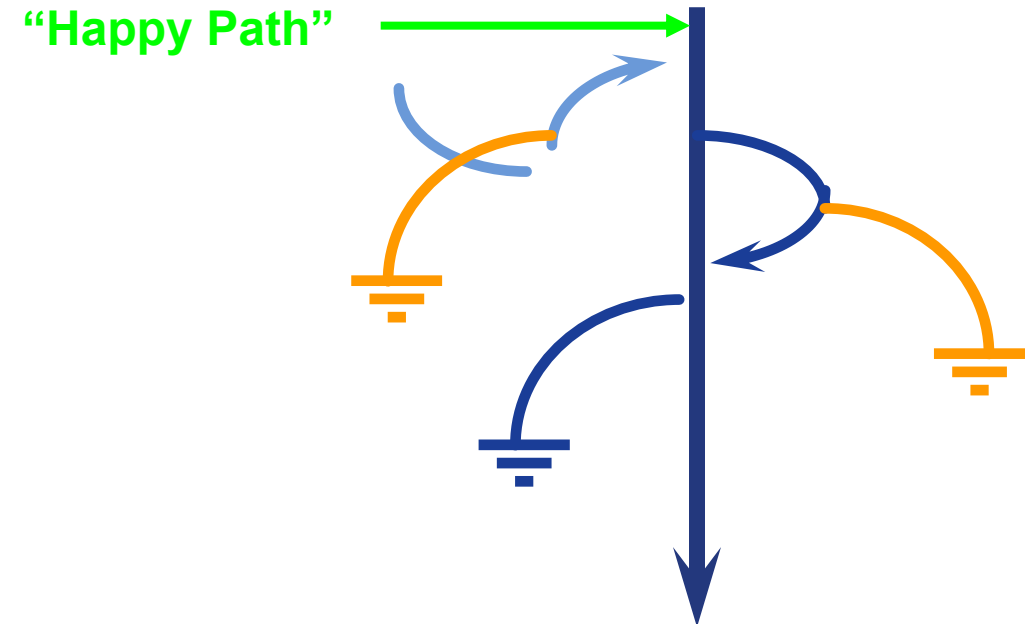
Postconditions: List the possible states of the system at the end of the use case. The system must be in one of those states when the use case ends

Use-case description

❖ **Basic flow** (“Happy Path”)

❖ **Alternative flows**

- Regular variants
- Odd cases
- Exceptional flows



Use-case description example

Title and brief description: **Borrow books;** The use case begins when a reader comes to borrow a book. The goal of the use case is to handle borrowing books for readers

Basic flow:

1. Identify reader library cards: staff ask the reader to present library card for checking
2. Determining book debt information: check information about books owed by the reader
3. Record borrowing information: update the reader's borrowing information
4. Send books and borrowing receipt to the reader

Alternative flows:

- ❑ Handling expired cards: if a reader's student card expires, the librarian will notify the reader and request a new card
- ❑ Handling non-lending (rejection): if the number of books that readers are borrowing > 3, librarians will refuse the borrowing

Preconditions:

- ❑ The system can scan the code of the reader card and the book
- ❑ Must be connected to the network; Receipt printing system is available

Postconditions: The system is still available for the next operation

Use-case description example

Title and brief description: **Transaction;** The use case begins when a customer inserts a card into an ATM. The system will allow customers to make transactions

Basic flow:

1. Validate card and check customer login (performed by login use case)
2. Display the menu interface asking customers to choose the service to perform
3. Customers choose the service to perform
4. Executing the corresponding service when the customer chooses (withdrawal, deposit, view account information: described in each respective use case)
5. Close the menu interface and notify the customer to withdraw the card
6. Customer withdraws card and system returns to ready state

Alternative flows:

Handling invalid card and failed login

Preconditions:

- ❑ ATM must be connected to the banking network
- ❑ ATM must have enough paper to print receipts

Postconditions: ATM must be available for a new customer for the next operation

Exercises

- ❖ Make a Use-case description for
 - Do appeal procedures (phúc khảo điểm)
 - Register for the course
 - Online goods order
 - Withdraw from ATM

Summarization

- ❖ A use-case model is a model of how different types of users interact with the system to solve a problem
- ❖ A Use-case diagram is used as the primary specification of the functional requirements for the system, as the basis for analysis and design, as an input to iteration planning, as the basis of defining test cases
- ❖ A use case is a sequence of actions a system performs
- ❖ An actor represents EXTERNAL thing that interacts with the system
- ❖ An association between an actor and a use case indicates that the actor and the use case somehow interact or communicate with each other

Summarization

- ❖ Generalization between use cases is similar to generalization between classes
- ❖ **<<extend>>** is a relationship that specifies how and when the behavior defined in usually supplementary (optional) extending use case can be inserted into the behavior defined in the extended use case
- ❖ **<<include>>** show that behavior of the included use case (the addition) is inserted into the behavior of the including (the base) use case
- ❖ System boundary is a conceptual boundary, to indicate the application of a Use-case to another entity
- ❖ Refine Use-case diagrams includes: add use cases that describe software-specific functionality, develop <<extend>> relationship, develop <<include>> relationship



Review questions

- ❖ What is the role of the requirements modeling with Use-case?
- ❖ What is an actor and its roles? How many type of actor are there? Give examples?
- ❖ What are the relationships between the Use-cases? Compare the differences between them.
- ❖ What is the Use-case refinement? What steps dose it include?