HUST

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

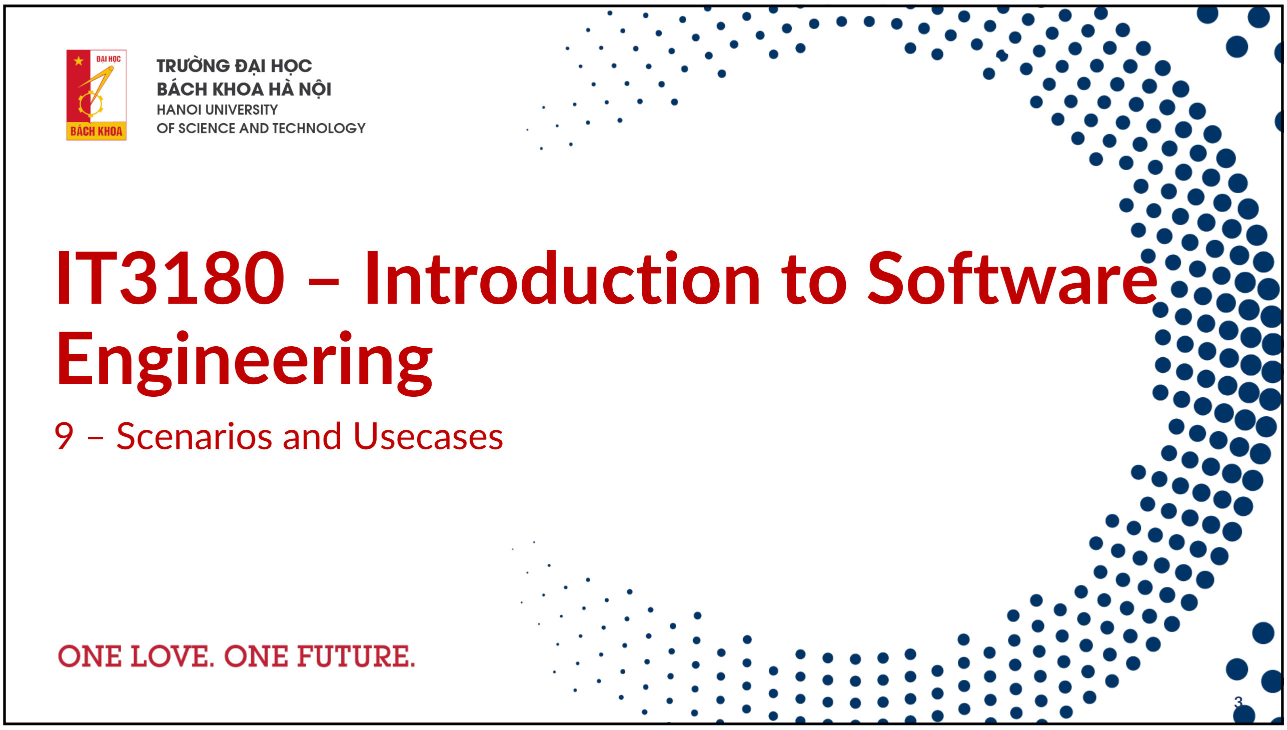
1

SOICT

School of Information and Communication Technology

ONE LOVE. ONE FUTURE.

2



TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

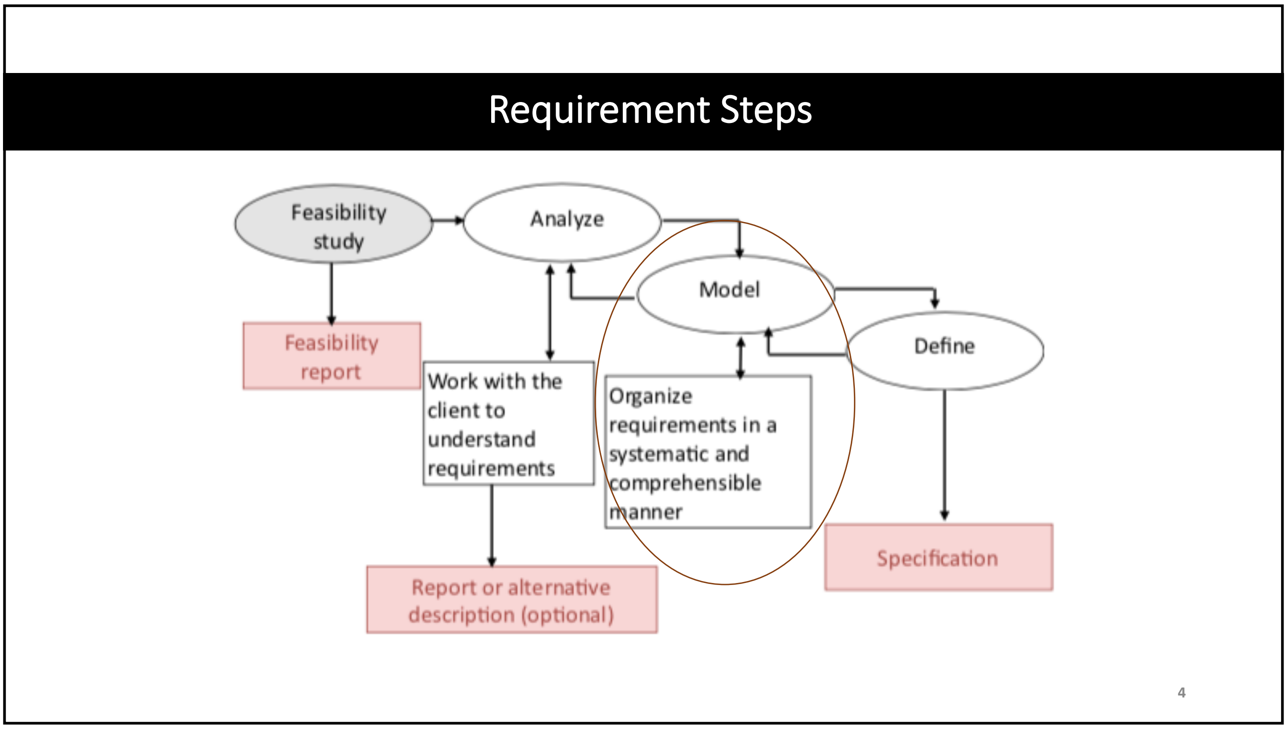
IT3180 – Introduction to Software Engineering

9 – Scenarios and Usecases

ONE LOVE. ONE FUTURE.

3

3



Requirement Steps

Feasibility study

Analyze

Model

Define

Feasibility report

Work with the client to understand requirements

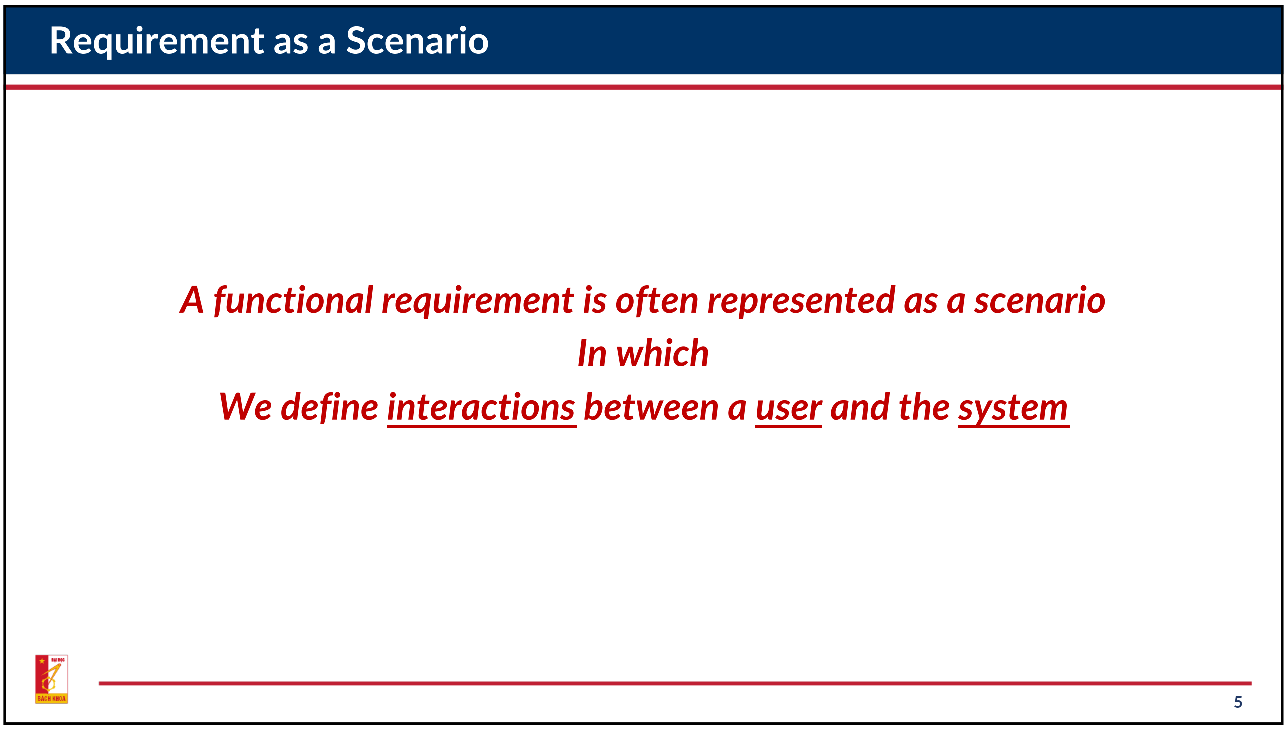
Organize requirements in a systematic and comprehensible manner

Report or alternative description (optional)

Specification

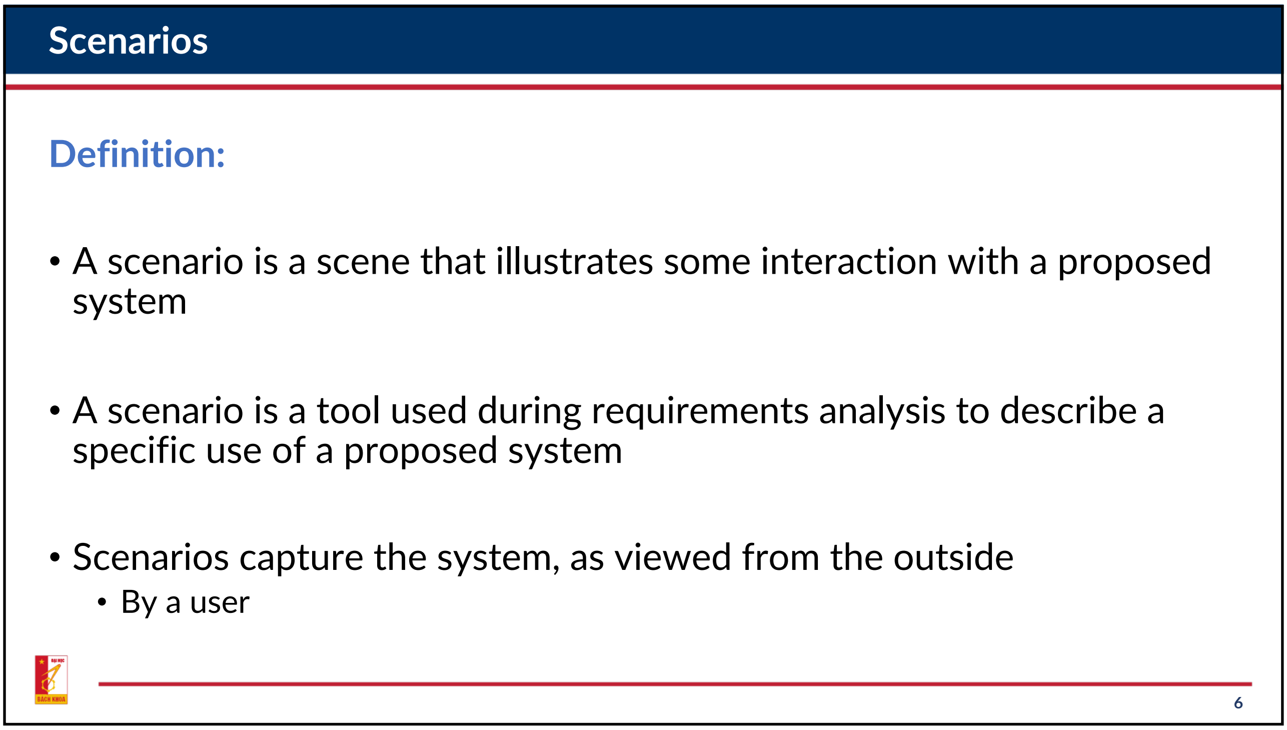
4

4



Requirement as a Scenario

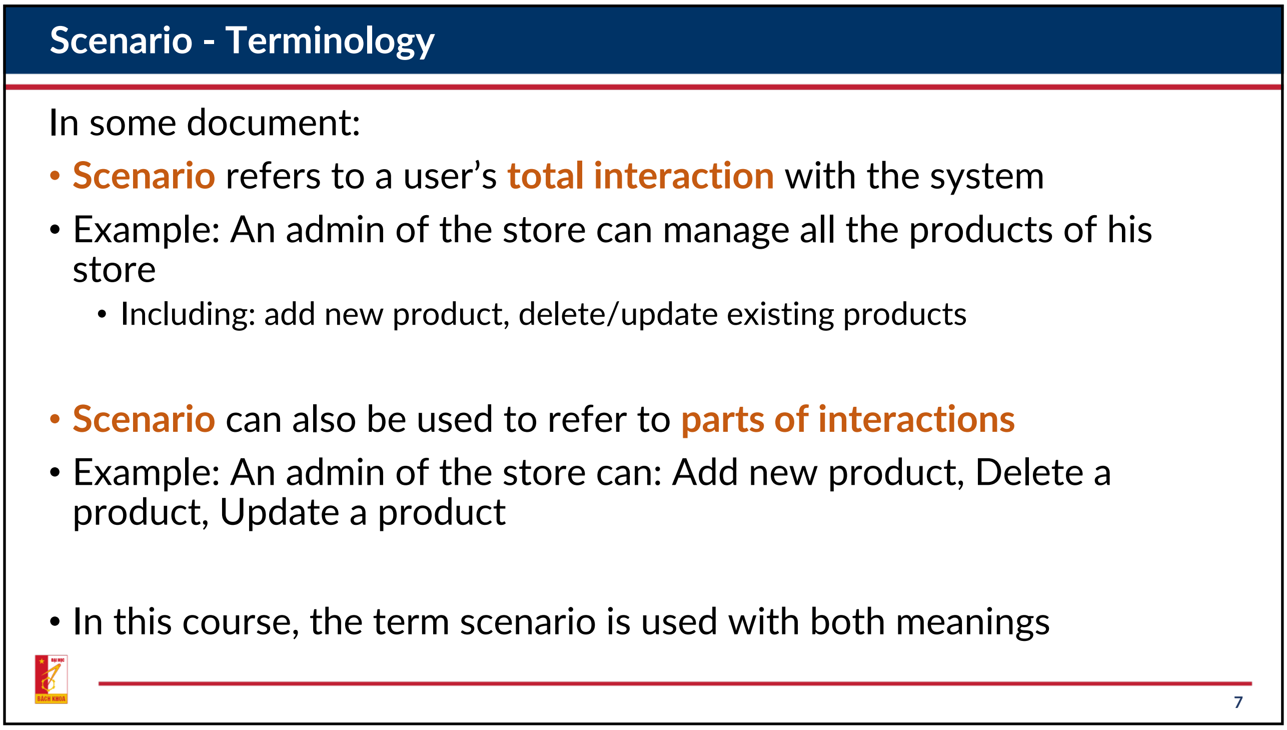
A functional requirement is often represented as a scenario
In which
We define interactions between a user and the system



Scenarios

Definition:

- A scenario is a scene that illustrates some interaction with a proposed system
- A scenario is a tool used during requirements analysis to describe a specific use of a proposed system
- Scenarios capture the system, as viewed from the outside
  - By a user



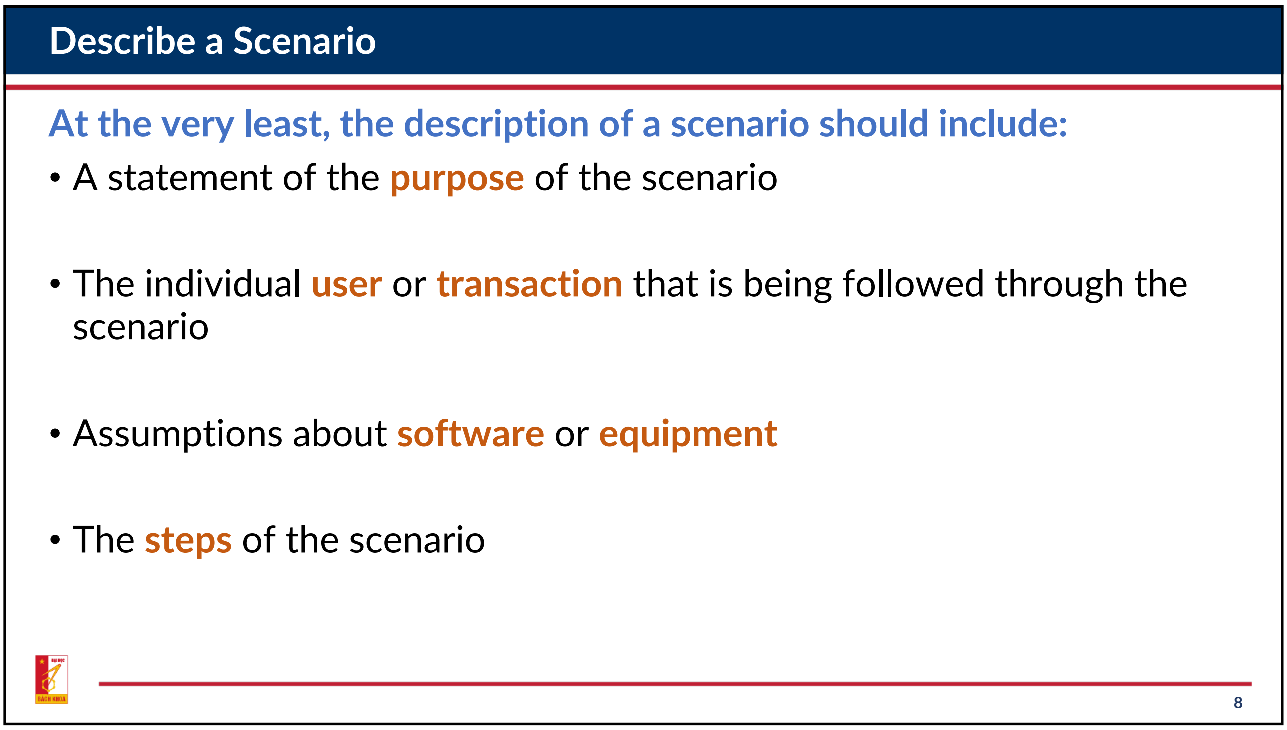
Scenario - Terminology

In some document:

- **Scenario** refers to a user's **total interaction** with the system
- Example: An admin of the store can manage all the products of his store
  - Including: add new product, delete/update existing products

- **Scenario** can also be used to refer to **parts of interactions**
- Example: An admin of the store can: Add new product, Delete a product, Update a product

- In this course, the term scenario is used with both meanings



Describe a Scenario

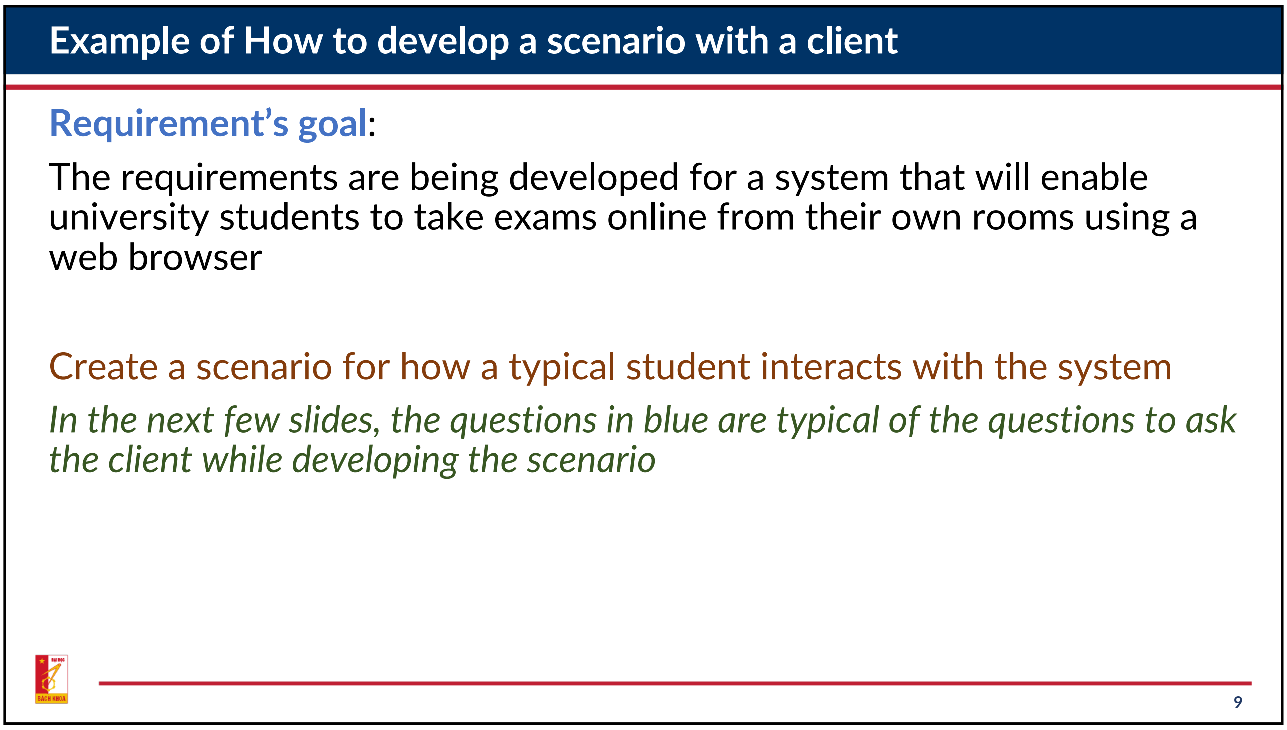
At the very least, the description of a scenario should include:

- A statement of the **purpose** of the scenario

- The individual **user** or **transaction** that is being followed through the scenario

- Assumptions about **software** or **equipment**

- The **steps** of the scenario



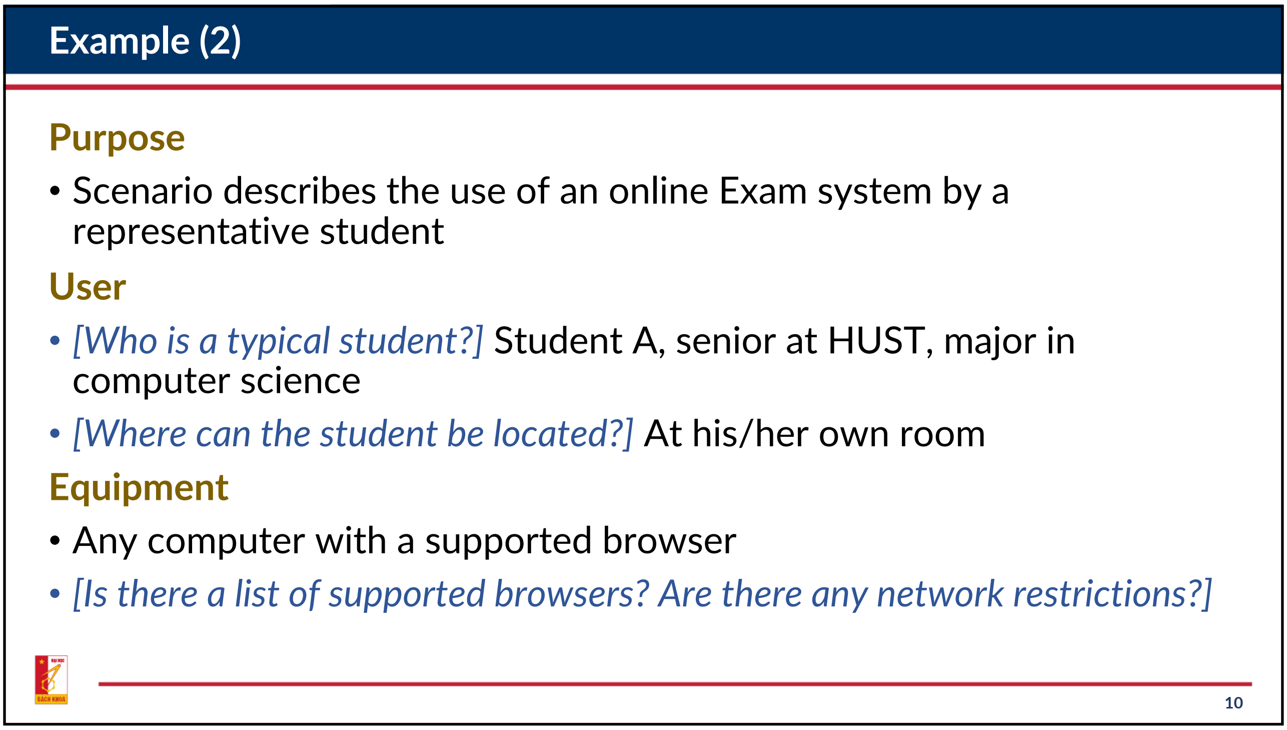
Example of How to develop a scenario with a client

Requirement's goal:

The requirements are being developed for a system that will enable university students to take exams online from their own rooms using a web browser

Create a scenario for how a typical student interacts with the system

In the next few slides, the questions in blue are typical of the questions to ask the client while developing the scenario



Example (2)

Purpose

- Scenario describes the use of an online Exam system by a representative student

User

- *[Who is a typical student?]* Student A, senior at HUST, major in computer science
- *[Where can the student be located?]* At his/her own room

Equipment

- Any computer with a supported browser
- *[Is there a list of supported browsers? Are there any network restrictions?]*

Example (3)

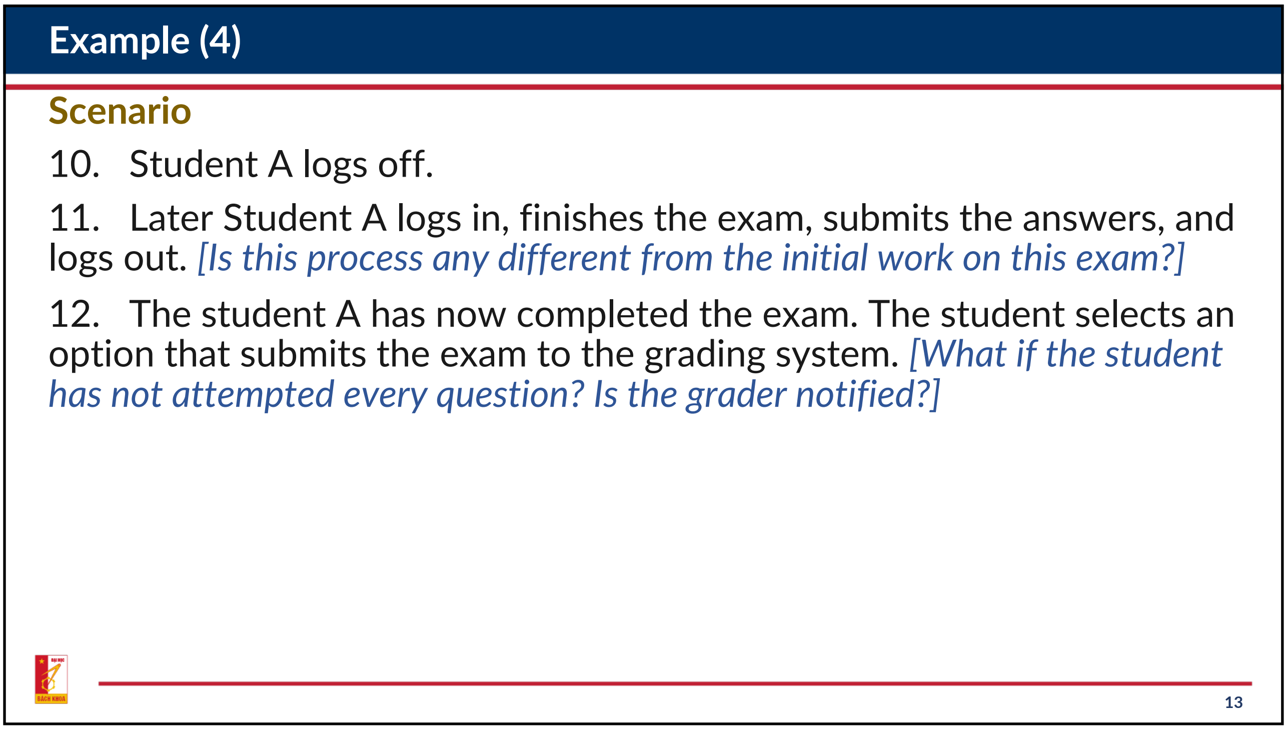
Scenario

1. Student A authenticates. *[How does a HUST student authenticate?]*
2. Student A starts browser and types URL of Exam system. *[How does the student know the URL?]*
3. Exam system displays list of options. *[Is the list tailored to the individual user?]*
4. Student A selects IT3180 Exam 1.
5. A list of questions is displayed, each marked to indicate whether completed or not. *[Can the questions be answered in any order?]*
6. Student A selects a question and chooses whether to submit a new answer or edit a previous answer *[Is it always possible to edit a previous answer? Are there other options?]*

Example (3)

Scenario

7. *[What types of questions are there: text, multiple choice, etc.?]* The first question requires a written answer. Student A is submitting a new answer. The student has a choice whether to type the solution into the browser or to attach a separate file. Student A decides to attach a file. *[What types of file are accepted?]*
8. For the second question, the student chooses to edit a previous answer. Student A chooses to delete a solution previously typed into the browser and to replace it with an attached file. *[Can the student edit a previous answer, or must it always be replaced with a new answer?]*
9. As an alternative to completing the entire exam in a single session, Student A decides to saves the completed questions to continue later. *[Is this always permitted?]*



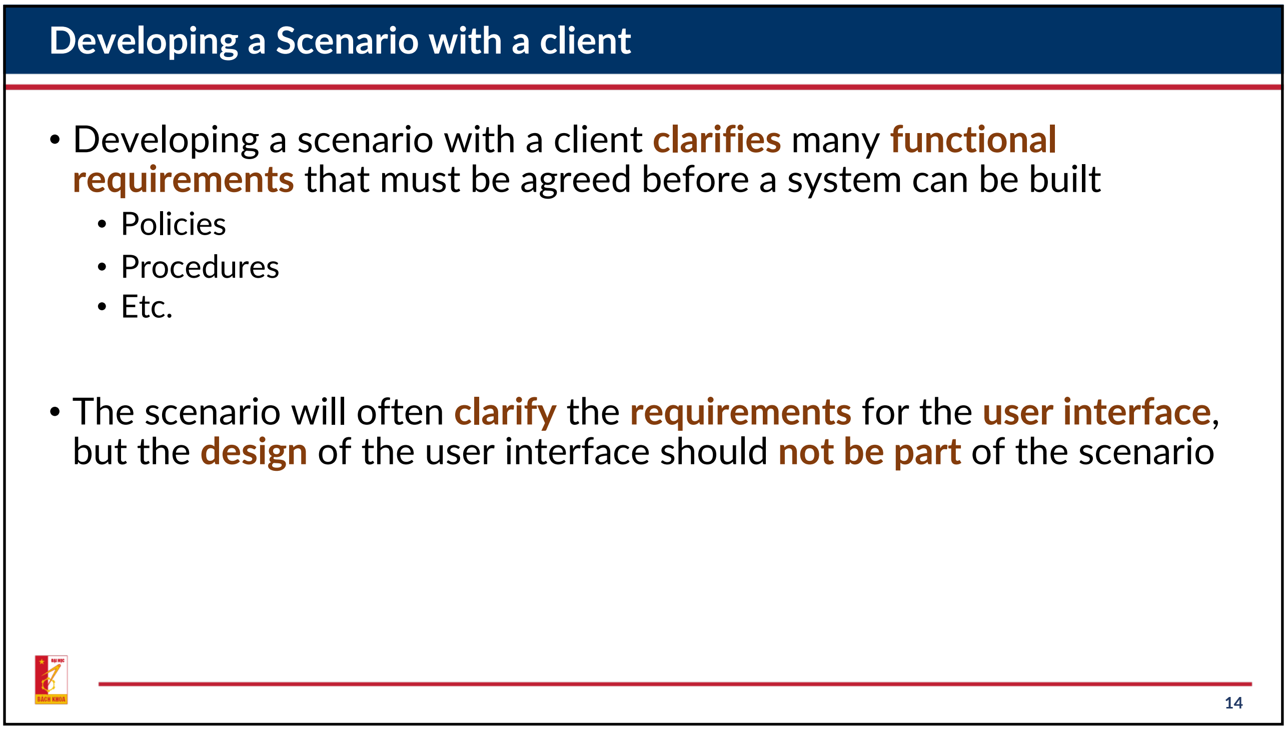
Example (4)

Scenario

10. Student A logs off.

11. Later Student A logs in, finishes the exam, submits the answers, and logs out. *[Is this process any different from the initial work on this exam?]*

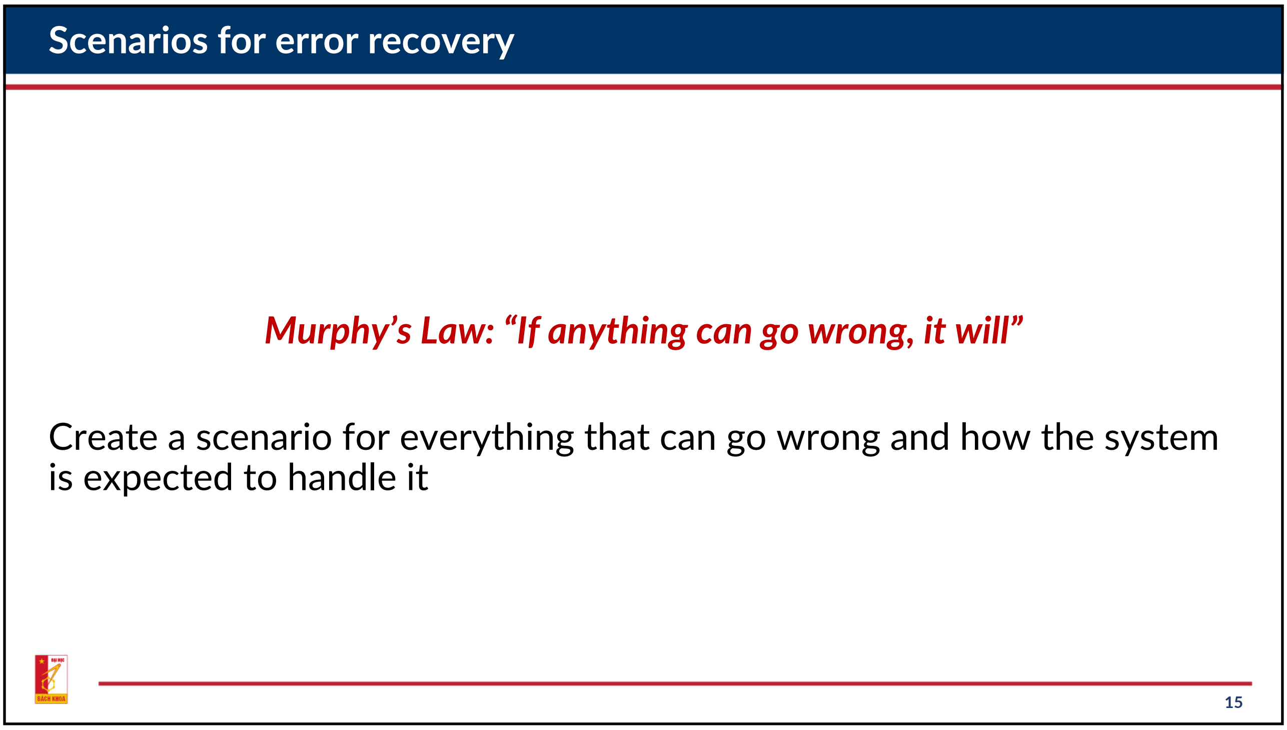
12. The student A has now completed the exam. The student selects an option that submits the exam to the grading system. *[What if the student has not attempted every question? Is the grader notified?]*



Developing a Scenario with a client

- Developing a scenario with a client **clarifies** many **functional requirements** that must be agreed before a system can be built
  - Policies
  - Procedures
  - Etc.

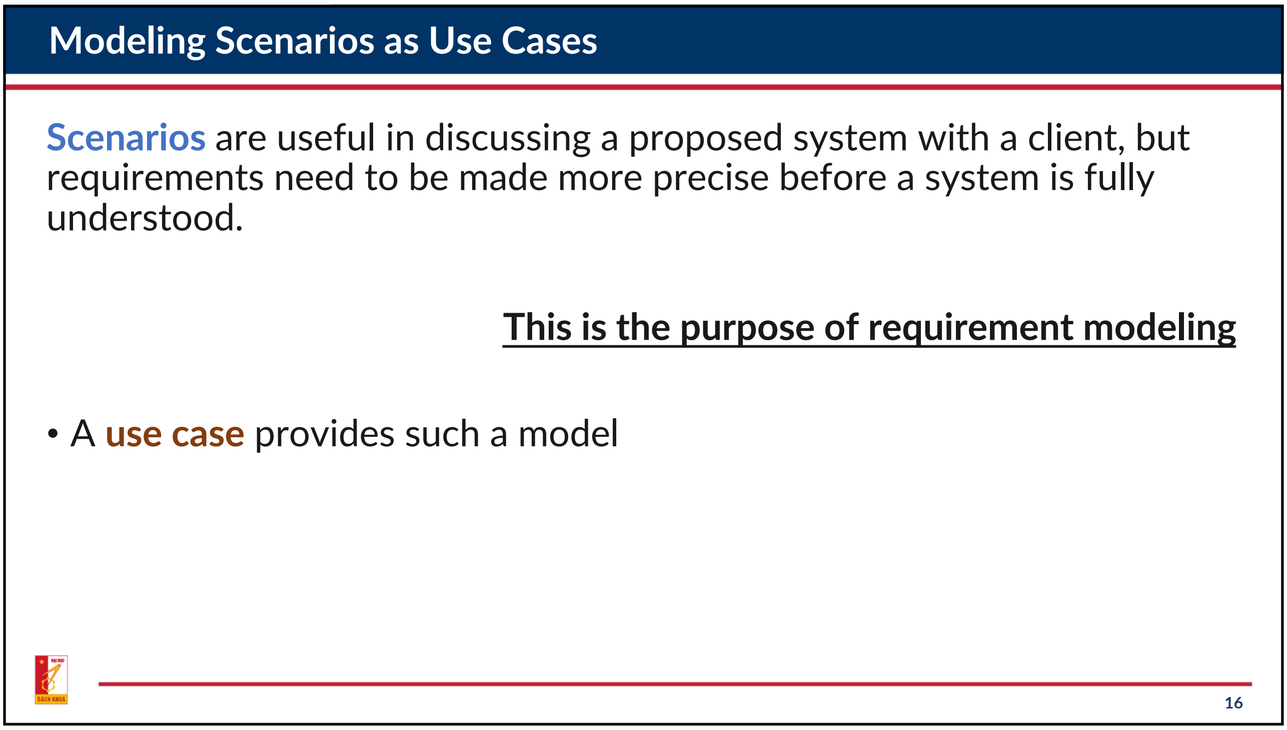
- The scenario will often **clarify** the **requirements** for the **user interface**, but the **design** of the user interface should **not be part** of the scenario



Scenarios for error recovery

Murphy's Law: "If anything can go wrong, it will"

Create a scenario for everything that can go wrong and how the system is expected to handle it



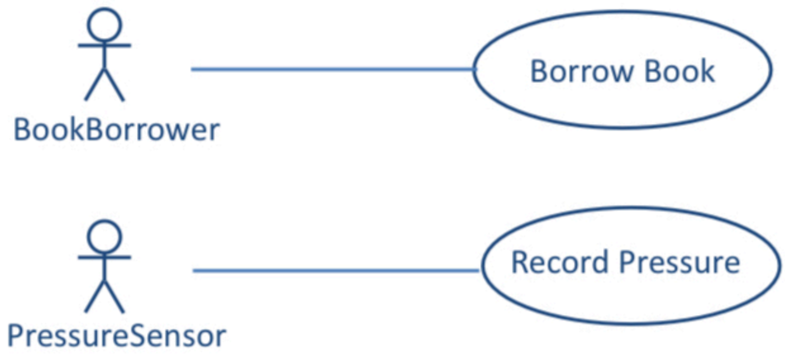
Modeling Scenarios as Use Cases

Scenarios are useful in discussing a proposed system with a client, but requirements need to be made more precise before a system is fully understood.

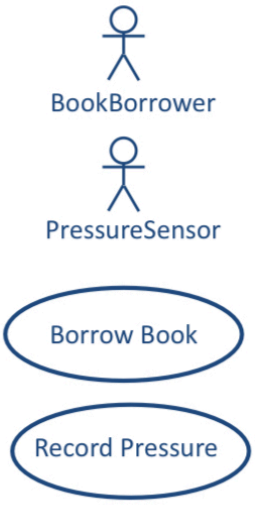
This is the purpose of requirement modeling

- A **use case** provides such a model

Two simple Use Cases



Actor and Use Case Diagram



- An **actor** is a user of a system in a particular **role**.

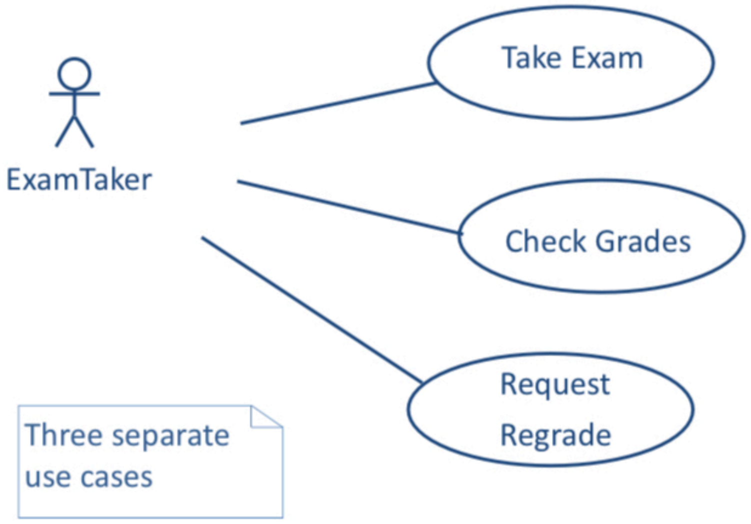
  An actor can be human or an external system.

- A **use case** is a a task that an actor needs to perform with the help of the system.

Use Cases and Actors

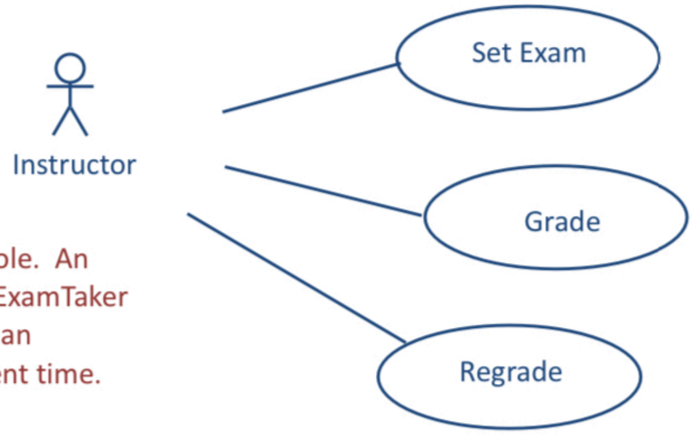
- Actor is role, not an individual
- E.g., A staff in a hotel can have many roles
  - Receptionist
  - Security Staff
  - Etc.
- Actor must be a **beneficiary** of the use case
- When naming actors, choose names that describe the role, not generic names, such as “user” or “client”

Use Cases for Exam System



Three separate use cases

Use cases for Exam System (2)



Note that actor is a role. An individual can be an ExamTaker on one occasion and an Instructor at a different time.

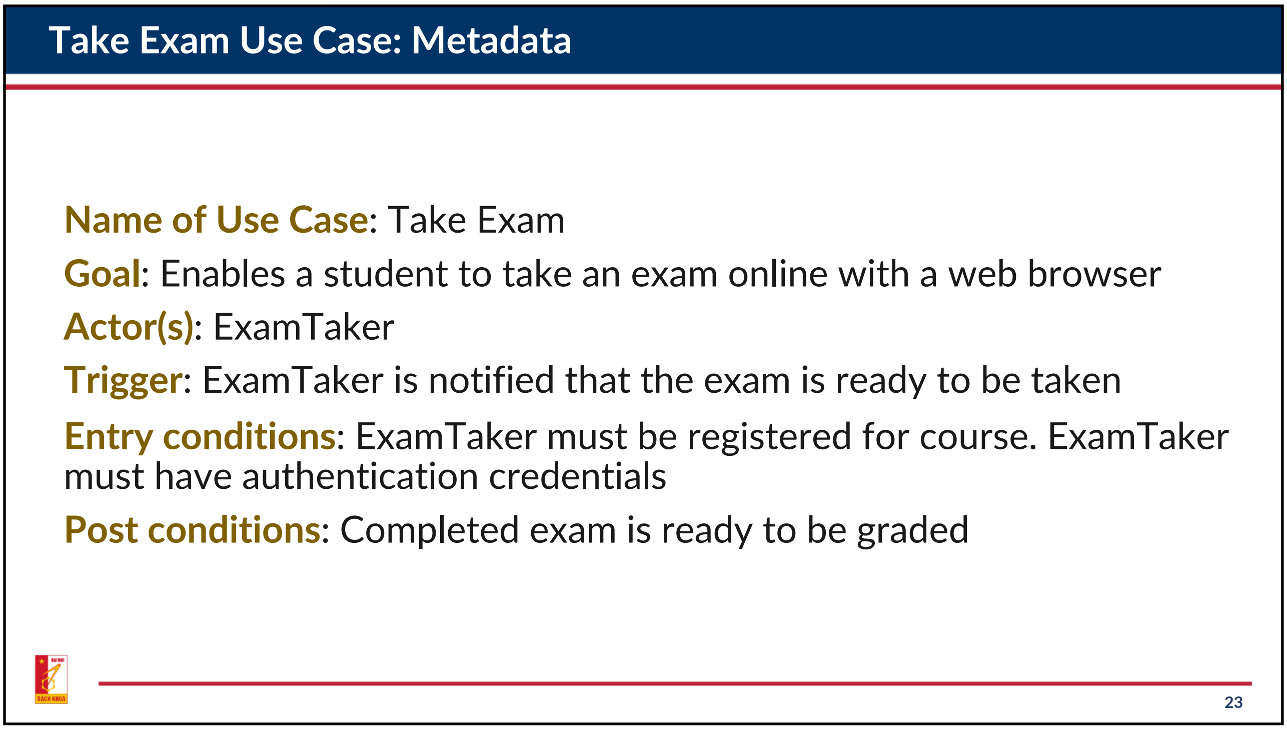
Describe a Use Case

Metadata

- The **name** of the use case
- **Goal** of the use case
- The **actor** or **actors**
- **Trigger**
- **Entry conditions** at beginning
- **Post conditions** at end

Flow of events

- The **basic flow** of events
- **Alternative flows** of events
- **Exceptions**



Take Exam Use Case: Metadata

Name of Use Case: Take Exam

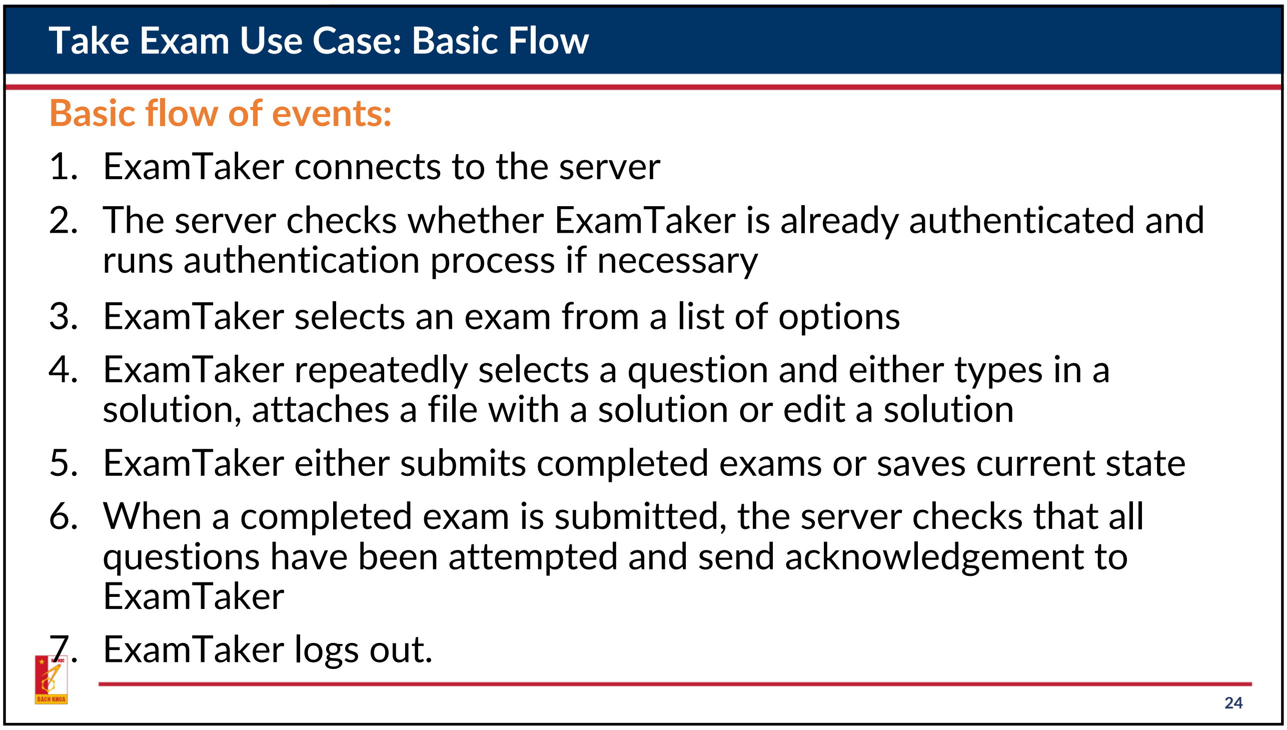
Goal: Enables a student to take an exam online with a web browser

Actor(s): ExamTaker

Trigger: ExamTaker is notified that the exam is ready to be taken

Entry conditions: ExamTaker must be registered for course. ExamTaker must have authentication credentials

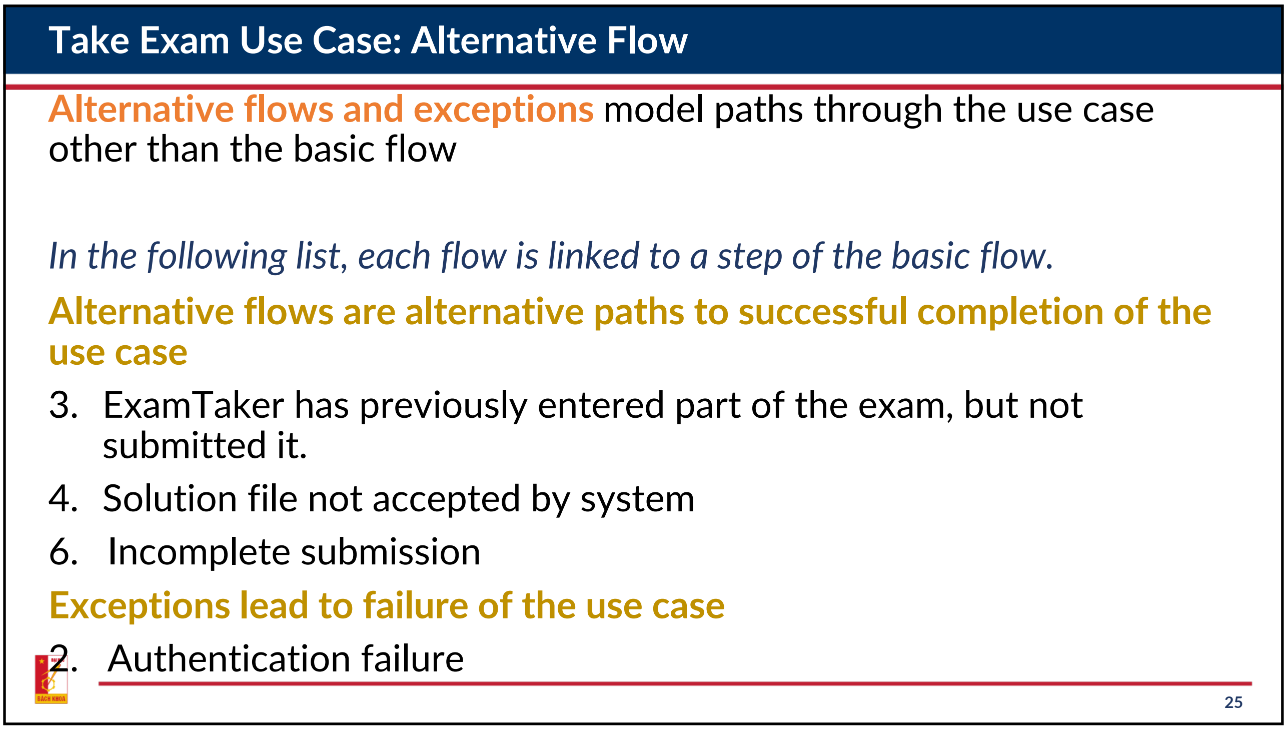
Post conditions: Completed exam is ready to be graded



Take Exam Use Case: Basic Flow

Basic flow of events:

1. ExamTaker connects to the server
2. The server checks whether ExamTaker is already authenticated and runs authentication process if necessary
3. ExamTaker selects an exam from a list of options
4. ExamTaker repeatedly selects a question and either types in a solution, attaches a file with a solution or edit a solution
5. ExamTaker either submits completed exams or saves current state
6. When a completed exam is submitted, the server checks that all questions have been attempted and send acknowledgement to ExamTaker
7. ExamTaker logs out.



Take Exam Use Case: Alternative Flow

Alternative flows and exceptions model paths through the use case other than the basic flow

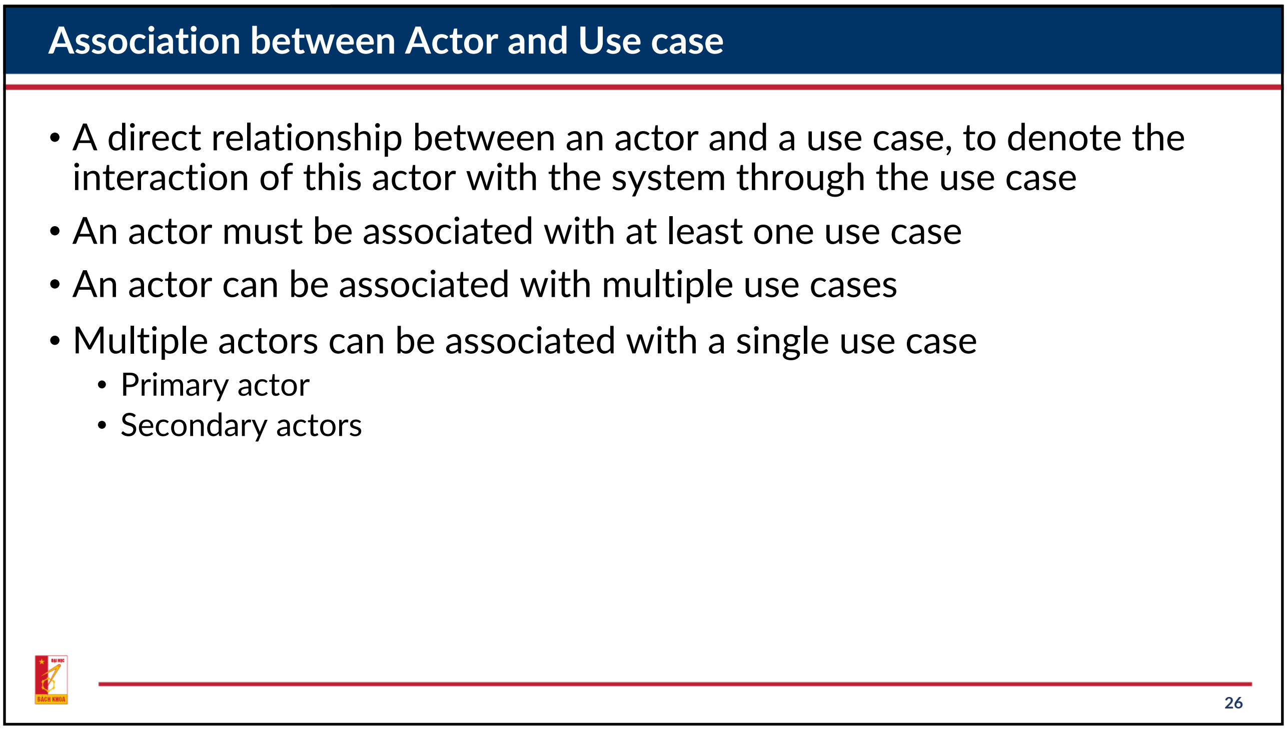
In the following list, each flow is linked to a step of the basic flow.

Alternative flows are alternative paths to successful completion of the use case

3. ExamTaker has previously entered part of the exam, but not submitted it.
4. Solution file not accepted by system
6. Incomplete submission

Exceptions lead to failure of the use case

2. Authentication failure



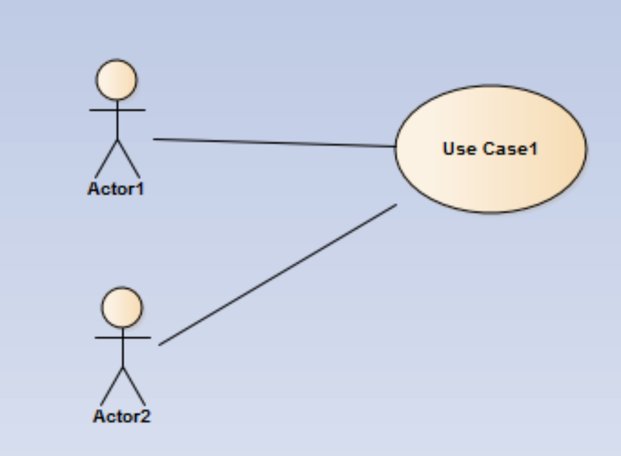
Association between Actor and Use case

- A direct relationship between an actor and a use case, to denote the interaction of this actor with the system through the use case
- An actor must be associated with at least one use case
- An actor can be associated with multiple use cases
- Multiple actors can be associated with a single use case
  - Primary actor
  - Secondary actors

Association (2)

Which statement is correct?

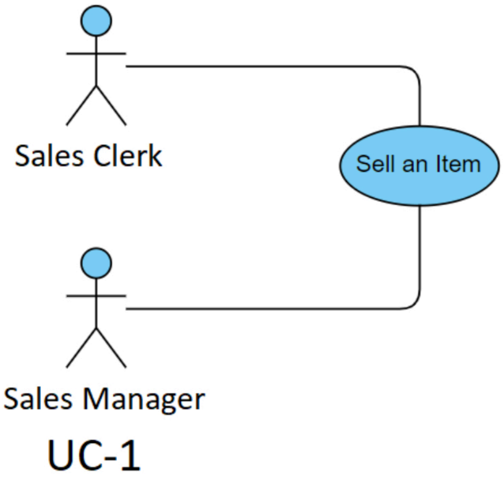
- Actor 1 and Actor 2 can interact the system through UC1
- Both Actor 1 and Actor 2 are needed to start UC1
- Actor 1 starts UC1 first then Actor 2 does something later or vice versa



Association (3)

Two actors Sales Clerk and Sales Manager are required to execute the use case Sell an Item

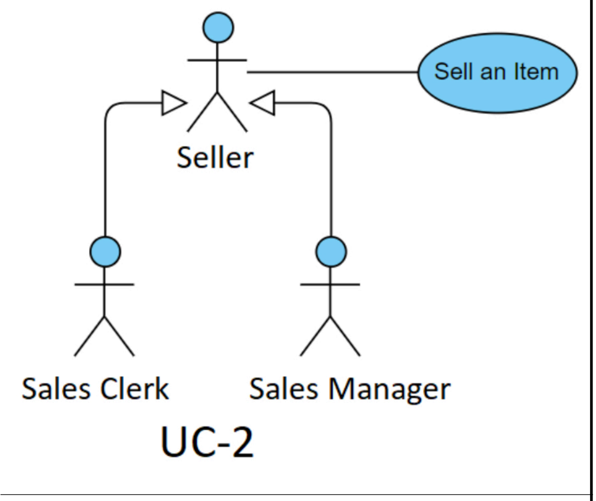
- But only one primary actor, who starts the UC, supposing Sales Clerk
- Every sale is performed by a clerk
- But it should be approved by a sales manager
- Sales Manager is the secondary actor who is involved in the execution



Association (4)

Both the Sales Manager and the Sales Clerk can sell an item (i.e., start the Sell an Item use case)

- Both actors Sales Manager and Sales Clerk can act as the seller (Seller)
- Using the inheritance relationship between Actor to denote that actors share the same use case

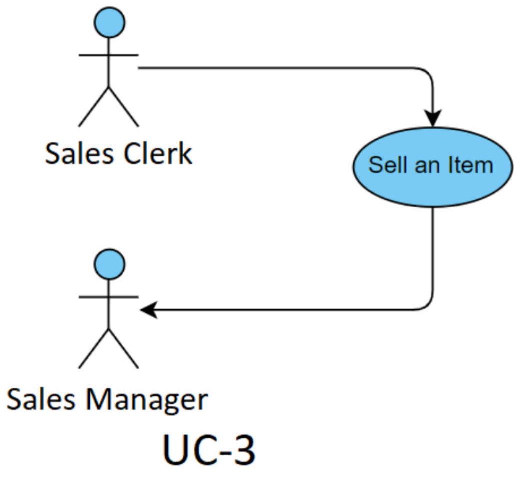


Association (5)

Same situation as UC1, but a minor difference

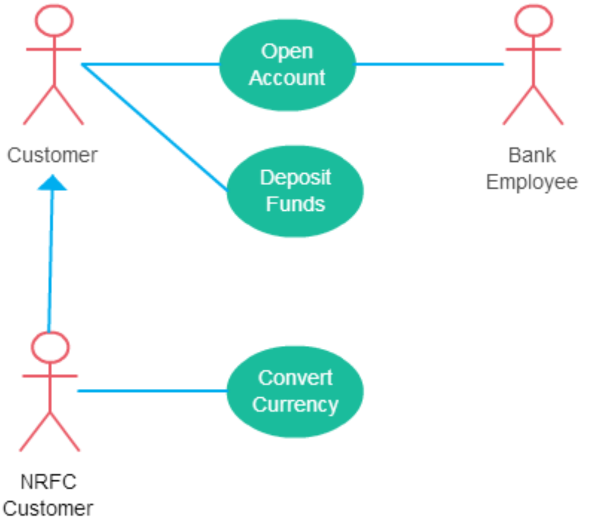
- The arrows indicate clearly who is the primary and the secondary actors
- However, these arrows are not standardized in UML
- But they are used as private notation of the organization

There is no differentiation between primary and secondary actors



Generalization between Actors

- One actor can inherit the role of the other actor
- The descendant inherits all the use cases of the ancestor
- The descendant can have one or more use cases that are specific to that role



Relationship between Use Cases

There are three kinds of relationship between Use cases:

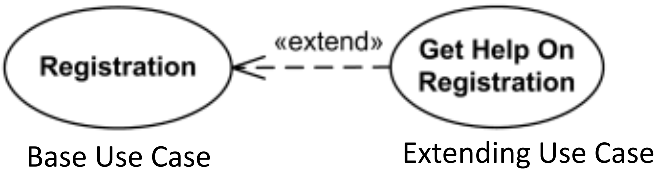
- Generalization
- Extension
- Inclusion

The <<extend>> Relationship

The <<extend>> relationship

Use cases can make use of other use cases

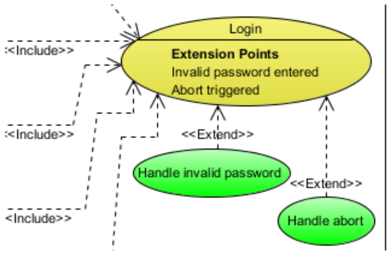
- Base use case: extended use case
- Extending use case: provides optional behavior



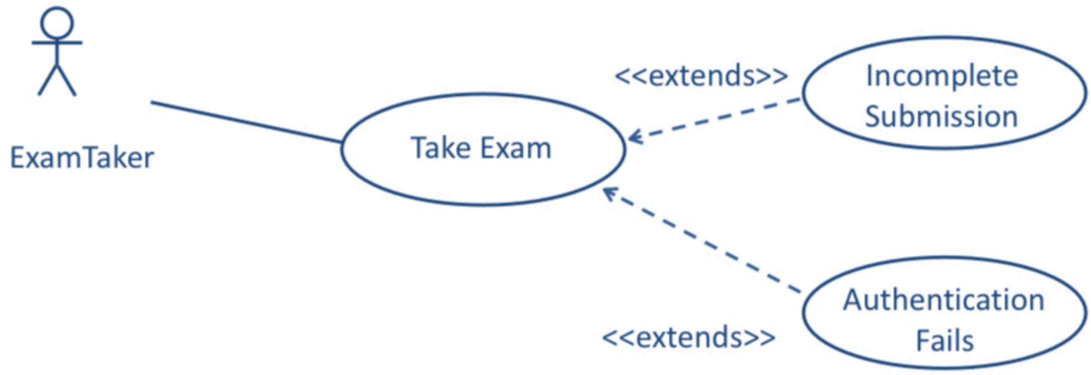
The <<extend>> Relationship (2)

Extension Point

- A feature of a use case that identifies a point where the behavior of this use case can be augmented with elements of another (extending) use case.
- If an alternative flow or an exception needs extra detail, it can be modeled as a separate use case using the **<<extend>>** relationship



Take Exam extensions



The <<include>> relationship

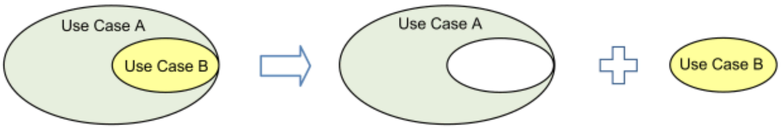
A **directed relationship** between two use cases which is used to show that behavior of the **included use case** (the addition) is **inserted** into the behavior of the **including (the base) use case**

The <<include>> relationship could be used:

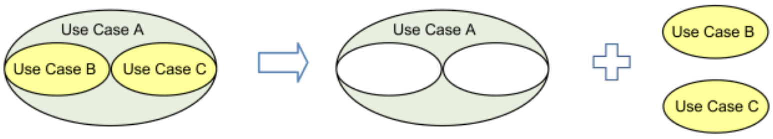
- To simplify **large use case** by splitting it into several use cases
- To extract **common parts** of the behaviors of two or more use cases

The <<include>> relationship (2)

A large use case is split into several use cases

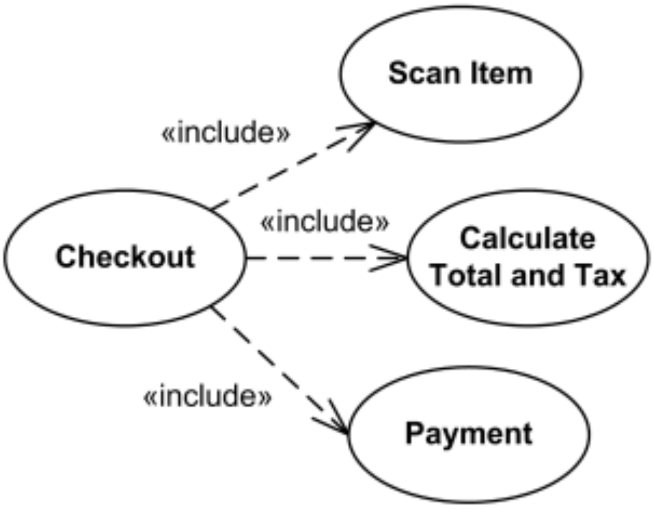


Use case B is extracted from larger use case A into a separate use case



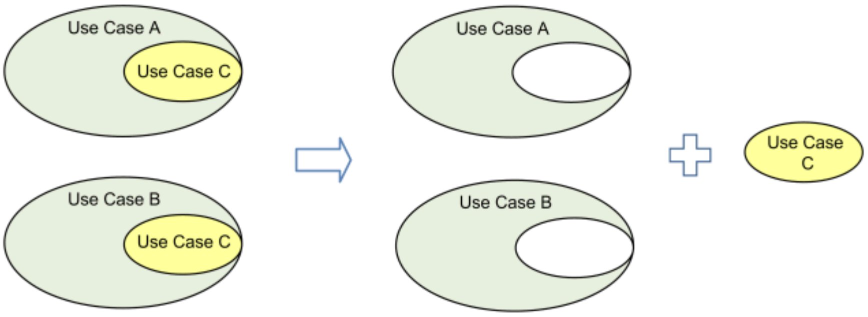
Use cases B and C are extracted from larger use case A into separate use cases

Example



The <<include>> relationship (3)

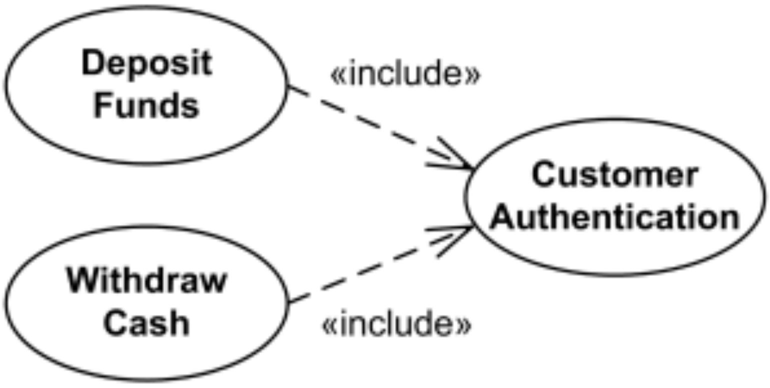
Extract **common parts** of the behaviors of a some use cases

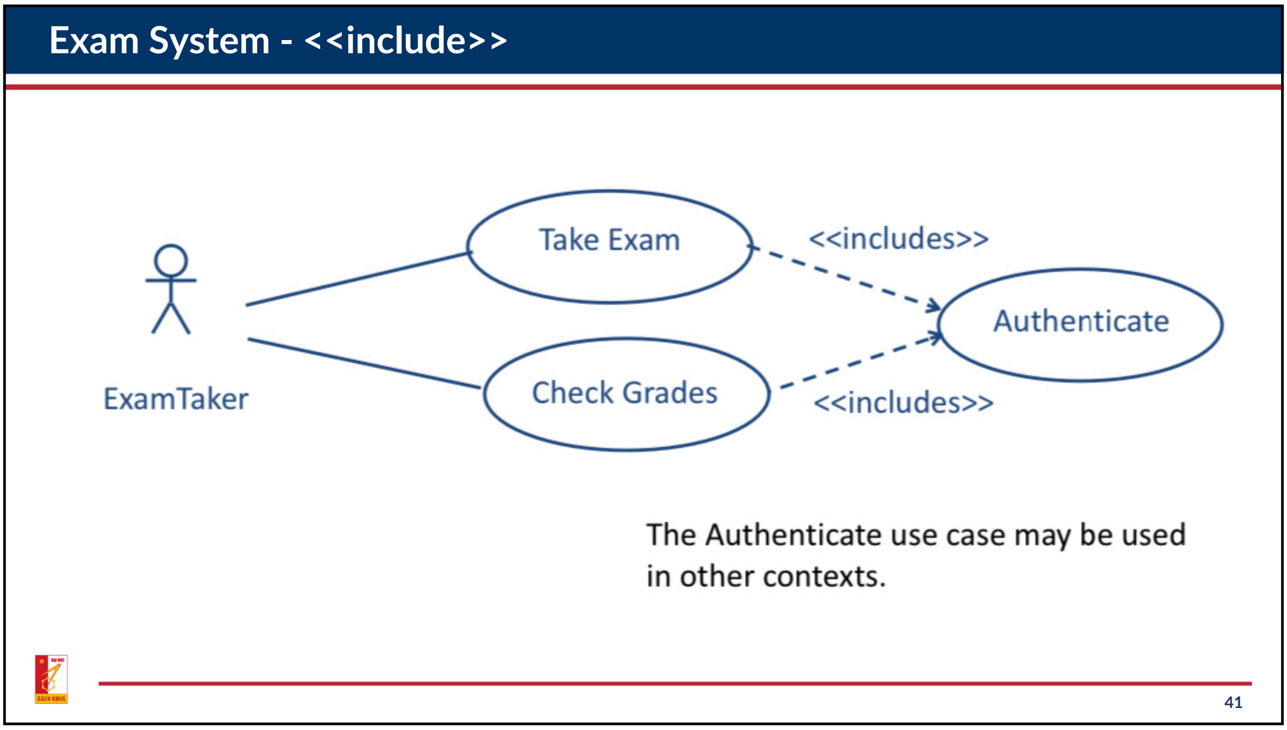


Use case C is extracted from use case A and B to be reused by both use cases A, B

There is no "**inclusion points**" to specify location or condition of inclusion for the <<include>> relationship

Example





Exam System - <<include>>

ExamTaker

Take Exam

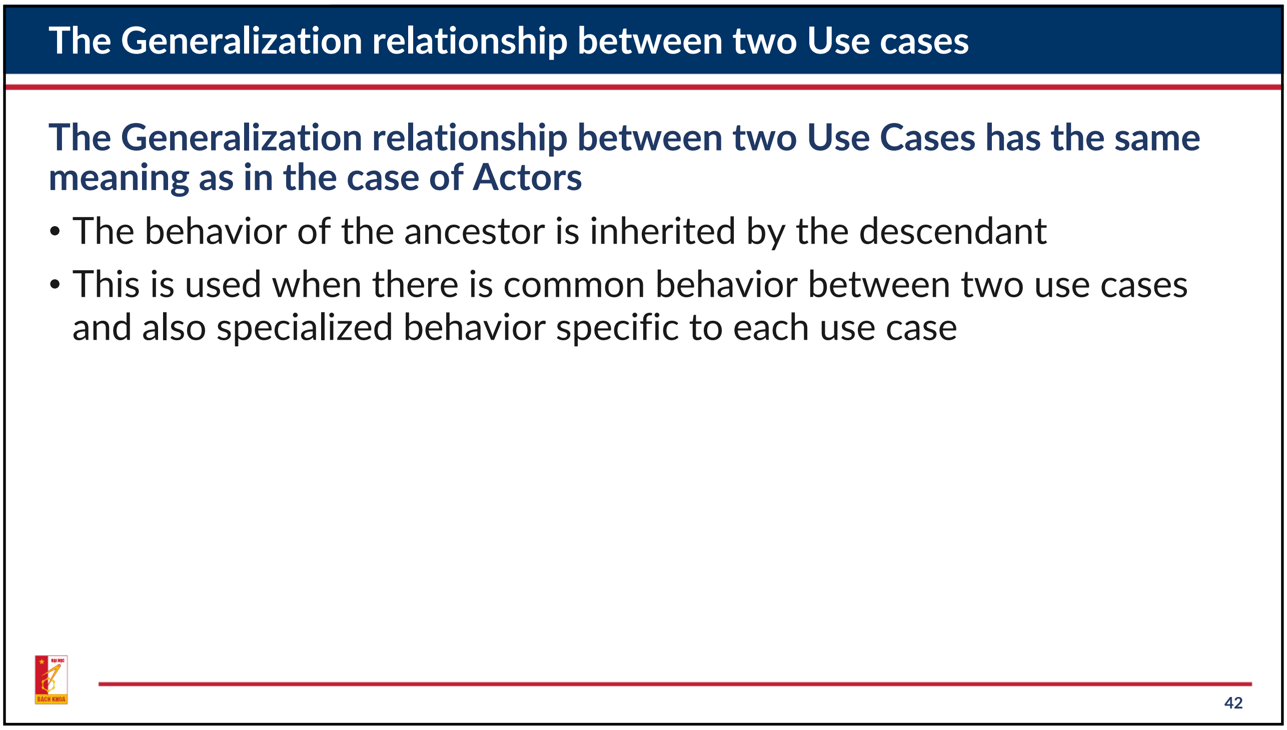
Check Grades

<<includes>>

<<includes>>

Authenticate

The Authenticate use case may be used in other contexts.



The Generalization relationship between two Use cases

The Generalization relationship between two Use Cases has the same meaning as in the case of Actors

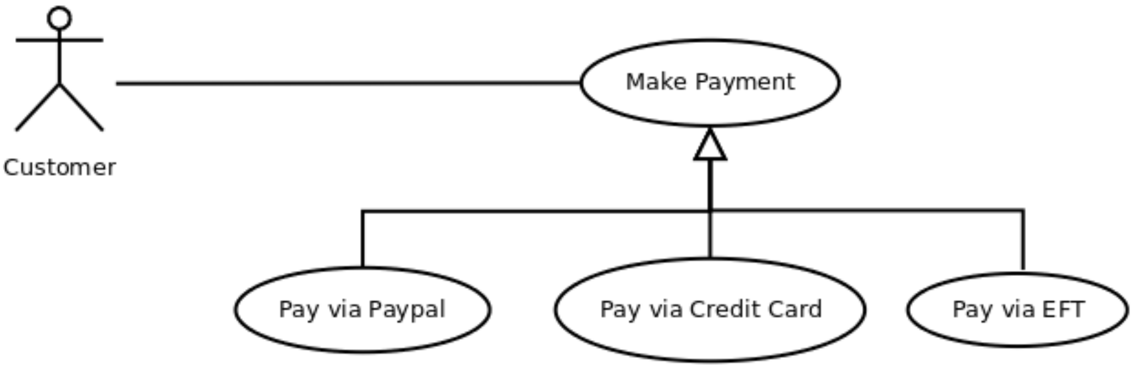
- The behavior of the ancestor is inherited by the descendant
- This is used when there is common behavior between two use cases and also specialized behavior specific to each use case

Example

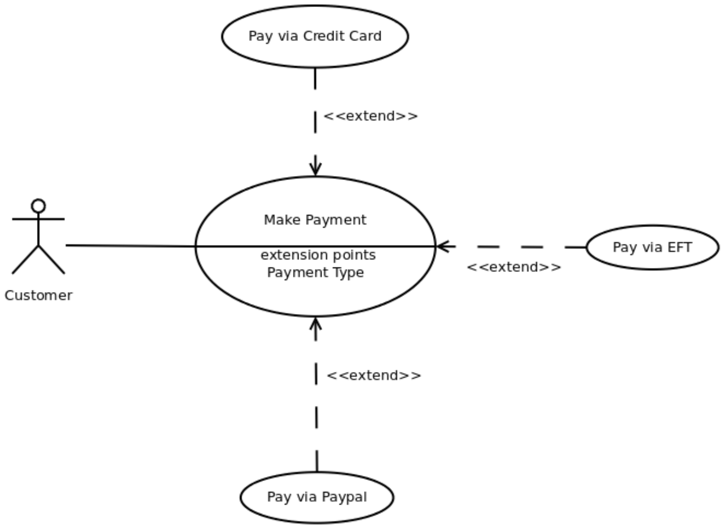
When checkout, a customer has to make a payment. He or she can select one of three payment methods:

- Pay via Paypal
- Pay via Credit Card
- Pay via EFT

Example (2)



Example (3)



Scenario and Use Cases in the Development cycle

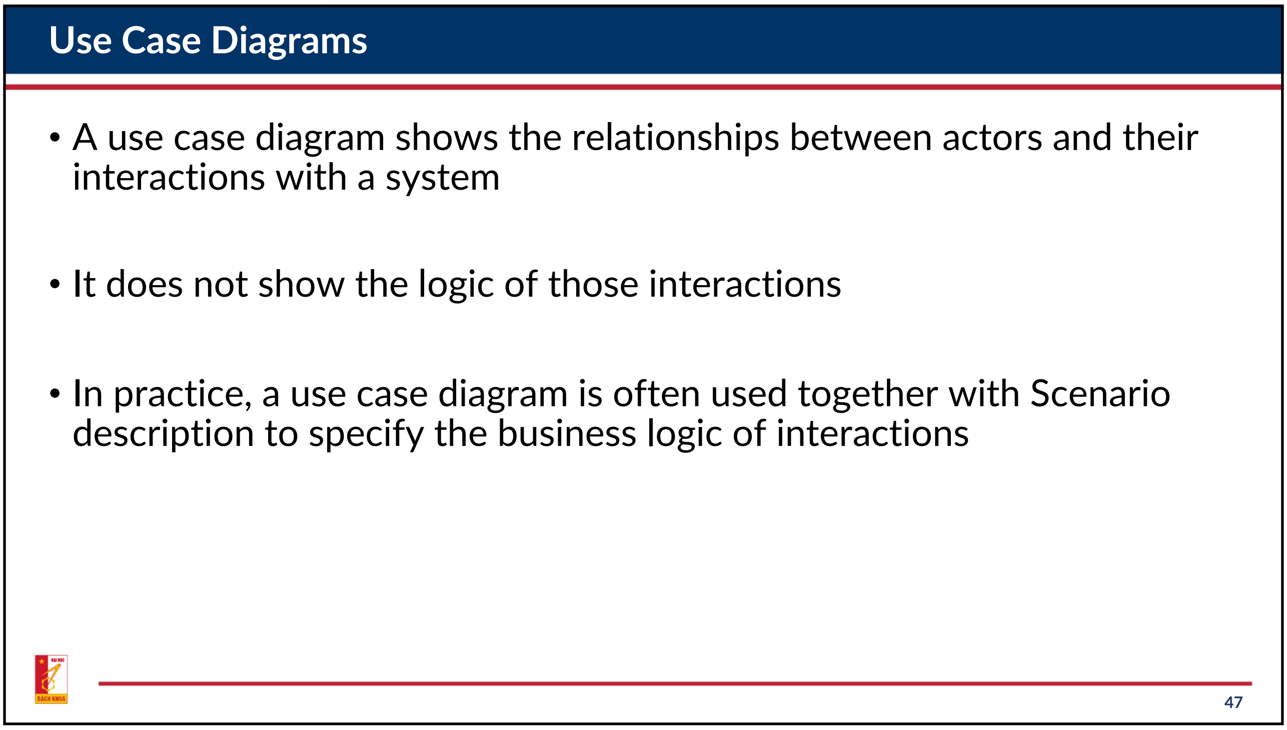
Scenarios and Use cases are both intuitive – easy to discuss with clients

Scenarios are a tool for **requirement analysis**

- They are useful to validate use cases and in checking the design of a system
- They can be used as test cases for acceptance testing

Use cases are a tool for **modeling requirements**

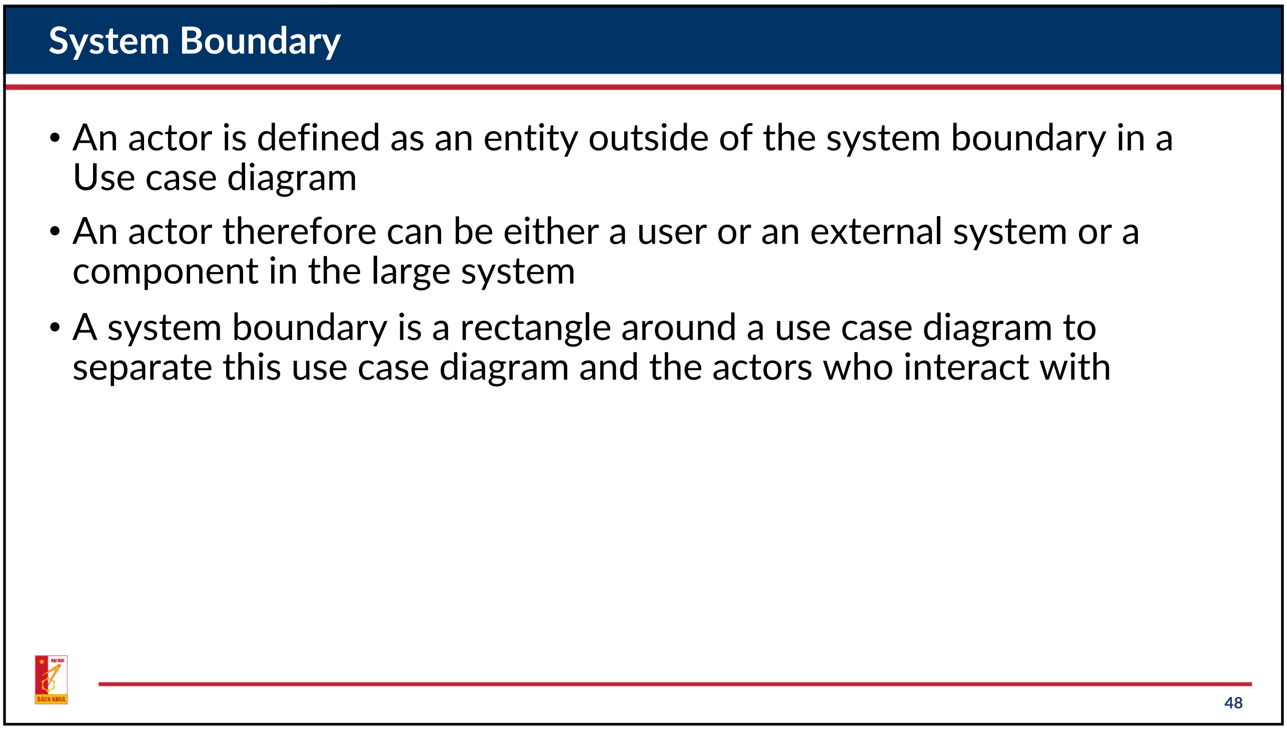
- A set of use cases can provide a framework for the requirement specification
- Use cases are the basis for system and program design, but are often hard to translate into class models



Use Case Diagrams

- A use case diagram shows the relationships between actors and their interactions with a system
- It does not show the logic of those interactions
- In practice, a use case diagram is often used together with Scenario description to specify the business logic of interactions

47



System Boundary

- An actor is defined as an entity outside of the system boundary in a Use case diagram
- An actor therefore can be either a user or an external system or a component in the large system
- A system boundary is a rectangle around a use case diagram to separate this use case diagram and the actors who interact with

48

Exercise 1 – Old Exam Question

The Pizza Ordering System

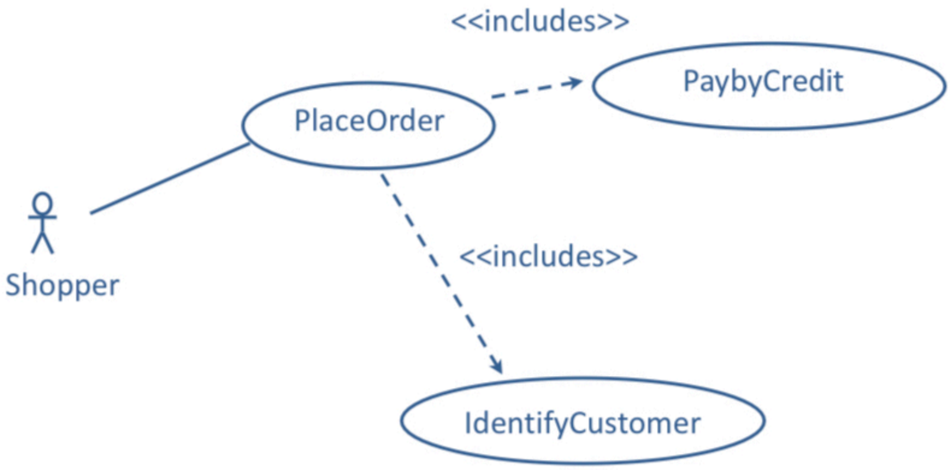
The Pizza Ordering System allows the user of a web browser to order pizza for home delivery. To place an order, a shopper searches to find items to purchase, adds items one at a time to a shopping cart, and possibly searches again for more items.

When all items have been chosen, the shopper provides a delivery address. If not paying with cash, the shopper also provides credit card information.

The system has an option for shoppers to register with the pizza shop. They can then save their name and address information, so that they do not have to enter this information every time that they place an order.

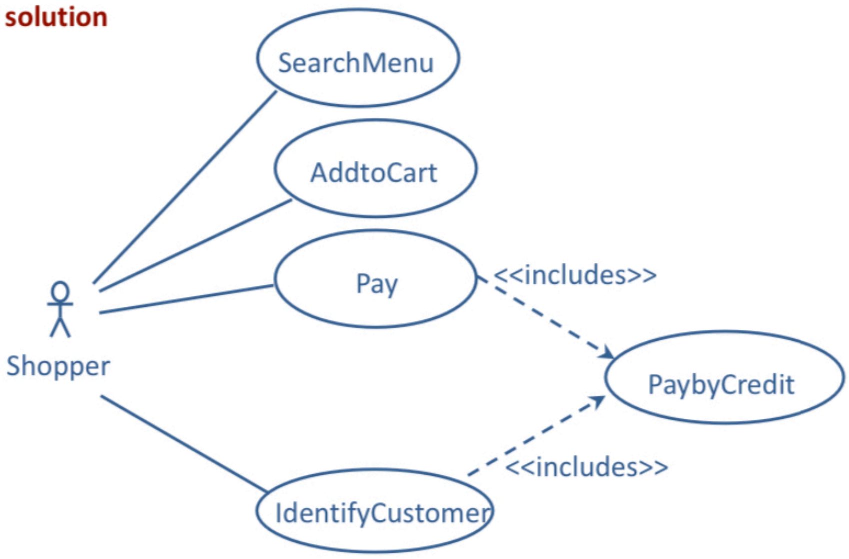
Develop a use case diagram, for a use case for placing an order, *PlaceOrder*. The use case should show a relationship to two previously specified use cases, *IdentifyCustomer*, which allows a user to register and log in, and *PaybyCredit*, which models credit card payments.

Exercise 1 – Correct Solution



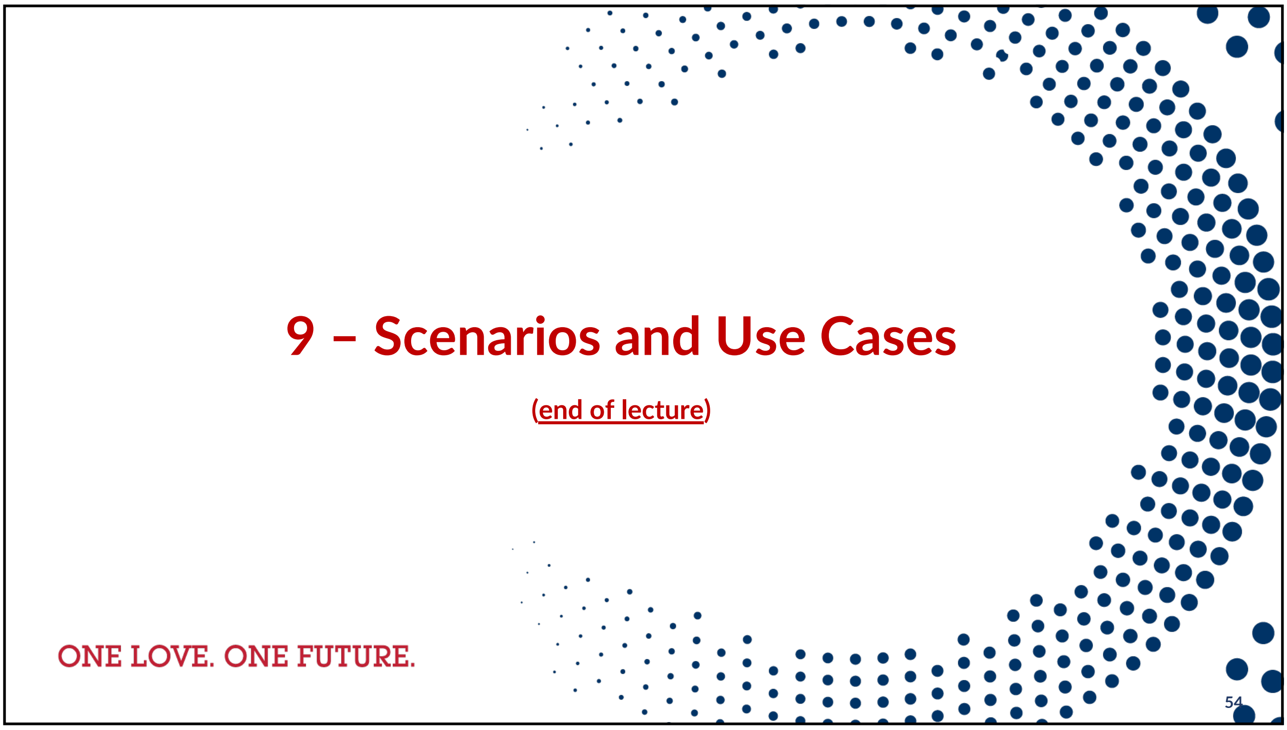
Exercise 1

Wrong solution



Exercise 2

- Modeling the following situation using use cases
- A general customer can come to the Bank X and ask for open an account. He or she will complete a form and the bank employee will validate the form to open his/her account.
- A customer can deposit funds, when the amount is over 10,000$ or his/her age is over 55, a bonus will be calculated and offered to the customer
- A NRFC customer can also open account, deposit funds but he or she can also convert currency



9 – Scenarios and Use Cases

(end of lecture)

ONE LOVE. ONE FUTURE.

54

54