Phân tích thiết kế quy trình nghiệp vụ doanh nghiệp

Chương 4. UML

NỘI DUNG

UML
Các dạng biểu đồ cơ bản trong UML
Nắm bắt yêu cầu
Mô hình hóa yêu cầu

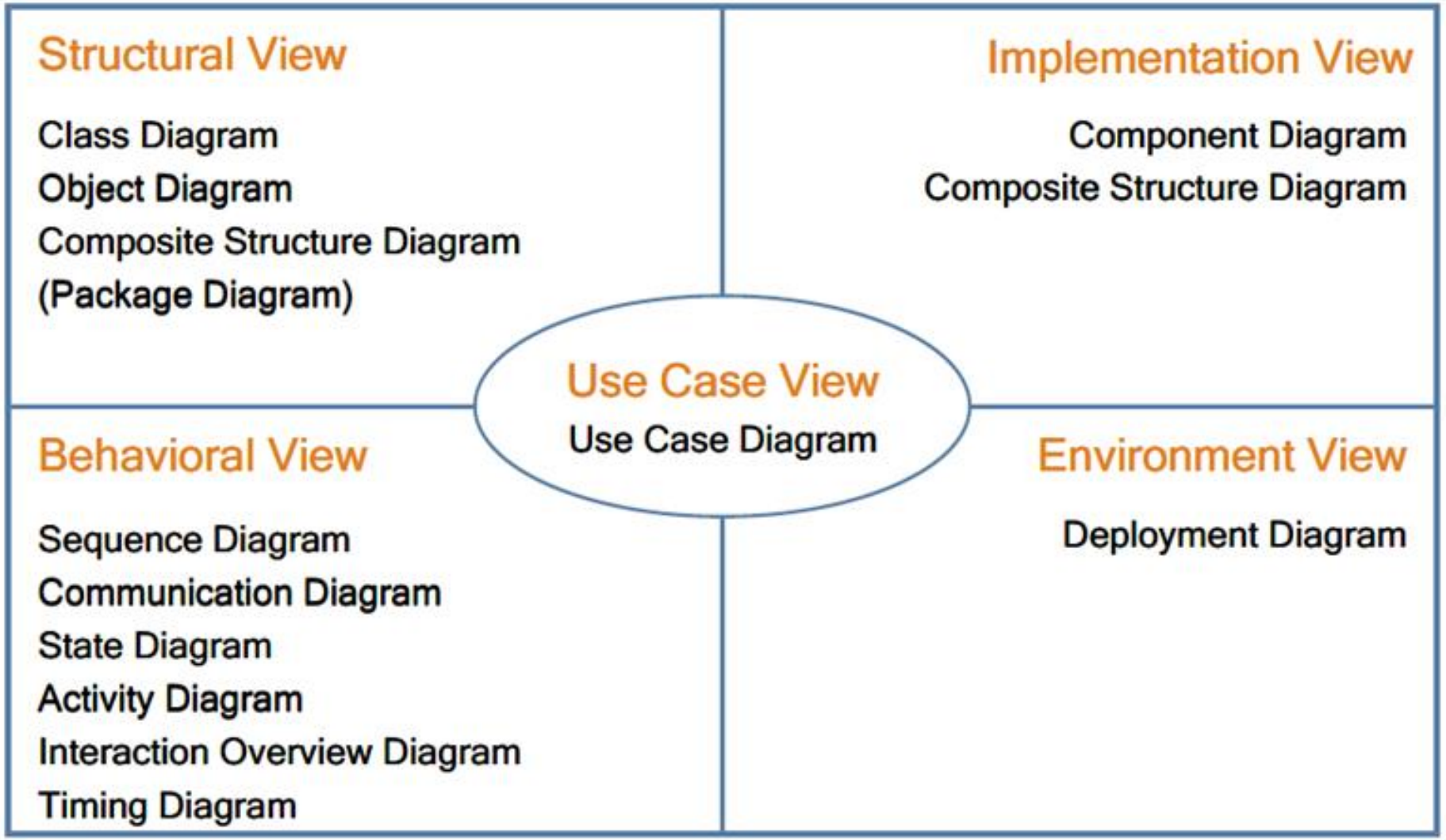
UML

UML

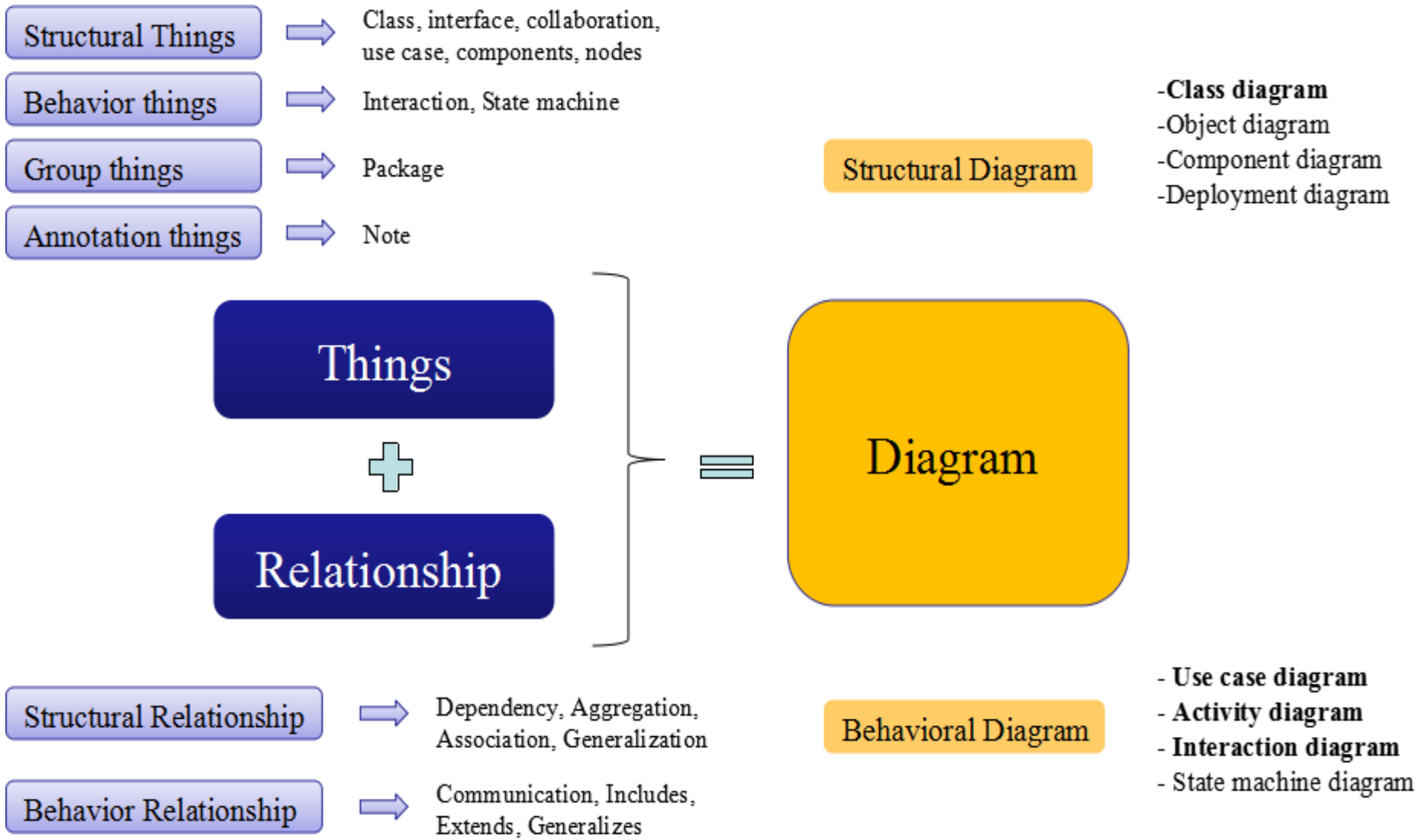
Khái niệm:

- Ngôn ngữ **mô hình hóa** hợp nhất (UML – Unified Modeling Language)
- Là ngôn ngữ được dùng để xác định, trực quan hóa, xây dựng và lập tài liệu cho các kết quả của quá trình xây dựng phần mềm
- Ưu điểm: trực quan, dễ hiểu, **theo tiêu chuẩn** và có thể được chia sẻ bởi các bên liên quan

UML


Structural View
Class Diagram
Object Diagram
Composite Structure Diagram
(Package Diagram)
Implementation View
Component Diagram
Composite Structure Diagram
Use Case View
Use Case Diagram
Behavioral View
Sequence Diagram
Communication Diagram
State Diagram
Activity Diagram
Interaction Overview Diagram
Timing Diagram
Environment View
Deployment Diagram

UML


Structural Things
Class, interface, collaboration, use case, components, nodes
Behavior things
Interaction, State machine
Group things
Package
Annotation things
Note
Things
Relationship
Diagram
Structural Diagram
-Class diagram
-Object diagram
-Component diagram
-Deployment diagram
Structural Relationship
Dependency, Aggregation, Association, Generalization
Behavior Relationship
Communication, Includes, Extends, Generalizes
Behavioral Diagram
- Use case diagram
- Activity diagram
- Interaction diagram
- State machine diagram

UML

Một số dạng biểu đồ phổ biến:

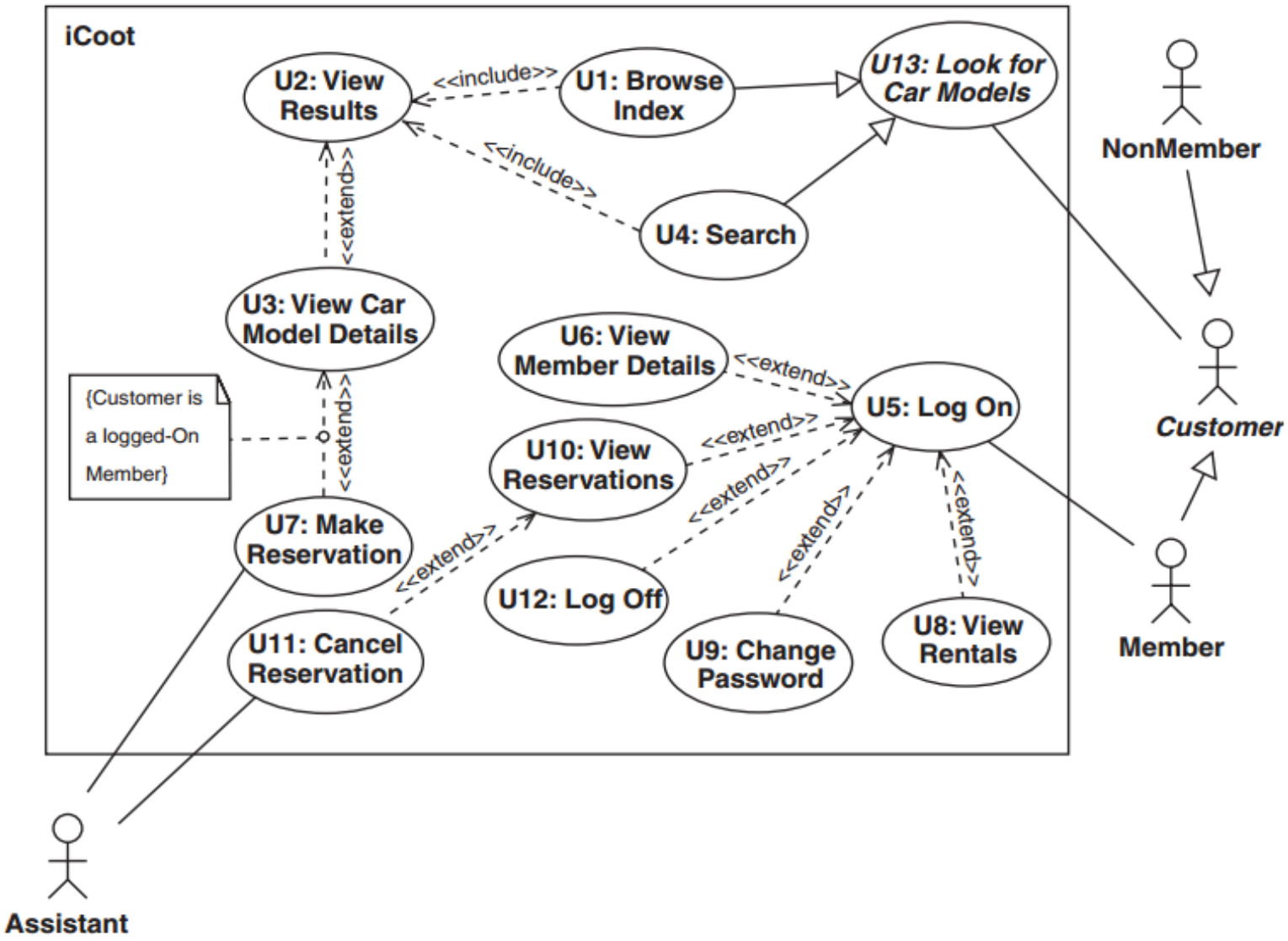
- Biểu đồ pha sử dụng (Use case Diagram)
- Biểu đồ lớp (Class Diagram)
- Biểu đồ đối tượng (Object Diagram)
- Biểu đồ tuần tự (Sequence Diagram)
- Biểu đồ tương tác (Communication Diagram)
- Biểu đồ hoạt động (Activity Diagram)
- Biểu đồ thành phần (Component Diagram)
- Biểu đồ triển khai (Deployment Diagram)

UML

Requirements	Business	Actor list (with descriptions) Use case list (with descriptions) Use case details Activity diagrams (optional) Communication diagrams (optional)	No No No Yes Yes
	System	Actor list (with descriptions) Use case list (with descriptions) Use case details Use case diagram Use case survey User interface sketches	No No No Yes No No
Analysis		Class diagram Communication diagrams	Yes Yes
Design	System	Deployment diagram Layer diagram	Yes No
	Subsystem	Class diagrams Sequence diagrams Database schema	Yes Yes No

CÁC DẠNG BIỂU ĐỒ CƠ BẢN TRONG UML

Use case Diagram



Use case Diagram

Tổng quát:

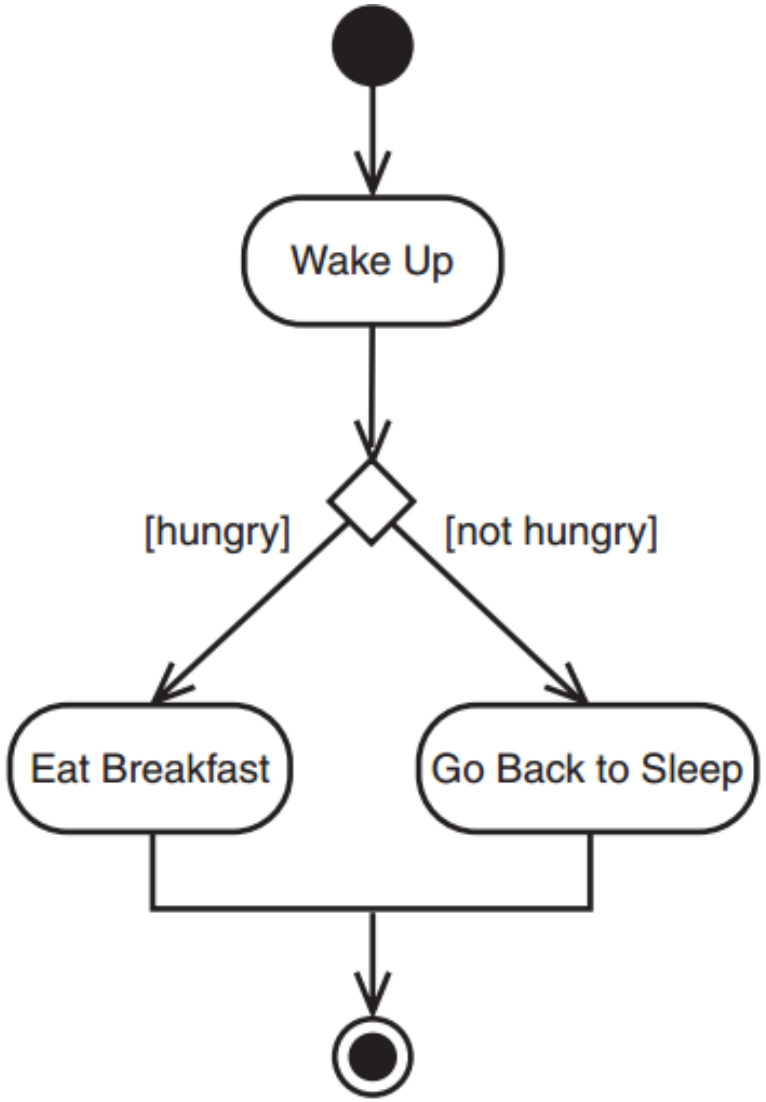
- Mô tả cách thức mà hệ thống hay nghiệp vụ được sử dụng bởi các bên liên quan
- Mô tả các trường hợp sử dụng (Use case) có mối liên hệ với nhau như thế nào
- Mô tả cách mà người dùng có thể tiếp cận các pha sử dụng đó

Use case Diagram

Tổng quát:

➢Là công cụ để thể hiện các yêu cầu từ phía người dùng nên các chức năng có trong Use case thì **phải có trong hệ thống**

Activity Diagram

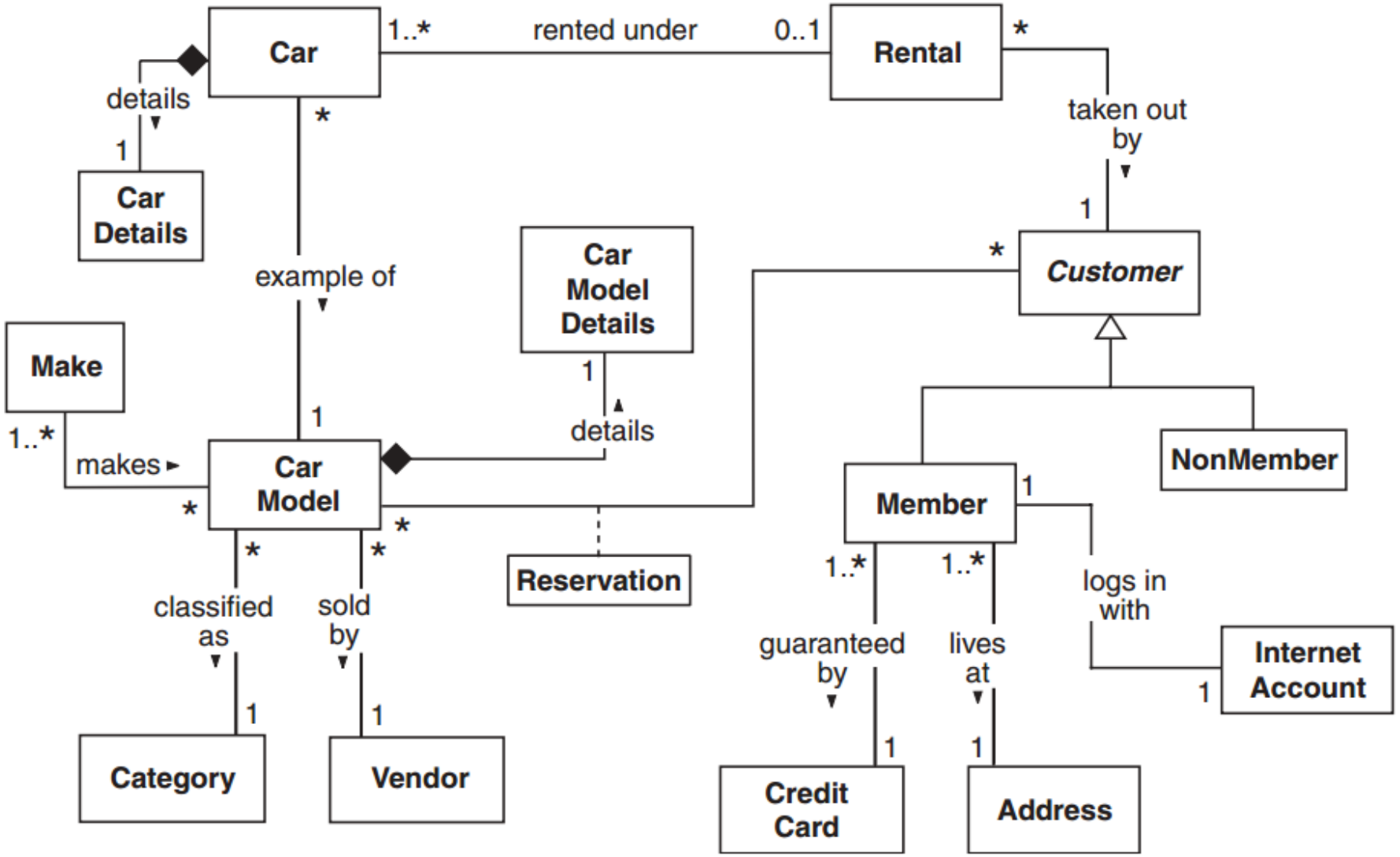


Activity Diagram

Tổng quát:

- Thể hiện những hành động xảy ra ứng với một thao tác
- Hỗ trợ lặp, tuần tự và song song
- Thường được sử dụng ngay trong giai đoạn lấy yêu cầu (requirement) để có cái nhìn rõ ràng về các hành vi mà người dùng mong muốn (ở dạng từng bước thực hiện)

Class Diagram – Analysis

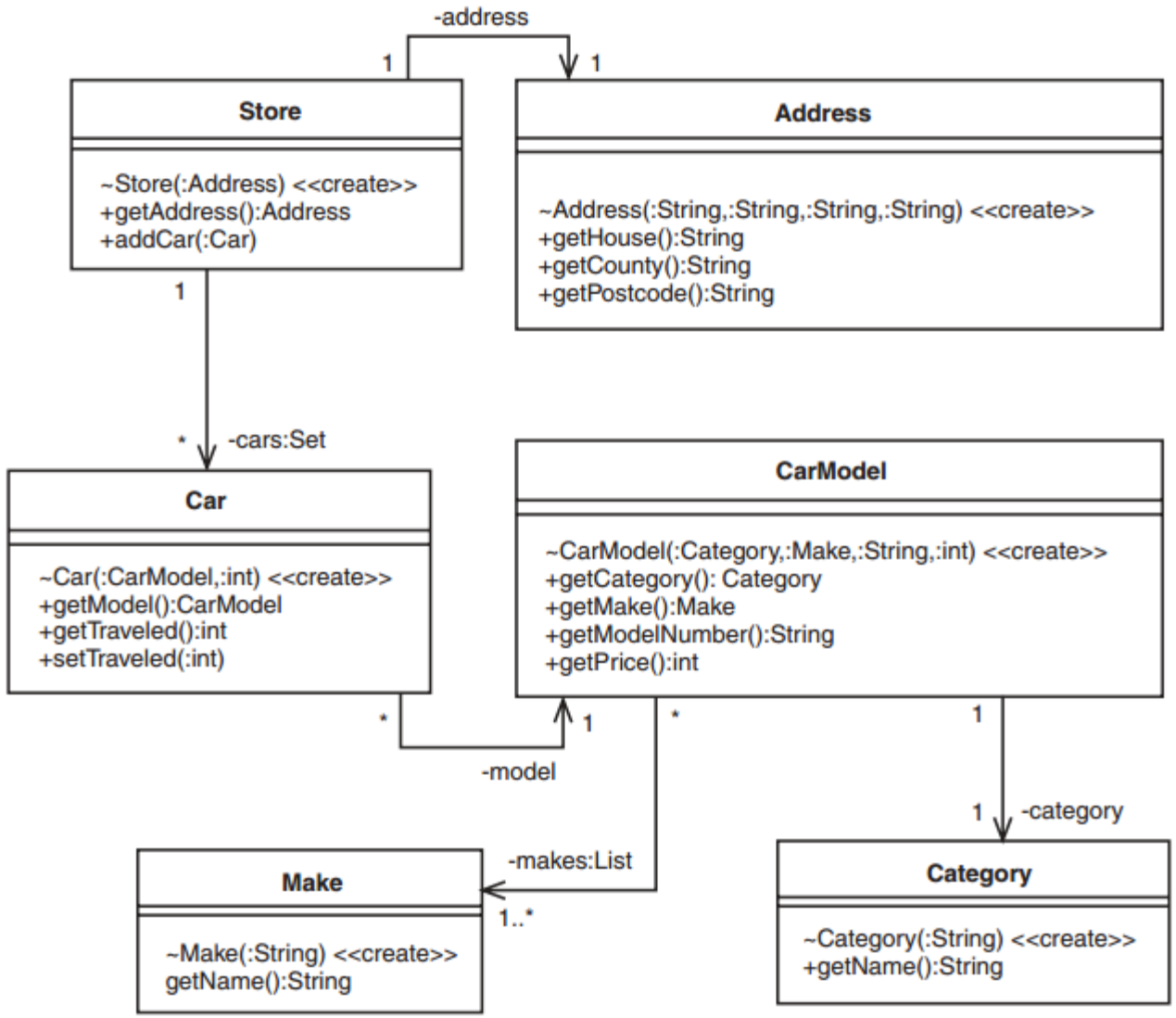


Class Diagram

Tổng quát:

- Ở giai đoạn phân tích, biểu đồ lớp chỉ ra các lớp sẽ xuất hiện trong hệ thống / nghiệp vụ đang xét
- Thể hiện mối liên hệ giữa các lớp → cũng là mối liên hệ giữa các thực thể trong từng lớp
- Ở giai đoạn **phân tích**, các lớp được thể hiện đơn giản và thường chưa bắt buộc bao gồm các thuộc tính / hành vi đi kèm

Class Diagram – Design

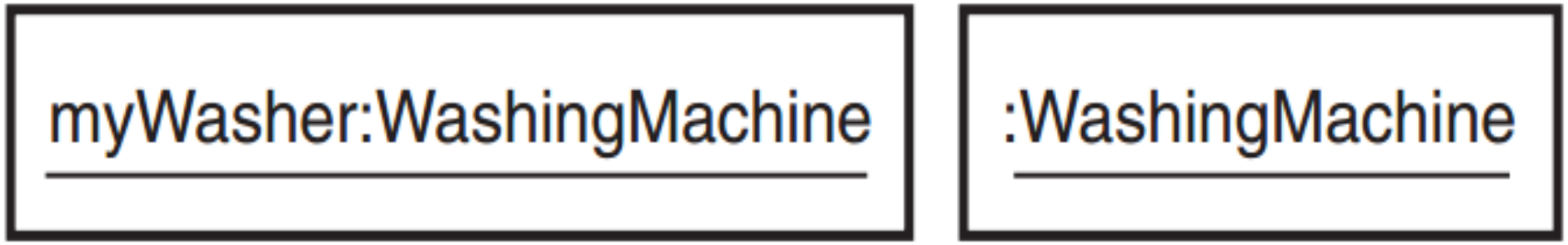


Class Diagram

Tổng quát:

- Ở giai đoạn thiết kế, biểu đồ lớp sẽ sử dụng thêm các kí hiệu mới
- Mục tiêu là thể hiện được chi tiết bên trong các lớp (gồm thuộc tính + hành vi)
- Thường được bổ sung: constructor, methods, ...

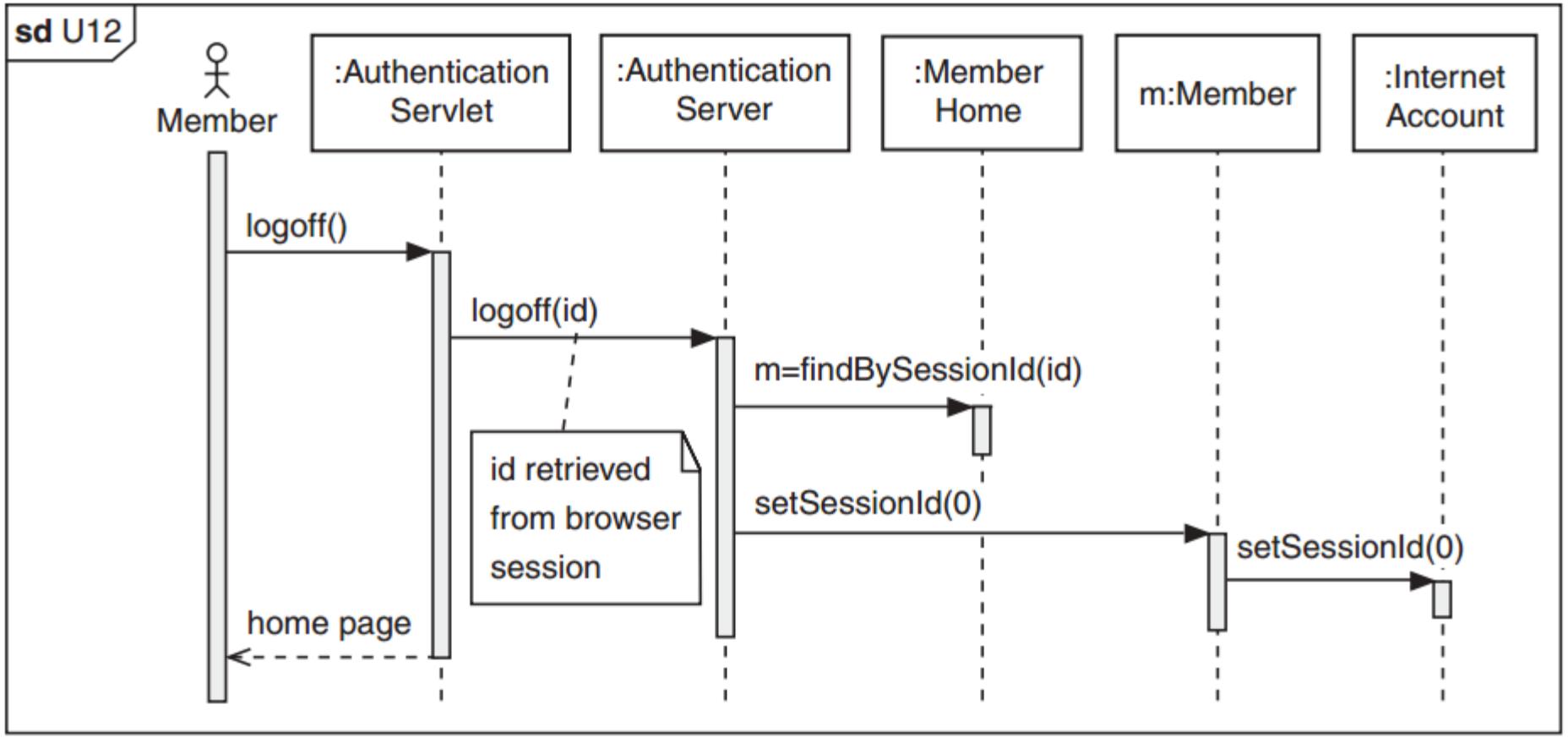
Object Diagram



Tổng quát:

- Gạch dưới tên
- Tên đối tượng cụ thể nằm phía tay trái (trước dấu hai chấm)
- Với **anonymous object**, ta không cần viết tên đối tượng

Sequence Diagram

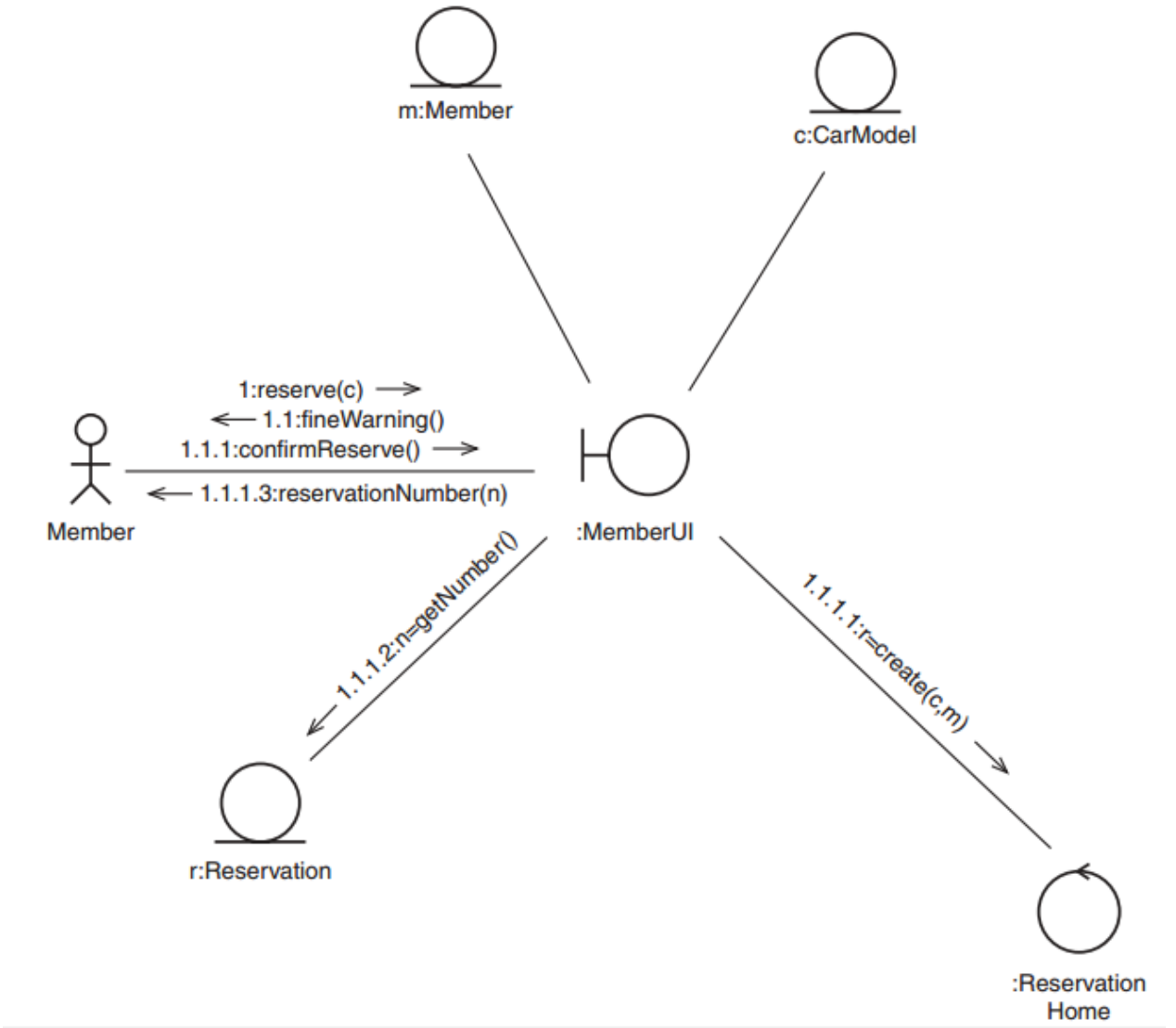


Sequence Diagram

Tổng quát:

- Biểu đồ tuần tự cho ta thấy sự tương tác giữa các objects
- Các thông điệp (Message) được biểu diễn dưới dạng các mũi tên
- Thể hiện các hành vi **theo thời gian**
- Trong phạm vi môn học ta sẽ sử dụng Sequence Diagram trong giai đoạn thiết kế

Communication Diagram

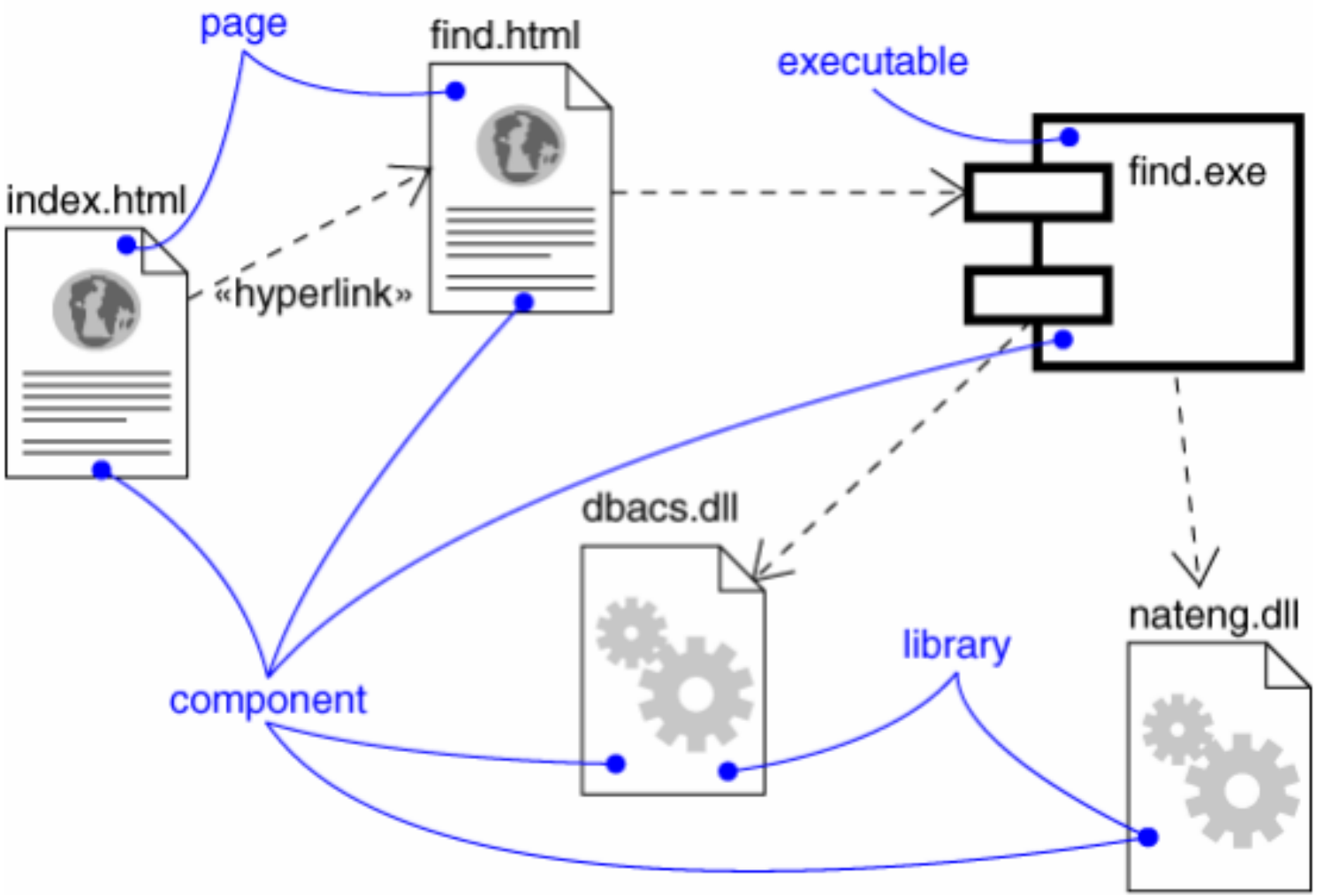


Communication Diagram

Tổng quát:

- Biểu đồ tương tác cho ta thấy sự tương tác giữa các objects
- Các thông điệp (Message) được biểu diễn dưới dạng các mũi tên
- Nhấn mạnh sự giao tiếp của các objects hơn là yếu tố thời gian

Component Diagram

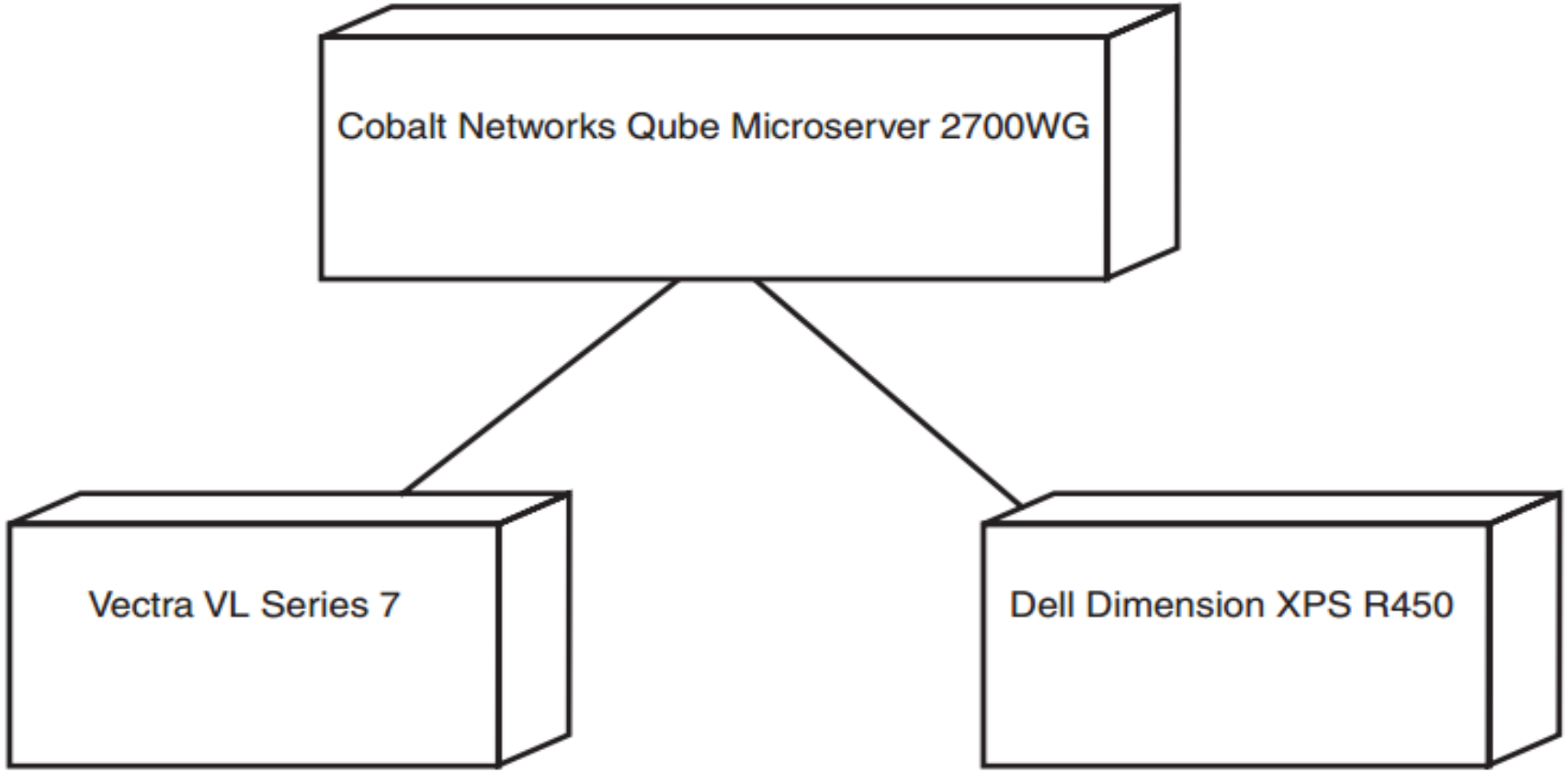


Component Diagram

Tổng quát:

- Biểu đồ thành phần cho ta cái nhìn vật lý về hệ thống
- Cho ta thấy các thành phần phần mềm và quan hệ giữa chúng trong hệ thống
- Mỗi thành phần có thể là: mã nguồn, file thực thi, thư viện, ...

Deployment Diagram

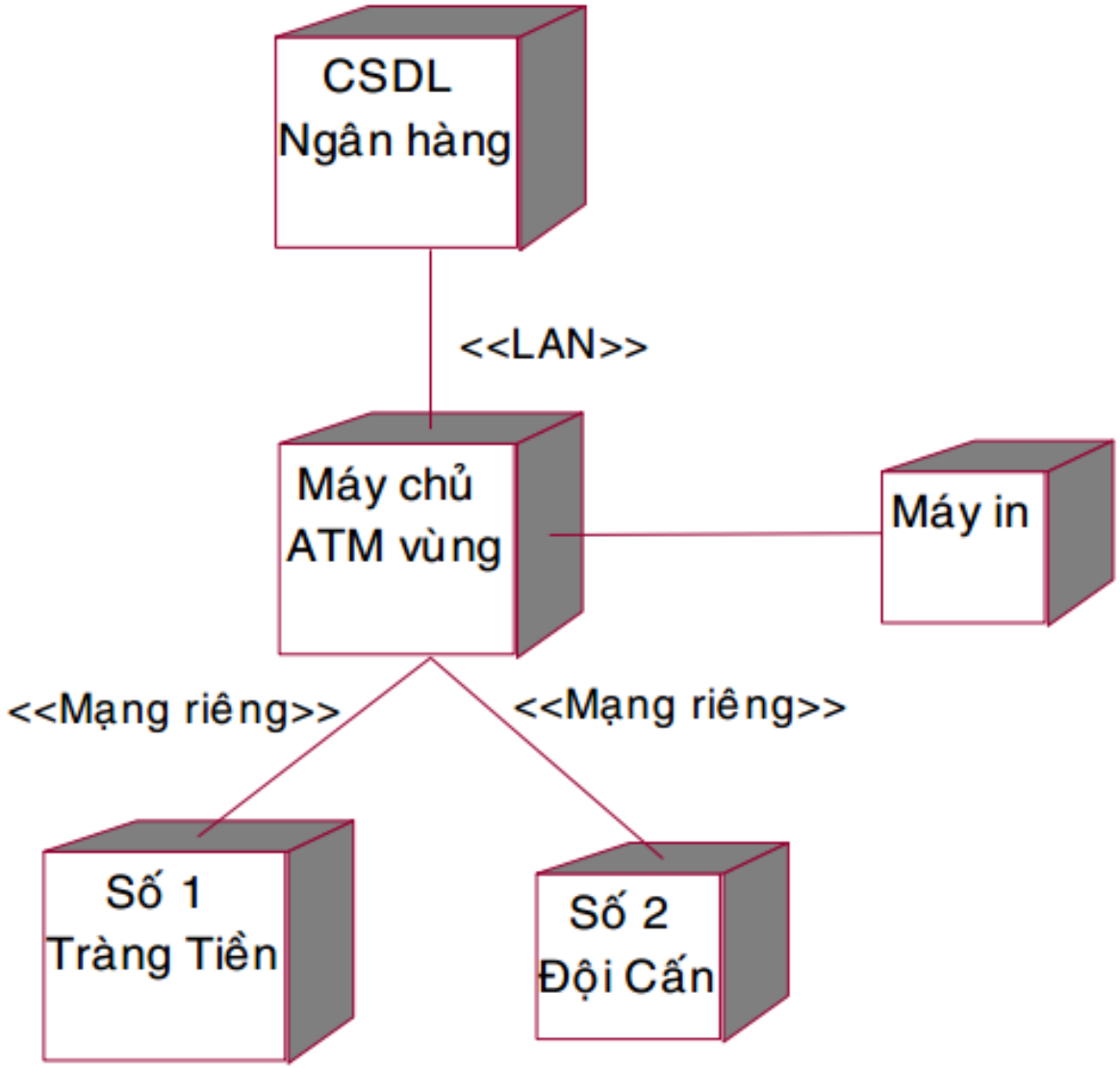


Deployment Diagram

Tổng quát:

- Biểu đồ triển khai cho thấy hệ thống cuối cùng được triển khai như thế nào
- Gồm các yếu tố như: máy móc, process, file

Deployment Diagram



Deployment Diagram

Tổng quát:

- Biểu đồ trên hình này cho thấy Máy trạm ATM sẽ chạy trên nhiều địa điểm khác nhau.
- Chúng giao tiếp với Máy chủ ATM qua mạng riêng. Máy chủ ATM sẽ giao tiếp với các Máy chủ CSDL thông qua mạng LAN.
- Như vậy, hệ thống ATM này có kiến trúc ba tầng: một tầng là CSDL, một tầng là máy chủ, tầng còn lại là máy trạm.

NẮM BẮT YÊU CẦU

Nắm bắt yêu cầu

Tại sao cần nắm bắt yêu cầu ?:

Nắm bắt yêu cầu

Mục tiêu:

- Kiểm tra lại ngữ cảnh phát triển phần mềm: làm rõ lý do tại sao cần phát triển phần mềm.
- Đảm bảo rằng ta đã hiểu rõ nghiệp vụ và cái hiểu này trùng khớp với các bên liên quan
- Mô tả được yêu cầu hệ thống (thông qua tài liệu **SRS**): chức năng + ràng buộc

Nắm bắt yêu cầu

Mục tiêu:

- Business Requirement: nắm được các các nghiệp vụ và tài liệu hóa
- System Requirement: chọn lựa các chức năng nên và không nên được cung cấp bởi hệ thống
- System Requirement thường chia làm 2 nhóm: yêu cầu chức năng và yêu cầu phi chức năng

MÔ HÌNH HÓA TRƯỜNG HỢP SỬ DỤNG – USE CASE

Use Case

Khái niệm:

- Được Ivar Jacobson đề xuất
- Mô tả các bên tương tác với hệ thống
- Mô tả tương tác giữa người sử dụng với hệ thống phần mềm để thực hiện thao tác giải quyết một công việc cụ thể nào đó

Use Case

Khái niệm:

- Không cho biết chi tiết thực hiện bên trong
- Không phải là bản thiết kế, không phải là bản cài đặt
- Tiến trình của hệ thống được chia nhỏ thành các Use Cases nhằm nhận ra được các bộ phận của hệ thống một cách rõ ràng và đầy đủ → dễ dàng hơn trong việc xây dựng và quản lý

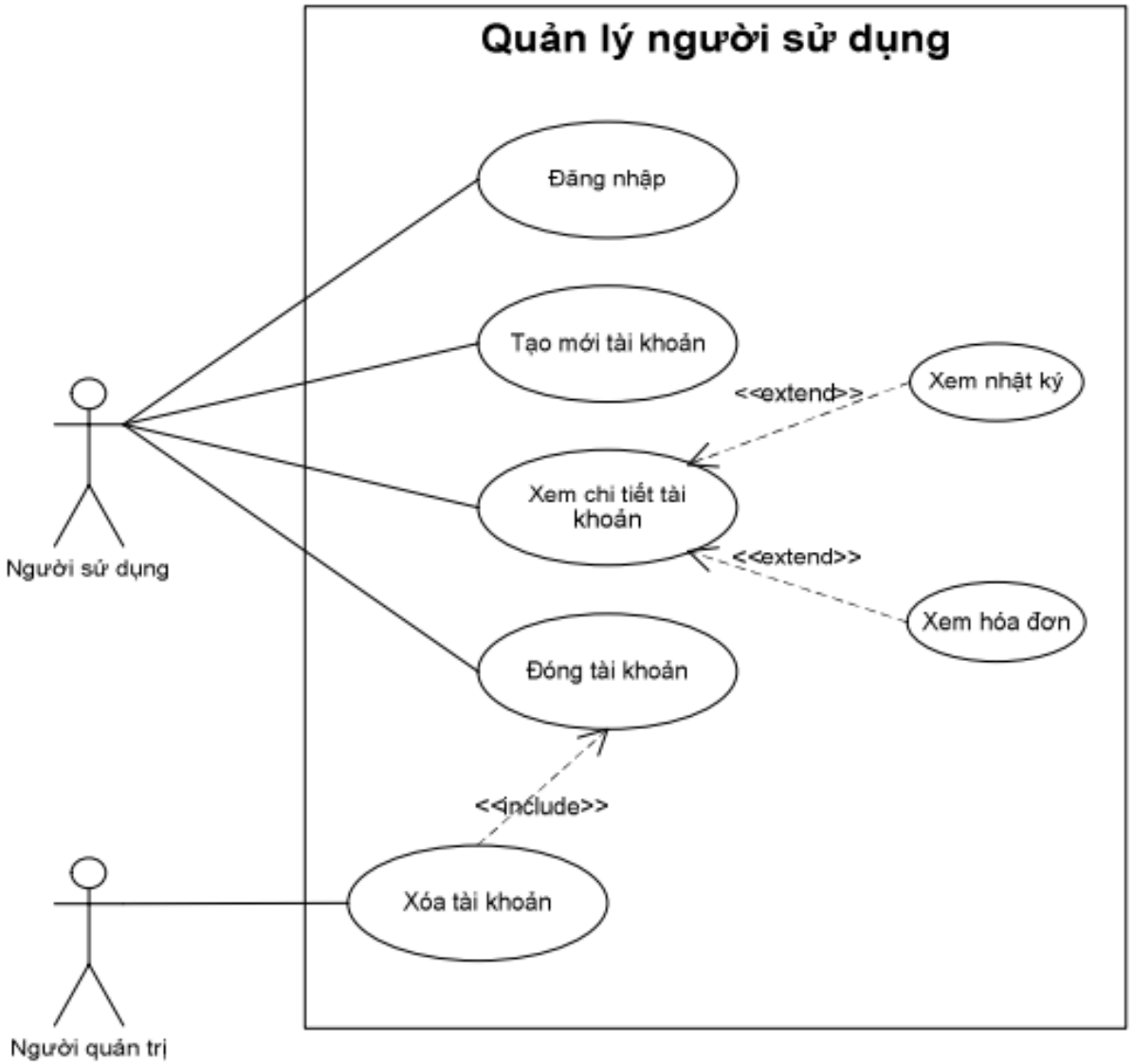
Use Case

Mục tiêu:

- Hình thành quyết định và mô tả yêu cầu chức năng hệ thống.
- Là kết quả của thỏa thuận giữa khách hàng và người phát triển hệ thống phần mềm.
- Mô tả rõ ràng và nhất quán cái hệ thống sẽ làm, sao cho mô hình có khả năng được sử dụng xuyên suốt quá trình phát triển
- Làm cơ sở để kiểm tra + Có khả năng mở rộng

Use Case

Ví dụ:



USE CASE DIAGRAM

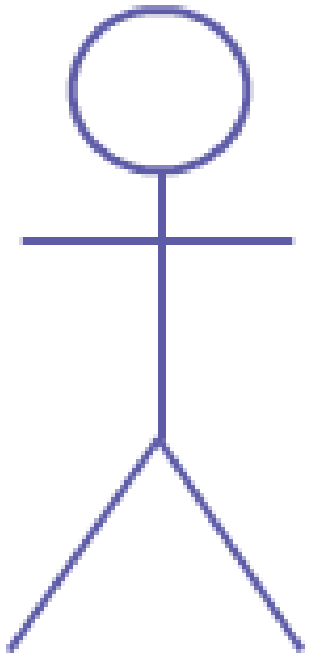
Use Case Diagram

Gồm các thành phần chính:

- Actor: tác nhân
- Use Case: trường hợp sử dụng
- System: hệ thống đang xét
- Các mối liên hệ giữa các thành phần

Actor

Kí hiệu trong UC Diagram:



Actor

Actor

Khái niệm:

- Là người sử dụng hệ thống.
- Người sử dụng hệ thống có thể là người, máy, hệ thống khác hoặc một hệ thống con trong mô hình.
- Bất cứ tương tác nào từ bên ngoài hay bên trong hệ thống đều được gọi là **Actor**

Actor

Khái niệm:

- Một tác nhân tương tác với hệ thống thông qua gửi và nhận thông điệp (message)
- Có thể phân loại: tác nhân chính (Primary Actor) hay tác nhân phụ (Secondary Actor)

Actor

Cách để xác định Actor – Trả lời các câu hỏi:

- **Ai** là người sử dụng chức năng chính của hệ thống ?
- **Ai** cần sự hỗ trợ từ hệ thống để thực hiện công việc thường nhật của họ ?
- **Ai** phải thực hiện công việc bảo dưỡng, quản trị và giữ cho hệ thống hoạt động ?
- Hệ thống đang xây dựng cần tương tác với những **hệ thống khác** hay không ?

Actor

Ví dụ:

Giả sử ta đang xây dựng một phần mềm quản lý máy ATM. Máy cho phép người sử dụng mở tài khoản, kiểm tra, rút tiền và sửa password. Nhân viên ngân hàng có thể sử dụng phần mềm để kiểm tra thông tin về số tiền trong máy. Hãy xác định các tác nhân có trong hệ thống ?

Actor

Ví dụ:

Giả sử ta đang xây dựng một phần mềm quản lý máy ATM. Máy cho phép **người sử dụng** mở tài khoản, kiểm tra, rút tiền và sửa password. **Nhân viên ngân hàng** có thể sử dụng phần mềm để kiểm tra thông tin về số tiền trong máy.

Actor

Ví dụ:

Ta cần xây dựng một chương trình quản lý thư viện. Chương trình cho phép bạn đọc tra cứu các thông tin về tài liệu, đặt mượn. Chương trình cũng cho phép thủ thư kiểm kê, xác nhận việc mượn sách. Hãy xác định các tác nhân có trong hệ thống ?

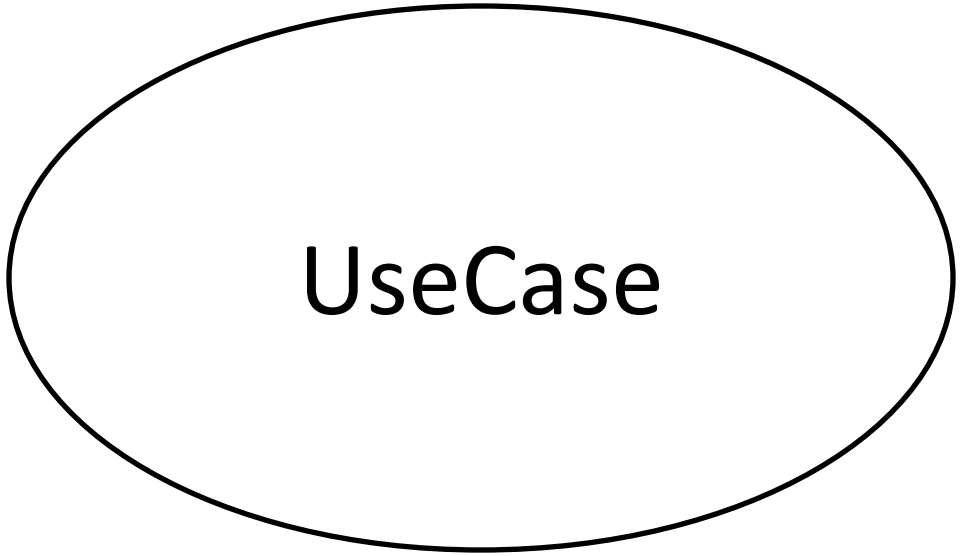
Actor

Ví dụ:

Ta cần xây dựng một chương trình quản lý thư viện. Chương trình cho phép **bạn đọc** tra cứu các thông tin về tài liệu, đặt mượn. Chương trình cũng cho phép **thủ thư** kiểm kê, xác nhận việc mượn sách.

Use Case

Kí hiệu trong UC Diagram:



Use Case

Khái niệm:

- Là một chức năng của hệ thống
- Được hình thành dựa trên quá trình mô tả các tương tác giữa Actor với hệ thống.
- Một Use Case bao giờ cũng được gây ra bởi một tác nhân, được thực hiện nhân danh một tác nhân nào đó

Use Case

Cách để xác định Use Case – Trả lời các câu hỏi :

- Tác nhân yêu cầu hệ thống thực hiện chức năng gì?
- Tác nhân cần đọc, tạo lập, bãi bỏ, lưu trữ, sửa đổi các thông tin nào trong hệ thống?
- Có cần thông báo cho tác nhân về sự kiện xảy ra trong hệ thống ? Có cần tác nhân thông báo cái gì đó cho hệ thống?

Use Case

Ví dụ:

Giả sử ta đang xây dựng một phần mềm quản lý máy ATM. Máy cho phép người sử dụng mở tài khoản, kiểm tra, rút tiền và sửa password. Nhân viên ngân hàng có thể sử dụng phần mềm để kiểm tra thông tin về số tiền trong máy. Hãy xác định các Use Case có trong hệ thống ?

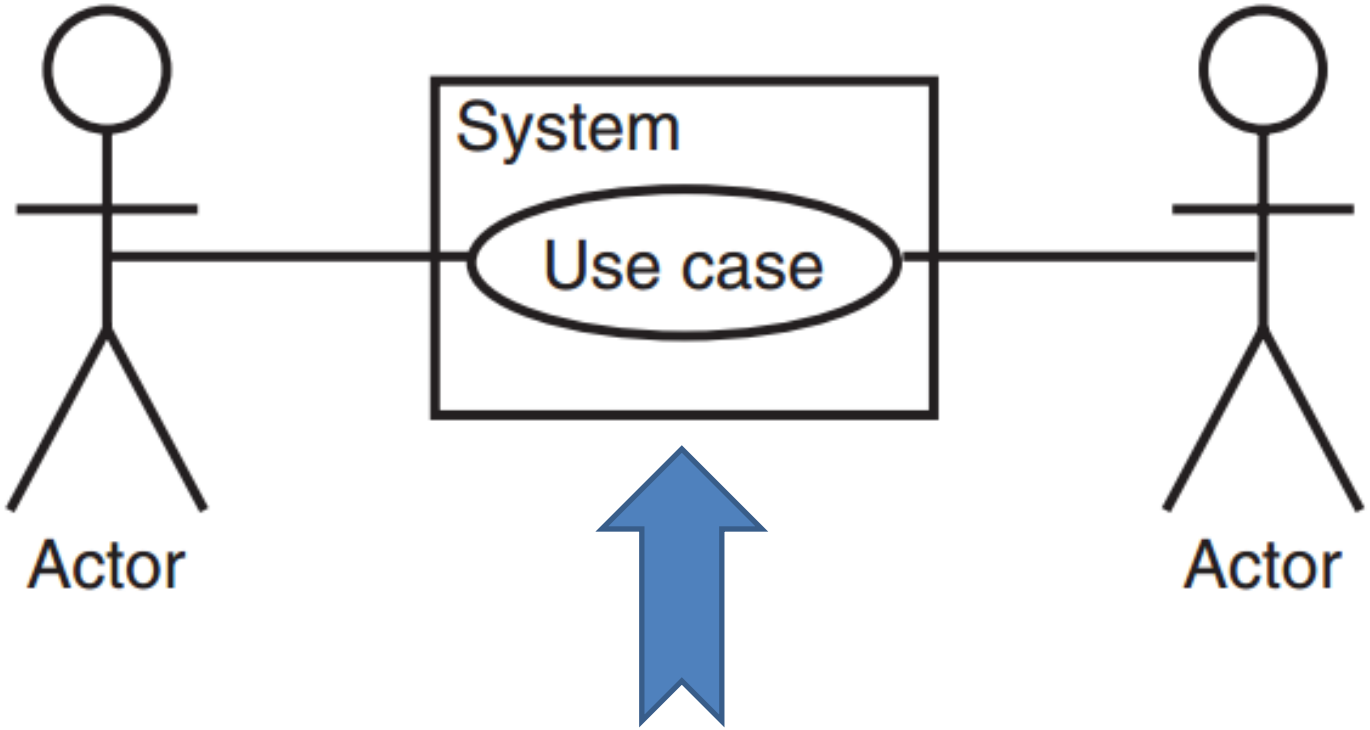
Use Case

Ví dụ:

Giả sử ta đang xây dựng một phần mềm quản lý máy ATM. Máy cho phép người sử dụng **mở tài khoản, kiểm tra, rút tiền và sửa password**. Nhân viên ngân hàng có thể sử dụng phần mềm để **kiểm tra thông tin** về số tiền trong máy.

System - Boundary

Kí hiệu trong UC Diagram:



System - Boundary

Khái niệm:

- Là ranh giới giữa hệ thống và bên ngoài
- Actor thường nằm bên ngoài system
- Use Case nằm trong system

Bài tập

Nội dung:

Một công ty xây dựng hệ thống bán và đặt vé. Người sử dụng có thể trực tiếp đặt vé trước, hủy đặt vé, thay đổi hạng vé. Các thao tác trên được hỗ trợ bởi một thư kí phòng vé.

Người dùng có thể đăng kí một tài khoản, sau đó đặt vé online bằng tài khoản đó. Người dùng có thể thanh toán online bằng thẻ tín dụng hoặc tài khoản ngân hàng. Hệ thống cũng cho phép kiểm tra thông tin tàu, vé và giờ khởi hành

Bài tập

Yêu cầu:

- Xác định hệ thống
- Xác định các Actor
- Xác định các UseCase tương ứng có thể có

CÁC DẠNG QUAN HỆ TRONG USE CASE DIAGRAM

Các dạng quan hệ trong UC Diagram

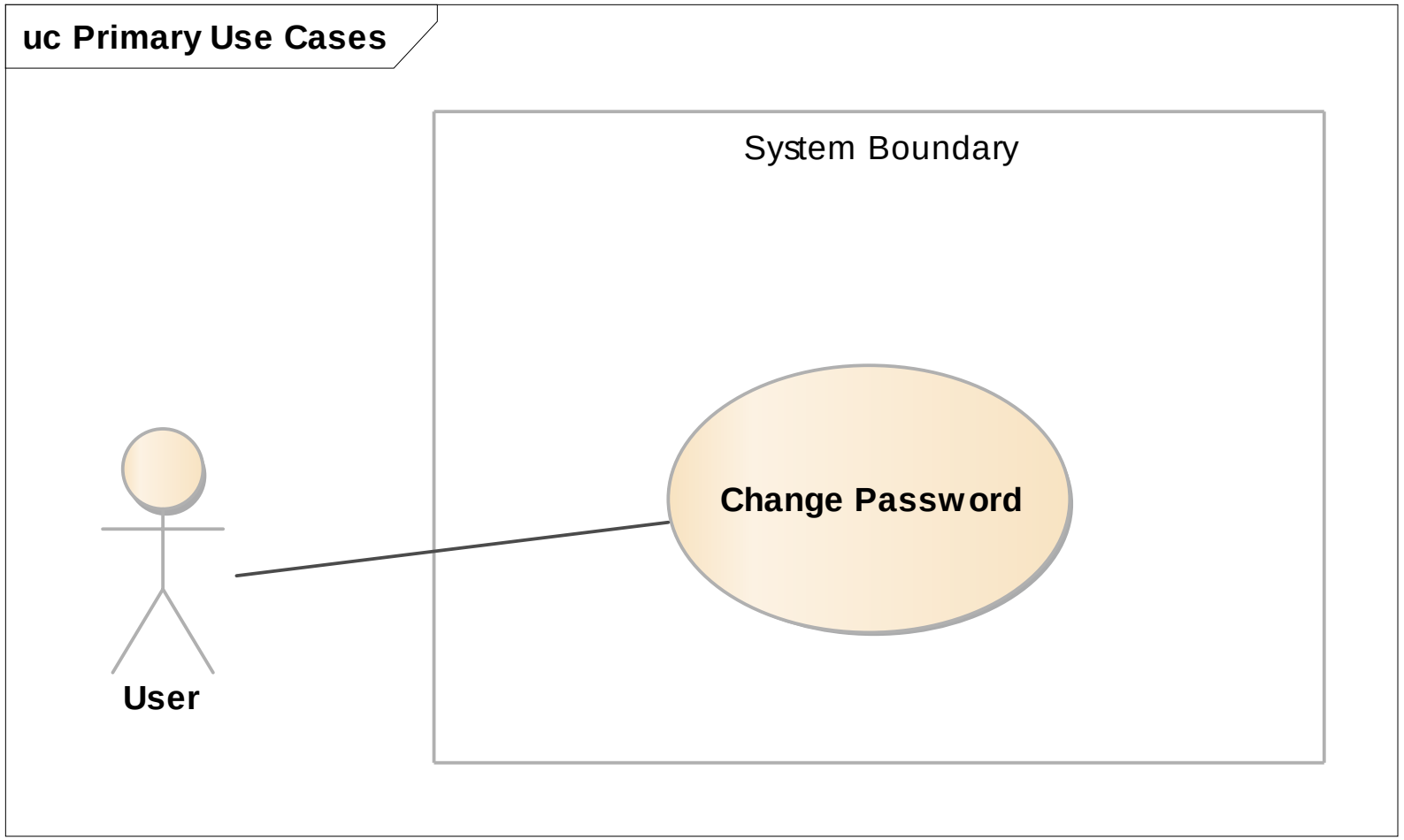
Gồm 4 dạng chính:

- Association
- Include
- Extend
- Generalization/Specializtion

ASSOCIATION

Các dạng quan hệ trong UC Diagram

Association:



Các dạng quan hệ trong UC Diagram

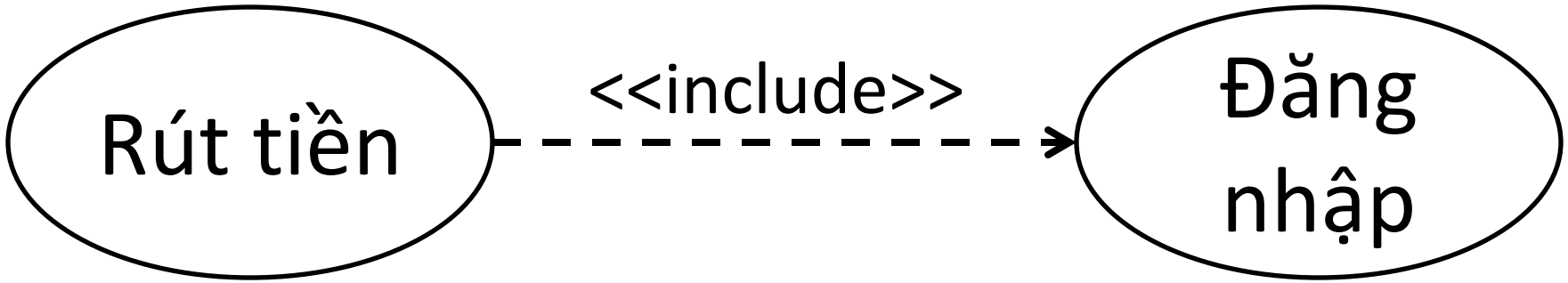
Association:

- Là mối liên hệ giữa **Actors và Use cases**
- Thể hiện tương tác giữa actors và use cases
- Đôi khi có mũi tên (thể hiện hướng thực thi)
- Cho biết Actor nào thực hiện Use Cases nào hoặc Use Cases nào có tác động đến Actor nào

INCLUDE

Các dạng quan hệ trong UC Diagram

Include – chứa:



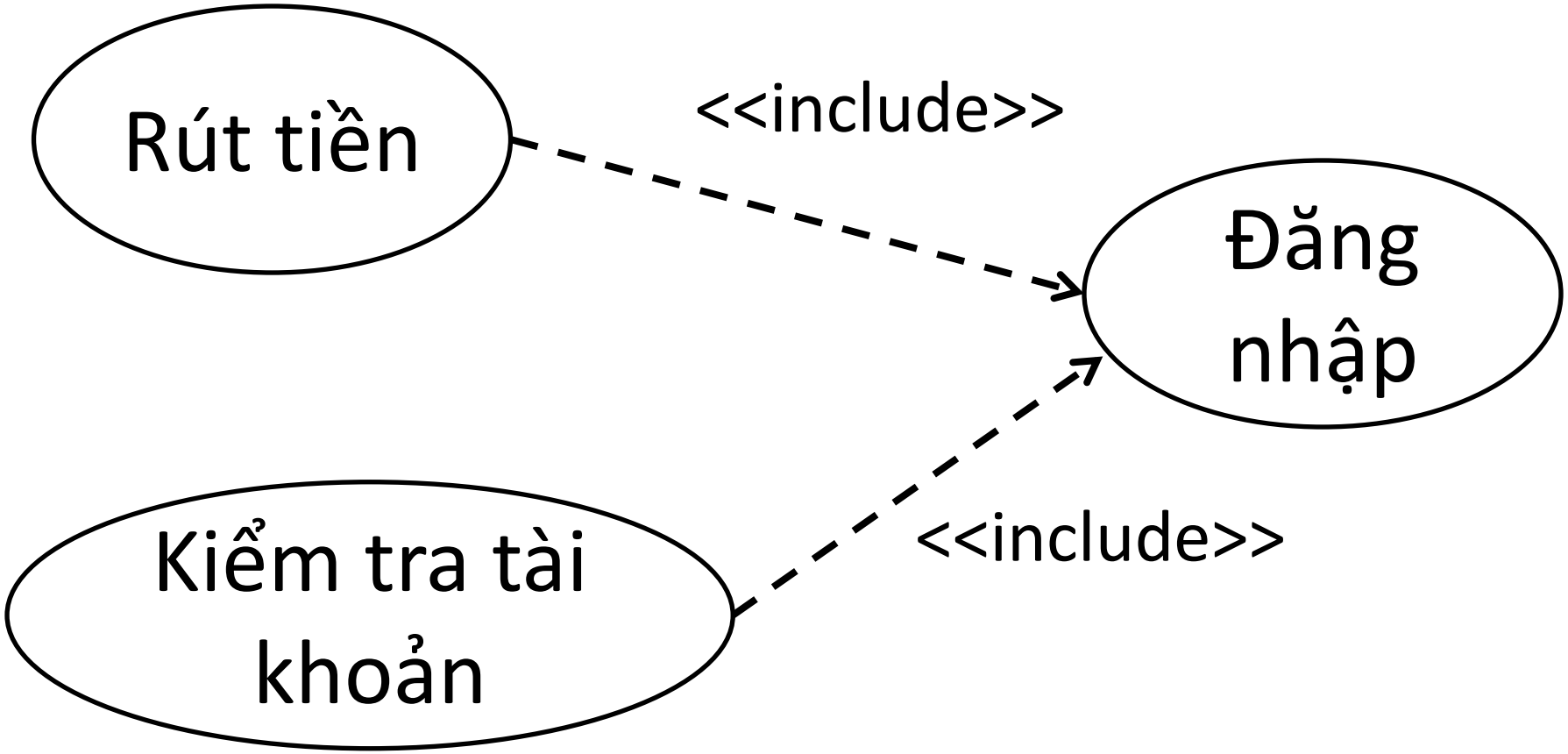
Các dạng quan hệ trong UC Diagram

Include:

- Là mối liên hệ giữa use case – use case
- Hành vi của use case include sẽ được **chèn** vào hành vi của use case được include
- Ở đây Use Case "Đăng nhập" là một phần không thể thiếu của Use Case "Rút tiền"
- Use Case "Rút tiền" **bắt buộc** phải thực hiện "Đăng nhập"
- **Tái sử dụng các chức năng phổ biến**

Các dạng quan hệ trong UC Diagram

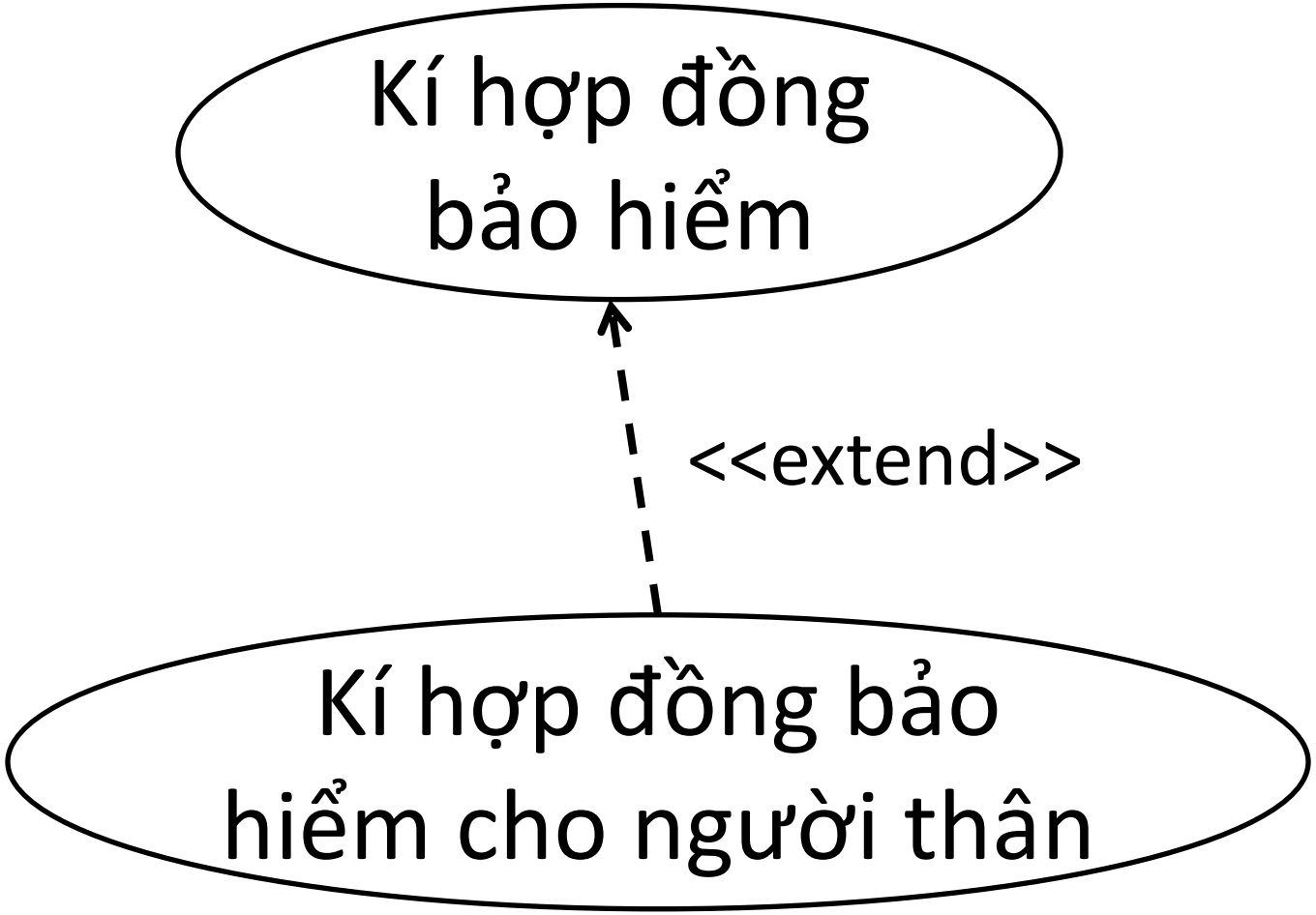
Include – chứa:



EXTEND

Các dạng quan hệ trong UC Diagram

Extension – mở rộng:



Các dạng quan hệ trong UC Diagram

Extension:

- Là mối quan hệ giữa Use Case – Use Case
- Hành vi của use case mở rộng (extension use case) được chèn vào hành vi của use case cơ sở (base use case) dưới một số điều kiện
- Extension use case **không bắt buộc** phải xảy ra
- Việc mở rộng chỉ có thể xảy ra ở một số điểm đã xác định trước. Các điểm này gọi là **extension points**

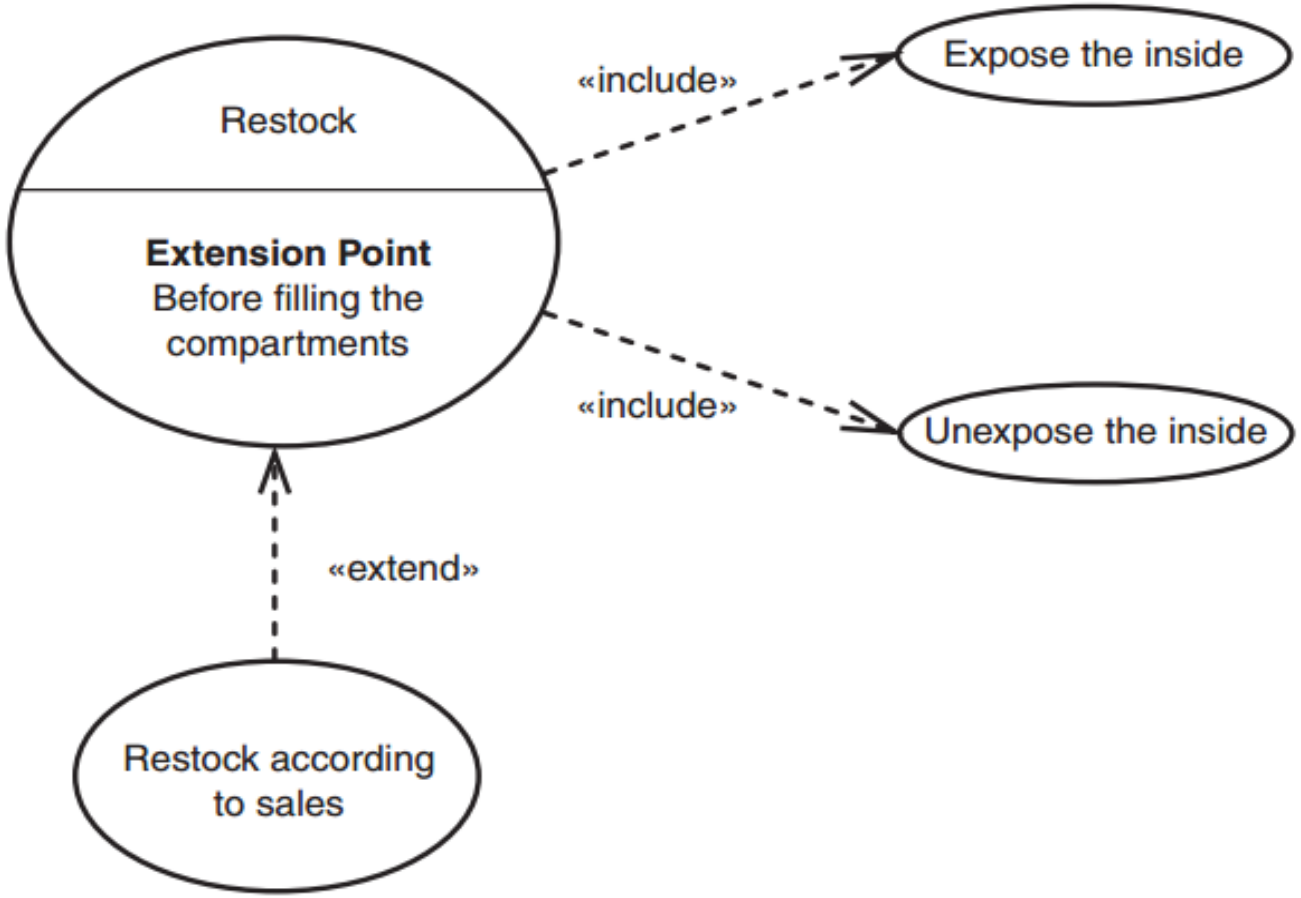
Các dạng quan hệ trong UC Diagram

Extension:

- Trong ví dụ trên, Use Case “Kí hợp đồng bảo hiểm” là Use Case cơ sở. Ở đây Actor hoàn toàn có thể chỉ cần thực hiện chức năng này mà thôi
- Trong lúc thực hiện Use Case “Kí hợp đồng bảo hiểm”, Actor có thể chọn thêm option “Kí hợp đồng bảo hiểm cho người thân” → ta nói rằng UseCase đó “mở rộng” chức năng cho Use Case cơ sở

Các dạng quan hệ trong UC Diagram

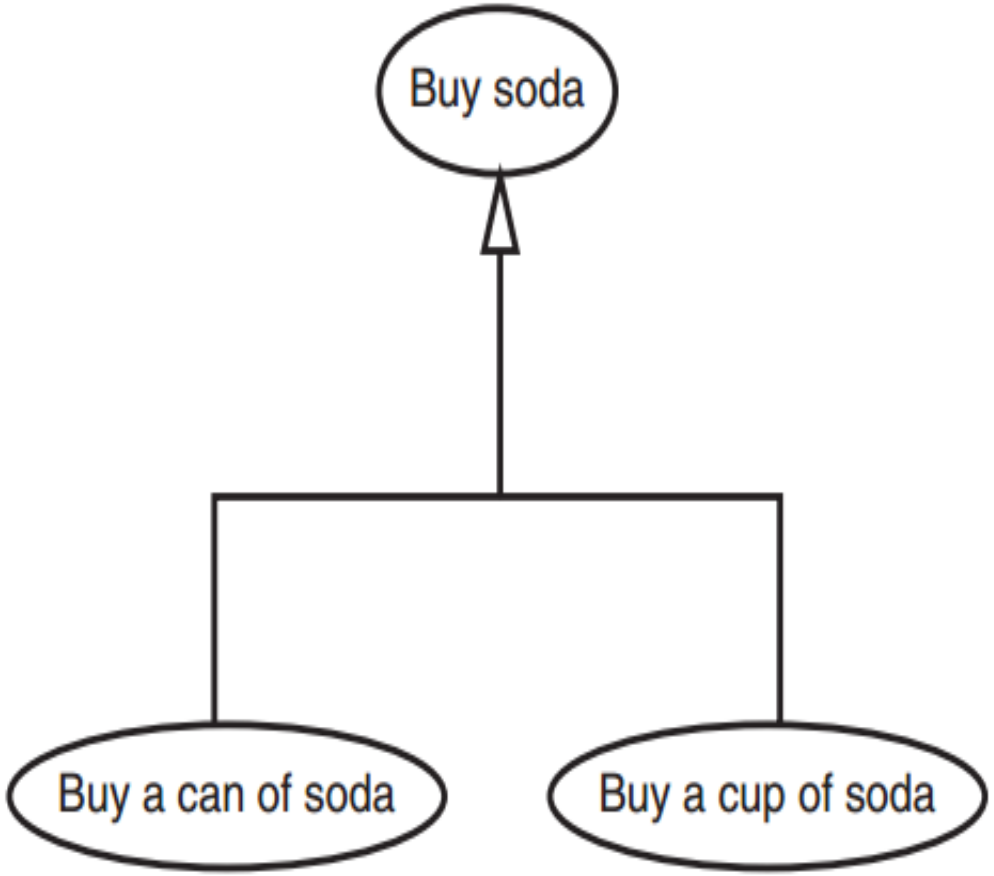
Extension – mở rộng (lưu ý chiều mũi tên):



GENERALIZATION

Các dạng quan hệ trong UC Diagram

Generalization



Các dạng quan hệ trong UC Diagram

Generalization:

- Là mối liên hệ giữa use case và use case, actor và actor
- Thể hiện tính chất thừa kế trong lập trình hướng đối tượng
- VD: Actor B thừa kế từ Actor A thì tất cả các quyền của Actor A sẽ được truyền cho Actor B

CASE STUDY 1

Case Study 1

Phần mềm quản lý máy bán Soda:

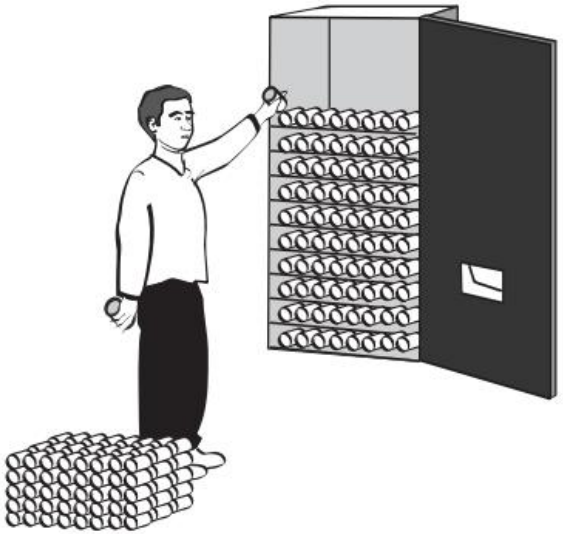
Case Study 1

Mô tả:

- Một khách hàng đến và mua một ly soda.
- Một người cung cấp hàng (supplier) có thể đến và đưa thêm soda vào máy
- Một người thu tiền (collector) có thể đến và lấy tiền từ máy ra
- Biết rằng để lấy tiền hoặc thêm soda thì đều phải thực hiện 2 hành vi là “mở máy soda” và “đóng máy soda” lại

Case Study 1

Xác định các Actor ?:



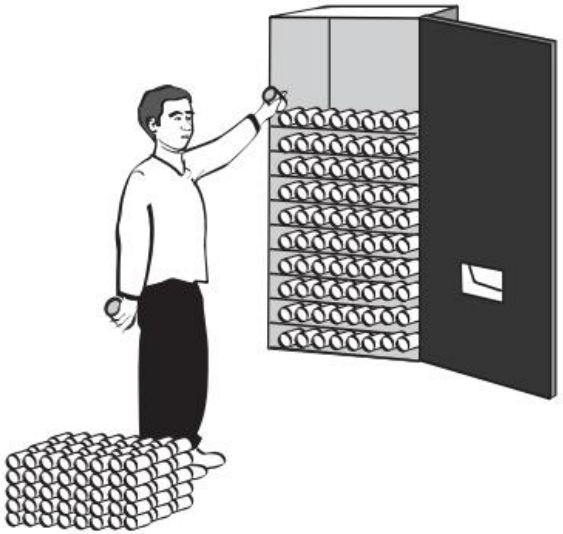
Case Study 1

Gồm 3 Actor chính:

- Khách hàng (Customer)
- Người cung cấp hàng (Supplier)
- Người thu tiền (Collector)

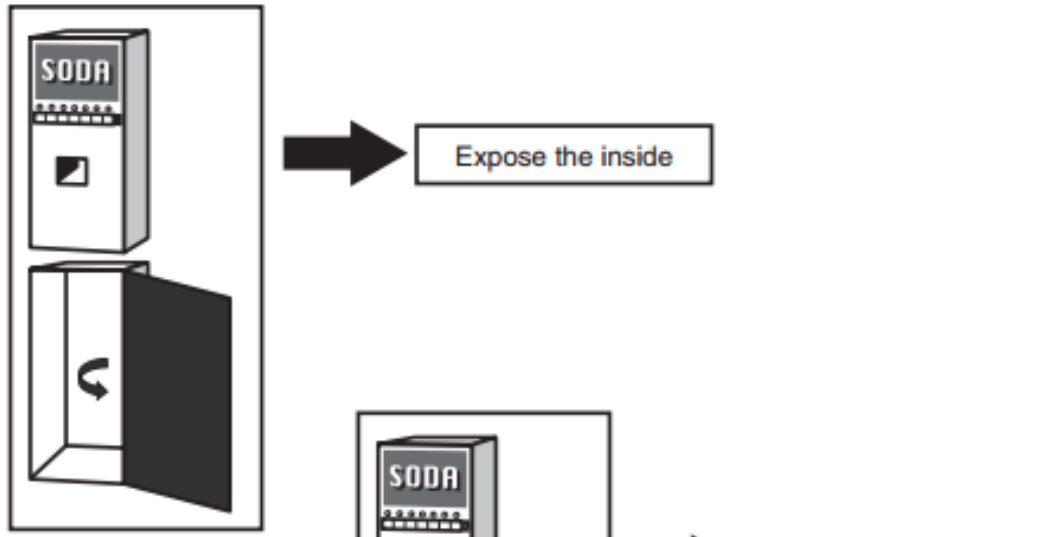
Case Study 1

Xác định các Use Case?:



Case Study 1

Xác định các Use Case?:



Case Study 1

Gồm 3 Use Cases chính:

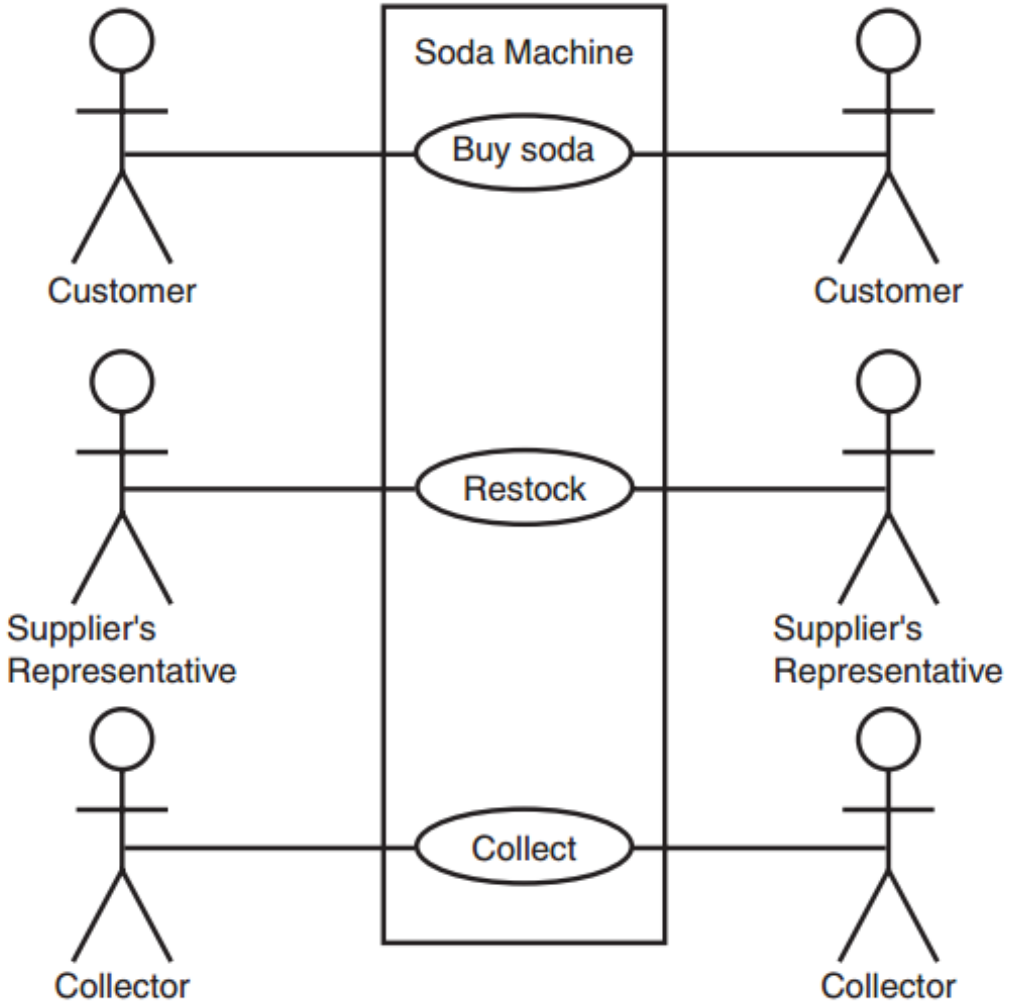
- Mua soda (Buy soda)
- Đưa soda vào máy (Restock)
- Thu tiền (Collect)

Gồm 2 Use Cases phụ:

- Mở máy (Expose the inside)
- Đóng máy (Unexpose the inside)

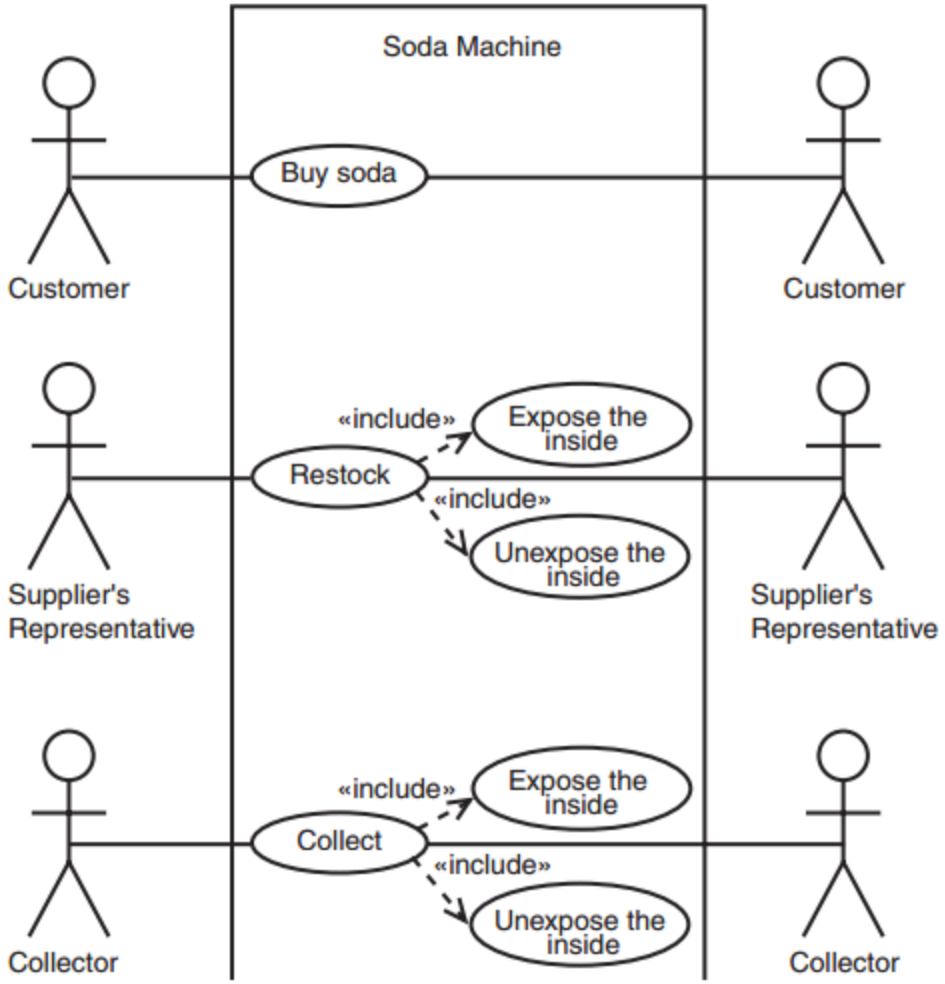
Case Study 1

Xác định các Mối liên hệ?:



Case Study 1

Xác định các Mối liên hệ?:

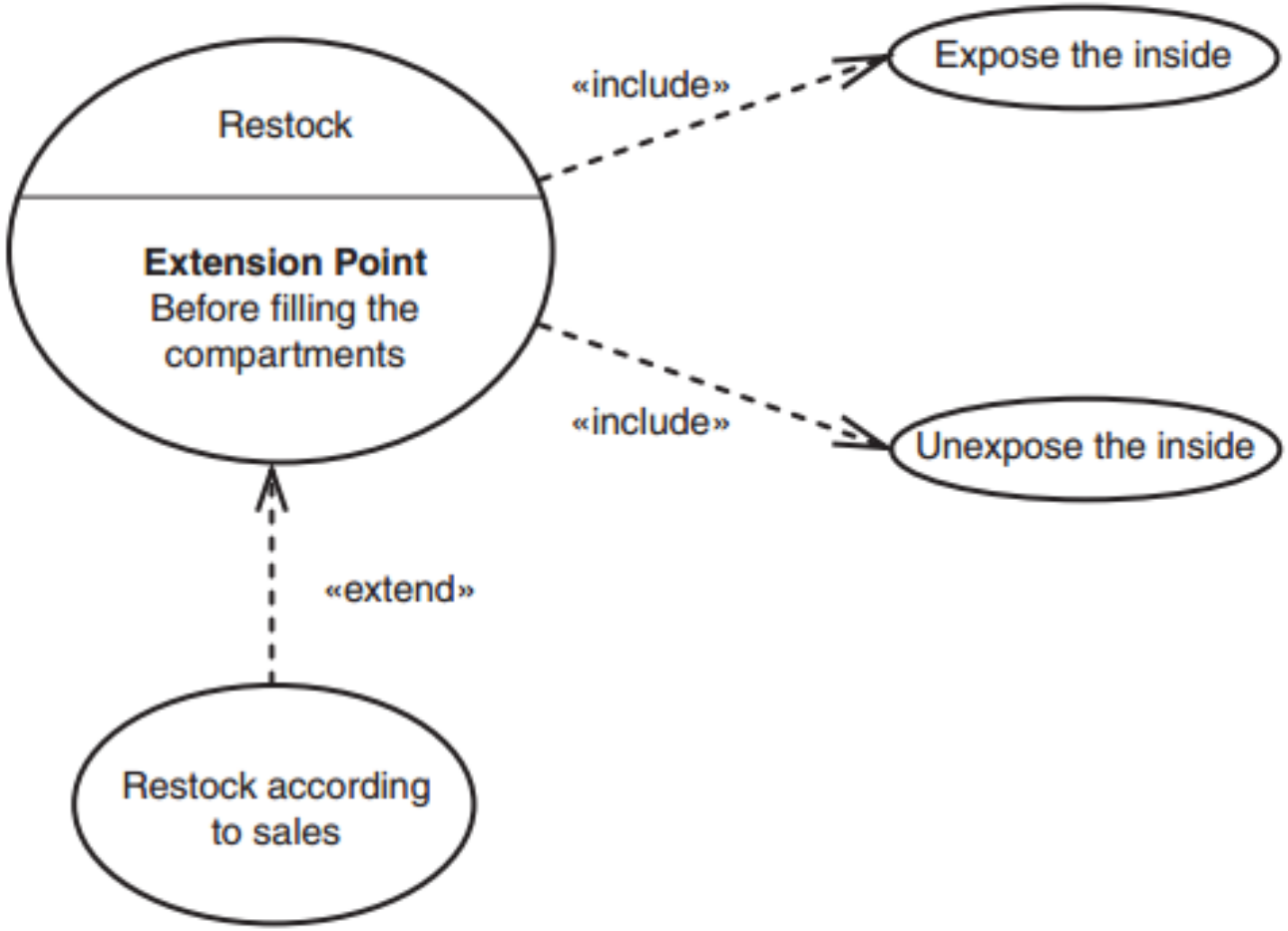


Case Study 1

Giả sử ta có thêm chức năng là thêm soda dựa trên doanh thu của một sản phẩm nào đó

Case Study 1

Xác định các Mối liên hệ?:



Bài tập

Nội dung:

Một công ty xây dựng hệ thống bán và đặt vé. Người sử dụng có thể trực tiếp đặt vé trước, hủy đặt vé, thay đổi hạng vé. Các thao tác trên được hỗ trợ bởi một thư kí phòng vé.

Người dùng có thể đăng kí một tài khoản, sau đó đặt vé online bằng tài khoản đó. Người dùng có thể thanh toán online bằng thẻ tín dụng hoặc tài khoản ngân hàng. Hệ thống cũng cho phép kiểm tra thông tin tàu, vé và giờ khởi hành

Bài tập

Yêu cầu:

➢Vẽ Use Case Diagram hoàn chỉnh cho bài tập trên

USE CASE DESCRIPTION

Use Case Description

Khái niệm:

- Là những **đoạn text** mô tả chức năng của use case dưới ngôn ngữ / thuật ngữ của user
- Không có UML format
- Mô tả "cái gì" (WHAT) chứ không mô tả "như thế nào" (HOW)
- Thường bao gồm: mục đích, khởi tạo, luồng sự kiện của use case
- Đóng vai trò cầu nối giữa stakeholder - dev

Use Case Description

Key components	Explanation
Name	*Clear, unique name of the use case (verb, goal-driven)*
Actors	*Someone or something that interacts with the use case*
Description	*Brief overview of the use case, describing the main idea*
Goal	*What the actors achieve with this use case*
Pre-condition	*State(s) the system can be in before the use case starts*
Trigger	*Event that causes the use case to be initiated*
Post-condition	*State(s) the system can be in after the use case finishes*
Normal flow	*Typical (primary) processing path*
Alternative flow	*Alternative (secondary) processing path*
Exception flow	*When things go wrong at the system level*
Others	Business rules, Assumption, Notes, etc.

Use Case Description

Flow – Luồng sự kiện:

- Khi xây dựng hệ thống ta cần mô tả UC chi tiết hơn, các chi tiết này (hành vi của UC) được viết trong tài liệu văn bản luồng sự kiện (flow of events).
- Tài liệu luồng sự kiện mô tả chi tiết người sử dụng sẽ làm gì và hệ thống sẽ làm gì. Tuy là chi tiết nhưng luồng sự kiện vẫn **độc lập đối với ngôn ngữ**

Use Case Description

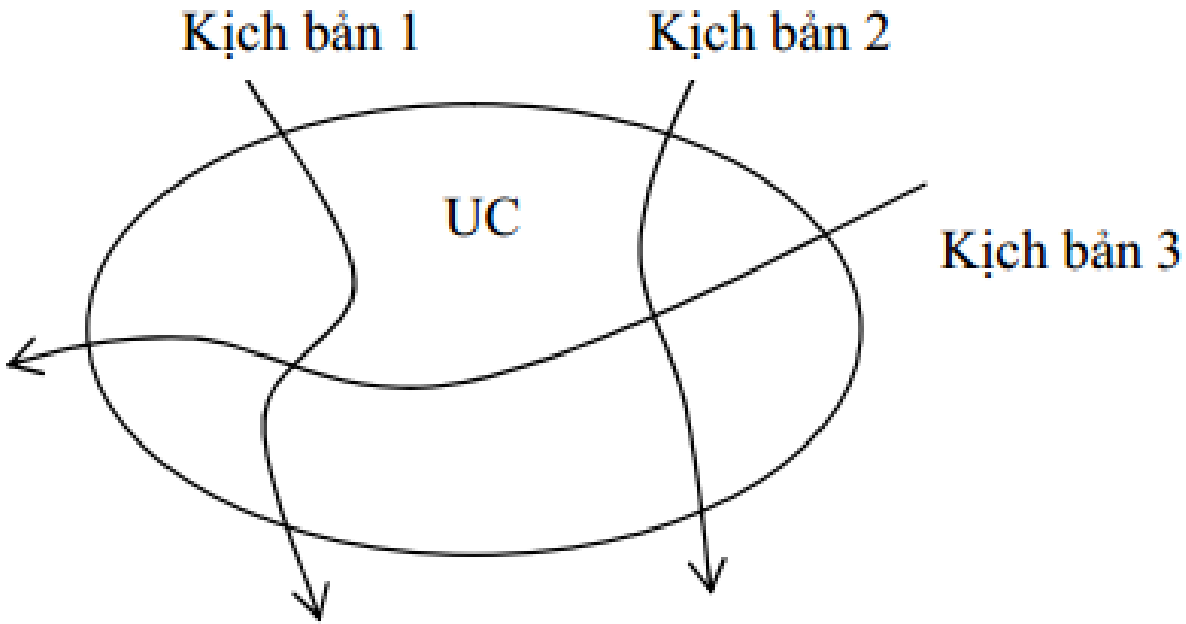
Kịch bản – Scenarios:

- Kịch bản đi xuyên suốt UC theo nhánh chính hay các nhánh phụ hoặc nhánh đặc biệt của đường đi
- Thí dụ, trong một hệ thống có UC tuyển nhân viên. Chức năng tác nghiệp này có nhiều biến thể: tuyển nhân viên mới, thuyên chuyển nhân viên từ cơ quan khác, tuyển nhân viên nước ngoài

Use Case Description

Kịch bản – Scenarios:

➢Mỗi kịch bản (Scenario) đó có thể theo một luồng sự kiện khác nhau (dù cùng thuộc một UC)



Use Case Description

Yêu cầu:

➢Làm UC Description cho UC “Đặt vé máy bay cho chuyến đi du lịch” ?

Use Case Description

Make a seat reservation use case	
Name	*Make reservation*
Actors	*Passenger*
Description	*Allows a passenger to book a plane seat for a journey from the Website*
Goal	*Reserve a seat*
Pre-condition	*Main Webpage is displayed successfully*
Trigger	*User clicks on "Reserve seat" button on the main Webpage*
Post-condition	• *A seat is booked* • *Number of available seats is reduced*

Use Case Description

Make a seat reservation use case	
Normal flow	*[User log in and reserve a seat successfully]* 1. User logs in 2. User specifies a flight and travel details 3. User specifies passenger details 4. User specifies payment details 5. User confirms transaction
Alternative flow	*[When no seat is available on the selected date]* • Show option to select another day • Repeat steps in normal flow
Exception flow	*[When a payment is failed]* • Notify error with the payment • Give an option to re-enter payment details or other payment method

ACTIVITY DIAGRAM

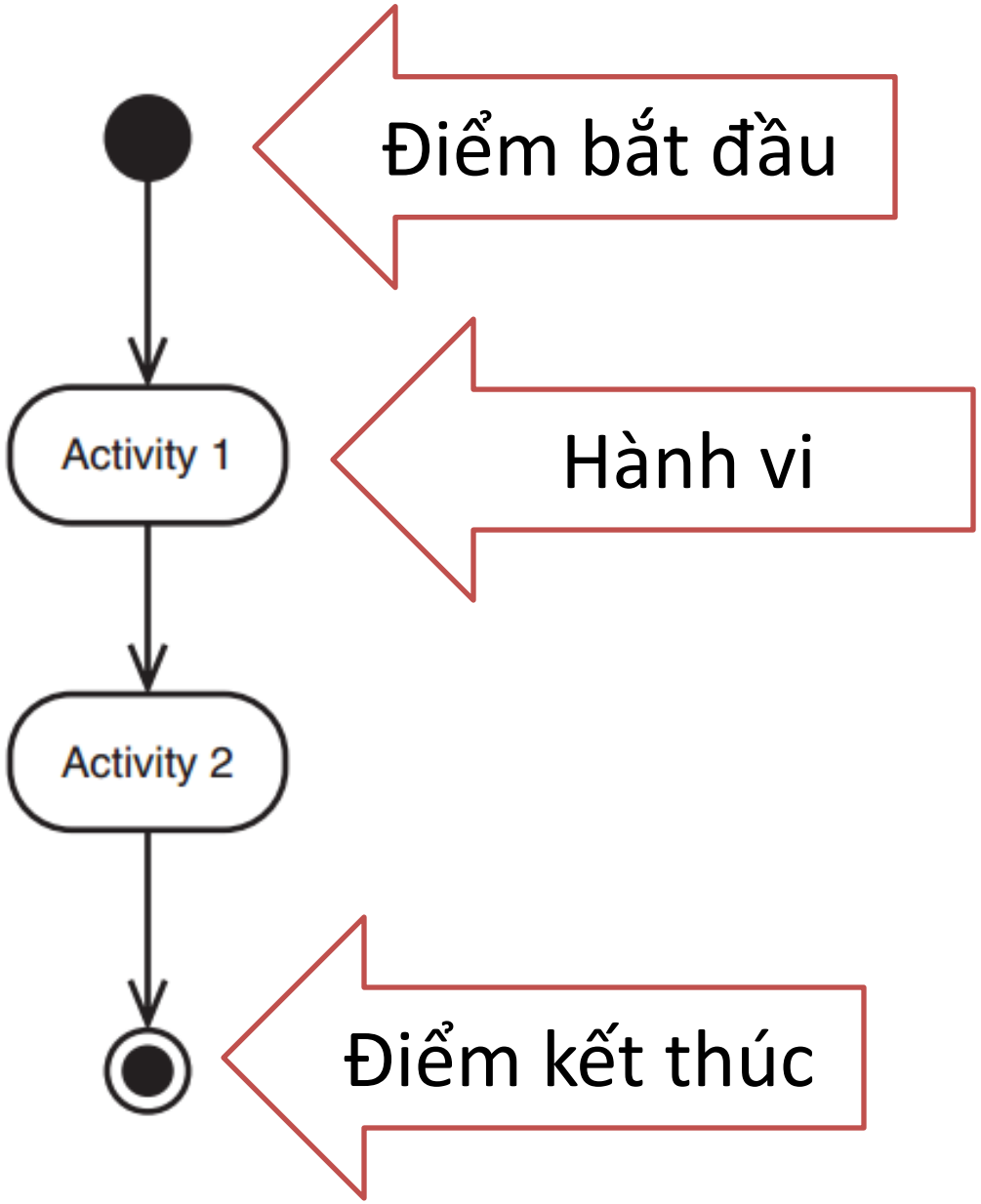
Activity Diagram

Khái niệm:

- Là một dạng biểu đồ hỗ trợ cho mô hình hóa nghiệp vụ
- Tương tự như flow chart (khi mô tả các bước của giải thuật)
- Bao gồm các khối hình chữ nhật cạnh tròn dùng để biểu diễn các hành vi
- Được dùng để hỗ trợ thêm cho mô tả Use Case

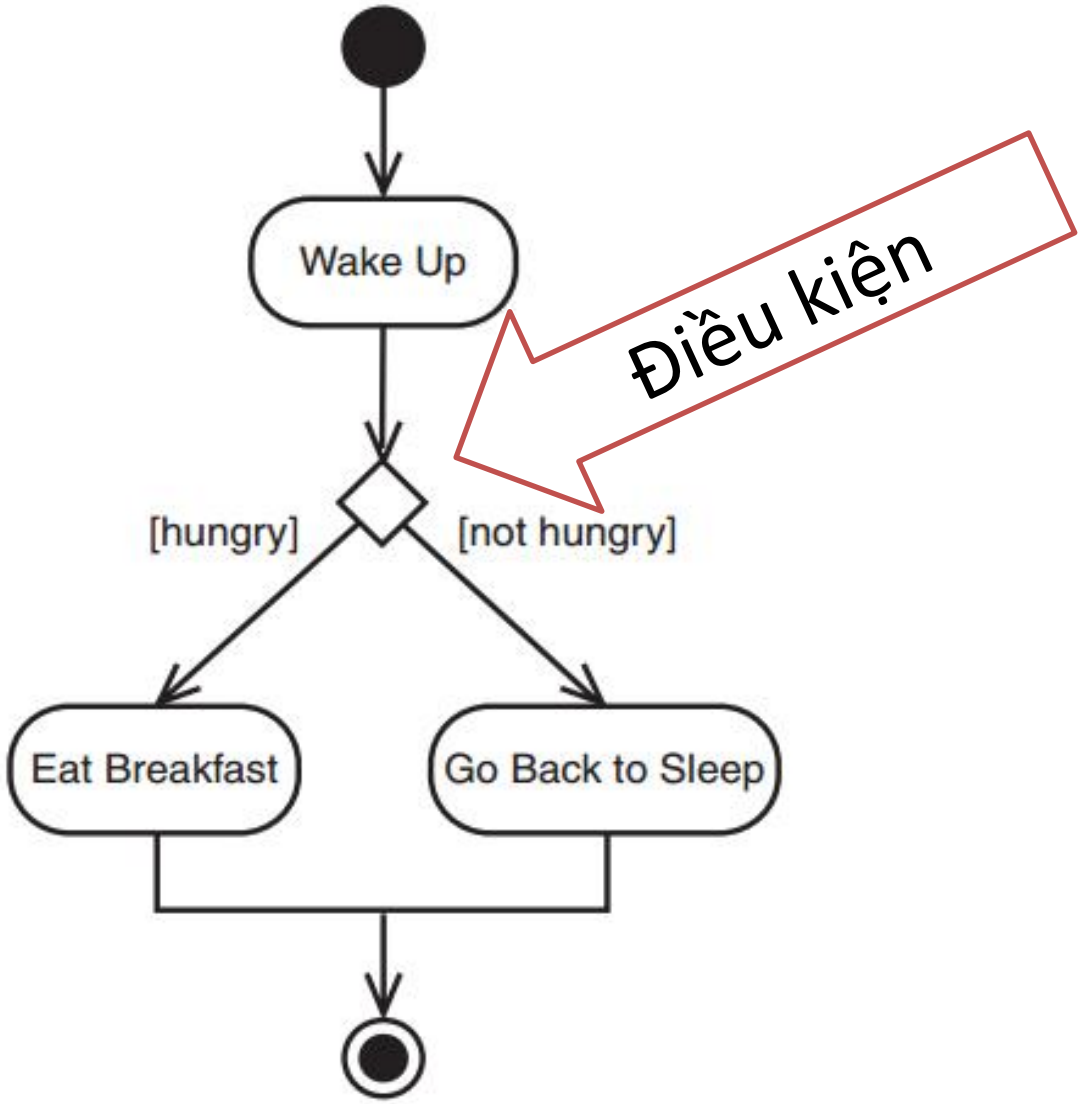
Activity Diagram

Minh họa:



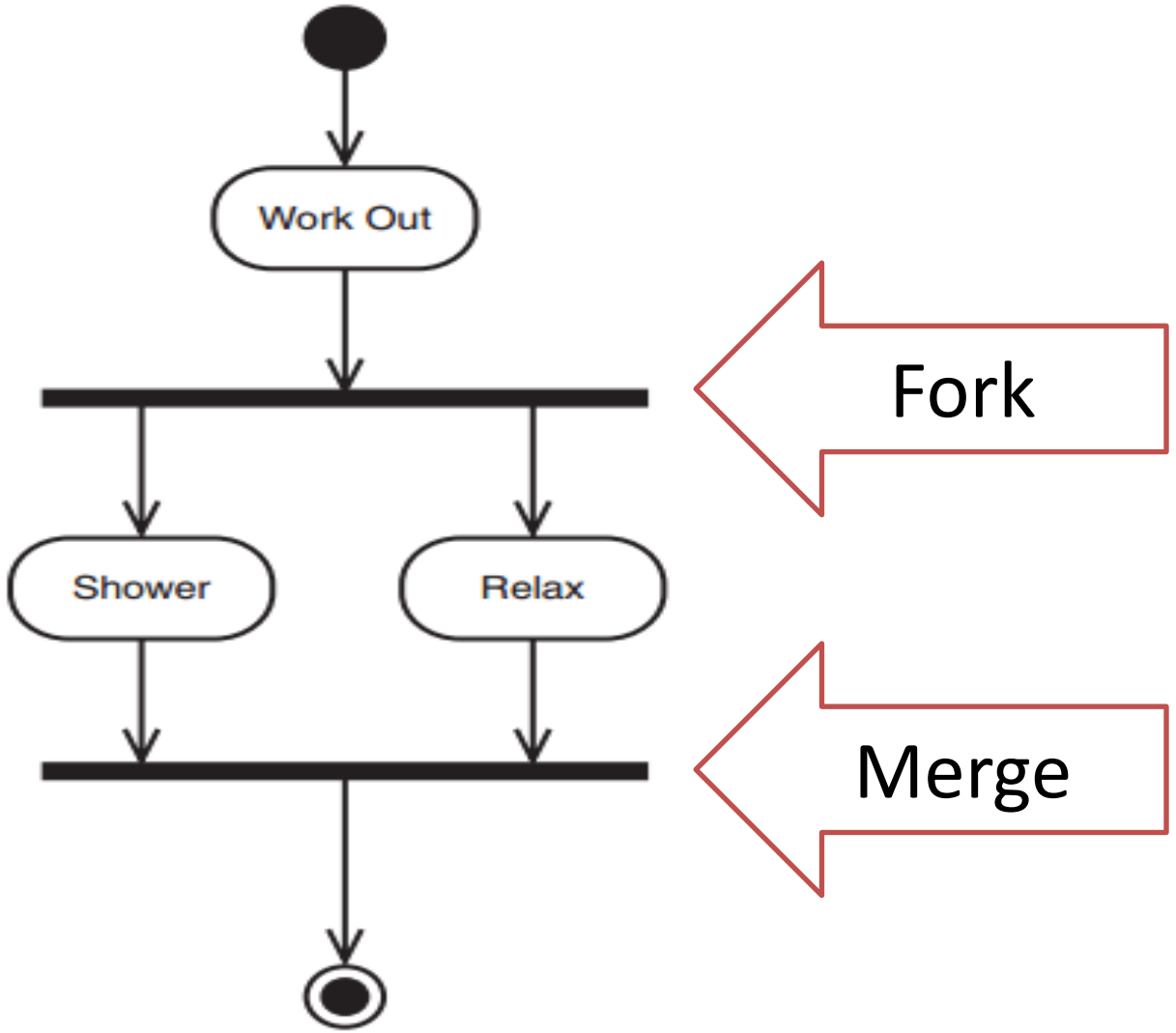
Activity Diagram

Rẽ nhánh:



Activity Diagram

Đồng thời:



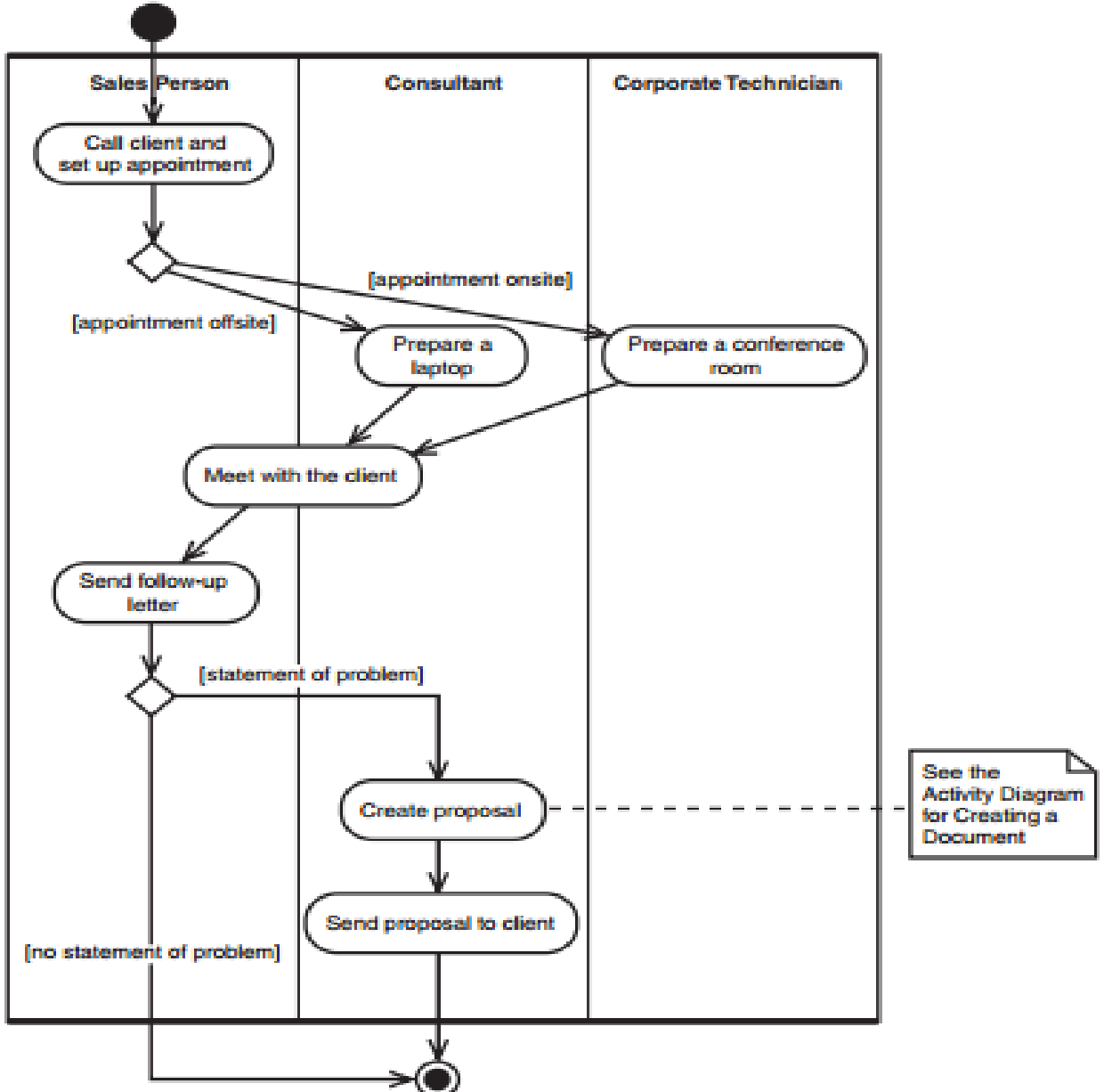
Activity Diagram

Ví dụ:

1. Open the word processing package.
2. Create a file.
3. Save the file under a unique name within its directory.
4. Type the document.
5. If graphics are necessary, open the graphics package, create the graphics, and paste the graphics into the document.
6. If a spreadsheet is necessary, open the spreadsheet package, create the spreadsheet, and paste the spreadsheet into the document.
7. Save the file.
8. Print a hard copy of the document.
9. Exit the office suite.

Activity Diagram – Swim Lanes

Ví dụ:



Activity Diagram

Bài tập 1:

You have to decide whether to eat at home or in a restaurant. If you eat in a restaurant, you go there, order your meal, and eat it. If you stay at home, you have to find a recipe, find the ingredients, and cook it. You need to lay the table at the same time. Finally you eat your dinner.

Draw an activity diagram for this description

Activity Diagram

Bài tập 2:

When I order a take-away meal, I phone the Chinese restaurant, give him my phone number, and his computer will automatically display my address if I've ordered before. But if I'm a new customer calling for the first time, he must get my address before he takes my order.

Draw an activity diagram for this description

Activity Diagram

Bài tập 3:

Activities

Actor		System	
Main Flow: Add new user successfully			
1	Từ màn hình bất kỳ, actor lựa chọn left menu **Quản lý người dùng/Thêm mới**		
		2	Load trang Thêm mới thông tin người dùng (SC05)
3	Nhập vào các thông tin được yêu cầu.		
		4	Validation các thông tin nhập vào.
5	Sửa lại những thông tin chưa đúng (nếu hệ thống check validation chưa đúng).		
		6	Lưu thông tin người dùng mới vào CSDL, thông báo thành công và chuyển sang trang Quản lý thông tin người dùng (SC02)


Q
A