


**BỘ MÔN CÔNG NGHỆ PHẦN MỀM
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài 10. Biểu đồ use case và Biểu đồ hoạt động

Nguyễn Thị Thu Trang
trangntt-fit@mail.hut.edu.vn

Nội dung

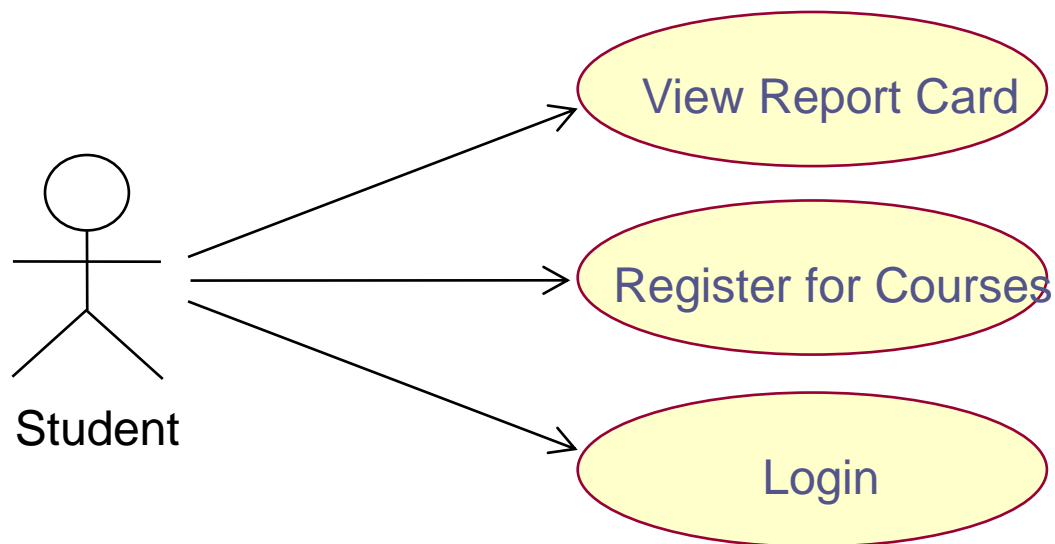
- 
1. Tổng quan về biểu đồ use case
 2. Các thành phần chính trong biểu đồ use case
 3. Biểu đồ hoạt động

1.1. Tổng quan về biểu đồ use case

- Mỗi hệ thống tương tác với con người hoặc các hệ thống khác để thực hiện nhiệm vụ
- Các hành vi của hệ thống có thể được mô tả trong các use case.
 - What, not How
 - Các use case mô tả các tương tác giữa hệ thống và môi trường của nó
 - Biểu đồ use case

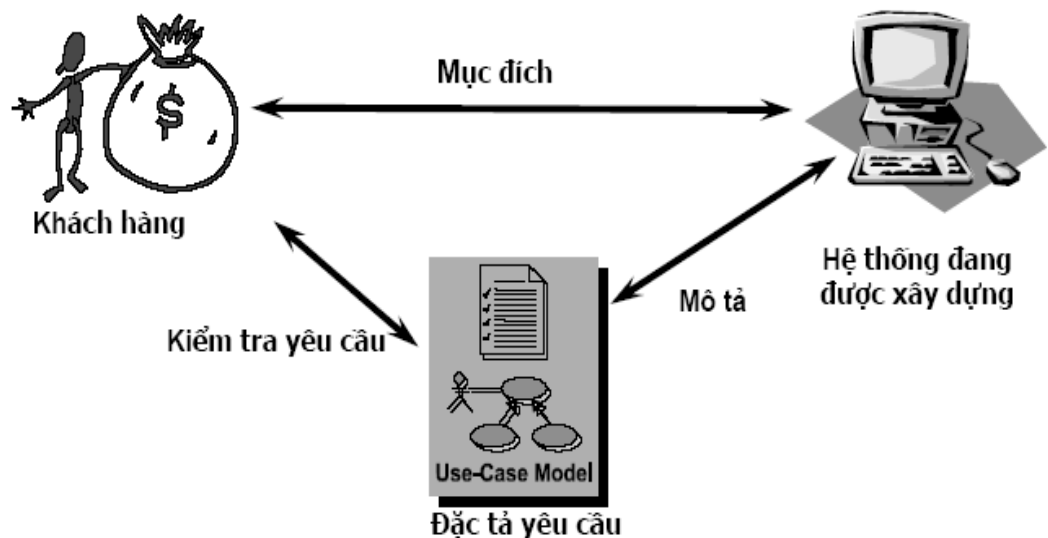
1.1. Tổng quan về biểu đồ use case (2)

- Biểu đồ mô tả các yêu cầu chức năng của hệ thống dưới dạng các use case.
- Bao gồm các chức năng mong đợi của hệ thống (use case) và môi trường (actor) của nó.



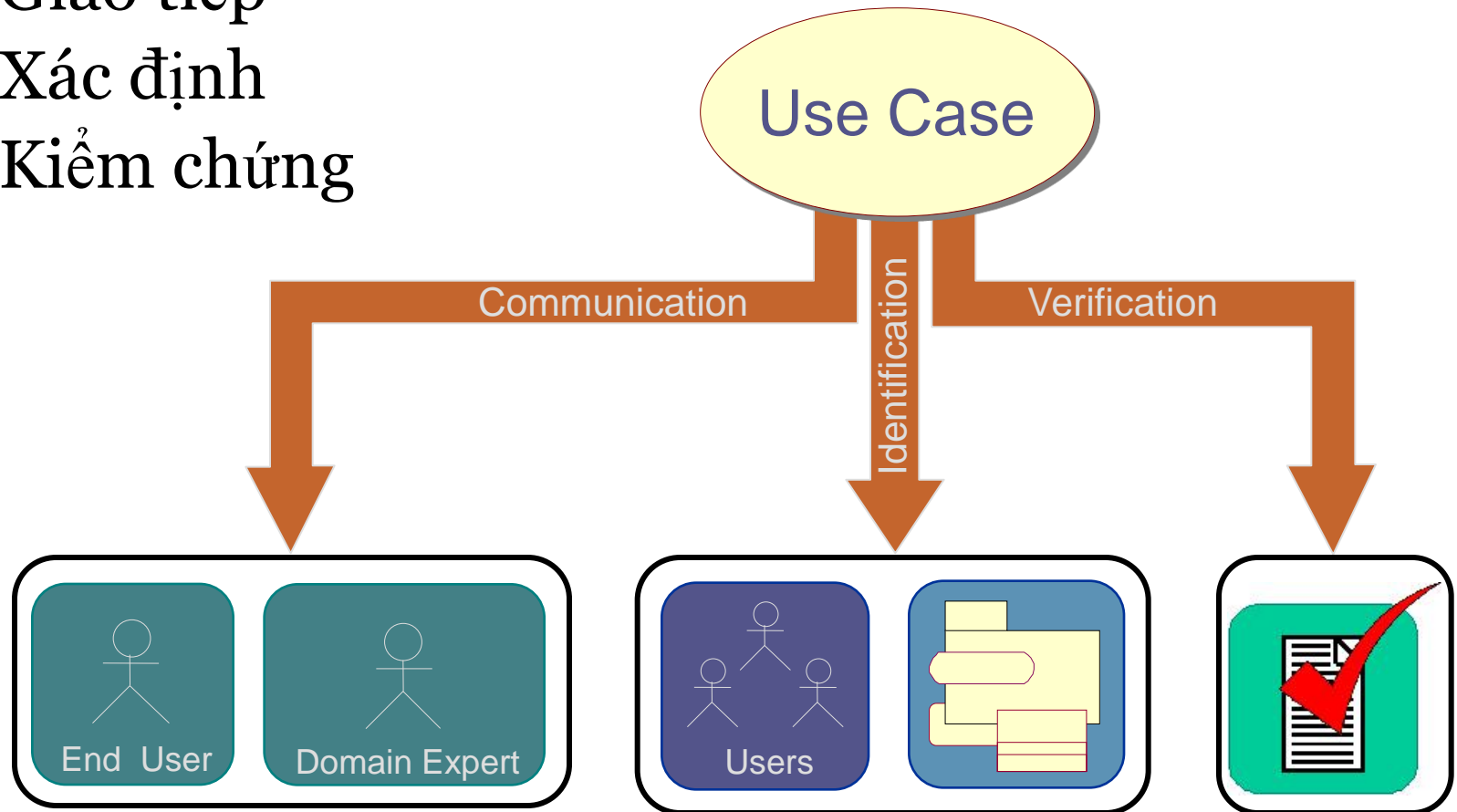
1.2. Lợi ích của biểu đồ use case

- Giống như một bản hợp đồng giữa người phát triển phần mềm và khách hàng.
- Là công cụ mạnh mẽ cho việc lập kế hoạch → Được dùng trong tất cả các giai đoạn trong quy trình phát triển hệ thống
 - Khách hàng của bạn phải phê chuẩn biểu đồ use-case
 - Sử dụng biểu đồ use case để thảo luận với khách hàng.
 - Các thành viên tham gia vào dự án, sử dụng mô hình này để hiểu rõ hơn về hệ thống



1.2. Lợi ích của biểu đồ use case (2)

- Giao tiếp
- Xác định
- Kiểm chứng



Nội dung

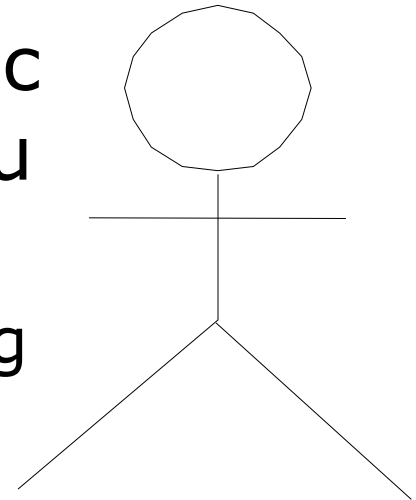
1. Tổng quan về biểu đồ use case

⇒ 2. Các thành phần chính trong biểu đồ use case

3. Biểu đồ hoạt động

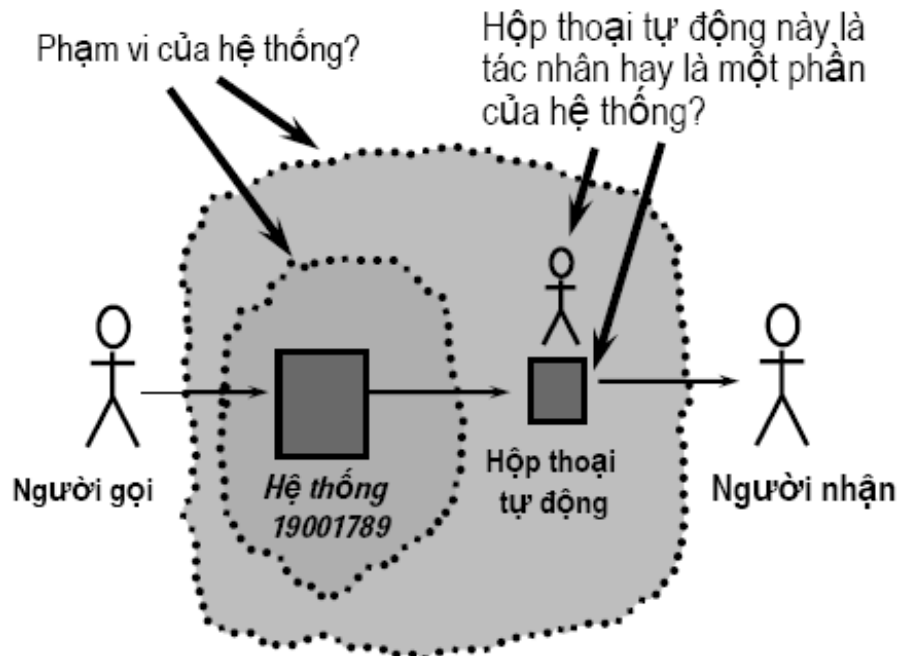
2.1. Tác nhân (actor)

- ◆ Tác nhân là bất kỳ thứ gì tương tác với hệ thống, có sự trao đổi dữ liệu với hệ thống
 - Là một lớp/loại người dùng chứ không phải một người cụ thể
 - Một người dùng cụ thể có thể đóng vai trò là các tác nhân khác nhau, có nghĩa là người đó có nhiều vai trò khác nhau trong hệ thống
 - Không phải là một phần của hệ thống
 - ◆ Actors are EXTERNAL.



Actor

Ví dụ về tác nhân



- Tác nhân **KHÔNG** phải là một phần của hệ thống!!!
- Giúp giới hạn hệ thống

Tác nhân có thể là:

- Người dùng,
- Thiết bị phần cứng
- Hệ thống phần mềm khác

- Tác nhân trao đổi thông tin với hệ thống:
 - Gửi thông tin tới hệ thống
 - Nhận thông tin từ hệ thống

Tìm kiếm tác nhân của hệ thống

- Đặt các câu hỏi sau để tìm ra tác nhân:
 - Nhóm người nào yêu cầu hệ thống làm việc giúp họ?
 - Nhóm người nào kích hoạt chức năng của hệ thống?
 - Nhóm người nào sẽ duy trì và quản trị hệ thống hoạt động?
 - Hệ thống có tương tác với các thiết bị hay phần mềm ngoại vi nào khác hay không?
- Thông tin về tác nhân:
 - Tên tác nhân phải mô tả vai trò của tác nhân đó một cách rõ ràng
 - Tên nên là danh từ
 - Cần mô tả khái quát khả năng của tác nhân đó

2.2. Use case

- ◆ Use case mô tả chức năng của hệ thống, là một chuỗi các hành động của hệ thống thực hiện nhằm thu được một kết quả dễ thấy tới một tác nhân nào đó.
 - Một use case mô hình hóa một hội thoại giữa một hoặc nhiều tác nhân với hệ thống
 - Một use case mô tả hành động của hệ thống thực hiện nhằm mang đến một giá trị nào đó cho tác nhân.



Use Case

Tìm use case của hệ thống

- Xem các yêu cầu chức năng để tìm ra các UC
- Đối với mỗi tác nhân tìm được, đặt các câu hỏi:
 - Các tác nhân yêu cầu những gì từ hệ thống
 - Các công việc chính mà tác nhân đó muốn HT thực thi?
 - Tác nhân đó có tạo ra hay thay đổi dữ liệu gì của HT?
 - Tác nhân đó có phải thông báo gì cho HT?
 - Tác nhân đó có cần thông tin thông báo gì từ HT?
- Thông tin về use case:
 - Tên của UC nên chỉ rõ kết quả của quá trình tương tác với tác nhân
 - Tên nên là động từ
 - Mô tả ngắn gọn về mục đích của UC

Những điều nên tránh khi tạo UC

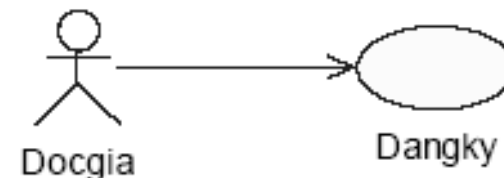
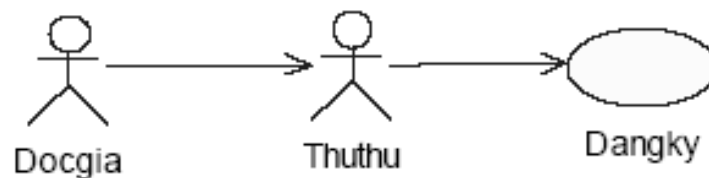
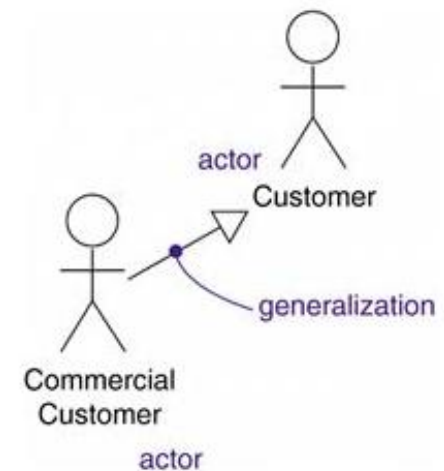
- Tạo ra các UC quá nhỏ
 - Hành động quá đơn giản mà chỉ cần mô tả bởi vài dòng
- Tạo ra quá nhiều Use case (hàng chục)
 - Nhóm các Use case liên quan thành một Use case tổng quát (mức 1)
 - Mô tả các Use Case tổng quát ở một sơ đồ khác (mức 2)
 - Ví dụ: “Quản lý sách” bao gồm “Nhập sách”, “Xuất sách”, “...”
- Sử dụng các Use-case quá cụ thể, hoặc làm việc với dữ liệu quá cụ thể. Ví dụ:
 - “Tìm sách theo tên” (nên là “Tìm sách”)
 - “Nhập Pin vào máy ATM” (nên là “Nhập PIN”)
 - “Thêm sách” (nên là “Quản lý sách” bao gồm “Thêm sách”)

2.3. Mỗi liên hệ (relationship)

- Mỗi liên hệ giữa các actor với nhau
 - Khái quát hóa (Generalization)
 - Giao tiếp
- Mỗi liên hệ giữa actor và use case
 - Giao tiếp
- Mỗi liên hệ giữa các use case với nhau
 - Generalization: Khái quát hóa
 - Include: Bao hàm
 - Extend: Mở rộng

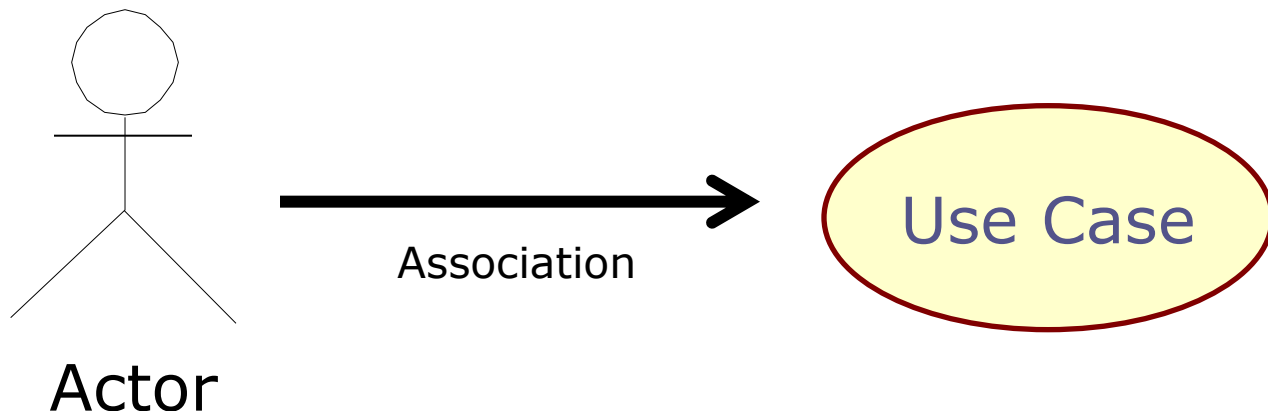
2.3.1. Mối liên hệ giữa các actor với nhau

- Khái quát hóa (Generalization)
 - Tác nhân con kế thừa tính chất và hành vi của tác nhân cha
- Giao tiếp
 - Xét sự khác nhau giữa hai biểu đồ sau



2.3.2. Mối liên hệ giữa actor với use case

- Thiết lập quan hệ giữa Tác nhân và Use Case
 - Chúng tương tác bằng cách gửi các tín hiệu cho nhau
- Một use case mô hình hóa một hội thoại giữa các tác nhân và hệ thống
- Một use case được bắt đầu bởi một tác nhân để gọi một chức năng nào đó trong hệ thống.



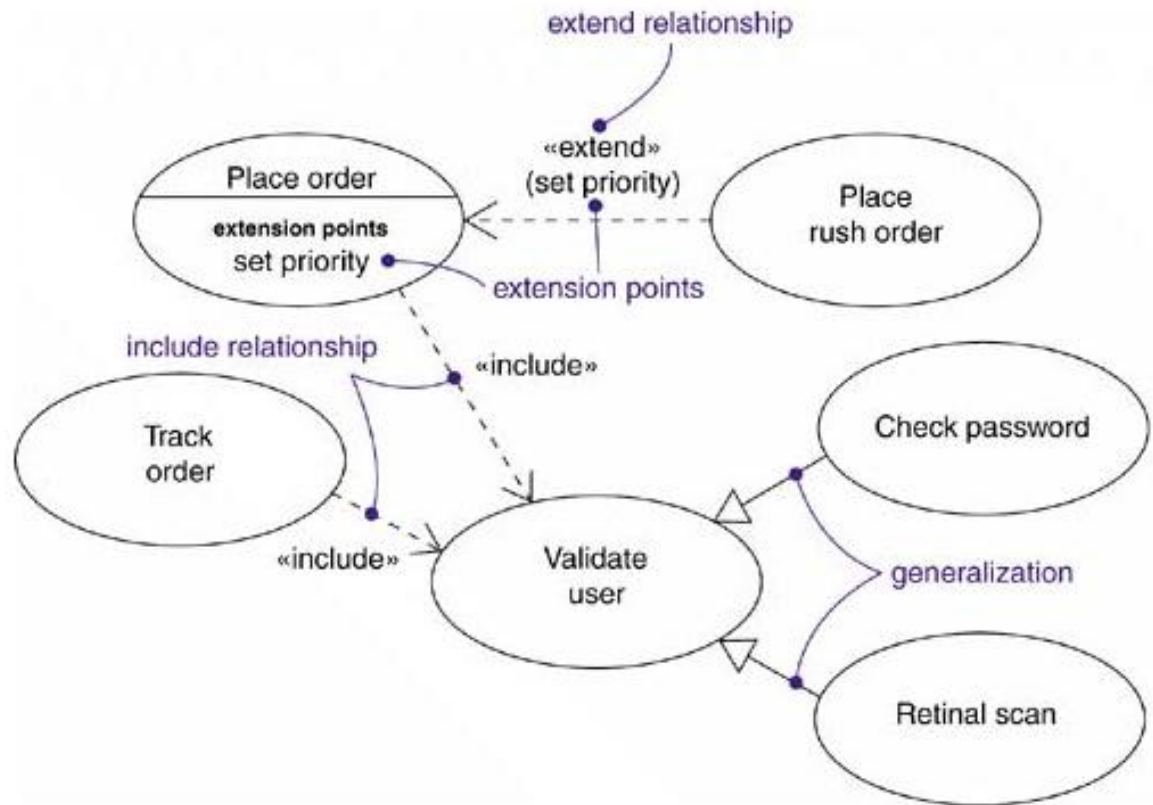
2.3.2. Mối liên hệ giữa actor với use case (2)

Chiều của quan hệ chính là chiều của tín hiệu gửi đi

- Từ tác nhân tới Use Case
 - Kích hoạt Use case
 - Hỏi thông tin nào đó trong hệ thống
 - Thay đổi thông tin nào đó trong hệ thống
 - Thông báo cho UC về một sự kiện đặc biệt nào đó xảy ra với hệ thống
- Từ Use Case tới tác nhân:
 - Nếu như có một điều gì đó xảy ra với HT và tác nhân đó cần được biết sự kiện đó
 - UC đôi khi cần hỏi thông tin nào đó từ một tác nhân trước khi UC đó đưa ra một quyết định

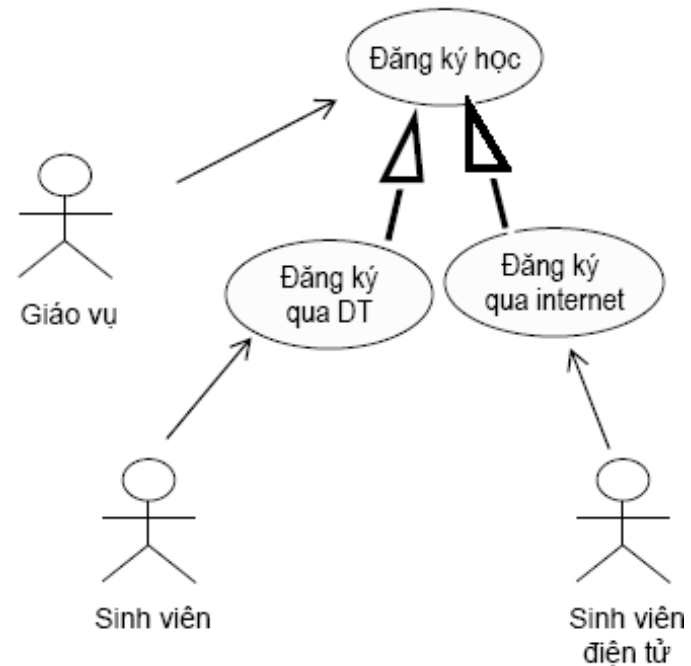
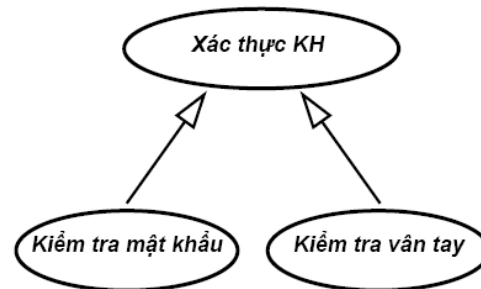
2.2.3. Mối liên hệ giữa các use case

- Generalization
- <<include>>
 - always use
- <<extend>>
 - sometime use



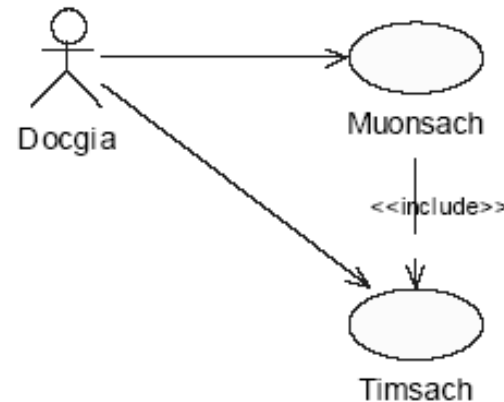
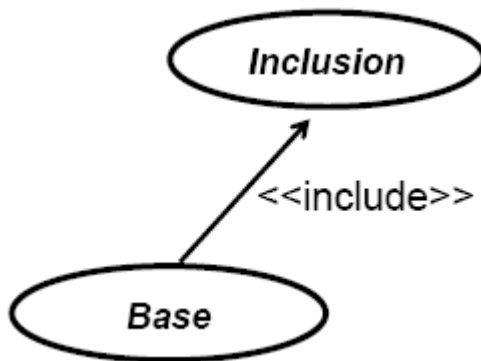
a. Quan hệ generalization

- Được sử dụng để chỉ ra một vài tính chất chung của một nhóm tác nhân hoặc UC
- Sử dụng khái niệm kế thừa
 - Mô tả hành vi chung (chia sẻ) trong UC cha
 - Mô tả hành vi riêng trong (các) UC con



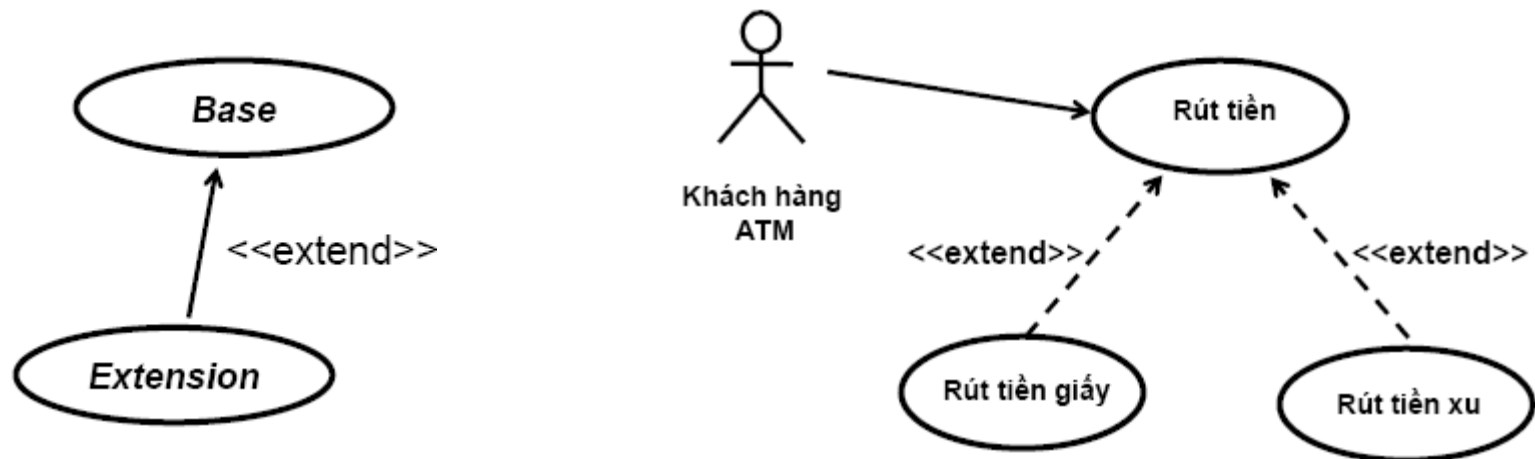
b. Quan hệ <<include>>

- Cho phép một UC sử dụng chức năng của UC khác
- Chức năng của UC Inclusion sẽ được gọi trong UC Base
- Sử dụng stereotype là <<include>>

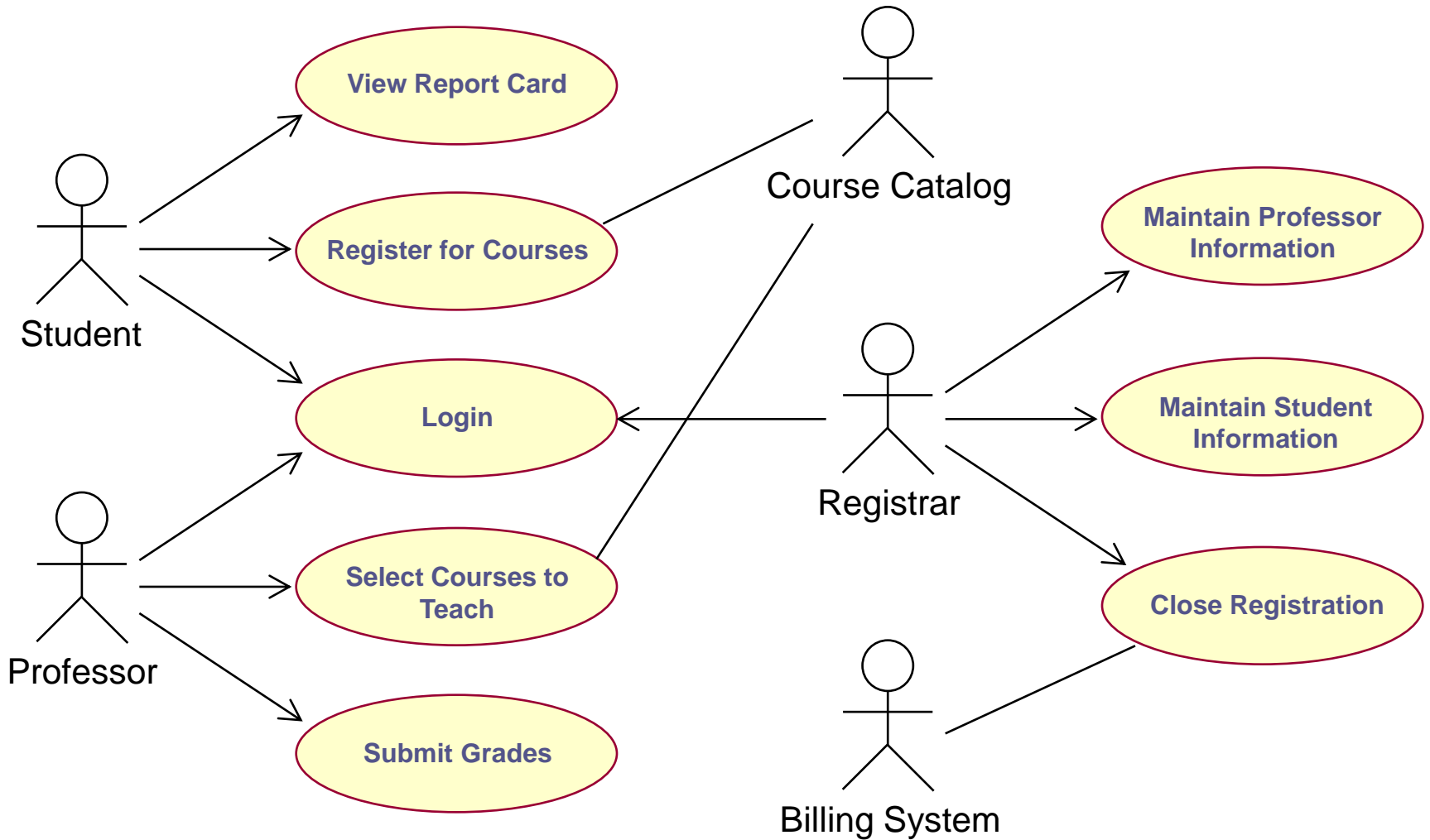


c. Quan hệ <<extend>>

- Cho phép mở rộng chức năng của một UC
- Chèn hành vi của UC Extension vào UC Base
- Chỉ chèn khi điều kiện extend đúng (mở rộng, phát sinh)
- Chèn vào lớp cơ sở tại điểm phát sinh (extension point)
- Sử dụng stereotype là <<extend>>



Đọc biểu đồ use case



Đọc biểu đồ use case

- Trả lời các câu hỏi sau:
 - Mô tả các chức năng của hệ thống
 - Sinh viên có thể tác động lên những use-case nào?
 - Giáo viên có thể tác động lên những use-case nào?
 - Nếu A vừa là sinh viên vừa là giáo viên, anh ta có thể thực hiện được những use-case nào?
 - Sơ đồ này không nói lên được những gì?
 - Những use-case nào cần thiết thực hiện đầu tiên?
- Biểu đồ use case có thể mô tả hết được không?

Nội dung

1. Tổng quan về biểu đồ use case
2. Các thành phần chính trong biểu đồ use case
- ⇒ 3. Biểu đồ hoạt động

3. Biểu đồ hoạt động

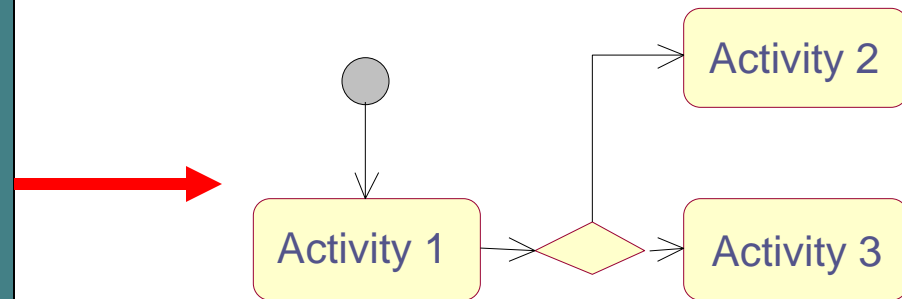
- Biểu đồ hoạt động (Activity Diagram – AD) được sử dụng để mô tả các hoạt động và các hành động được thực hiện trong một use case
 - Biểu đồ luồng (flow chart): Chỉ ra luồng điều khiển từ hoạt động/hành động này đến hoạt động/hành động khác.

Flow of Events

This use case starts when the Registrar requests that the system close registration.

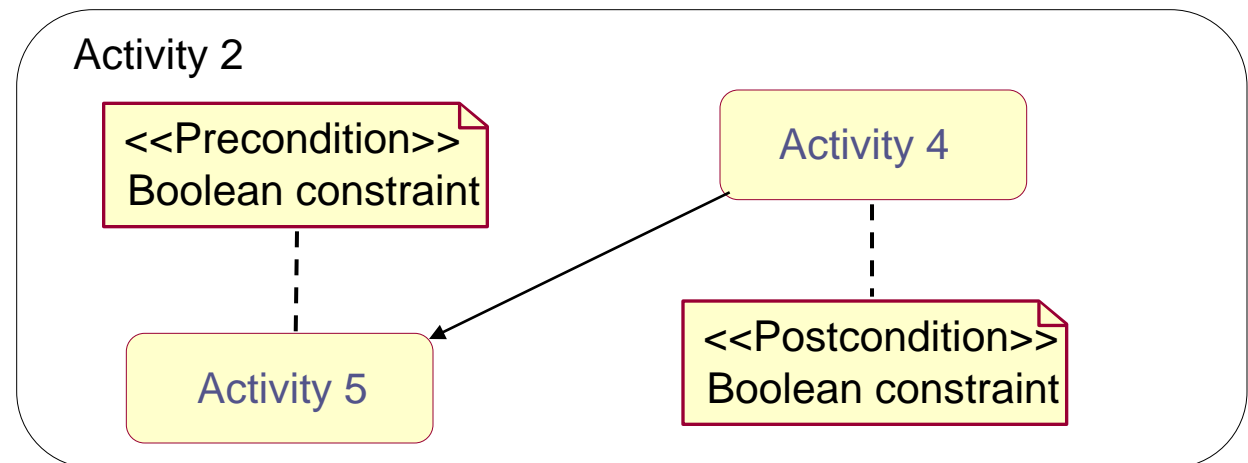
1. The system checks to see if registration is in progress. If it is, then a message is displayed to the Registrar and the use case terminates. The Close Registration processing cannot be performed if registration is in progress.

2. For each course offering, the system checks if a professor has signed up to teach the course offering and at least three students have registered. If so, the system commits the course offering for each schedule that contains it.

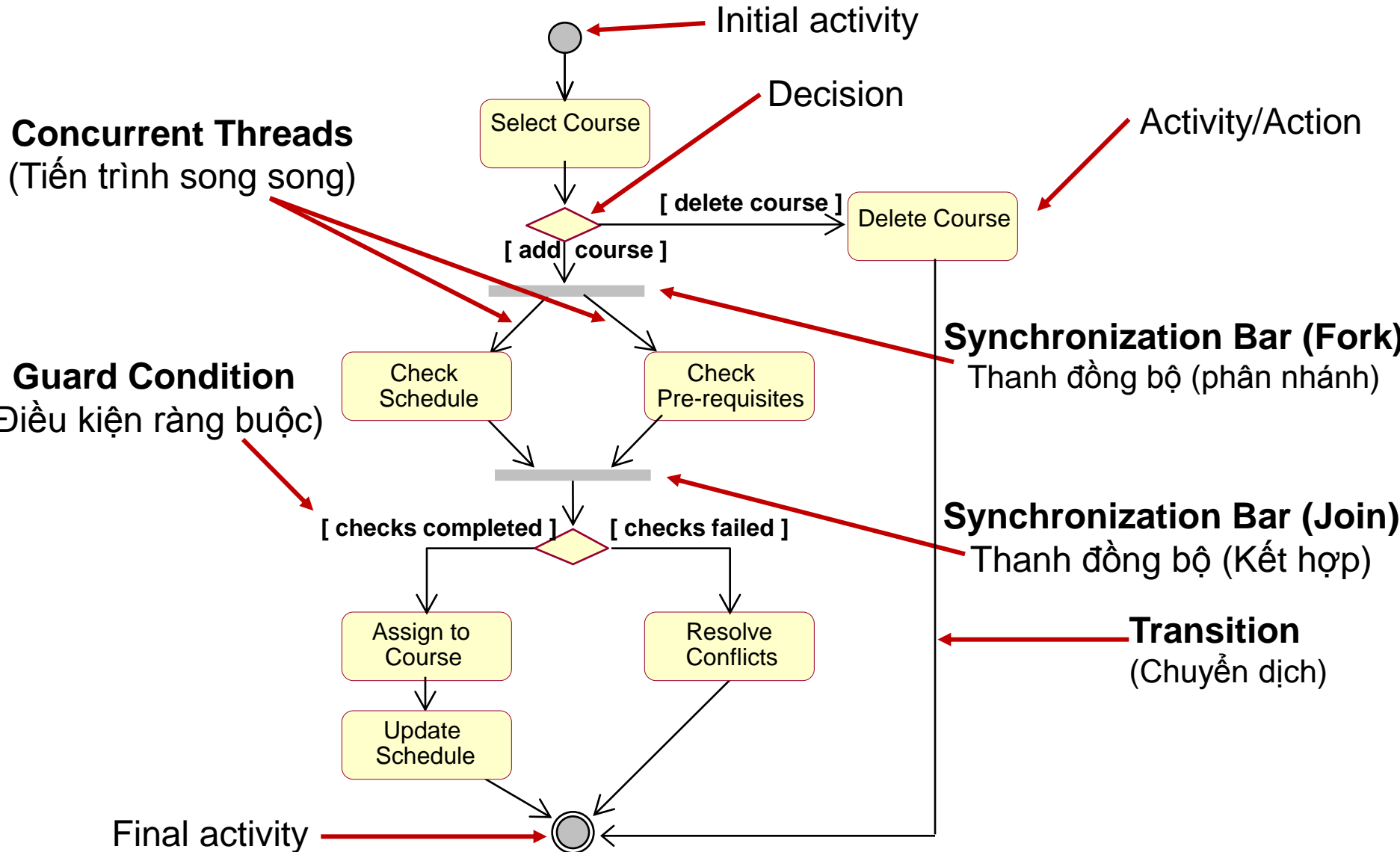


3. Biểu đồ hoạt động (2)

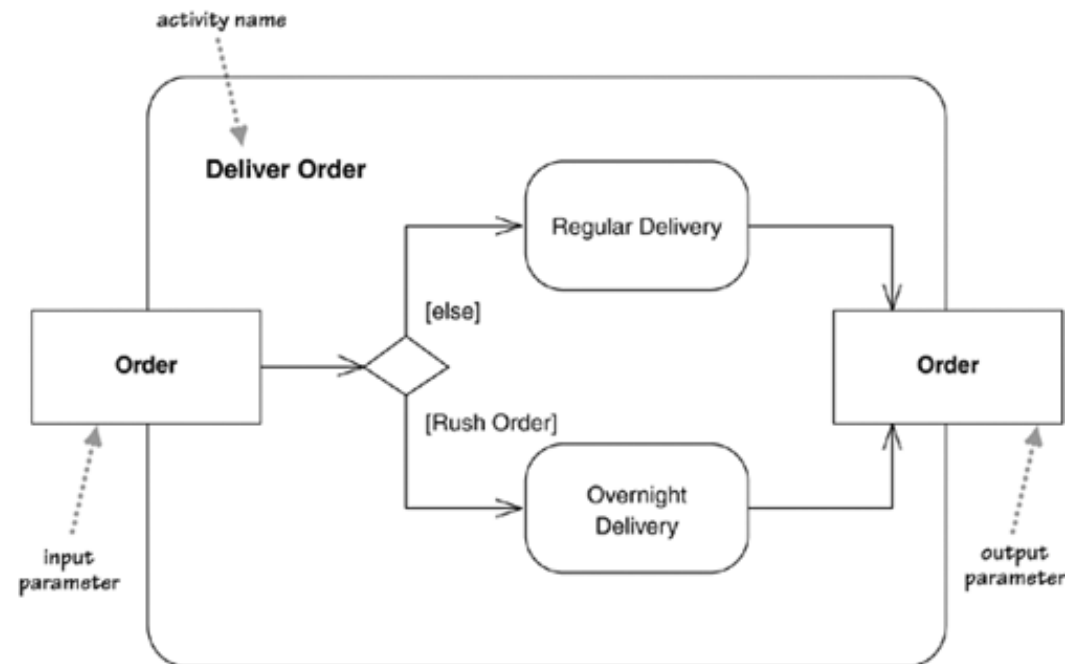
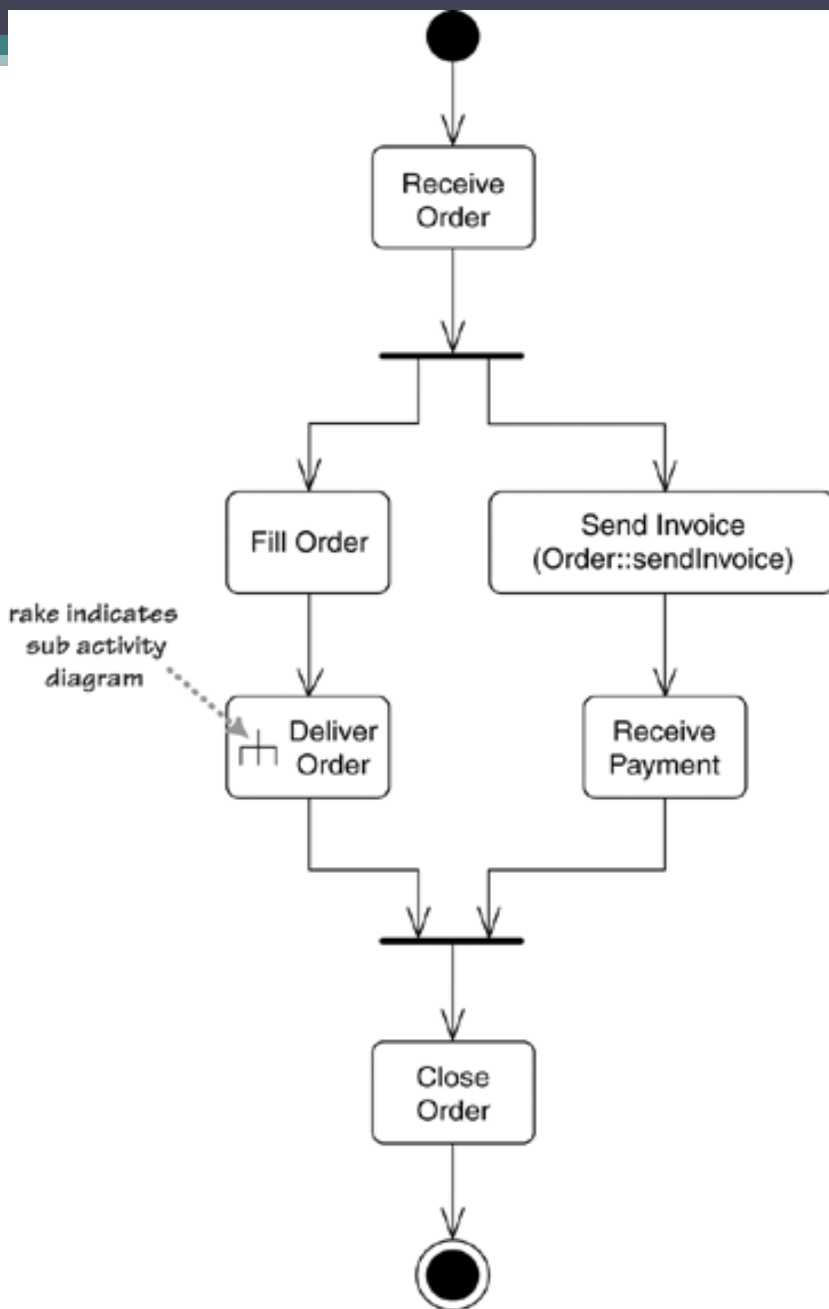
- Hoạt động
 - Đặc tả cho hành vi được diễn tả như một luồng thực thi thông qua sự sắp xếp thứ tự của các đơn vị nhỏ hơn.
 - Các đơn vị nhỏ hơn bao gồm các hoạt động lồng nhau và các hành động riêng lẻ cơ bản
- Có thể chứa các ràng buộc biểu thức logic khi hoạt động được gọi hoặc kết thúc



AD for Register for course use case



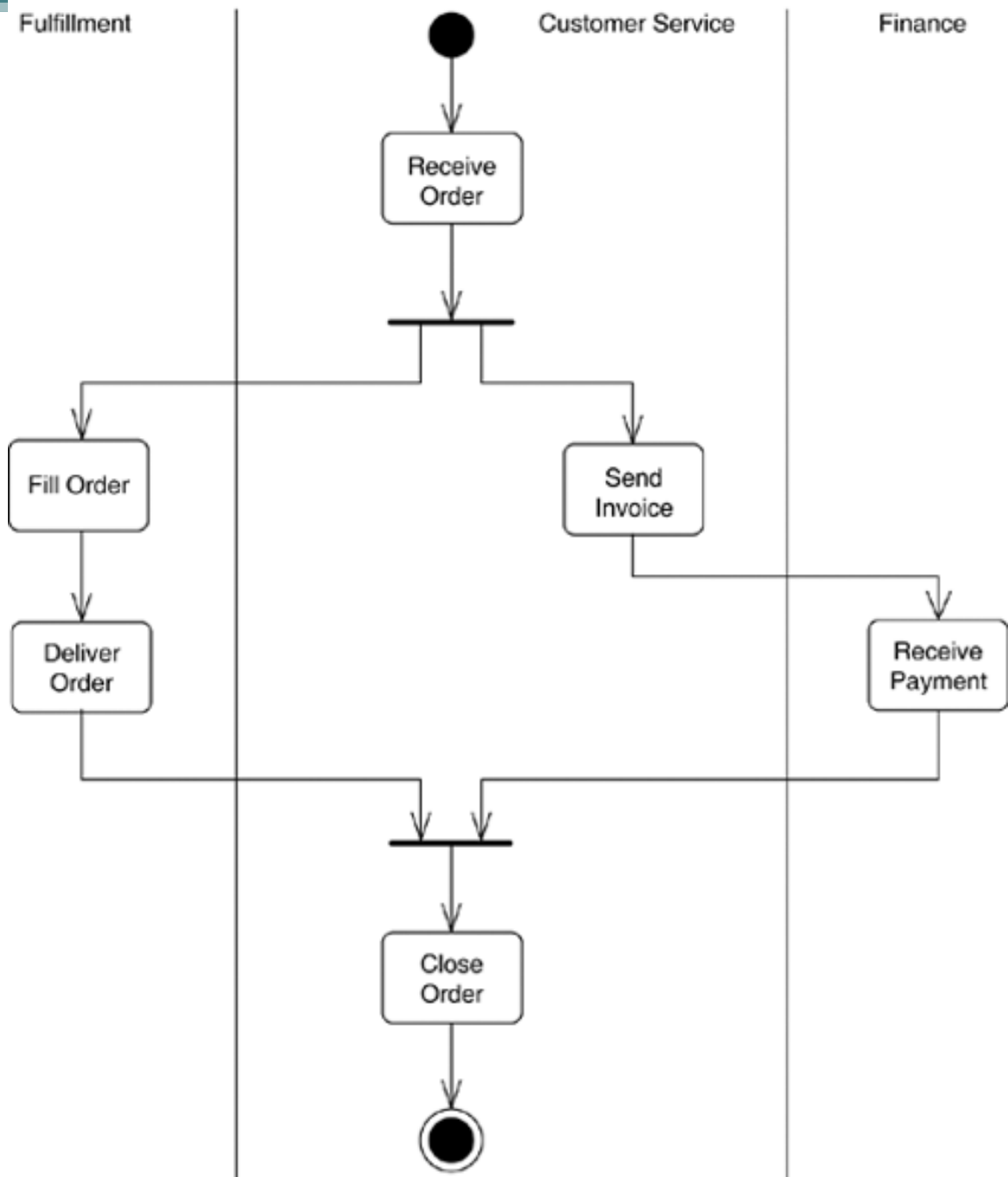
Gọi một AD khác



Phân chia (Partition)

- Biểu đồ hoạt động chỉ mô tả điều gì xảy ra chứ không mô tả ai làm gì
- Nếu muốn chỉ ra ai làm gì thì có thể phân chia thành các phần bao gồm các hoạt động do ai làm
- Có thể phân chia theo một chiều (hàng hoặc cột) hoặc hai chiều (cả hàng và cột)

Phân chia một
chiều hay còn gọi
là swim lane

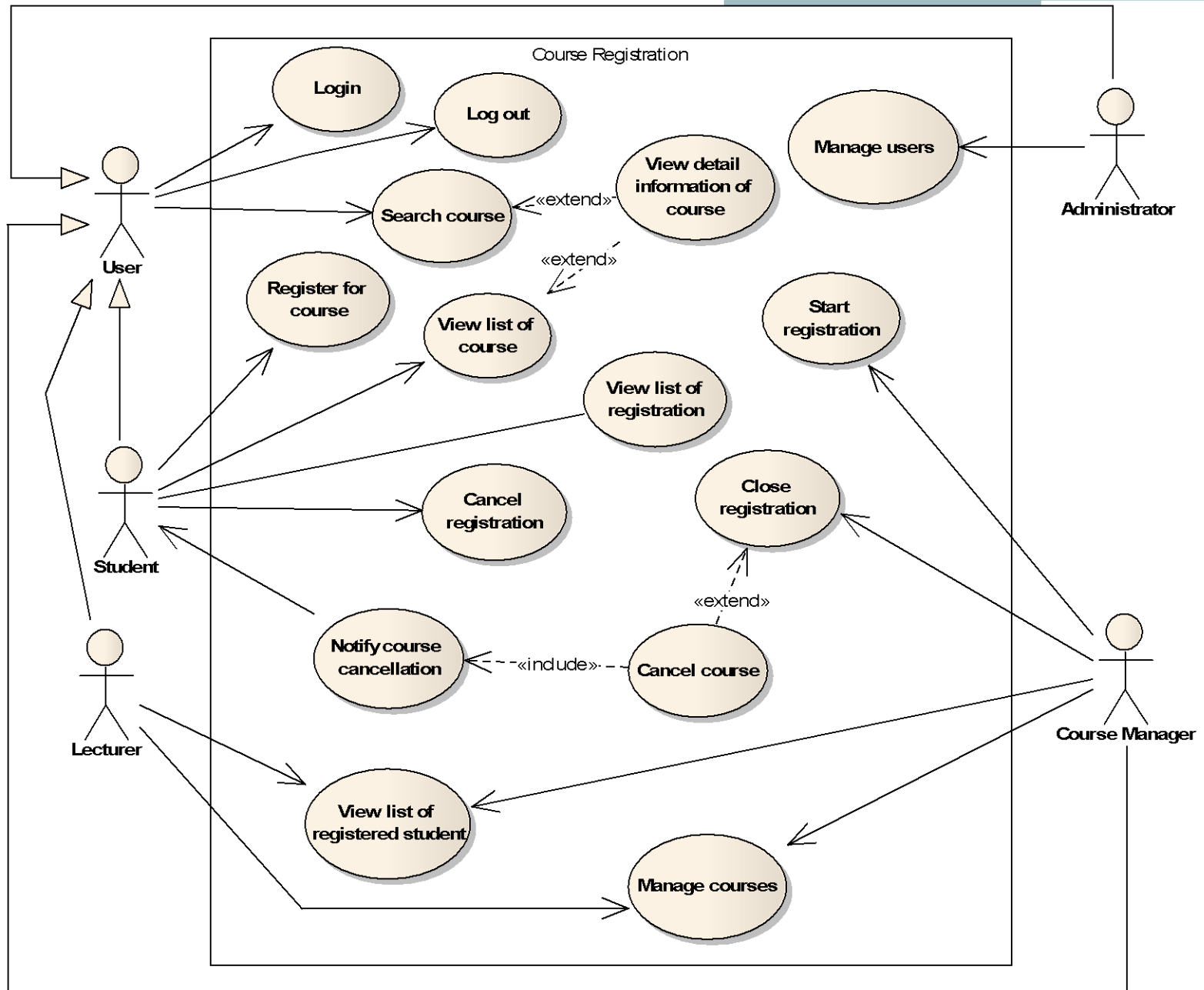


Case study: Hệ thống đăng ký học đơn giản

- Xác định các tác nhân của hệ thống:
 - Sinh viên (Student)
 - Giảng viên (Lecturer)
 - Người quản lý khóa học (Course Manager)
 - Người dùng thông thường (User)
 - Người quản trị hệ thống (Administrator)

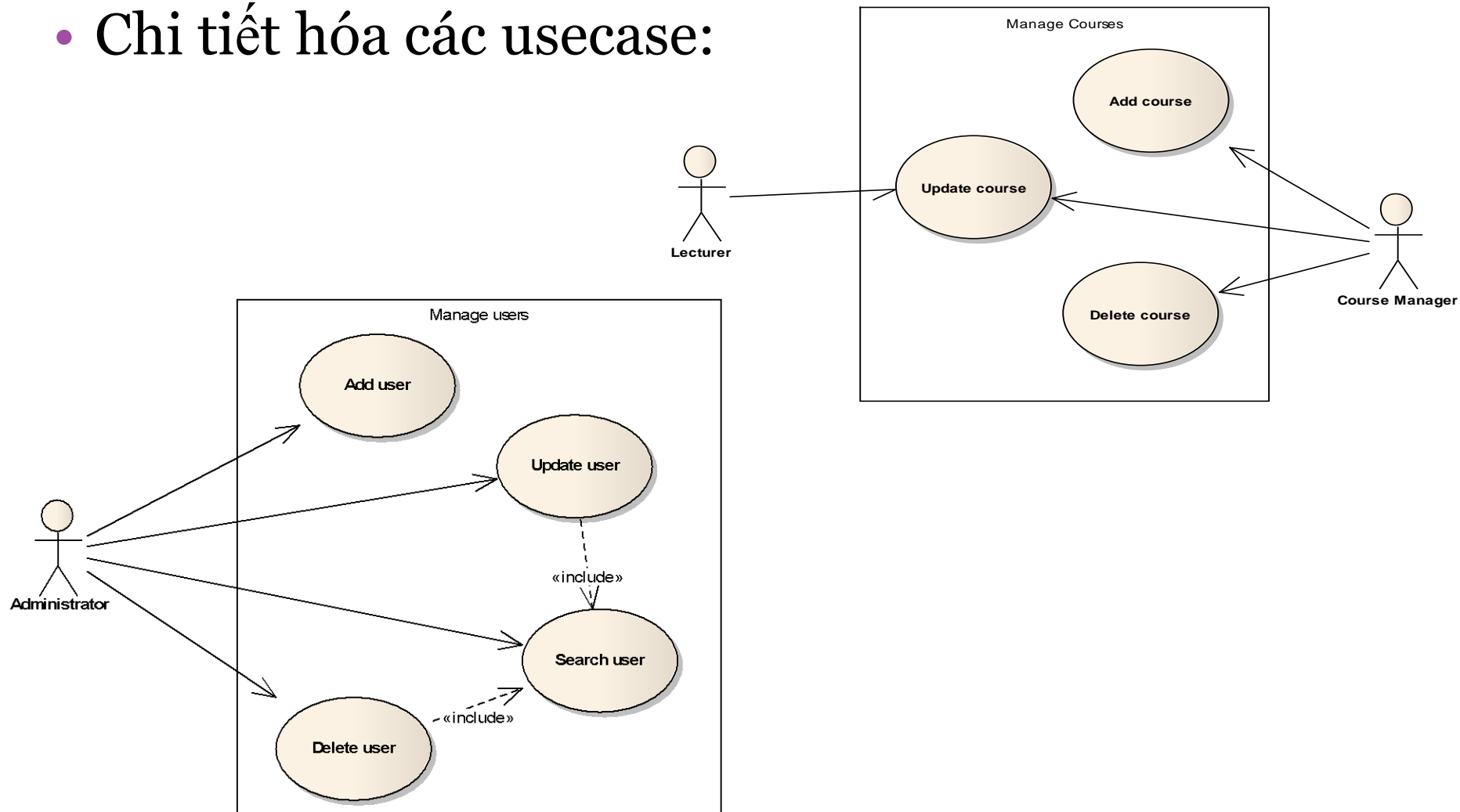
Case study: Hệ thống đăng ký học đơn giản

- Xác định các usecase cho từng tác nhân và các usecase chung, xây dựng biểu đồ usecase tổng quát:



Case study: Hệ thống đăng ký học đơn giản

- Chi tiết hóa các usecase:



Case study: Hệ thống đăng ký học đơn giản

- Đặc tả chi tiết các usecase:
 - Mã usecase
 - Tên usecase
 - Các tác nhân tham gia
 - Điều kiện trước
 - Luồng thực thi chính
 - Luồng mở rộng
- Ví dụ đặc tả chi tiết cho usecase “Thêm khóa học”

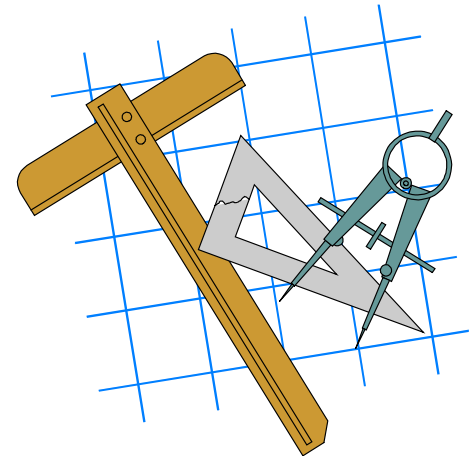
Mã UC (UC #)	UC001	Tên usecase	Thêm khóa học
Tác nhân	Người quản lý khóa học (Course Manager)		
Điều kiện trước	Tác nhân phải đăng nhập thành công và chức năng đăng ký chưa được mở		
Luồng thực thi chính			
No.	Thực hiện	Hành động	
1	Course manager	Chọn chức năng thêm khóa học	
2	System	Khởi tạo dữ liệu	
3	Course manager	Nhập thông tin khóa học	
4	Course manager	Yêu cầu thêm mới khóa học	
5	System	Kiểm tra các trường quan trọng trong thông tin khóa học (mã và tên KH)	
6	System	Kiểm tra tính hợp lệ của Ngày bắt đầu và ngày kết thúc	
7	System	Kiểm tra mã khóa học đã tồn tại	
8	System	Thông báo thêm khóa học thành công	
Luồng thực thi mở rộng			
No.	Thực hiện	Hành động	
6a.	System	notifies error: Chưa nhập đủ thông tin	
7a.	System	notifies error: Thông tin ngày bắt đầu và ngày kết thúc không hợp lệ	
8a.	System	notifies error: Mã khóa học đã tồn tại	

Case study: Hệ thống đăng ký học đơn giản

- Luyện tập: đặc tả usecase “Đăng ký khóa học” và “Hủy bỏ đăng ký”

- Cho:

- Các tác nhân: Người mua, Hệ thống E-mail, Hệ thống cho vay và Hệ thống báo cáo tín dụng
- Các use case: Tìm người môi giới, Quản lý hồ sơ cá nhân, Tìm kiếm nhà và Yêu cầu vay
- Các mối liên kết:
 - Từ người mua tới Tìm người môi giới
 - Từ người mua tới Quản lý hồ sơ cá nhân
 - Từ người mua tới Tìm kiếm nhà
 - Từ người mua tới Yêu cầu vay
 - Quản lý hồ sơ cá nhân tới Hệ thống e-mail
 - Tìm kiếm nhà tới Hệ thống e-mail
 - Yêu cầu vay tới Hệ thống e-mail, Hệ thống cho vay
 - Yêu cầu vay tới Hệ thống báo cáo tín dụng



- Hãy vẽ:

- Biểu đồ use-case

- Cho:

- Các trạng thái hành động:

- Chọn hồ sơ
 - Tìm hồ sơ người mua
 - Tạo hồ sơ mới
 - Đăng nhập

- Luồng hoạt động:

- Bắt đầu từ Chọn hồ sơ tới Tìm hồ sơ người mua rồi đi từ Tìm hồ sơ người mua đến Tạo hồ sơ mới nếu hồ sơ không tồn tại. Nếu hồ sơ tồn tại thì có thể Đăng nhập

- Hãy vẽ:

- Biểu đồ hoạt động

