

Chương 3: Phân tích chức năng

Nguyễn Hồng Hạnh, MSc
Bộ môn Công nghệ Phần mềm
Khoa CNTT – ĐH Xây Dựng Hà Nội
Email: hanhnh@huce.edu.vn

1

1

Nội dung trình bày

1. Mục đích phân tích chức năng (PTCN)
2. Các biểu đồ dùng để mô hình hóa chức năng
 1. Mô hình hóa (MHH) với biểu đồ ca sử dụng (Use-case Diagram)
 2. Mô hình hóa với biểu đồ hoạt động (Activity Diagram)
3. Bài tập thực hành



2

2

Nội dung trình bày

1. Mục đích phân tích chức năng (PTCN)
2. Các biểu đồ dùng để mô hình hóa chức năng
 1. Mô hình hóa (MHH) với biểu đồ ca sử dụng (Use-case Diagram)
 2. Mô hình hóa với biểu đồ hoạt động (Activity Diagram)
3. Bài tập thực hành



3

3

1. Mục đích Phân tích chức năng

- Cho một cái nhìn sơ bộ về tổng quan chức năng của hệ thống/ phần mềm.
- Sự phân tích tập trung trả lời câu hỏi What – hệ thống đáp ứng được những gì cho người dùng, không đi sâu vào câu hỏi How – hệ thống làm việc đó như thế nào.
- Làm tài liệu đặc tả chi tiết cho các chức năng làm căn cứ trong quá trình trao đổi xây dựng hệ thống, dẫn dắt ảnh hưởng đến toàn bộ các góc nhìn phát triển hệ thống còn lại, đảm bảo hệ thống phát triển theo đúng mong đợi của người dùng.

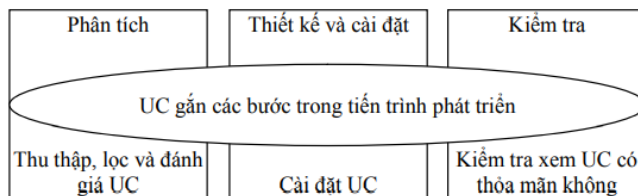


4

4

1. Mục đích Phân tích chức năng

Làm rõ các chức năng lớn của HT mà người dùng mong đợi (gọi là các ca sử dụng - usecase).



Hình 3.1 UC và tiến trình phát triển

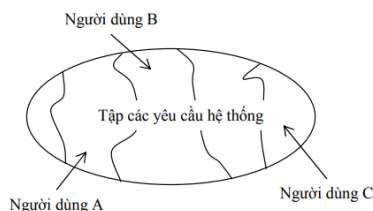


5

5

1. Mục đích Phân tích chức năng

- Làm sao để tìm kiếm các yêu cầu chức năng của hệ thống?
- Phân tích từ:
 - Phỏng vấn người sử dụng cuối
 - Khảo sát tài liệu họ đang sử dụng
 - Làm việc với các chuyên gia, người quản lý dự án có kinh nghiệm xử lý trong lĩnh vực đang xem xét



Hình 3.3 Phân hoạch yêu cầu bằng UC



6

6

1. Mục đích Phân tích chức năng

- Diễn tả kết quả phân tích bằng một biểu đồ ca sử dụng (use case diagram) trong UML, sử dụng Activity Diagram để đặc tả hành vi của từng usecase thực hiện
- Trình tự công việc MHH tính năng hệ thống như sau:
 - MHH môi trường của HT bằng các tác nhân (xác định tác nhân)
 - MHH nhu cầu bằng các ca sử dụng (xác định ca sử dụng)
 - Thiết lập các mối liên quan (mối quan hệ)
 - Lập biểu đồ ca sử dụng
 - Đặc tả chi tiết cho từng ca sử dụng (có thể dung AD hỗ trợ)



7

7

Nội dung trình bày

1. Mục đích phân tích chức năng (PTCN)

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa (MHH) với biểu đồ ca sử dụng (Use-case Diagram)

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)

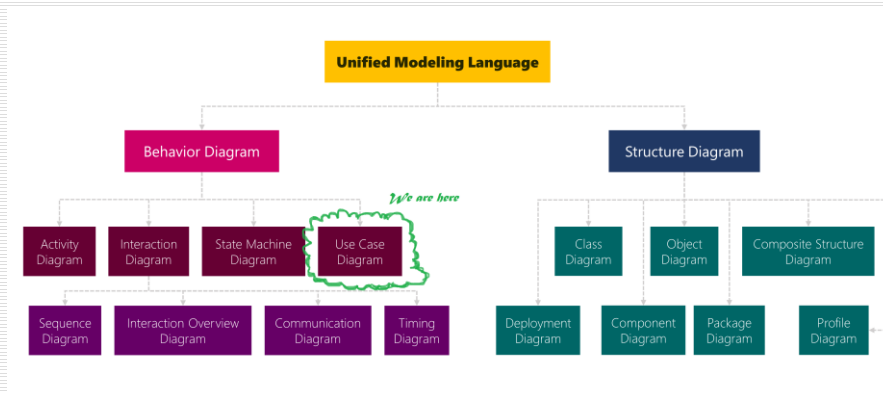
3. Bài tập thực hành



8

8

2. Các biểu đồ dùng để mô hình hóa chức năng



9

9

Nội dung trình bày

1. Mục đích phân tích chức năng (PTCN)
2. Các biểu đồ dùng để mô hình hóa chức năng
 - 2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)
 - 2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)
3. Bài tập thực hành



10

10

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



11

11

2.1 MHH với biểu đồ ca sử dụng (UseCase Diagram)

2.1.1 UseCase – Ca sử dụng là gì?

*Use Case là kỹ thuật dùng để **mô tả sự tương tác** giữa người dùng và hệ thống với nhau, trong một **môi trường cụ thể** và vì một **mục đích cụ thể**.*

Sự tương tác ở đây có thể là:

•Người dùng tương tác với hệ thống như thế nào?

•Hoặc, hệ thống tương tác với các hệ thống khác như thế nào?

Môi trường cụ thể, tức là *nằm trong một bối cảnh, phạm vi chức năng cụ thể, hoặc rộng hơn là trong một hệ thống/ phần mềm cụ thể.*

Mục đích cụ thể. *Diễn tả được yêu cầu theo góc nhìn cụ thể từ phía người dùng*



12

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



13

13

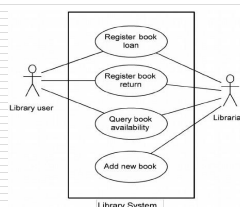
2.1 MHH với biểu đồ ca sử dụng (UseCase Diagram)

2.1.2 Hình thức thể hiện của Usecase

Tồn tại dưới 2 hình thức:

- **Biểu đồ UseCase (Use Case Diagram)**

- Dạng **hình vẽ/ ký hiệu**



- **Đặc tả Use Case (Use Case**

Specification/Use Case Scenarios)

- Dạng một **bảng ghi chú** mô tả **rất chi tiết các thông tin về Use Case**

USE CASE SPECIFICATION	
Use Case ID	UC-1.1
Use Case Name	Công việc
Description	Là người dùng có quyền đăng nhập vào ứng dụng để có quyền truy cập vào ứng dụng.
Actor(s)	Người dùng, Quản lý, Khách hàng
Priority	Thấp
Trigger	Người dùng muốn đăng nhập vào ứng dụng.
Preconditions	<ul style="list-style-type: none"> • Hệ thống phải được khởi tạo thành công. • Người dùng phải có tài khoản đăng ký. • Tài khoản phải được kích hoạt.
Postconditions	<ul style="list-style-type: none"> • Người dùng đã đăng nhập thành công. • Hệ thống ghi nhận trạng thái đăng nhập thành công của người dùng.
Basic Flow	<ol style="list-style-type: none"> 1. Người dùng nhập tên đăng nhập và mật khẩu. 2. Người dùng nhấn vào nút đăng nhập. 3. Người dùng được chuyển đến trang chủ. 4. Hệ thống xác thực thông tin đăng nhập và chuyển người dùng đến trang chủ.



14

14

2.1 MHH với biểu đồ ca sử dụng (UseCase Diagram)

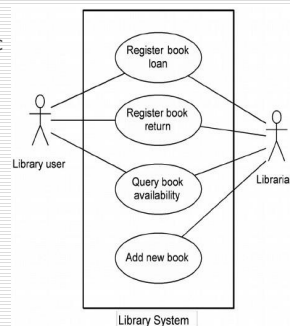
2.1.2 Hình thức thể hiện của Usecase

Biểu đồ Use case

- ✓ Sử dụng một tập các ký hiệu để thể hiện nội dung phân tích hệ thống về mặt chức năng
- ✓ Gồm tổ hợp: người dùng hệ thống (actors) và những tương tác của họ với hệ thống
- ✓ Mô tả hệ thống phải làm được những gì (what) không mô tả làm như thế nào (how)

UC giúp thể hiện được:

- ✓ Những ngữ cảnh tương tác của hệ thống/ ứng dụng với người dùng, với tổ chức khác, với hệ thống khác.
- ✓ Những tính năng mà hệ thống/ ứng dụng cung cấp cho người dùng
- ✓ Mô tả được những luồng sự kiện chính trong tính năng
- ✓ Phạm vi thực hiện của hệ thống



15

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase Diagram

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



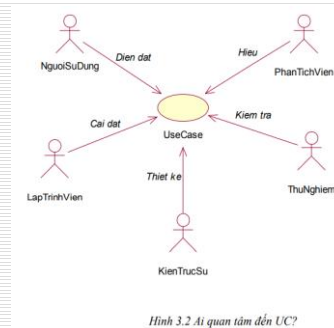
16

16

2.1 MHH với biểu đồ ca sử dụng (Use-case Diagram)

2.1.3 Mục đích sử dụng UC Diagram

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



17

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



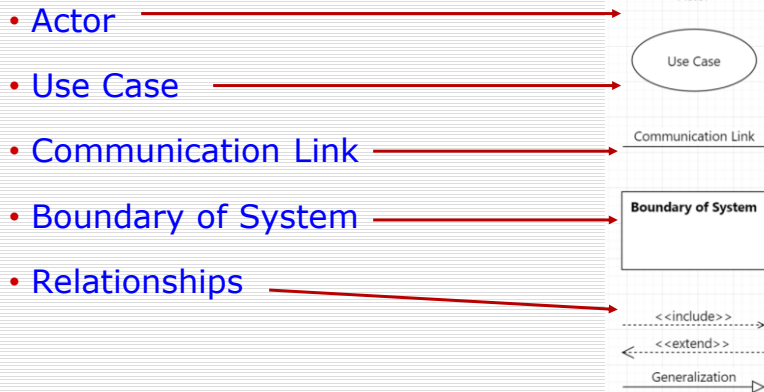
18

18

2.1 MHH với biểu đồ ca sử dụng (Use-case Diagram)

2.1.4 Các ký pháp chính của biểu đồ Usecase

5 ký pháp thường dùng:



19

19

2.1.4 Các ký pháp chính của biểu đồ Usecase

Actor – Tác nhân



20

20

2.1.4 Các ký pháp chính của biểu đồ Usecase

• Tác nhân (Actor)

là bất kỳ thứ gì tương tác với hệ thống, có sự trao đổi dữ liệu với hệ thống, tác nhân có thể là:

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



Lưu ý: Actor là một lớp/loại người dùng (có vai trò trong hệ thống) chứ không phải một người cụ thể

- Ví dụ: trong hệ thống thông tin bệnh viện
- Tác nhân: bác sỹ, y tá, bệnh nhân ...
- Dr. John, Mrs. Brown không phải là tác nhân



21

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quy tắc xác định tác nhân của hệ thống

- Đặt các câu hỏi sau
 - Ai là người sử dụng hệ thống?
 - Ai sẽ là Admin của hệ thống (tức người cài đặt, quản lý, bảo trì... hệ thống)?
 - Hệ thống này có được sử dụng bởi bất kỳ một hệ thống nào khác không? (*)
 - Hệ thống lưu trữ dữ liệu, vậy ai là người input dữ liệu vào hệ thống?
 - Hệ thống lưu trữ dữ liệu, vậy ai là người cần những dữ liệu output?
- Thông tin về tác nhân
 - Tên tác nhân phải mô tả vai trò của tác nhân đó một cách rõ ràng
 - Tên actor nên là **danh từ (Noun)**
 - Cần hình dung khái quát khả năng của tác nhân đó



22

2.1.4 Các ký pháp chính của biểu đồ Usecase

UseCase – Ca sử dụng



23

23

2.1.4 Các ký pháp chính của biểu đồ Usecase

• Ký pháp mô tả Use case

- Một use case mô tả hành động của hệ thống nhằm mang đến một giá trị nào đó cho tác nhân.
- Ký hiệu: hình Oval mô tả chức năng thực hiện của hệ thống
- Tên use case: **cụm động từ (Verb +Noun)**


 Usecase

- Một use case phải bao hàm một chuỗi hành động (flow of events) giúp thực hiện một mục tiêu cụ thể, kết quả trả lại cho tác nhân ví dụ:

Ví dụ: *Đổi điểm thành viên*, *Rút tiền*, *Chuyển tiền nội địa*, *Chuyển tiền quốc tế*, *Duyệt nhận xét bài viết*.



24

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quy tắc xác định use case của hệ thống

- 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?



25

2.1.4 Các ký pháp chính của biểu đồ Usecase

Communication Link & Boundary System



26

26

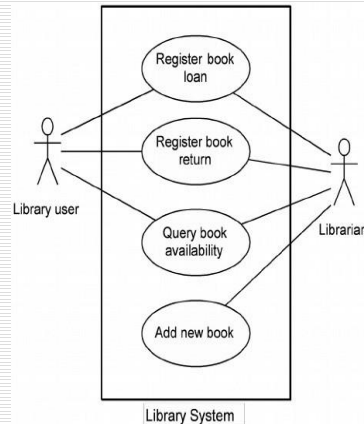
2.1.4 Các ký pháp chính của biểu đồ Usecase

Communication Link thể hiện sự tương tác giữa Actor với System. Nối giữa Actor với Use Case.

Boundary of System là phạm vi các tương tác xảy ra, có thể là trong một hệ thống, một module, hoặc một tính năng bất kỳ.

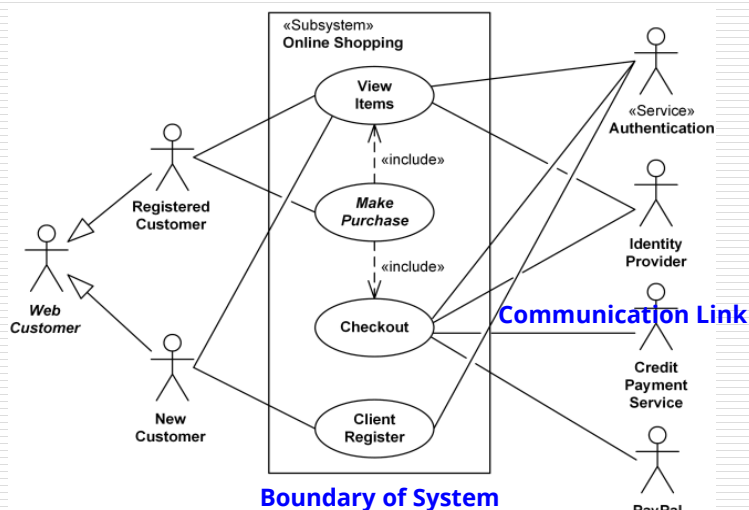
Lưu ý:

- Luôn phải xác định tên của khung hệ thống.
- UseCase đặt bên trong khung hệ thống
- Actors đặt bên ngoài khung hệ thống



27

2.1.4 Các ký pháp chính của biểu đồ Usecase



28

2.1.4 Các ký pháp chính của biểu đồ Usecase

Relationships – Quan hệ



29

29

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quan hệ (relationship)

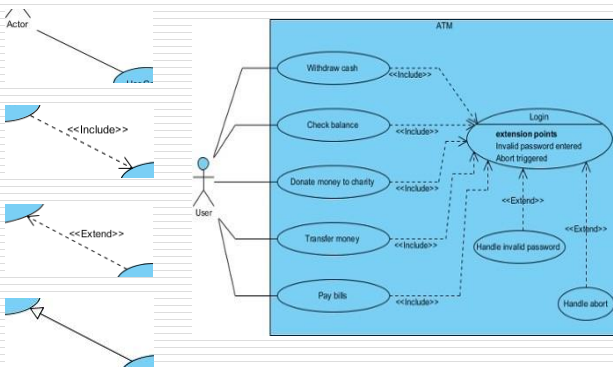
4 loại mối quan hệ:

✓ **Association**

✓ **Include**

✓ **Extend**

✓ **Generalization**

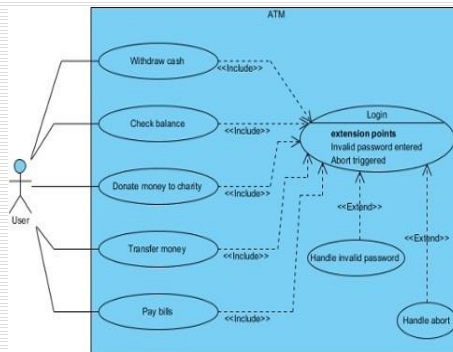
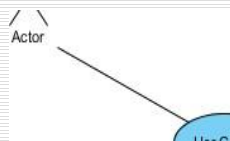


30

2.1.4 Các ký pháp chính của biểu đồ Usecase

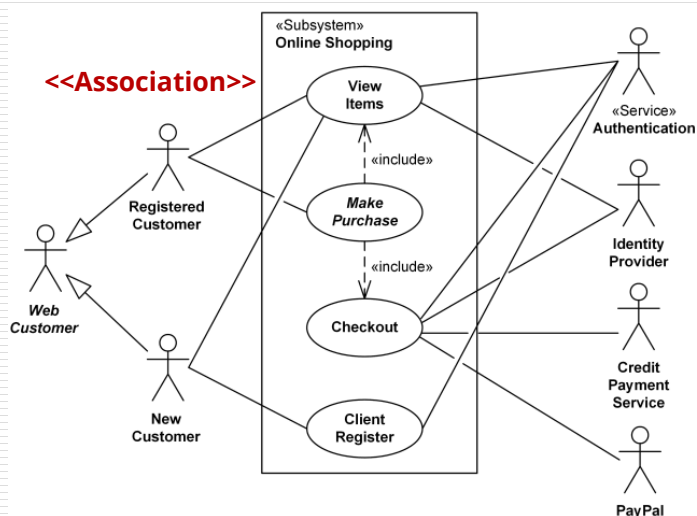
Quan hệ <<Association>>

- ✓ **Association:** Đường thẳng nối giữa actors và use cases thể hiện cho sự tương tác, gắn kết giữa chúng (*Communication Link*)



31


2.1.4 Các ký pháp chính của biểu đồ Usecase

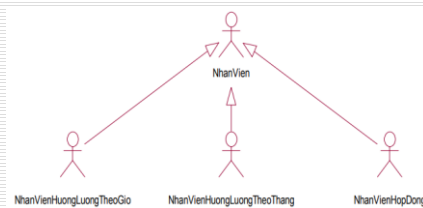
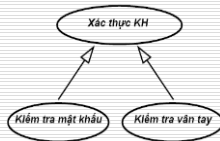


32

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quan hệ <<generalization>>

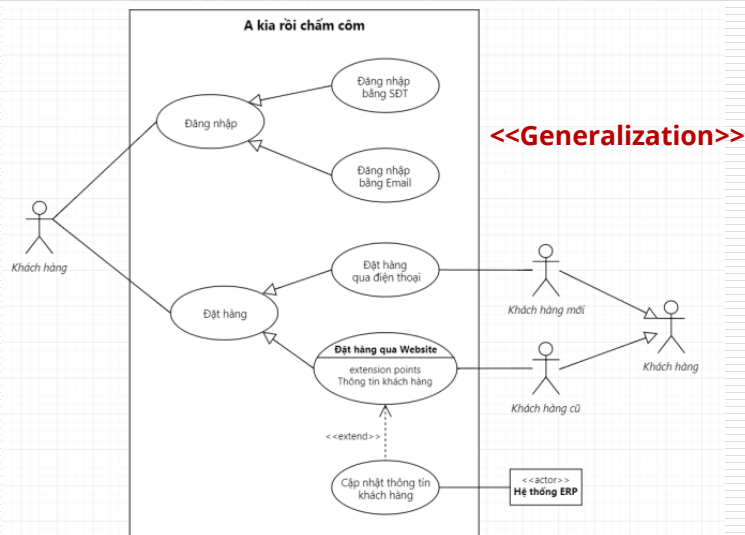
- thể hiện mối quan hệ kế thừa giữa các thành phần cùng loại của mô hình - chỉ ra một vài tính chất chung của một nhóm tác nhân hoặc UC
- Sử dụng khái niệm kế thừa
 - Mô tả hành vi chung (chia sẻ) trong UC cha
 - Mô tả hành vi riêng trong (các) UC con
- Biểu diễn: 



33

2.1.4 Các ký pháp chính của biểu đồ Usecase

- Ví dụ

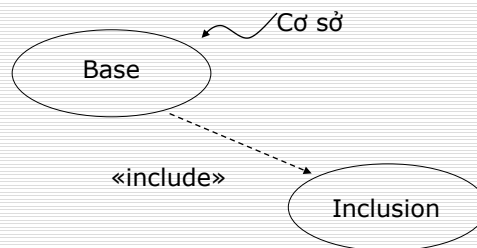


34

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quan hệ <<include>>

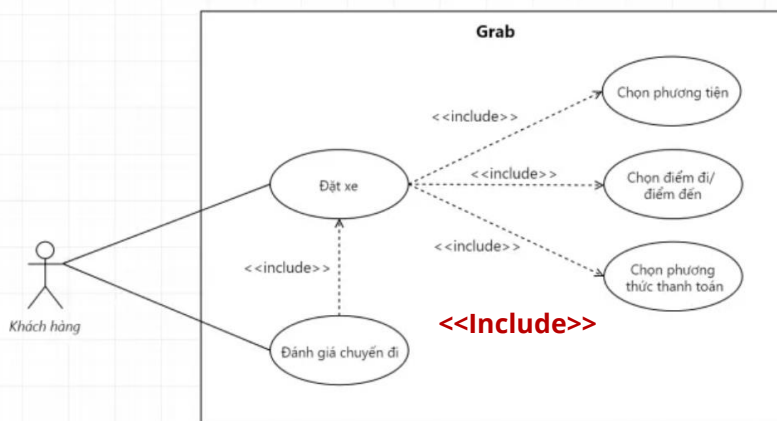
- xác định rằng trong một usecase này bao hàm hành vi được định nghĩa trong một use case khác - mỗi quan hệ **bắt buộc phải có** giữa các Use Case với nhau.
- Chức năng của UC Inclusion sẽ được gọi thực hiện trong UC Base



35

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quan hệ <<include>> - ví dụ



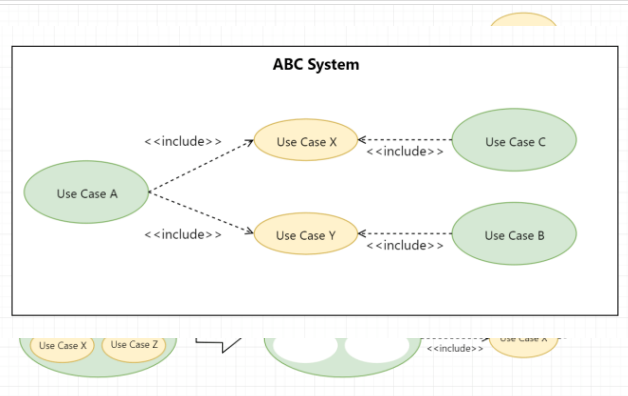
36

2.1.4 Các ký pháp chính của biểu đồ Usecase

Điểm cần lưu ý khi vẽ Include

- Chỉ nên tách Use Case khi UC có **độ phức tạp lớn** và những thứ tách ra được có thể được **tận dụng ở các Use Case khác**.

Quan hệ <<include>>

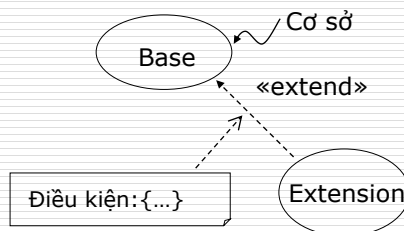


37

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quan hệ <<extend>>

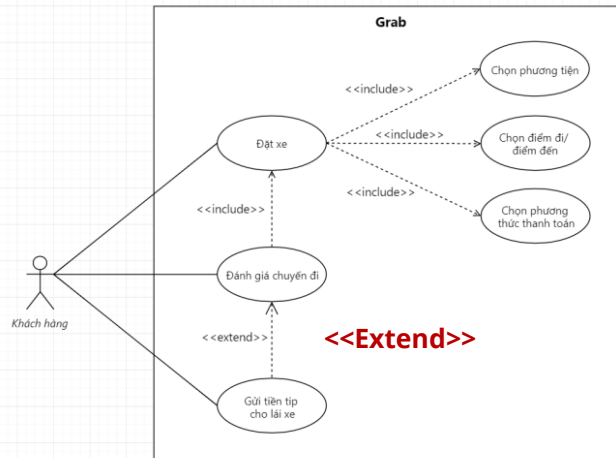
- Chỉ định một usecase có thể mở rộng thêm những use case khác
- Thể hiện mối quan hệ **có thể có** hoặc **có thể không** giữa các Use Case với nhau.
- Chèn hàng vi của UC Extension vào UC Base
 - Chỉ chèn khi điều kiện extend đúng (mở rộng, phát sinh)
 - Chèn vào lớp cơ sở tại điểm phát sinh (extension point)



38

2.1.4 Các ký pháp chính của biểu đồ Usecase

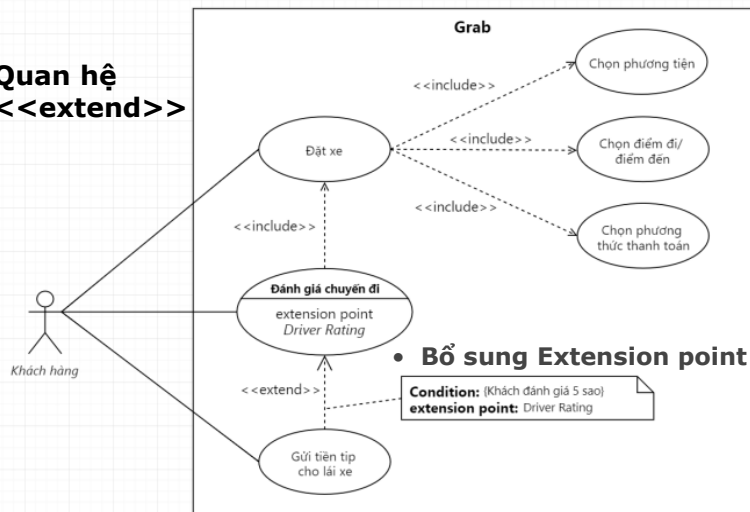
- Ví dụ 1:



39

2.1.4 Các ký pháp chính của biểu đồ Usecase

Quan hệ
<<extend>>

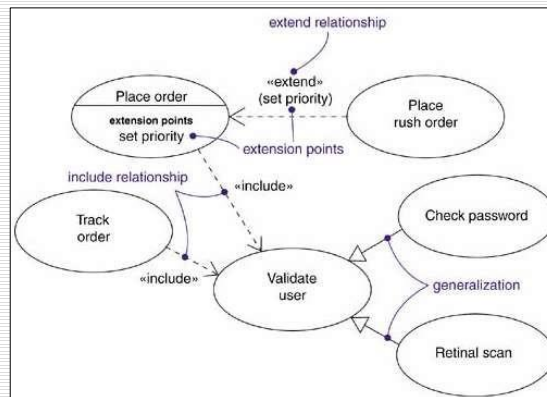


40

2.1.4 Các ký pháp chính của biểu đồ Usecase

Khác biệt giữa Extend và Include

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



41

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



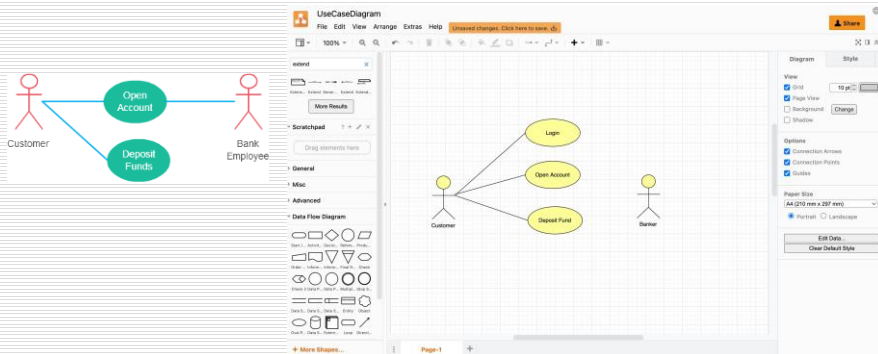
42

42

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

Thực hành vẽ biểu đồ usecase

- ✓ Tạo biểu đồ usecase qua công cụ:
- ✓ Công cụ: <https://app.diagrams.net/>

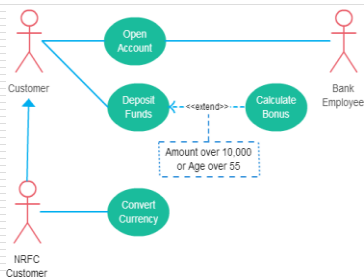


43

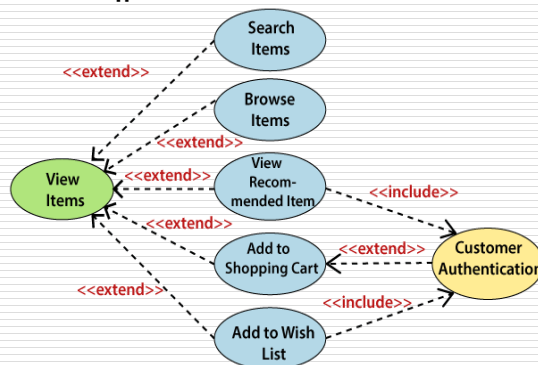
2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

Tạo lập và đọc hiểu các biểu đồ dưới đây.

Bài tập 1



Bài tập 2



44

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

Những gì use case thể hiện

- ✓ Ai là người sử dụng hệ thống
- ✓ Người dùng muốn làm được gì trong hệ thống
- ✓ Mục tiêu của người dùng
- ✓ Các bước thực hiện mà người dùng phải bám sát để thực hiện một việc cụ thể

Những thứ use cases không thể hiện

- ✓ Ngôn ngữ cài đặt hệ thống cụ thể
- ✓ Mô tả chi tiết về giao diện người dùng và hay các màn hình làm việc của hệ thống



45

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

Các bước để phát triển và vẽ biểu đồ use case:

- ✓ **Bước 1:** Xác định tất cả những người/ nhóm người sẽ sử dụng hệ thống
- ✓ **Bước 2:** Chọn ra một trong số những người/ nhóm người trong bước 1
- ✓ **Bước 3:** Xác định xem người dùng đó muốn làm gì với hệ thống. Mỗi việc người đó làm trên hệ thống được coi là một use case
- ✓ **Bước 4:** Đối với mỗi use case, xác định các chuỗi sự kiện thường xuyên xảy ra khi người dùng thực hiện nó.
- ✓ **Bước 5:** Mô tả một bản đặc tả đơn giản cho use case về việc người dùng sẽ làm được gì và hệ thống sẽ đáp ứng gì cho họ
- ✓ **Bước 6:** Xem xét các sự kiện trong các tình huống thay thế và thêm sự kiện đầy vào biểu đồ như là một use case mở rộng (extend)
- ✓ **Bước 7:** Tìm kiếm điểm tương đồng giữa các use case. Tách điểm tương đồng ra và tập hợp lại thành các usecase tổng quát.
- ✓ **Bước 8:** Lặp lại từ bước 2 đến bước 7 cho các người dùng còn lại.



46

Vẽ biểu đồ use case theo mô tả hệ thống TMDT sau:

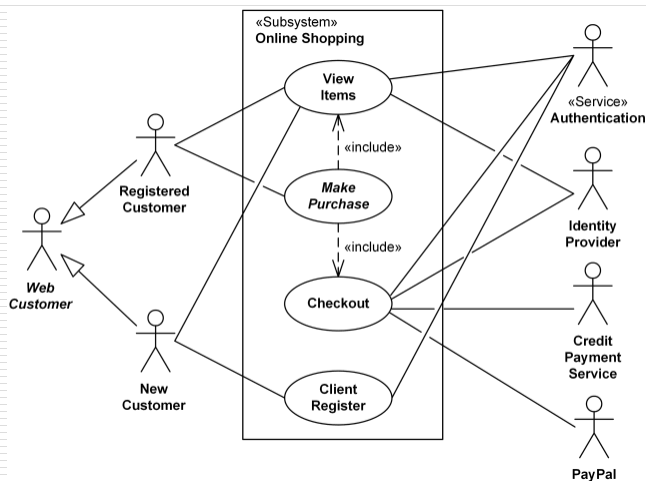
Khi khách hàng ghé thăm trang web họ có thể duyệt xem sản phẩm bán trên trang hoặc tìm kiếm sản phẩm mình quan tâm. Khi thấy sản phẩm mong muốn họ có thể vào xem chi tiết về sản phẩm đó và nếu quyết định lựa chọn sản phẩm, họ sẽ đưa sản phẩm đó vào giỏ hàng và có thể quay lại trang chủ để xem tiếp các sản phẩm khác. Ngoài ra, trong quá trình duyệt xem sản phẩm nếu khách hàng đã đăng nhập thì có thể đọc thêm các đánh giá về sản phẩm hoặc đưa sản phẩm đang quan tâm vào danh sách sản phẩm yêu thích của riêng mình. Để xác thực khách hàng, khách hàng có thể đăng nhập hệ thống bằng nhiều hình thức: đăng nhập thông qua tài khoản đã đăng ký trực tiếp trên hệ thống, đăng nhập thông qua giao thức SSO (Single Sign On) – hệ thống quản lý liên kết đa danh tính.

Khi sản phẩm đã ở trong giỏ hàng, khách hàng có thể tùy ý xem lại thông tin giỏ hàng ở bất kỳ thời điểm nào, thay đổi, thêm bớt số lượng sản phẩm cho đến khi quyết định đặt mua. Hệ thống sẽ tự động tính toán tổng số tiền cần thanh toán cho khách hàng trên giỏ hàng. Nếu khách hàng đặt mua, hệ thống sẽ hỗ trợ quy trình thanh toán hóa đơn gồm tính toán thuế để xuất hóa đơn VAT (nếu khách có nhu cầu), tính toán phí ship và lựa chọn thanh toán trực tuyến qua thẻ ngân hàng hoặc qua các đơn vị thanh toán trung gian là PayPal



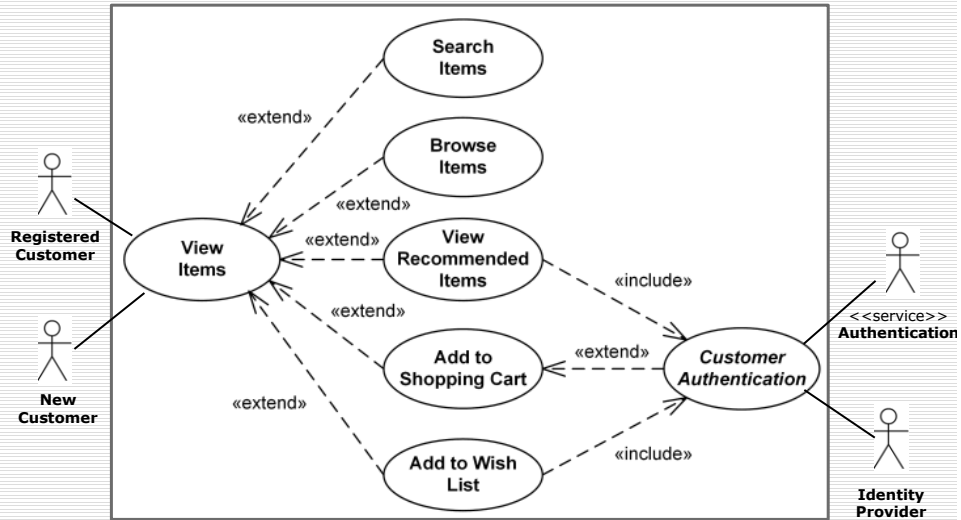
47

Vẽ biểu đồ use case cho hệ thống TMDT



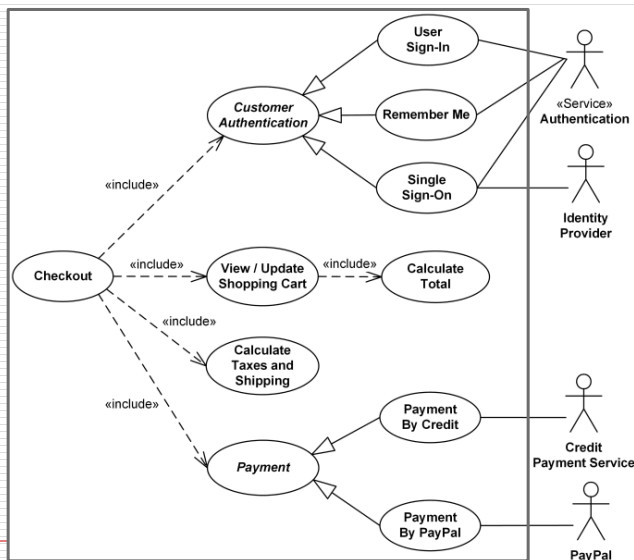
48

Vẽ biểu đồ use case cho hệ thống TMDT



49

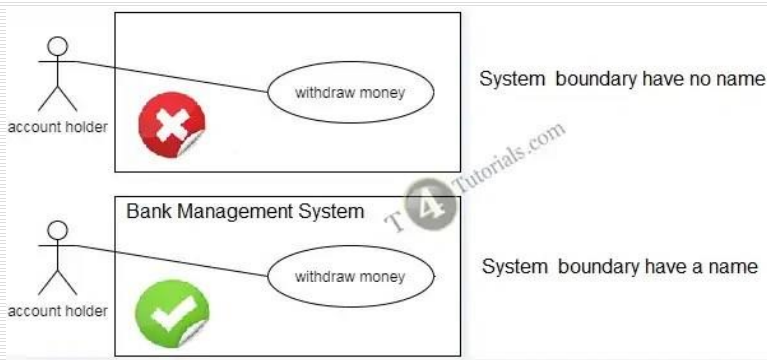
Vẽ biểu đồ use case cho hệ thống TMDT



50

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

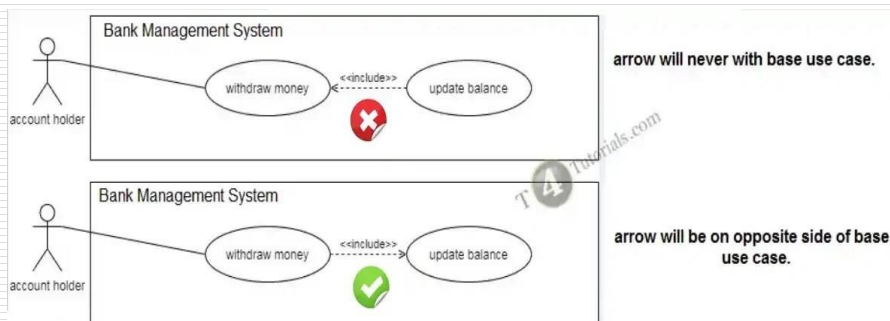
Lỗi thường gặp khi vẽ biểu đồ (1)



51

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

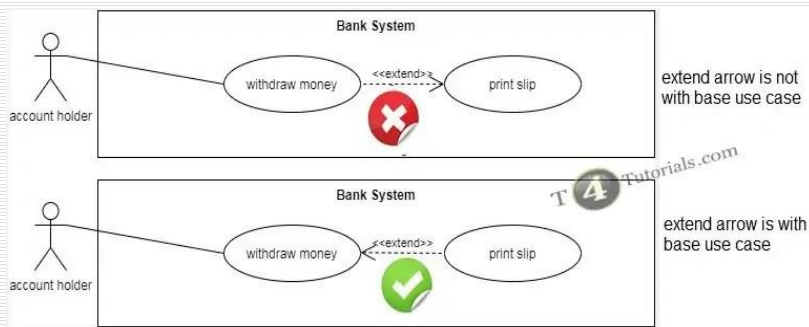
Lỗi thường gặp khi vẽ biểu đồ (2)



52

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

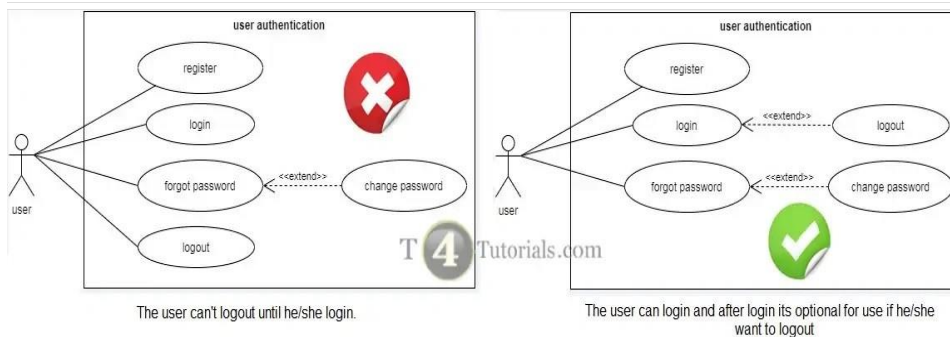
Lỗi thường gặp khi vẽ biểu đồ (3)



53

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

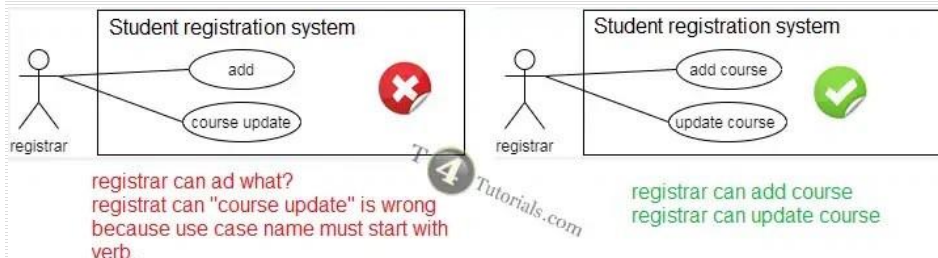
Lỗi thường gặp khi vẽ biểu đồ (4)



54

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

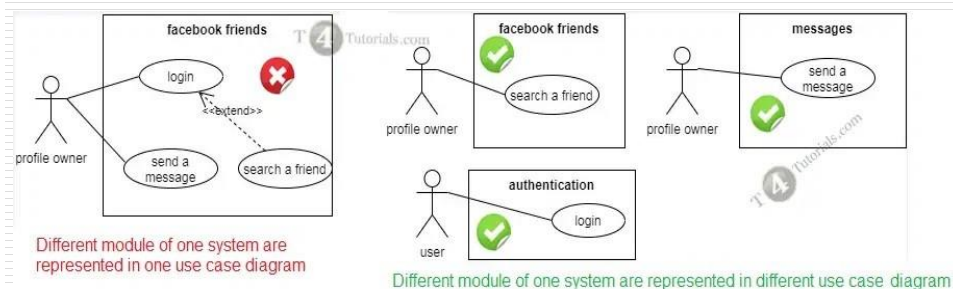
Lỗi thường gặp khi vẽ biểu đồ (5)



55

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

Lỗi thường gặp khi vẽ biểu đồ (6)



56

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

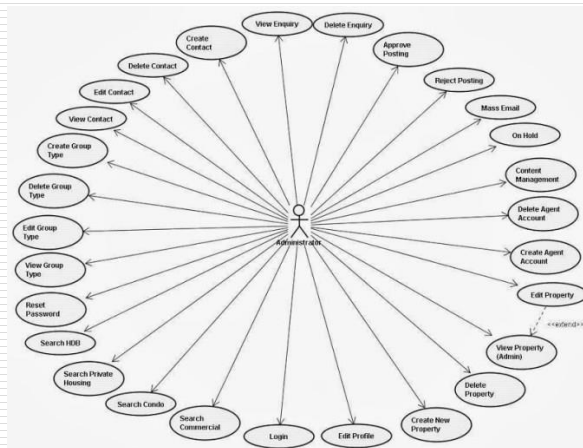
Lỗi thường gặp khi vẽ biểu đồ (7)



57

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

Lỗi thường gặp khi vẽ biểu đồ (8)



58

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

- Nếu Hệ thống phức tạp, quá nhiều Use case (hàng chục), nên chia Usecase Diagram thành nhiều mức theo quy tắc:

Level 0: Chỉ chứa những Usecase chính của hệ thống (có thể ở mức trừu tượng)

Level 1: Chỉ chứa các usecase ở mức phụ cho những usecase chính ở Level 0, có thể có nhiều usecase diagram ở mức level 1

Level n...

+ Ví dụ: "Quản lý sách" bao gồm "Nhập sách", "Xuất sách", "..."

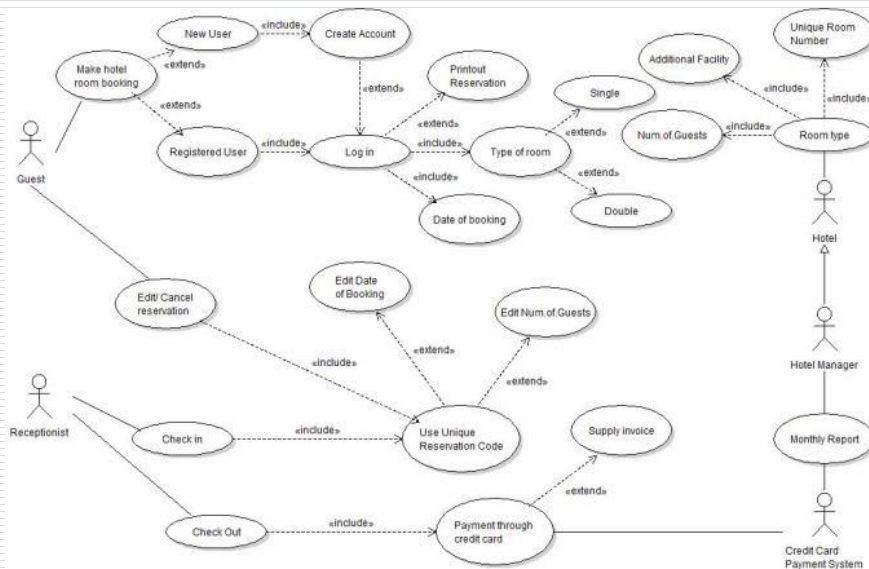
- Sử dụng các Use-case quá cụ thể, hoặc làm việc với dữ liệu quá cụ thể. Ví dụ:

- "Tìm sách theo tên" (nên là "Tìm sách")
- "Nhập Pin vào máy ATM" (nên là "Nhập PIN")



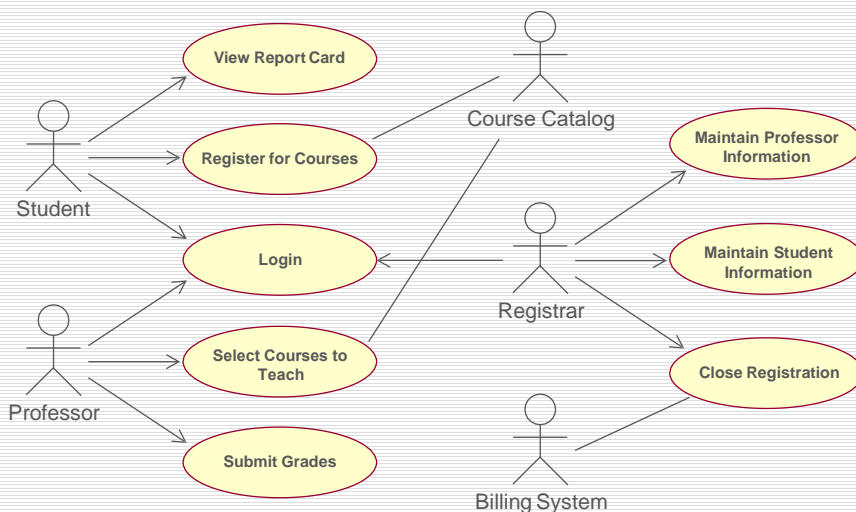
59

Tìm lỗi trong biểu đồ dưới đây và sửa lại cho đúng



60

Đọc biểu đồ use case sau đây



61

Đọc biểu đồ use case sau đây

- 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?



62

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



63

63

2.1.6 Các thông số cần quan tâm khi đặc tả UC

Cần mô tả UC để hiểu được chi tiết hoạt động, các thông số mô tả được gom thành 3 nhóm:

A. Summary: Mô tả cơ bản

B. Flow: Luồng thực hiện

C. Addition Information: Thông tin bổ sung



64

2.1.6 Các thông số cần quan tâm khi đặc tả UC

SUMMARY

Mô tả cơ bản



65

2.1.6 Các thông số cần quan tâm khi đặc tả UC

A. SUMMARY (Mô tả cơ bản)

- **Use Case Name:** Tên Use Case
- **Use Case ID:** Mã Use Case
- **Use Case Description:** Tóm gọn nhanh sự tương tác được thể hiện trong Use Case là gì (viết theo mẫu User Story).
- **Actor:** Những đối tượng thực hiện sự tương tác trong Use Case.
- **Priority:** Mức độ ưu tiên của Use Case so với các Use Case còn lại trong dự án (Must Have/Nice to have/High/ Medium/Low)
- **Trigger:** Điều kiện kích hoạt Use Case xảy ra.
- **Pre-Condition:** Điều kiện cần để Use Case thực hiện thành công.
- **Post-Condition:** Những thứ sẽ xuất hiện sau khi Use Case được thực hiện thành công.



66

2.1.6 Các thông số cần quan tâm khi đặc tả UC

A. SUMMARY Ví dụ Use Case rút tiền tại máy ATM.

Use Case:	Rút tiền ATM
Use Case ID	UC01
Actor	Khách hàng
Description	Là Khách hàng tôi muốn rút tiền để lấy được tiền mặt từ TK ngân hàng của tôi
Trigger	Khách hàng thực hiện lệnh rút tiền (bấm nút rút)..
Pre-Condition	<ul style="list-style-type: none"> Khách hàng đã đút thẻ vào máy ATM. Số dư lớn hơn 50,000đ
Post-Condition	<ul style="list-style-type: none"> Khách hàng nhận được tiền mặt sau khi thực hiện rút tiền. Số dư của khách hàng đã bị trừ đi khoảng tiền đã rút.
Priority	<ul style="list-style-type: none"> Must Have (Bắt buộc phải có)



67

Phân biệt Trigger và Pre-Condition

Trigger nghĩa là một thứ gì đó để **kích hoạt** cho Use Case chạy, **khởi xướng** cho Use Case chạy.

Pre-Condition nghĩa là một thứ gì đó, mà **phải có nó** thì **Use Case mới chạy** được.

Lưu ý:

- ✓ Use Case có thể có Pre-Condition hoặc không, nhưng Trigger thì phải có.
- ✓ Output (hoặc Post-Condition) của Use Case này có thể là trigger của Use Case khác.

Ví dụ:

Use Case (UC01): Rút tiền tại máy ATM đã được thực hiện xong. Post condition của UC01 cũng là trigger cho UC02 tiếp theo: **Gửi tin nhắn SMS thông báo cho khách hàng.**



68

2.1.6 Các thông số cần quan tâm khi đặc tả UC

FLOW

Luồng thực hiện



69

2.1.6 Các thông số cần quan tâm khi đặc tả UC

CÓ 3 LOẠI LƯỒNG THỰC HIỆN:

•**Basic Flow:** luồng tương tác CHÍNH giữa các Actor và System để Use Case thực hiện thành công.

•**Alternative Flow:** luồng tương tác THAY THẾ giữa các Actor và System để Use Case **thực hiện thành công**.

•**Exception Flow:** luồng tương tác NGOẠI LỆ giữa các Actor và System mà Use Case **thực hiện thất bại**.

	BASIC FLOW	ALTERNATIVE FLOW	EXCEPTION FLOW
Ý ĐỒ THỂ HIỆN	Luồng đi đơn giản nhất	Các luồng đi khác	Các luồng đi khác
KẾT QUẢ DẪN TỚI	Use Case xảy ra thành công	Use Case xảy ra thành công	Use Case xảy ra thất bại



70

2.1.6 Các thông số cần quan tâm khi đặc tả UC

Ví dụ Flow

Trong mô hình quản lý e-Learning, có **Use Case: Hủy kích hoạt tài khoản học viên**



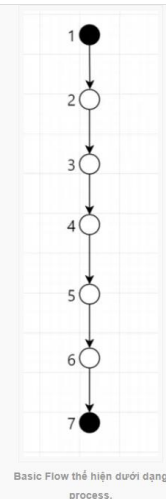
71

2.1.6 Các thông số cần quan tâm khi đặc tả UC

Use Case: Hủy kích hoạt tài khoản học viên

BASIC FLOW

- 1.Admin mở tài khoản học viên cần hủy kích hoạt
- 2.Hệ thống hiển thị màn hình thông tin học viên
- 3.Admin chọn lệnh hủy kích hoạt
- 4.Hệ thống yêu cầu nhập mã OTP để xác nhận
- 5.Admin nhập đúng mã OTP để xác nhận lệnh hủy kích hoạt
- 6.Hệ thống kiểm tra mã OTP và tiến hành hủy kích hoạt
- 7.Hệ thống hiển thị thông báo đã hủy kích hoạt.



72

2.1.6 Các thông số cần quan tâm khi đặc tả UC

ALTERNATIVE FLOW

1a. Admin chọn học viên cần hủy kích hoạt ở lưới học viên.

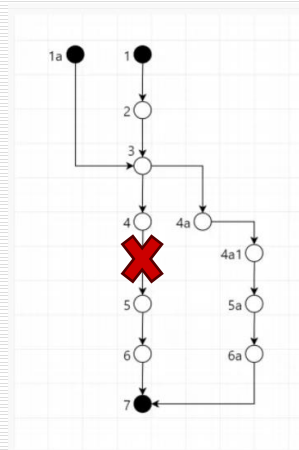
Use Case tiếp tục bước 3.

4a. Admin chọn phương thức xác nhận khác: Xác nhận qua mã reCaptcha

4a1. Hệ thống hiển thị mã reCaptcha và yêu cầu nhập mã reCaptcha để xác nhận

5a. Admin nhập đúng mã reCaptcha để xác nhận lệnh hủy kích hoạt

6a. Hệ thống kiểm tra mã reCaptcha và tiến hành hủy kích hoạt *Use Case tiếp tục bước 7.*



Alternative Flow thể hiện dưới dạng process.



73

2.1.6 Các thông số cần quan tâm khi đặc tả UC

EXCEPTION FLOW

5b. Admin nhập sai mã reCaptcha.

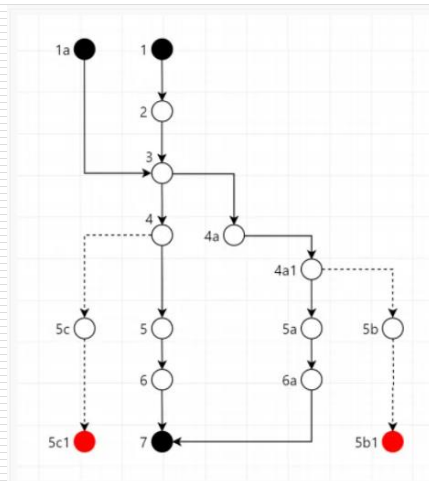
5b1. Hệ thống báo lỗi và hủy bỏ lệnh hủy kích hoạt học viên.

Use Case dừng lại.

5c. Admin nhập sai mã OTP.

5c1. Hệ thống báo lỗi và hủy bỏ lệnh hủy kích hoạt học viên.

Use Case dừng lại.



Exception Flow thể hiện dưới dạng process.



74

2.1.6 Các thông số cần quan tâm khi đặc tả UC

Addition Information

Thông tin bổ sung



75

2.1.6 Các thông số cần quan tâm khi đặc tả UC

C. Additional Information

•**Business Rule:** các quy định về mặt Business mà hệ thống bắt buộc phải nghe theo, làm theo.

•**Non-Functional Requirement:** Vì Use Case chỉ dùng để thể hiện [Functional Requirement](#), nên phải bổ sung các [yêu cầu về Non-Functional](#) ở đây.

Lưu ý:

Business Rule là các quy định, hoặc các policy của khách hàng. Có sao ghi vậy. Vì đây là giai đoạn tài liệu hóa yêu cầu thông qua *Use Case*, nên cứ liệt kê hết các Business Rule có liên quan trong tài liệu đặc tả.



76

2.1.6 Các thông số cần quan tâm khi đặc tả UC

C. Additional Information

Ví dụ:

Business Rules	BR1.1-1: Người dùng nhập sai thông tin đăng nhập ở lần thứ 6 liên tiếp sẽ bị khóa tài khoản 30 phút.
Non-Functional Requirement	NFR1.1-1: Time out cho màn hình đăng nhập dưới 60 giây. NFR1.1-2: Mật khẩu của người dùng phải được hash bằng MD5.



77

Tổng kết: Lợi ích của UseCase

1. Use Case giúp **thể hiện được rõ Requirement theo góc nhìn của người dùng cuối** (rất quan trọng, vì nó giúp hiểu rõ bản chất vấn đề).
2. Những gì được thể hiện trong Use Case rất **tự nhiên**, ai đọc cũng hiểu.
3. Use Case có thể **chia nhỏ phạm vi** theo nhiều phân hệ, hoặc cụm tính năng. Và bản thân nó cũng có thể nhìn dưới **góc độ high-level**. Do đó, dễ hơn rất nhiều trong việc mô tả bao trùm đủ các yêu cầu trong một dự án lớn.
4. Use Case là **bước đệm tuyệt vời** giữa việc *mô tả tổng quát* và *mô tả chi tiết sự tương tác* thông qua Sequence Diagram (biểu đồ hành vi thuộc họ UML).
5. Use Case được dùng để tạo các Epic, và các User Stories trong dự án Scrum, làm mọi thứ được nhất quán và rất chặt chẽ.
6. Use Case còn được dùng để tạo các Test Case.



78

Nội dung trình bày

2. Các biểu đồ dùng để mô hình hóa chức năng

2.1 Mô hình hóa với biểu đồ ca sử dụng (Usecase Diagram)

2.1.1 Use Case là gì?

2.1.2 Hình thức thể hiện của Use Case

2.1.3 Mục đích sử dụng UseCase

2.1.4 Các ký pháp sử dụng trong biểu đồ UC

2.1.5 Cách vẽ biểu đồ UC và lỗi sai thường gặp

2.1.6 Các thông số cần quan tâm khi đặc tả UC

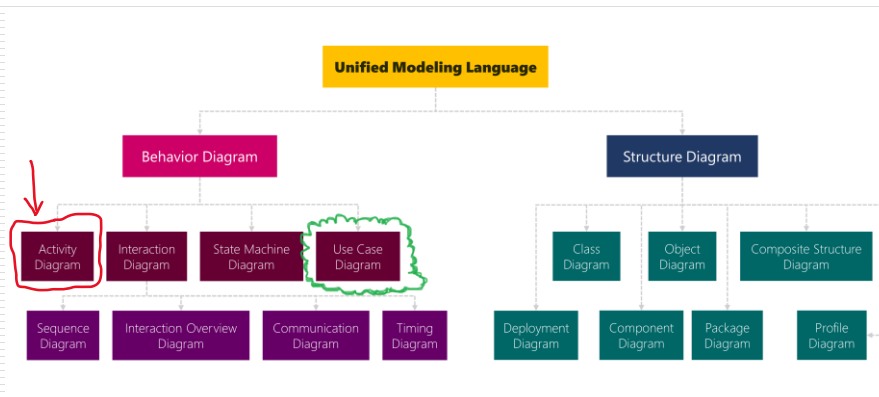
2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)



79

79

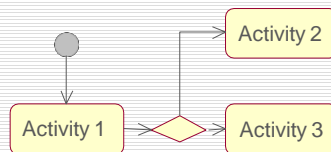
2.2 MHH với biểu đồ hoạt động (Activity Diagram)



80

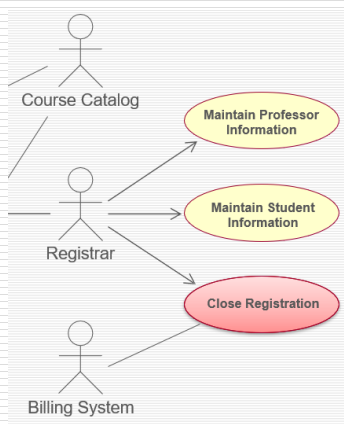
2.2 MHH với biểu đồ hoạt động (Activity Diagram)

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



81

2.2 MHH với biểu đồ hoạt động (Activity Diagram)

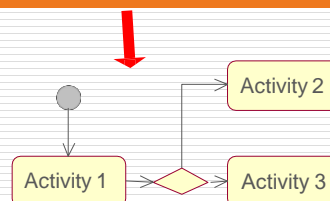


Flow of Events

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

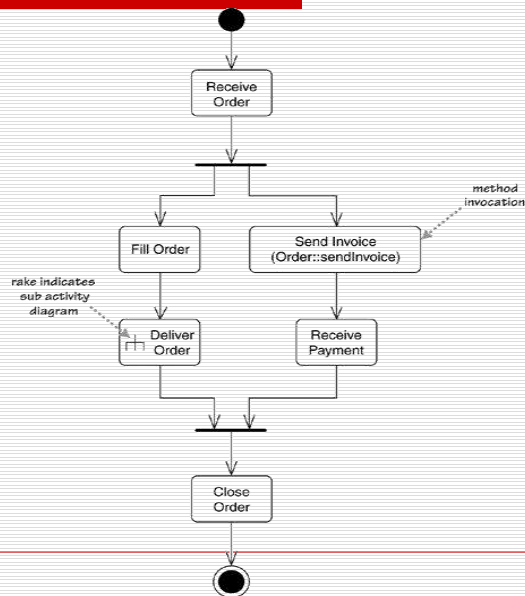
1. The system checks to see if registration is in progress. If it is, then a message is displayed to the Registrar and the use case terminates. The Close Registration processing cannot be performed if registration is in progress.

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.



82

2.2 MHH với biểu đồ hoạt động (Activity Diagram)

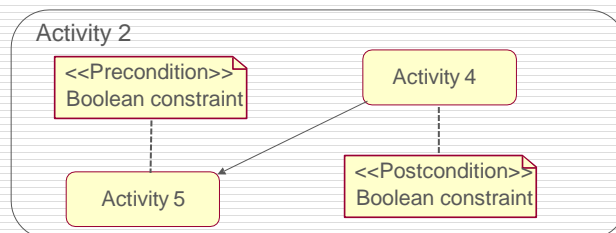


83

83

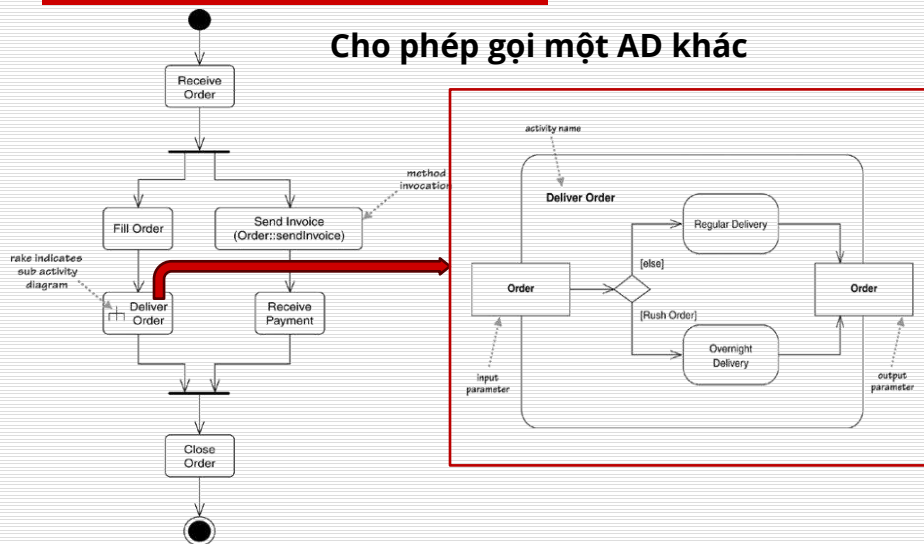
2.2 MHH với biểu đồ hoạt động (Activity Diagram)

- 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



84

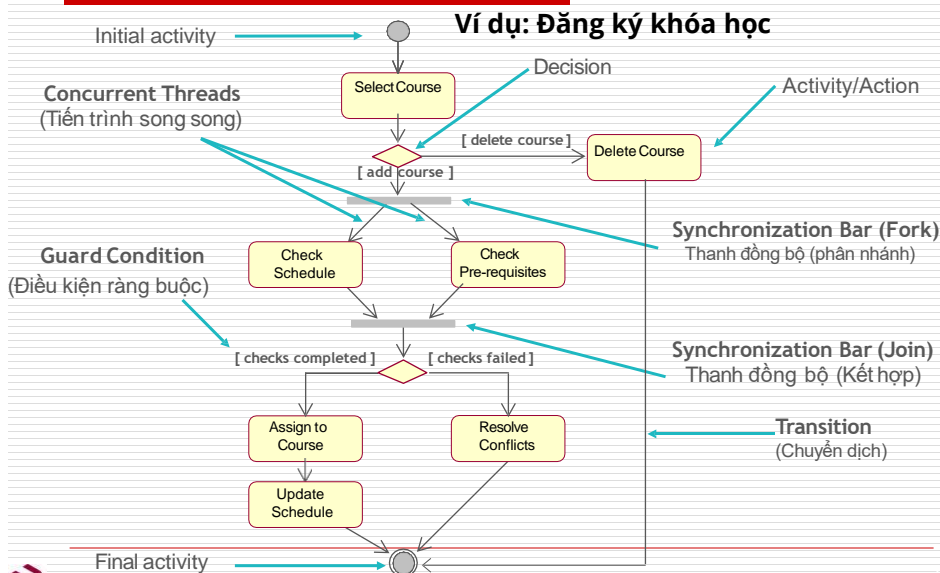
2.2 MHH với biểu đồ hoạt động (Activity Diagram)



85

85

2.2 MHH với biểu đồ hoạt động (Activity Diagram)



86

Nội dung trình bày

1. Mục đích phân tích chức năng (PTCN)
2. Các biểu đồ dùng để mô hình hóa chức năng
 - 2.1 Mô hình hóa (MHH) với biểu đồ ca sử dụng (Use-case Diagram)
 - 2.2 Mô hình hóa với biểu đồ hoạt động (Activity Diagram)
3. Bài tập thực hành



87

87

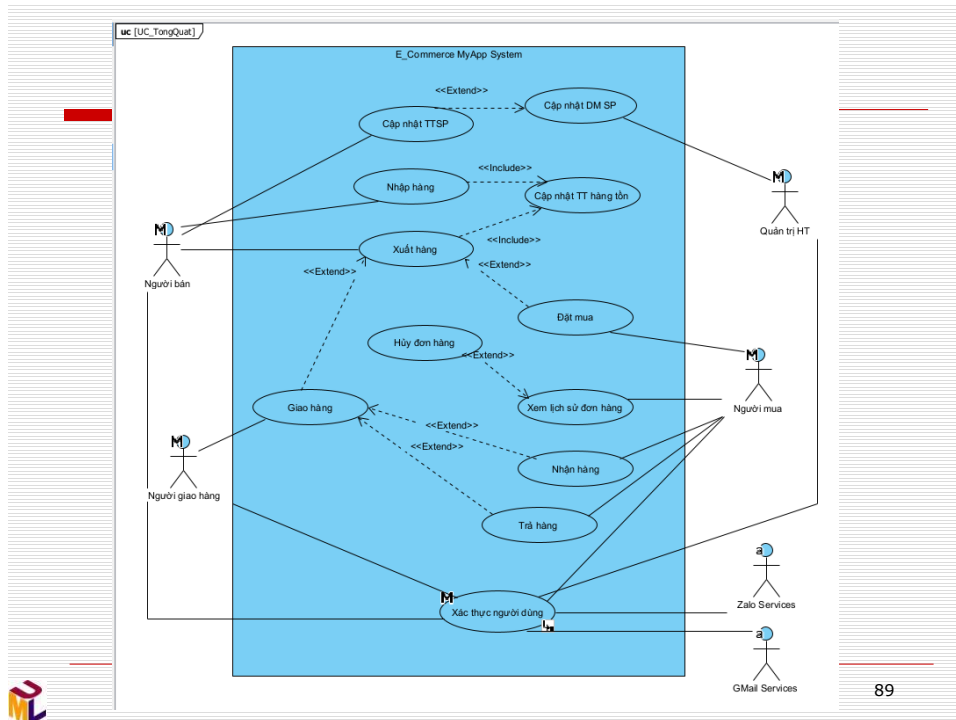
3. Bài tập thực hành

Ví dụ 1 : Hoạt động E_commerce MyApp

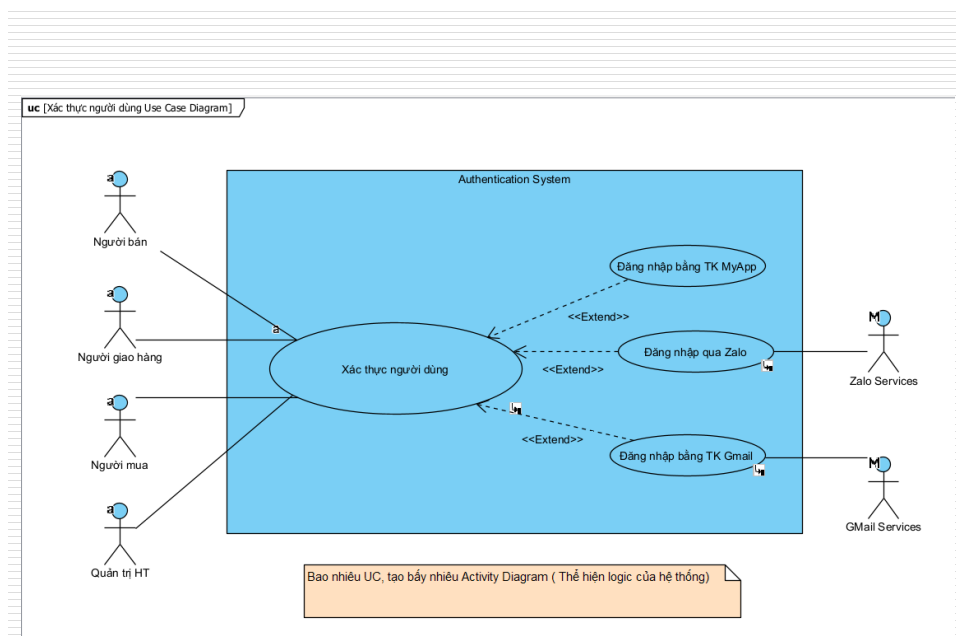


88

88



89



90

90

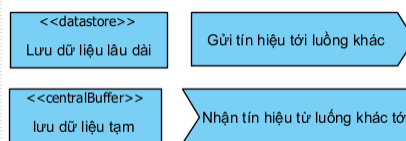
Kịch bản luồng chính

USECASE Xác thực người dùng

TRIGGER: Người dùng đang đứng sẵn trong trang Đăng Nhập.

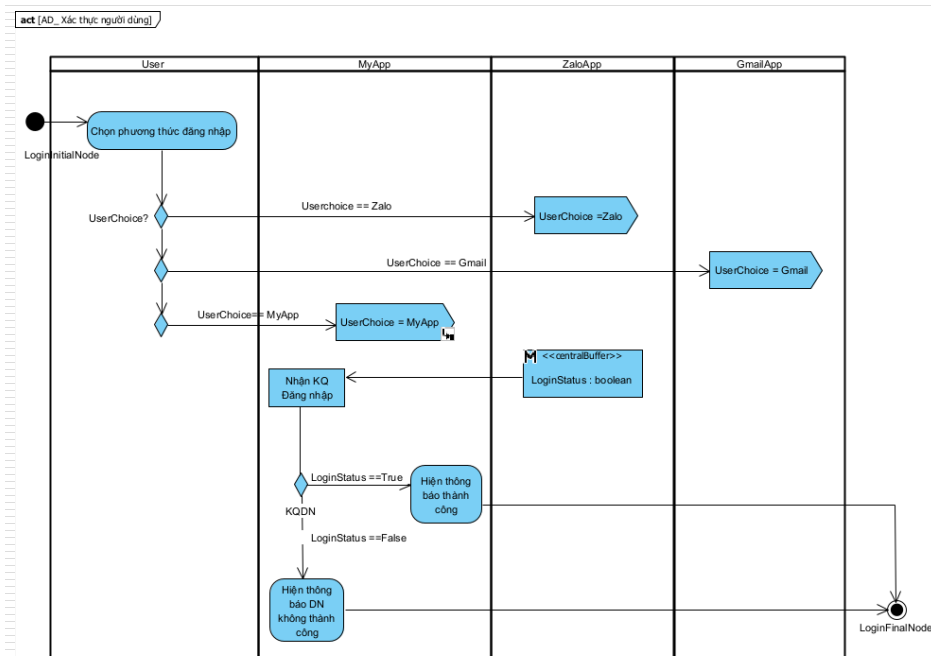
FLOW:

- Trang Đăng nhập có 3 lựa chọn: ĐN = TK Zalo; ĐN = TK Gmail; ĐN = UD MyApp của HT.
- Tùy theo User chọn đăng nhập theo cách nào thì sẽ gọi Activity của luồng con tương ứng
- Sau khi đăng nhập xong trong các luồng con, thì luồng con sẽ trả về LoginStatus cho biết trạng thái đăng nhập
- Luồng chính sẽ sử dụng Login Status để hiện thông báo cho người dùng



91

91



92

92

Kịch bản luồng con:

Kịch bản:

Nếu nhận được tín hiệu UserChoice = MyApp

(1) hệ thống sẽ hiển thị Form Đăng nhập TK bằng MyApp

(2) User nhập TT đăng nhập

(3) Xác nhận đăng nhập

3.1 Nếu User nhấn nút Cancel HT sẽ ghi nhận ĐN không thành công hệ thống sẽ kết thúc quá trình đăng nhập ghi nhận LoginStatus == False và

3.2 Nếu nhấn nút SignIn thì chuyển sang (4)

(4) Kiểm tra so khớp TT đăng nhập

4.1 TK Không hợp lệ : Hiện thông báo lỗi và Hiển thị lại form đăng nhập

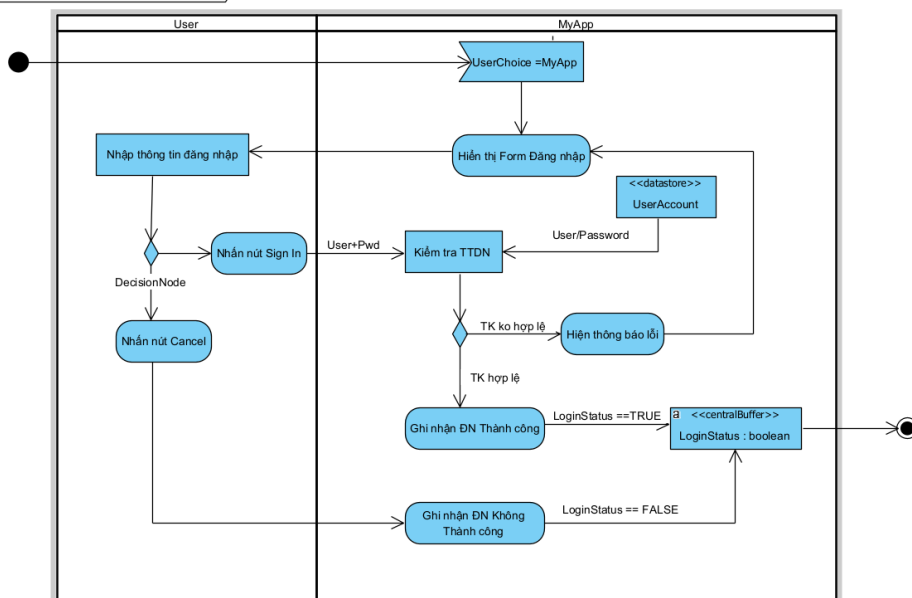
4.2 TK Hợp lệ: Ghi nhận ĐN thành công và ghi nhận Login Status == True



93

93

act [Đăng nhập bằng TK MyApp Activity Diagram]



94

94

3. Bài tập thực hành

Ví dụ 2: (A) MHH nghiệp vụ Hệ thống yêu cầu đào tạo

Đầu bài: Để nâng cấp hệ thống thông tin của mình, một công ty muốn, trong bước đầu, vạch ra một quy trình đào tạo (QTĐT) cho nhân viên của mình, để sau đó tin học hoá một phần quy trình này.

- 1) QTĐT bắt đầu khi có một nhân viên (NV) gửi đến người phụ trách đào tạo (PTĐT) một đề nghị được đi đào tạo. Người PTĐT xem xét đề nghị này và đưa ra trả lời đồng ý hay không đồng ý.
- 2) Nếu đồng ý, người PTĐT tìm trong một catalô một nơi có các lớp đào tạo thích hợp, thông báo nội dung đào tạo lại cho NV đã xin đi đào tạo, cùng với một danh sách các kỳ học sẽ mở tới đây. Khi người NV đã chọn kỳ học, người PTĐT gửi một đăng ký cho NV đó tới cơ sở đào tạo.



3. Bài tập thực hành

(A) MHH nghiệp vụ (2)

Đầu bài (tt)

- 3) Nếu có gì cản trở, người NV phải báo sớm cho người PTĐT để huỷ đăng ký hay huỷ đề nghị đào tạo.
- 4) Sau khi đào tạo xong, người NV nộp lại cho người PTĐT một giấy xác nhận sự có mặt và giấy nhận xét kết quả học tập.
- 5) Người PTĐT kiểm tra lại hoá đơn mà cơ sở đào tạo gửi tới, trước khi chuyển cho kế toán trả tiền.

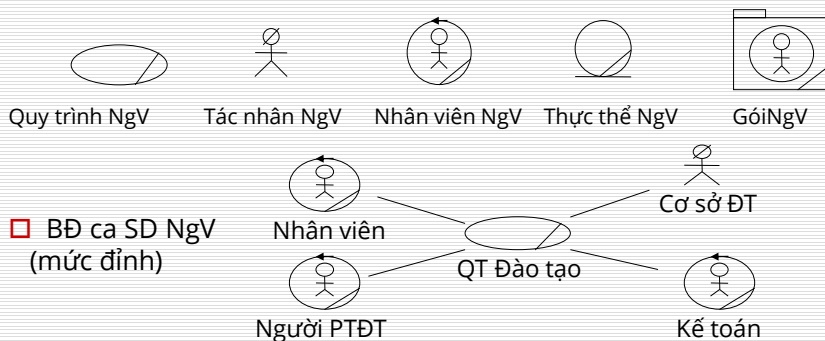


3. Bài tập thực hành

(A) MHH nghiệp vụ (3)

Câu hỏi 1: MHH quy trình đào tạo trên bằng một BD ca sử dụng (mức đỉnh).

□ Để MHH nghiệp vụ (NgV) ta dùng các biểu tượng sau, do Jacobson đề xuất (vẫn được dùng trong RUP và trong Rational Rose 2000):



□ BD ca SD NgV (mức đỉnh)



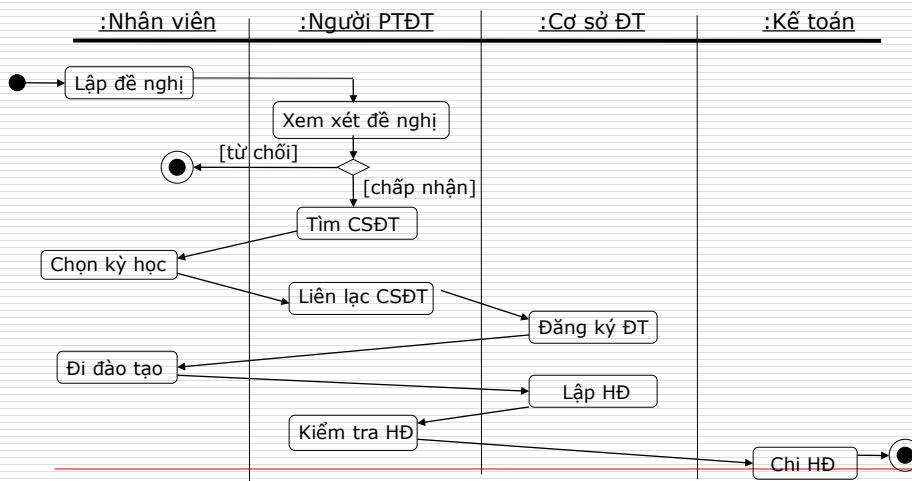
97

97

3. Bài tập thực hành

(A) MHH nghiệp vụ (4)

Câu hỏi 2: Mô tả QTĐT bằng một biểu đồ hoạt động, có phân tuyến.



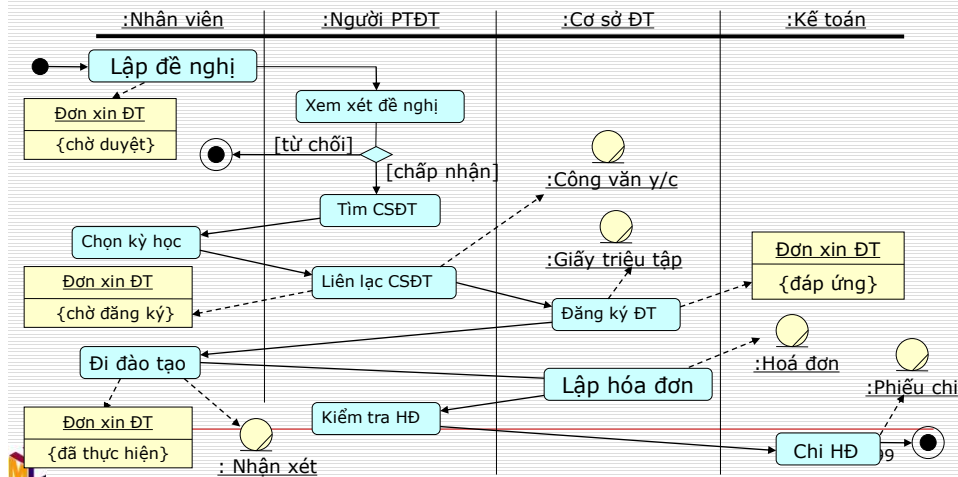
98

98

3. Bài tập thực hành

(A) MHH nghiệp vụ (5)

Câu hỏi 2 (tt): Thêm vào BD hoạt động các đối tượng NgV đã được sản sinh cùng với trạng thái của chúng nếu có.

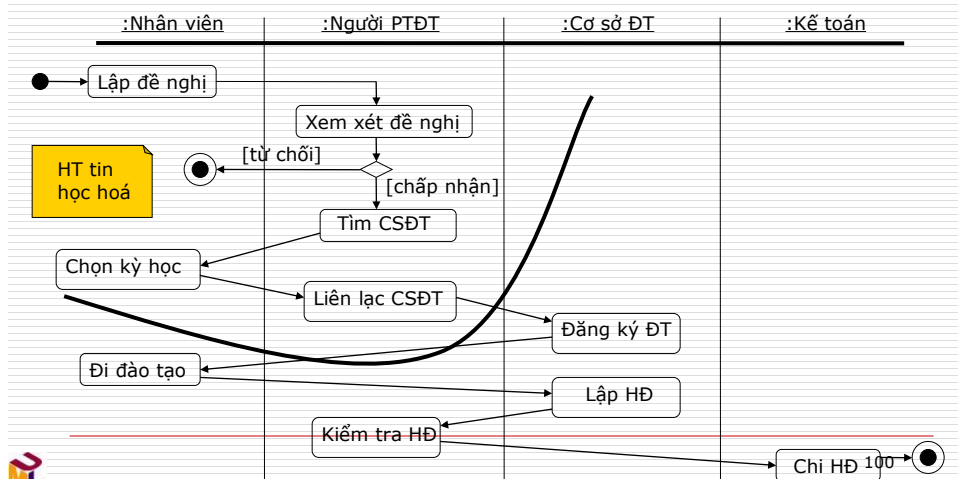


99

3. Bài tập thực hành

(B) XD phạm vi, nhu cầu

Câu hỏi 3: Từ mô hình nghiệp vụ, hãy xác định phạm vi của HT tin học hoá.



100

3. Bài tập thực hành

(B) XD phạm vi, nhu cầu (2)

Câu hỏi 4: *Hãy xác định các nhu cầu của hệ thống tin học hoá (hệ YCĐT)*

- ☐ Hệ YCĐT cho phép bắt đầu một yêu cầu DT và theo dõi nó cho đến khi hoàn tất việc đăng ký DT cho một NV.
- ☐ Hệ YCĐT phải tin học hoá các hoạt động nghiệp vụ sau:
 - Soạn thảo một đề nghị DT (nhân viên).
 - Kiểm nhận một đề nghị (người PTĐT).
 - Tìm nơi DT (người PTĐT).
 - Chọn lớp DT (nhân viên).
 - Đăng ký DT (người PTĐT).
- ☐ Muốn thể HT phải quản lý một catalô, mà người NV có thể đọc, còn người PTĐT có thể viết vào đó và tổ chức nó theo chủ đề.
- ☐ *Lưu ý:* người NV có thể xin huỷ đăng ký hay huỷ y/c DT.



3. Bài tập thực hành

(B) XD phạm vi, nhu cầu (3)

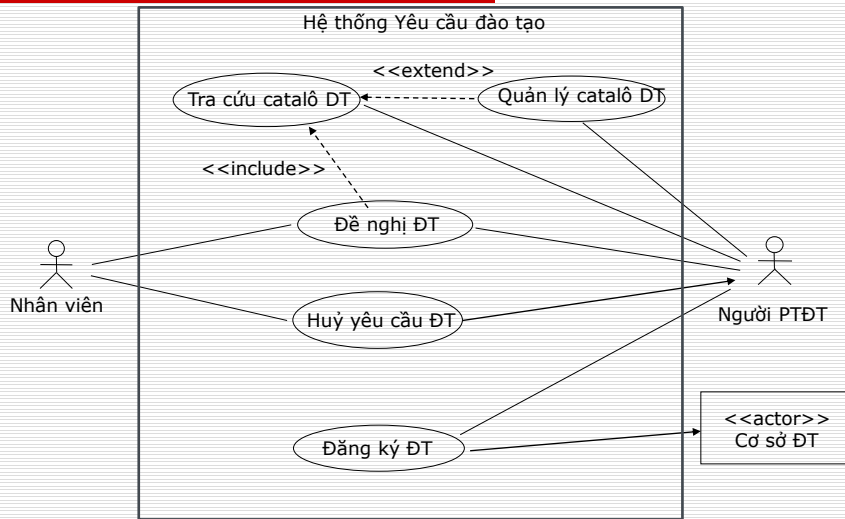
Câu hỏi 5: *Hãy lập BĐ ca SD cho hệ YCĐT*

Từ các hoạt động nghiệp vụ, phát hiện các ca SD sau:

- ☐ Đề nghị được DT
- ☐ Đăng ký DT (tra cứu catalô DT)
- ☐ Huỷ yêu cầu
- ☐ Quản lý catalô DT



3. Bài tập thực hành



103

103

Hỏi - đáp



104

104

Lời hay ý đẹp

**"Kẻ trách mình có thể làm nên điều
hay cho người khác. Kẻ trách người
thật là làm thêm cái dở cho mình"**

HỨA HÀNH

