CHƯƠNG 3
NGÔN NGỮ MÔ HÌNH HÓA THỐNG NHẤT (UML)

GV biên soạn: ThS. Trần Kim Hương

NỘI DUNG

1. Mô hình hóa
2. Lịch sử UML
3. Mục đích UML
4. Khung nhìn UML
5. Phân loại sơ đồ
6. Quy trình RUP

1.1 Mô hình hóa

- Mô hình là một dạng biểu diễn trừu tượng của một hệ thống thực.
- Mô hình cho phép:
  - Cái nhìn trực quan về hệ thống đang có hoặc hướng tới
  - Kiểm chứng hệ thống bởi khách hàng
  - Cung cấp những chỉ dẫn để xây dựng hệ thống
  - Tài liệu hóa hệ thống

→Diễn tả hệ thống bằng mô hình (khi phân tích và thiết kế) được gọi là mô hình hóa.

1.1 Mô hình hóa

- Tồn tại nhiều cách mô hình hóa một hệ thống
  - Mô hình phù hợp sẽ làm cho việc giải bài toán dễ dàng hơn
- Có nhiều mức chính xác của mô hình
  - Mô hình trừu tượng → làm mịn →mô hình chi tiết
- Không có mô hình nào là đầy đủ
  - Cần tiếp cận (hiểu) hệ thống thông qua nhiều mô hình khác nhau
- Mô hình tốt phải là mô hình phù hợp với thế giới thực

1.2 Mô hình hóa hướng đối tượng

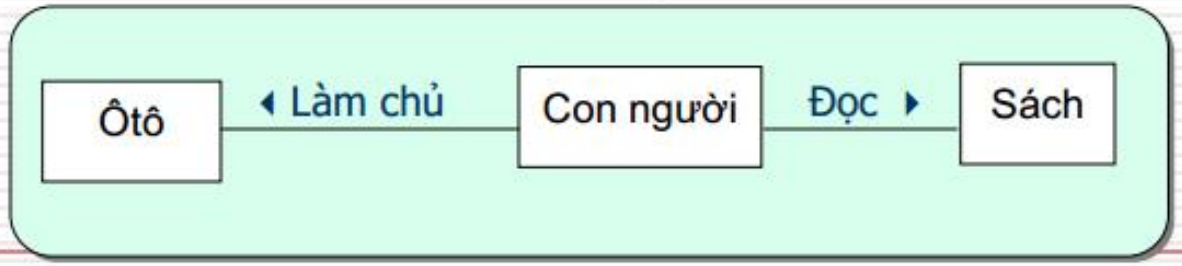
- Tăng tính độc lập của mô hình với các chức năng yêu cầu
- Dễ dàng hơn trong việc thay đổi hoặc thêm bớt các chức năng
- Gần với thế giới thực

1.3 Ví dụ về mô hình

Thế giới thực

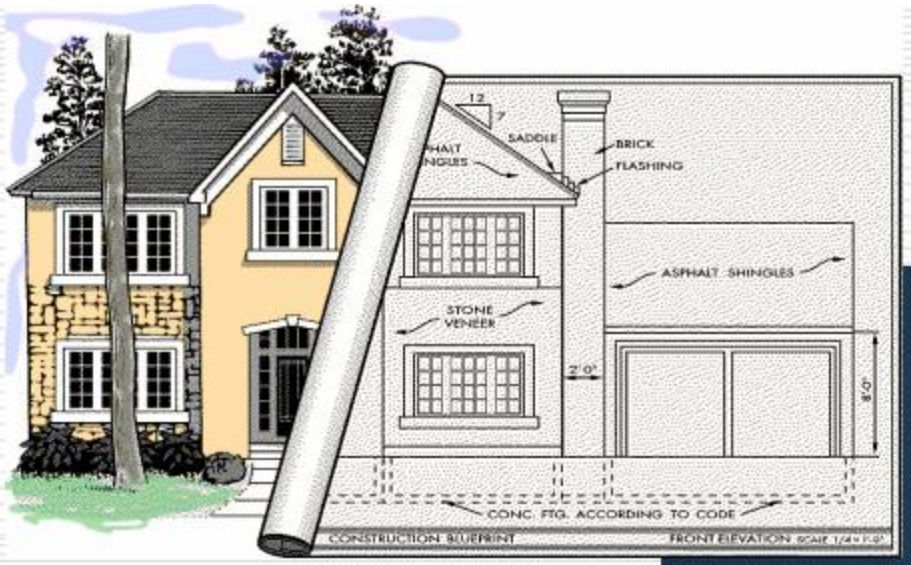
Mô hình: Quả địa cầu học sinh

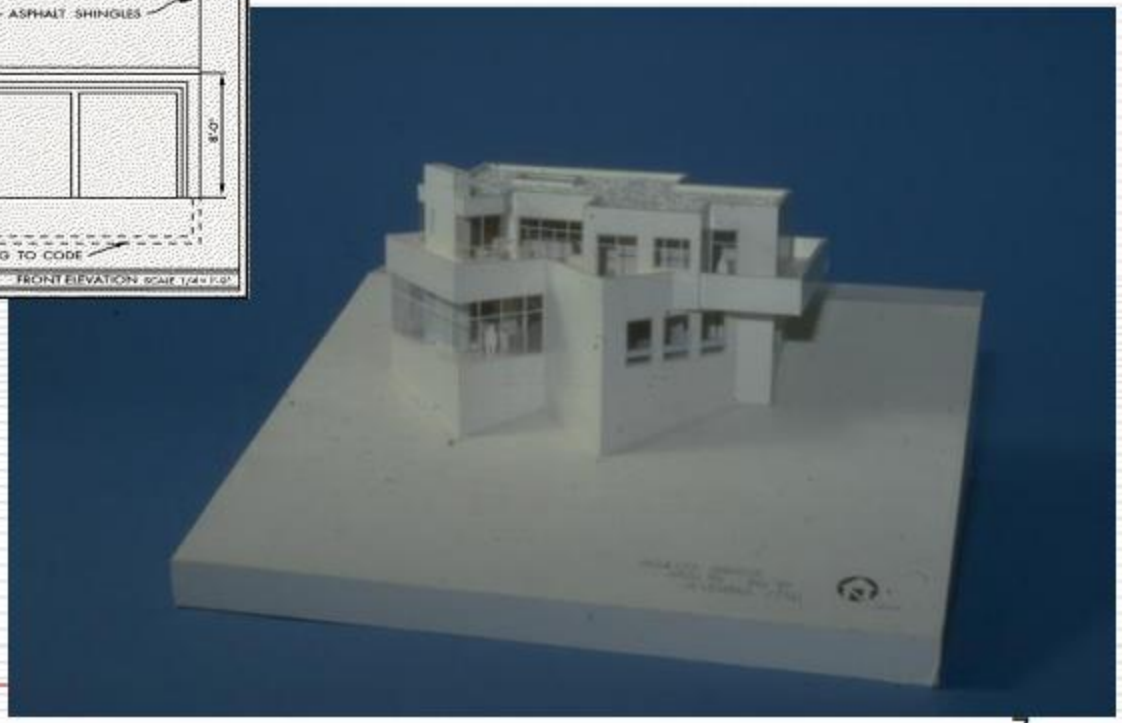
Thế giới thực



Mô hình

1.3 Ví dụ


12
7
SADDLE
BRICK
FLASHING
ASPHALT SHINGLES
STONE VENEER
2'-0"
8'-0"
CONC. FTG. ACCORDING TO CODE
CONSTRUCTION BLUEPRINT
FRONT ELEVATION



UML là gì?

"The Unified Modeling Language is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system"

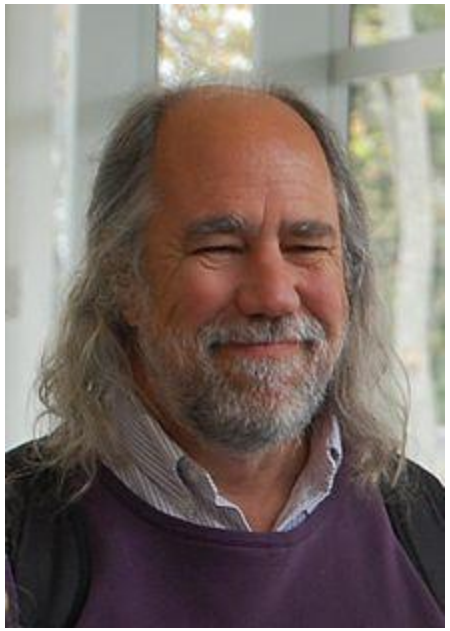
Grady Booch, Ivar Jacobson and James Rumbaugh

UML là gì?

- UML (Unified Modeling Language)
  - Là ngôn ngữ mô hình hóa hướng đối tượng
  - Được thừa nhận như một chuẩn mặc định của ngành CNTT
  - Có nhiều công cụ và phương pháp dựa trên UML
    - Enterprise Architecture / Rational Rose
    - Rational Unified Process (RUP)
    - StarUML
    - Draw.io
    - Astah
    - dbdiagram.io

2. Lịch sử UML

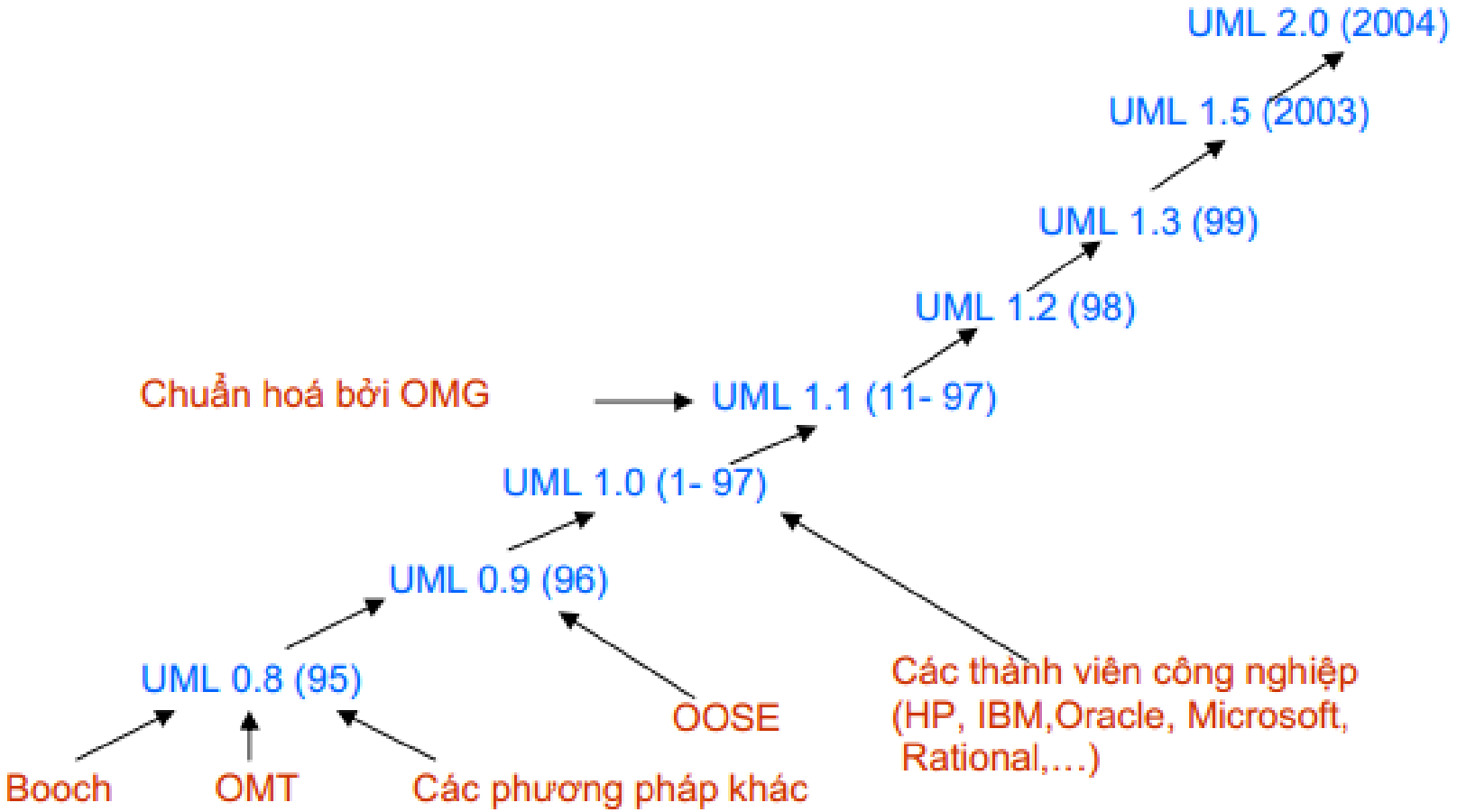
- Grady Booch (born February 27, 1955) is an American software engineer, best known for developing the Unified Modeling Language (UML) with Ivar Jacobson and James Rumbaugh. He is recognized internationally for his innovative work in software architecture, software engineering, and collaborative development environments.



Grady Booch in 2011

https://en.wikipedia.org/wiki/Grady_Booch

2. Lịch sử UML



https://en.wikipedia.org/wiki/Unified_Modeling_Language

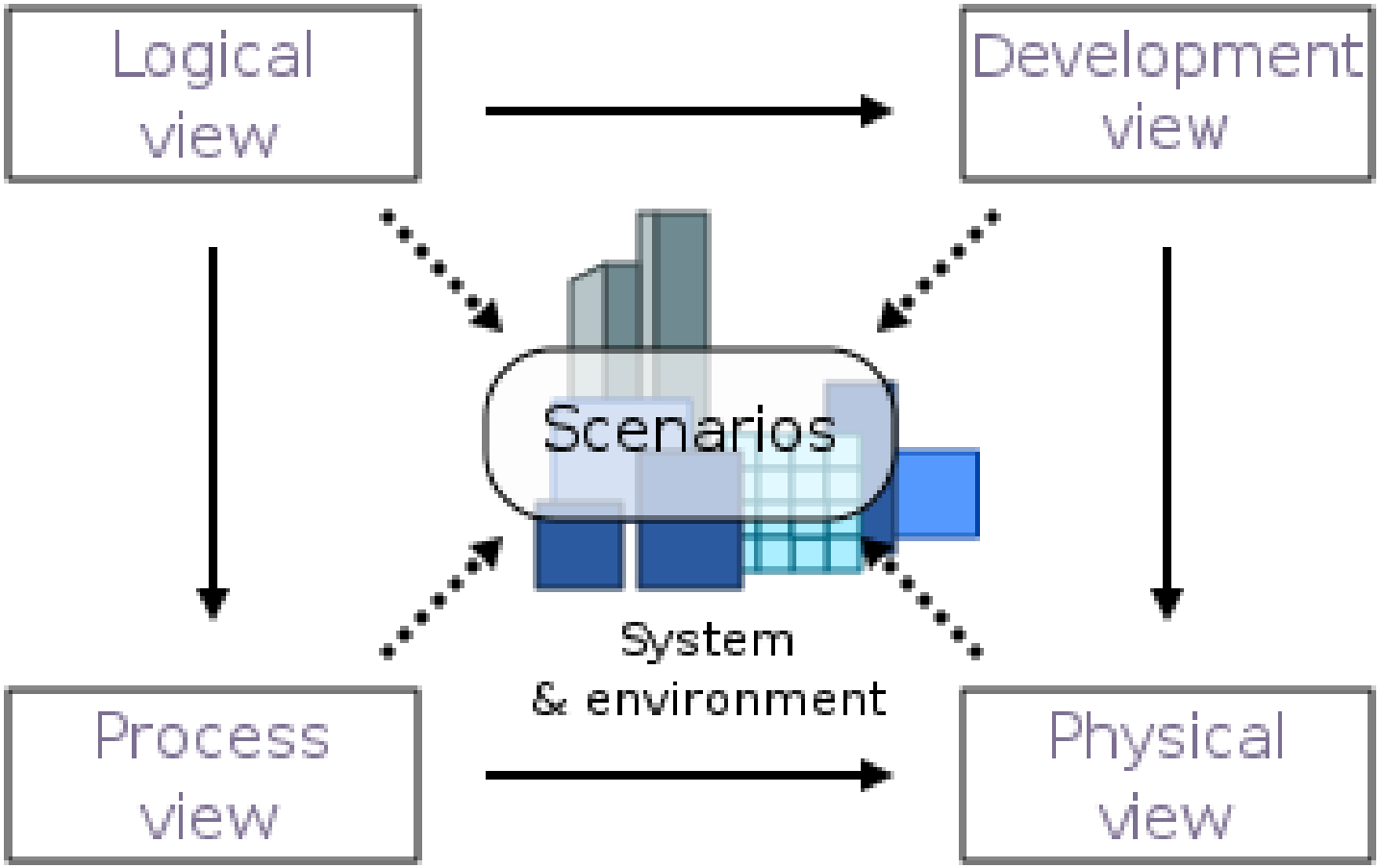
3. Mục đích của UML

- Để đặc tả phần mềm hướng đối tượng
- Phát triển phần mềm hướng đối tượng
- Các tài liệu đính kèm trong phát triển phần mềm hướng đối tượng.
- UML giúp người phát triển hiểu rõ và ra quyết định liên quan đến phần mềm cần xây dựng.
- UML bao gồm một tập các khái niệm, các ký hiệu, các biểu đồ và hướng dẫn.
- UML hỗ trợ xây dựng hệ thống hướng đối tượng dựa trên việc nắm bắt *khía cạnh cấu trúc tĩnh* và các *hành vi động* của hệ thống.

Xây dựng

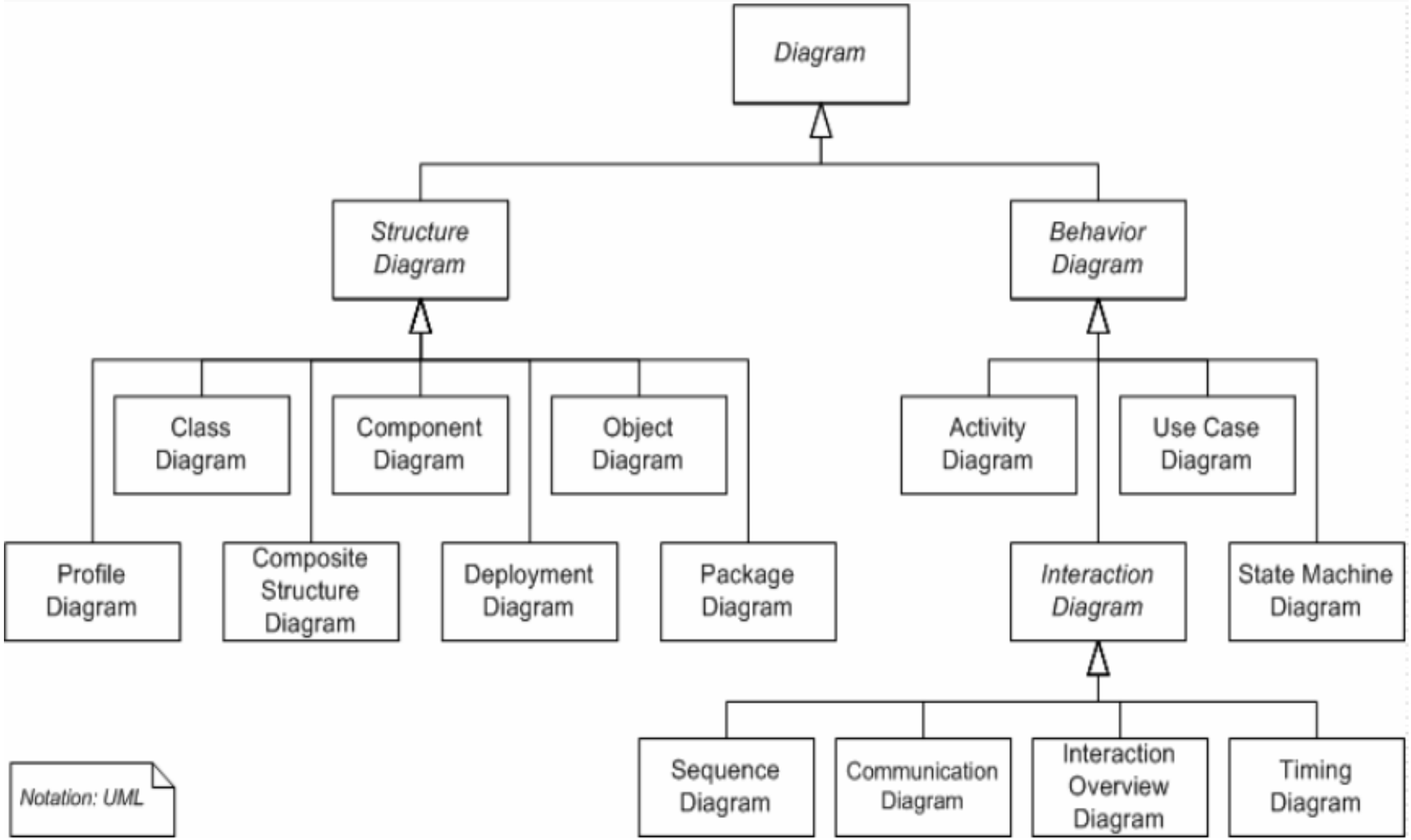
- Các mô hình UML có thể ánh xạ sang
  - Một ngôn ngữ lập trình: Java, C++, C#,…
  - Một bảng trong CSDL quan hệ (R-DBMS)
  - Một lưu trữ bền vững của CSDL HĐT (OO-DBMS)
- Forward engineering

4. Khung nhìn UML



https://en.wikipedia.org/wiki/4%2B1_architectural_view_model

5. Phân loại sơ đồ UML



5. Phân loại sơ đồ UML

- Dựa vào tính chất của các sơ đồ, UML chia các sơ đồ thành hai lớp mô hình:
  - Sơ đồ mô hình hóa hành vi (Behavioral Modeling Diagrams).
  - Sơ đồ mô hình hóa cấu trúc (Structural Modeling Diagrams).

https://en.wikipedia.org/wiki/4%2B1_architectural_view_model

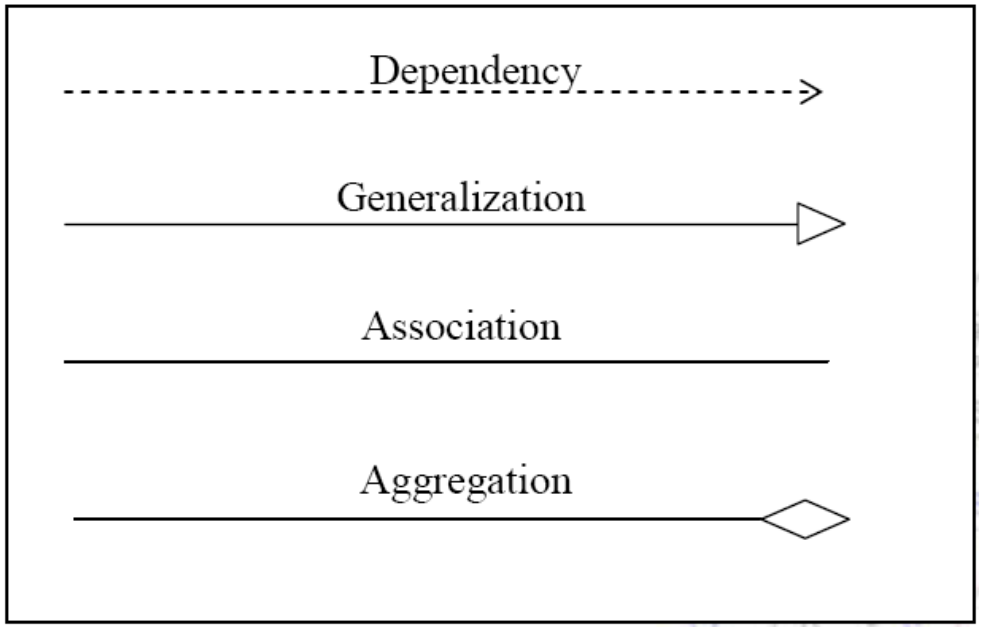
5. Phân loại sơ đồ UML

- Sơ đồ mô hình hóa hành vi (Behavioral Modeling Diagrams):
  - Nắm bắt các hoạt động và hành vi của hệ thống, sự tương tác giữa các phần tử bên trong và bên ngoài hệ thống.
  - Behaviour Diagram thường dùng:
    - Use Case Diagram
    - Activity Diagram
    - Sequence Diagram
    - State machine Diagram

5. Phân loại sơ đồ UML

- Sơ đồ mô hình hóa cấu trúc (Structural Modeling Diagrams).
  - Biểu diễn các cấu trúc tĩnh của hệ thống phần mềm được mô hình hoá.
  - Structure Diagram thường dùng:
    - Class Diagram
    - Component Diagram
    - Package Diagram
    - Deployment Diagram

5. Phân loại sơ đồ UML



Quan hệ phụ thuộc

Quan hệ tổng quát

Quan hệ liên kết

Quan hệ kết tập

Hình 1: Một số dạng quan hệ trong UML

5.1 Sơ đồ Use-Case

a) Ý nghĩa

- Sơ đồ Use-case:

  + Biểu diễn sơ đồ chức năng của hệ thống.

  + Tương tác giữa các *tác nhân* và hệ thống thông qua các use-case.

- Trong đó:

  + Mỗi *use-case* mô tả một chức năng hệ thống

  + *Tác nhân* là con người hay hệ thống thực khác cung cấp thông tin hay tác động tới hệ thống.

5.1 Sơ đồ Use-Case

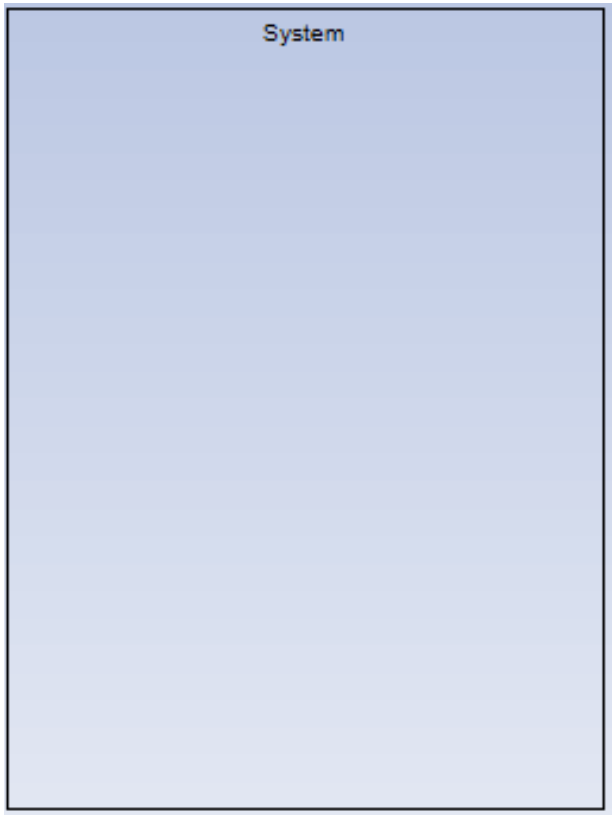
b) Tập ký hiệu

Một sơ đồ Use-case chứa các phần tử mô hình biểu thị hệ thống, tác nhân cũng như các trường hợp sử dụng và các mối quan hệ giữa các Use-case.

5.1 Sơ đồ Use-Case

b) Tập ký hiệu

- Hệ thống:

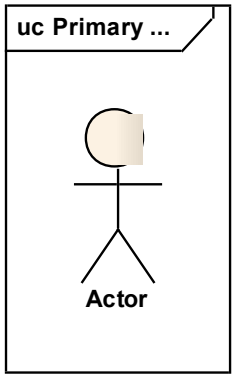


- Hệ thống: ranh giới giữa bên trong và bên ngoài của một chủ thể trong phần mềm chúng ta đang xây dựng.

- Hệ thống không nhất thiết là phần mềm: có thể là một chiếc máy, một doanh nghiệp, trường ĐH,…

5.1 Sơ đồ Use Case

- *Actor*



+ Là người dùng của hệ thống.

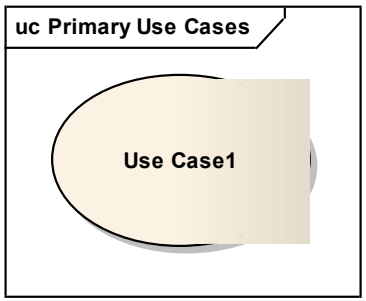
+ Có thể là một người dùng thực hoặc các hệ thống máy tính khác.

+ Thực hiện các use-case

+ Một tác nhân có thể thực hiện nhiều use-case và ngược lại một use-case cũng có thể được thực hiện bởi nhiều tác nhân.

5.1 Sơ đồ Use-Case

- *Use-Case*



+ Là thành phần cơ bản của biểu đồ use-case.

+ Use case được biểu diễn bởi các hình elip.

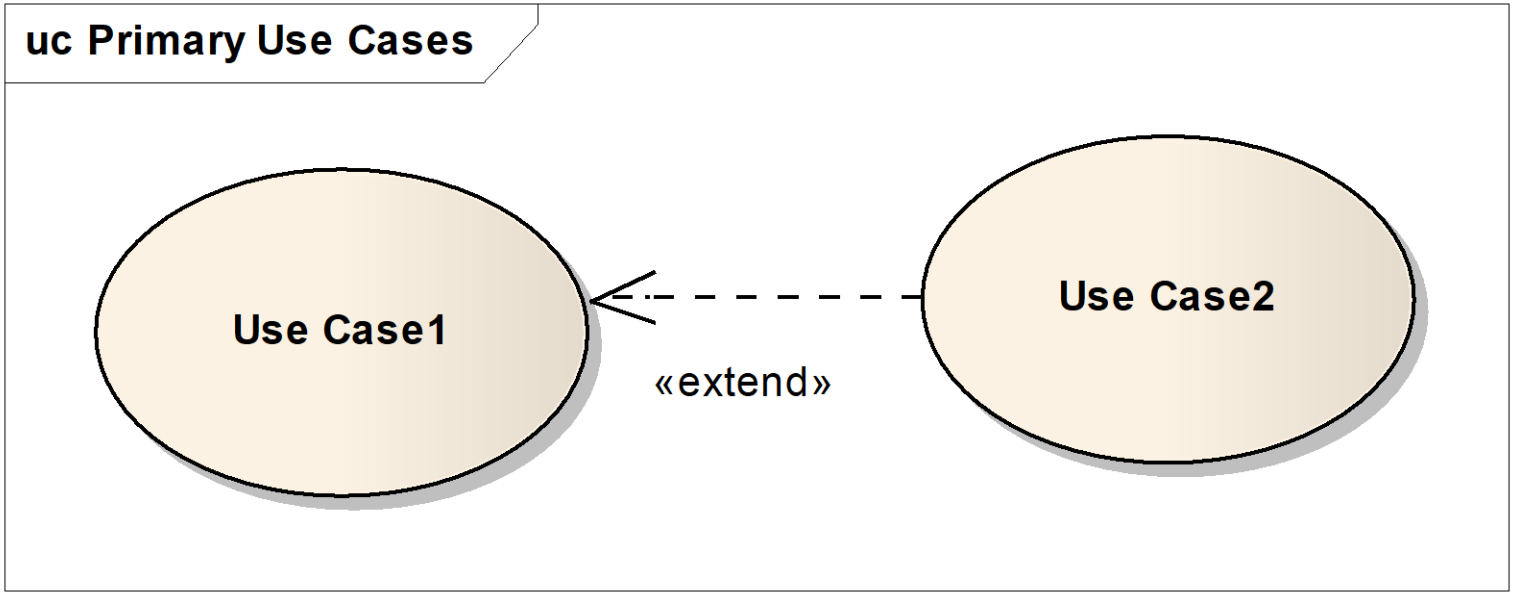
+ Tên các use-case thể hiện một chức năng xác định của hệ thống.

+ Một use-case cũng có thể được thực hiện bởi nhiều tác nhân.

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa các Use-case*

- Extend: (Mở rộng)

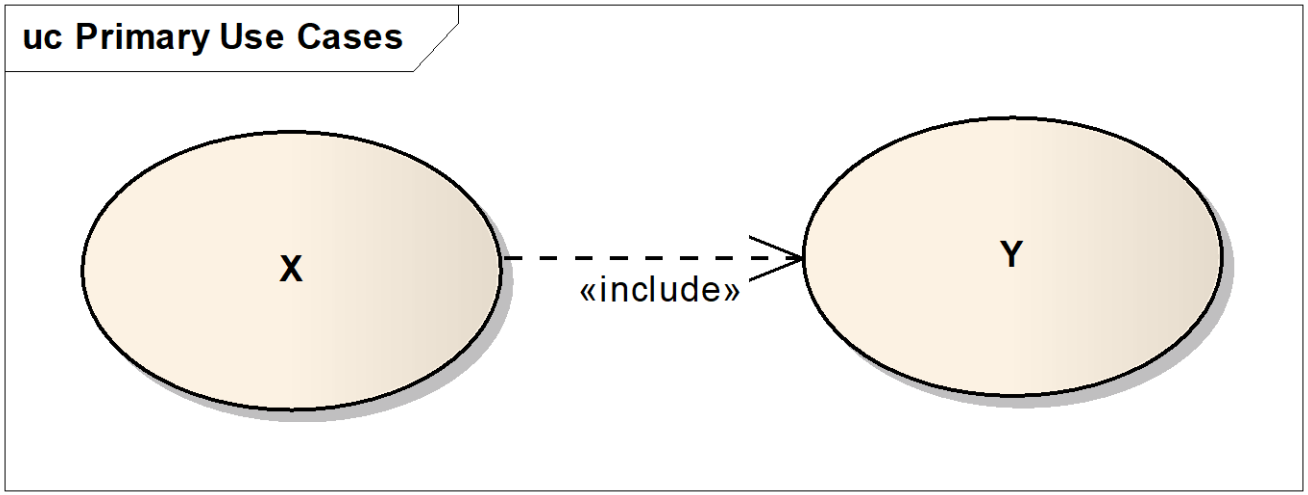


+ Use case này mở rộng từ use case kia bằng cách thêm vào một chức năng cụ thể.

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa các Use-case*

- include: (Bao hàm)



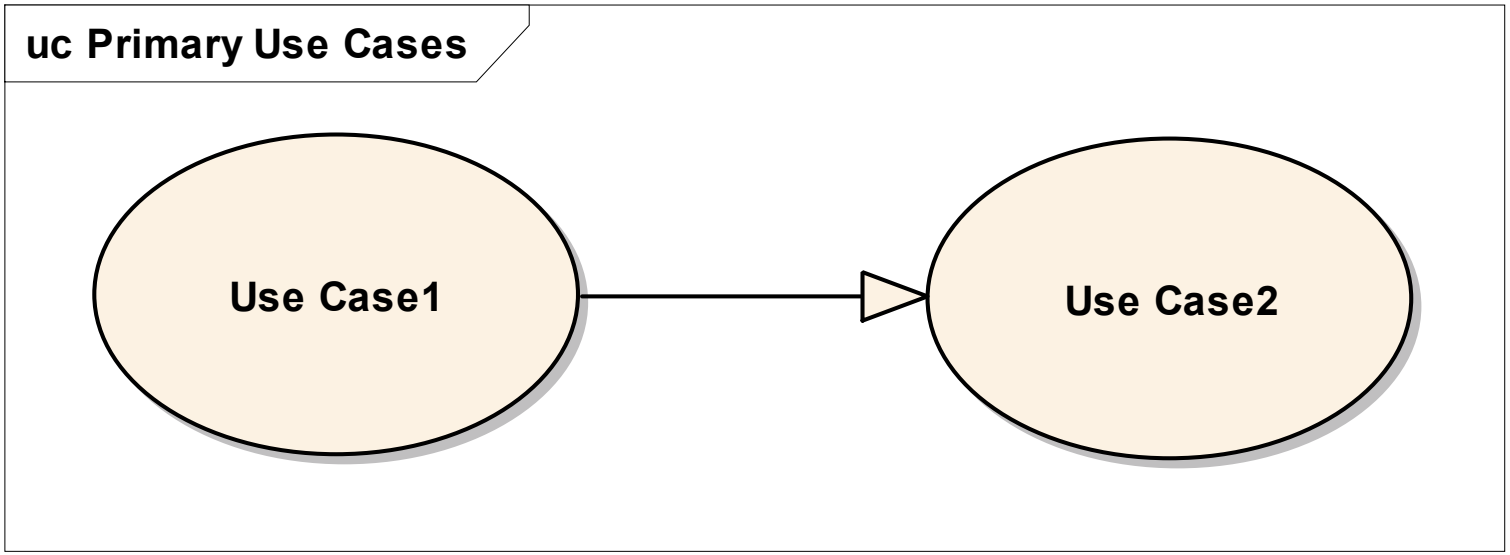
+ X<<include >>Y chỉ ra rằng tiến trình thực hiện X luôn luôn liên quan đến việc thực hiện Y ít nhất một lần.

+ X phải đáp ứng các điều kiện tiền của Y trước khi bao hàm nó.

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa các Use-case*

- Generalization: (Khái quát)

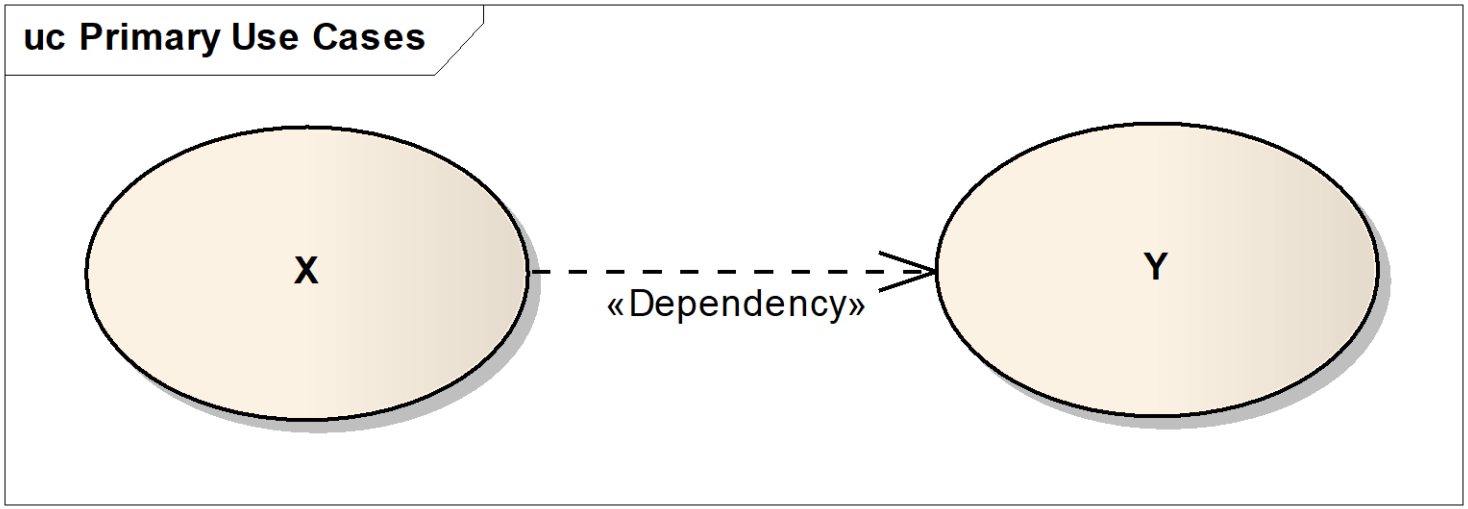


+ Use case này được kế thừa các chức năng từ use case kia

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa các Use-case*

- Dependency: (phụ thuộc)

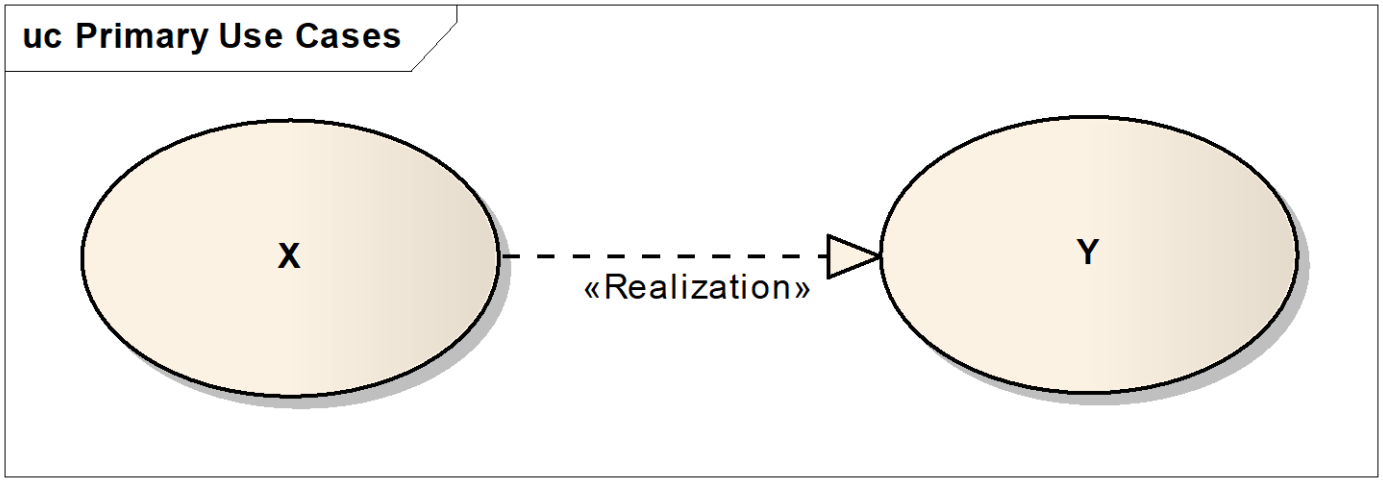


+ Phụ thuộc là mối quan hệ có nghĩa rằng một hoặc một tập mô hình các phần tử yêu cầu mô hình các phần tử khác cho các đặc tả hoặc thực thi của chúng.

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa các use case*

- Realization: (thực thi)



+ Thực thi là mối quan hệ trừu tượng chuyên biệt giữa 2 tập mô hình các phần tử: một là đại diện cho đặc điểm kỹ thuật (Nhà cung cấp) và hai là đại diện cho việc thực thi sau này (Khách hàng).

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa các Actor*

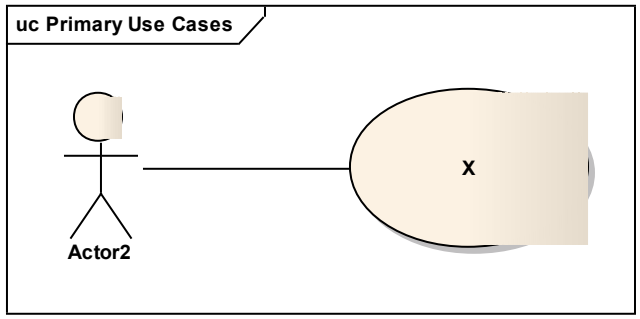
- Generalization: (Khái quát)

Khái quát là một mối quan hệ phân loại giữa phân loại tổng quát và một phân loại chi tiết. Mỗi thể hiện của phân loại chi tiết cũng là một thể hiện gián tiếp của phân loại tổng quát. Vì vậy, phân loại chi tiết kế thừa các tính năng của phân loại tổng quát.

5.1 Sơ đồ Use Case

- *Mối quan hệ giữa Actor và use case*

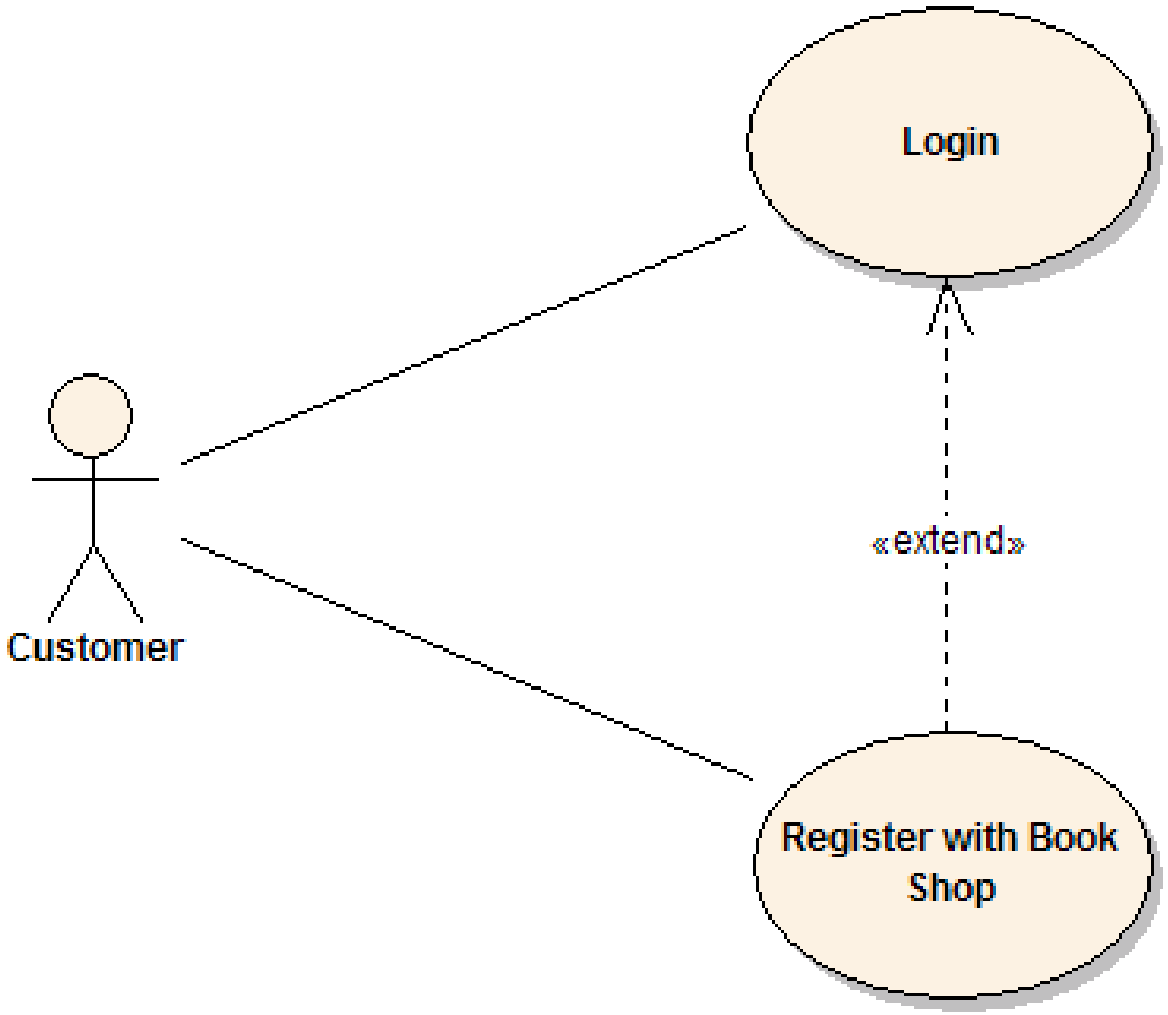
- Association: (Liên kết)



Liên kết đặc tả một mối quan hệ ngữ nghĩa mà nó có thể xảy ra ở các dạng thể hiện. Nó có ít nhất hai đầu cuối đại diện bởi các thuộc tính, mỗi trong số đó được liên kết với dạng của kết thúc.

5.1 Sơ đồ Use Case

c) Ví dụ:



Bài tập UseCase

- Hãy vẽ sơ đồ UseCase cho hệ thống sinhvien.dthu.edu.vn

Bài tập UseCase

- Hoạt động mượn trả sách thư viện trường gồm các hoạt động như sau:
  - Thủ thư thực hiện quy trình mượn sách, trả sách, thống kê sách mượn nhiều, trễ hạn, sách hỏng
  - Độc giả có thể tìm kiếm sách
  - Ban Giám Đốc có thể cập nhật quy định mượn trả sách

5.2 Sơ đồ Lớp (Class diagram)

a) Ý nghĩa

- Lớp: *một nhóm đối tượng có chung một số thuộc tính và phương thức.*

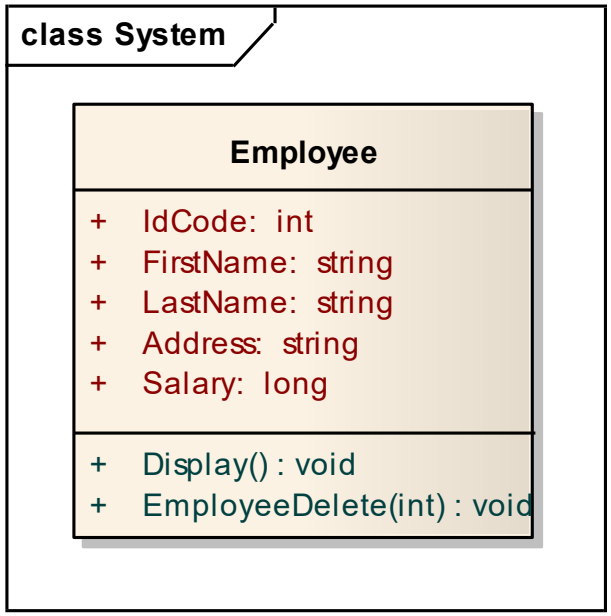
- *Các lớp (bao gồm cả các thuộc tính và phương thức) cùng với các mối quan hệ sẽ tạo thành sơ đồ lớp.*

- Sơ đồ lớp là một sơ đồ dạng mô hình tĩnh nhằm mô tả các khái niệm lớp, các thuộc tính, phương thức và mối quan hệ giữa chúng với nhau.

5.2 Sơ đồ Lớp

b) Ký hiệu

Ký hiệu lớp:



-Trong UML, mỗi lớp được biểu diễn bởi hình chữ nhật gồm 3 phần:

+Tên lớp

+Các thuộc tính: phạm vi truy cập của thuộc tính, tên thuộc tính, giá trị thuộc tính

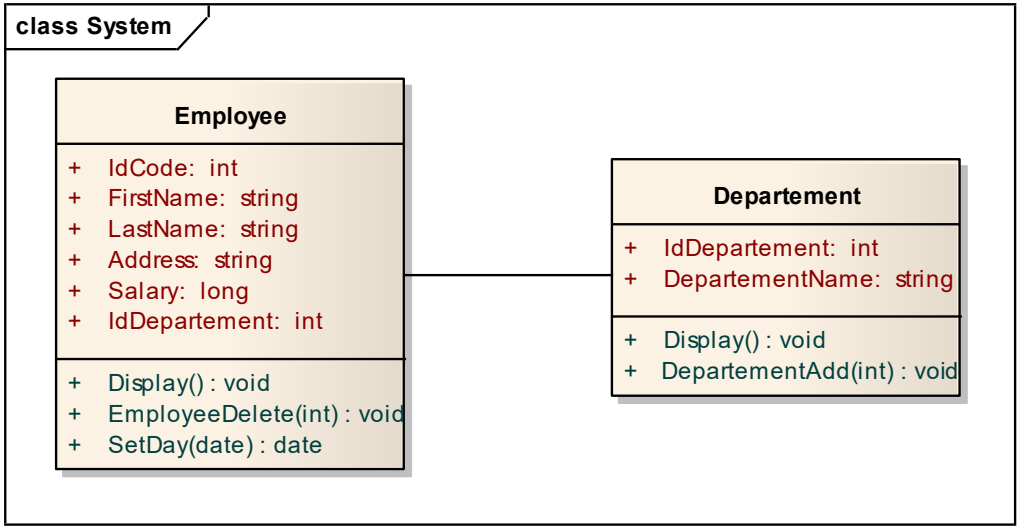
+Các phương thức.

5.2 Sơ đồ Lớp

Các mối quan hệ trong sơ đồ lớp:

-Quan hệ liên kết (Association):

+ Là một sự nối kết giữa các lớp, cũng có nghĩa là sự nối kết giữa các đối tượng của các lớp này.



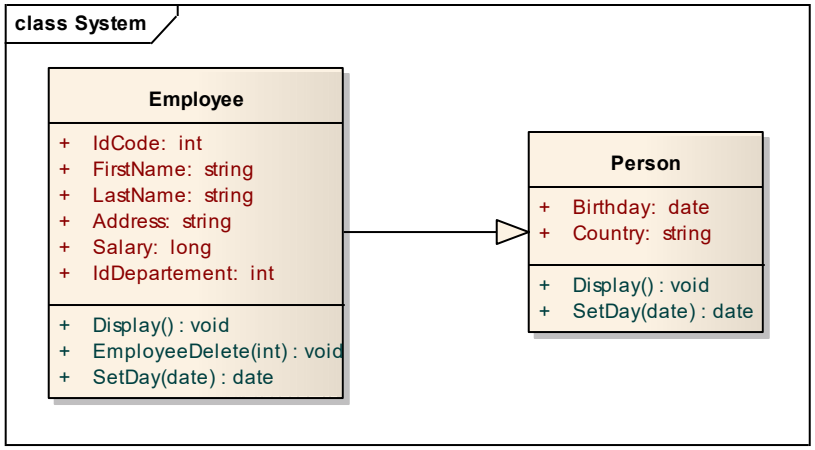
5.2 Sơ đồ Lớp

Các mối quan hệ trong sơ đồ lớp(tt)

- Quan hệ khái quát (Generalization):

+ Là mối quan hệ giữa một lớp có các đặc trưng mang tính khái quát cao hơn và một lớp có tính chất đặc biệt hơn.

+ Trong sơ đồ lớp, mối quan hệ khái quát chính là sự kế thừa của một lớp từ lớp khác.



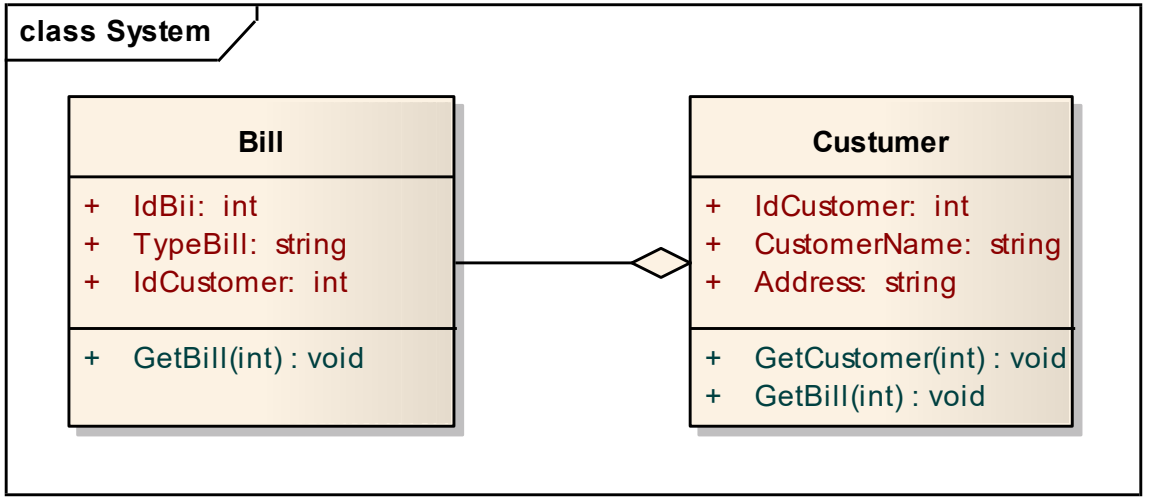
5.2 Sơ đồ Lớp

Các mối quan hệ trong sơ đồ lớp(tt)

- Quan hệ kết tập (Aggregation):

+ Là dạng quan hệ mô tả một lớp A là một phần của lớp B và lớp A có thể tồn tại độc lập.

+ Quan hệ kết tập được biểu diễn bằng một mũi tên gắn hình thoi rỗng ở đầu hướng về lớp bao hàm.



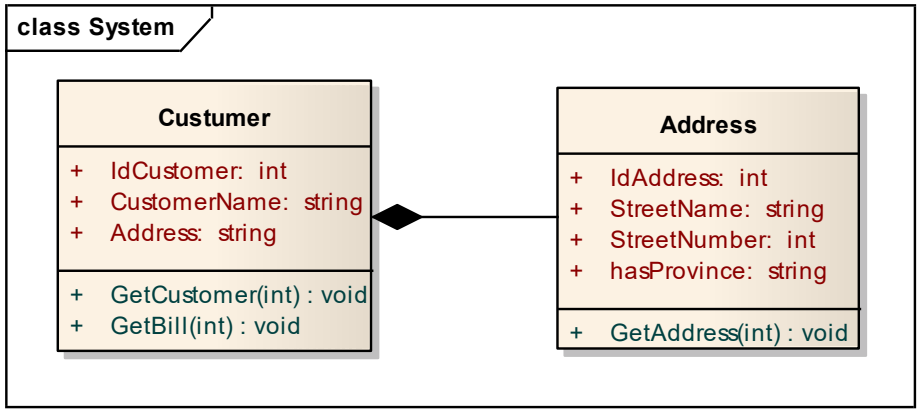
5.2 Sơ đồ Lớp

Các mối quan hệ trong sơ đồ lớp(tt)

- Quan hệ hợp thành (Composition):

+ Một quan hệ hợp thành biểu diễn một quan hệ kiểu tổng thể-bộ phận.

+ Lớp A có quan hệ hợp thành với lớp B nếu lớp A là một phần của lớp B và sự tồn tại của đối tượng lớp B điều khiển sự tồn tại của đối tượng lớp A.



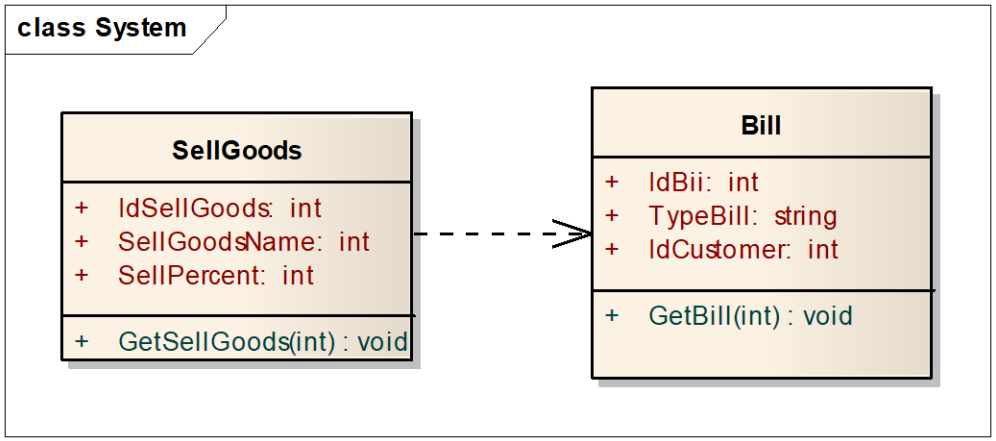
5.2 Sơ đồ Lớp

Các mối quan hệ trong sơ đồ lớp:

- Quan hệ phụ thuộc (Dependency):

+ Phụ thuộc là mối quan hệ giữa hai lớp đối tượng.

+ Một lớp đối tượng A có tính độc lập và một lớp đối tượng B phụ thuộc vào A; một sự thay đổi của A sẽ ảnh hưởng đến lớp phụ thuộc B.



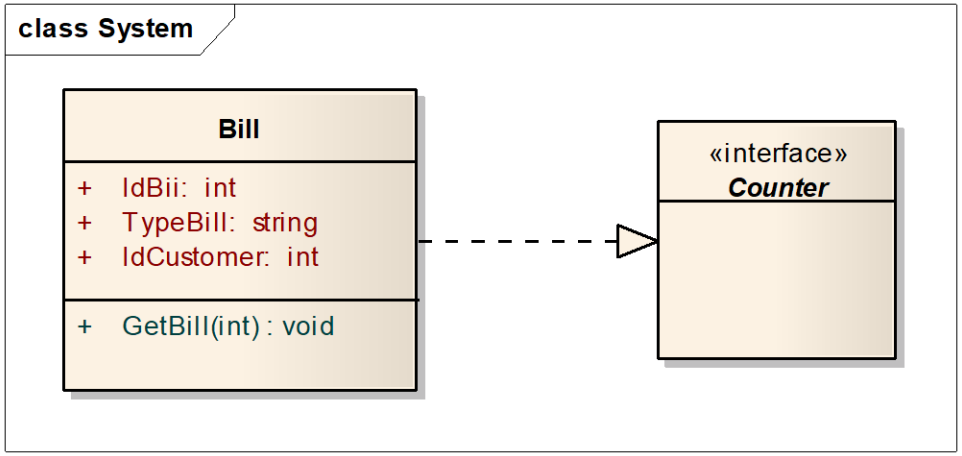
5.2 Sơ đồ Lớp

Các mối quan hệ trong sơ đồ lớp:

- Quan hệ thực thi (Realization):

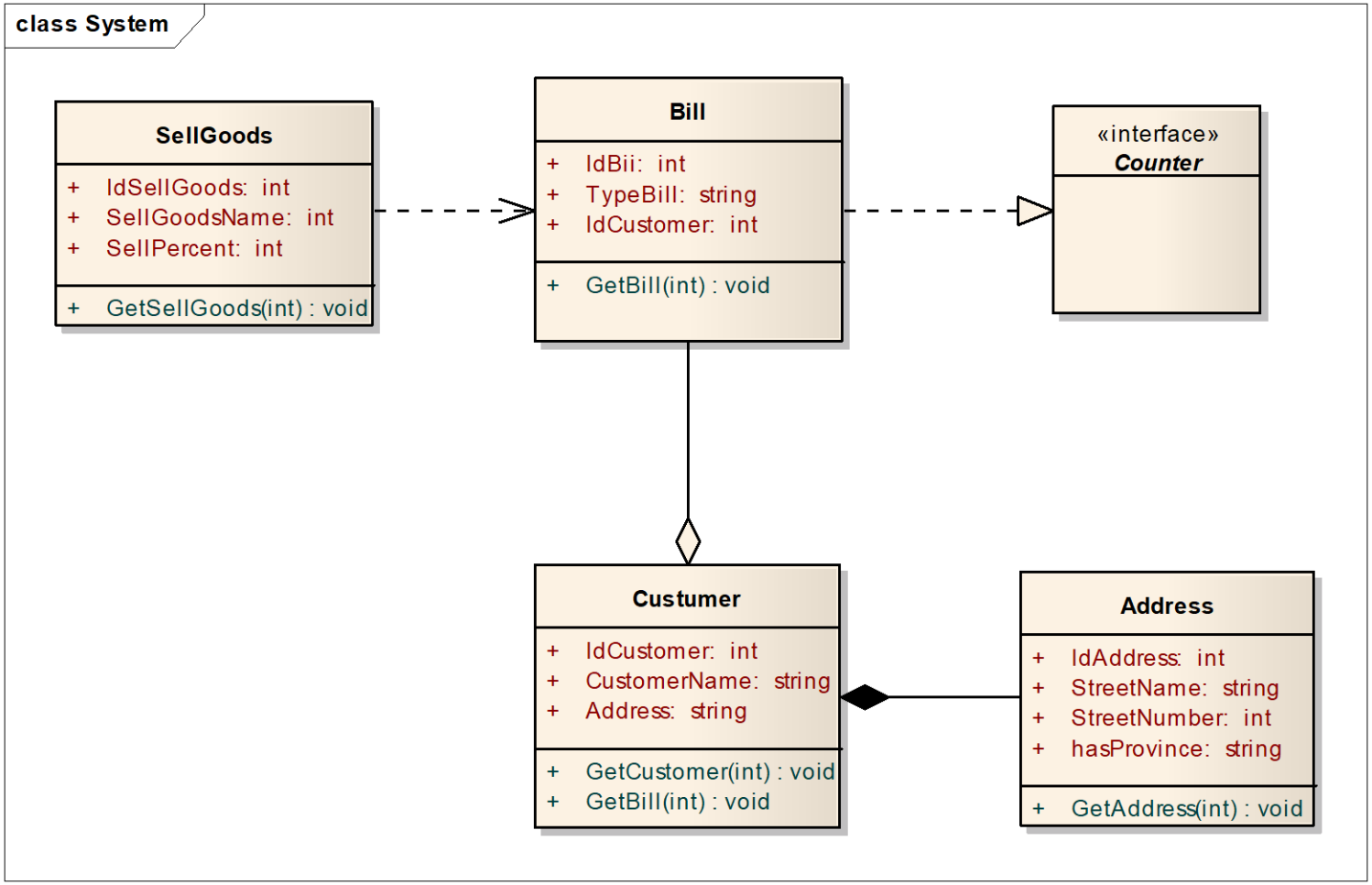
+ Biểu diễn mối quan hệ ngữ nghĩa giữa các thành phần của biểu đồ lớp.

+ Một thành phần mô tả một công việc dạng hợp đồng và thành phần còn lại thực hiện hợp đồng đó.



5.2 Sơ đồ Lớp

c) Ví dụ



Bài tập

- Thư viện cần quản lý các dữ liệu sau:
  - Quyển Sách gồm các thông tin: Mã quyển sách, tình trạng sách, ghi chú. Mỗi quyển sách sẽ thuộc 1 đầu sách
  - Đầu sách gồm mã đầu sách, tựa sách, kích thước, trị giá, tên tác giả. Mỗi đầu sách thuộc 1 lĩnh vực cụ thể.
  - Lĩnh vực gồm các thông tin mã lĩnh vực, tên lĩnh vực, mô tả
  - Sinh viên cần có thẻ thư viện khi mượn sách, thông tin thẻ gồm: mã thẻ, họ tên sv,lớp, niên khóa, ngày lập thẻ, thời hạn thẻ
  - Mượn sách thông qua phiếu mượn gồm: mã phiếu, ngày mượn, ngày trả
  - Mỗi thẻ thư viện sẽ được mượn 3 quyển cho 1 lần mượn.
  - Thư viện cần quản lý thủ thư nào đã cho sv mượn sách, thông tin thủ thư gồm: mã số, họ tên, ngày sinh, số điện thoại, ngày vào làm

5.3 Sơ đồ Trạng thái (State Diagram)

a) Ý nghĩa

- Biểu diễn các trạng thái và sự chuyển tiếp giữa các trạng thái của các đối tượng trong một lớp xác định.

- Thông thường, mỗi lớp sẽ có một biểu đồ trạng thái (trừ lớp trừu tượng là lớp không có đối tượng)

5.3 Sơ đồ Trạng thái

Có hai dạng sơ đồ trạng thái:

- *Sơ đồ trạng thái cho một use case:* mô tả các trạng thái và chuyển tiếp trạng thái của một đối tượng thuộc một lớp nào đó trong hoạt động của một use case cụ thể

- *Sơ đồ trạng thái hệ thống:* mô tả tất cả các trạng thái của một đối tượng trong toàn bộ hoạt động của cả hệ thống

5.3 Sơ đồ Trạng thái

- *Trạng thái (state):* miêu tả các biến trạng thái hoặc các hành động (action) tương ứng với trạng thái đó.

- *Trạng thái khởi đầu (initial state):* trạng thái đầu tiên khi kích hoạt đối tượng.

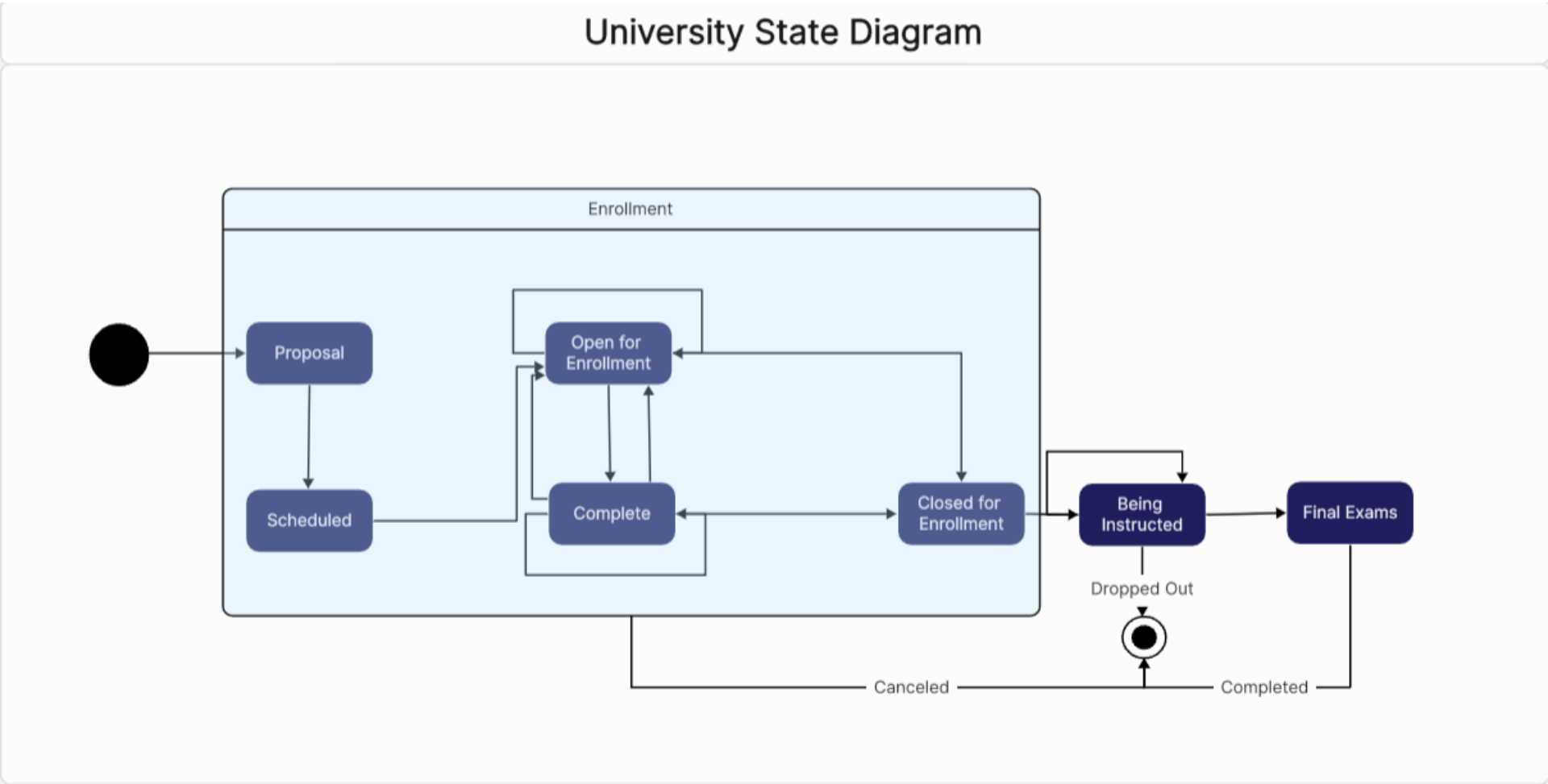
- *Trạng thái kết thúc (final state):* kết thúc vòng đời đối tượng.

- *Các chuyển tiếp (transition):* biểu diễn các chuyển đổi giữa các trạng thái.

- *Sự kiện (event):* sự kiện tác động gây ra sự chuyển đổi trạng thái.

5.3 Sơ đồ Trạng thái

c) Ví dụ:



Bài tập

- Vẽ **sơ đồ trạng thái cho Phiếu mượn sách** trong hệ thống thư viện
  - Các trạng thái chính:
    - Sẵn sàng (Available): Phiếu mượn đã được tạo
    - Đã mượn (Borrowed): phiếu mượn ở trạng thái đang sử dụng.
    - Đang gia hạn (Renewing): Phiếu mượn đang gia hạn
    - Trả sách (Returned): phiếu mượn hoàn tất.
    - Quá hạn (Overdue): Phiếu mượn bị quá hạn
  - Các sự kiện chuyển trạng thái:
    - Mượn sách (Borrow): Sự kiện làm phiếu mượn chuyển từ trạng thái Sẵn sàng sang Đã mượn.
    - Gia hạn (Renew): Sự kiện làm phiếu mượn chuyển từ Đã mượn sang Đang gia hạn.
    - Trả sách (Return): Sự kiện làm phiếu mượn chuyển từ Đang gia hạn sang Trả sách.
    - Quá hạn (Expire): Sự kiện làm phiếu mượn chuyển từ Đã mượn sang Quá hạn.

5.4 Sơ đồ Hoạt động (Activity)

a) Ý nghĩa

- Sơ đồ hoạt động:

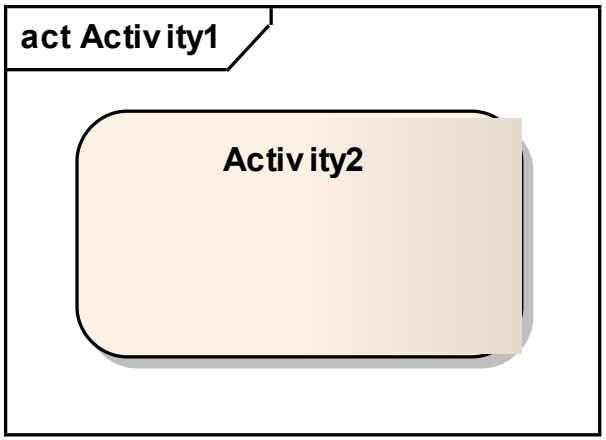
+ Biểu diễn các hoạt động và sự đồng bộ, chuyển tiếp các hoạt động của hệ thống trong một lớp,

+ Kết hợp giữa các lớp với nhau trong một chức năng cụ thể.

5.4 Sơ đồ Hoạt động

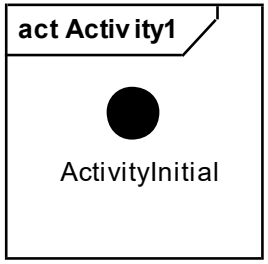
b) Ký hiệu

- *Hoạt động* (*Activity*): là một quy trình được định nghĩa rõ ràng, có thể được thực hiện bởi một hàm hoặc một nhóm đối tượng. Hoạt động được thể hiện bằng hình chữ nhật tròn cạnh.

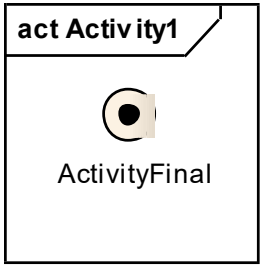


5.4 Sơ đồ Hoạt động

- *Trạng thái khởi đầu (Intial):* là nút điều khiển cho phép bắt đầu khi một hoạt động được gọi.

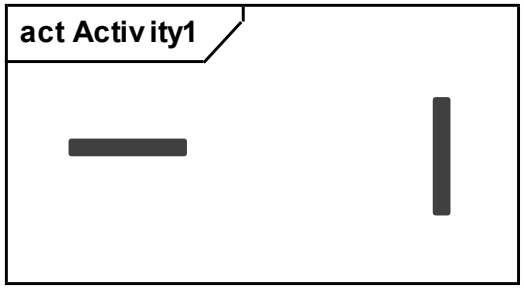


- Trạng thái kết thúc (Final Intial): Một hoạt động có thể có nhiều hơn một trạng thái kết thúc. Trạng thái kết thúc đầu tiên dừng lại tất cả các luồng hoạt động.

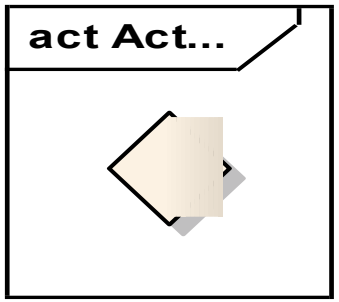


5.4 Sơ đồ Hoạt động

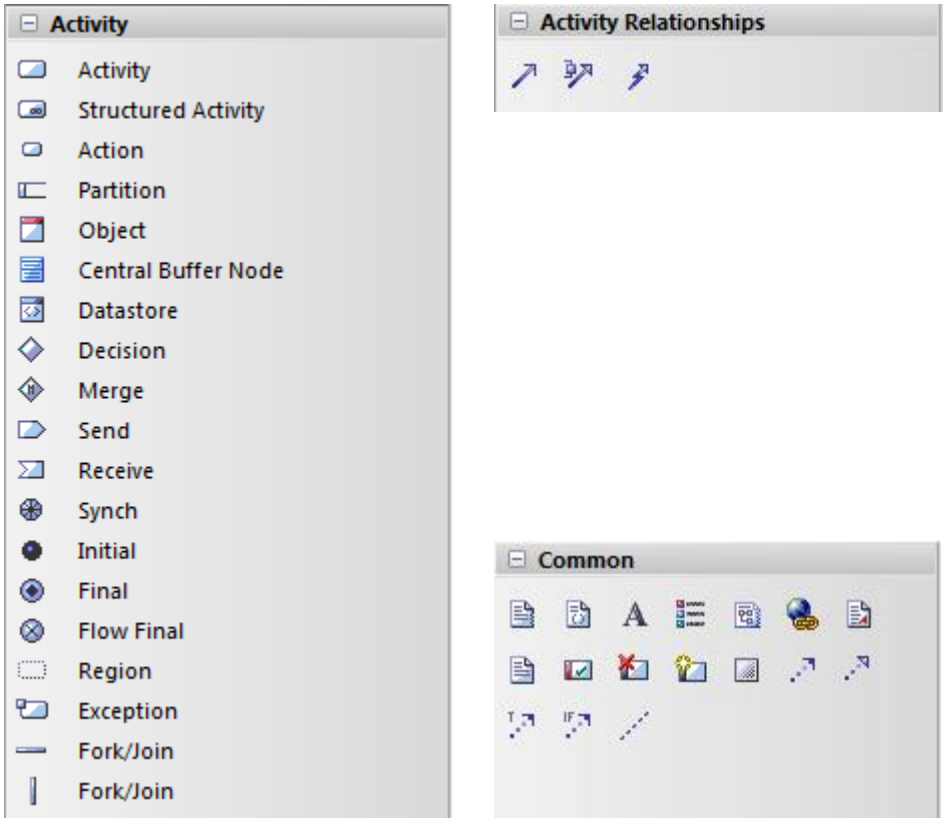
- *Thanh đồng bộ hóa (Synchronisation bar):* cho phép ta mở ra hoặc là đóng lại các nhánh chạy song song trong tiến trình.

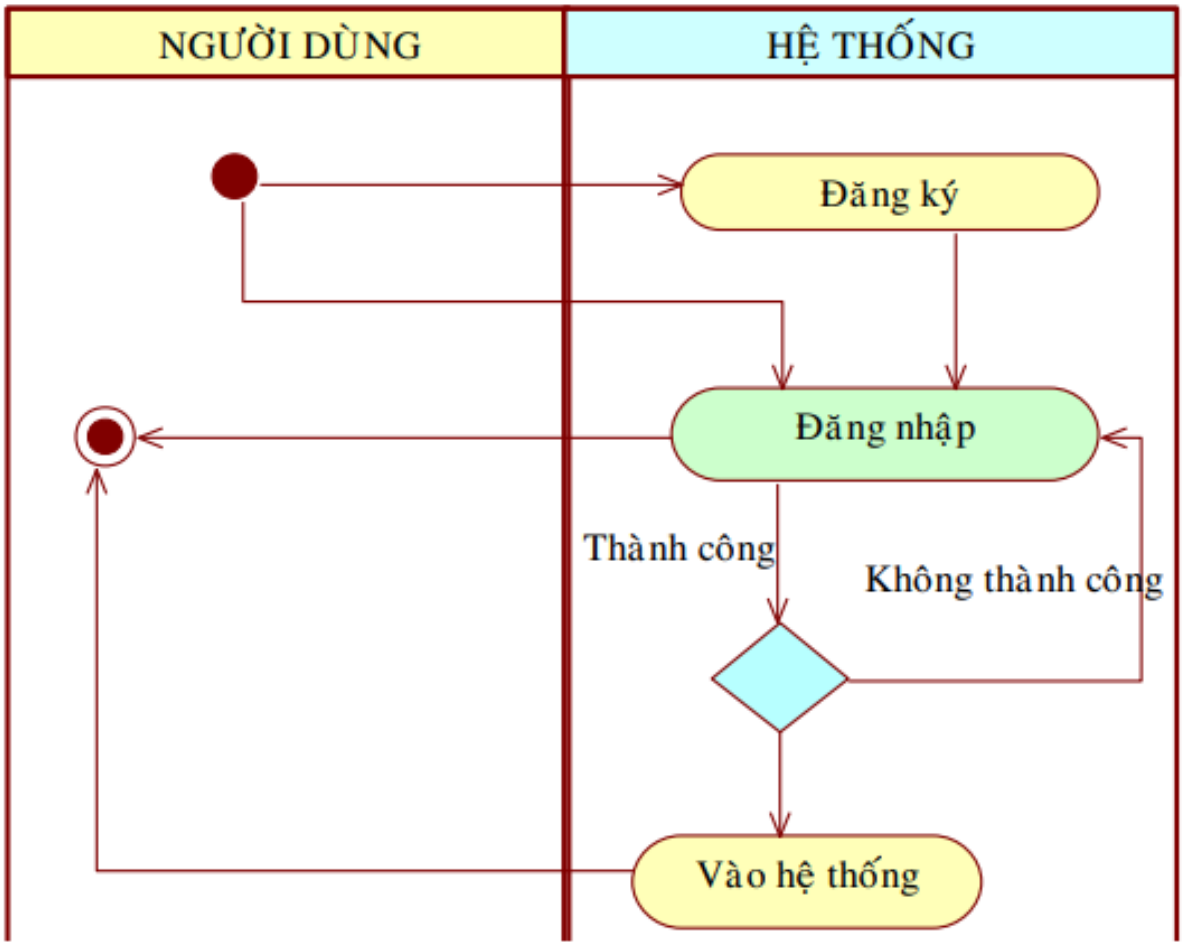


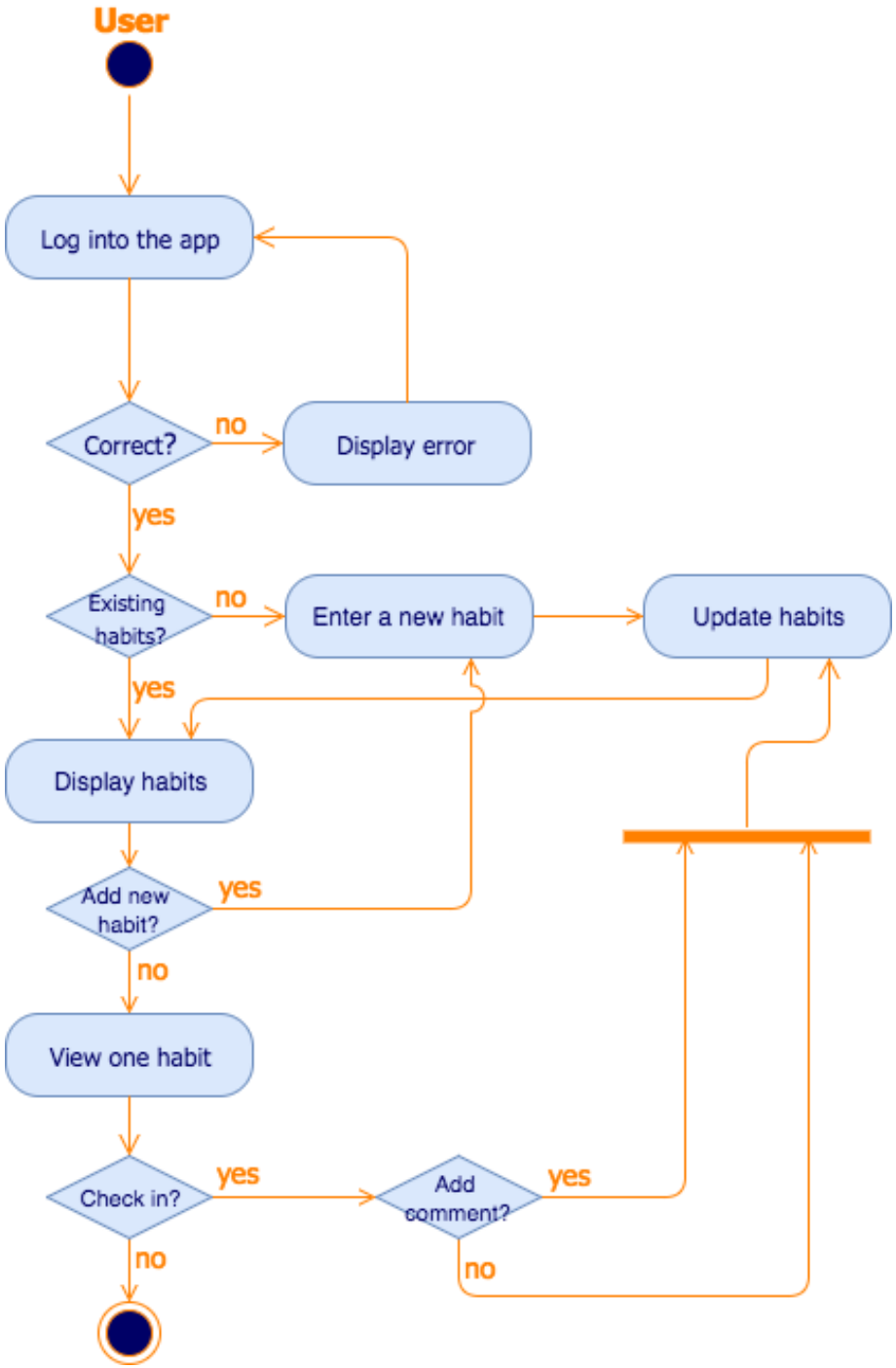
- *Quyết định (Decission):* Mô tả một lựa chọn điều kiện.

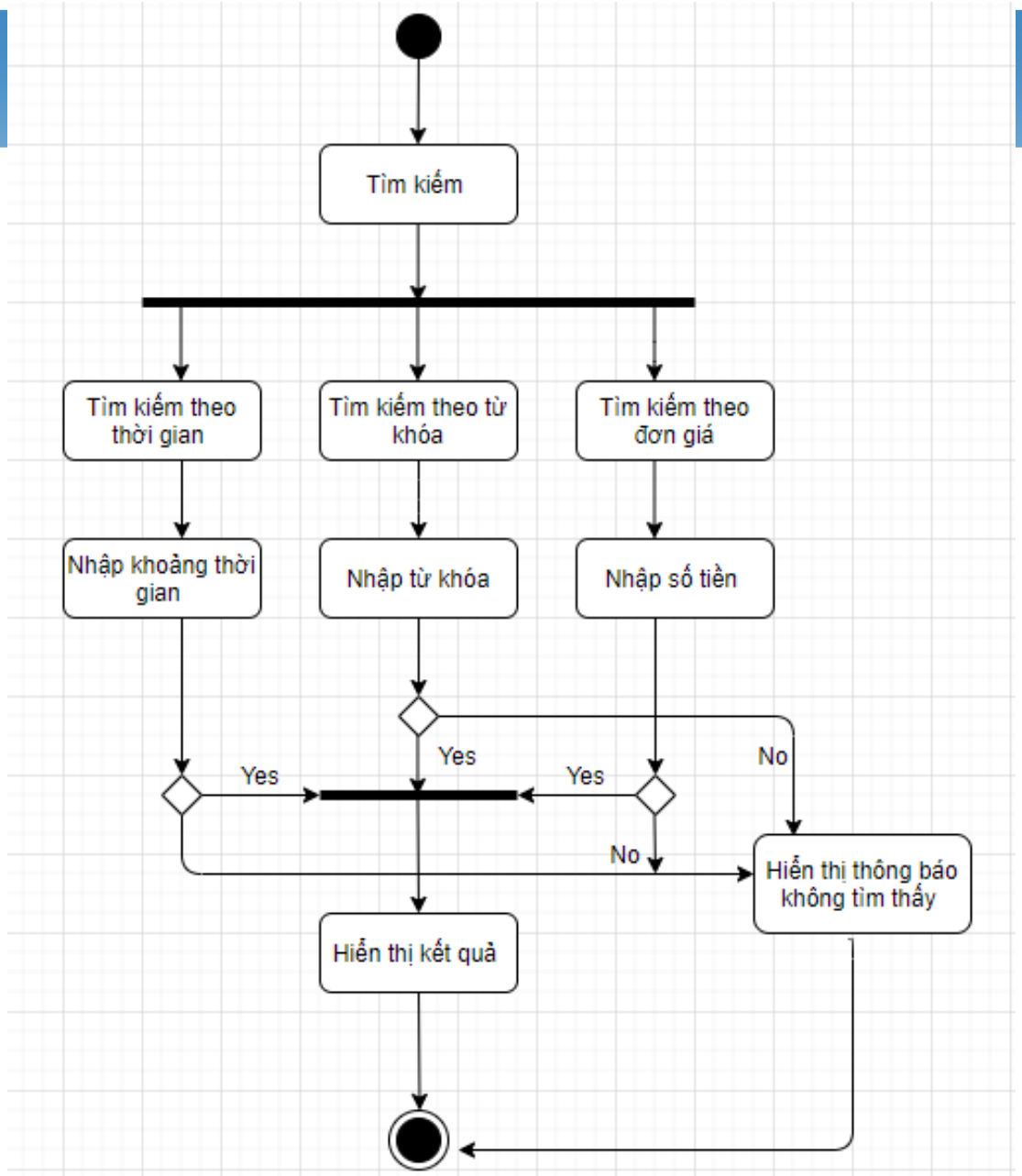


5.4 Sơ đồ Hoạt động




NGƯỜI DÙNG
HỆ THỐNG
Đăng ký
Đăng nhập
Thành công
Không thành công
Vào hệ thống


User
Log into the app
Correct?
no
Display error
yes
Existing habits?
no
Enter a new habit
Update habits
yes
Display habits
Add new habit?
yes
no
View one habit
Check in?
yes
Add comment?
yes
no
no


Tìm kiếm
Tìm kiếm theo thời gian
Tìm kiếm theo từ khóa
Tìm kiếm theo đơn giá
Nhập khoảng thời gian
Nhập từ khóa
Nhập số tiền
Yes
Yes
Yes
No
No
Hiển thị thông báo không tìm thấy
Hiển thị kết quả

Bài tập UML

Dựa trên sơ đồ UseCase đã vẽ cho trang sinhvien.dthu.edu.vn hãy vẽ sơ đồ hoạt động cho các use-case sau:

1) Vẽ sơ đồ hoạt động (Activity Diagram) cho use-case "Đăng nhập"

2) Vẽ sơ đồ hoạt động (Activity Diagram) cho use-case "Lập kế hoạch học tập"

Bài tập UML

- Quy trình mượn sách ở thư viện bao gồm các bước sau:
  - Để thực hiện chức năng mượn sách thì hệ thống yêu cầu phải đăng nhập thành công vào hệ thống
  - Kiểm tra thẻ thư viện: hệ thống yêu cầu nhập mã thẻ để kiểm tra thẻ thư viện này hợp lệ không (thẻ thật?, còn hạn? Vi phạm?...). Nếu thẻ không hợp lệ thì thông báo và kết thúc.
  - Tìm kiếm sách: Nếu thẻ hợp lệ thì hệ thống mở chức năng tìm kiếm sách theo tiêu chí (tên sách, tác giả, thể loại).
  - Chọn sách: Hệ thống hiển thị kết quả tìm sách và yêu cầu chọn sách mình muốn mượn (có thể chọn nhiều sách)
  - Xác nhận mượn sách: Hệ thống yêu cầu người dùng xác nhận thông tin sách và số lượng mượn.
  - Ghi nhận mượn sách: Hệ thống ghi nhận thông tin mượn sách vào cơ sở dữ liệu và cập nhật số lượng sách còn lại.
  - In phiếu mượn: Hệ thống in ra phiếu mượn sách cho người dùng.
  - Kết thúc mượn sách: Người dùng kết thúc quy trình mượn sách.

5.4 Sơ đồ Tương tác dạng tuần tự (Sequence)

a) Ý nghĩa

- Sơ đồ tuần tự:

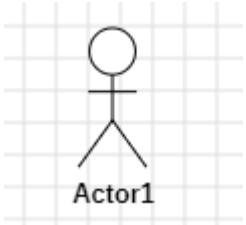
+ Biểu diễn mối quan hệ giữa các đối tượng, giữa các đối tượng và tác nhân theo thứ tự thời gian.

+ Biểu đồ tuần tự nhấn mạnh thứ tự thực hiện của các tương tác.

5.4 Sơ đồ Tương tác dạng tuần tự

b) Ký hiệu

- Đối tượng (object): được biểu diễn bởi các hình chữ nhật, bên trong là tên của đối tượng.



- Các thông điệp (message): được biểu diễn bằng các mũi tên hướng từ đối tượng gửi sang đối tượng nhận.

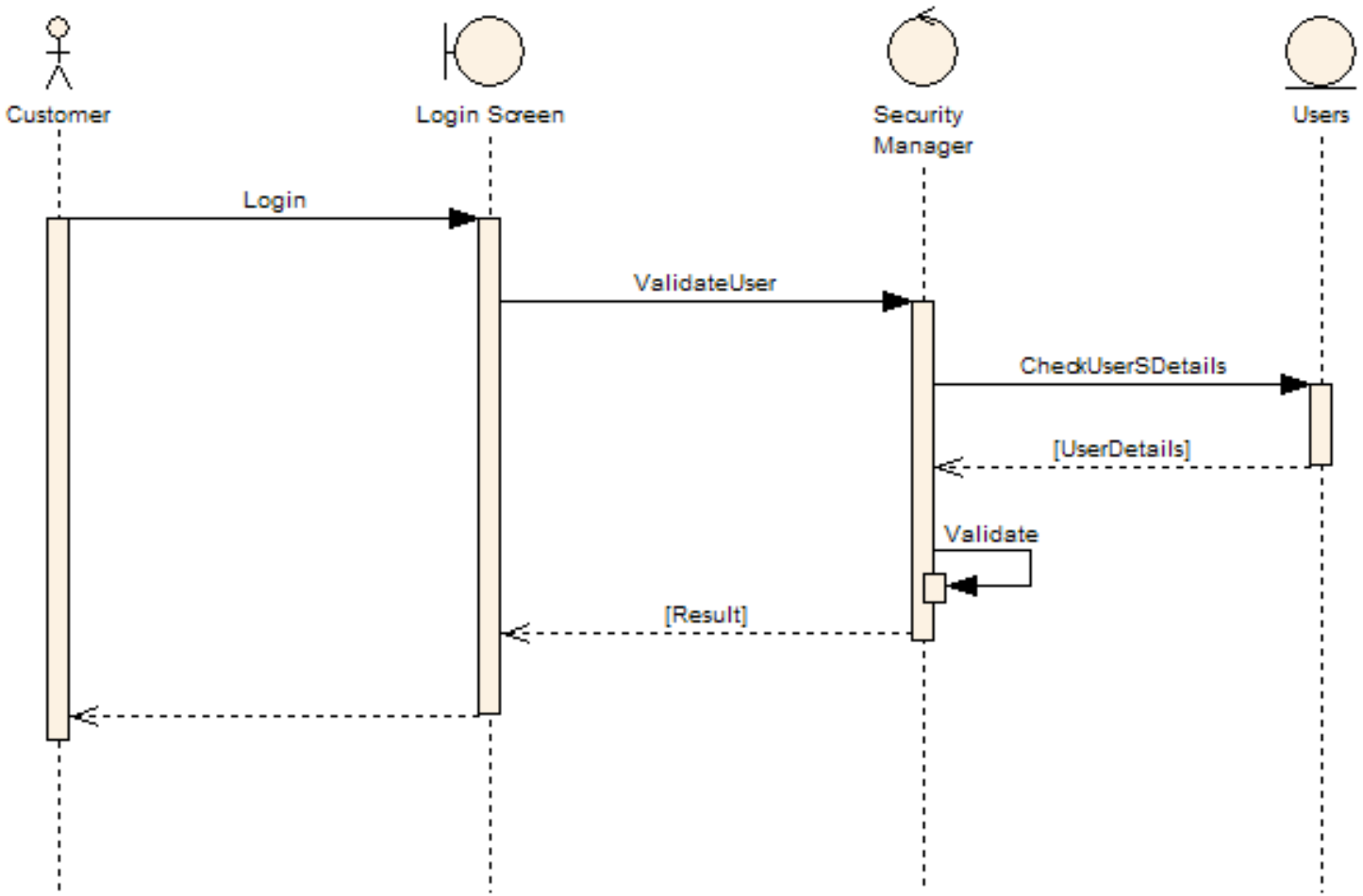
5.4 Sơ đồ Tương tác dạng tuần tự

- Đường lifeline: là một đường kẻ nối dài phía dưới đối tượng, mô tả quá trình của đối tượng trong tương tác thuộc sơ đồ.

- Chú thích: sơ đồ tuần tự cũng có thể có chú thích để người đọc dễ dàng hiểu được nội dung của biểu đồ đó.

5.4 Sơ đồ Tương tác dạng tuần tự

c) Ví dụ



Bài tập Sequence

Dựa trên sơ đồ UseCase đã vẽ cho trang sinhvien.dthu.edu.vn hãy vẽ sơ đồ hoạt động cho các use-case sau:

1) Vẽ sơ đồ tuần tự (sequence Diagram) cho use-case "Đăng nhập"
2) Vẽ sơ đồ tuần tự (Sequence Diagram) cho use-case "Lập kế hoạch học tập"

Bài tập Sequence

- Quy trình mượn sách ở thư viện bao gồm các bước sau:
  - Để thực hiện chức năng mượn sách thì hệ thống yêu cầu phải đăng nhập thành công vào hệ thống
  - Kiểm tra thẻ thư viện: hệ thống yêu cầu nhập mã thẻ để kiểm tra thẻ thư viện này hợp lệ không (thẻ thật?, còn hạn? Vi phạm?...). Nếu thẻ không hợp lệ thì thông báo và kết thúc.
  - Tìm kiếm sách: Nếu thẻ hợp lệ thì hệ thống mở chức năng tìm kiếm sách theo tiêu chí (tên sách, tác giả, thể loại).
  - Chọn sách: Hệ thống hiển thị kết quả tìm sách và yêu cầu chọn sách mình muốn mượn (có thể chọn nhiều sách)
  - Xác nhận mượn sách: Hệ thống yêu cầu người dùng xác nhận thông tin sách và số lượng mượn.
  - Ghi nhận mượn sách: Hệ thống ghi nhận thông tin mượn sách vào cơ sở dữ liệu và cập nhật số lượng sách còn lại.
  - In phiếu mượn: Hệ thống in ra phiếu mượn sách cho người dùng.
  - Kết thúc mượn sách: Người dùng kết thúc quy trình mượn sách.

5.5 Sơ đồ Tương tác dạng cộng tác

a) Ý nghĩa

- Biểu diễn mối quan hệ giữa các đối tượng; giữa các đối tượng và tác nhân, vai trò của các đối tượng trong tương tác.

- Biểu đồ cộng tác cũng có các thông điệp với nội dung tương tự như trong biểu đồ tuần tự.

5.5 Sơ đồ Tương tác dạng cộng tác

b) Ký hiệu

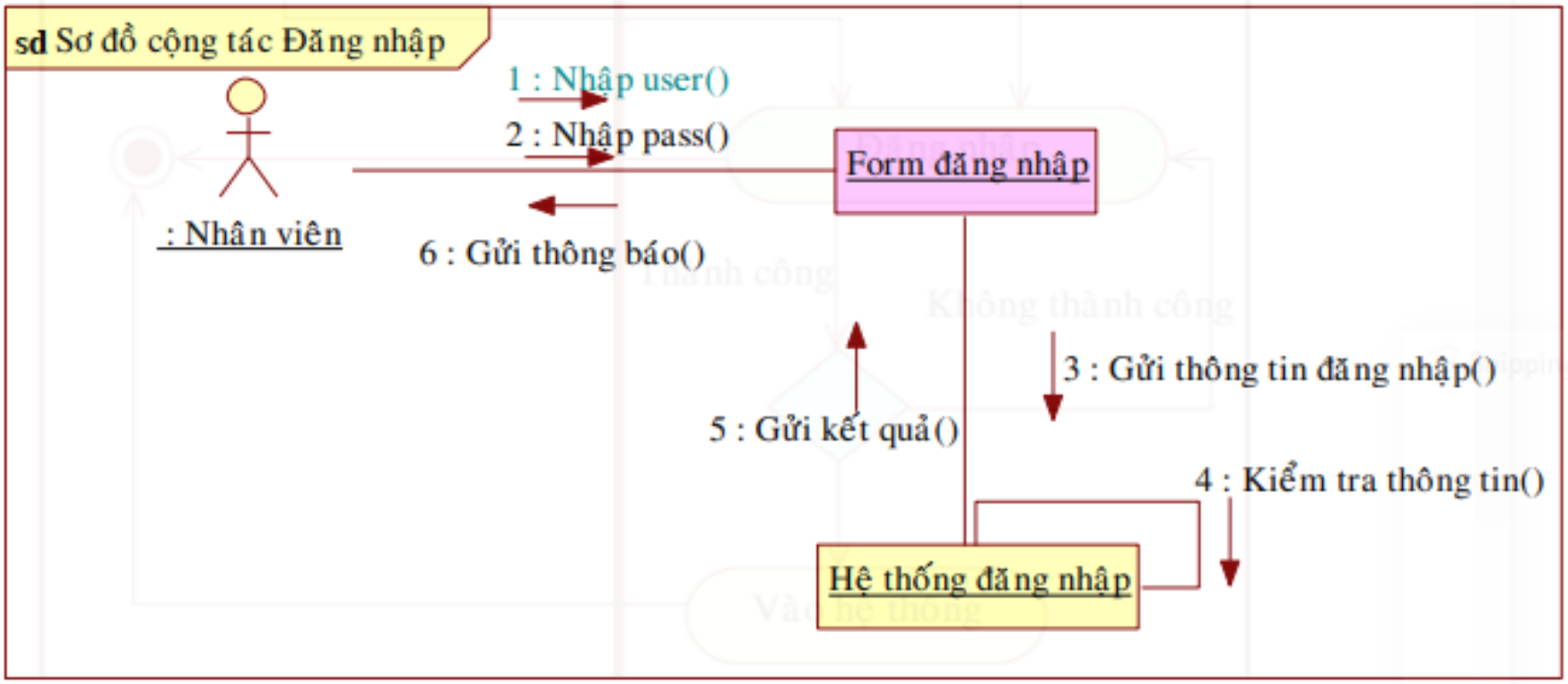
- *Các đối tượng:* được biểu diễn bởi các hình chữ nhật, bên trong là tên của đối tượng.

- *Các liên kết*: giữa hai đối tượng có tương tác sẽ có một liên kết nối 2 đối tượng đó. Liên kết này không có chiều.

- *Các thông điệp:* biểu diễn bằng các mũi tên hướng từ đối tượng gửi sang đối tượng nhận bên cạnh liên kết giữa 2 đối tượng đó.

- Các thông điệp: được đánh số thứ tự theo thứ tự xuất hiện trong kịch bản mô tả use case tương ứng.

5.5 Sơ đồ Tương tác dạng cộng tác



5.7 Sơ đồ Thành phần (Component)

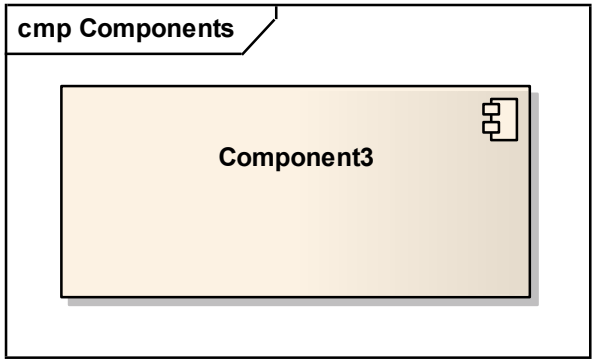
a) Ý nghĩa

- Sơ đồ thành phần: được sử dụng để biểu diễn các thành phần phần mềm cấu thành nên hệ thống.

5.7 Sơ đồ Thành phần

b) Ký hiệu

- *Thành phần*: Mô tả một thành phần của sơ đồ, mỗi thành phần có thể chứa nhiều lớp hoặc nhiều chương trình con.

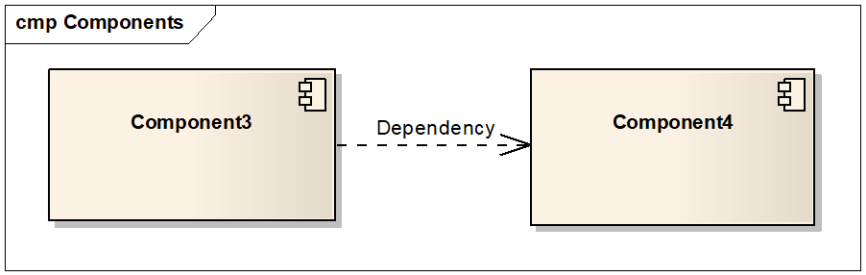


- *Giao tiếp:* Mô tả giao tiếp gắn với mỗi thành phần. Các thành phần trao đổi thông tin qua các giao tiếp.

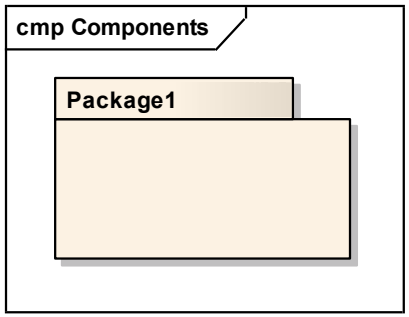
5.7 Sơ đồ Thành phần

b) Ký hiệu

- Mối quan hệ phụ thuộc giữa các thành phần (Dependency):

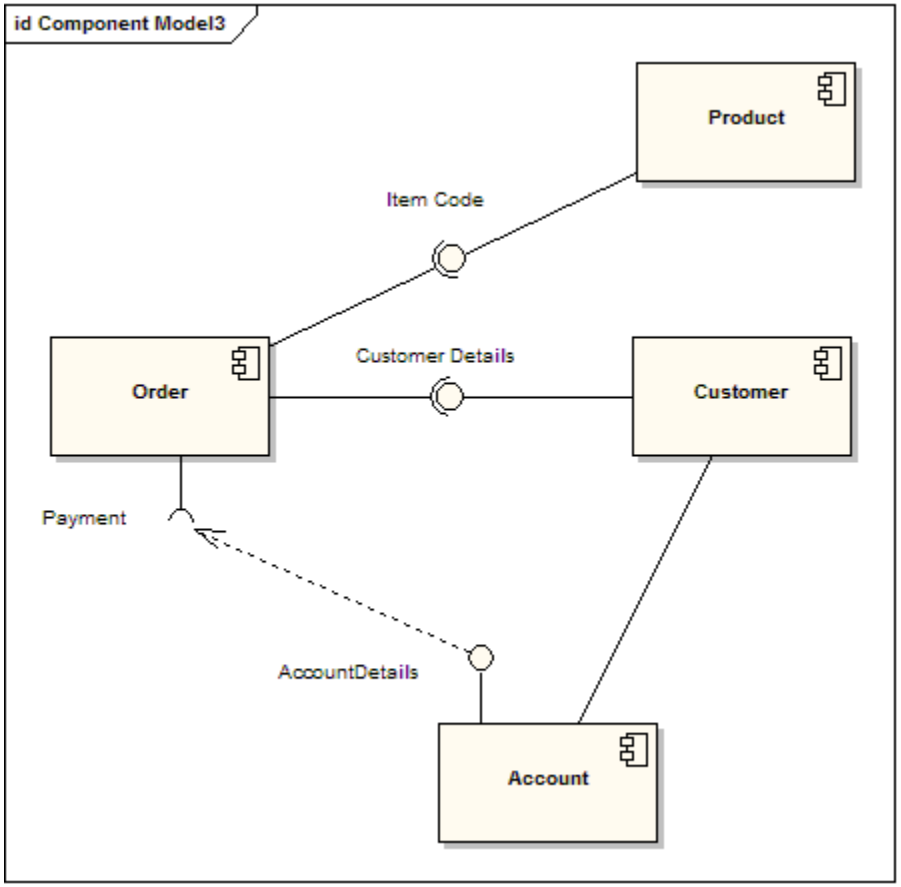


- Gói (Package): Để nhóm một số thành phần lại với nhau.

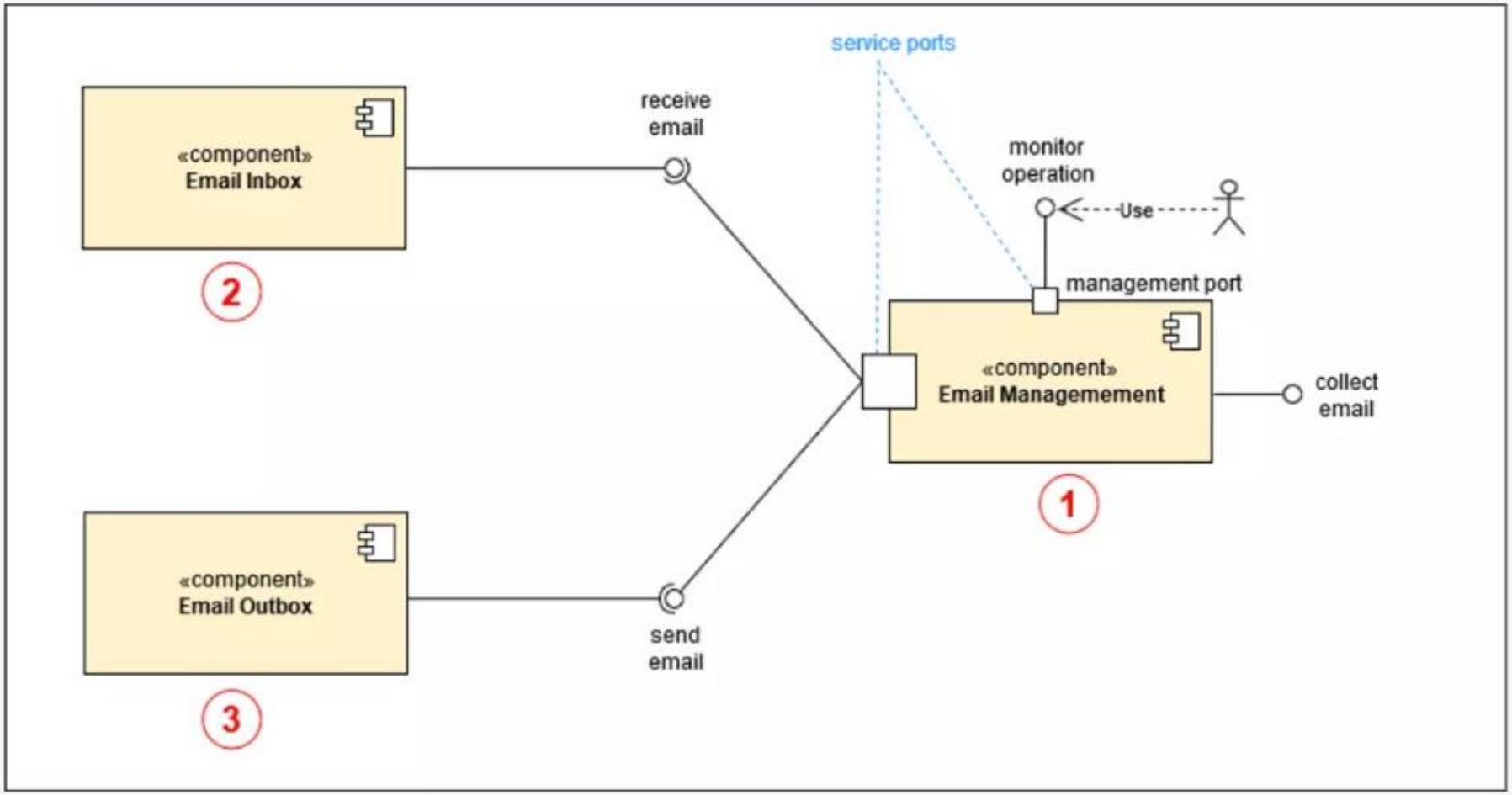


5.7 Sơ đồ Thành phần

c) Ví dụ



Example: Component diagram for an email software



Bài tập

- Thư viện của một trường đại học cần xây dựng một hệ thống phần mềm để quản lý việc mượn/trả sách và quản lý thông tin tài liệu. Hệ thống cần có các thành phần chính sau:
  - Thành phần Quản lý người dùng
    - Quản lý độc giả: đăng ký, cập nhật, xoá thông tin độc giả.
    - Quản lý thủ thư: thêm, cập nhật, phân quyền cho nhân viên thư viện.
  - Thành phần Quản lý tài liệu
    - Quản lý sách, báo, tạp chí, luận văn.
    - Tìm kiếm tài liệu theo tên, tác giả, thể loại.
  - Thành phần Quản lý mượn/trả
    - Xử lý nghiệp vụ mượn sách: ghi nhận ngày mượn, hạn trả.
    - Xử lý nghiệp vụ trả sách: cập nhật tình trạng sách, tính phí trễ hạn.
  - Thành phần Thanh toán & báo cáo
    - Xử lý việc thu phí trễ hạn.
    - Xuất báo cáo: số sách mượn, số sách trả muộn, tình hình sử dụng tài liệu.
  - Cơ sở dữ liệu
    - Lưu trữ thông tin độc giả, sách, phiếu mượn, phiếu trả, báo cáo.
  - Giao diện người dùng
    - Ứng dụng web hoặc desktop cho độc giả và thủ thư.

5.8 Sơ đồ Triển khai (Deployment Diagram)

a) Ý nghĩa

- Biểu diễn kiến trúc cài đặt và triển khai hệ thống dưới dạng các nodes,

- Các mối quan hệ giữa các node đó.

- Các nodes kết nối với nhau thông qua các liên kết truyền thông:

+ Kết nối mạng

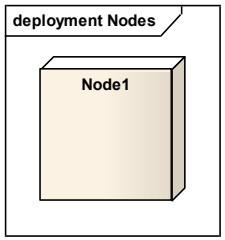
+ Liên kết TCP/IP

+ Microwave,… và được đánh số theo thứ tự thời gian tương tự như trong biểu đồ cộng tác.

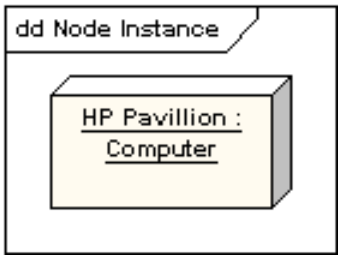
5.8 Sơ đồ Triển khai hệ thống

b) Ký hiệu

- *Các nodes:* Node hoặc là một thành phần phần cứng hoặc phần mềm. Nó được hiển thị như là một hình hộp ba chiều, như hình dưới đây.

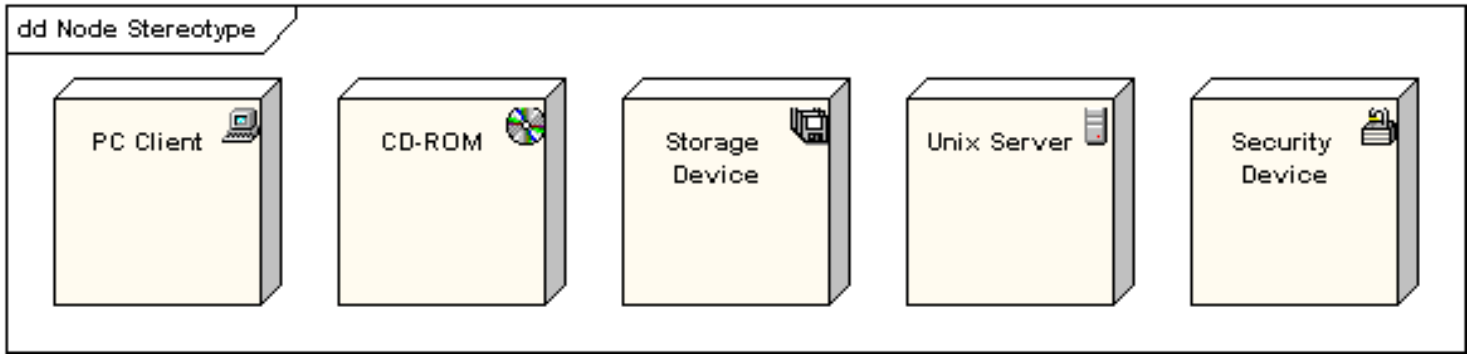


- *Node minh họa* (Instance Node): Node minh họa có thể biểu diễn bằng một sơ đồ. Nó có thể phân biệt với một node thực tế là tên được gạch chân và có dấu 2 chấm trước node cơ sở.



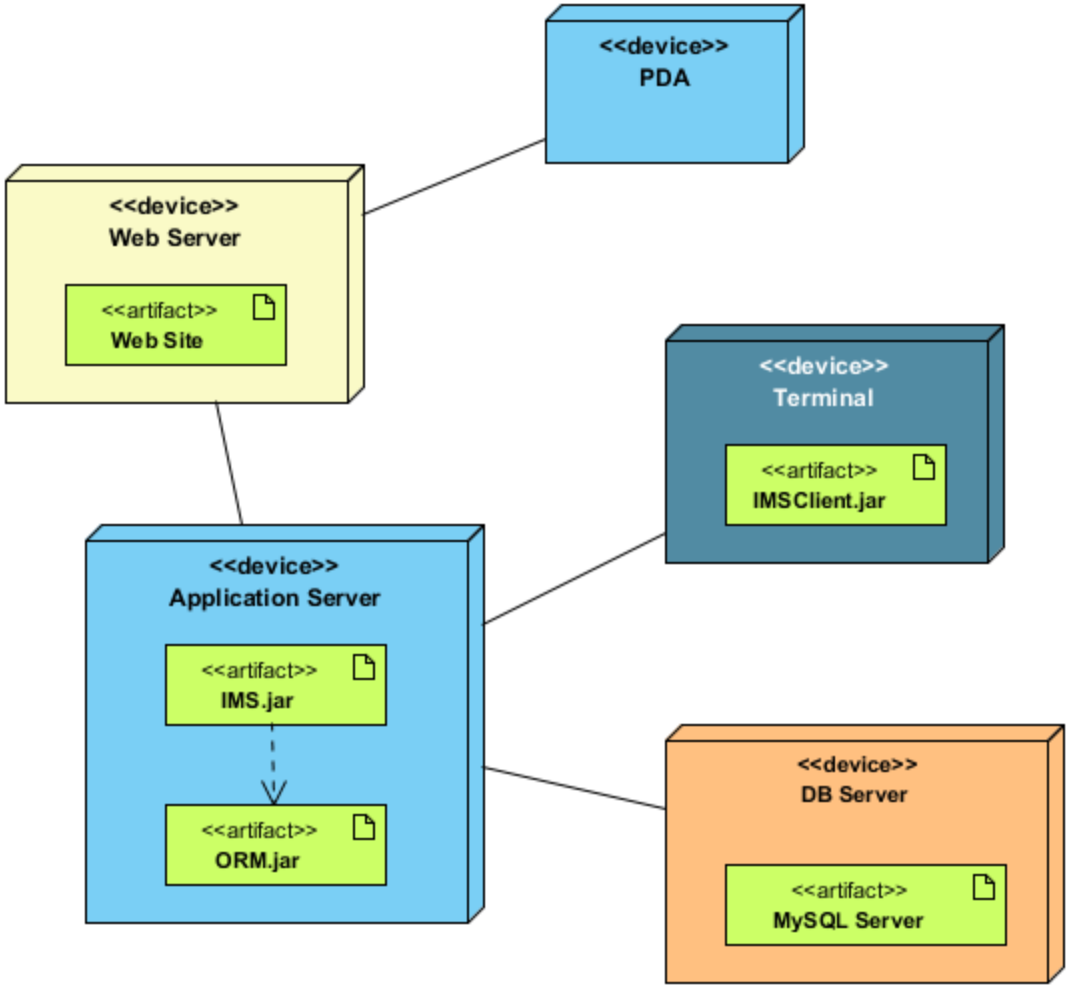
5.8 Sơ đồ Triển khai hệ thống

- *Node khuôn mẫu (Stereotypes Node):* Một số khuôn mẫu chuẩn được cung cấp cho các node: «cd-rom»,«computer»,«disk array», «pc», «pc client»,«pc server»,«secure»,«server», «storage», «unix server»,«user pc».



5.8 Sơ đồ Triển khai hệ thống

c) Ví dụ



Bài tập

- Sơ đồ triển khai cho hệ thống quản lý mượn trả sách thư viện:
  - Các node vật lý như:
    - Máy chủ ứng dụng (Application Server)
    - Máy chủ cơ sở dữ liệu (Database Server)
    - Máy trạm người dùng (User Workstation)
    - Máy in (Printer)
  - Các thành phần phần mềm được triển khai trên các node này:
    - Ứng dụng quản lý thư viện và giao diện người dùng trên máy chủ ứng dụng hoặc máy trạm người dùng
    - Cơ sở dữ liệu trên máy chủ cơ sở dữ liệu
    - Module in phiếu mượn trên máy in hoặc máy chủ ứng dụng
  - Các kết nối mạng giữa các node, ví dụ kết nối TCP/IP giữa máy trạm với máy chủ, hoặc kết nối in ấn từ máy chủ đến máy in.

Tool UML

- **StarUML**
  - https://staruml.io/
- Draw.io (online)
  - https://app.diagrams.net/
- Astah
  - https://astah.net/products/astah-uml/
- Dbdiagram.io (online)
  - https://dbdiagram.io/home

- Vẽ sơ đồ hoạt động (Activity) cho use-case “Lập kế hoạch học tập”
- Vẽ sơ đồ tuần tự (Sequence) cho use-case “Lập kế hoạch học tập”

Bài tập UML

- Quy trình mượn sách ở thư viện bao gồm các bước sau:
  - Để thực hiện chức năng mượn sách thì hệ thống yêu cầu phải đăng nhập thành công vào hệ thống
  - Kiểm tra thẻ thư viện: hệ thống yêu cầu nhập mã thẻ để kiểm tra thẻ thư viện này hợp lệ không (thẻ thật?, còn hạn? Vi phạm?...). Nếu thẻ không hợp lệ thì thông báo và kết thúc.
  - Tìm kiếm sách: Nếu thẻ hợp lệ thì hệ thống mở chức năng tìm kiếm sách theo tiêu chí (tên sách, tác giả, thể loại).
  - Chọn sách: Hệ thống hiển thị kết quả tìm sách và yêu cầu chọn sách mình muốn mượn (có thể chọn nhiều sách)
  - Xác nhận mượn sách: Hệ thống yêu cầu người dùng xác nhận thông tin sách và số lượng mượn.
  - Ghi nhận mượn sách: Hệ thống ghi nhận thông tin mượn sách vào cơ sở dữ liệu và cập nhật số lượng sách còn lại.
  - In phiếu mượn: Hệ thống in ra phiếu mượn sách cho người dùng.
  - Kết thúc mượn sách: Người dùng kết thúc quy trình mượn sách.

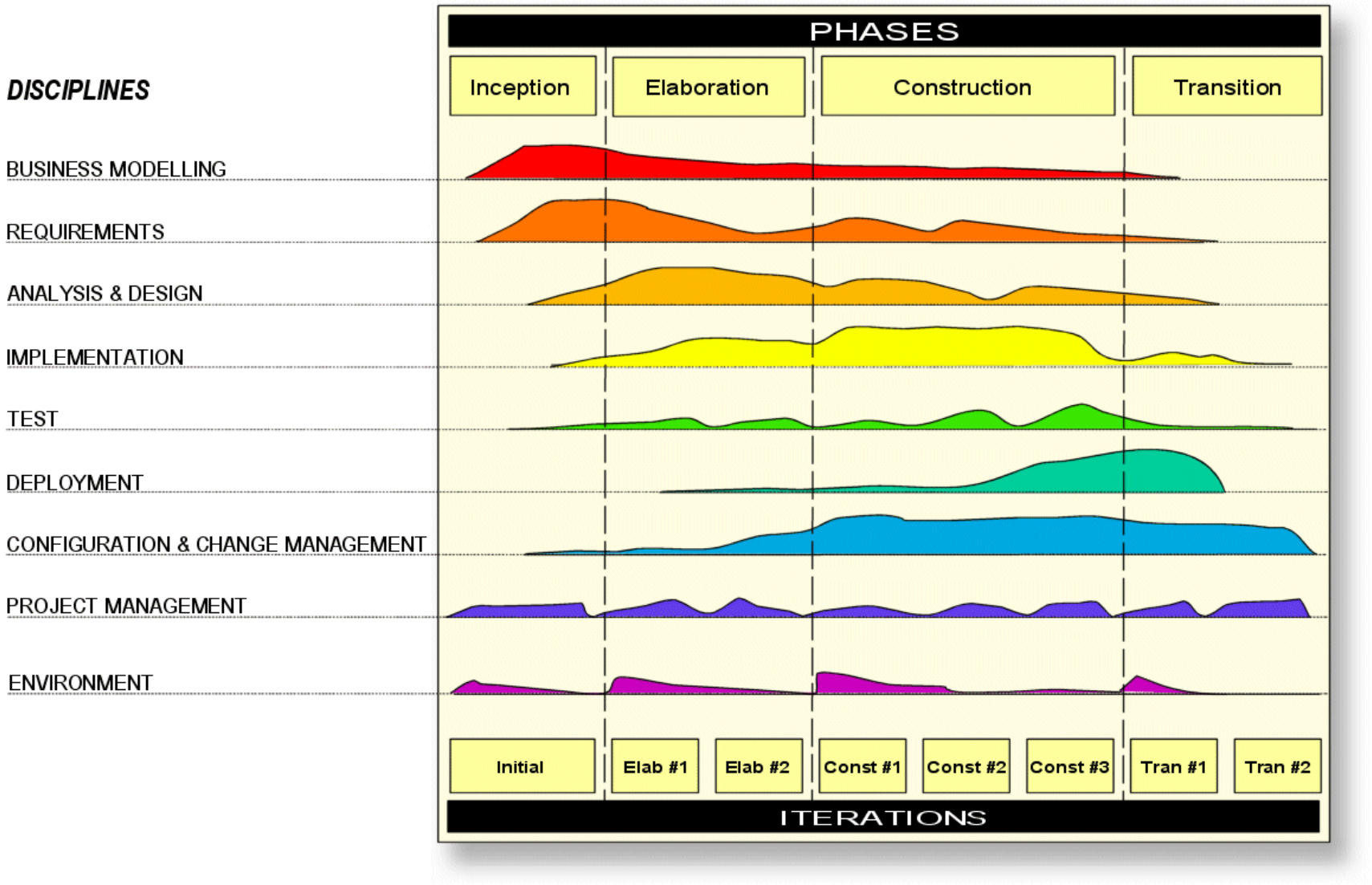
Bài tập UML – yêu cầu

- Sử dụng UML, hãy vẽ các sơ đồ sau:
  - Use Case
  - Activity Diagram
  - Sequence Diagram
  - Class Diagram
  - Component Diagram
  - State Machine Diagram
  - Deployment Diagram
  - Package Diagram
- Hãy sử dụng StarUML để vẽ lại các sơ đồ trên

6. Quy trình RUP

- Giới thiệu RUP – Rational Unified Process
  - RUP Là quy trình công nghệ phần mềm được phát triển bởi hãng Rational
  - RUP hỗ trợ các hoạt động giữa các nhóm, phân chia công việc cho từng thành viên trong nhóm, trong từng giai đoạn khác nhau của quá trình phát triển phần mềm.
  - RUP sử dụng hệ thống ký hiệu trực quan của UML
  - RUP được phát triển song song với UML.

6. Quy trình RUP



6. Quy trình RUP

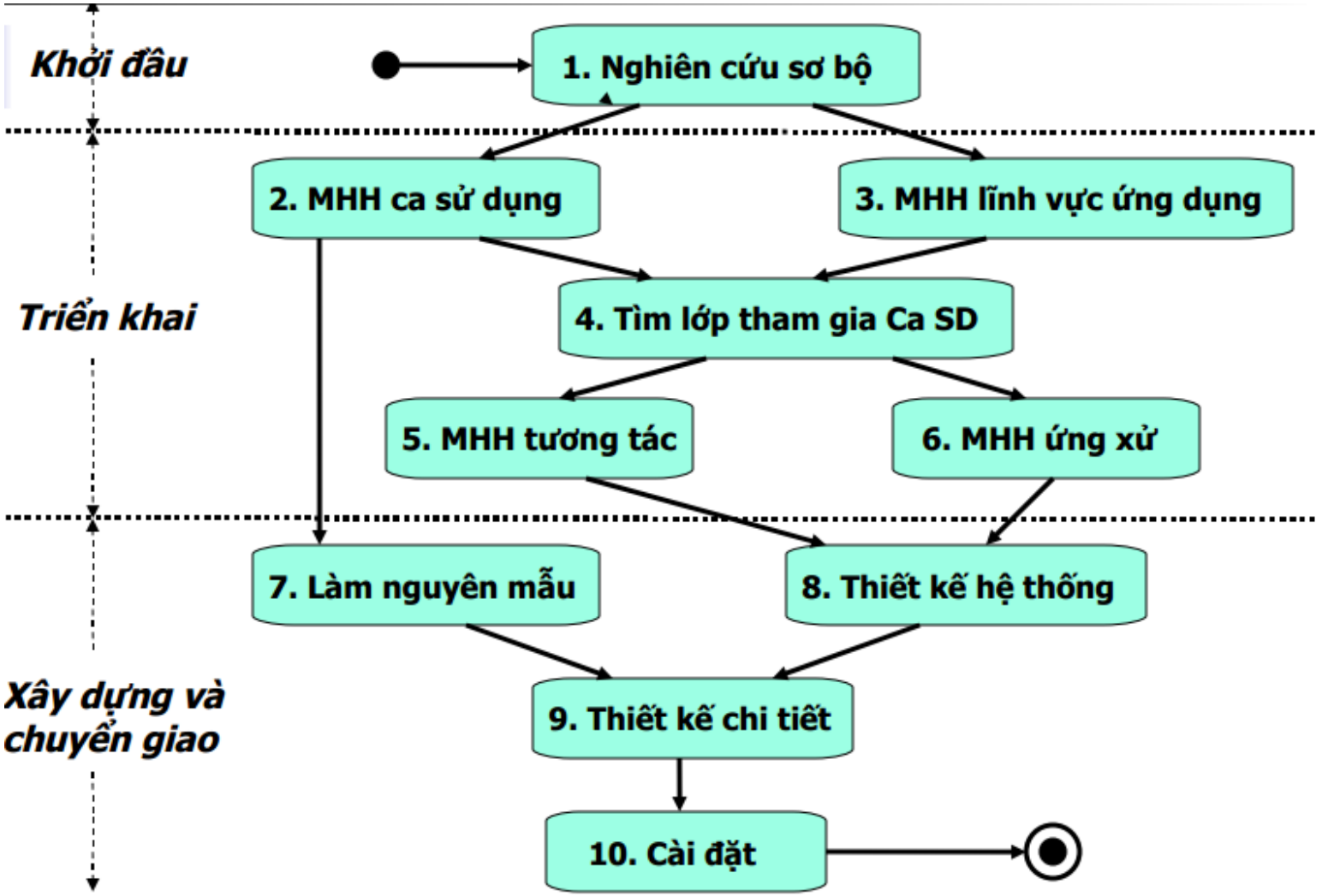
- Các nguyên tắc cơ bản của RUP
  - Lặp và tăng trưởng
  - Tập trung vào kiến trúc
  - Dẫn dắt theo các ca sử dụng
  - Khống chế bởi các nguy cơ

6. Quy trình RUP

- Các pha và công đoạn của RUP
  - RUP gồm 4 pha:
    - Khởi đầu (inception)
    - Triển khai (elaboration)
    - Xây dựng (construction)
    - Chuyển giao (transition)
  - Mỗi pha gồm nhiều vòng lặp.
  - Mỗi vòng lặp thường gồm năm cộng đoạn:
    - Xác định nhu cầu
    - Phân tích
    - Thiết kế
    - Cài đặt
    - Đánh giá

6. Quy trình RUP

Một tiến trình 10 bước



6. Quy trình RUP

- Khởi đầu (inception)
  - Định nghĩa mục đích, yêu cầu
  - Tìm hiểu vấn đề và phác thảo phương án thực hiện
  - Đánh giá lợi ích
  - Xác định các nguy cơ
  - Định nhĩa các tiêu chuẩn đánh giá
  - Đánh giá thời gian thực hiện và chi phí
  - Xây dựng phương án thực hiện
  - Đánh giá tính khả thi về kỹ thuật

6. Quy trình RUP

- Triển khai (elaboration)
  - Phân tích yêu cầu
  - Xác định các chức năng của hệ thống
  - Lựa chọn kiến trúc của hệ thống
  - Xây dựng kế hoạch thực hiện
  - Thực hiện bởi nhà tin học
  - Chủ yếu dựa vào các usecase
  - Giúp người sử dụng hiểu rõ cái họ cần
  - Chi tiết hóa dần các usecase
  - Xây dựng mô hình các lớp

6. Quy trình RUP

- Triển khai (elaboration) – kết quả
  - Mô tả chức năng của hệ thống (usecase, mô tả usecase và sơ đồ lớp)
  - Kiến trúc thực thi được của hệ thống
  - Kế hoạch hoàn chỉnh để phát triển hệ thống
  - Kế hoạch chi tiết các bước lặp

6. Quy trình RUP

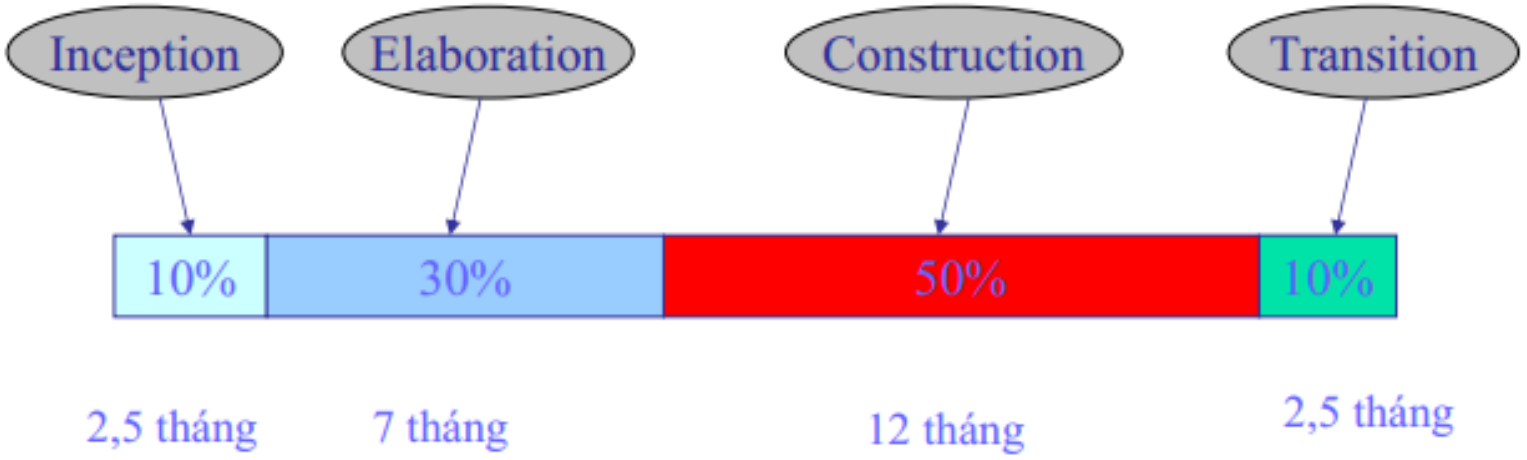
- Xây dựng (construction)
  - Phát triển phần mềm cho người sử dụng
  - Thực hiện bởi nhiều bước lặp
    - Xây dựng kiến trúc chi tiết
    - Phát triển một phần hệ thống
    - Kiểm thử một phần hệ thống
    - Cài đặt một phần hệ thống
  - Mỗi bước lặp cho một nguyên mẫu thực thi được (executable prototype)
  - Thêm dần dần các chức năng

6. Quy trình RUP

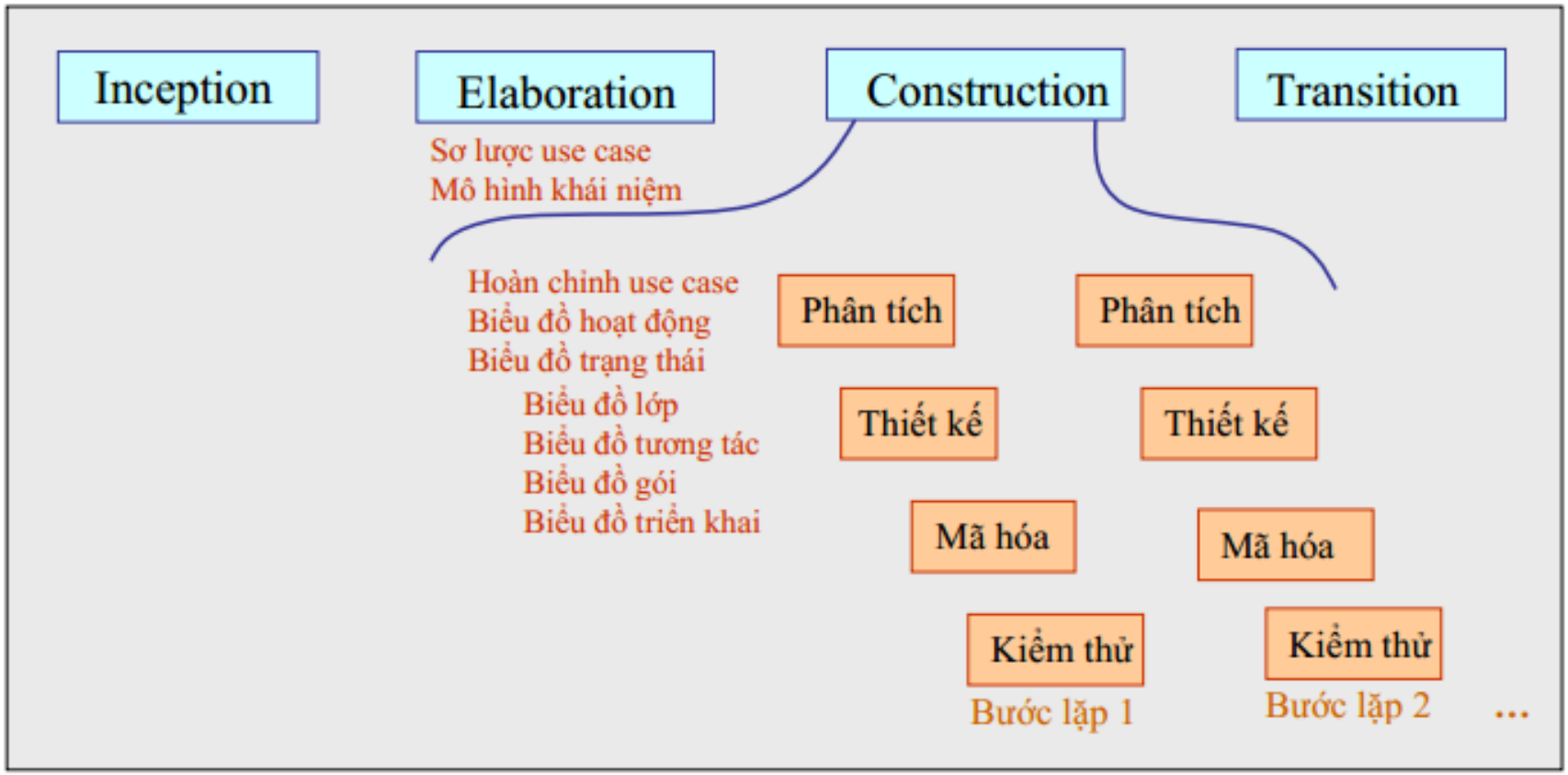
- Chuyển giao (transition)
  - Giai đoạn phát triển gần như kết thúc
  - Đánh giá dự án
  - Chuyển giao phần mềm cho người sử dụng
  - Huấn luyện người sử dụng
  - Kết quả:
    - Hệ thống thực thi được: phiên bản β, phiên bản chính thức
    - Hướng dẫn cài đặt
    - Hướng dẫn sử dụng

6. Quy trình RUP

- Ví dụ: một dự án kéo dài 2 năm



UML và RUP


Inception
Elaboration
Construction
Transition
Sơ lược use case
Mô hình khái niệm
Hoàn chỉnh use case
Biểu đồ hoạt động
Biểu đồ trạng thái
Biểu đồ lớp
Biểu đồ tương tác
Biểu đồ gói
Biểu đồ triển khai
Phân tích
Thiết kế
Mã hóa
Kiểm thử
Bước lặp 1
Phân tích
Thiết kế
Mã hóa
Kiểm thử
Bước lặp 2
...

Câu hỏi thảo luận