



TRƯỜNG ĐẠI HỌC MỞ
TP HỒ CHÍ MINH

Bắt đầu lúc 7h15

BÀI GIẢNG
**PHÂN TÍCH THIẾT KẾ HỆ THỐNG
THÔNG TIN**

**Chương 5: Phân tích thiết kế
hướng đối tượng**

ThS. Nguyễn Thị Phương Trang
trang.ntp@ou.edu.vn



Nội dung



Giới thiệu tổng quan về UML



Một số Case tool hỗ trợ UML



Một số biểu đồ UML cơ bản



Cài đặt (ánh xạ) biểu đồ



Giới thiệu Visual Paradigm

Hướng phân tích hệ thống



- Hướng không đối tượng
 - ▣ Các mô hình xử lý
 - Dựa trên các hành vi và hoạt động
 - ▣ Các mô hình dữ liệu
 - Dựa trên thể hiện tiêu biểu tinh của dữ liệu.
- Hướng đối tượng
 - ▣ Kết nối giữa xử lý và dữ liệu
 - ▣ Thực hiện một cách “tự nhiên hơn”



Các đặc tính cơ bản của hệ thống HĐT

- Lớp và đối tượng
- Các phương thức và thông điệp
- Tính bao bọc và che dấu thông tin
- Tính kế thừa
- Tính đa hình

Ký hiệu và viết tắt



- C (Classes) – Lớp
- O Objects – Đối tượng
- M (Methods and Messages) – Phương thức và thông điệp
- P (Polymorphism) – Tính đa hình
- I (Inheritance) – Tính kế thừa
- E (Encapsulation) – Tính bao bọc

Đối tượng



- Là thành phần trọng tâm của cách tiếp cận hướng đối tượng, biểu diễn từ thế giới thực sang thể hiện của tin học.
- Là một đại diện của bất kỳ sự vật nào cần mô hình trong hệ thống: diễn đạt một thực thể vật lý hoặc thực thể quan niệm, hoặc thực thể phần mềm.
- Đóng một vai trò xác định trong lĩnh vực ứng dụng
- Đối tượng có thể là một thực thể hữu hình trực quan (một con người, một vị trí, một sự vật,...) hoặc một khái niệm, một sự kiện (phòng ban, bộ phận,...)

Đối tượng



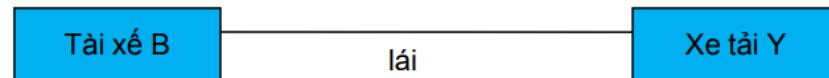
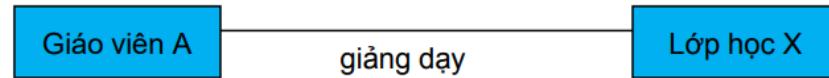
- Một đối tượng phải thỏa 3 nguyên lý:
 - Phân biệt (distinction): đơn vị duy nhất (định danh)
 - Bền vững(permanence): quá trình sống (trạng thái)
 - Hoạt động (activity): vai trò, hành vi

Đối tượng = định danh + trạng thái + hành vi

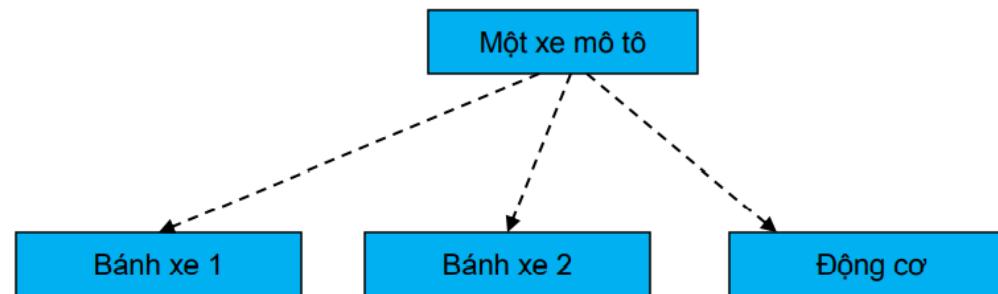
Đối tượng



- Liên kết giữa các đối tượng
 - Mối kết hợp (association): liên kết ngữ nghĩa



- Phân cấp (hierarchy): liên kết cấu trúc



Điểm khác biệt giữa PTTK hướng cấu trúc – PTTK hướng đối tượng



	Hướng cấu trúc	Hướng đối tượng
Cách tiếp cận	<ul style="list-style-type: none"> - Phân chia chương trình chính thành nhiều chương trình con nhằm đến thực một công việc xác định, - Cách thực hiện : Phương pháp thiết kế từ trên xuống (top-down). 	<ul style="list-style-type: none"> - Tập trung vào cả hai khía cạnh của hệ thống là dữ liệu và hành động - Phương pháp thiết kế từ dưới lên (bottom-up) .
Ưu điểm	<ul style="list-style-type: none"> - Phân tích được các chức năng của hệ thống. - Dễ theo dõi luồng dữ liệu. 	<ul style="list-style-type: none"> - Gần gũi với thế giới thực - Đóng gói che giấu thông tin làm cho hệ thống tin cậy hơn. - Tái sử dụng dễ dàng.
Nhược điểm	<ul style="list-style-type: none"> - Không hỗ trợ việc sử dụng lại - Không phù hợp cho phát triển các phần mềm lớn 	<ul style="list-style-type: none"> - Phương pháp này khá phức tạp, khó theo dõi được luồng dữ liệu do có nhiều luồng dữ liệu ở đầu vào.

1. Tổng quan về UML



- ❖ UML (Unified Model Language) là một ngôn ngữ dùng cho phân tích thiết kế hướng đối tượng (OOAD – Object Oriented Analysis and Design)
- ❖ Được duy trì và phát triển bởi OMG (Object Management Group), do **Jacobson**, **Booch**, **Rumbaugh** sáng lập. Ngoài ra còn có hàng trăm các tập đoàn lớn khác bảo trợ phát triển.
- ❖ UML 2.0 có 13 loại biểu đồ để thể hiện các khung nhìn khác nhau (View) về hệ thống.
- ❖ Các biểu đồ UML cho ta cái nhìn rõ hơn về hệ thống (cả cái nhìn tĩnh và động)

1. Tổng quan về UML....



- ❖ Hiện nay UML được sử dụng rất phổ biến trong các dự án phần mềm.
- ❖ UML thể hiện phương pháp phân tích hướng đối tượng nên không lệ thuộc ngôn ngữ LT.
- ❖ Có rất nhiều công cụ phần mềm hỗ trợ phân tích thiết kế dùng UML.
- ❖ Nhiều công cụ có thể sinh ra mã từ UML và ngược lại (từ mã thành UML-Reverse Eng)
- ❖ UML không phải là ngôn ngữ lập trình !.

UML dùng để làm gì ?



- ❖ UML là một ngôn ngữ dùng để:
 1. Trực quan hóa (Visualizing)
 2. Đặc tả (Specifying)
 3. Xây dựng (Constructing)
 4. Viết tài liệu (Documenting)

Trực quan hóa-Visualizing



- ❖ Dùng tập các ký hiệu đồ họa phong phú để biểu diễn hệ thống đang được nghiên cứu.
- ❖ Hệ thống ký hiệu đều có ngũ nghĩa chặt chẽ, có thể hiểu bởi nhiều công cụ khác nhau.
- ❖ Giúp cho các nhà thiết kế, nhà lập trình khác biệt về ngôn ngữ đều có thể hiểu được.

UML là ngôn ngữ cho đặc tả - specifying



- ❖ UML giúp xây dựng các mô hình chính xác, đầy đủ và không nhập nhằng.
- ❖ Tất cả các công đoạn từ phân tích, thiết kế cho đến triển khai đều có các biểu đồ UML biểu diễn.
- ❖ Use case (dùng cho phân tích); Class, Sequence, Activity... (cho thiết kế); Component, Deployment (cho triển khai).

Xây dựng - Constructing



- ❖ Các mô hình của UML có thể kết nối với nhiều ngôn ngữ lập trình. Tức là có thể ánh xạ các mô hình UML về một ngôn ngữ lập trình như C++, Java...
- ❖ Việc chuyển các mô hình trong UML thành Code trong ngôn ngữ lập trình ➔ Forward engineering
- ❖ Việc chuyển ngược trở lại code trong một ngôn ngữ lập trình thành UML ➔ Reverse Engineering.
- ❖ Cần công cụ để chuyển đổi “xuôi” & “ngược”

Viết tài liệu (Documenting)



- ❖ Giúp xây dựng tài liệu đặc tả - requirements
- ❖ Tài liệu kiến trúc (architecture)
- ❖ Tài liệu thiết kế
- ❖ Source code
- ❖ Tài liệu để kiểm thử - Test
- ❖ Tài liệu mẫu - Prototype
- ❖ Tài liệu triển khai – Deployment

Các đặc trưng của một tiến trình sử dụng UML



- Tập trung vào người dùng (user – concentrated):
 - Phân tích viên xác định các tính năng của hệ thống thông qua các use case
 - Người dùng xác nhận các use case này
 - Thiết kế viên và người phát triển hiện thực hoá các use case
 - Người thử nghiệm kiểm tra hệ thống về việc thỏa mãn các use case được đặt ra.

Các đặc trưng của một tiến trình sử dụng UML



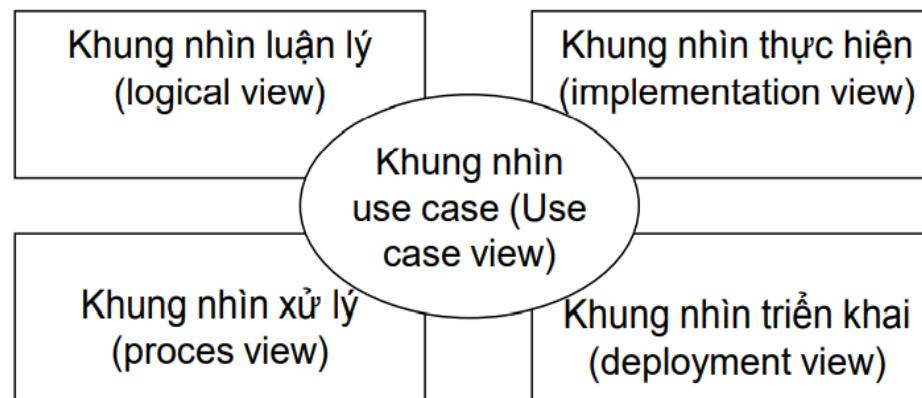
□ Các khung nhìn về hệ thống:

Người dùng

Chức năng

Lập trình viên

Quản trị phần mềm



Quản trị viên tích hợp hệ thống

Hiệu năng
Tính co giãn
Thông lượng

Thiết kế viên hệ thống

Hình thái hệ thống
Chuyển giao, cài đặt
Truyền thông

2. Một số Case tool (Công cụ) hỗ trợ UML



- ❖ Rational Rose (của hãng Rational) <http://www-128.ibm.com/developerworks/downloads/r/rsd/?>
- ❖ Visual Paradiagm <http://www.visual-paradigm.com>
- ❖ Microsoft Visio www.microsoft.com
- ❖ Power designer <http://www.sybase.com>
- ❖ Visual Case <http://www.visualcase.com>
- ❖ Pacestar UML Diagrammer www.peacestar.com
- ❖

3. Một số biểu đồ UML cơ bản



- ❑ Các sơ đồ mô tả khía cạnh tĩnh
 - ❑ Sơ đồ đối tượng (object diagram)
 - ❑ Sơ đồ lớp (class diagram)
 - ❑ Sơ đồ use case (use case diagram)
 - ❑ Sơ đồ thành phần (component diagram)
 - ❑ Sơ đồ triển khai (deployment diagram)
- ❑ Các sơ đồ mô tả khía cạnh động
 - ❑ Các sơ đồ tương tác (interaction diagram)
 - Sơ đồ tuần tự (sequence diagram)
 - Sơ đồ hợp tác (collaboration diagram)
 - ❑ Sơ đồ hoạt động (activity diagram)
 - ❑ Sơ đồ chuyển dịch trạng thái (state transition diagram)

3. Một số biểu đồ UML cơ bản



1

- ❖ Biểu đồ ca sử dụng
Use Case
Diagram

Component

Deployment

Communication/
Collaboration

Timing

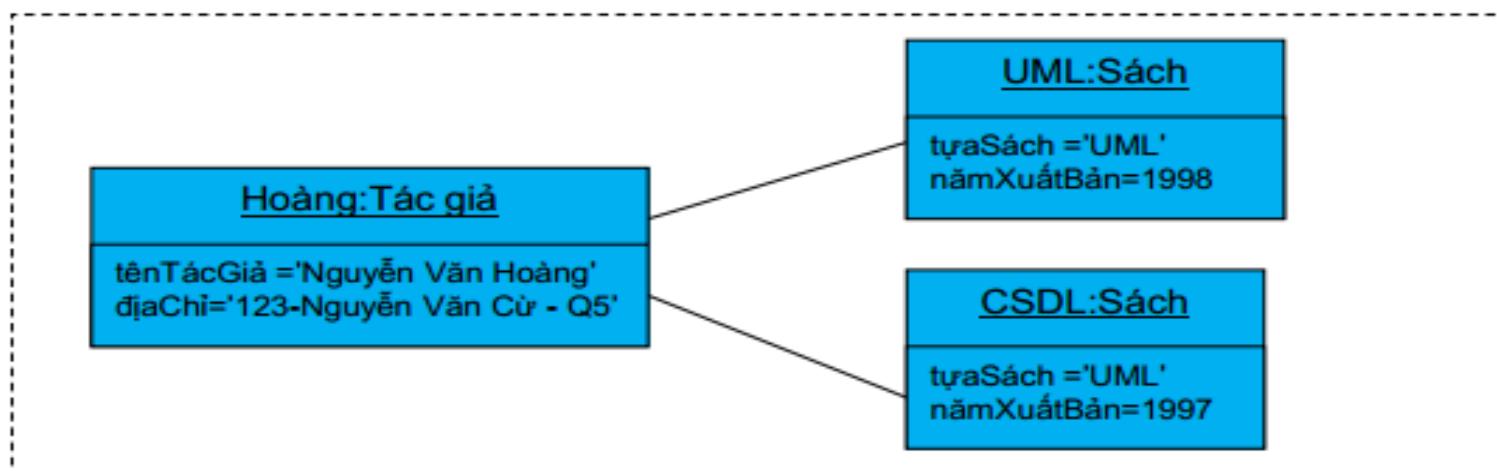
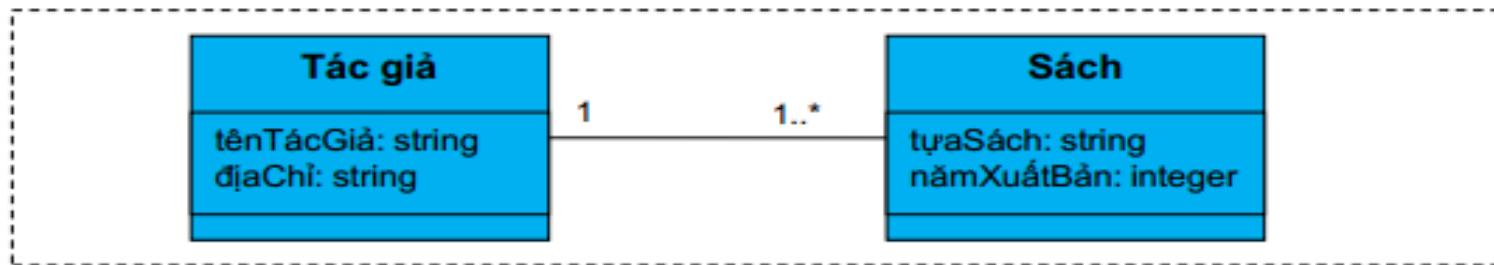
Interaction

State

3. Một số biểu đồ UML cơ bản



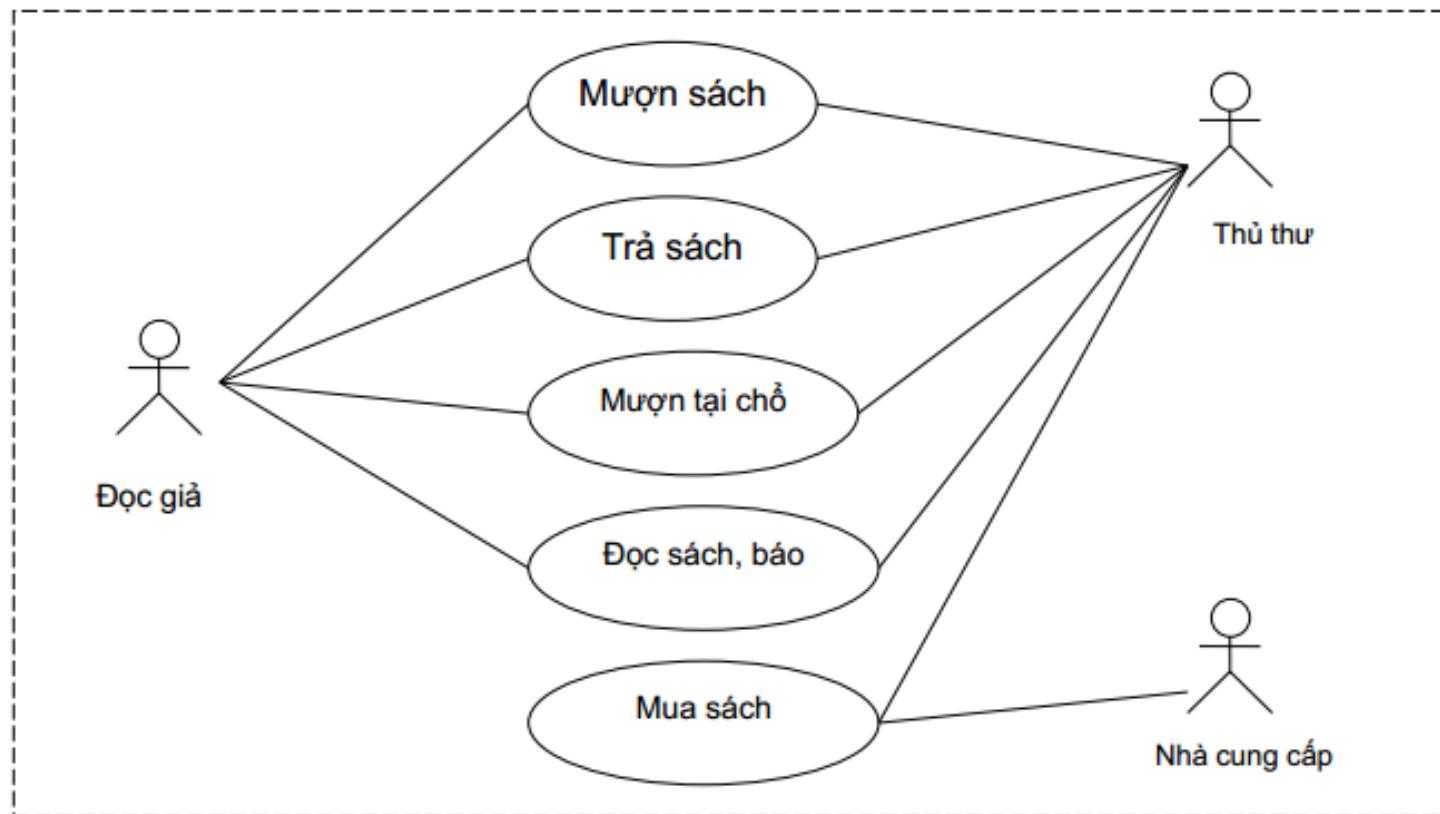
□ Sơ đồ lớp và đối tượng





3. Một số biểu đồ UML cơ bản

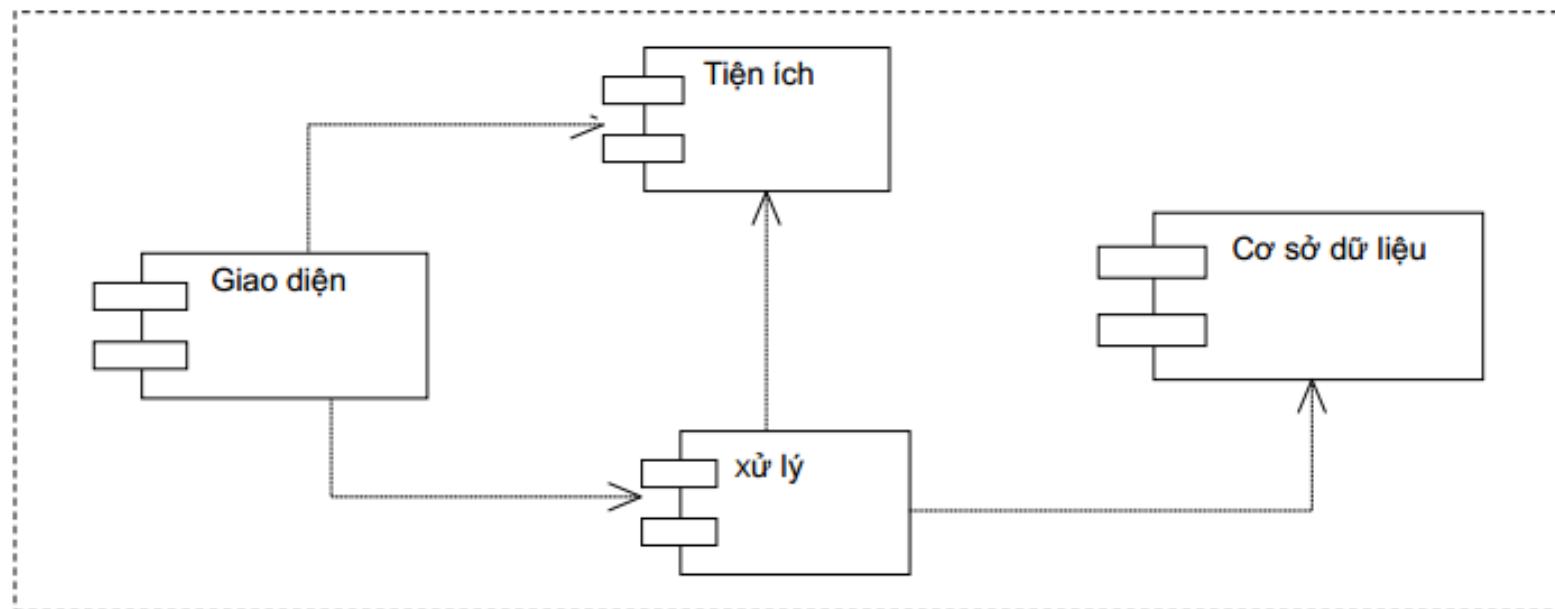
□ Sơ đồ Use case



3. Một số biểu đồ UML cơ bản



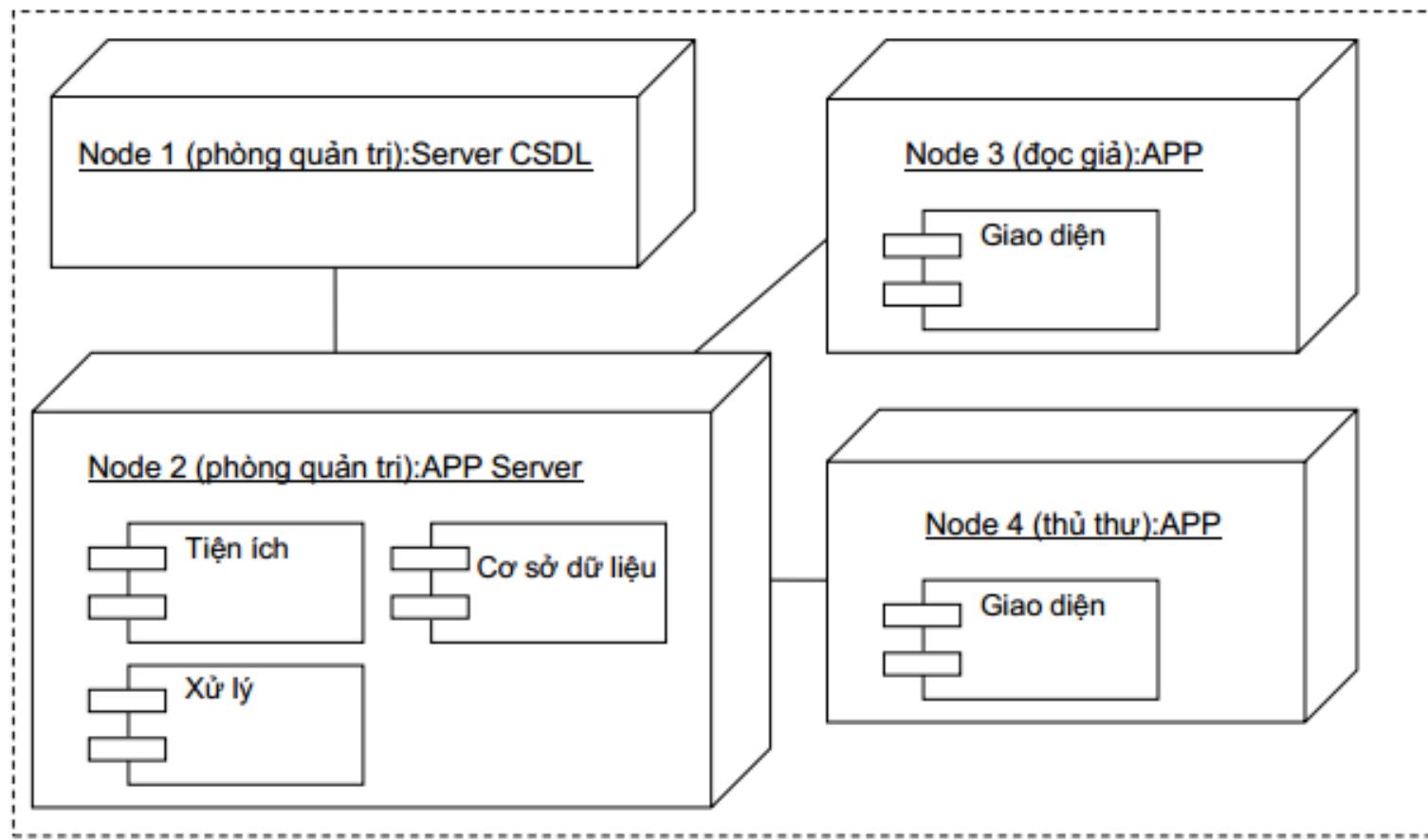
▫ Sơ đồ thành phần



3. Một số biểu đồ UML cơ bản



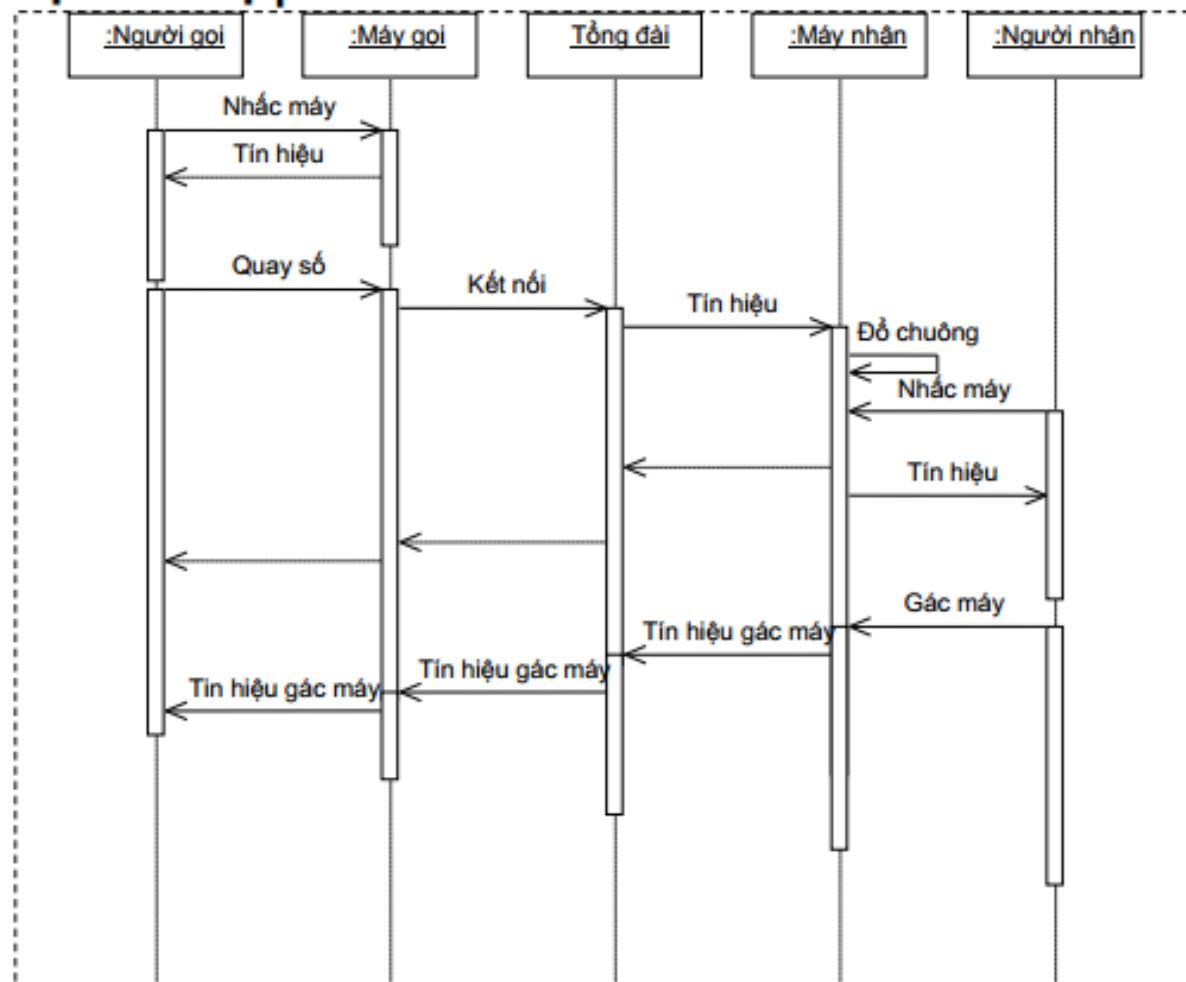
□ Sơ đồ triển khai



3. Một số biểu đồ UML cơ bản



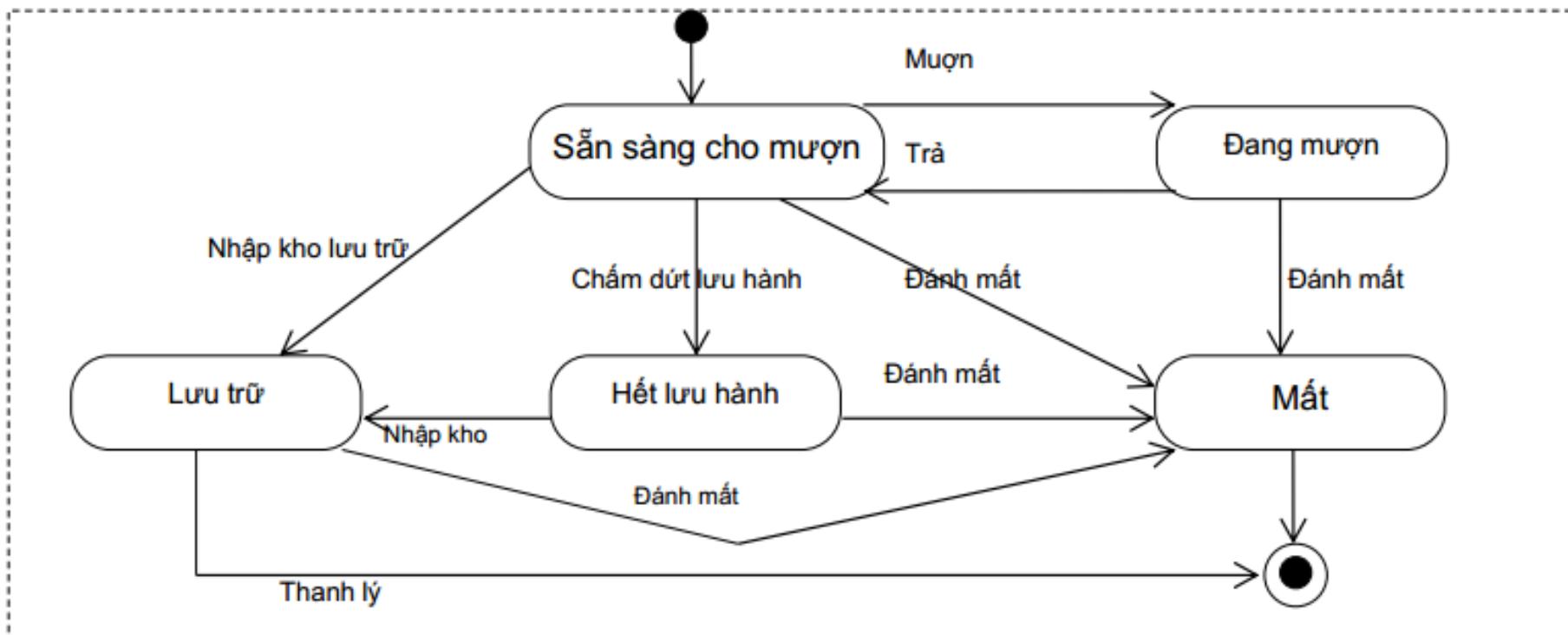
□ Sơ đồ tuần tự và hợp tác



3. Một số biểu đồ UML cơ bản



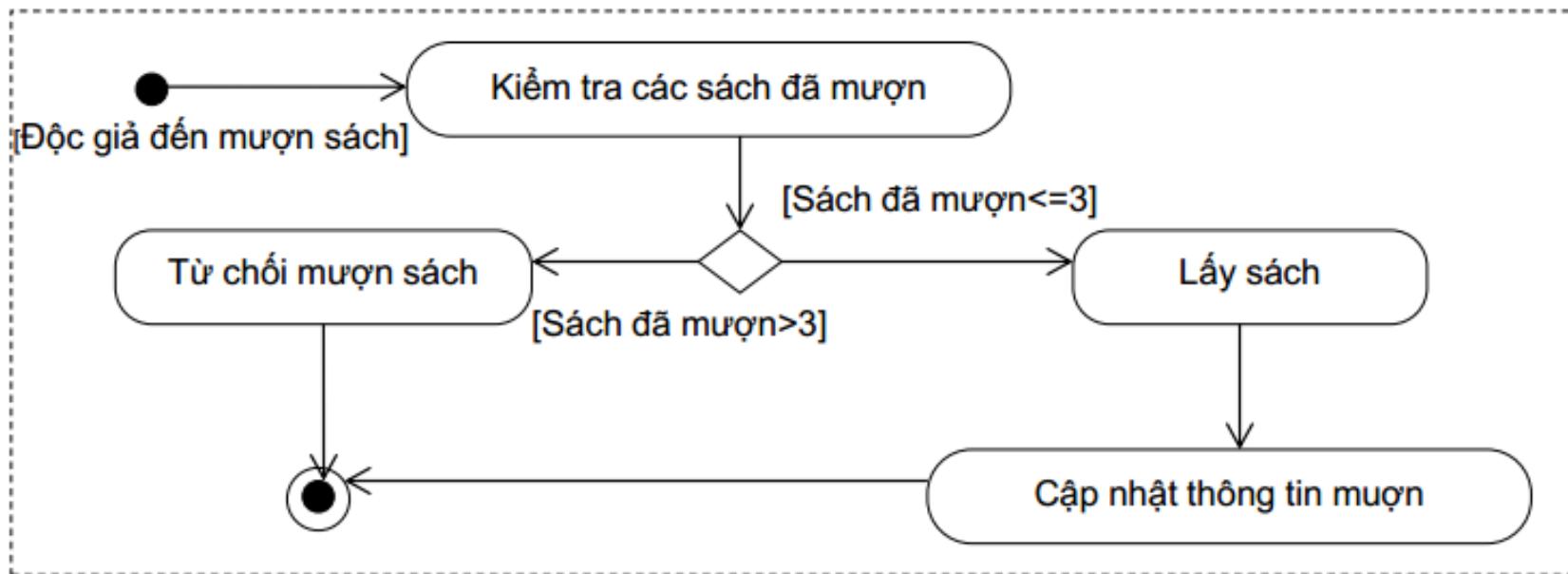
□ Sơ đồ chuyển đổi trạng thái



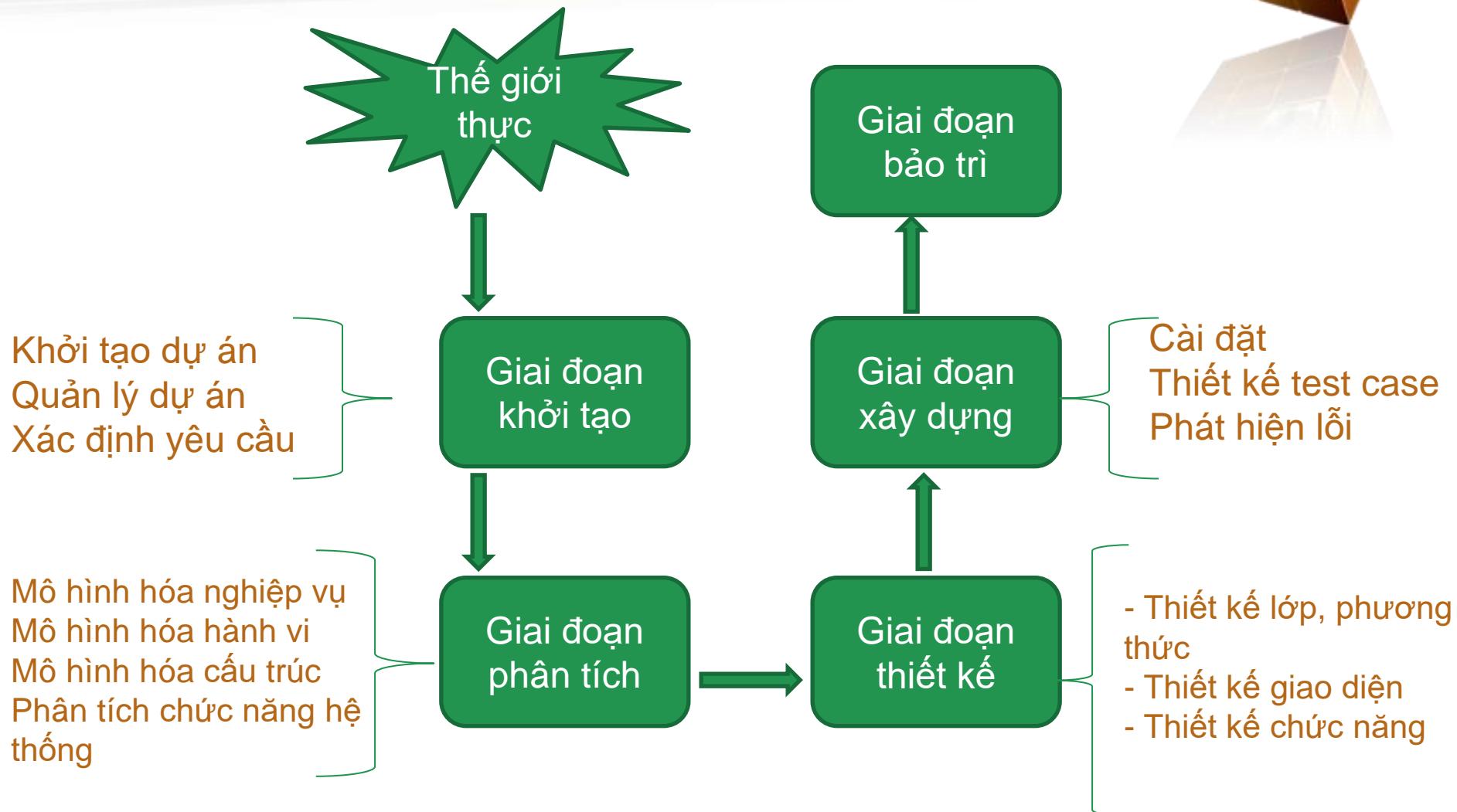
3. Một số biểu đồ UML cơ bản



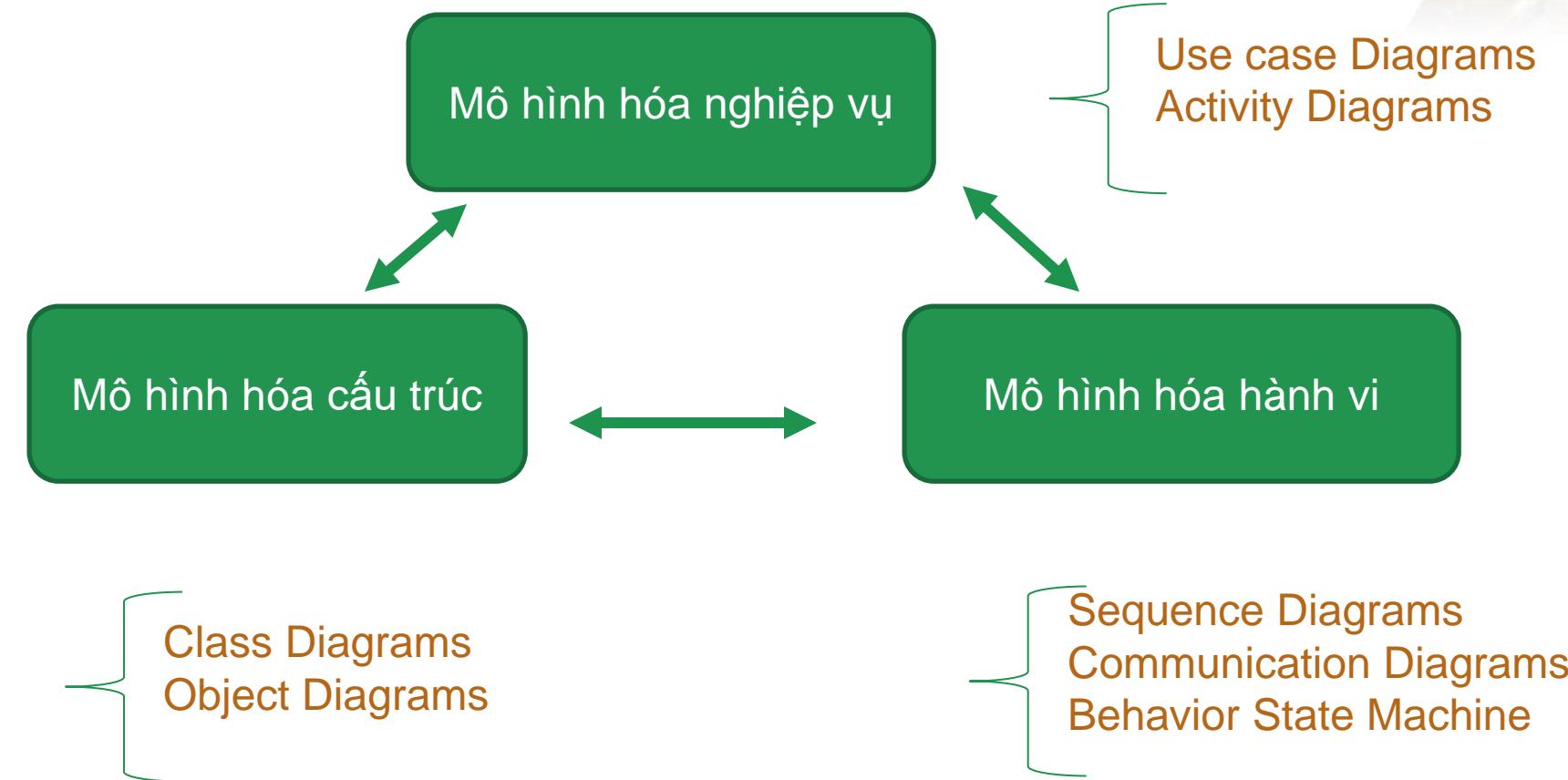
□ Sơ đồ hoạt động



Bức tranh của PTTK HTTT theo HĐT



Bức tranh giai đoạn phân tích



3. Một số biểu đồ UML cơ bản



1

❖ Biểu đồ sử dụng
Use Case Diagram

2

❖ Biểu đồ
Class Diagram

- Mô tả các chức năng của hệ thống dựa trên quan điểm người sử dụng.
- Mô tả sự tương tác giữa người dùng và hệ thống.
- Cho biết hệ thống được sử dụng như thế nào ?

Component

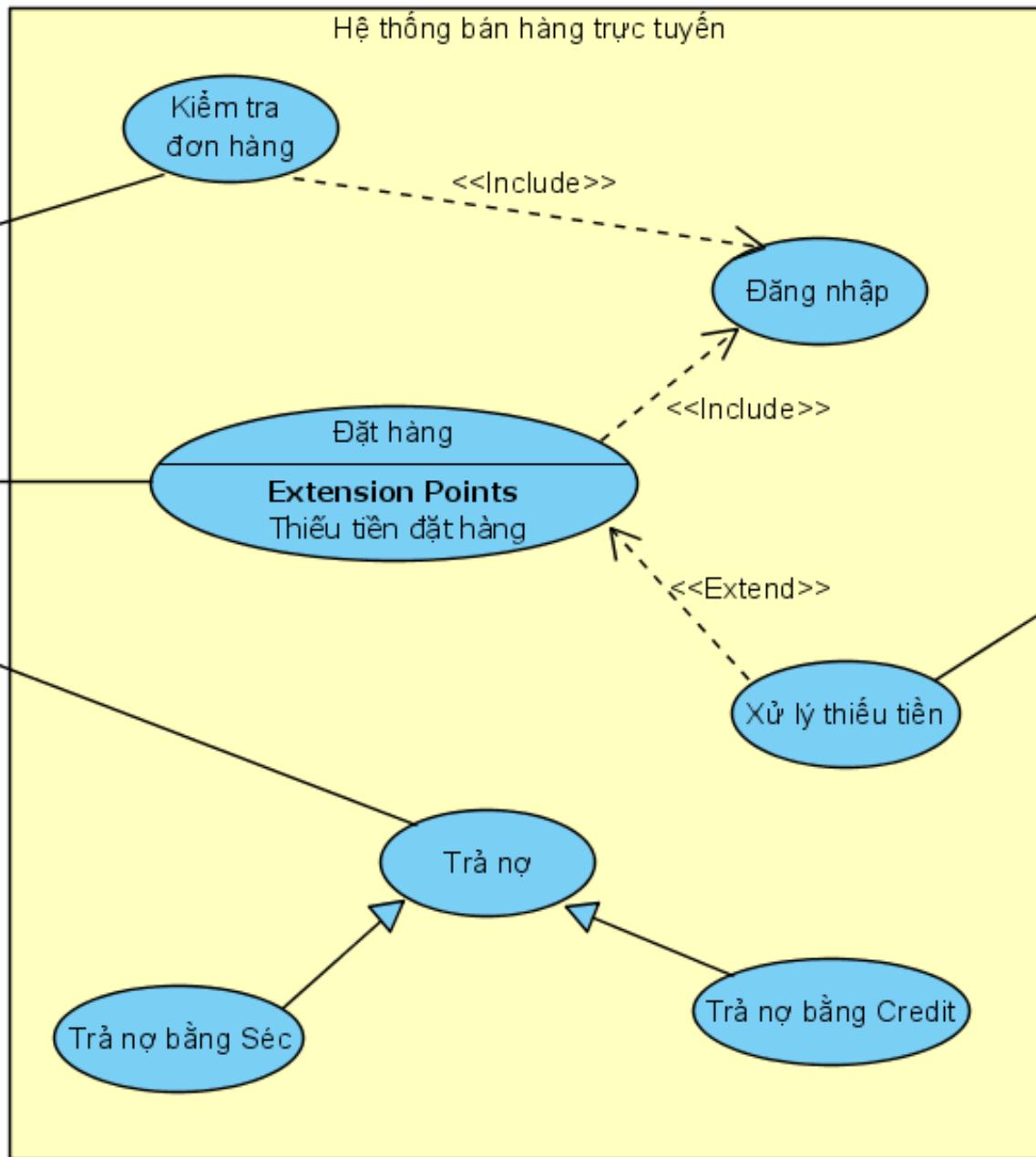
Deployment

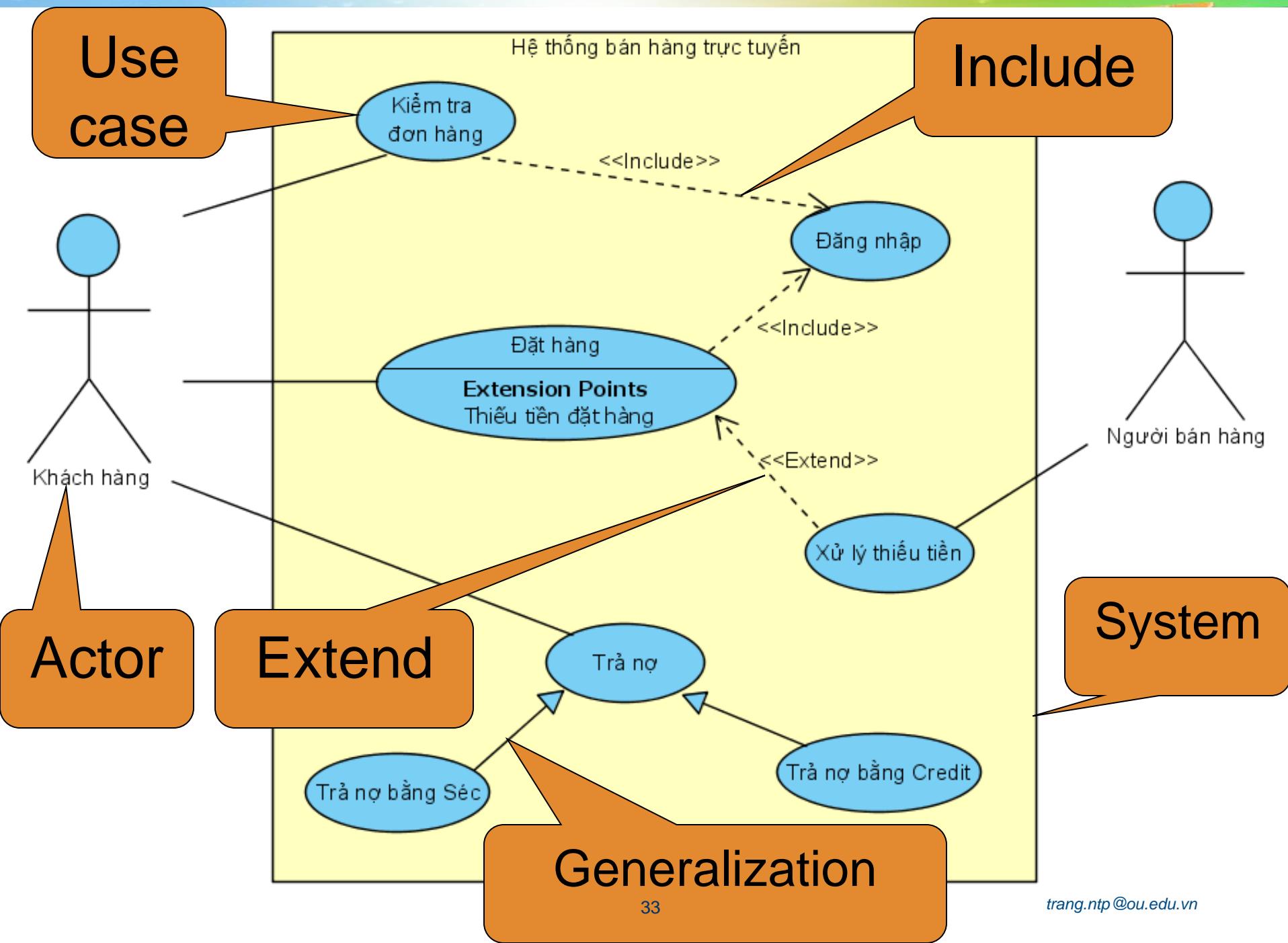
Communication

Collaboration

Timing

State





Chú ý:

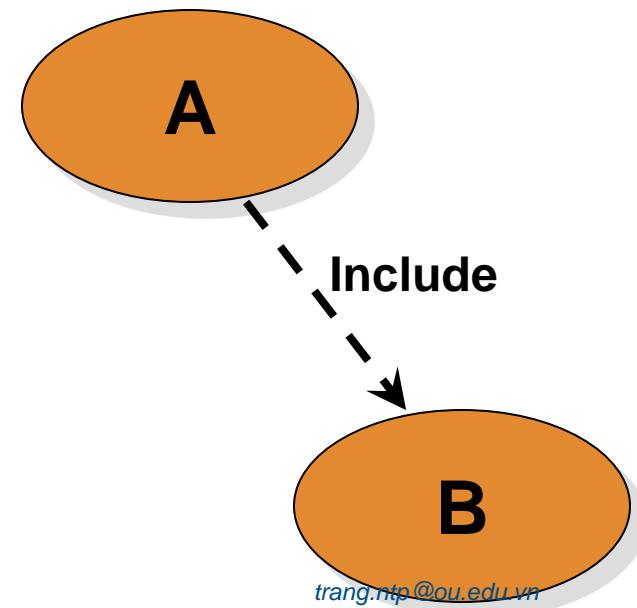


- ❖ Khi nào thì vẽ quan hệ <Include> (bao hàm)
→ **Use case A được gọi là Include B nếu trong xử lý của A có gọi đến B ít nhất 1 lần !**

❖ Minh họa thông qua Code

```
Class B { public void X () { .... } }
```

```
Class A {  
    Public void Y () {  
        B objB = new B(); objB.X (); ...  
    }  
}
```



Chú ý:

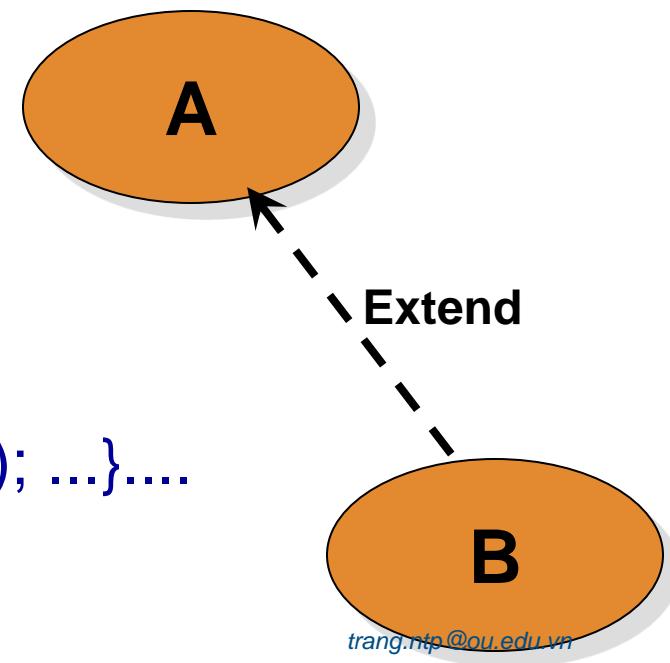


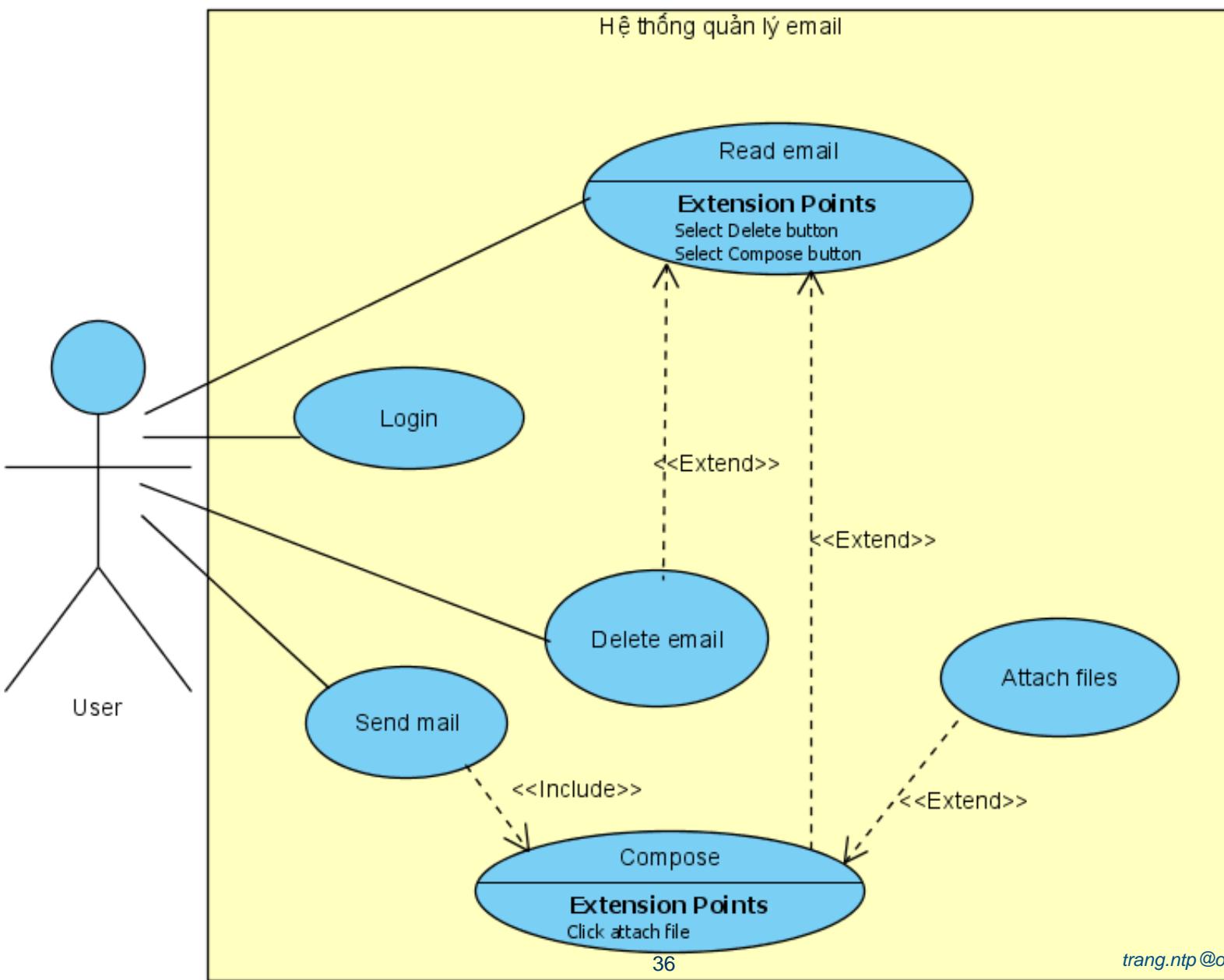
- ❖ Khi nào thì vẽ quan hệ <Extend> (mở rộng)
→ **Use case B được gọi là Exntend A nếu use case B được gọi bởi A nếu thỏa mãn điều kiện nào đó.**

❖ Minh họa thông qua Code

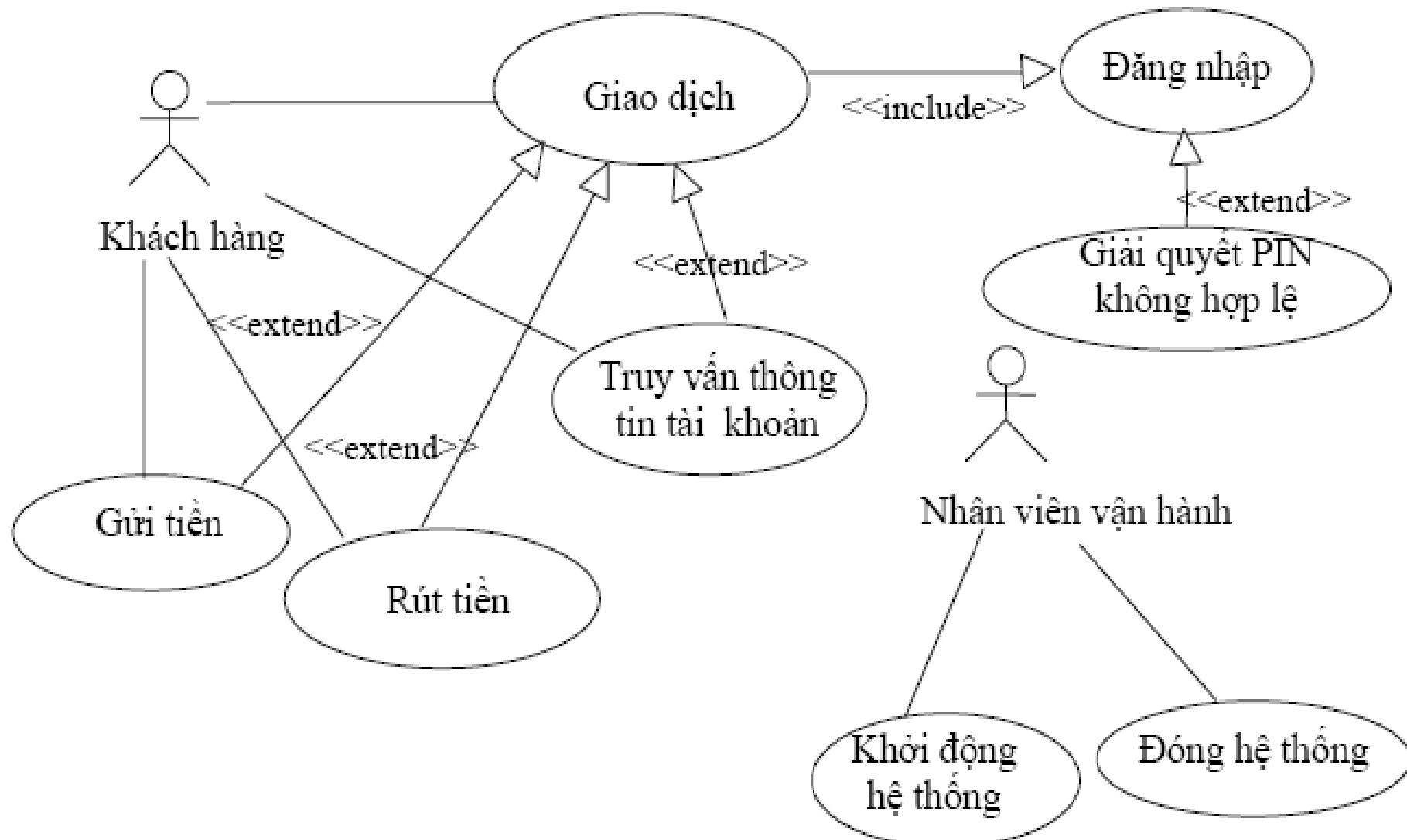
```
Class B { public void InẤn () { .... } }
```

```
Class A {  
    Public void XemDSSV () {  
        ... If (Click_Nút_InẤn)  
        { B objB = new B(); objB.InẤn(); .... }  
    }  
}
```

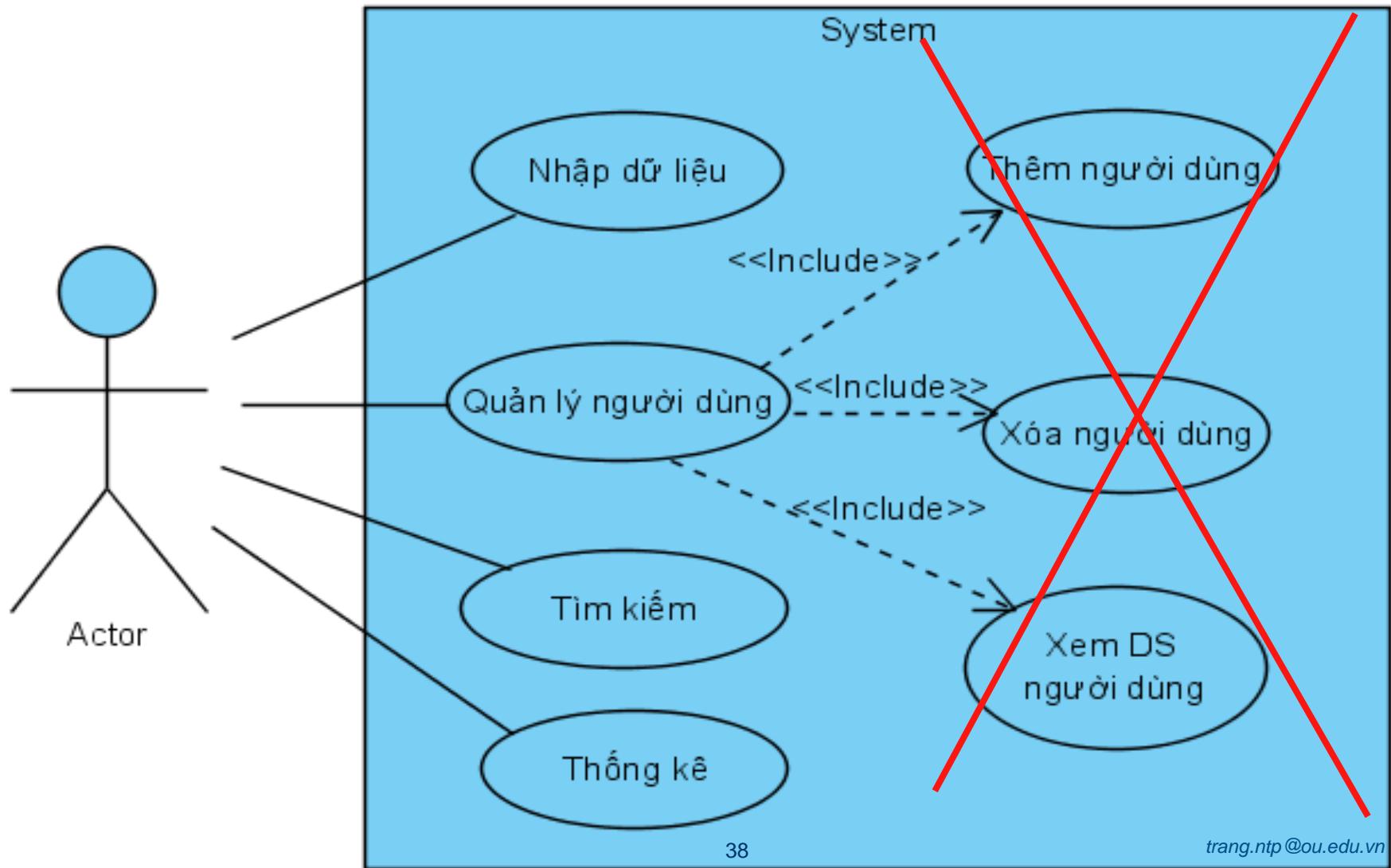




Một số hình vẽ đúng



Một số hình vẽ sai



Vẽ quan hệ tổng quát hóa (thừa kế)



- ❖ Khi nào thì vẽ quan hệ <Generalization> (tổng quát hóa)
→ **Use case A được gọi là Generalization B nếu B là một trường hợp riêng của A !**
- ❖ Nếu A Generalization B thì code có dạng như thế nào

Class A {

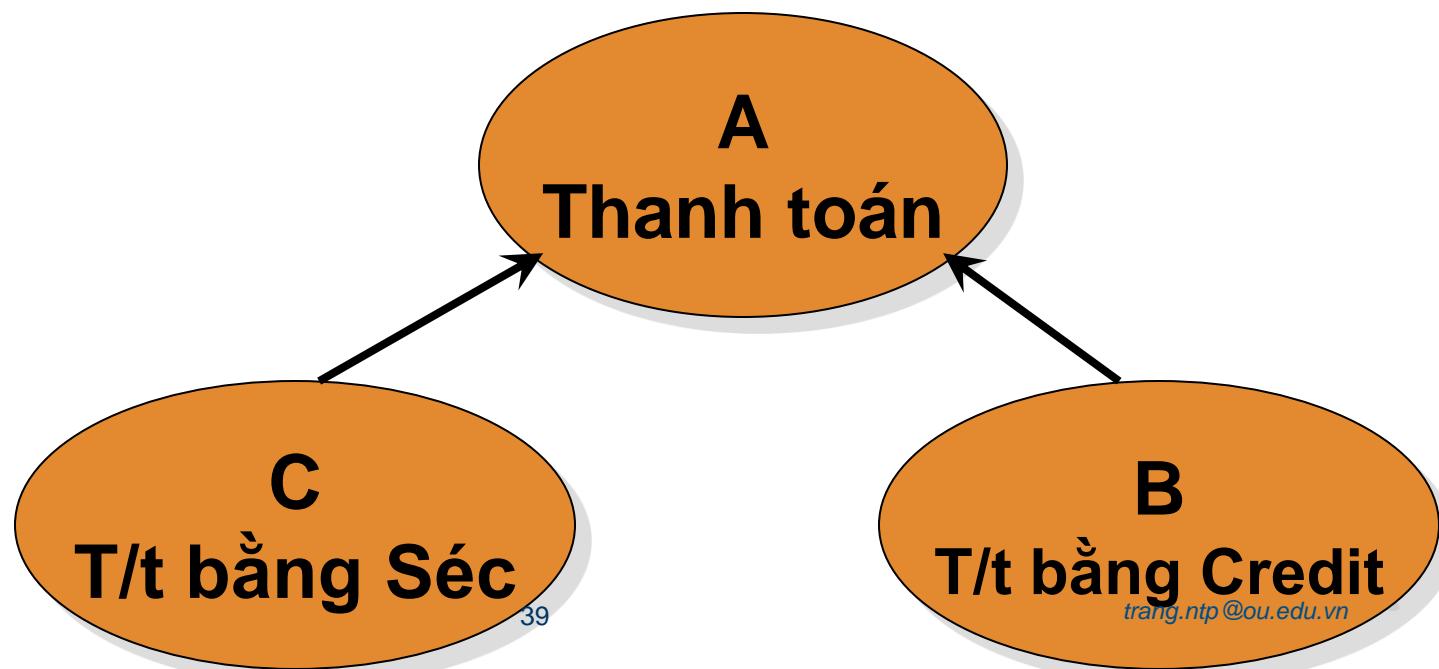
.....

}

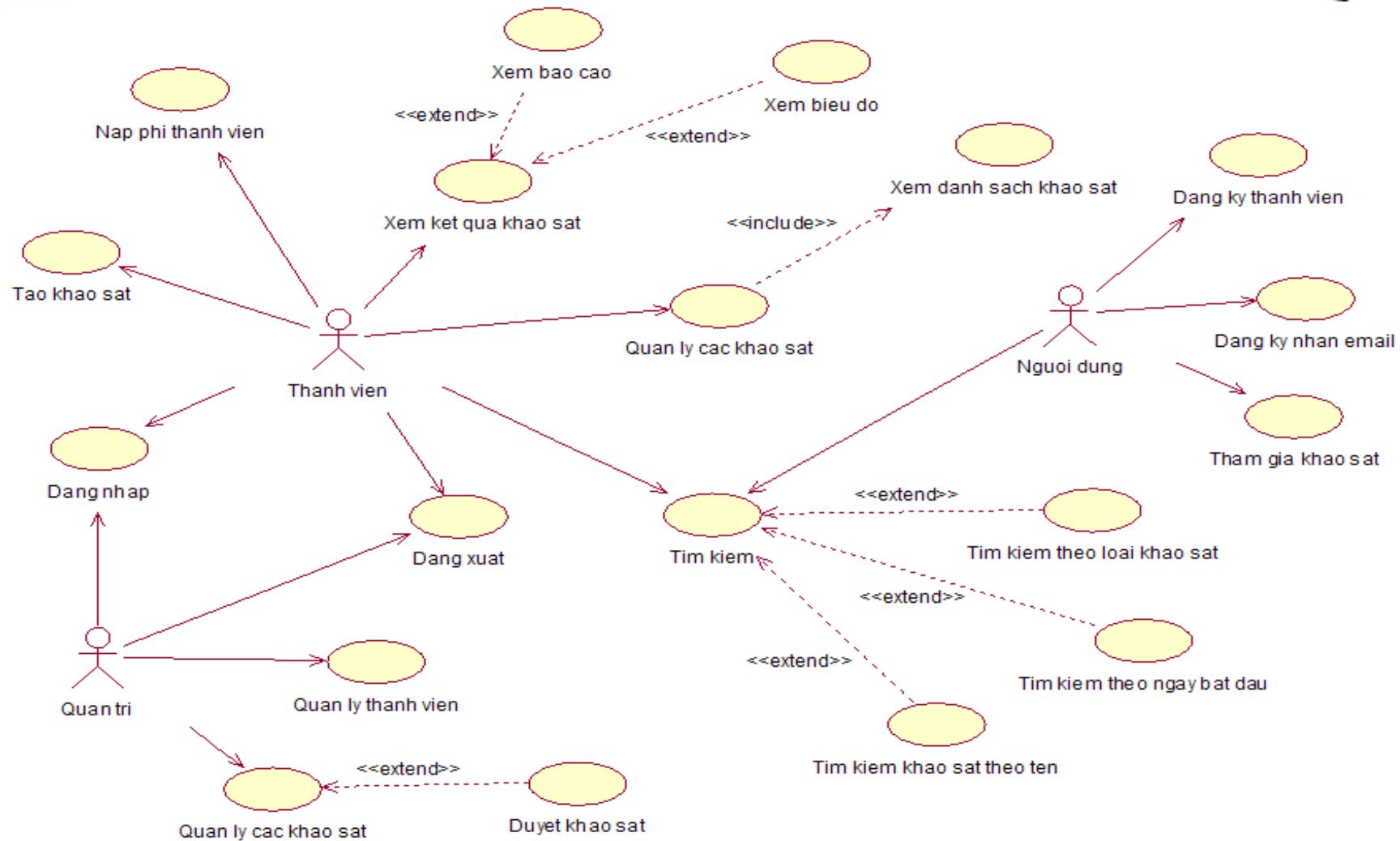
Class B : A

{

}



Đọc biểu đồ use case sau



Name	Tạo khảo sát
Description	Thành viên tạo khảo sát
Actor	Thành viên
Pre conditions	<ul style="list-style-type: none"> Đăng nhập vào tài khoản thành viên Hiển thị trang tạo khảo sát
Post conditions	Thông báo kết quả tạo khảo sát
Flow of events	<ol style="list-style-type: none"> Hệ thống lấy thông tin tài khoản người dùng Hệ thống kiểm tra tài khoản. Hệ thống hiển thị tùy chọn tạo khảo sát có tính phí hay không. Người dùng nhập tên khảo sát Người dùng nhập giới thiệu khảo sát Chọn ngày bắt đầu khảo sát Chọn ngày kết thúc khảo sát Nhấn nút tạo khảo sát <ol style="list-style-type: none"> Hệ thống lưu nội dung khảo sát xuống cơ sở dữ liệu Không lưu được: A1 Thông báo kết quả lên trang web
Alternative flow	<p>A1: Thông báo không tạo được khảo sát. Trở lại trang tạo khảo sát</p> <p>A2: Thông báo không lưu được khảo sát Trở lại trang tạo khảo sát</p>



TRƯỜNG ĐẠI HỌC MỞ
TP HỒ CHÍ MINH

Nghỉ giải lao đến 9:15

BÀI GIẢNG
**PHÂN TÍCH THIẾT KẾ HỆ THỐNG
THÔNG TIN**

**Chương 5: Phân tích thiết kế
hướng đối tượng**

ThS. Nguyễn Thị Phương Trang
trang.ntp@ou.edu.vn



3. Một số biểu đồ UML cơ bản

1

❖ Biểu đồ ca
sử dụng
**Use Case
Diagram**

- Mô tả các luồng công việc, qui trình nghiệp vụ.
- Mô tả các hoạt động của UC
- Hỗ trợ việc mô tả các xử lý song song.

4

Biểu đồ
Hoạt động
**Activity
Diagram**

Component

Deployment

Communication

Collaboration

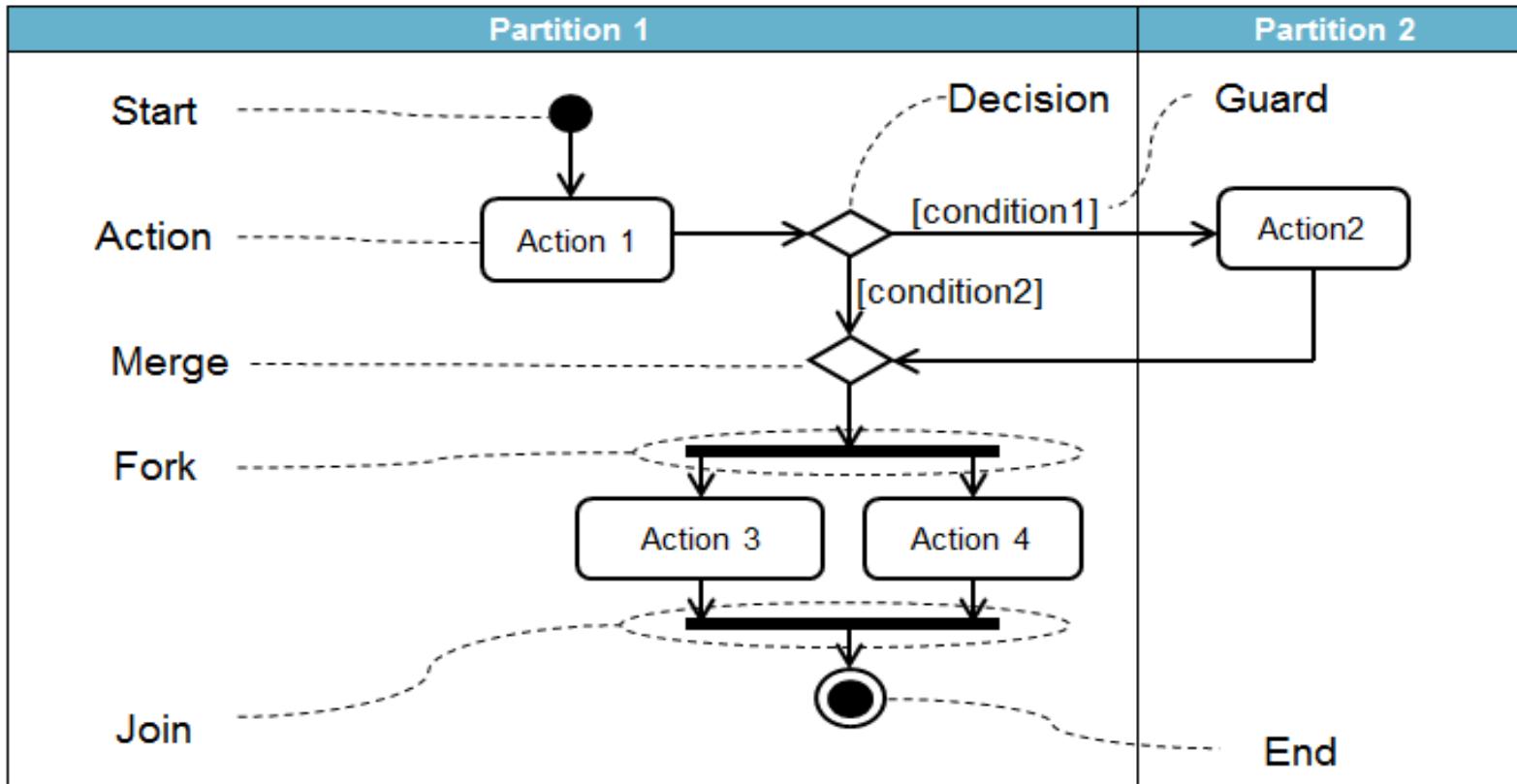
Timing

State

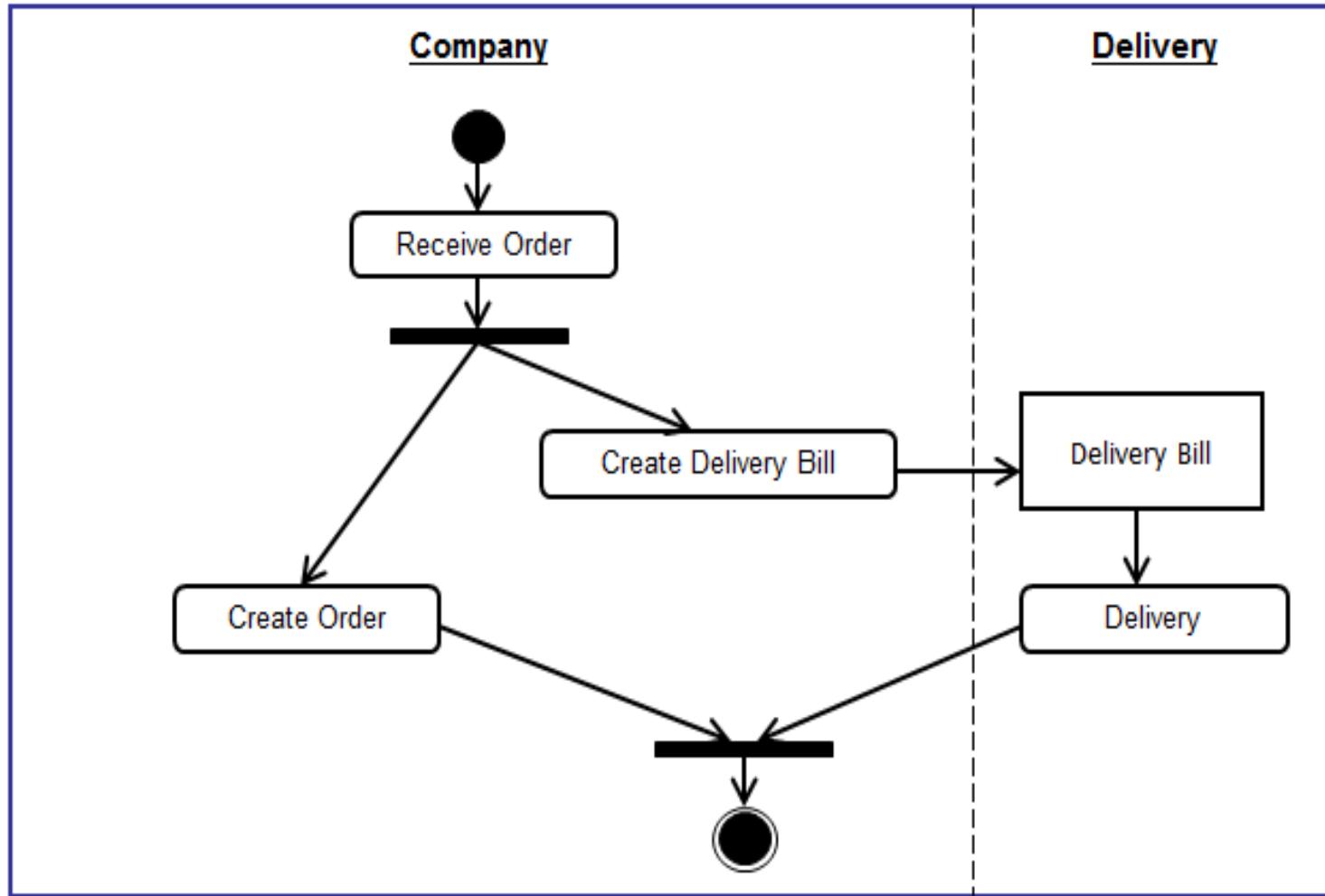
BIỂU ĐỒ HOẠT ĐỘNG – ACTIVITY DIAGRAM



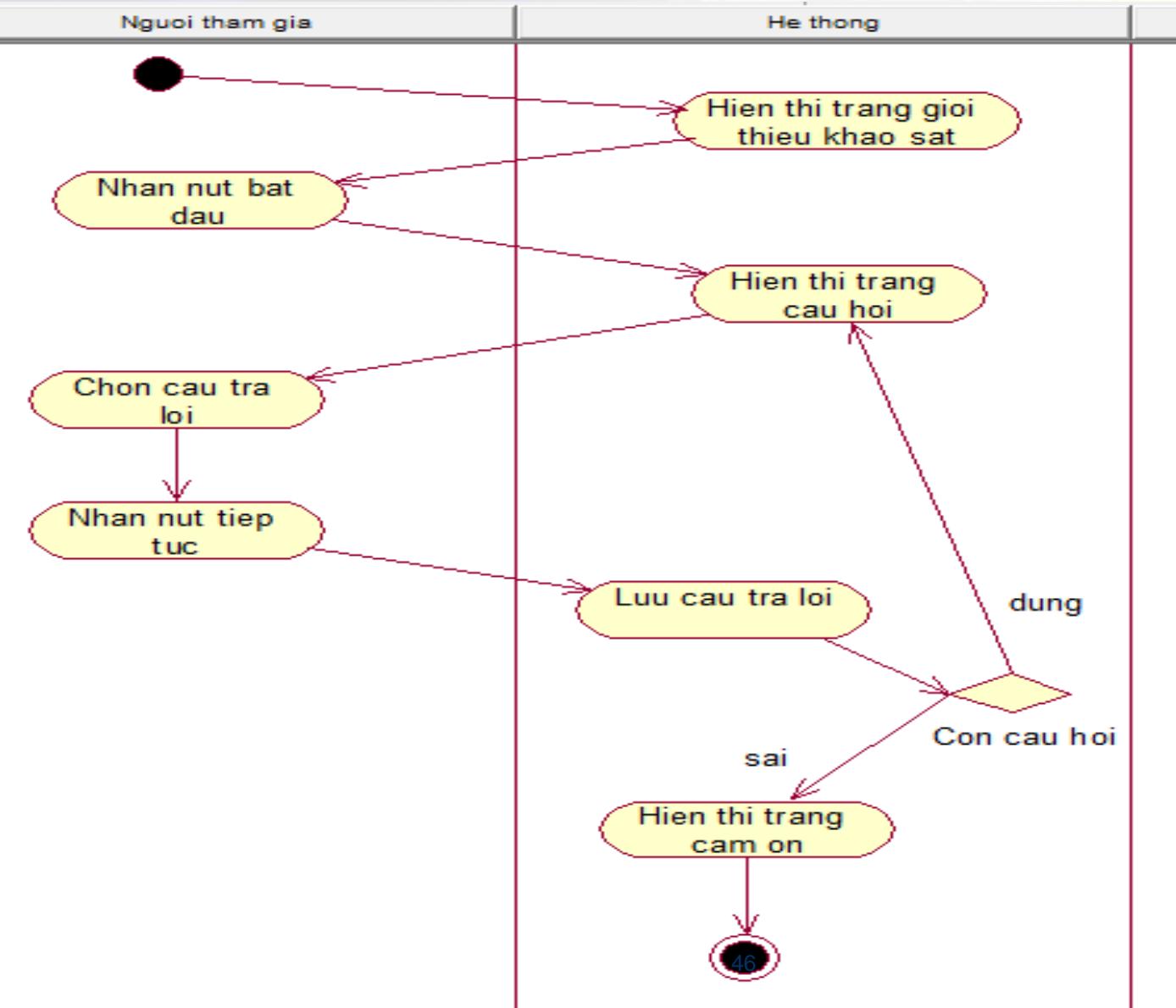
Các ký hiệu



Mô tả nghiệp vụ hệ thống



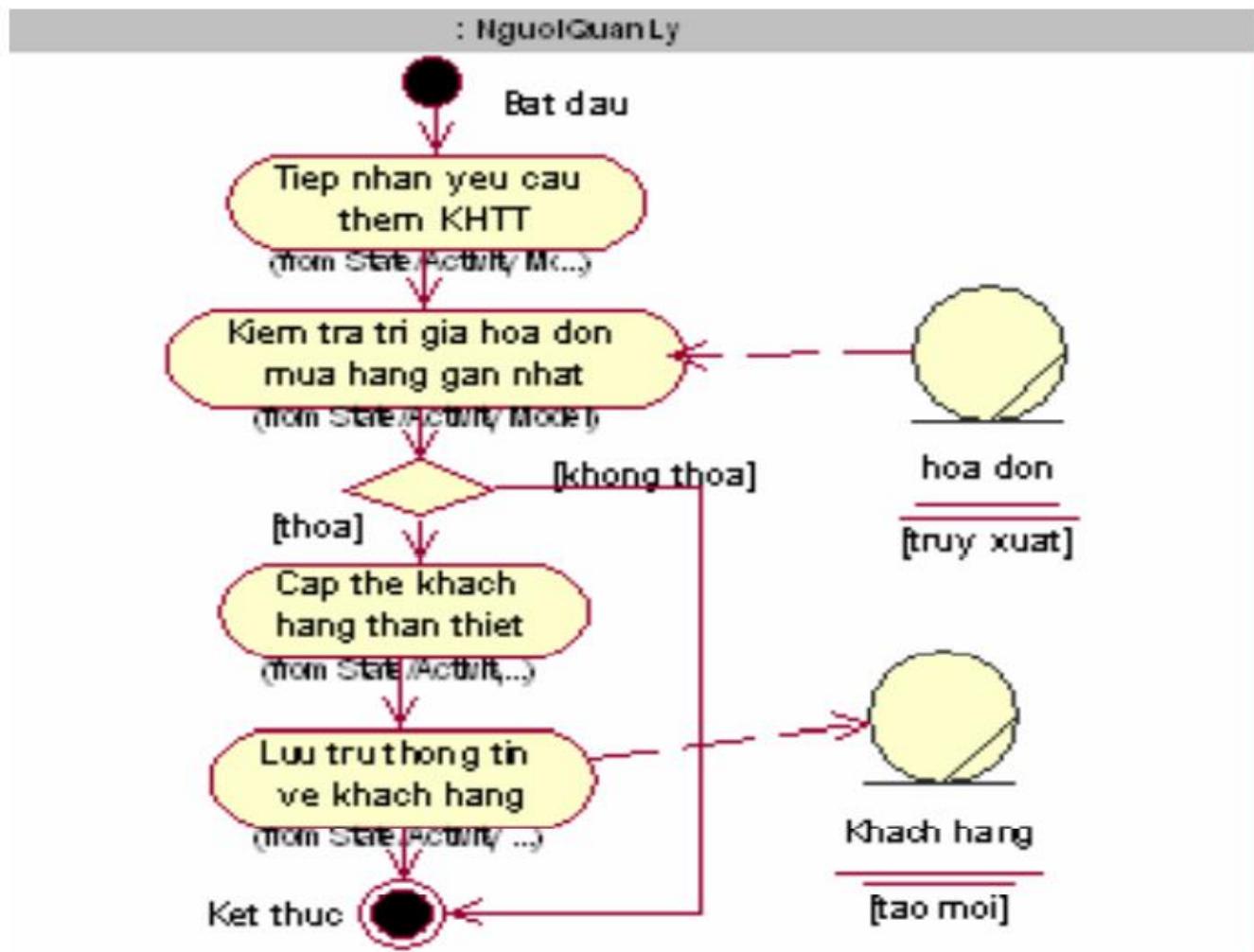
Mô tả hoạt động của Use Case



AD đặc tả use case: QLKH



Lược đồ Activity: QLKH



3. Một số biểu đồ UML cơ bản



1

❖ Biểu đồ sử dụng
Use Case Diagram

2

❖ **Biểu đồ Lớp Class Diagram**

- Là biểu đồ quan trọng nhất
 - Mô tả các đối tượng và mối quan hệ của chúng trong hệ thống.
 - Mô tả các thuộc tính và các hành vi (Behavior) của đối tượng.
 - Có biểu đồ lớp mức phân tích và mức cài đặt.

tích và mức cài đặt.

Component

Deployment

Communication

Collaboration

Timing

State

Hai dạng lớp: phân tích và thiết kế



Analysis

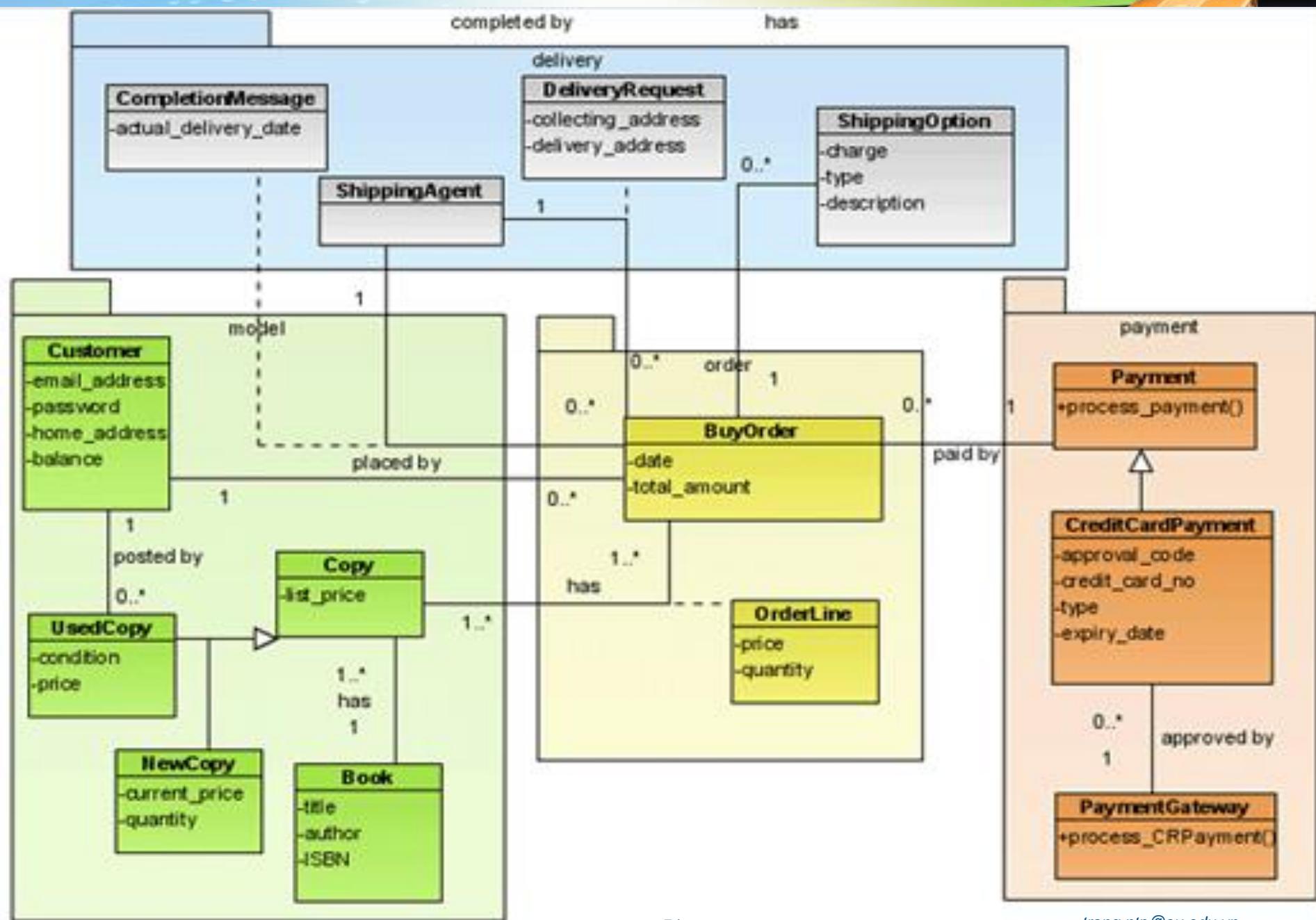
Order
Placement Date
Delivery Date
Order Number
Calculate Total
Calculate Taxes

Design

Order
<ul style="list-style-type: none">- deliveryDate: Date- orderNumber: int- placementDate: Date- taxes: Currency- total: Currency <pre># calculateTaxes(Country, State): Currency # calculateTotal(): Currency getTaxEngine() {visibility=implementation}</pre>

Bỏ qua các chi tiết
không cần thiết

Phải đầy đủ & chi
tiết các thành phần

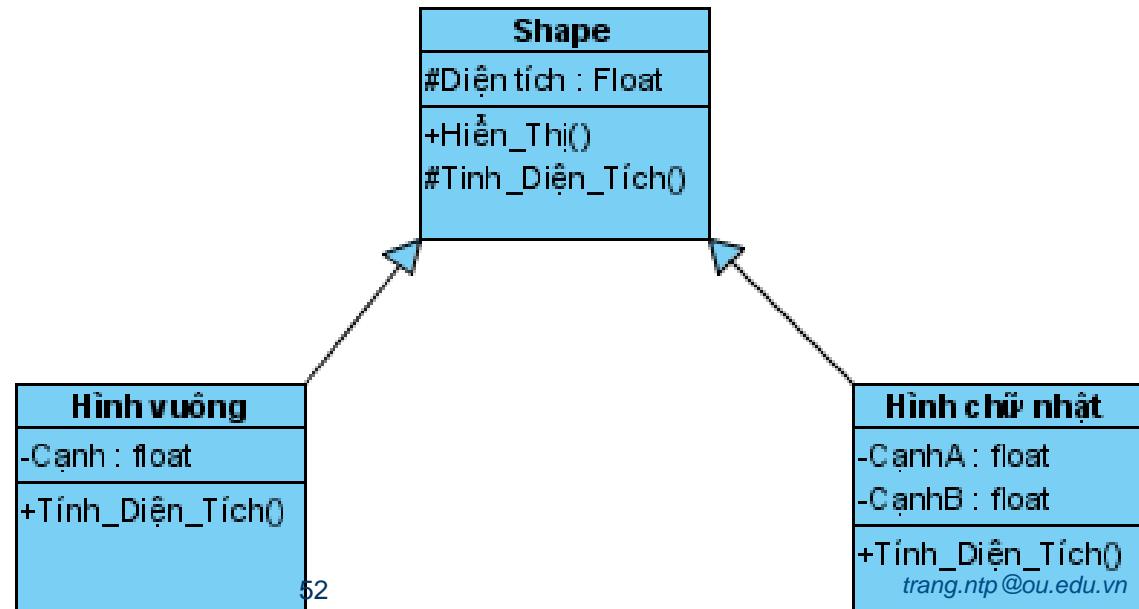


Các quan hệ trong biểu đồ lớp



- ❖ Quan hệ **Generalization**: Thể hiện rằng một lớp A kế thừa từ một lớp B (Hay A là trường hợp riêng của B; B là tổng quát của A)
- ❖ Gọi là quan hệ **Là một (Is a)**

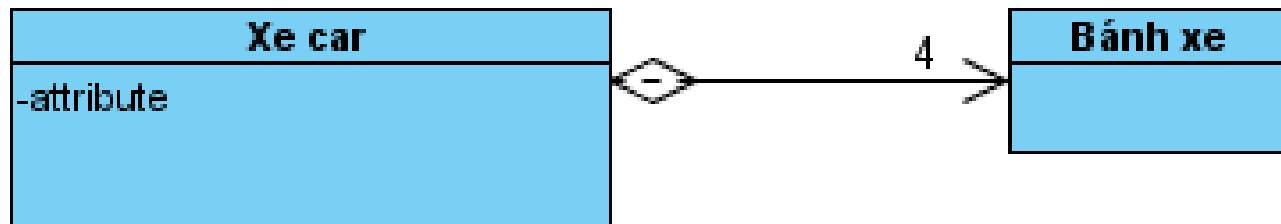
- ❖ Thể hiện:



Các quan hệ trong biểu đồ lớp (2)



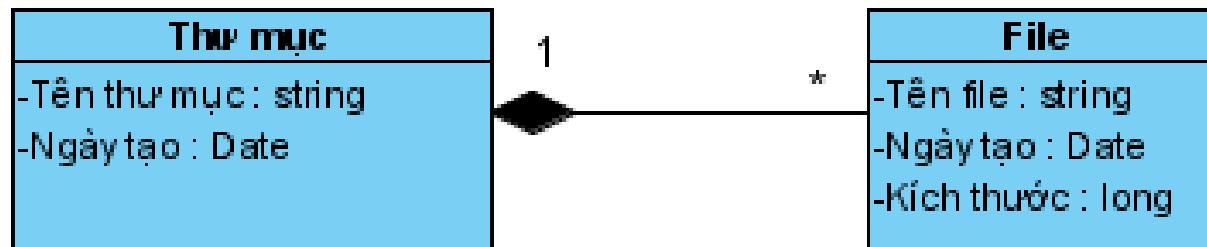
- ❖ **Quan hệ Aggregation:** Thể hiện rằng một lớp A nào đó bao gồm lớp B. Lớp B này có thể tồn tại độc lập mà không cần lớp A.
- ❖ Còn gọi là mối quan hệ: **Có một (Has a)**
- ❖ Thể hiện:



Các quan hệ trong biểu đồ lớp (3)



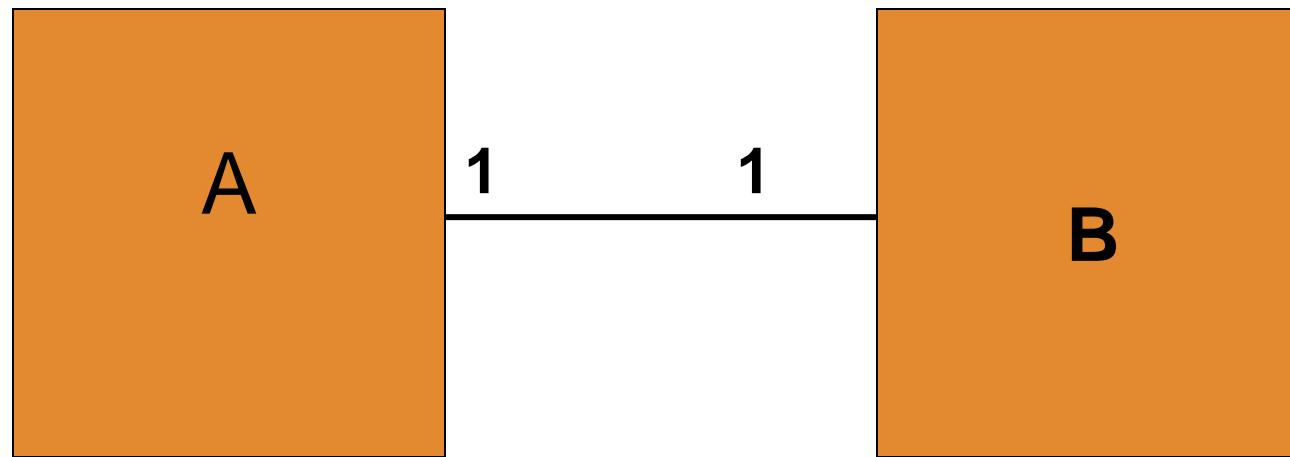
- ❖ **Quan hệ Composition:** thể hiện rằng một lớp A bao hàm lớp B. Nhưng lớp B không thể tồn tại độc lập (Tức không thuộc lớp nào). Tức là, nếu có B thì phải suy ra được A.
- ❖ **Thể hiện:**



Ứng số (Multiplicity)



- ❖ Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?

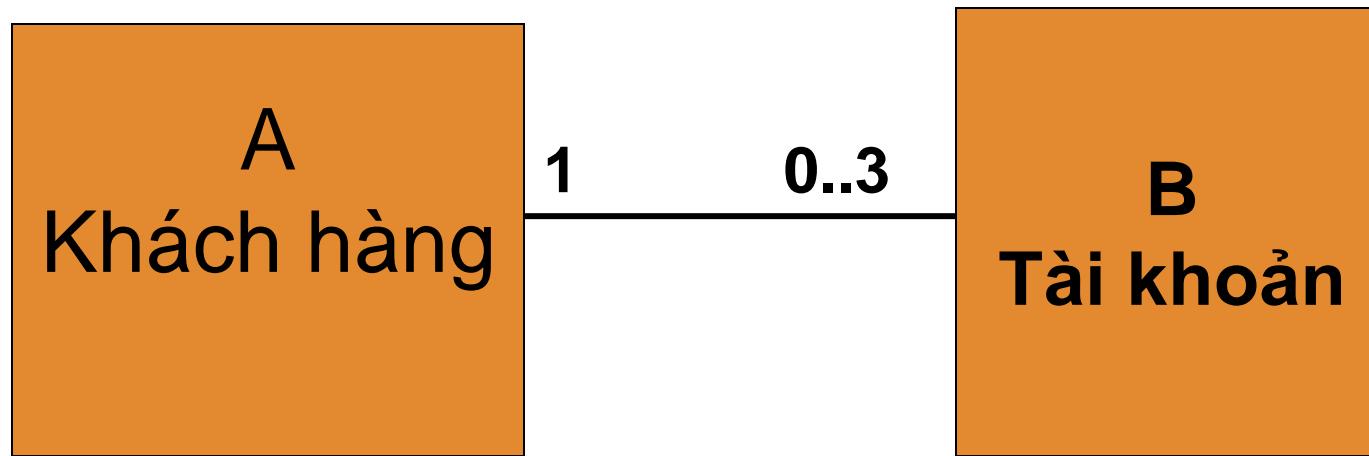


Một phần tử lớp A có 1 phần tử lớp B

Ứng số (Multiplicity)



- ❖ Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



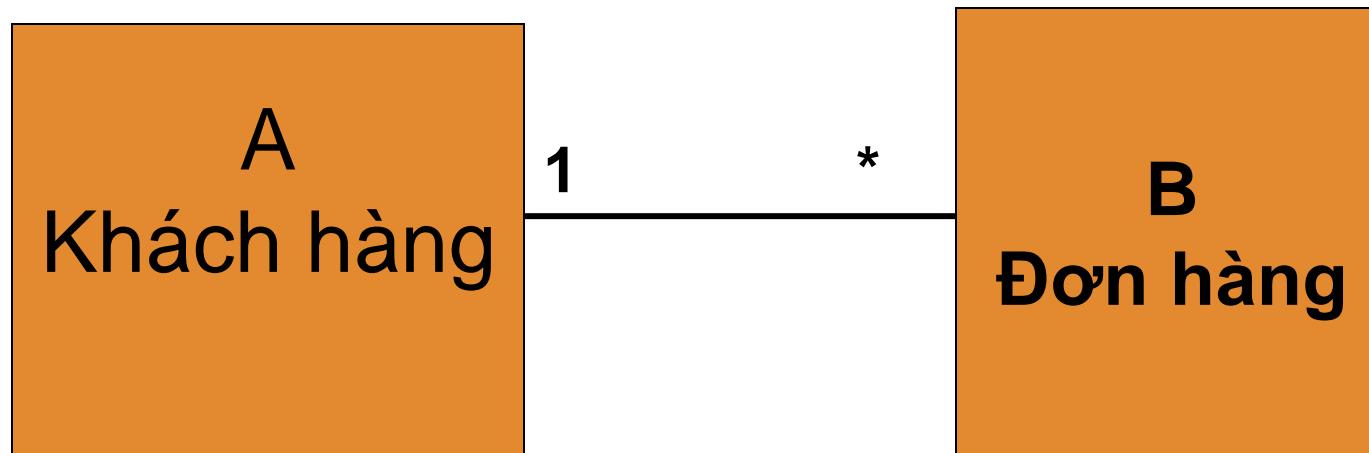
Một phần tử lớp A có tối đa 3 phần tử lớp B

Mỗi phần tử lớp B có đúng 1 phần tử lớp A

Ứng số (Multiplicity)



- ❖ Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



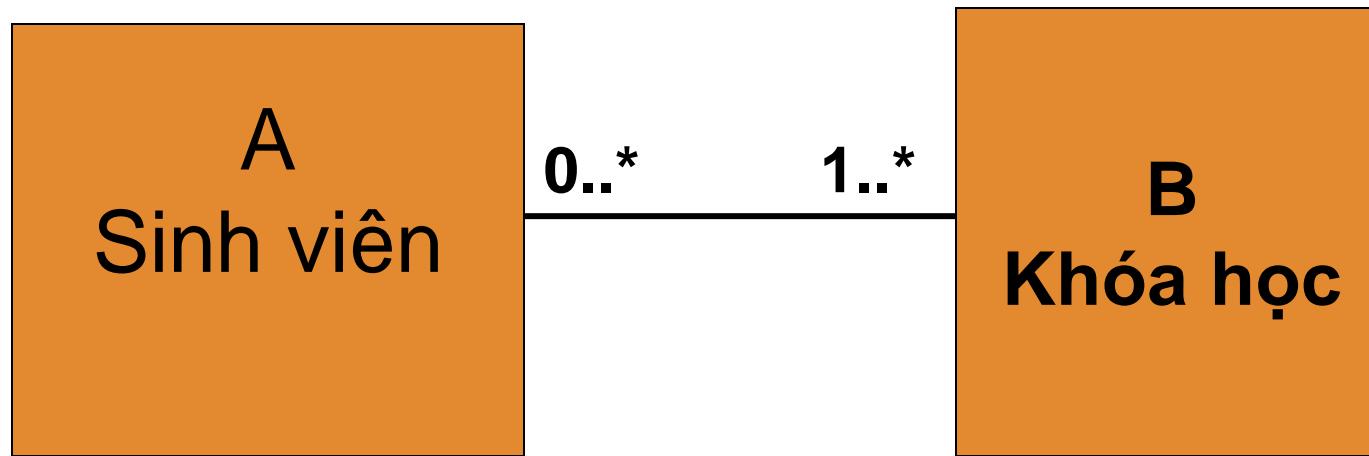
Một phần tử lớp A có nhiều phần tử lớp B

Mỗi phần tử lớp B có đúng 1 phần tử lớp A

Ứng số (Multiplicity)



- ❖ Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



Mỗi sinh viên tham gia ít nhất 1 khóa học

Mỗi khóa học có thể có 0 hoặc nhiều sv tham gia

3. Một số biểu đồ UML cơ bản



- Mô tả sự tương tác của các đối tượng theo trình tự về thời gian.
- Có sự liên kết chặt chẽ với biểu đồ lớp.
- Mỗi biểu đồ tuần tự mô tả một tình huống xử lý.

3

Biểu đồ
Tuần tự
**Sequence
Diagram**

4

Biểu đồ
Hoạt động
**Activity
Diagram**

Component

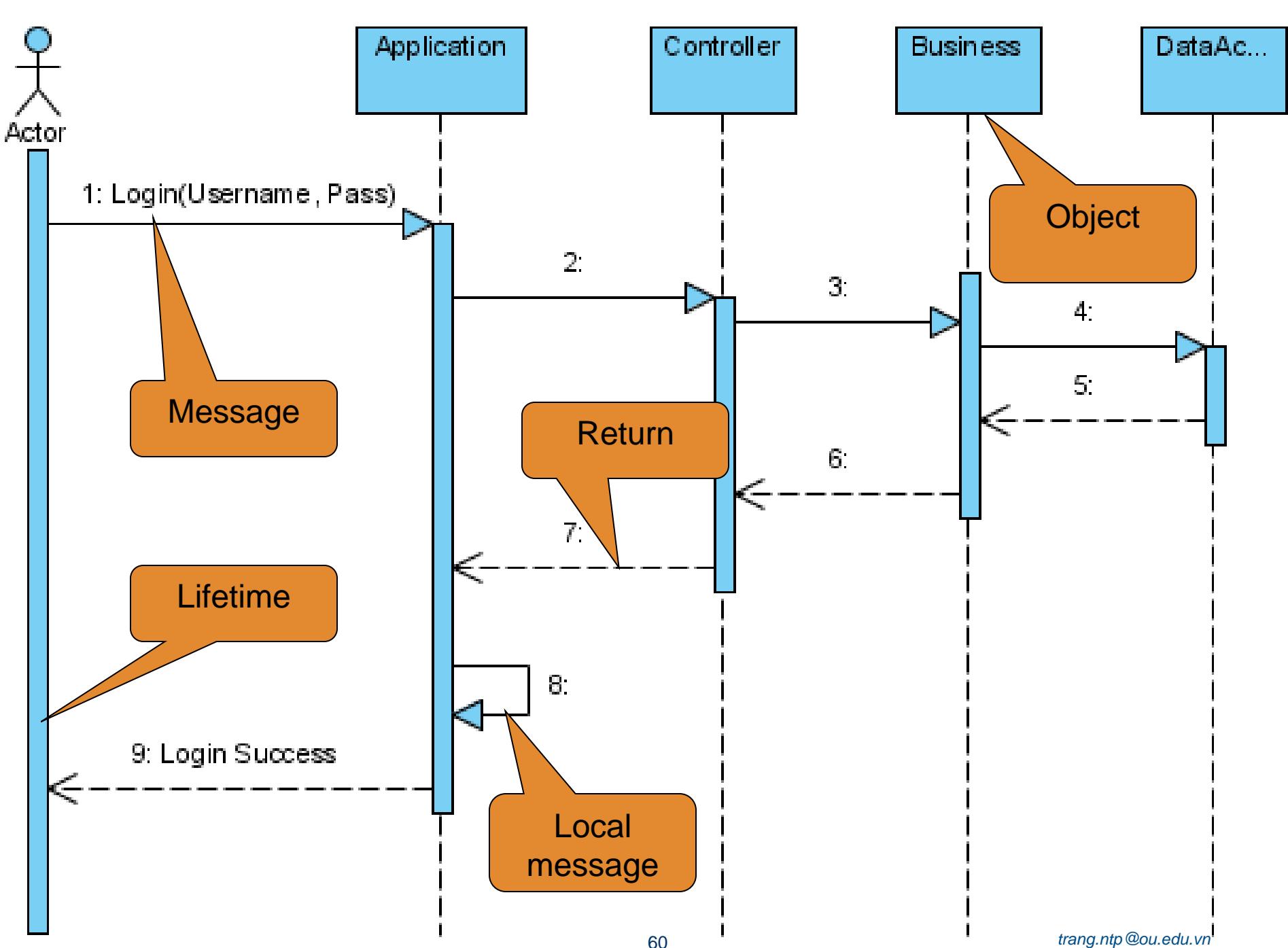
Deployment

Communication

Collaboration

Timing

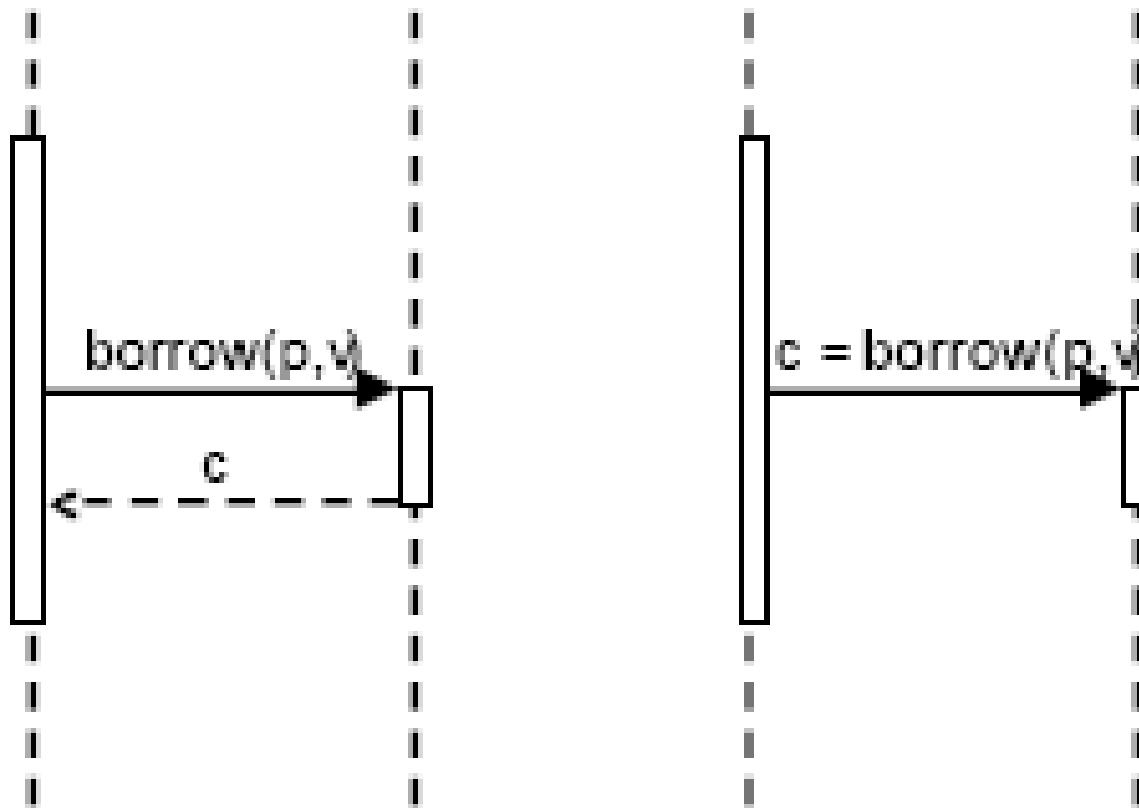
State



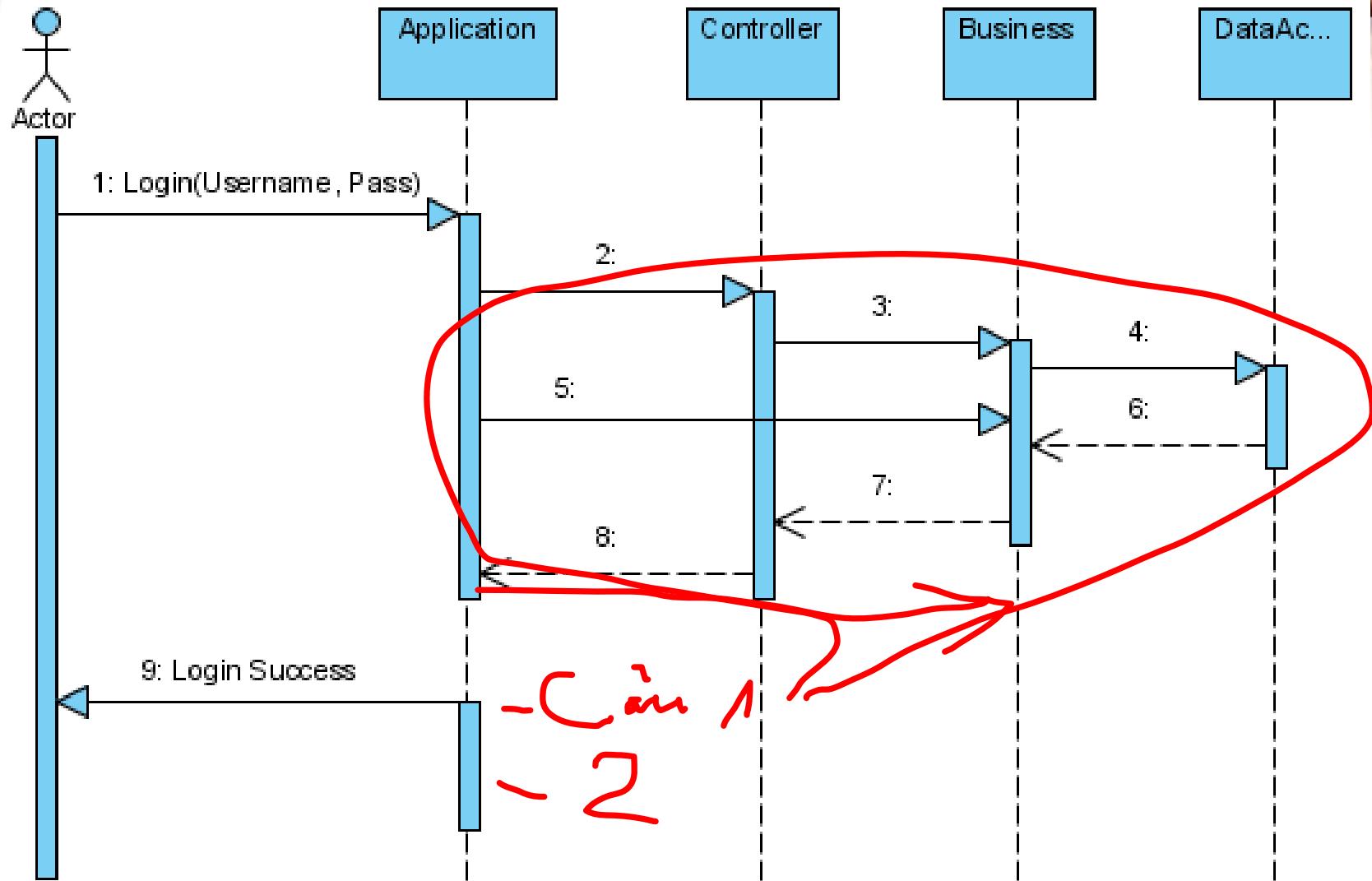
Vẽ biểu đồ tuần tự



- ❖ Chú ý: có thể vẽ một trong 2 dạng



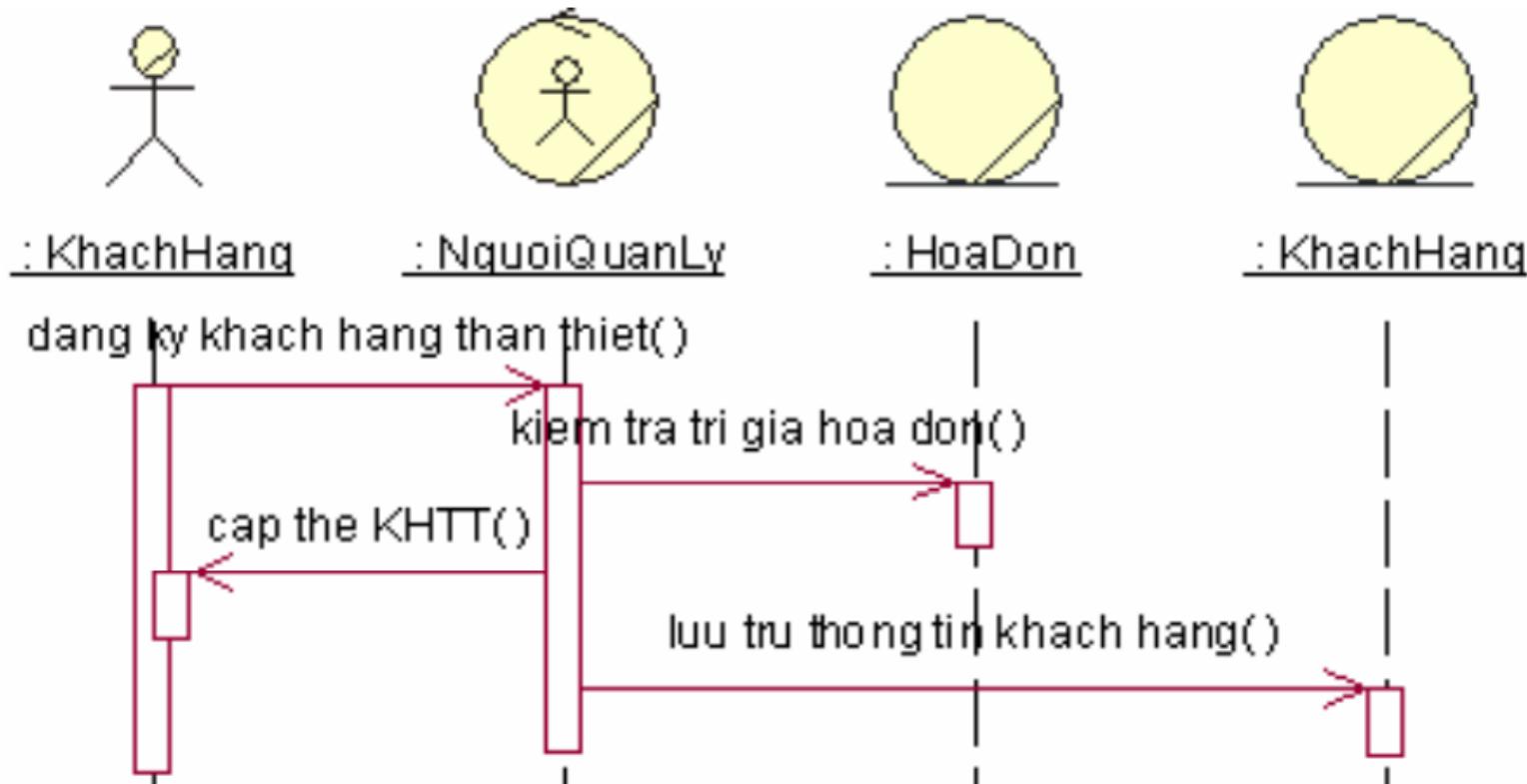
Ví dụ về sai !



Lược đồ sequence



Quản lý khách hàng thân thiết

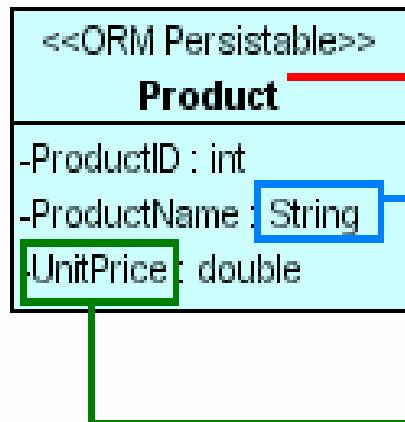


Một số biểu đồ khác



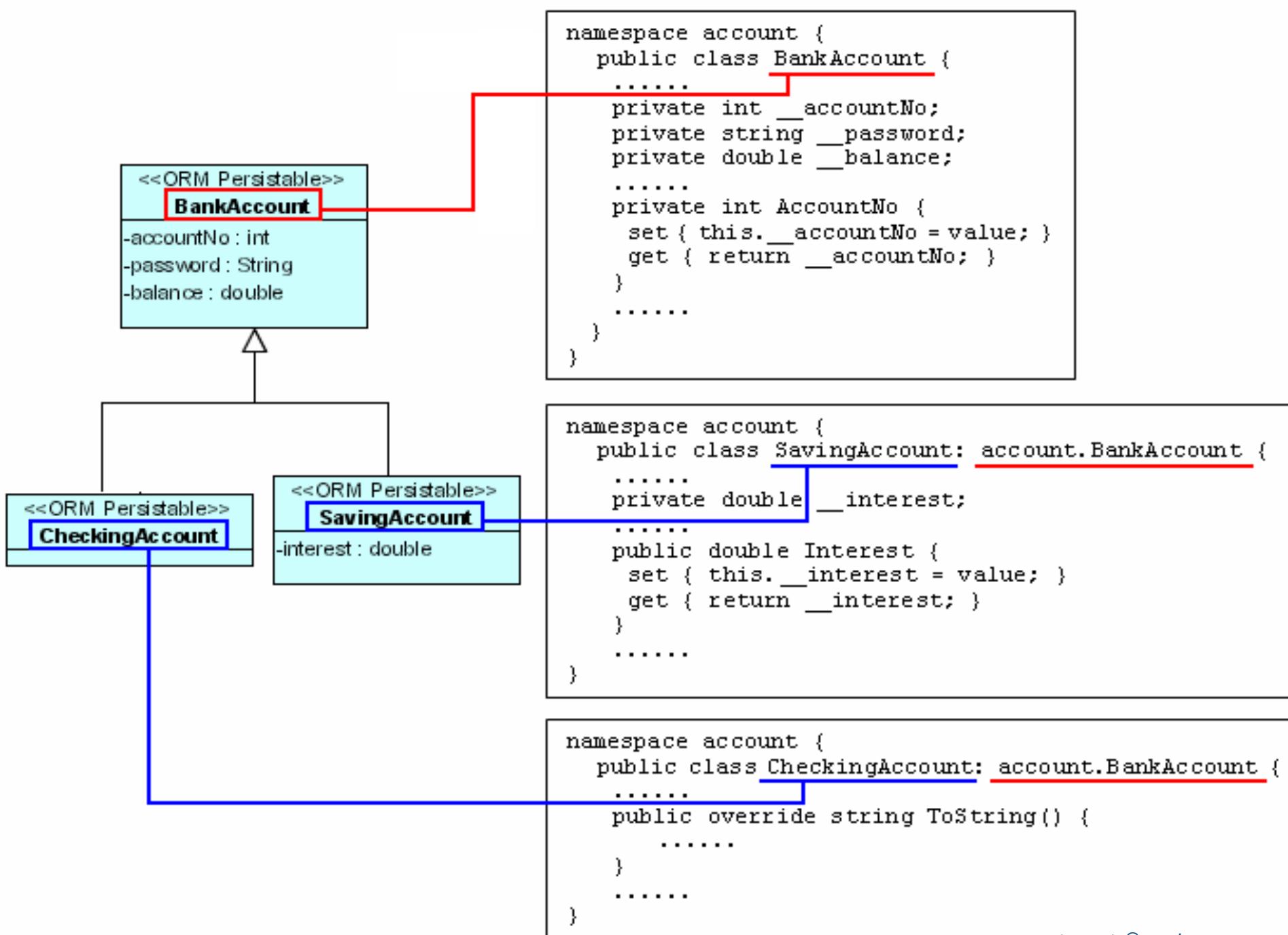
- ❖ Biểu đồ truyền thông: Communication diagram*
- ❖ Biểu đồ tương tác: Interaction Diagram
- ❖ Biểu đồ thời gian – Timming diagram*
- ❖ Biểu đồ trạng thái – State Diagram
- ❖ Biểu đồ đối tượng – Object Diagram
- ❖ Biểu đồ gói - Package Diagram
- ❖ Biểu đồ cấu trúc kết hợp – Composite Structured*
- ❖ Biểu đồ thành phần – Component Diagram
- ❖ Biểu đồ triển khai – Deployment Diagram

Ánh xạ biểu đồ sang Code



```
/// <summary>
/// ORM-Persistable Class
/// </summary>
public class Product {
    .....
    private int _ProductID;
    private string _ProductName;
    private double _UnitPrice;
    .....
}
```

- Mapping Classes, Attributes and Data Type

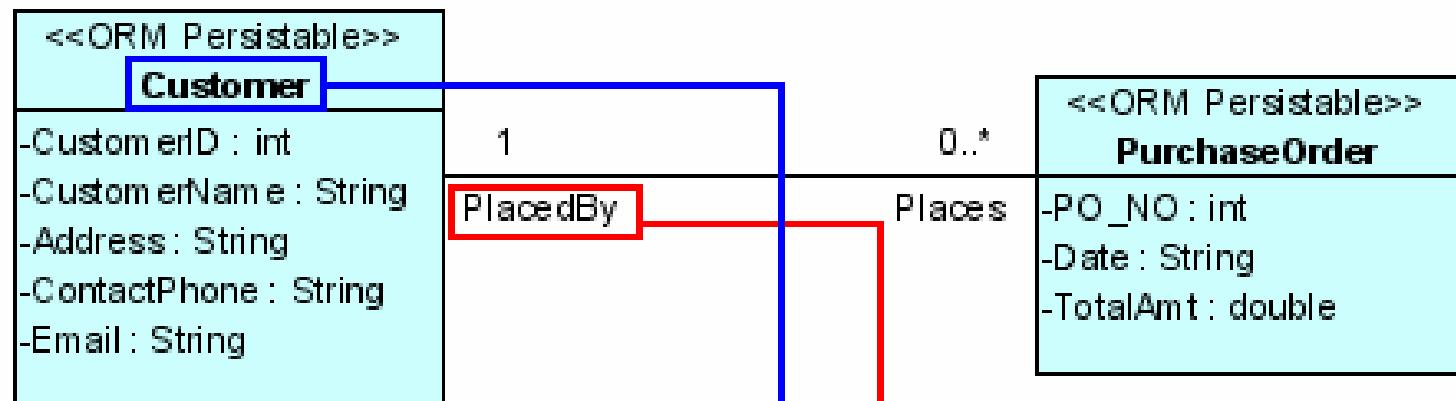


Ánh xạ khách hàng- đơn hàng



<p><<ORM Persistable>></p> <p>Customer</p> <p>-CustomerID : int -CustomerName : String -Address : String -ContactPhone : String -Email : String</p>	1	0..*	<p><<ORM Persistable>></p> <p>PurchaseOrder</p> <p>-PO_NO : int -Date : String -TotalAmt : double</p>
--	---	------	--

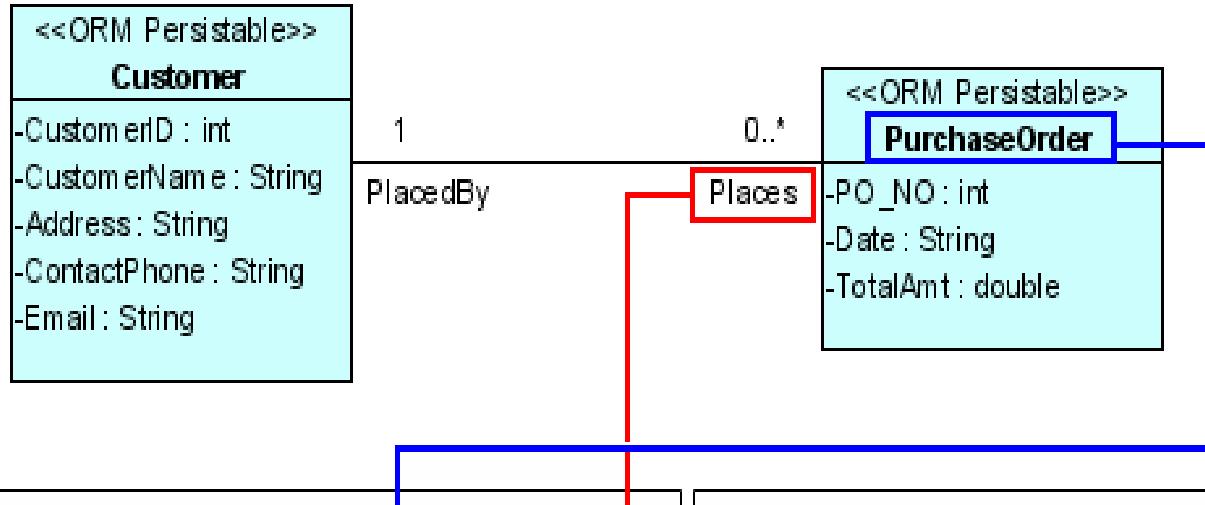
The Classes with association and multiplicity



```

public class PurchaseOrder {
    .....
    private int __PO_NO;
    private string __Date;
    private double __TotalAmt;
    private shoppingcart.Customer __PlacedBy;
    .....
    public shoppingcart.Customer PlacedBy {
        set { value.Places.Add(this); }
        get { return __PlacedBy; }
    }
    .....
}

```



```

public class Customer {
    .....
    private int _CustomerID;
    private string _CustomerName;
    private string _Address;
    private string _ContactPhone;
    private string _Email;
    .....
    public readonly
        shoppingcart.PurchaseOrderSetCollection Places:
    .....
}

```

```

public class PurchaseOrderSetCollection
: Orm.Util.ORMSet {
    .....
    public void Add(PurchaseOrder value) {
        .....
    }
    public void Remove(PurchaseOrder value) {
        .....
    }
    public bool Contains(PurchaseOrder value) {
        .....
    }
    .....
}

```

Mapping the One-to-many Associations

```

<<ORM Persistable>>
Software
-ID : int
-SWName : String
-Producer : String
-OSPlatform : String

```

1 belongsTo

Mapping



```

<<ORM Persistable>>
License
-SerialNo : String
-Licensee : String
-PurchasedDate : String

```

1 contains

```

public class Software{
    private int __ID;
    private string __SWName;
    private string __Producer;
    private string __OSPlatform;
    private License __contains;
    .....
    public License Contains{
        set {
            .....
        }
        get {
            .....
        }
    }
}

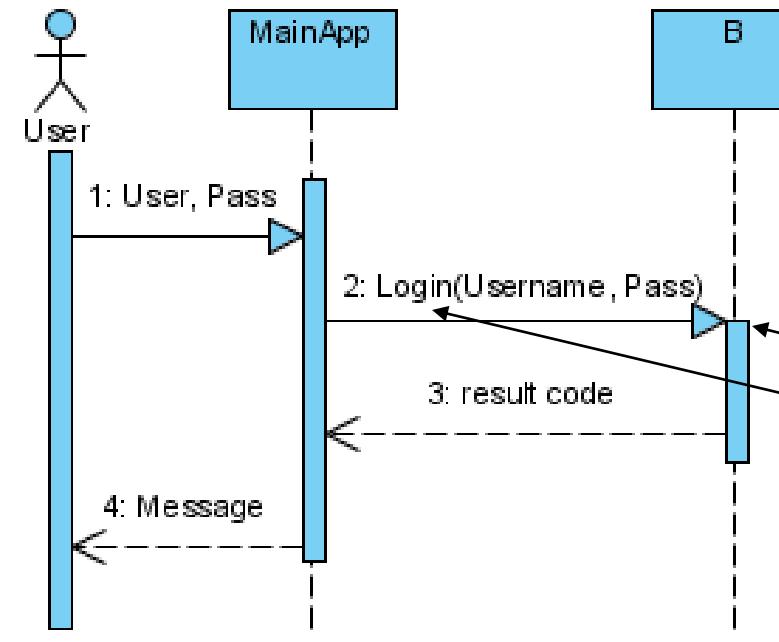
```

```

public class License{
    private string __SerialNo;
    private string __Licensee;
    private string __PurchasedDate;
    private Software __belongsTo;
    .....
    public Software BelongsTo{
        set {
            .....
        }
        get {
            .....
        }
    }
}

```

Ánh xạ biểu đồ tuần tự sang Code



Class B { public int Login(string UID, Pass)

{
}

MainApp

{ Nhập User name, password

B objB = new B();

bool Result = objB.Login(UID, Password)

if (Result == true)

}

Bài tập



- ❖ **Xây dựng sơ đồ Use case cho một hệ thống thương mại điện tử (ECommerce) như sau**

“Một công ty chuyên kinh doanh về các thiết bị điện tử và công nghệ thông tin trong nhiều năm nay và đã có một lượng khách hàng nhất định. Để mở rộng hoạt động kinh doanh của mình, công ty mong muốn xây dựng một hệ thống thương mại điện tử nhằm mở rộng phạm vi kinh doanh trên mạng Internet.

Hệ thống mới phải đảm bảo cho **khách hàng** viếng thăm Website dễ dàng lựa chọn **các sản phẩm**, xem các khuyến mãi cũng như mua hàng. Việc **thanh toán** có thể được thực hiện qua **mạng** hoặc **thanh toán** trực tiếp tại **cửa hàng**. Khách hàng có thể nhận hàng tại cửa hàng hoặc sử dụng dịch vụ chuyển hàng có phí của công ty.

Ngoài ra, hệ thống cũng cần có phân hệ để đảm bảo cho công ty **quản lý** các hoạt động kinh doanh như **số lượng hàng** có trong **kho**, quản lý đơn đặt hàng, tình trạng giao hàng, thanh toán v.v...



Hướng dẫn

❖ **Bước 1: Thu thập kiến thức liên quan đến hệ thống sẽ xây dựng**

Trước hết, để phân tích hệ thống trên bạn phải có kiến thức về hệ thống thương mại điện tử, chúng ta có thể tìm hiểu thông qua các nguồn sau:

- Xem các trang Web bán hàng qua mạng như amazon, lazada.vn, bkc.vn v.v..
- Xem các hệ thống mẫu về thương mại điện tử nguồn mở như Magento, OpenCart, Spree Commerce v.v...
- Đọc sách, báo về eCommerce
- Hỏi những người chuyên về lĩnh vực này (hỏi chuyên gia)



Hướng dẫn

❖ **Bước 2: Xác định các Actor: *Hãy trả lời cho câu hỏi***

- “Ai sử dụng hệ thống này?”
- “Hệ thống nào tương tác với hệ thống này?”

=>Actor của hệ thống gồm: Khách hàng tiềm năng, khách hàng, Người bán hàng, Quản lý Kho, Quản trị hệ thống, Cổng thanh toán



Hướng dẫn

❖ **Bước 3: Xác định Use Case**

Bạn cần trả lời câu hỏi

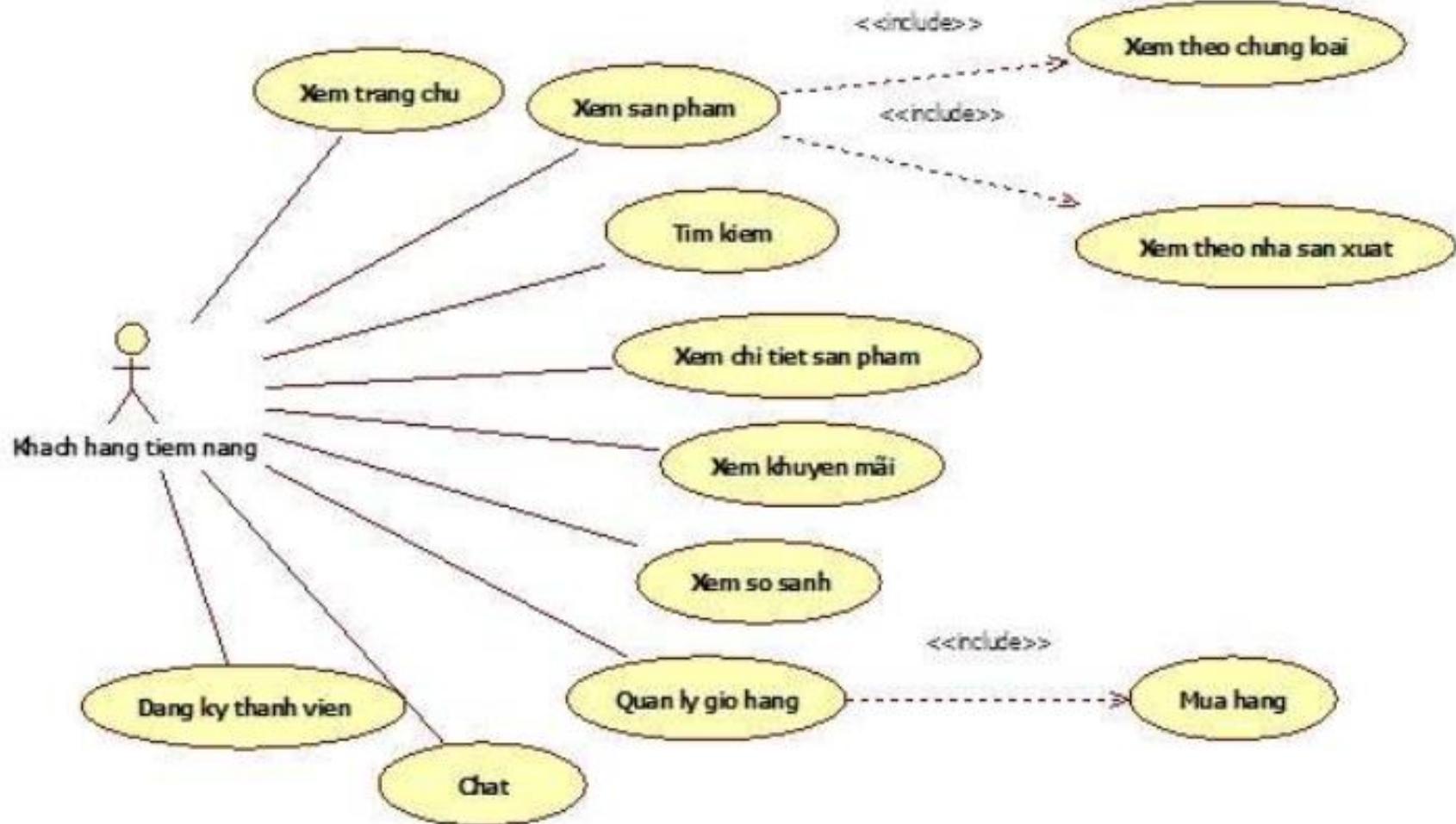
- “Actor sử dụng chức năng gì trên hệ thống?”



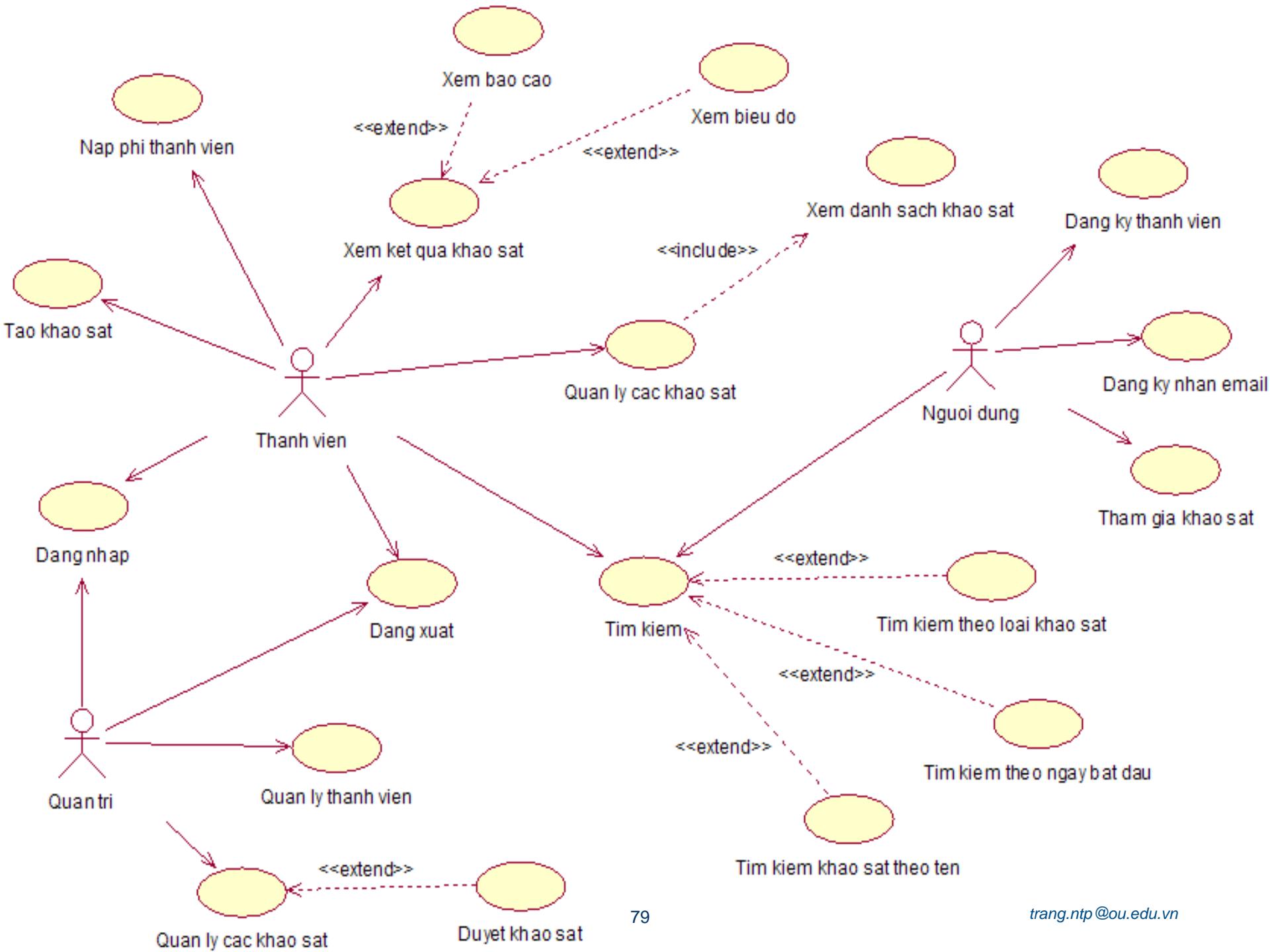
Hướng dẫn

❖ Bước 4: Vẽ Use Case

Hướng dẫn

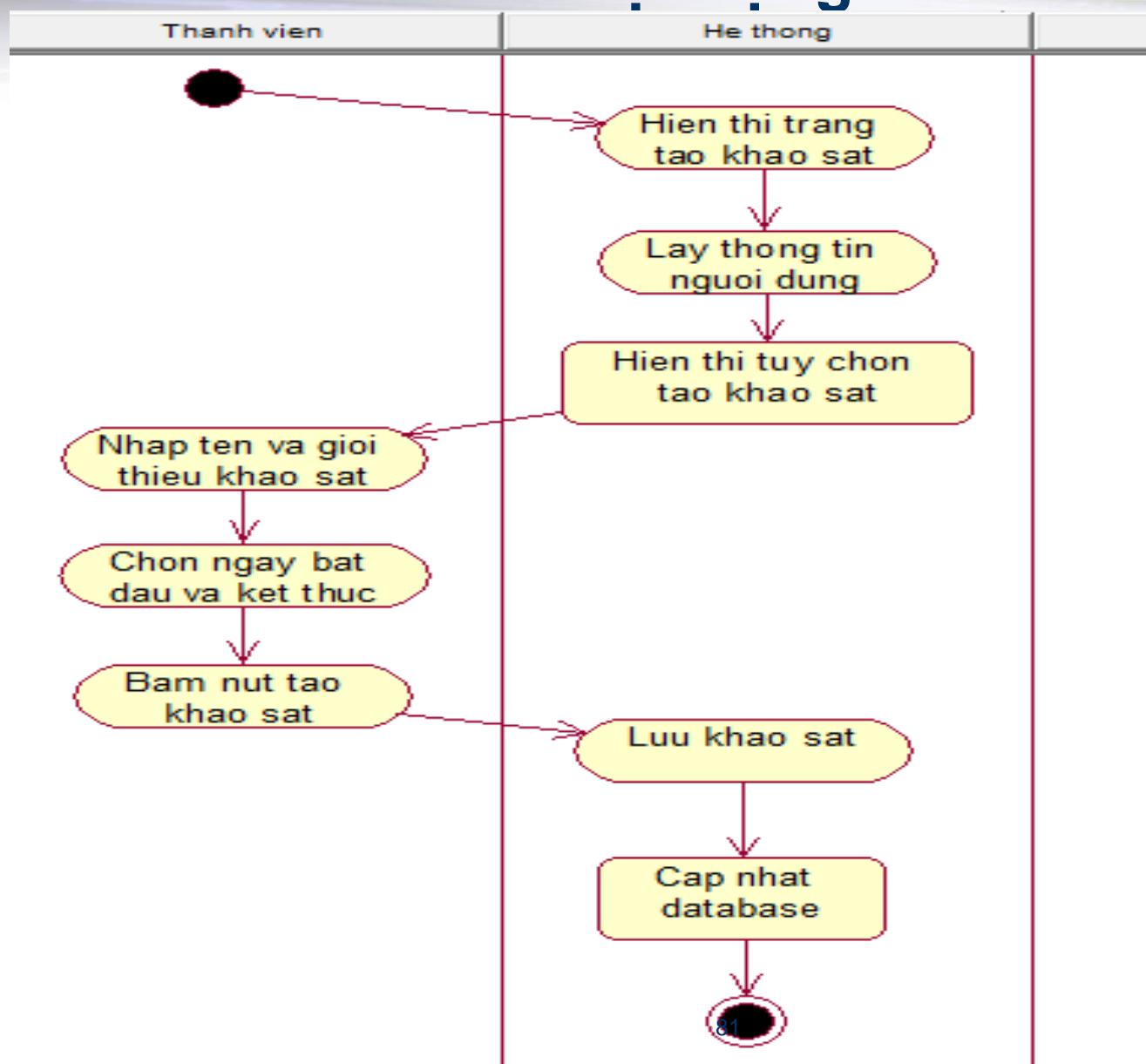




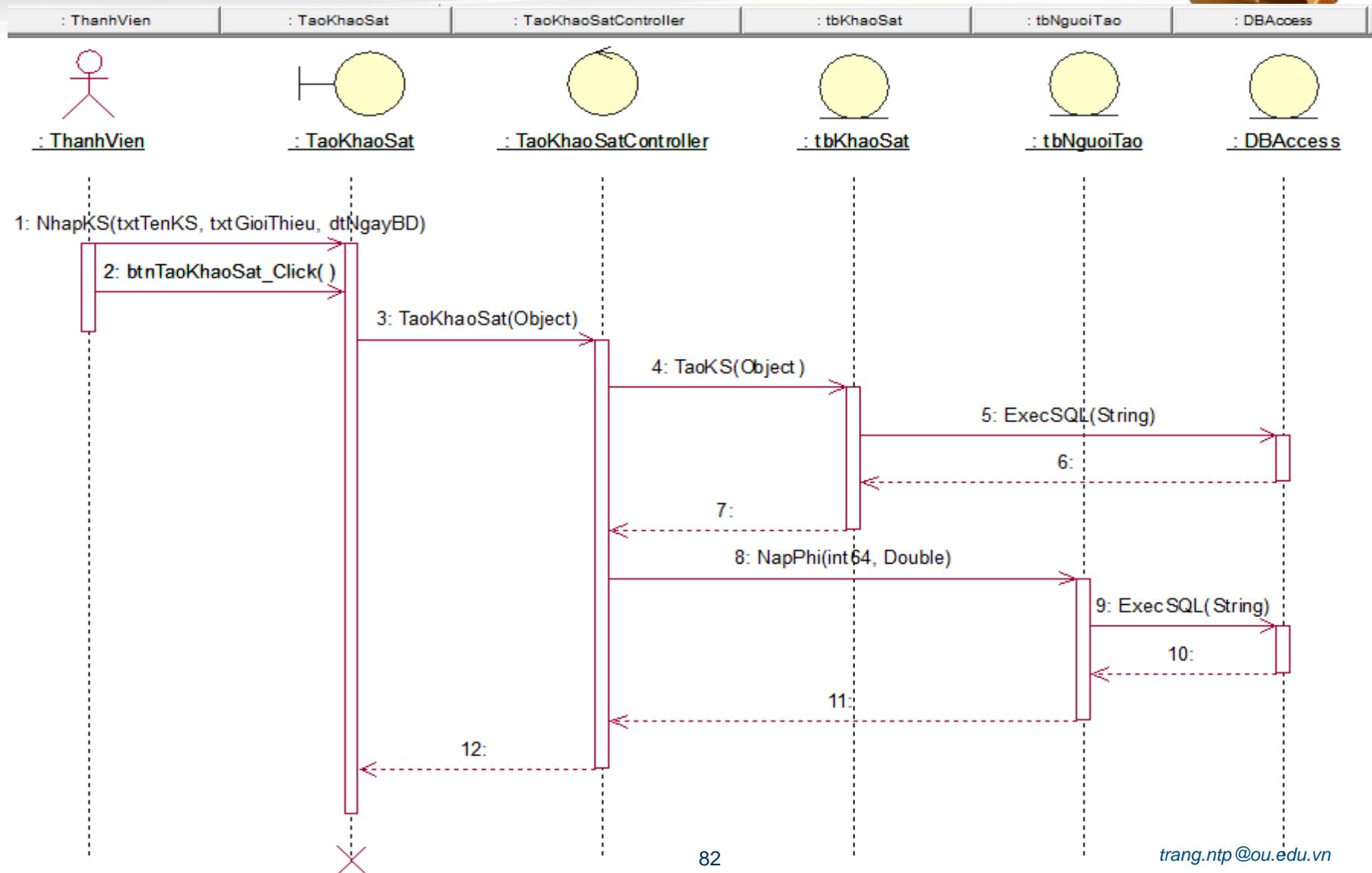


Name	Tạo khảo sát
Description	Thành viên tạo khảo sát
Actor	Thành viên
Pre conditions	<ul style="list-style-type: none"> Đăng nhập vào tài khoản thành viên Hiển thị trang tạo khảo sát
Post conditions	Thông báo kết quả tạo khảo sát
Flow of events	<ol style="list-style-type: none"> Hệ thống lấy thông tin tài khoản người dùng Hệ thống kiểm tra tài khoản. Hệ thống hiển thị tùy chọn tạo khảo sát có tính phí hay không. Người dùng nhập tên khảo sát Người dùng nhập giới thiệu khảo sát Chọn ngày bắt đầu khảo sát Chọn ngày kết thúc khảo sát Nhấn nút tạo khảo sát <ol style="list-style-type: none"> Hệ thống lưu nội dung khảo sát xuống cơ sở dữ liệu Không lưu được: A1 Thông báo kết quả lên trang web
Alternative flow	<p>A1: Thông báo không tạo được khảo sát. Trở lại trang tạo khảo sát</p> <p>A2: Thông báo không lưu được khảo sát Trở lại trang tạo khảo sát</p>

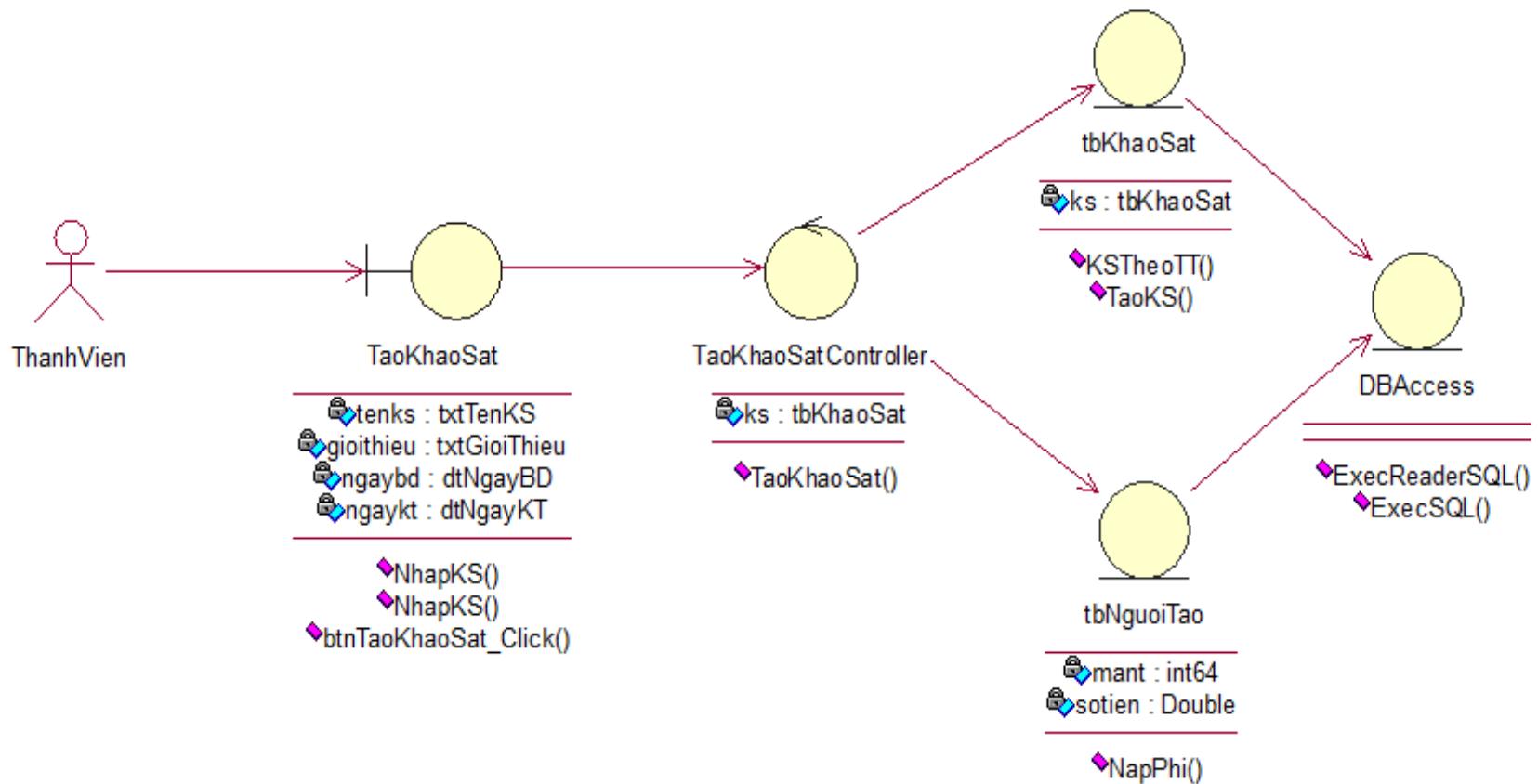
Biểu đồ hoạt động



Biểu đồ trình tự

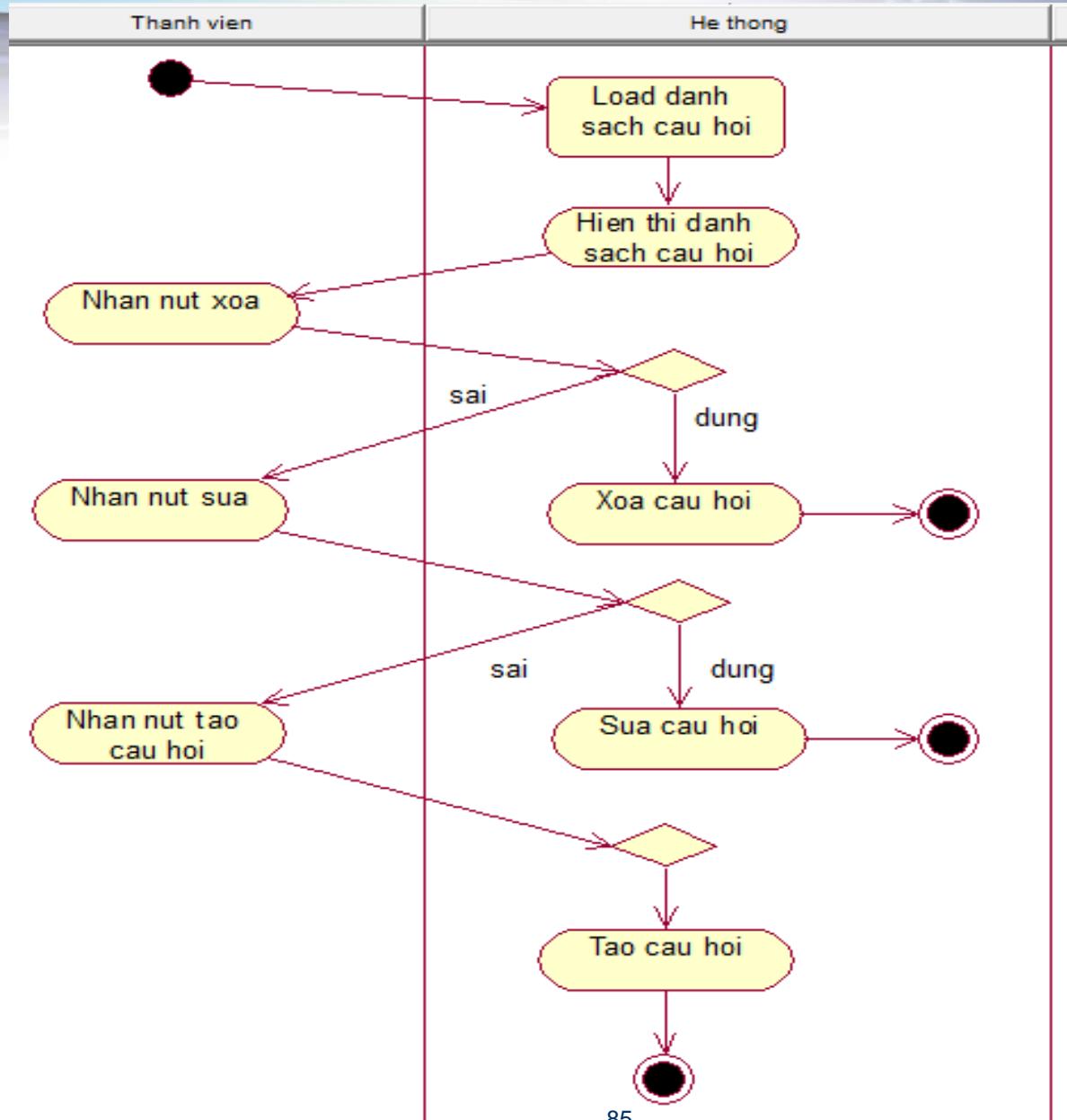


Biểu đồ class



Name	Quản lý câu hỏi
Description	Thành viên quản lý tạo, cập nhật, xoá câu hỏi
Actor	Thành viên
Pre conditions	<ul style="list-style-type: none"> Đăng nhập vào tài khoản thành viên Hiển thị trang quản lý câu hỏi
Post conditions	Thông báo kết quả thao tác
Flow of events	<ol style="list-style-type: none"> Hệ thống lấy danh sách câu hỏi theo khảo sát và người tạo khảo sát. Hệ thống hiển thị danh sách câu hỏi Không hiển thị được: A1 Người dùng nhấn nút tạo câu hỏi Hệ thống chuyển sang trang tạo câu hỏi mới Người dùng nhấn nút chỉnh sửa của câu hỏi Hệ thống hiển thị trang chỉnh sửa câu hỏi <ol style="list-style-type: none"> Không thể sửa: A2 Người dùng nhấn nút xoá câu hỏi <ol style="list-style-type: none"> Không thể xóa: A3
Altenative flows	<p>A1: Thông báo không hiển thị được.</p> <p>A2: Thông báo không sửa được câu hỏi</p> <p>A3: Thông báo ⁴không xóa được câu hỏi.</p>

Biểu đồ hoạt động



Biểu đồ trình tự



: ThanhVien

: DSCauHoi

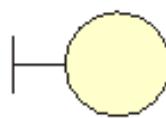
: DSCauHoiController

: tbCauHoi

: DBAccess



: ThanhVien



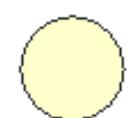
: DSCauHoi



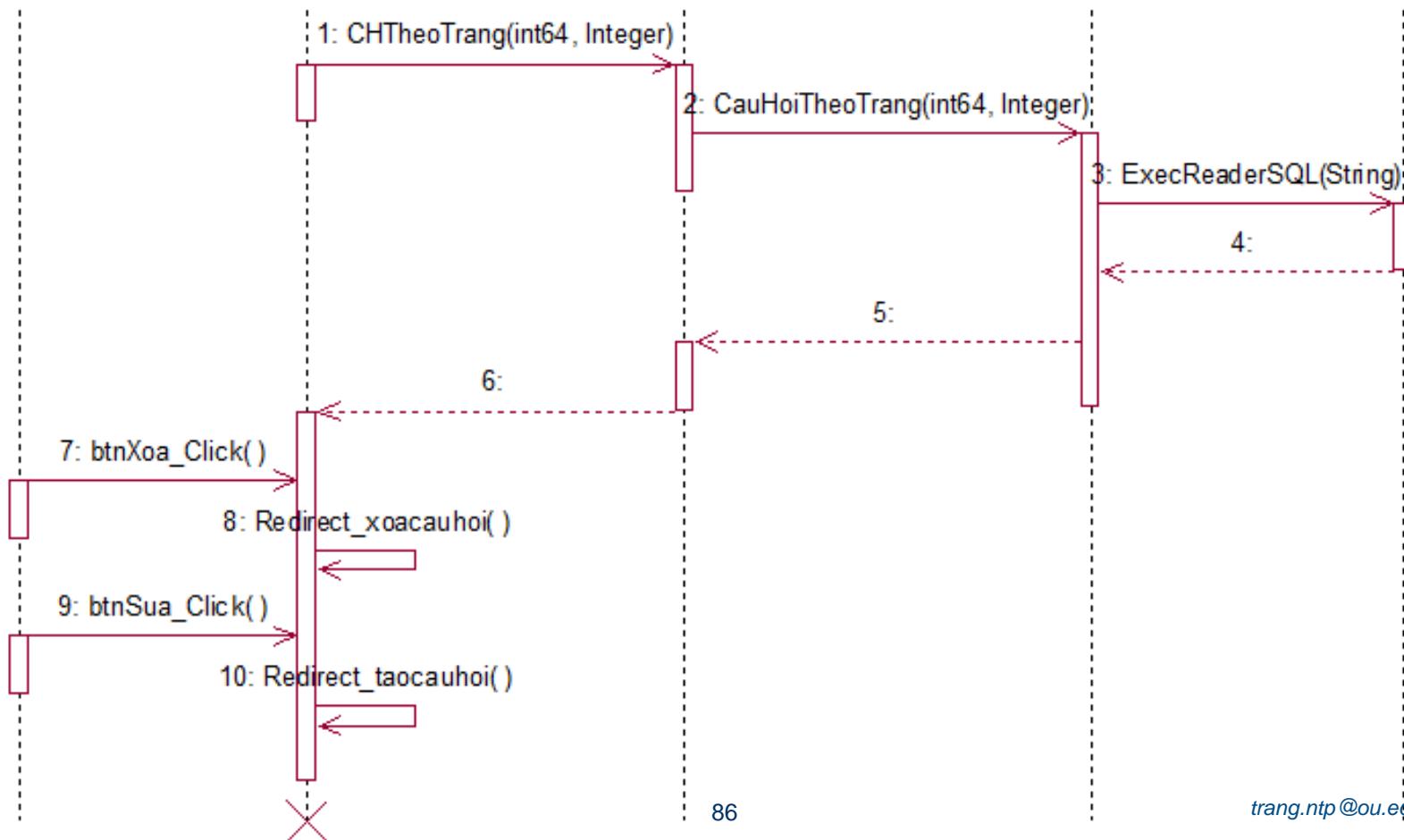
: DSCauHoiController



: tbCauHoi

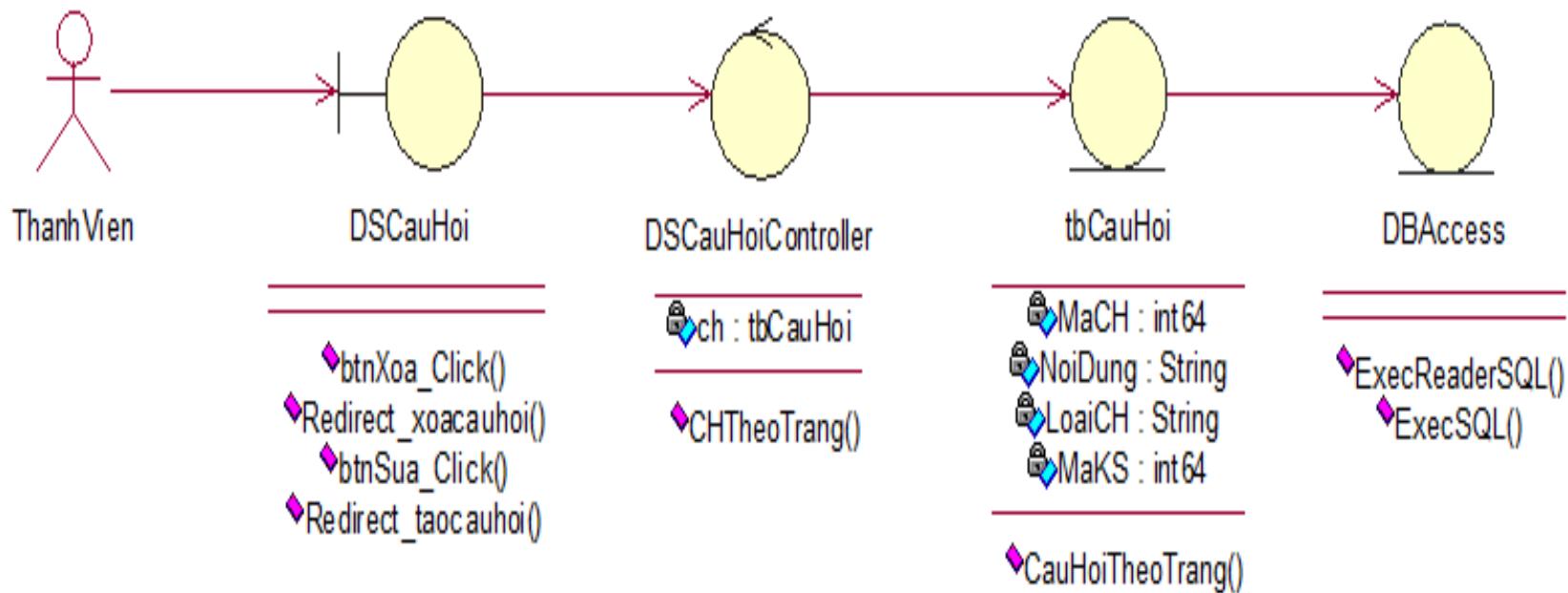


: DBAccess





Biểu đồ class





TRƯỜNG ĐẠI HỌC MỞ
TP HỒ CHÍ MINH

Bắt đầu lúc 7h15

BÀI GIẢNG
**PHÂN TÍCH THIẾT KẾ HỆ THỐNG
THÔNG TIN**

**Chương 5: Phân tích thiết kế
hướng đối tượng**

ThS. Nguyễn Thị Phương Trang
trang.ntp@ou.edu.vn

Bài tập



Hệ thống Quản lý Khám chữa bệnh, được mô tả như sau:

A. TIẾP NHẬN BỆNH NHÂN:

Bệnh nhân tiến hành mua phiếu khám điền thông tin đầy đủ vào phiếu khám. Nhân viên tiếp nhận tiến hành kiểm tra dựa trên thông tin phiếu khám xem bệnh nhân đã có trong hồ sơ bệnh nhân chưa.

1. Nếu chưa có thì Tạo hồ sơ mới cho bệnh nhân.
2. Nếu đã có thì cập nhật ngày khám vào hồ sơ đã có.

B. KHÁM BỆNH:

Bệnh nhân tiến hành khám bệnh theo thứ tự phiếu khám. Bác sĩ tiến hành nghe triệu chứng, tiền sử bệnh, yêu cầu bệnh nhân làm các xét nghiệm khi cần thiết rồi đưa ra kết quả khám bệnh và đưa ra pháp đồ điều trị. Kết quả khám được bác sĩ lưu vào trong Hồ sơ bệnh nhân.

1. Nếu bệnh nhân cần nhập viện điều trị thì bác sĩ yêu cầu bệnh nhân nhập viện để tiến hành điều trị nội trú. Bệnh nhân liên hệ với phòng quản lý nội trú để được xếp phòng bệnh theo giấy kết quả khám của bác sĩ.
2. Nếu không cần bác sĩ sẽ kê đơn cho bệnh nhân tiến hành điều trị ngoại trú.

C. ĐIỀU TRỊ:

Khi điều trị nội trú Bác sĩ tiến hành việc điều trị theo phác đồ. Các thông tin về tình trạng bệnh nhân được lưu lại trong CSLD. Trong quá trình điều trị Bác sĩ có thể tiến hành kê đơn, yêu cầu bệnh nhân xét nghiệm hay chuyển khoa. Các thông tin về đơn thuốc cũng được lưu lại trong hồ sơ đơn thuốc.

D. XUẤT VIỆN:

Khi bệnh nhân yêu cầu hoặc do bác sĩ yêu cầu bệnh nhân chuyển viện hay xuất viện Nhân viên thu ngân sẽ tiến hành tính toán và đưa ra hóa đơn viện phí theo dữ liệu về đơn thuốc và ngày nhập viện, ngày xuất viện trong hồ sơ bệnh nhân. Nhân viên thu ngân tiến hành cập nhật ngày xuất hoặc chuyển viện vào trong hồ sơ bệnh nhân.

1. Vẽ biểu đồ use case tổng quát
2. Xây dựng biểu đồ hoạt động cho từng use case
3. Xây dựng biểu đồ class
4. Xây dựng biểu đồ tuần tự cho từng use case

Bài tập cộng điểm nhóm: Hệ thống Quản lý khám bệnh



A. TIẾP NHẬN BỆNH NHÂN:

Bệnh nhân tiến hành mua phiếu khám điền thông tin đầy đủ vào phiếu khám. Nhân viên tiếp nhận tiến hành kiểm tra dựa trên thông tin phiếu khám xem bệnh nhân đã có trong hồ sơ bệnh nhân chưa.

1. Nếu chưa có thì Tạo hồ sơ mới cho bệnh nhân.
2. Nếu đã có thì cập nhật ngày khám vào hồ sơ đã có.

B. KHÁM BỆNH:

Bệnh nhân tiến hành khám bệnh theo thứ tự phiếu khám. Bác sĩ tiến hành nghe triệu chứng, tiền sử bệnh, yêu cầu bệnh nhân làm các xét nghiệm khi cần thiết rồi đưa ra kết quả khám bệnh và đưa ra pháp đồ điều trị. Kết quả khám được bác sĩ lưu vào trong Hồ sơ bệnh nhân.

1. Nếu bệnh nhân cần nhập viện điều trị thì bác sĩ yêu cầu bệnh nhân nhập viện để tiến hành điều trị nội trú. Bệnh nhân liên hệ với phòng quản lý nội trú để được xếp phòng bệnh theo giấy kết quả khám của bác sĩ.
2. Nếu không cần bác sĩ sẽ kê đơn cho bệnh nhân tiến hành điều trị ngoại trú.

C. ĐIỀU TRỊ:

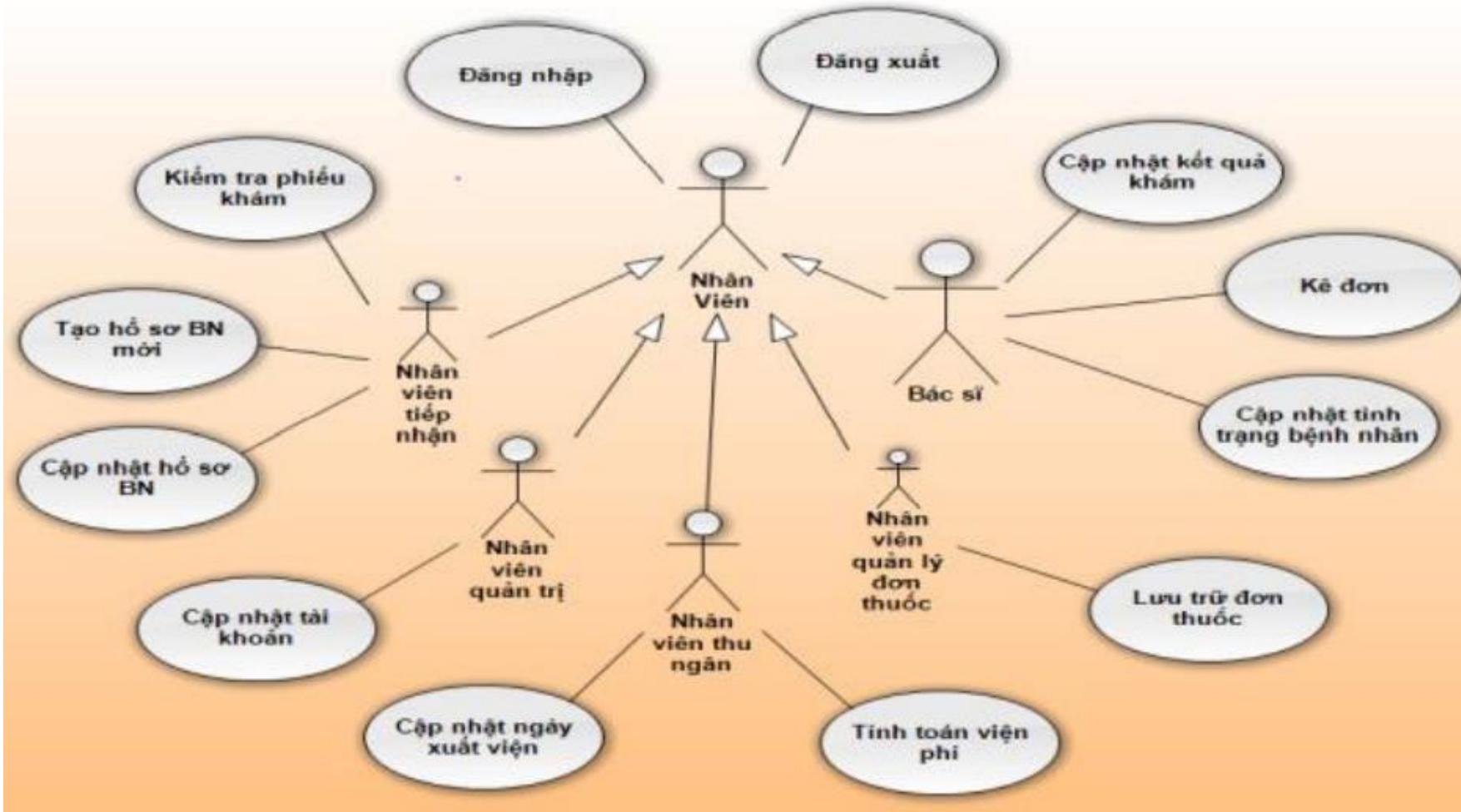
Khi điều trị nội trú Bác sĩ tiến hành việc điều trị theo phác đồ. Các thông tin về tình trạng bệnh nhân được lưu lại trong CSLD. Trong quá trình điều trị Bác sĩ có thể tiến hành kê đơn, yêu cầu bệnh nhân xét nghiệm hay chuyển khoa. Các thông tin về đơn thuốc cũng được lưu lại trong hồ sơ đơn thuốc.

D. XUẤT VIỆN:

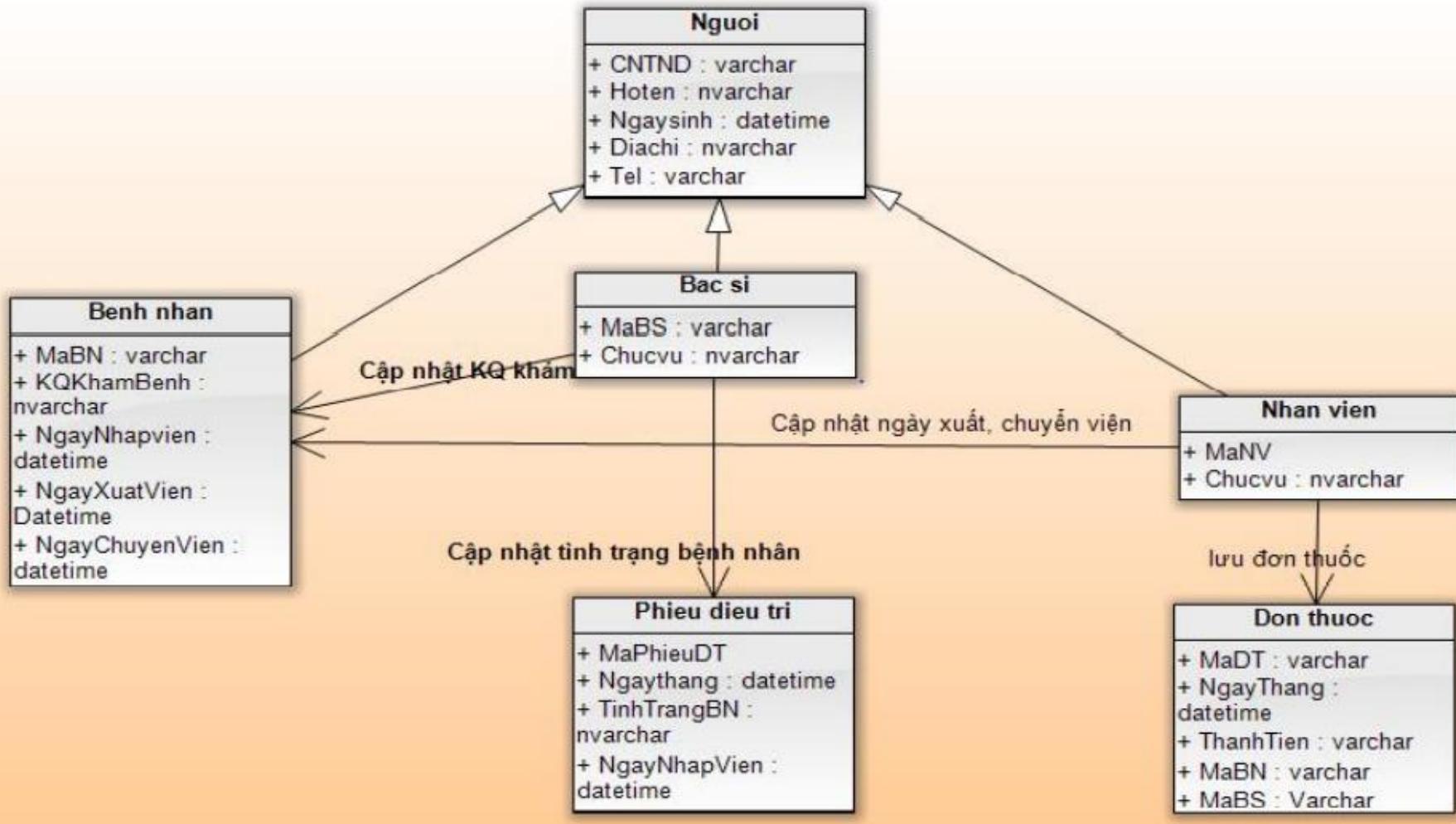
Khi bệnh nhân yêu cầu hoặc do bác sĩ yêu cầu bệnh nhân chuyển viện hay xuất viện Nhân viên thu ngân sẽ tiến hành tính toán và đưa ra hóa đơn viện phí theo dữ liệu về đơn thuốc và ngày nhập viện, ngày xuất viện trong hồ sơ bệnh nhân. Nhân viên thu ngân tiến hành cập nhật ngày xuất hoặc chuyển viện vào trong hồ sơ bệnh nhân.

1. Vẽ biểu đồ use case tổng quát
2. Xây dựng biểu đồ hoạt động cho từng use case
(Use case đăng nhập, use case Tạo hồ sơ mới, cập nhật thông tin chuyển viện)
3. Xây dựng biểu đồ class
(Biểu đồ class tổng thể, biểu đồ class Tạo hồ sơ mới, cập nhật thông tin chuyển viện)
4. Xây dựng biểu đồ tuần tự cho từng use case
(Use case đăng nhập, use case Tạo hồ sơ mới, cập nhật thông tin chuyển viện)

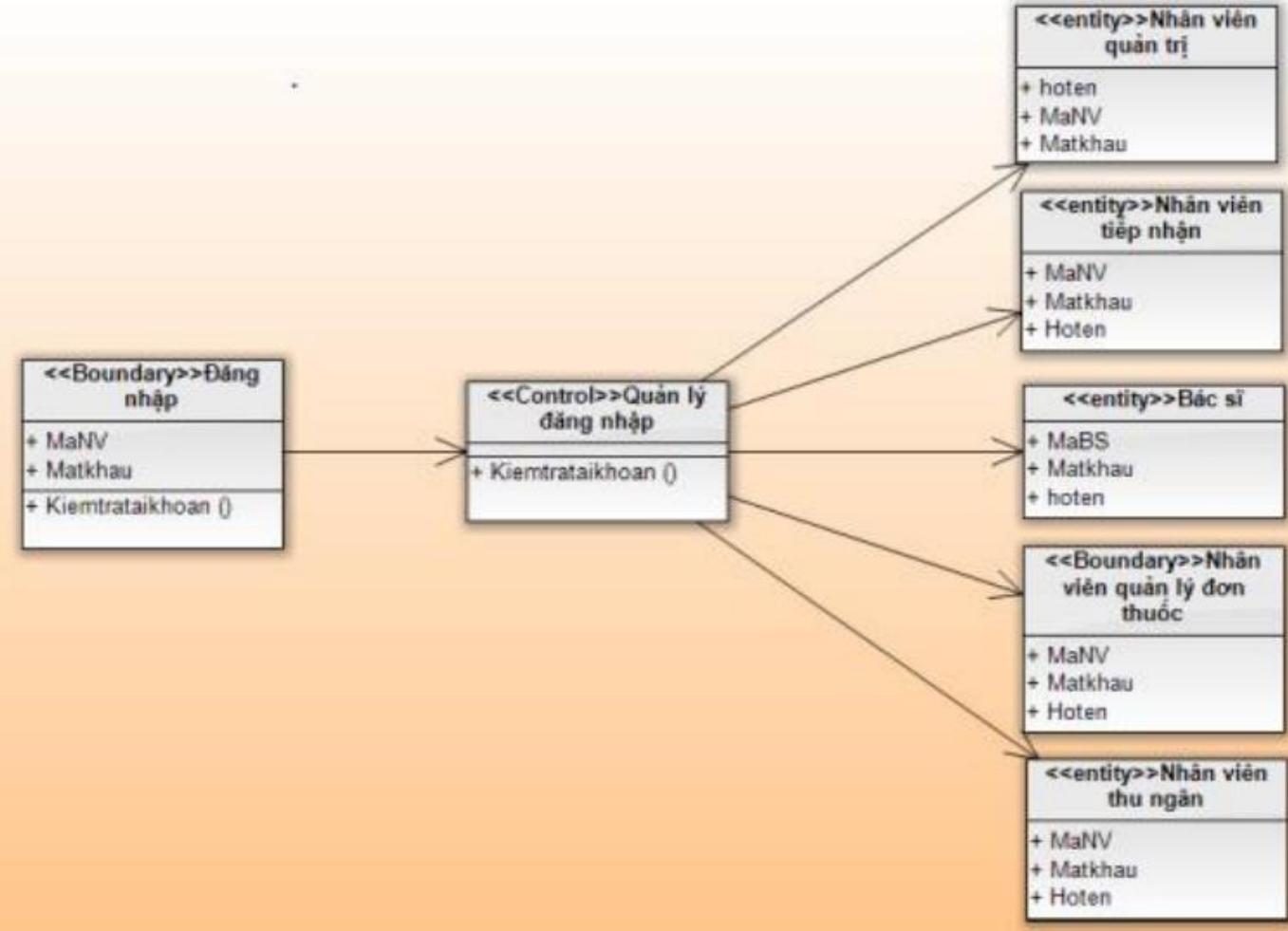
Biểu đồ Use case



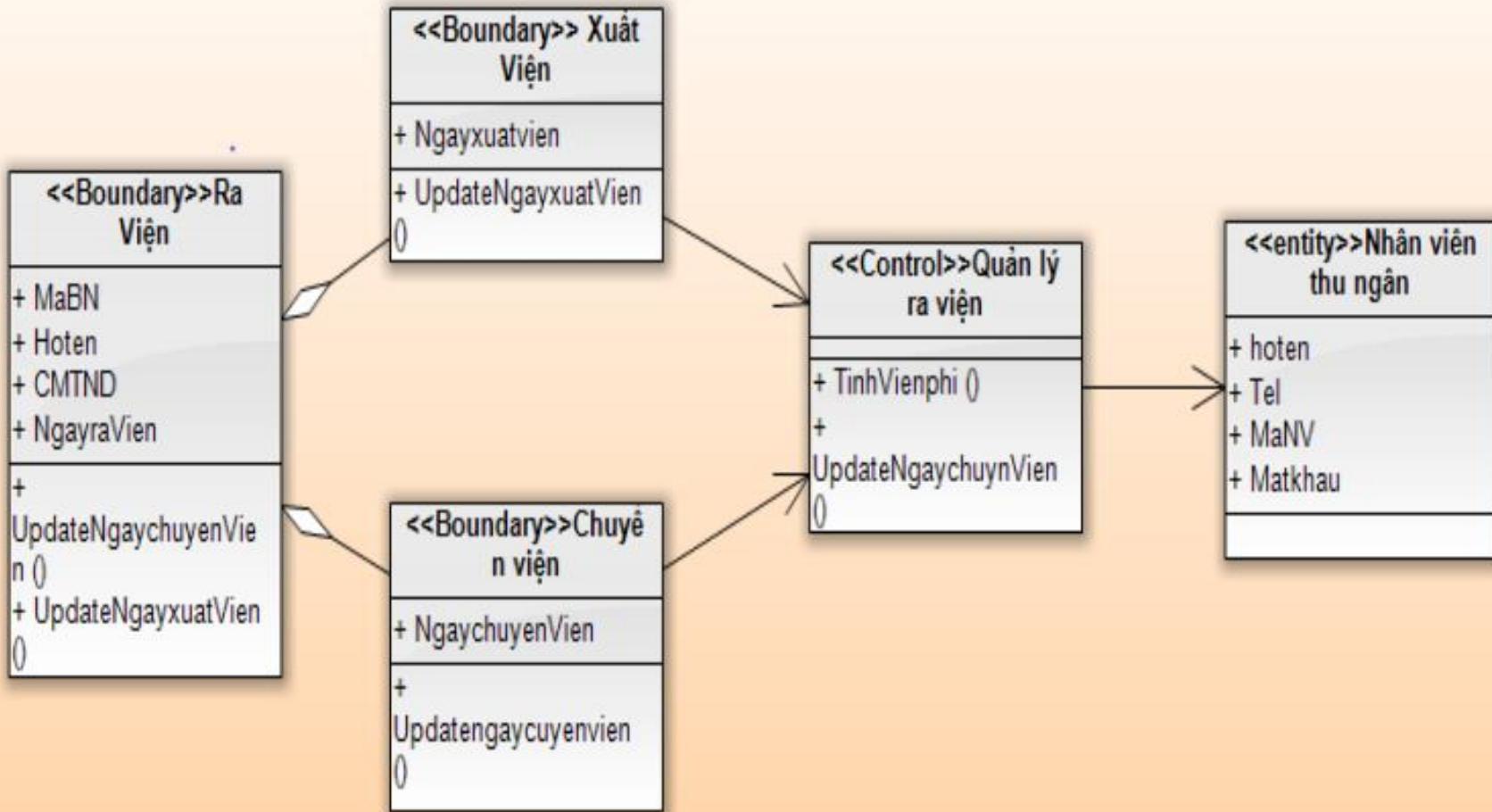
Biểu đồ lớp



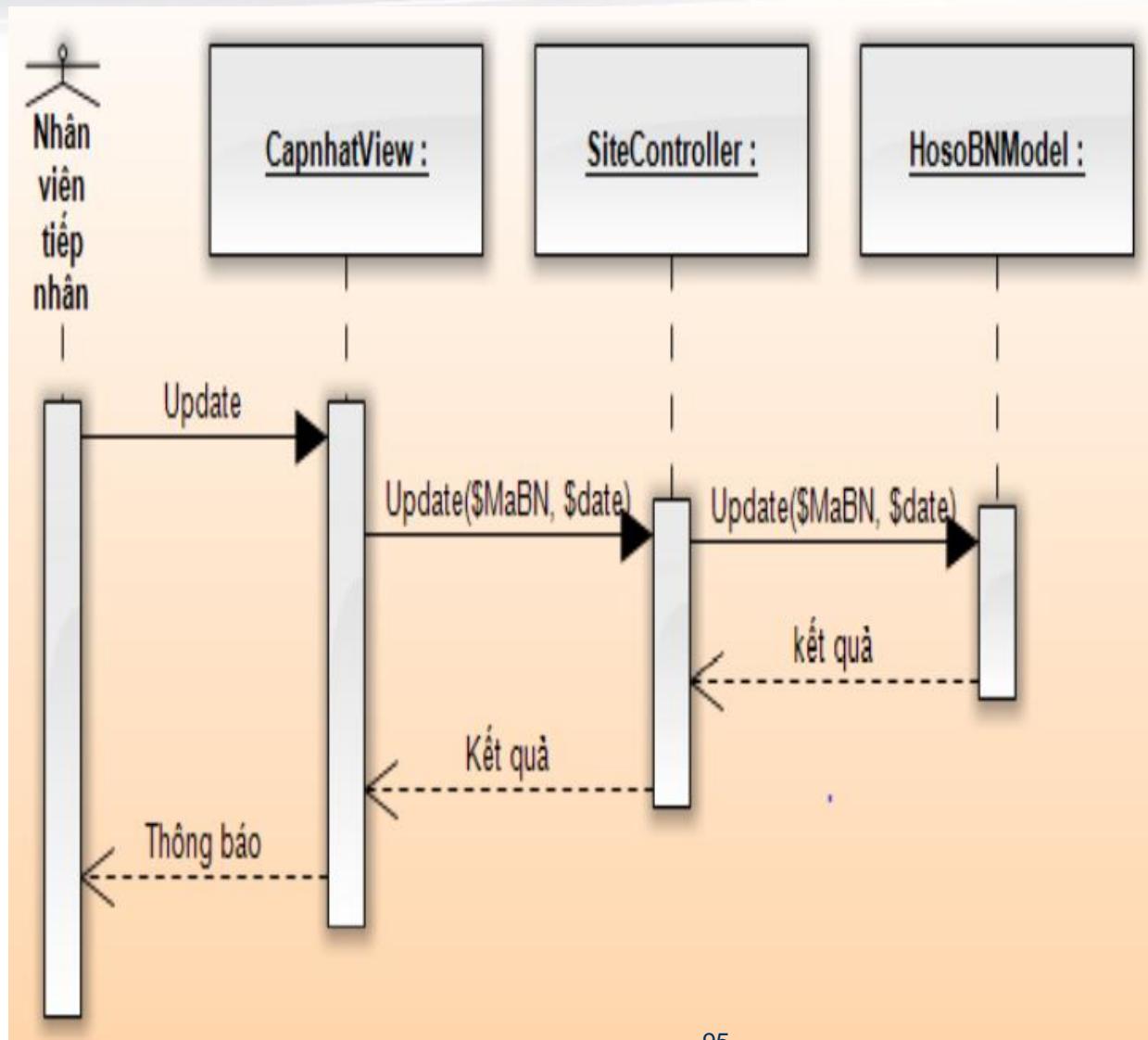
Biểu đồ lớp – Đăng nhập



Biểu đồ lớp – CNTT Chuyển viện



Biểu đồ trình tự - CN Ngày khám





Thank You !