

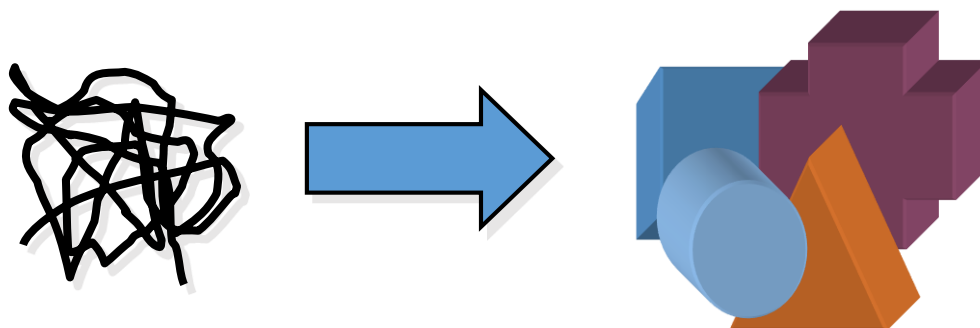


## Chủ đề 2: UML



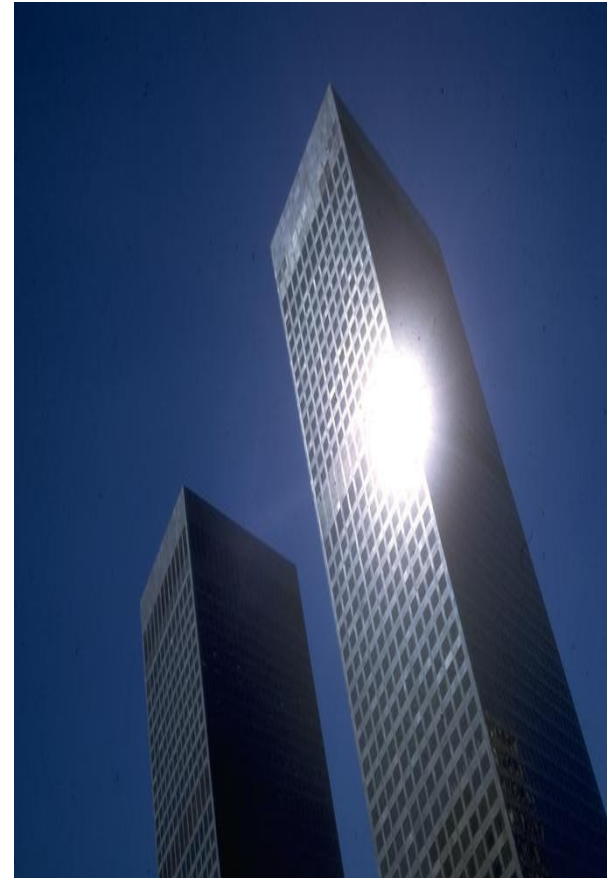
# What is modeling?

- To represent how a business works and functions in such a way that it can productively be used as a means to simulate the real world
- A prediction of what would happen under a certain set of conditions





# The importance of modeling





# Why do we modeling?

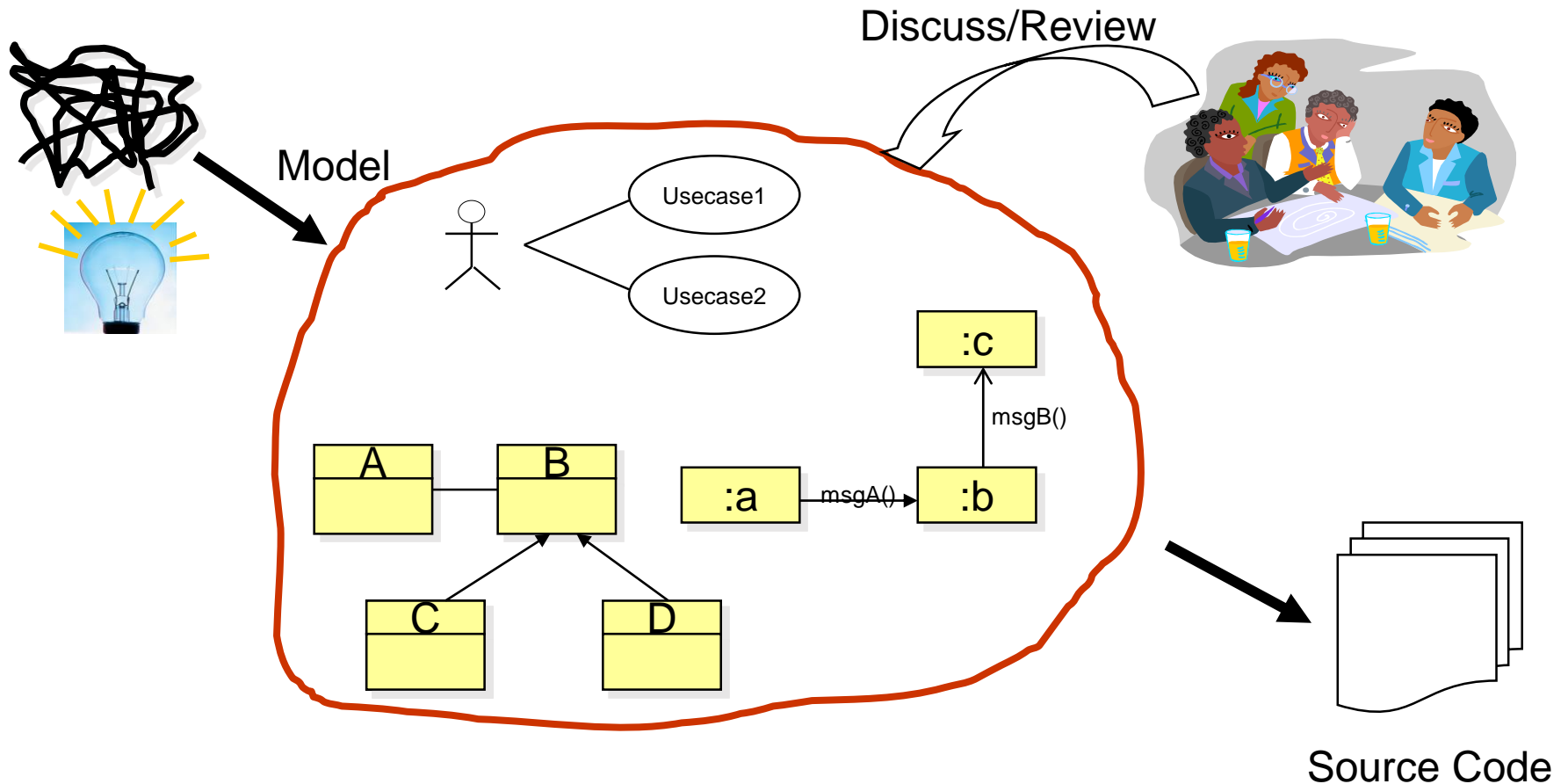
- We build models to better understand the system we are developing
- Modeling
  - Helps us to visualize a system as we want it to be
  - Permits us to specify the structure or behavior of a system
  - Gives us a template that guides us in constructing a system
  - Documents the decisions we have made
- We build models of complex systems because we cannot comprehend such a system in its entirety



- There are some problems that occur in software development
  - Diversification of software according to service and product which becomes more feature and high function
  - Trouble on communication because of different culture
  - Correspond with the changing requirement of short cycle and fast changing in business environment
  - The increasing of cost and the difficulty in maintenance of software



# Model in software





# The advantage of using model:

- It is easy to send the thinking image by clearing the small part
- It is able to sharing thing created by relationship between related person
- Can understand the outline of complicated software
- To adjust the problem, can correspond flexible to the changing of requirement

**Model is to decrease complication, share information between related persons, correspond to changing requirement, effective in maintain software**



# Software modeling

- There many methodologies:
  - DFD: Data Flow Diagram
  - ERD: Entity Relationship Diagram
  - UML: Unified Modeling Language



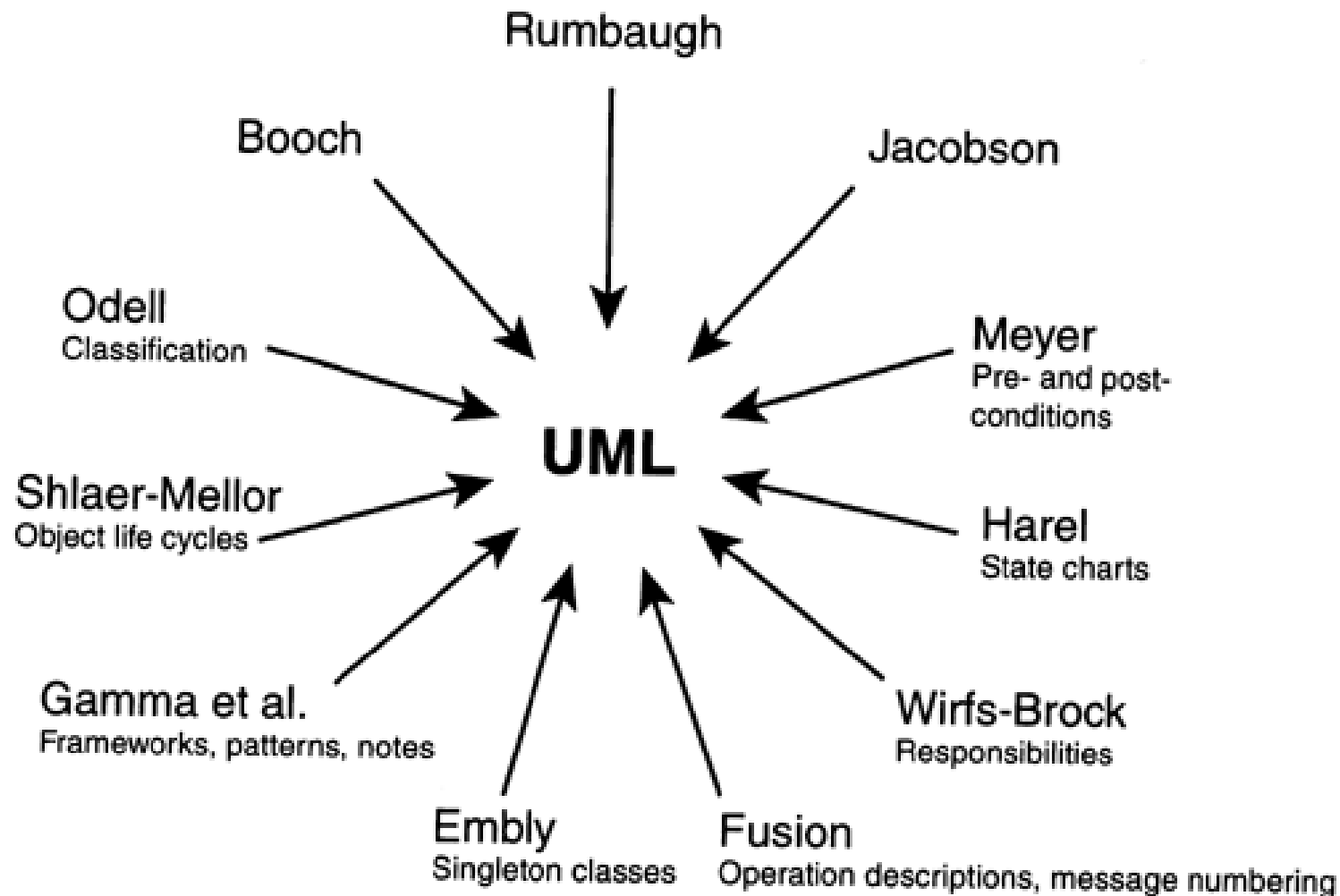


# Giới thiệu về UML





# History of UML





# 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.x 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)



# 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 lập trình.
- 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!
- Phiên bản mới nhất của UML là 2.5 ([www.omg.org](http://www.omg.org))



# 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 ngữ 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 UML có thể kết nối với nhiều ngôn ngữ lập trình. Tức là có thể ánh xạ các mô hình UML về một NNLT như C++, Java, C#...
- 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”





# UML là ngôn ngữ giúp viết tài liệu

- 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
- .....



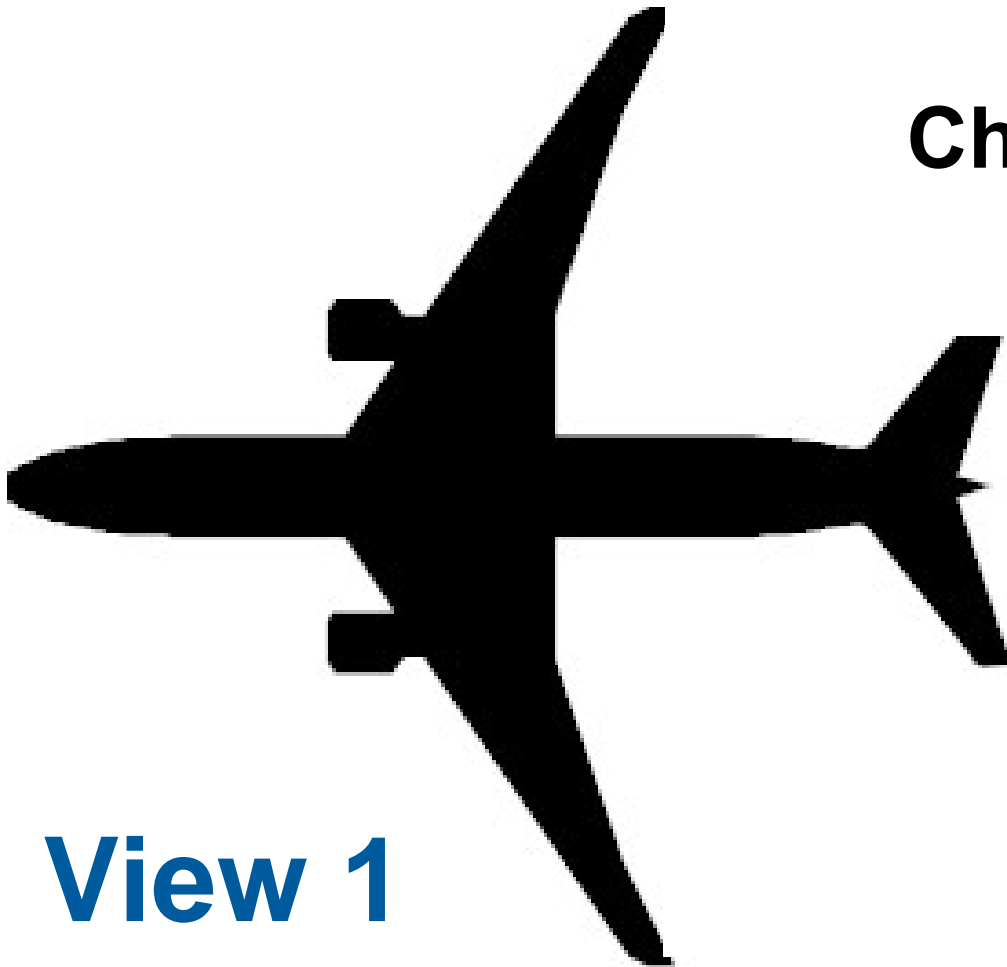
# Một số Case Tool hỗ trợ UML

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



# Một số biểu đồ UML cơ bản

Chiều cao ? ? ?



**View 1**



# Một số biểu đồ UML cơ bản

Chiều cao phía sau ?



View 2



# Một số biểu đồ UML cơ bản



**View 3**

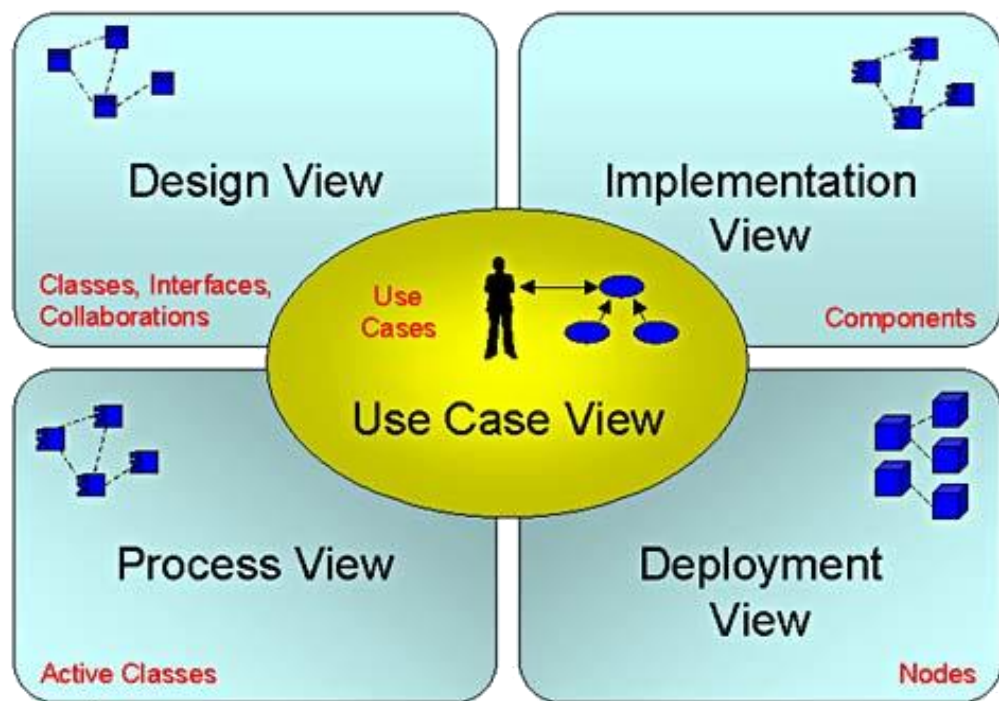
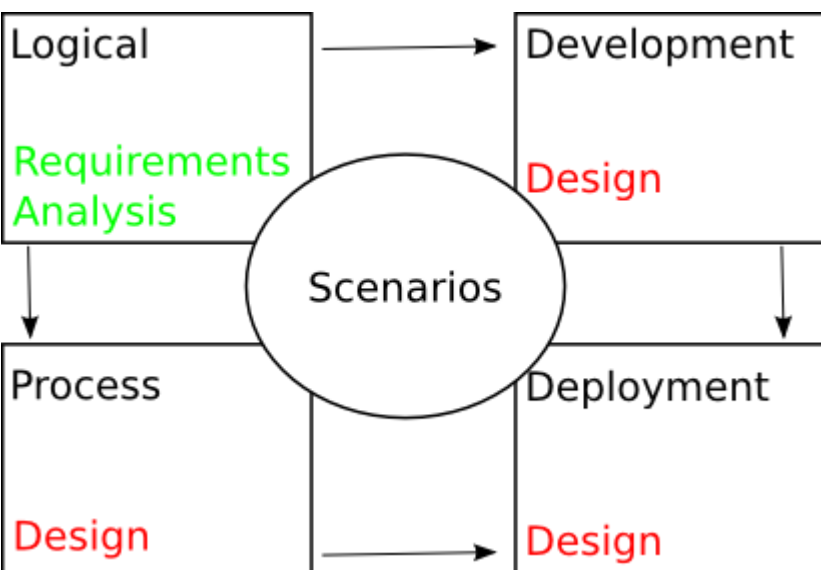


# Thành phần của ngôn ngữ UML

- Hướng nhìn (View)
- Biểu đồ (diagram)
- Phần tử mô hình hóa (model element)



# “4 + 1” view





# Khung nhìn (View)

- **Use case view:** chỉ ra chức năng hệ thống, nhìn từ tác nhân bên ngoài.
- **Logical view:** chỉ ra chức năng được thiết kế bên trong hệ thống ntn qua các khái niệm cấu trúc tĩnh cũng như ứng xử động của hệ thống.
- **Component view:** chỉ ra khía cạnh tổ chức của các thành phần code.
- **Concurrency view:** chỉ ra sự tồn tại đồng thời trong hệ thống, hướng đến việc giao tiếp và đồng bộ hóa trong hệ thống.
- **Deployment view:** chỉ ra khía cạnh triển khai hệ thống tập trung vào kiến trúc vật lý

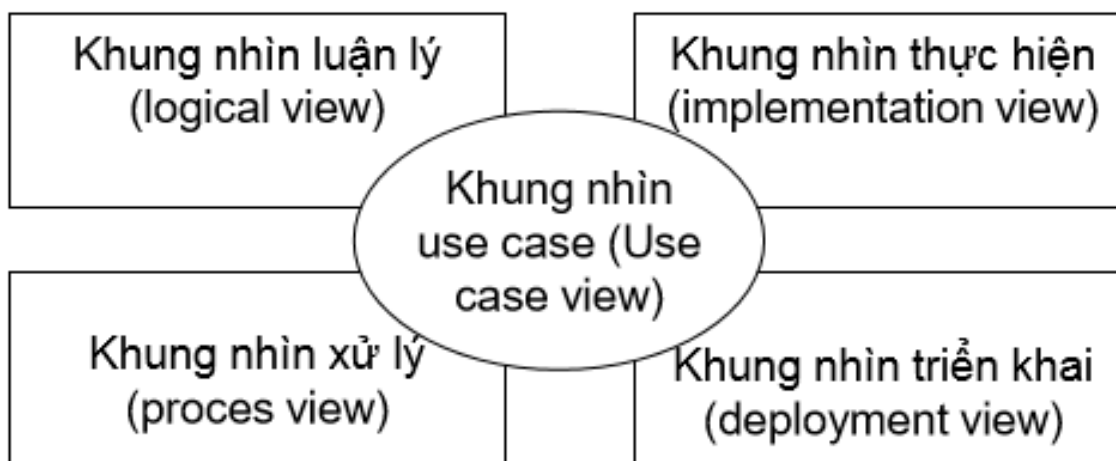




# Khung nhìn (view)

**Người dùng**  
Chức năng

**Lập trình viên**  
Quản trị phần mềm

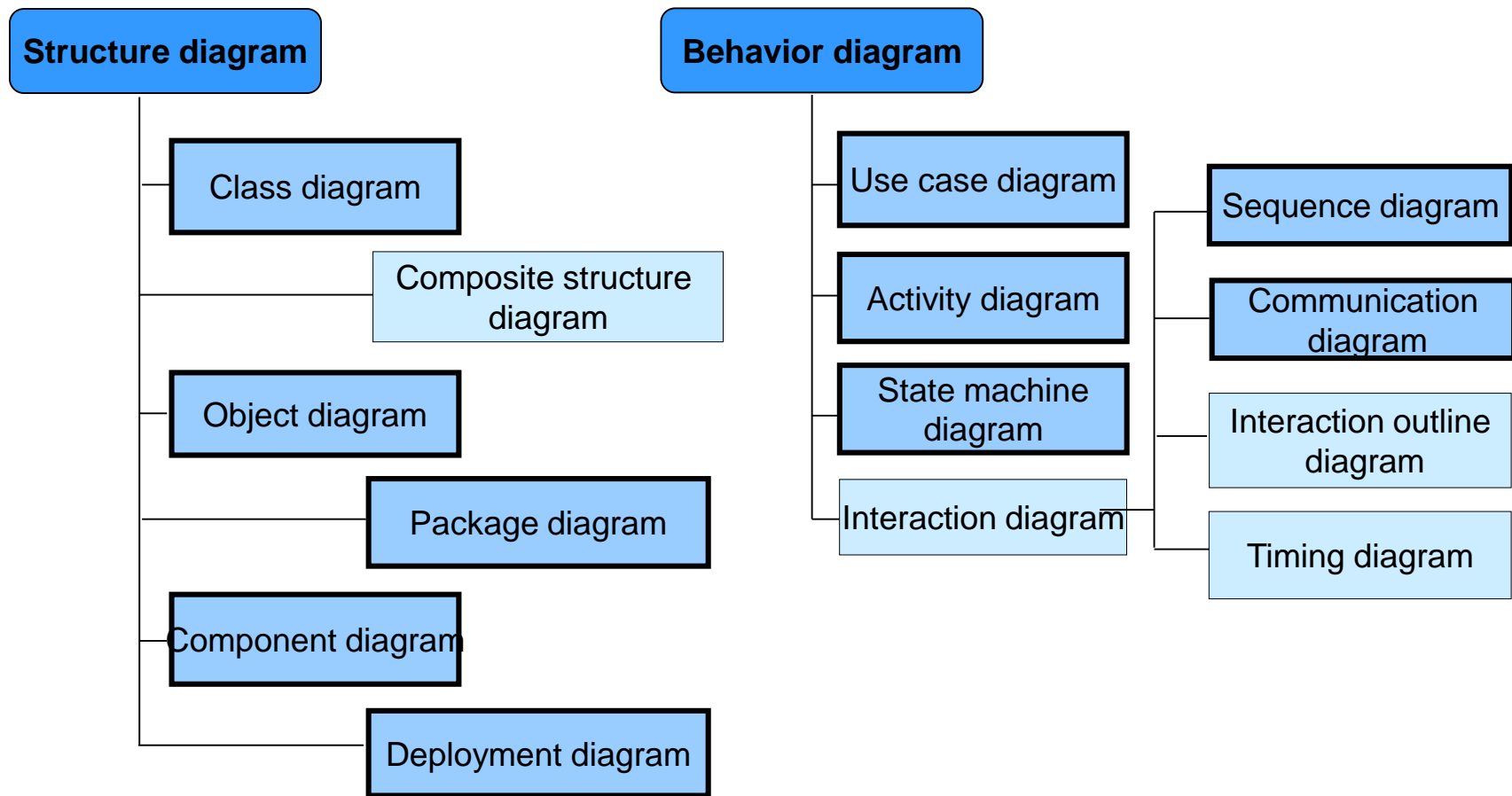


**Quản trị viên tích hợp hệ thống**  
Hiệu năng  
Tính co giãn  
Thông lượng

**Thiết kế viên hệ thống**  
Hình thái hệ thống  
Chuyển giao, cài đặt  
Truyền thông



# The outline of diagrams





# Các biểu đồ UML chính

- Các biểu đồ cấu trúc:
  - Biểu đồ lớp (Class diagram)
  - Biểu đồ đối tượng (Object diagram)
  - Biểu đồ gói (Package diagram)
  - Biểu đồ thành phần (Component diagram)
  - Biểu đồ triển khai (Deployment diagram)
- Các biểu đồ hành vi:
  - Biểu đồ ca sử dụng (Use-case diagram)
  - Biểu đồ hoạt động (Activity diagram)
  - Biểu đồ máy trạng thái (State machine diagram)
  - Biểu đồ trình tự (Sequence diagram)
  - Biểu đồ giao tiếp (Communication diagram)



# Một số biểu đồ UML cơ bản

1

❖ Biểu đồ ca  
sử dụng  
Use Case  
Diagram

Component

Deployment

Communication/  
Collaboration

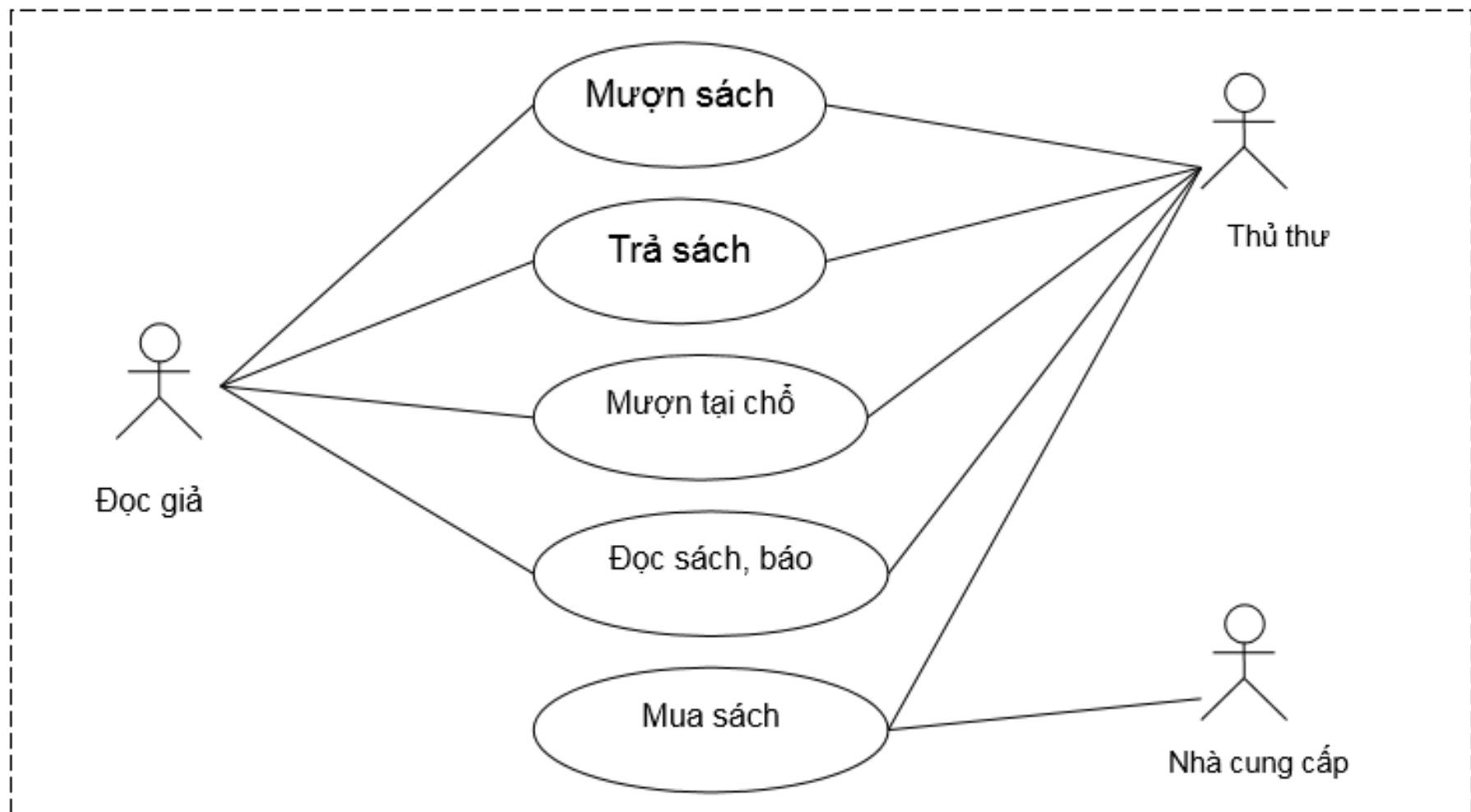
Timing

Interaction

State

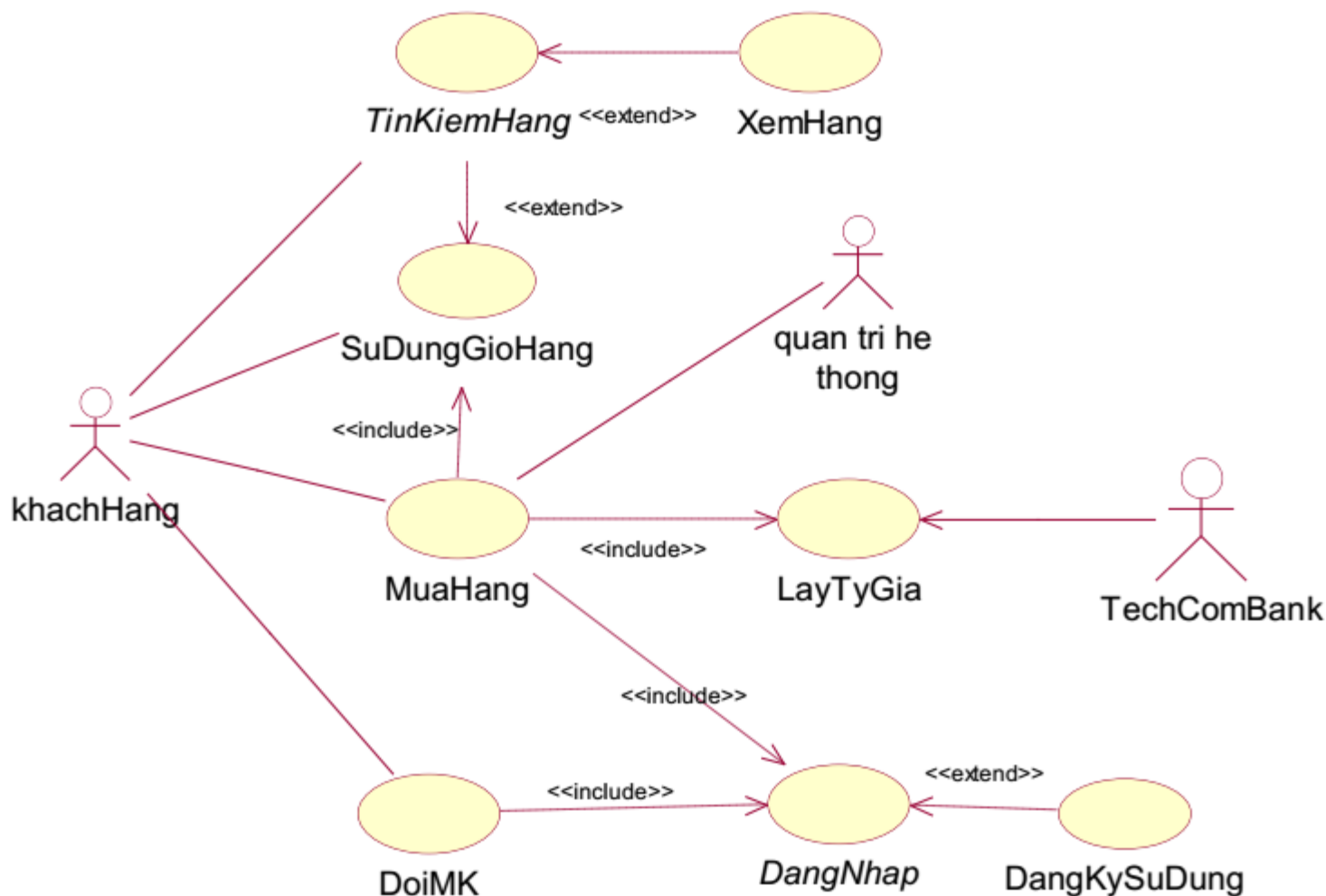


# Sơ đồ Use Case



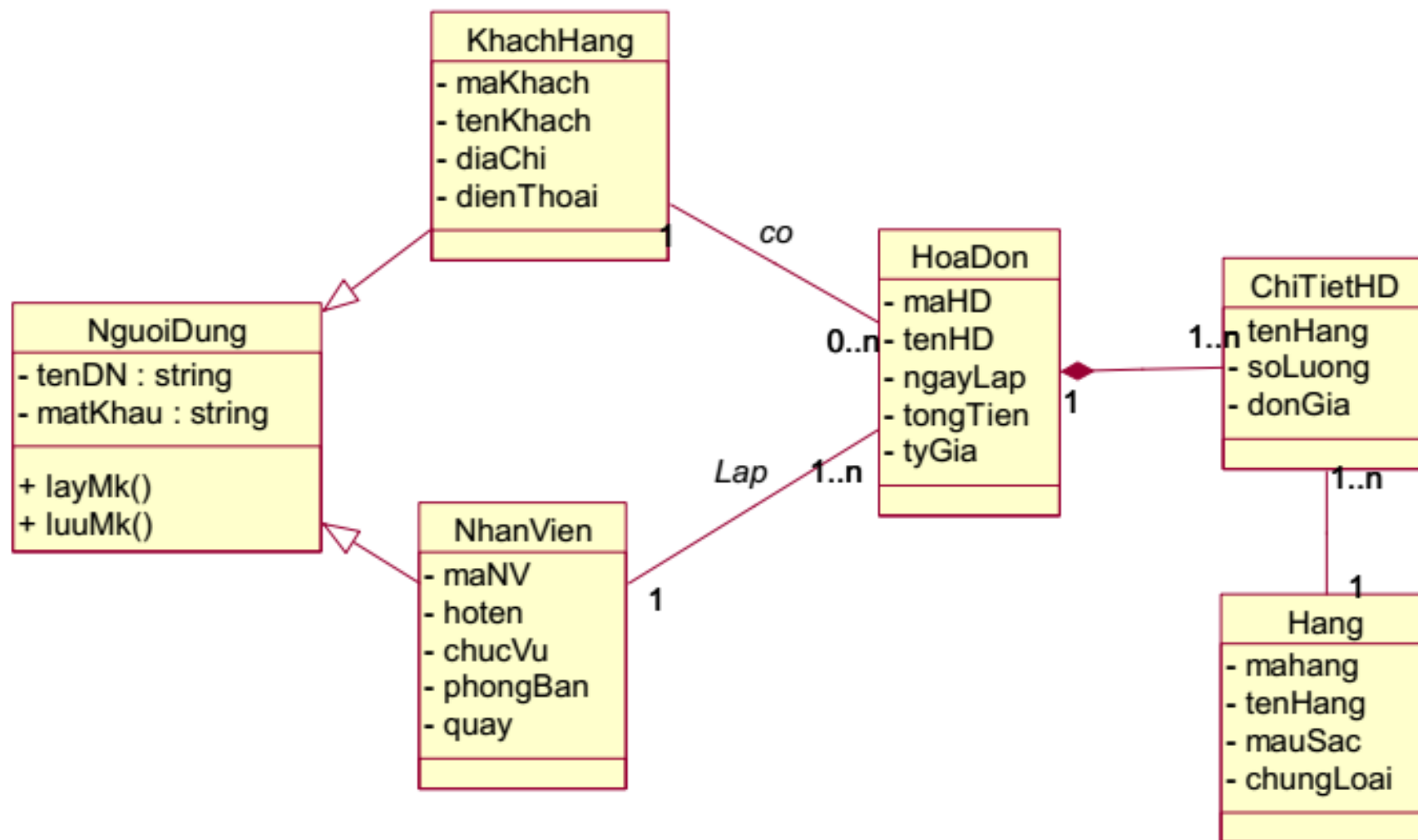


# Sơ đồ Use Case



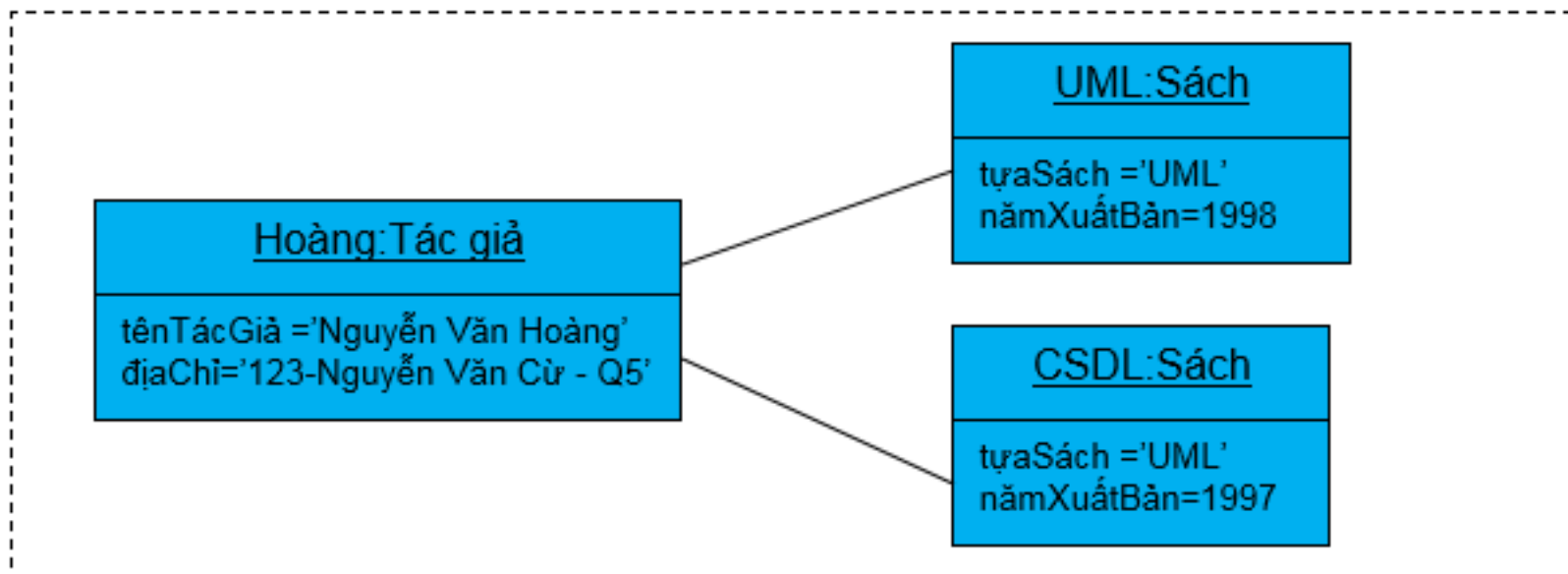
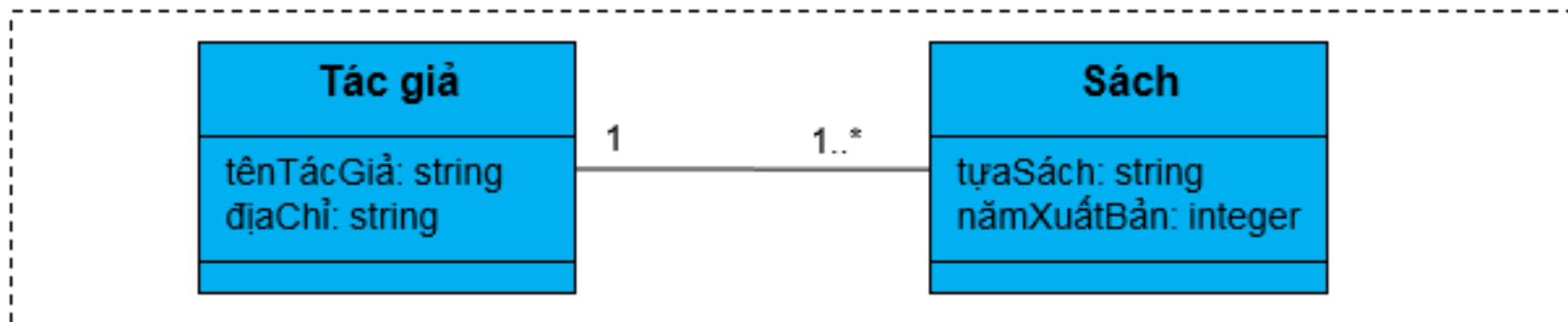


# Sơ đồ lớp





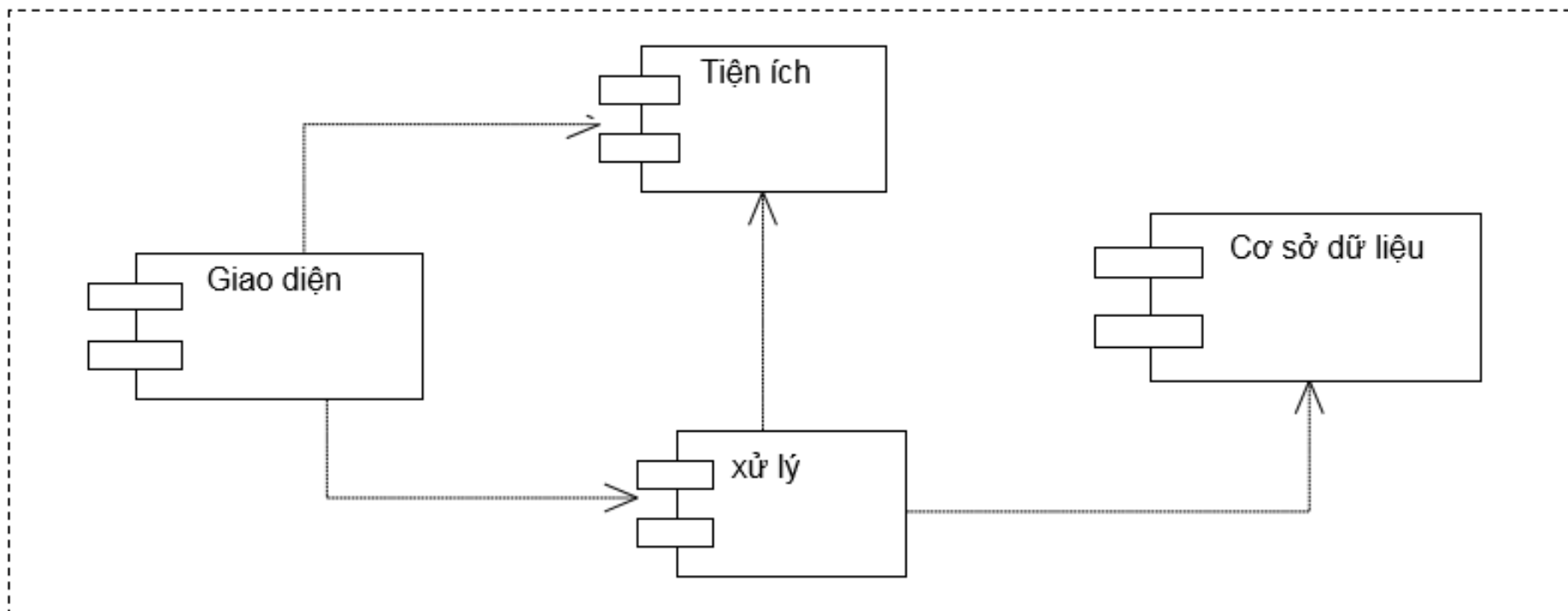
# Sơ đồ lớp và đối tượng





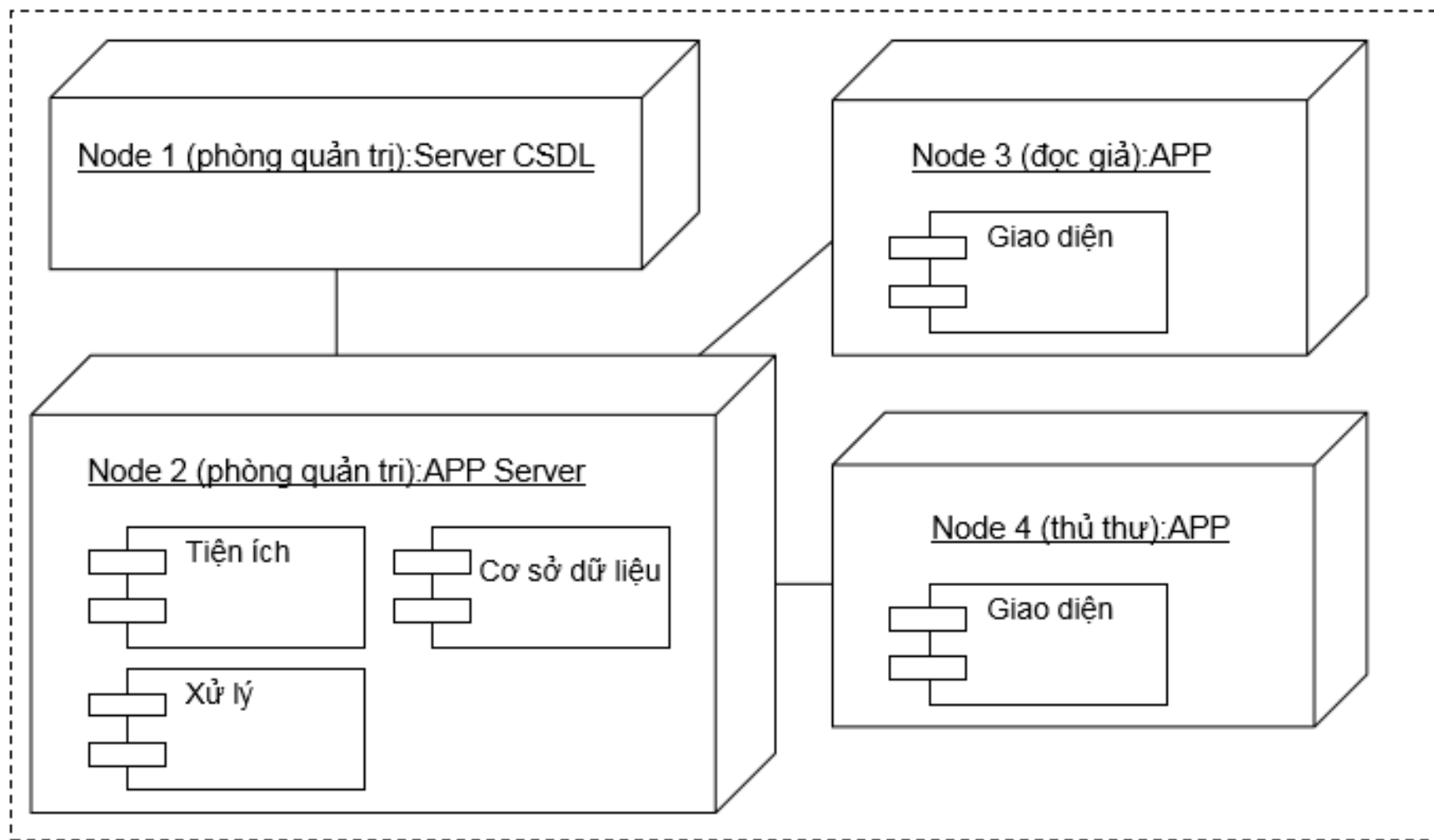


# Sơ đồ Thành phần



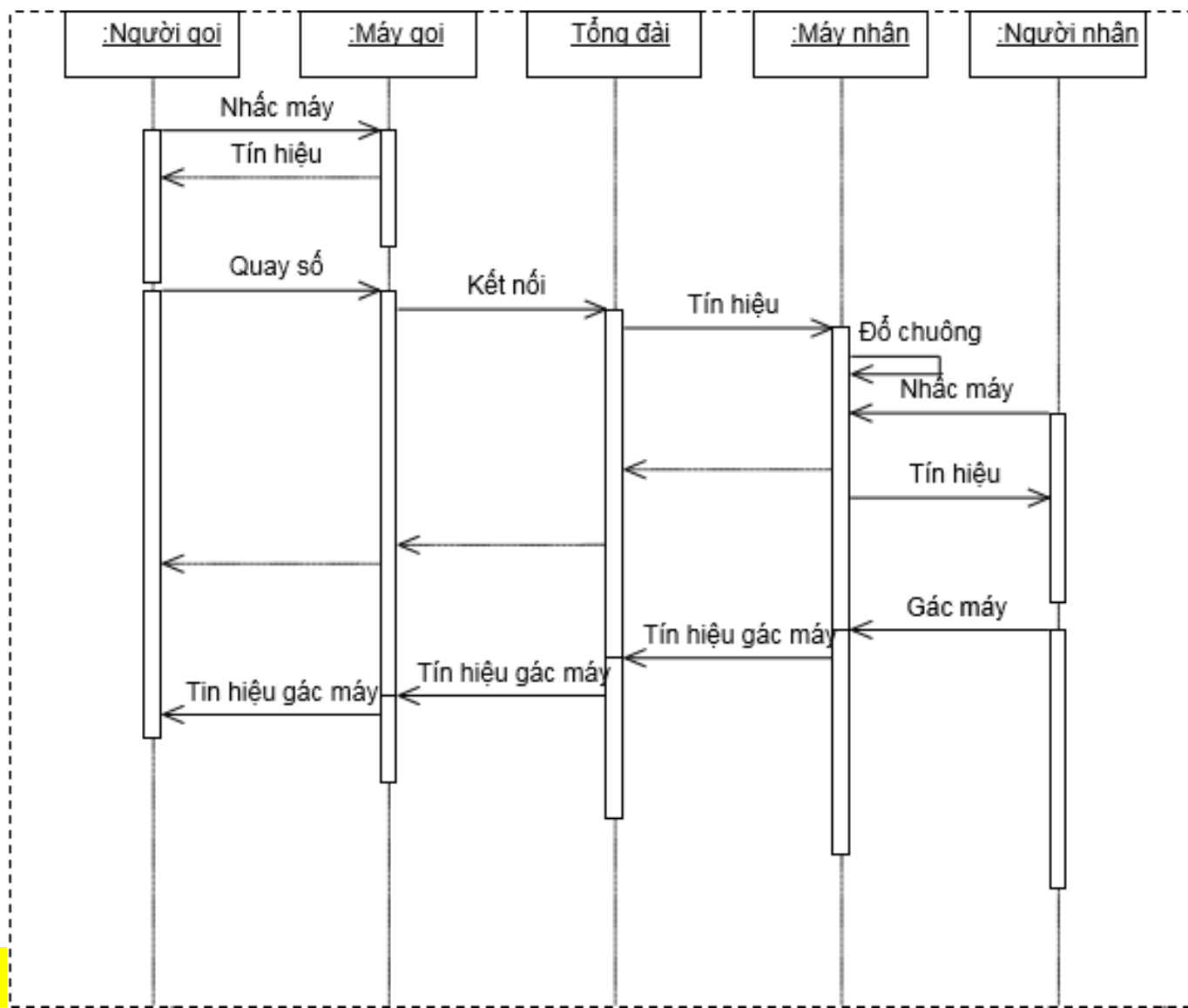


# Sơ đồ triển khai



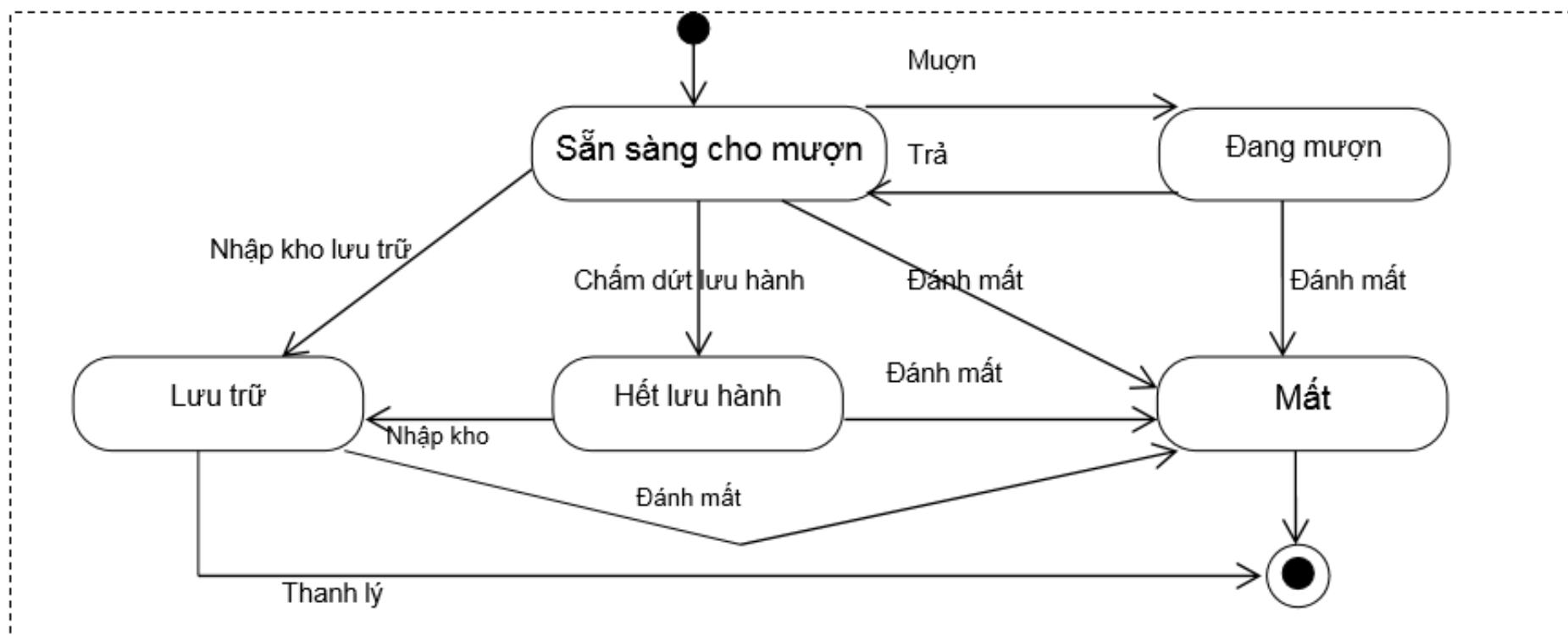


# Sơ đồ tuần tự và hợp tác



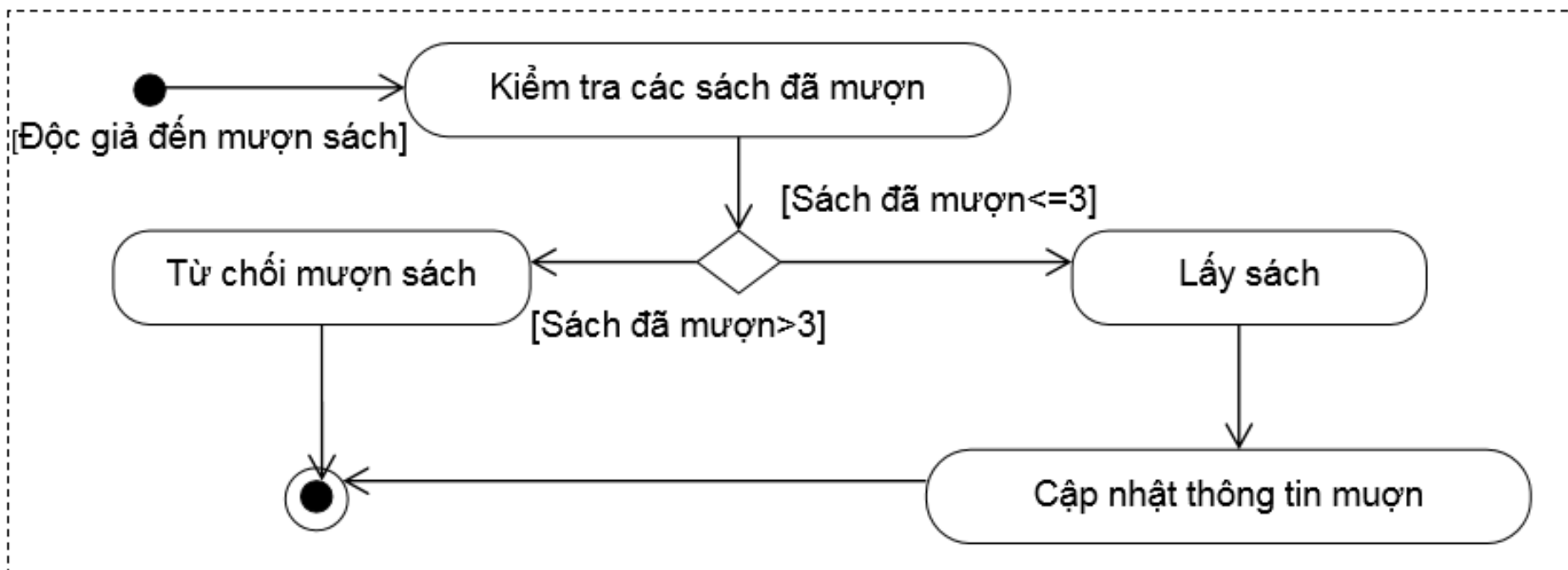


# Sơ đồ trạng thái





# Sơ đồ hoạt động





# References

Bài giảng này có tham khảo:

- Slide bài giảng “Xây dựng phần mềm hướng đối tượng”, **Trần Minh Triết**, ĐH KHTN TpHCM.
- Slide bài giảng “Phân tích Thiết kế HTTT hướng đối tượng”, **Nguyễn Trần Minh Thư**, ĐH KHTN TpHCM.
- Slide bài giảng Kỹ nghệ phần mềm, ĐH Công nghệ, ĐHQG Hà Nội.
- UML, Nguyễn Văn Quý, APTECH, Hưng Yên.



# Câu hỏi và thảo luận





**Thank you!!!**

