



# **Chương 4:**

## **Mô hình hóa yêu cầu**

# Tài liệu tham khảo

- ❖ Bài giảng dựa chính trên bài giảng “Xây dựng phần mềm hướng đối tượng” - ThS Trần Minh Triết – ĐH KHTN
- ❖ Bài giảng “Phân tích và thiết kế hướng đối tượng” – TS Phạm Ngọc Nam – ĐH BK HN
- ❖ Bài giảng “Phân tích và thiết kế hướng đối tượng” – ThS Phạm Nguyễn Cường – ĐH KHTN
- ❖ ....

# Nội dung

- ❖ Mô hình hóa yêu cầu:
  - Lược đồ Use-case
  - Khái niệm Actor và Usecase
  - Ví dụ
- ❖ Activity Diagram
- ❖ Sequence Diagram

# Mở đầu

## ❖ Đặt vấn đề:

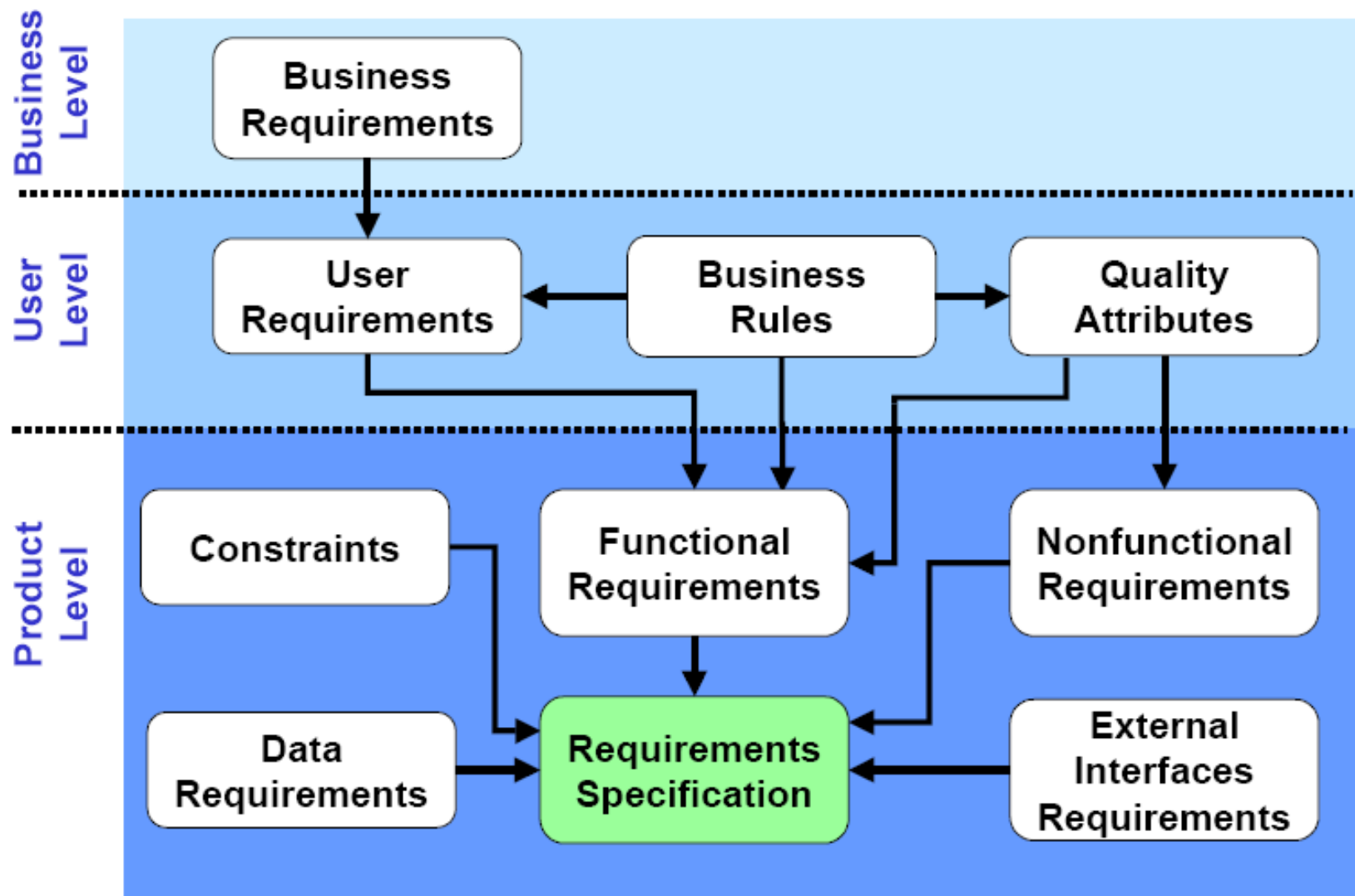
- Các mô tả về yêu cầu trong giai đoạn xác định yêu cầu:
    - Chỉ mô tả chủ yếu các thông tin liên quan đến việc thực hiện các nghiệp vụ trong thế giới thực, chưa thể hiện rõ nét việc **thực hiện** các nghiệp vụ **trên máy tính**
    - Mô tả thông qua các văn bản dễ gây ra nhầm lẫn và **không trực quan**
- ➔ Mô hình hóa yêu cầu

# Mục đích của mô hình

- ❖ Trực quan hóa hệ thống
- ❖ Chỉ rõ cấu trúc và ứng xử của hệ thống
- ❖ Cho chúng ta một khuôn mẫu để hướng dẫn chúng ta trong quá trình xây dựng hệ thống
- ❖ Đưa ra các dẫn chứng bằng tài liệu về các quyết định mà chúng ta đã đưa ra trong quá trình thiết kế hệ thống

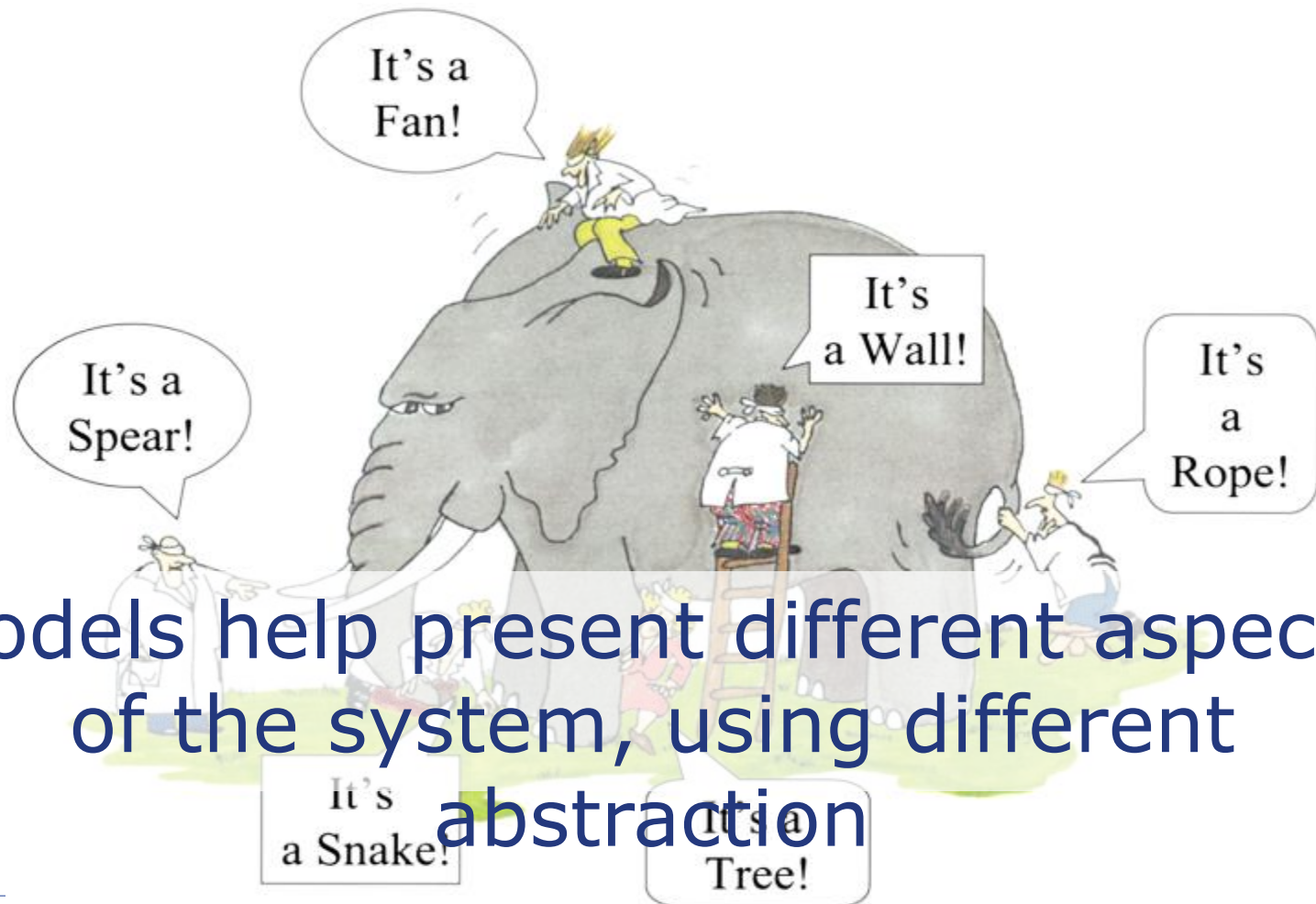
# Mô hình hóa yêu cầu

# Requirement Concepts



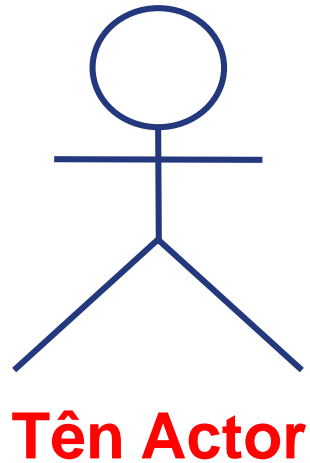
# Model different perspectives

## ❖ Model different perspectives





# Khái niệm Actor



❑ Tác nhân **BÊN NGOÀI** hệ thống

❑ Có tương tác với hệ thống



Con người



**Phần mềm**



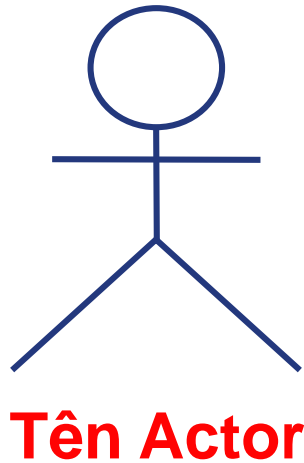
**Phần cứng**



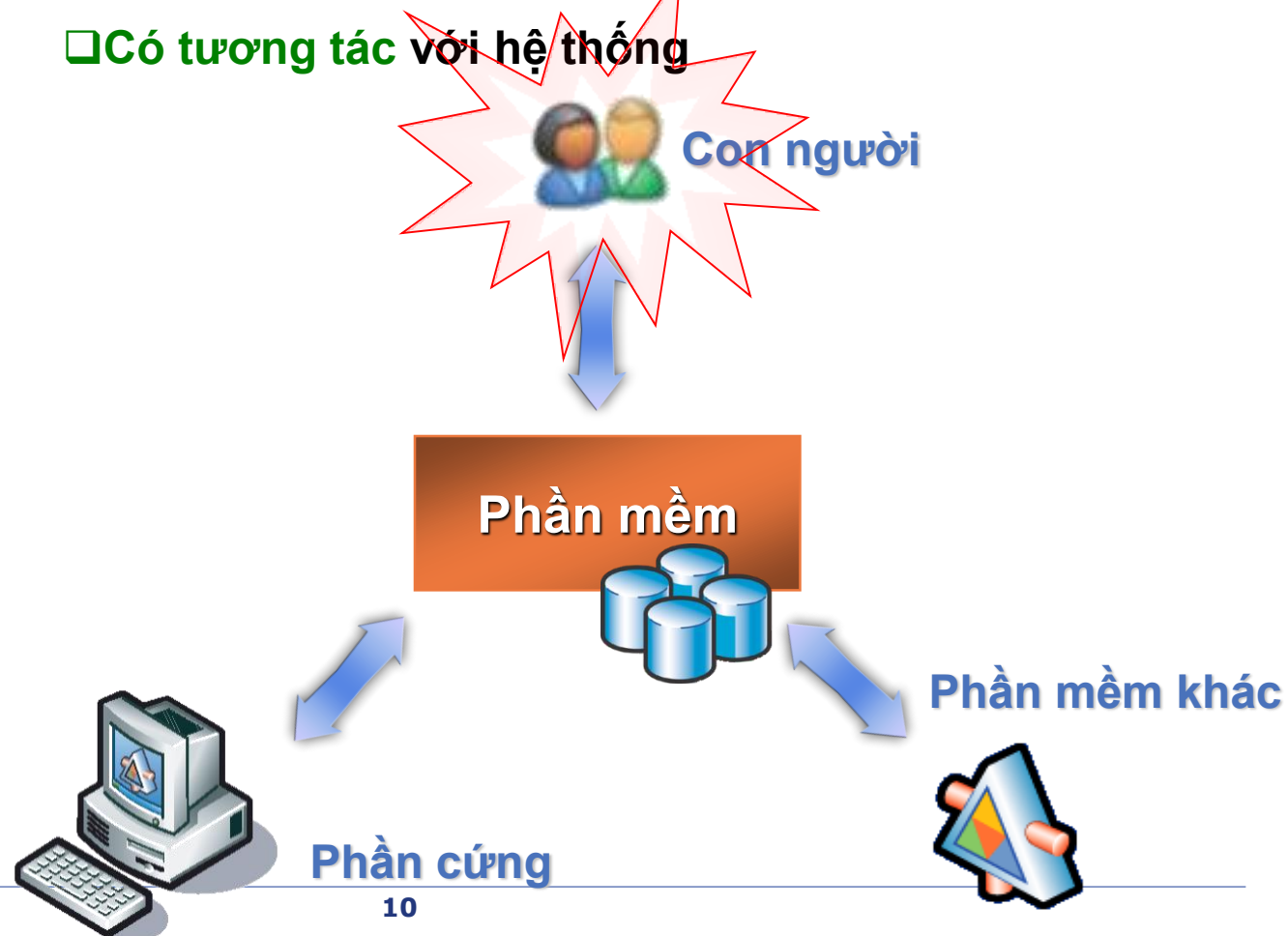
**Phần mềm khác**



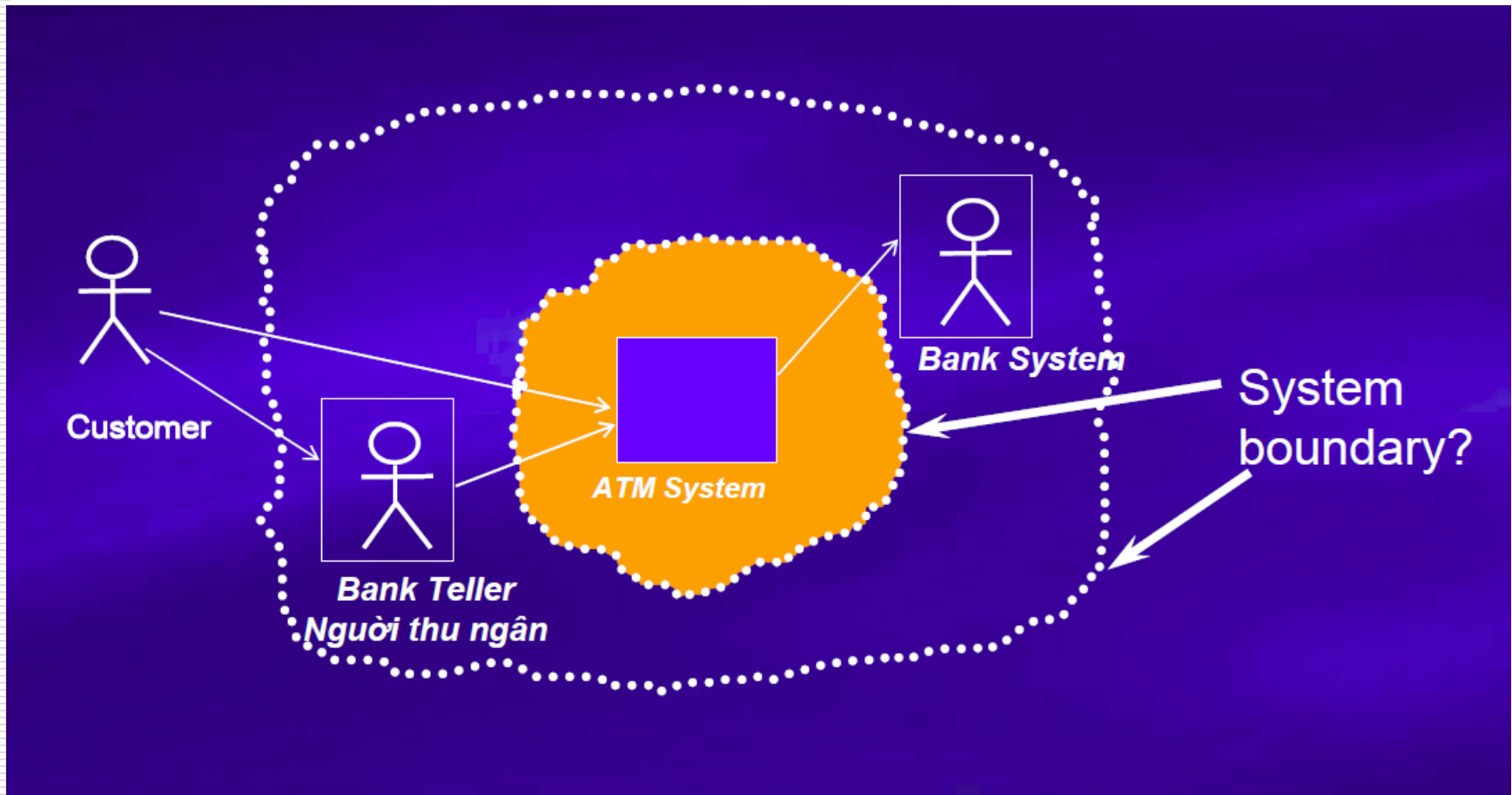
# Actor ⇔ Nhóm người sử dụng



- ❑ Tác nhân **BÊN NGOÀI** hệ thống
- ❑ Có tương tác với hệ thống

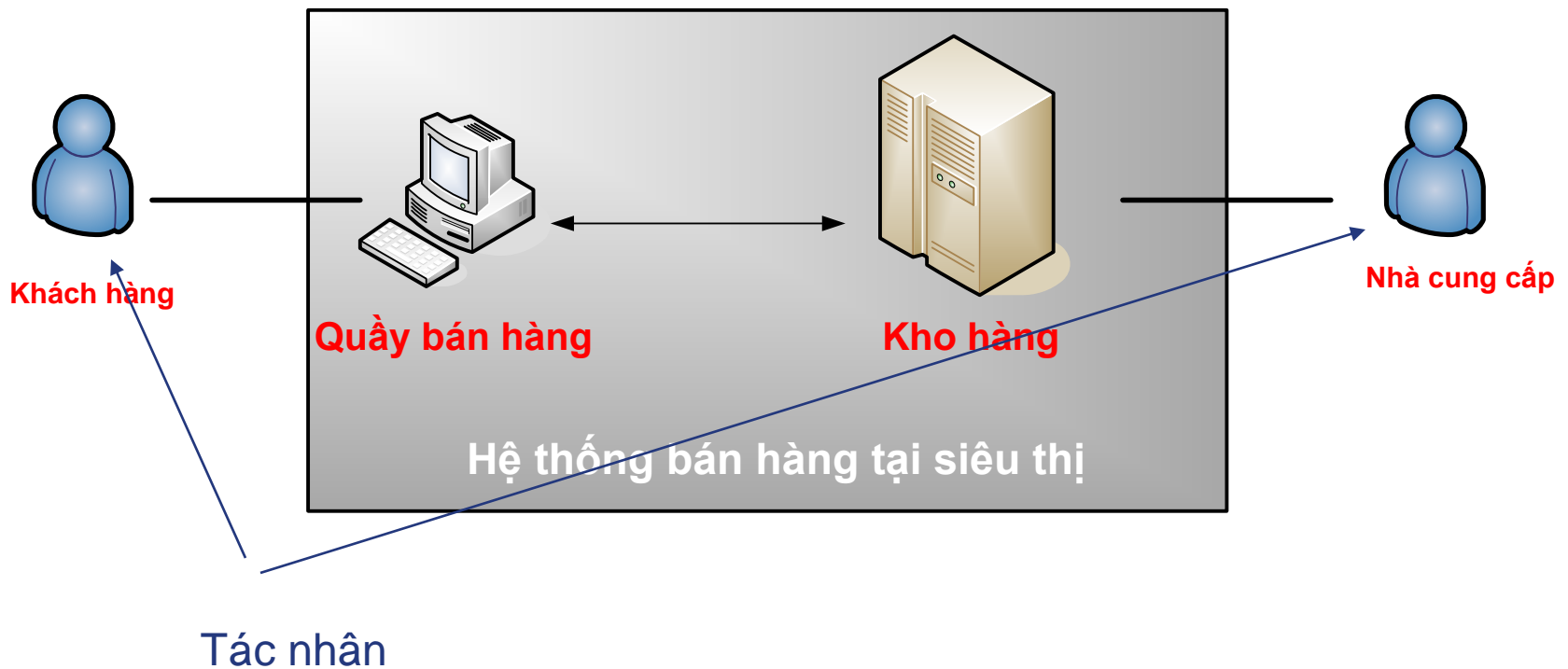


# Actors & System Boundary









System Boundary – Giới hạn của hệ thống

# Actor ⇔ Nhóm người sử dụng



# Ví dụ

## Xét phần mềm Quản lý học sinh cấp III

STT	Yêu cầu		Nhóm người dùng
1	Tiếp nhận học sinh		Giáo vụ?
2	Lập danh sách lớp		Giáo vụ?
3	Tra cứu học sinh		Mọi người? Phụ huynh? Học sinh?
4	Nhận bảng điểm môn		Giáo viên? Giáo vụ?
5	Xem báo cáo tổng kết		Ban giám hiệu?
6	Thay đổi quy định		Ban giám hiệu? Quản trị hệ thống?







# Ví dụ

## Xét phần mềm Quản lý học sinh cấp III

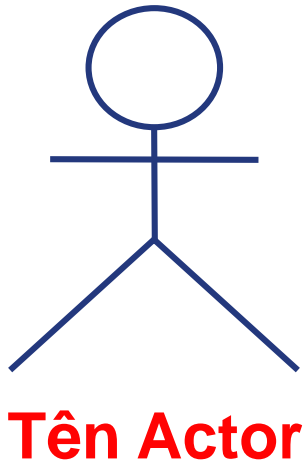
- ❖ Một nhóm người dùng tương ứng với một Actor
  - ❖ Mỗi Nhóm người dùng (Actor) được quyền sử dụng một hay nhiều chức năng trong hệ thống
  - ❖ Một chức năng có thể cho phép nhiều Nhóm người dùng sử dụng
  - ❖ Nhiều nhóm người dùng có cùng các quyền hạn giống nhau
- ➔ Nên xét là 1 Actor hay nhiều Actor?
- ❖ Việc xác định Actor phụ thuộc ngữ cảnh và quy trình thực tế

# Ví dụ

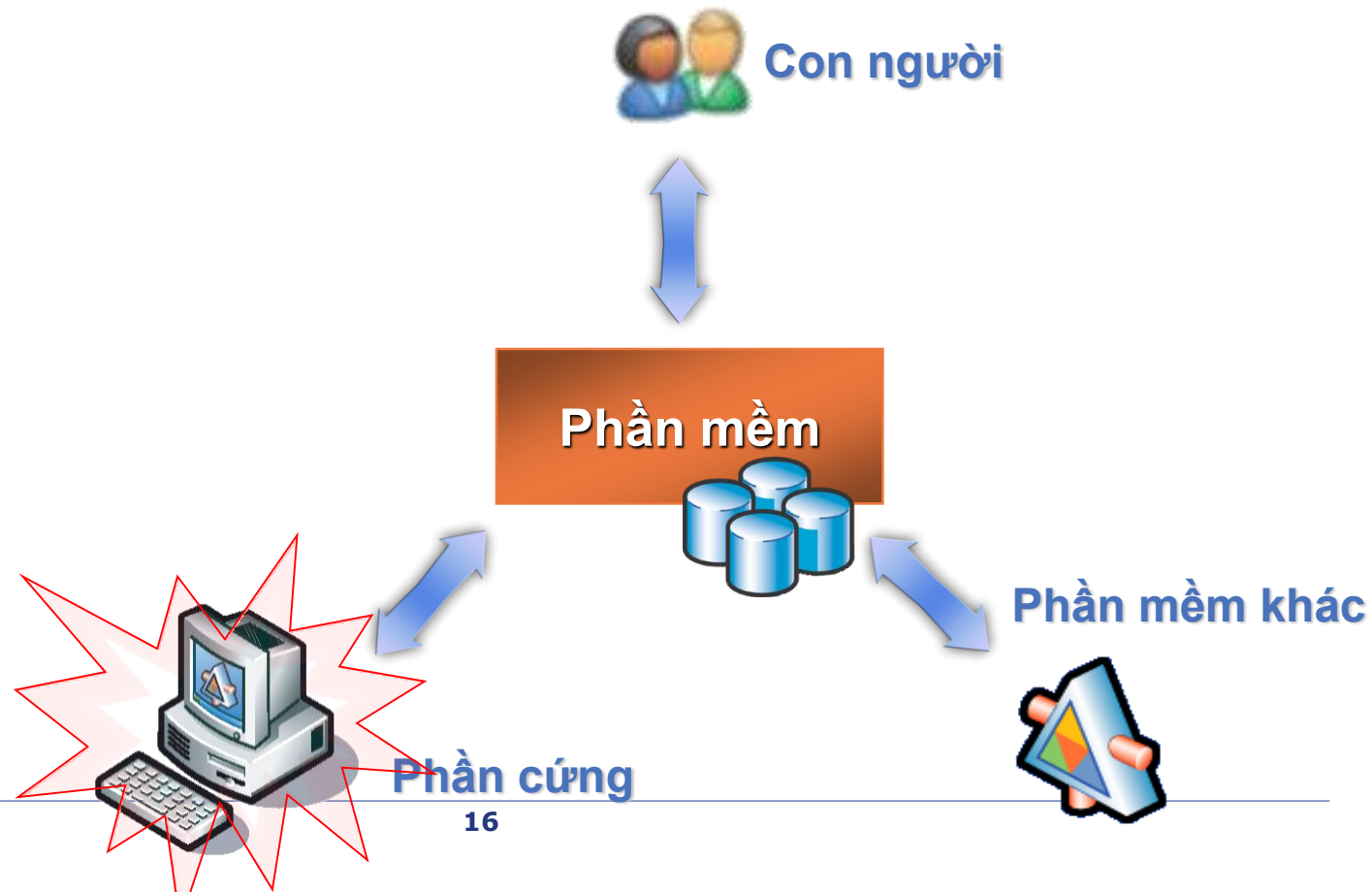
## Xét phần mềm Quản lý thư viện

STT	Yêu cầu		Nhóm người dùng
1	Lập thẻ độc giả		Thủ thư
2	Nhận sách mới		Thủ thư
3	Tra cứu học sinh		Thủ thư? Độc giả? Khách bất kỳ?
4	Lập phiếu mượn		Thủ thư
5	Nhận trả sách		Thủ thư
6	Thay đổi quy định		Thủ thư? Quản trị hệ thống?

# Actor ⇔ Phần cứng ngoại vi



- ❑ Tác nhân **BÊN NGOÀI** hệ thống
- ❑ Có tương tác với hệ thống






# Ví dụ

## ❖ Ví dụ:

- Phần mềm quản lý Siêu thị:
  - Đọc thông tin từ thiết bị đọc mã vạch
- Phần mềm quản lý cửa tự động:
  - Đọc thông tin từ camera
  - Phát lệnh điều khiển mở cửa
- Phần mềm quản lý ra vào các phòng trong công sở
  - Đọc tín hiệu từ đầu đọc thẻ từ
  - Phát lệnh điều khiển mở cửa
- Phần mềm chống trộm
  - Đọc tín hiệu từ camera, sensor
  - Phát lệnh điều khiển ra loa, đèn, điện thoại...

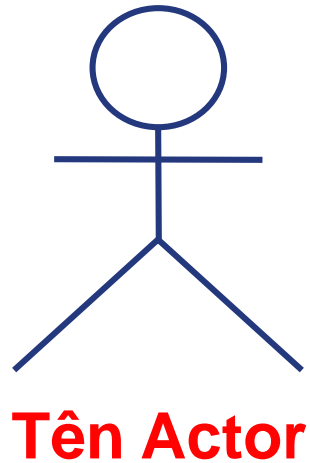


Các thiết bị ngoại vi  
mà phần mềm  
cần tương tác



Có cần liệt kê  
tất cả thiết bị ngoại vi?

# Actor $\Leftrightarrow$ Phần mềm khác



❑ Tác nhân **BÊN NGOÀI** hệ thống

❑ Có tương tác với hệ thống



Con người



Phần mềm



Phần cứng

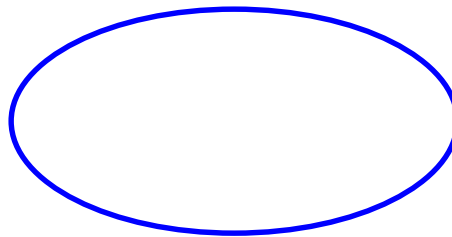
Phần mềm khác



# Ví dụ

- ❖ Kết xuất/nạp dữ liệu từ Excel
- ❖ Kết xuất dữ liệu báo cáo ra phần mềm gửi email (Microsoft Outlook, Outlook Express...)
- ❖ Phần mềm trung gian kết nối để chuyển đổi email từ dạng Web-based sang POP3 (ví dụ Yahoo!Pop)
- ❖ ...

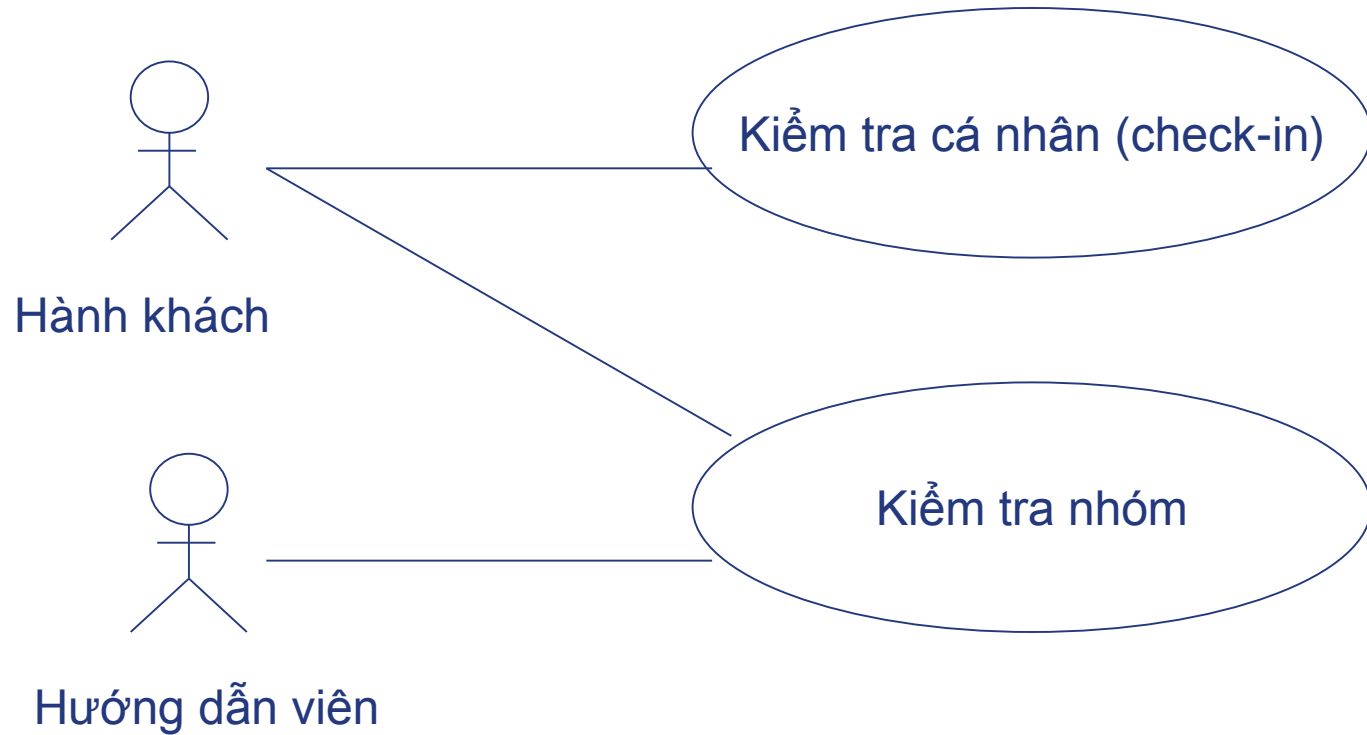
# Khái niệm Use-Case



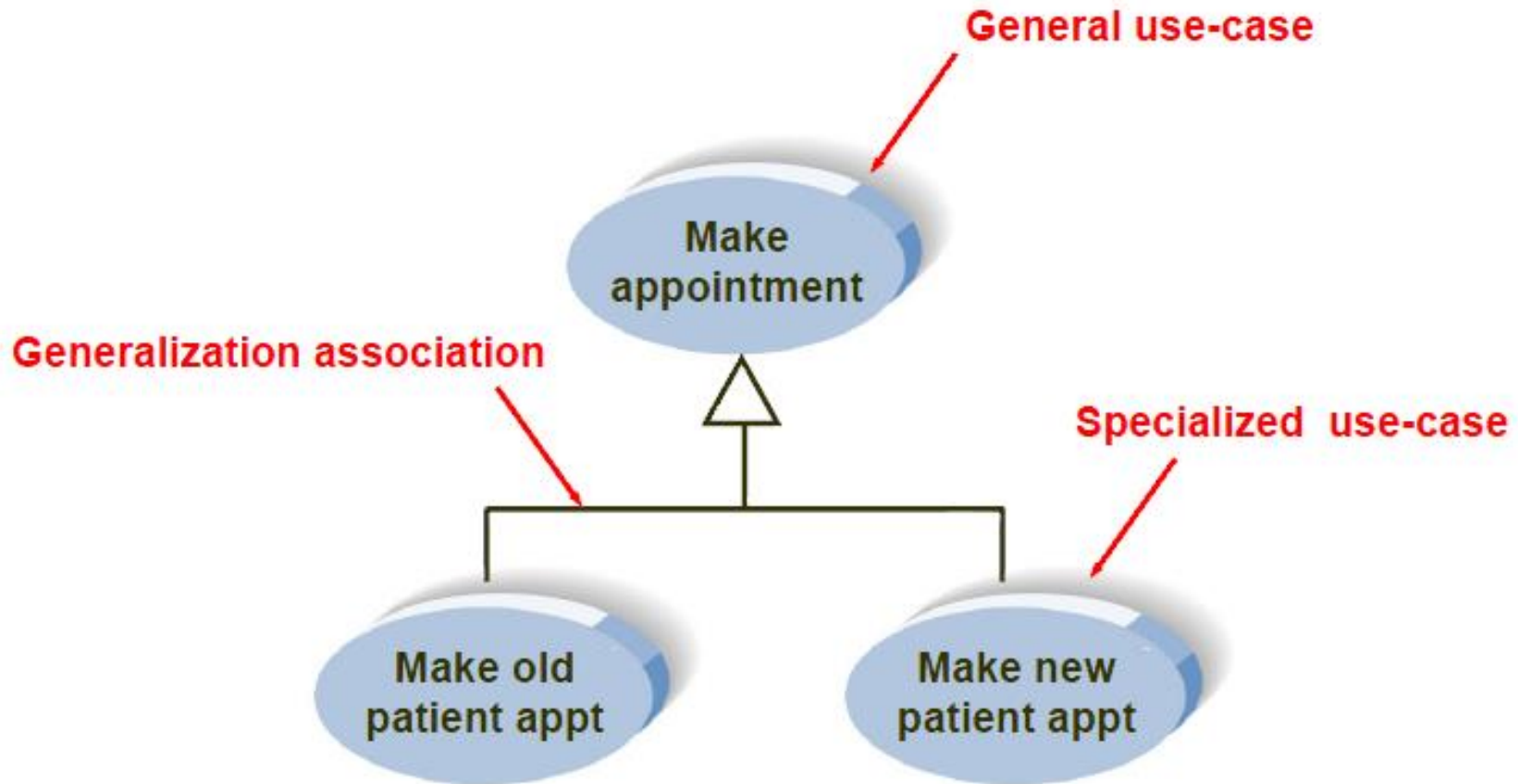
## Use-Case

- Một **Use-Case** là một chuỗi các hành động mà hệ thống thực hiện mang lại một kết quả quan sát được đối với actor.
- Có thể hiểu một Use-Case là một chức năng của hệ thống, mang một ý nghĩa nhất định đối với người dùng

# Use - Case



# Tổng quát hóa giữa các *use-case*



# Ví dụ

## Xét phần mềm Quản lý thư viện

STT	Yêu cầu
1	Lập thẻ đọc giả
2	Nhận sách mới
3	Tra cứu học sinh
4	Lập phiếu mượn
5	Nhận trả sách
6	Mua thêm sách

❖ Có bao nhiêu Use-case trong Ví dụ này?

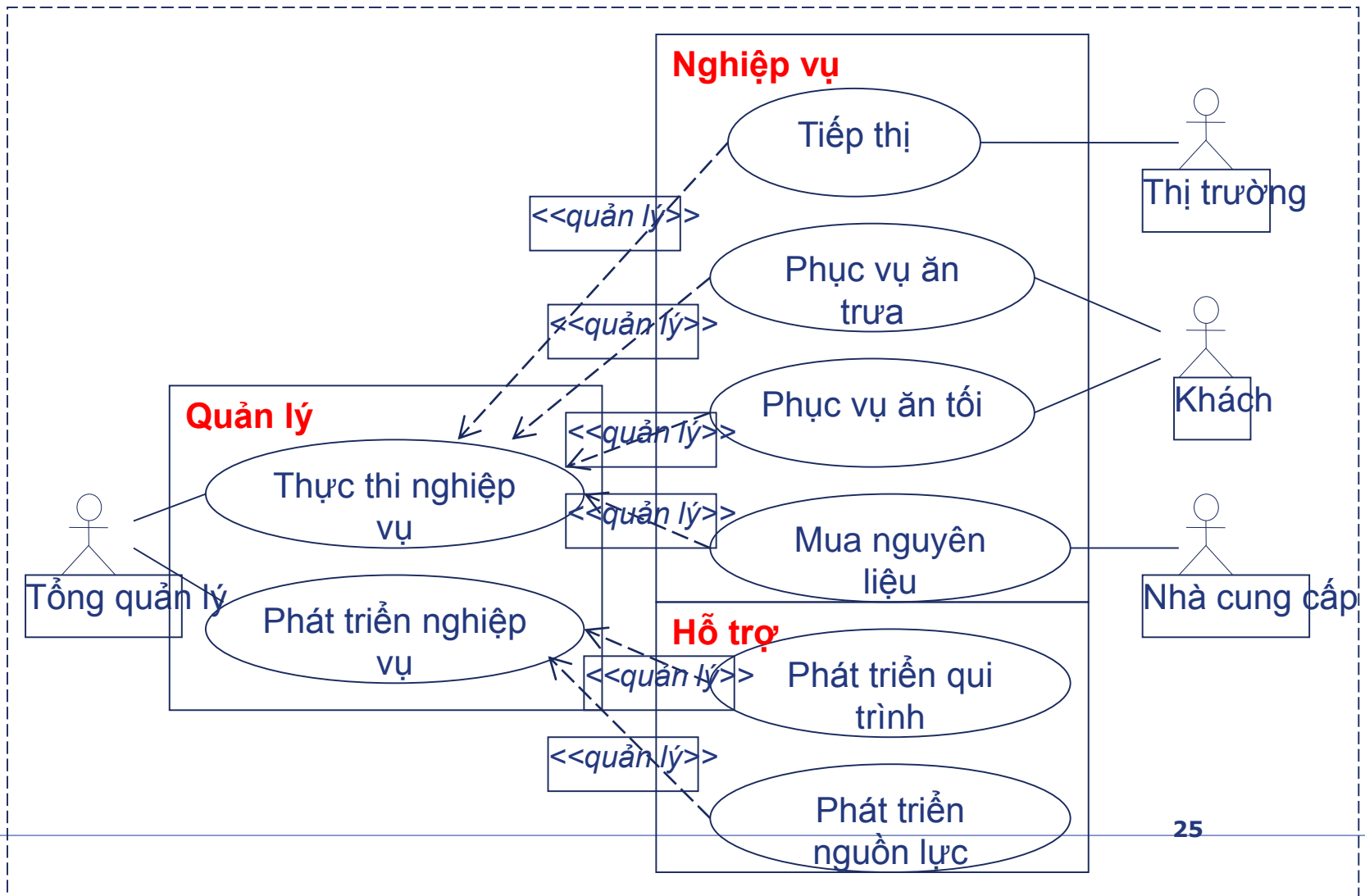


Các use case của hệ thống thư viện



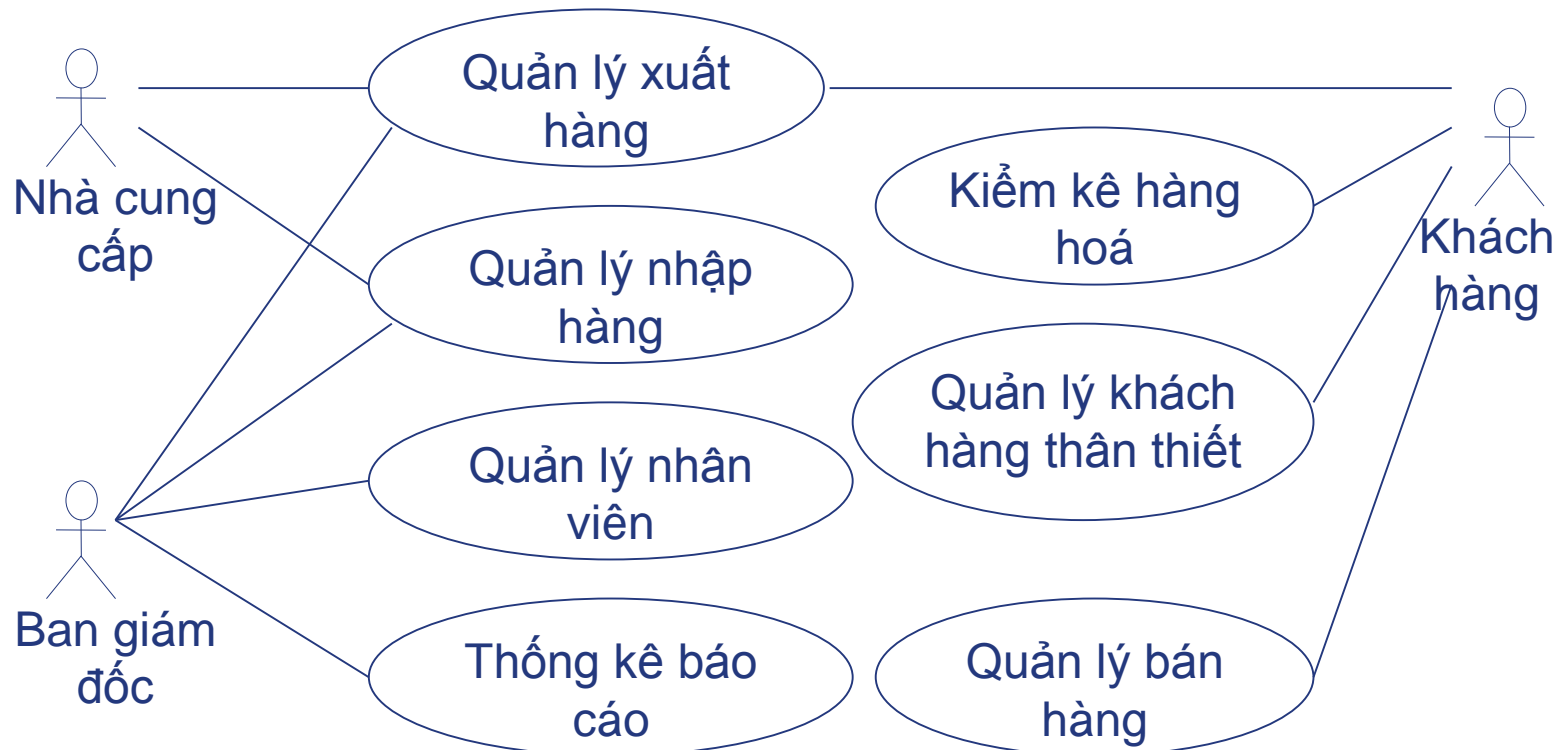
# Xác định tác nhân và use case nghiệp vụ

Ví dụ: các use case trong một tổ chức nhà hàng



# Mô hình use case

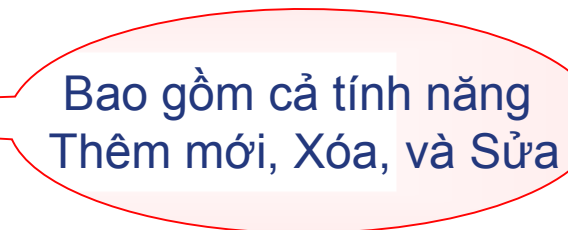
Ví dụ: mô hình use case của siêu thị - sơ đồ ngữ cảnh



# Ví dụ

## Xét phần mềm Quản lý học sinh cấp III

STT	Yêu cầu
1	Tiếp nhận học sinh
2	Lập danh sách lớp
3	Tra cứu học sinh
4	Nhận bảng điểm môn
5	Xem báo cáo tổng kết
6	Thay đổi quy định



Bao gồm cả tính năng  
Thêm mới, Xóa, và Sửa

❖ Có bao nhiêu Use-case trong Ví dụ này?

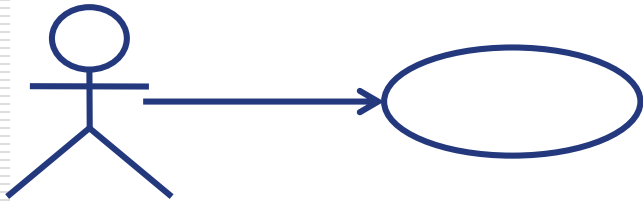
# Ví dụ

## Phần mềm thí nghiệm mạch điện

STT	Yêu cầu
1	Sắp đặt mạch điện
2	Cung cấp nguồn điện
3	Thay đổi thông số
4	Lưu bài thí nghiệm
5	Lấy lại thí nghiệm
6	Thay đổi quy định

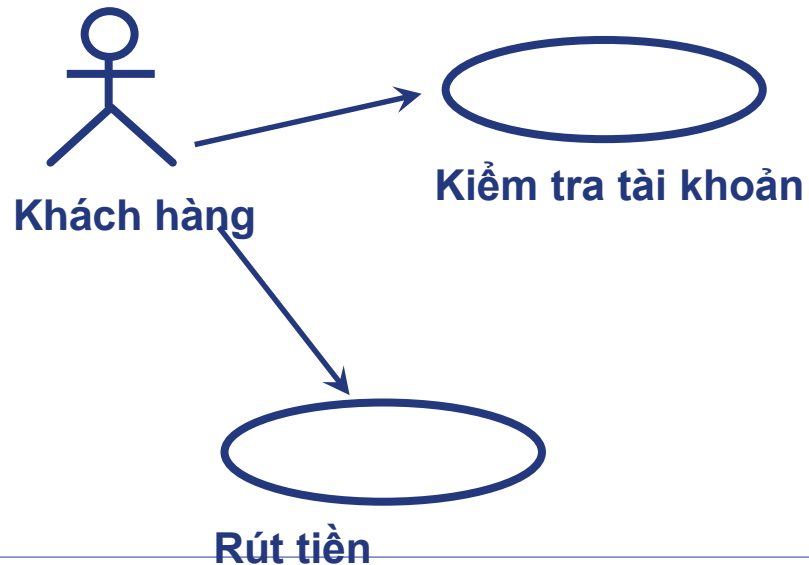
❖ Có bao nhiêu Use-case trong Ví dụ này?

# Sơ đồ Use-case

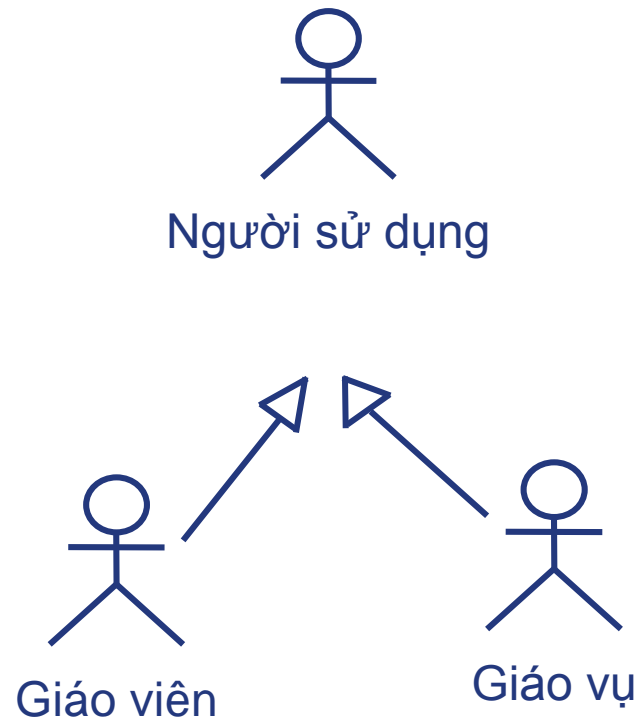


Sự tương tác giữa Actor và Use-case

Chiều của mũi tên thể hiện vai trò chủ động trong sự tương tác



# Tổng quát hóa giữa các Actor



# Mô tả use case

- ❖ Nội dung của một use case được mô tả ban đầu theo hai phần:

Giới thiệu về use case

Các dòng cơ bản (basic flow): bao gồm các hoạt động chính và thứ tự mô tả nội dung chính của use case

Các thay thế (alternative flow): mô tả các nhánh hoạt động bất thường để xử lý ngoại lệ ngoài các dòng chính

## ❖ Ví dụ: mô tả use case mượn sách

Use case bắt đầu khi một có đọc giả đến mượn sách. Mục tiêu của use case nhằm xử lý mượn sách cho đọc giả

### Các dòng cơ bản:

1. Xác định thẻ thư viện của đọc giả: nhân viên yêu cầu đọc giả xuất trình thẻ thư viện để kiểm tra
2. Xác định thông tin nợ sách: kiểm tra thông tin các sách đang nợ của đọc giả
3. Ghi nhận thông tin lần mượn: cập nhật vào hệ thống thông tin về lần mượn của đọc giả
4. Gởi sách cho đọc giả và thông báo ngày giới hạn trả sách



# Mô tả use case

## ❖ Ví dụ: mô tả use case mượn sách

Các dòng thay thế:

- Xử lý thẻ hết hạn: nếu thẻ sinh viên của đọc giả hết hạn, thủ thư sẽ thông báo cho đọc giả và yêu cầu làm thẻ mới
- Xử lý không cho mượn: nếu số lượng sách mà đọc giả đang mượn  $> 3$ , thủ thư sẽ từ chối lần mượn của đọc giả

## ❖ Ví dụ: mô tả use case Đăng ký

Use case mô tả hoạt động đăng kiểm tại quầy đăng ký khi hành khách tới để đăng ký đi chuyến bay của mình.

### Các dòng cơ bản:

1. Tìm kiếm chỗ ngồi: sau khi nhận vé từ hành khách, nhân viên sẽ tìm một chỗ ngồi cho hành khách từ hệ thống. Hệ thống sẽ đánh dấu chỗ đó không còn trống.
2. In thẻ lên máy bay: in thẻ lên máy bay cho hành khách.
3. Xử lý hành lý: kiểm tra và xác nhận hành lý, in ra thẻ đánh dấu hành lý và thẻ kiểm soát hành lý cho nhân viên.

### Các dòng thay thế

1. Xử lý hành lý đặc biệt: xử lý các hành lý chứa một loại hàng đặc biệt hoặc quá nặng (được mô tả trong use case ....).

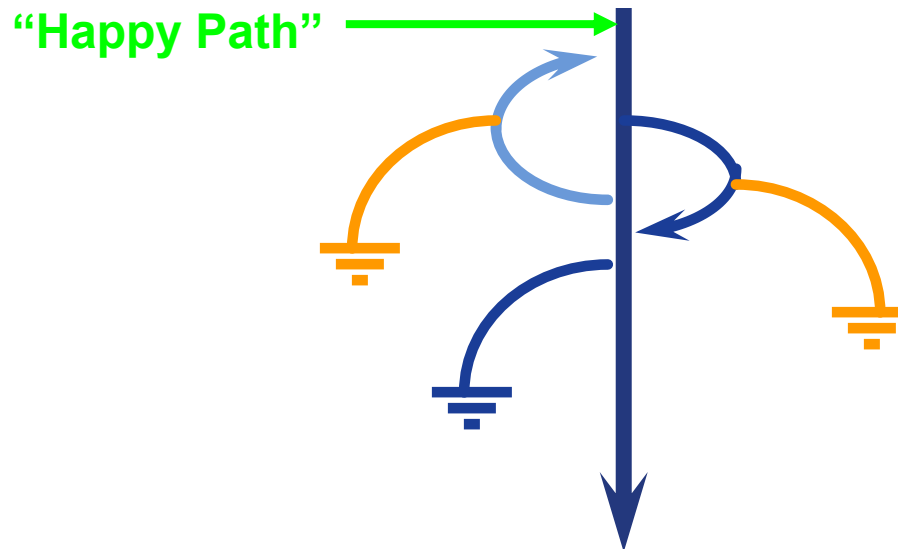
# Mô tả Use-case

## Ví dụ: mô tả use case Rút tiền ATM

1. Use-Case bắt đầu khi khách hàng đưa thẻ tín dụng vào. Hệ thống đọc và thẩm tra thông tin của thẻ.
2. Hệ thống nhắc nhập số PIN. Hệ thống kiểm tra số PIN.
3. Hệ thống hỏi tác vụ nào khách hàng muốn thực hiện. Khách hàng chọn “Rút tiền.”
4. Hệ thống hỏi số lượng. Khách hàng nhập số lượng.
5. Hệ thống yêu cầu nhập kiểu tài khoản. Khách hàng chọn checking hoặc savings.
6. Hệ thống liên lạc với ATM network . . .

# Mô tả use-case

- ❖ *basic flow* (“Happy Path”)
- ❖ Một số *alternative flows*
  - Các biến thể thường gặp (Regular variants)
  - Các trường hợp bất thường (Odd cases)
  - **Exceptional flows** xử lý các tình huống lỗi



## Các câu hỏi thường dùng để xác định actor

- ❖ Đối với một vấn đề cụ thể nào đó thì Ai là người quan tâm ?
- ❖ Hệ thống được dùng ở nơi nào trong tổ chức?
- ❖ Ai là người được lợi khi sử dụng hệ thống?
- ❖ Ai là người cung cấp thông tin cho hệ thống, sử dụng thông tin của hệ thống và xóa các thông tin đó?
- ❖ Ai là người hỗ trợ và bảo trì hệ thống?

## Các câu hỏi thường dùng để xác định actor

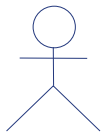
- ❖ Hệ thống có sử dụng nguồn lực nào từ bên ngoài?
- ❖ Có người nào đóng một vài vai trò trong hệ thống? *(Có thể phân thành 2 actor)*
- ❖ Có vai trò nào mà nhiều người cùng thể hiện?  
*Có thể chỉ là một actor*
- ❖ Hệ thống có tương tác với các hệ thống nào khác không?

# Các nhóm Actor

- ❖ Nhóm đòi hỏi sự trợ giúp của hệ thống trong hoạt động của mình.
- ❖ Nhóm cần thiết cho sự thực hiện các chức năng của hệ thống.
- ❖ Nhóm các thiết bị hay hệ thống mềm có liên quan đến hệ thống.
- ❖ Nhóm thực hiện các chức năng phụ trợ như quản trị duy tu hệ thống

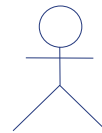
# Các nhóm Actor

- ❖ Nhóm đòi hỏi sự trợ giúp của hệ thống trong hoạt động của mình.



**Khách  
hàng**

**Trong hệ thống ATM**



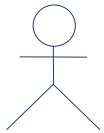
**Thủ thư**

**Trong hệ thư viện**



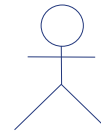
# Các nhóm Actor

- ❖ Nhóm cần thiết cho sự thực hiện các chức năng của hệ thống.



**Nhân  
viên vận  
hành**

**Trong hệ thống ATM**

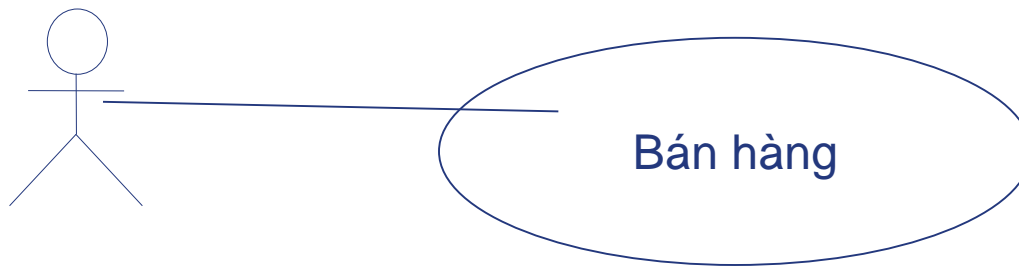


**Quản trị  
hệ thống**

**Trong hệ thư viện**

# Các nhóm Actor

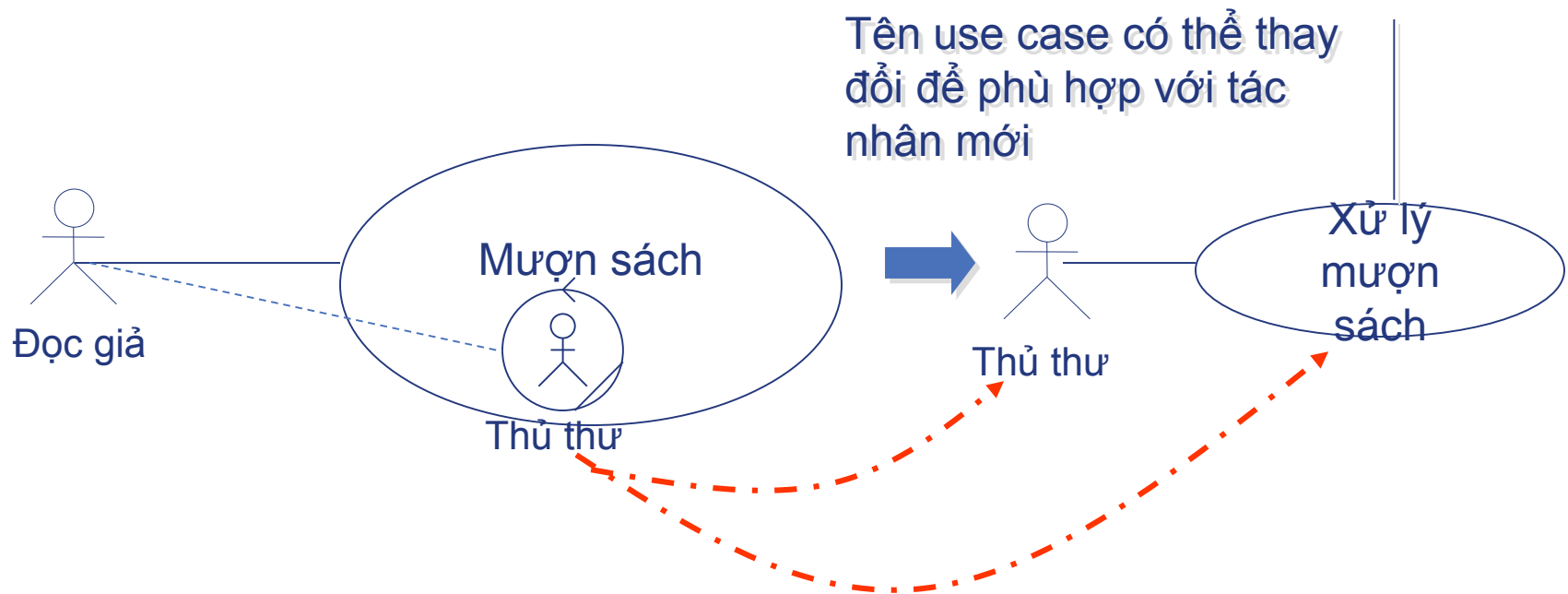
- ❖ Nhóm các thiết bị hay hệ thống mềm có liên quan đến hệ thống.



# Xác định tác nhân hệ thống

## ❖ Xác định tác nhân (actor)

- Xác định tác nhân từ kết quả của mô hình hoá nghiệp vụ
  - Ví dụ: trong hệ thống thư viện

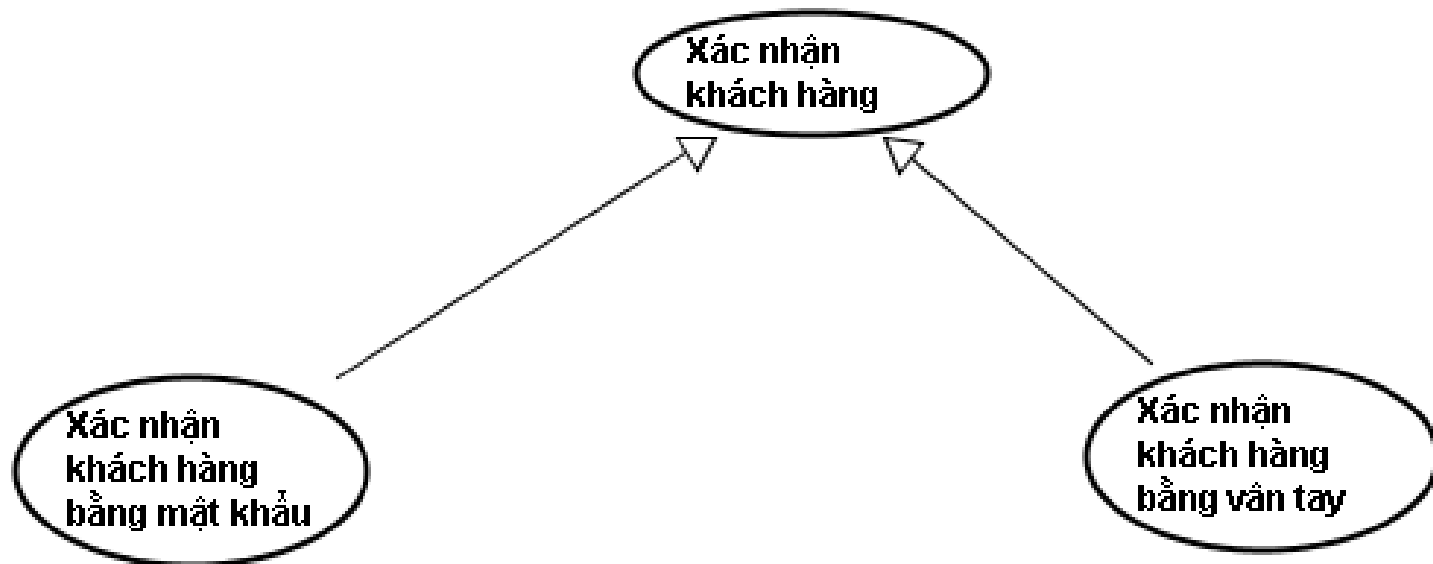


## Các câu hỏi thường dùng xác định Use Case

- ❖ Nhiệm vụ của mỗi actor là gì?
- ❖ Có actor nào sẽ tạo, lưu trữ, thay đổi, xóa hoặc đọc thông tin trong hệ thống?
- ❖ Có actor nào cần báo tin cho hệ thống về một thay đổi đột ngột từ bên ngoài?
- ❖ Có actor nào cần được thông báo về một sự việc cụ thể xảy ra trong hệ thống?
- ❖ Use - Case nào sẽ hỗ trợ và bảo trì hệ thống?
- ❖ Tất cả các yêu cầu về mặt chức năng có được thể hiện hết thông qua các trường hợp sử dụng chưa?

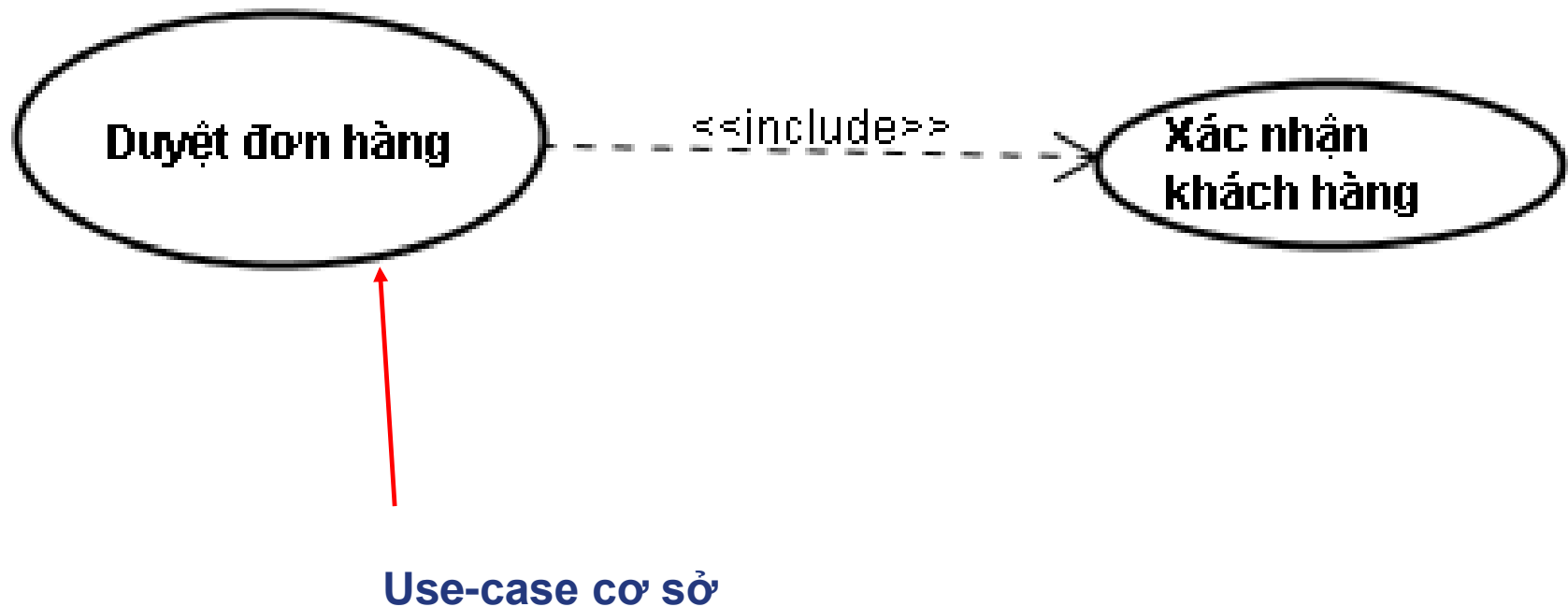
# Liên quan giữa các Use - Case

## ❖ Liên quan khái quát hóa



# Liên quan giữa các Use - Case

## ❖ Liên quan bao hàm:

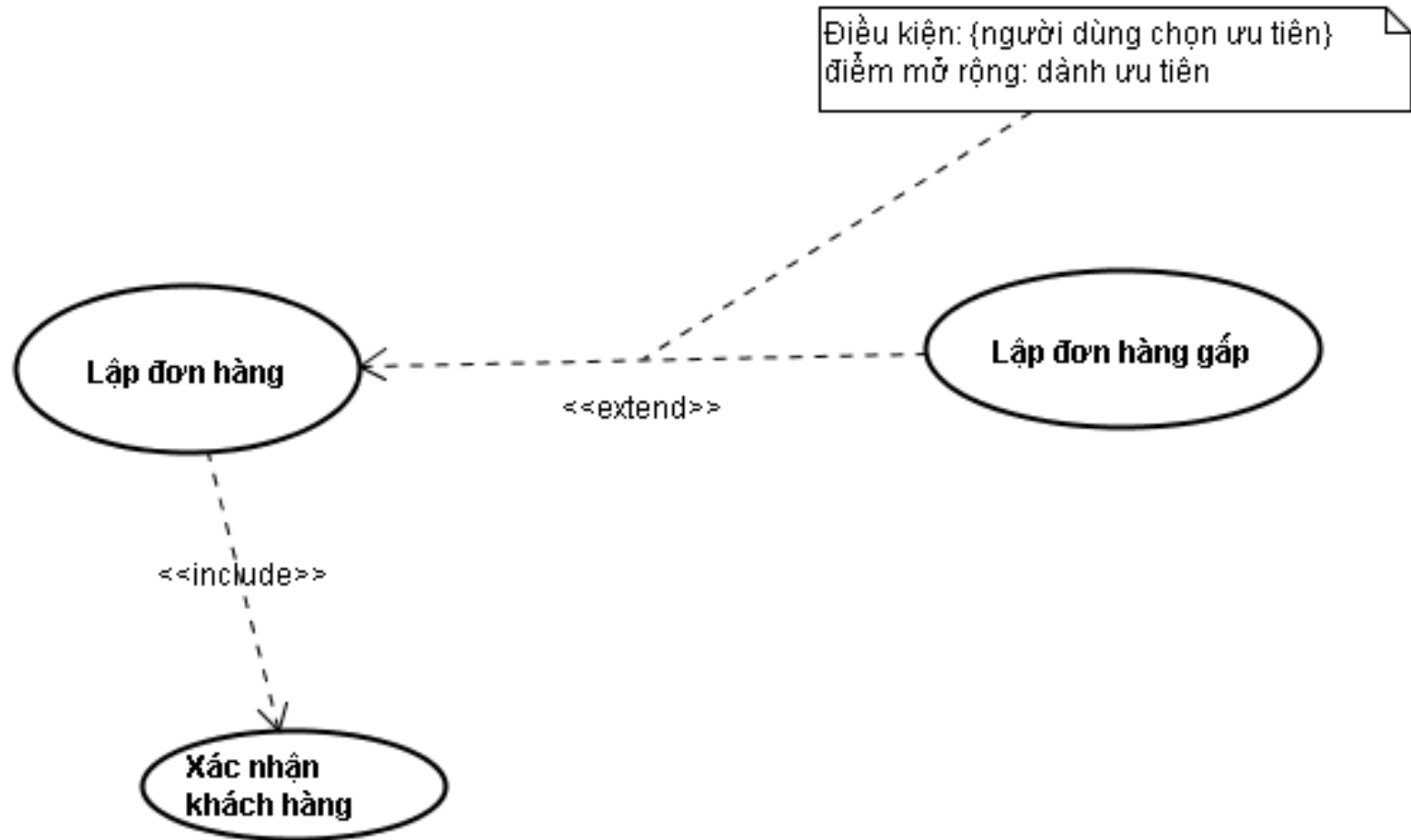


## <<include>>

- ❖ Mỗi liên kết <<include>> là một kết nối giữa hai *use-case*, cho thấy một *use-case* được sử dụng bởi một *use-case* khác
- ❖ Liên kết đến các chức năng thường dùng, được nhiều *use-case* khác sử dụng
- ❖ Mũi tên đi từ *use-case* cơ sở (*base usecase*) đến *use-case* được sử dụng (*used use-case*).

# Liên quan giữa các Use - Case

## ❖ Liên quan mở rộng:



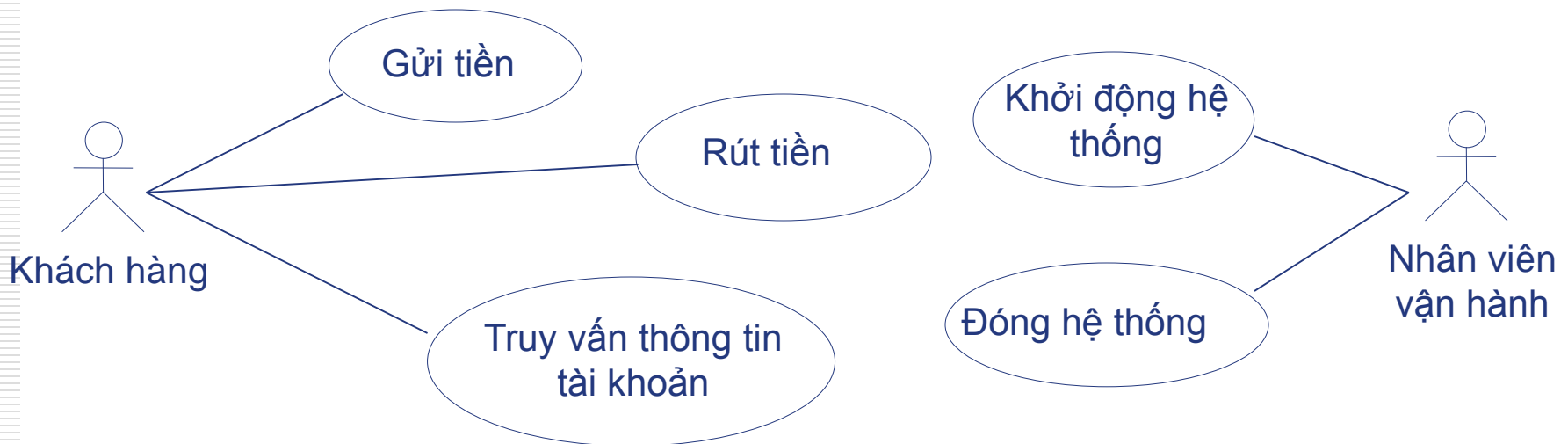


## <<extend>>

- ❖ Mỗi liên kết <<extend>> là một kết nối giữa hai *use-case*, mở rộng một *use-case* bằng cách thêm vào một hành vi mới.
- ❖ *Use-case chuyên biệt hóa (specialized use-case)* mở rộng *use-case phổ quát (general use-case)*.
- ❖ Mũi tên đi từ *use-case chuyên biệt hóa* đến *use-case phổ quát*.

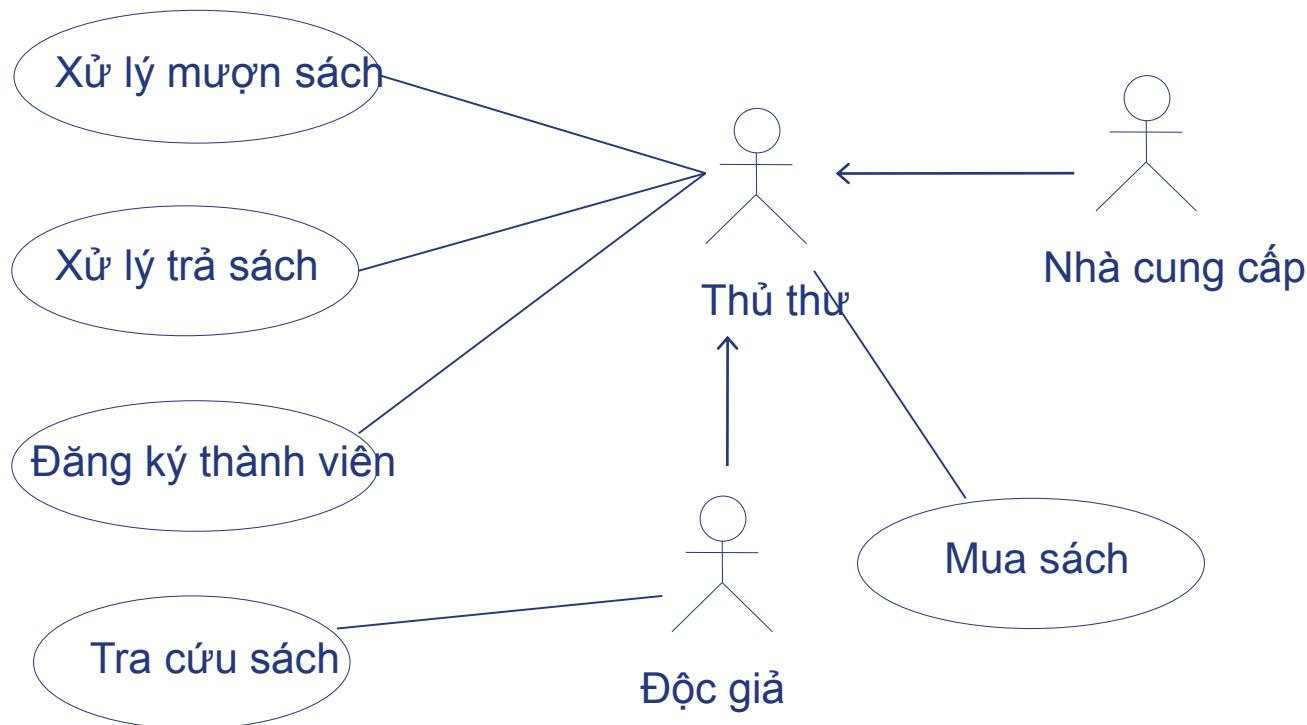
# Xác định use case

- ❖ Xác định use case thông qua tác nhân
  - Tìm các nhiệm vụ và chức năng mà tác nhân sẽ thi hành hoặc hệ thống cần tác nhân để thi hành và mô hình hoá nó như là use case
  - Ví dụ: hệ thống ATM



# Xác định use case

- ❖ Xác định use case thông qua tác nhân
  - Ví dụ: hệ thống thư viện



# Xác định use case

- ❖ Xác định use case thông qua tác nhân
  - Đặt tên use case: tên use case nên đặt nhằm phản ánh một mô tả tổng quan về chức năng của use case
    - Tên use case = động từ (do) + danh từ (what)

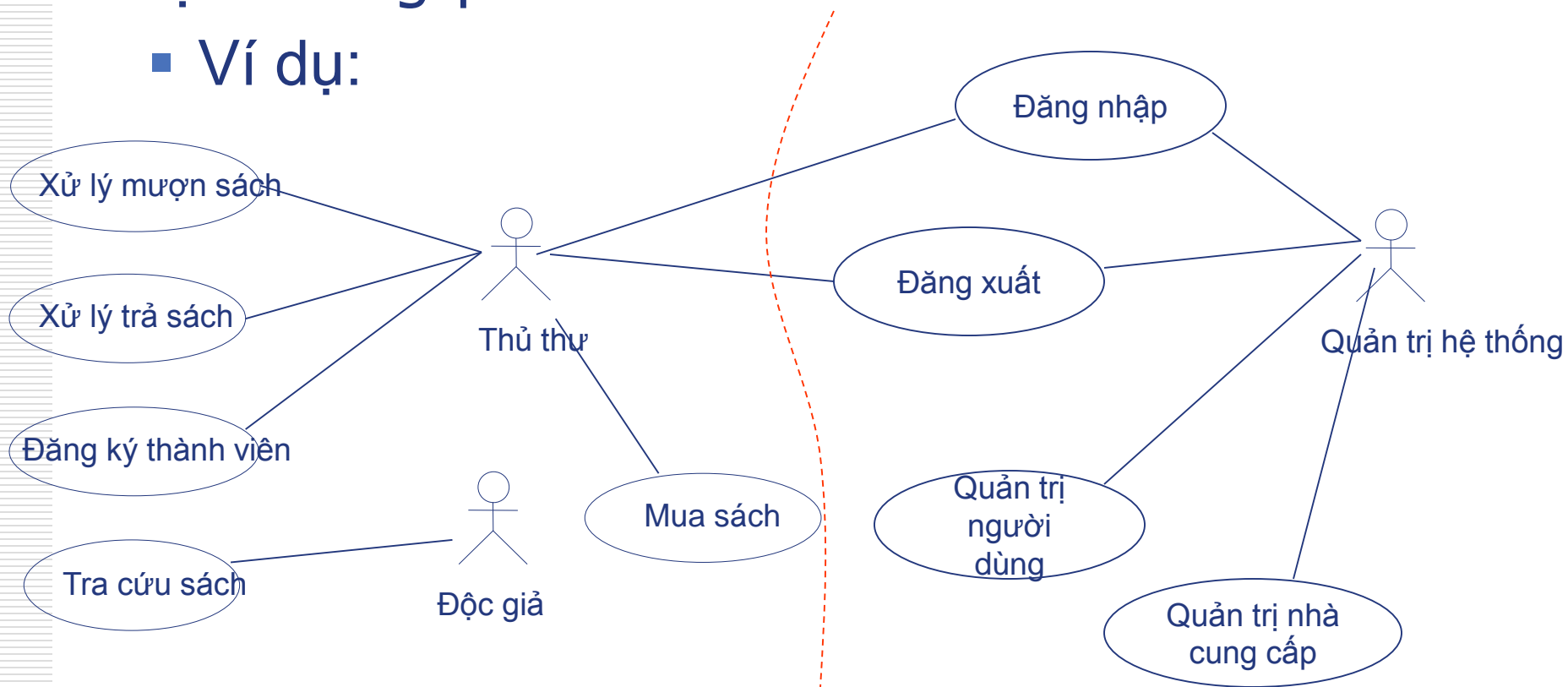
# Tinh chế và nâng cấp sơ đồ use case

- ❖ Bổ sung các use case mô tả chức năng đặc trưng phần mềm:
  - Các use case quản trị hệ thống:
    - Quản trị người dùng
    - Quản trị thông số hệ thống
  - Các use case quản trị dữ liệu
    - Quản trị danh mục dữ liệu
    - Quản trị an toàn (backup/restore)
  - ...

# Tinh chế và nâng cấp sơ đồ use case

❖ Bổ sung các use case mô tả chức năng đặc trưng phần mềm:

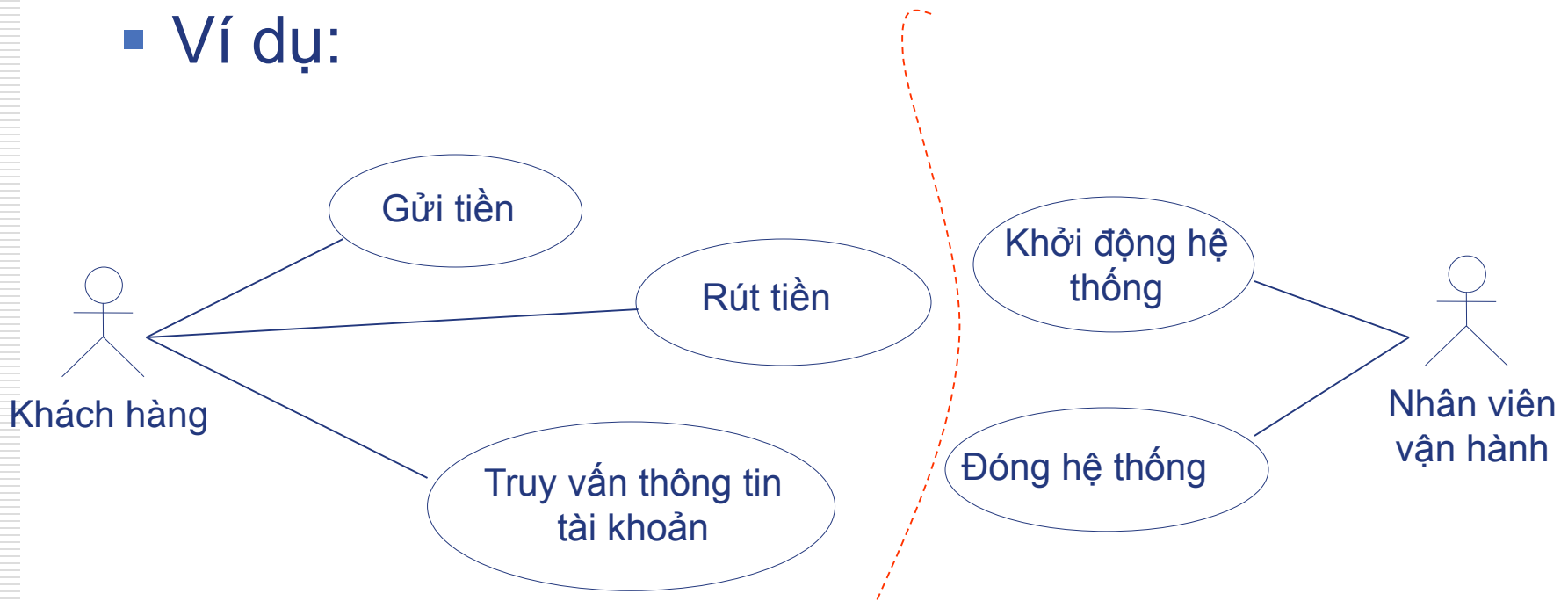
▪ Ví dụ:



# Tinh chế và nâng cấp sơ đồ use case

❖ Bổ sung các use case mô tả chức năng đặc trưng phần mềm:

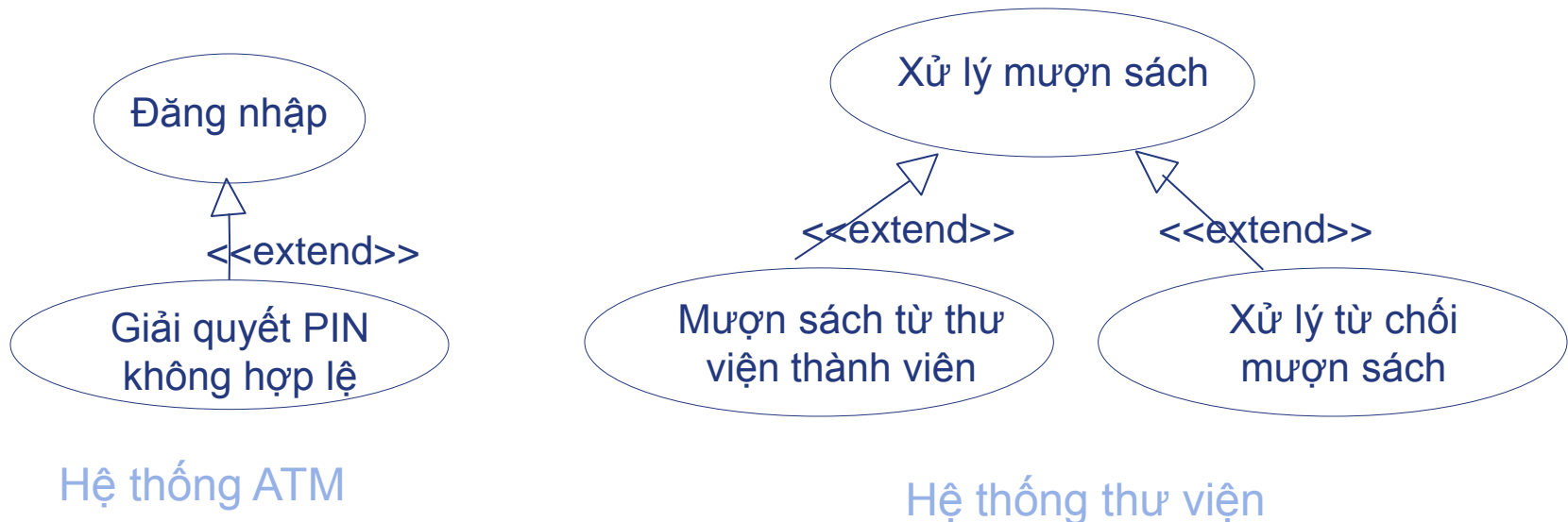
▪ Ví dụ:



# Tinh chế và nâng cấp sơ đồ use case

## ❖ Tinh chế use case

- Phát triển các liên kết <<extend>>:
  - tách những dòng xử lý đặc biệt → use case mở rộng

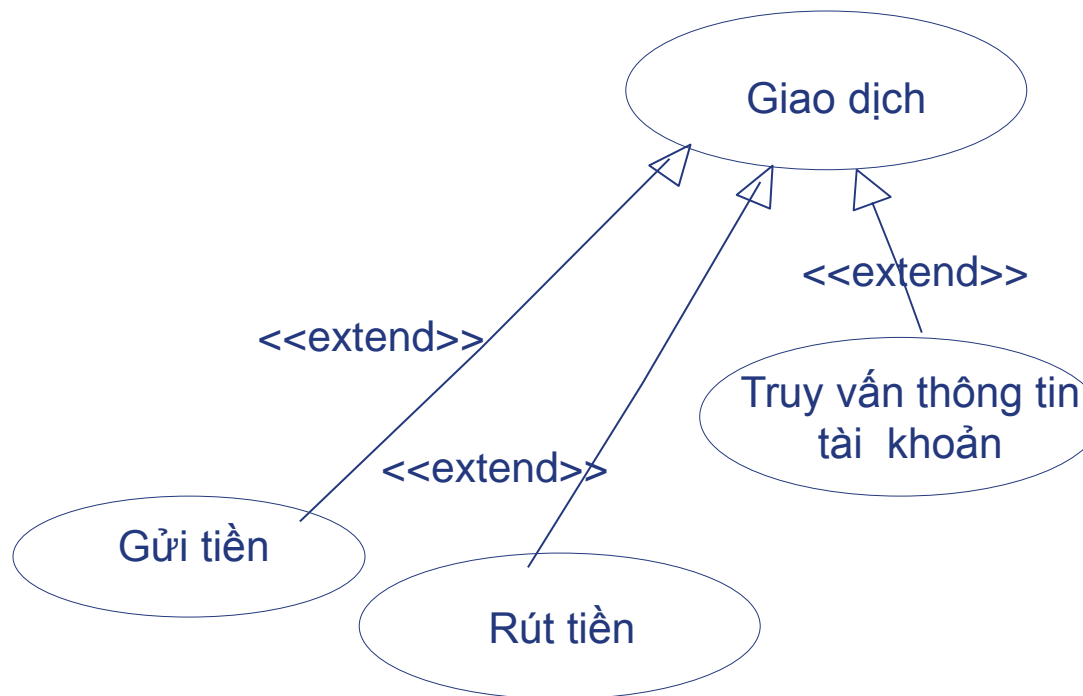




# Tinh chế và nâng cấp sơ đồ use case

## ❖ Tinh chế use case

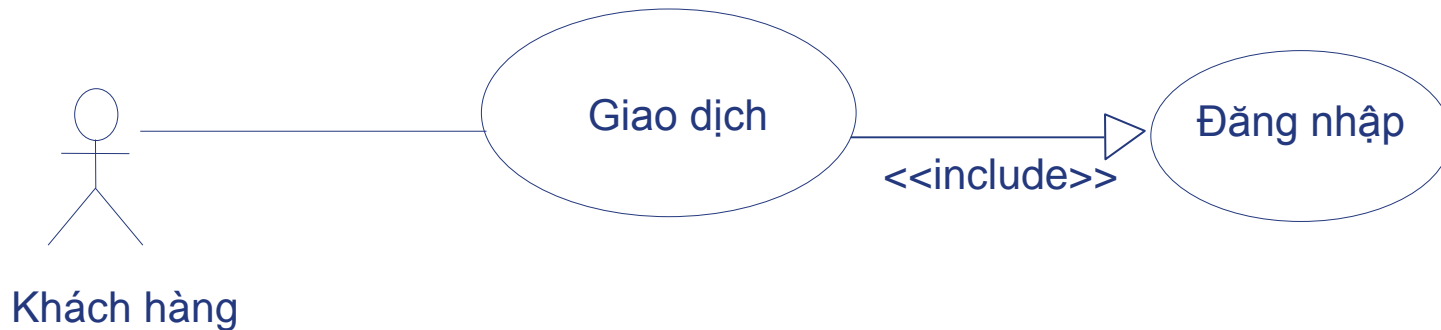
- Phát triển các liên kết <<extend>>:
  - Tổng quát hoá các use case có dòng xử lý chung  
→ use case tổng quát



# Tinh chế và nâng cấp sơ đồ use case

## ❖ Tinh chế use case

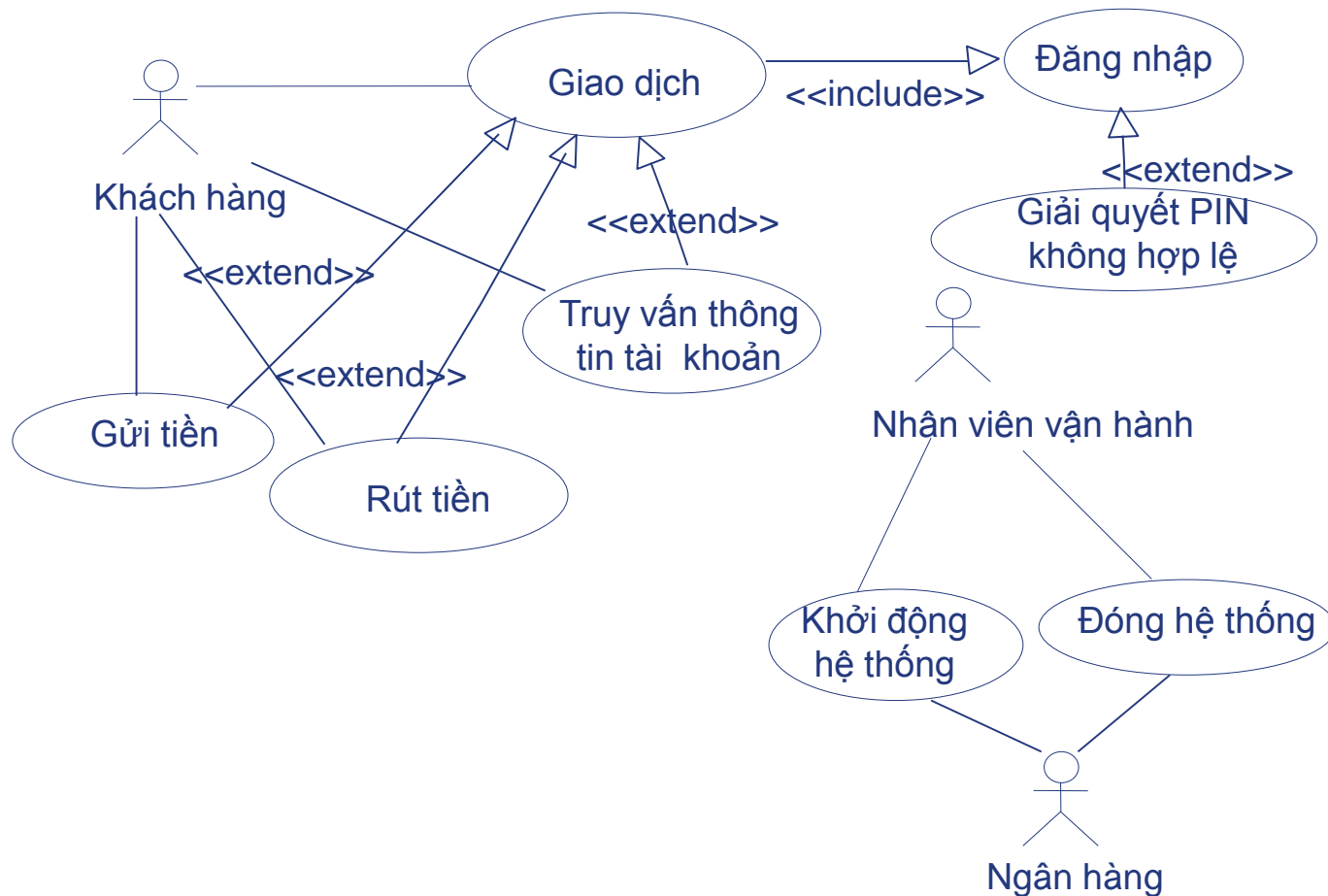
- Phát triển các liên kết <<include>>:



Hệ thống ATM

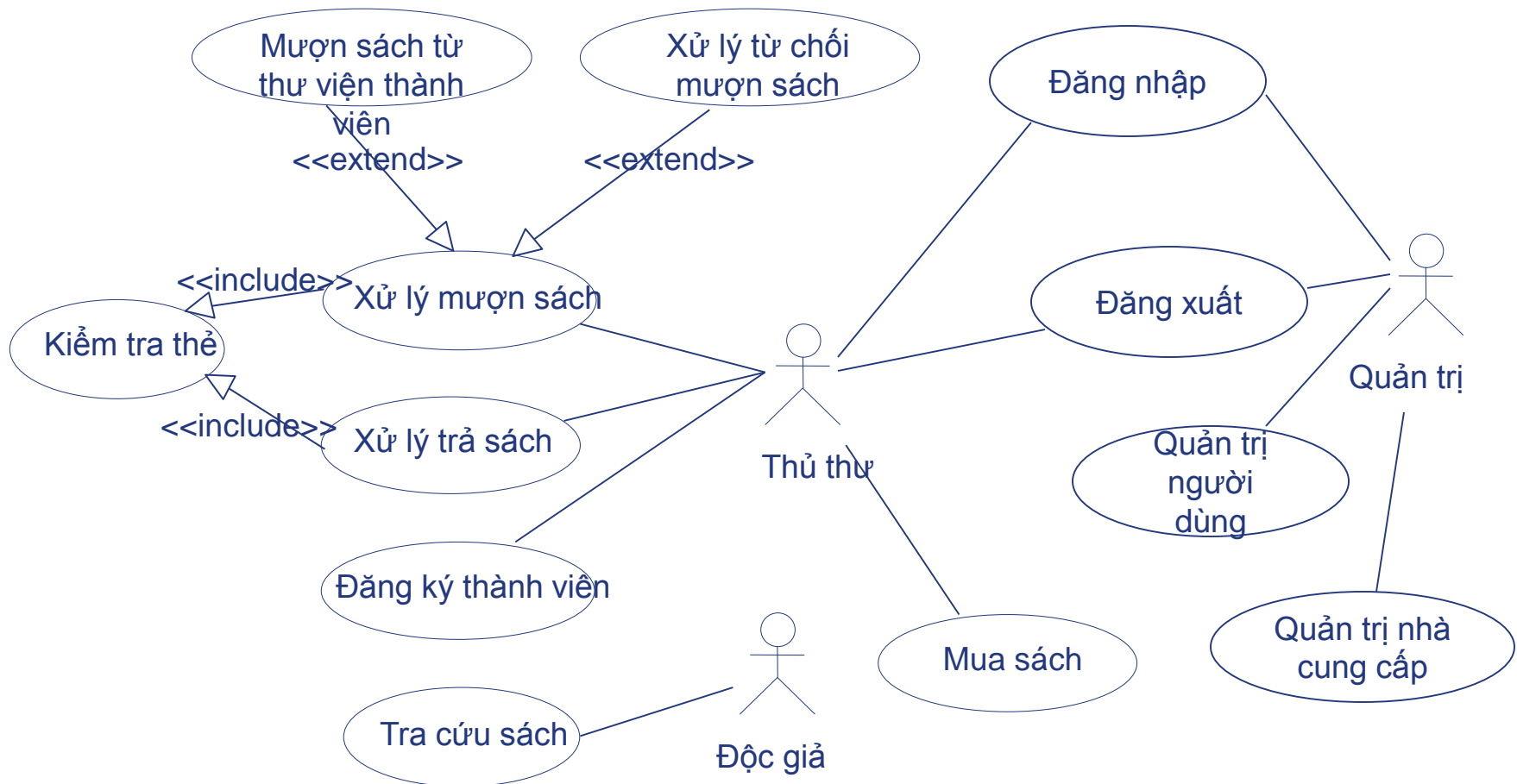
# Tình chế và nâng cấp sơ đồ use case

## ❖ Sơ đồ use case của hệ thống ATM



# Tính chế và nâng cấp sơ đồ use case

## ❖ Sơ đồ use case của hệ thống thư viện



# Mô tả use case

**Tóm tắt:** tóm tắt nội dung hoạt động của use case

**Dòng sự kiện chính:** các sự kiện và hoạt động xảy ra bình thường của use case

**Dòng sự kiện phụ:** các sự kiện và những hoạt động bất thường của use case ngoài những hoạt động chính

**Tiền điều kiện (pre-condition):** mô tả trạng thái của hệ thống phải đạt được để use case có thể bắt đầu

**Hậu điều kiện (post-condition):** liệt kê các trạng thái có thể của hệ thống tại cuối use case. Hệ thống phải thuộc một trong những trạng thái đó khi use case kết thúc

# Mô tả use case

## ❖ Ví dụ: mô tả use case **Giao dịch**

**Tóm tắt:** use case bắt đầu khi có khách hàng đưa thẻ vào máy ATM. Hệ thống sẽ cho phép khách hàng thực hiện các giao dịch

### **Dòng sự kiện chính:**

1. Xác nhận thẻ và kiểm tra đăng nhập của khách hàng (thực hiện bởi use case đăng nhập)
2. Hiển thị giao diện thực đơn yêu cầu khách hàng chọn dịch vụ thực hiện
3. Khách hàng chọn dịch vụ thực hiện
4. Thi hành dịch vụ tương ứng khi khách hàng chọn (rút tiền, gửi tiền, xem thông tin tài khoản: được mô tả trong từng use case tương ứng)
5. Đóng giao diện thực đơn và thông báo khách hàng rút thẻ
6. Khách hàng rút thẻ và hệ thống trở về trạng thái sẵn sàng

# Mô tả use case

## ❖ Ví dụ: mô tả use case **Giao dịch**

### **Dòng sự kiện phụ:**

1. Xử lý thẻ không hợp lệ và đăng nhập không thành công

### **Tiền điều kiện:**

1. ATM phải thông kết nối với mạng ngân hàng
2. ATM phải có đủ giấy để in hoá đơn

### **Hậu điều kiện:**

1. ATM phải ở trạng thái sẵn sàng cho một khách hàng mới

# Mô tả use case

## ❖ Ví dụ: mô tả use case **Rút tiền**

**Tóm tắt:** use case bắt đầu khi khách hàng chọn thực hiện rút tiền từ thực đơn chính. Hệ thống sẽ xử lý yêu cầu rút tiền của khách hàng

### **Dòng sự kiện chính:**

1. Yêu cầu xác định tài khoản
2. Yêu cầu nhập số tiền cần rút
3. Khách hàng nhập số tiền
4. Kiểm tra số tiền nhập với số dư tài khoản
5. Chuyển tiền ra cho khách hàng
6. In hoá đơn rút tiền cho khách hàng



# Mô tả use case

## ❖ Ví dụ: mô tả use case **Rút tiền**

### Dòng sự kiện phụ:

1. Nếu số tiền vượt quá số dư thì thông báo cho khách hàng điều chỉnh hoặc huỷ bỏ giao dịch

### Tiền điều kiện:

1. ATM phải có đủ tiền mặt trong máy
2. ATM phải có đủ giấy để in hoá đơn

### Hậu điều kiện:

1. ATM phải ở trạng thái cho thực hiện một giao dịch mới

# Sequence diagram

- ❖ Mô hình hóa khía cạnh động của hệ thống
- ❖ Nhấn mạnh sự tương tác giữa các đối tượng theo thời gian.
- ❖ Biểu đồ chỉ ra tập các đối tượng và các thông điệp trao đổi giữa chúng, diễn tả các hoạt động như thế nào theo trình tự thời gian
- ❖ Phân biệt:
  - Sequence diagram tập trung vào luồng điều khiển
  - Collaboration diagram tập trung vào luồng dữ liệu

# Sequence diagram

- ❖ Biểu đồ trình tự có thể sử dụng cho nhiều cách:
  - Lập mô hình tương tác ở mức cao giữa các đối tượng hoạt động trong một hệ thống
  - Dùng để mô hình hóa giao tác giữa các hệ thống con
  - Lập mô hình tương tác giữa các đối tượng nhằm thực hiện một use case
  - ...

# Sequence diagram

- ❖ Biểu đồ trình tự gồm các phần tử mô hình như sau:
  - Đối tượng (object)
  - Đường sống, sinh tồn (life line)
  - Thông điệp (message)
  - Vùng hoạt động (activation)
  - Chú thích (note) và ràng buộc (constraint)
- ❖ Ngoài ra biểu đồ trình tự chứa các tác nhân kích hoạt biểu đồ. Tuy nhiên tác nhân không phải là phần tử của biểu đồ

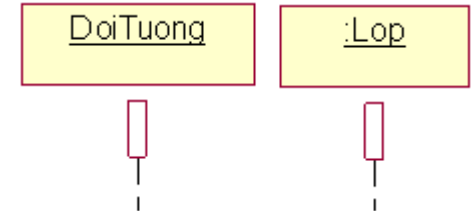
# Sequence diagram

- ❖ Biểu đồ chứa một loạt các đối tượng sắp xếp theo hàng ngang. Còn thời gian biểu diễn theo trục thẳng đứng
- ❖ Chu kỳ sống của đối tượng được biểu diễn bằng các đường thẳng đứng gọi là đường sinh tồn
- ❖ Các thông điệp là các đường ngang nối liền các đường sinh tồn giữa các đối tượng
- ❖ Có nhiều loại thông điệp có ký hiệu và ngữ nghĩa khác nhau.

# Sequence diagram

## ❖ Đối tượng:

- Biểu diễn bằng hình chữ nhật, bên trong là tên đối tượng



## ❖ Đường sinh tồn:

- Biểu diễn bằng đường gạch thẳng đứt nét cho biết thời gian sống của đối tượng

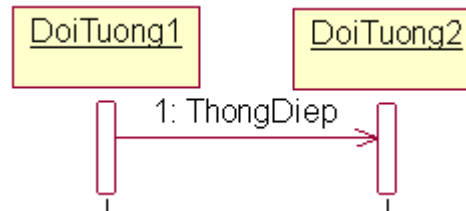
## ❖ Vùng hoạt động:

- Biểu diễn bằng hình chữ nhật hẹp dọc theo đường sinh tồn
- Cho biết thời gian thực thi một hoạt động tương ứng.

# Sequence diagram

## ❖ Thông điệp đơn giản

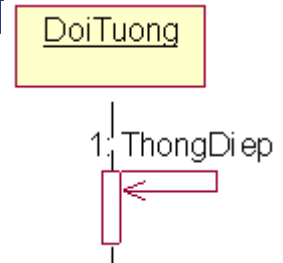
- Thông điệp được gửi đi giữa các đối tượng, đây là thao tác lớp được định nghĩa sau này.
- Thông điệp biểu diễn một đối tượng gọi thao tác của đối tượng khác



# Sequence diagram

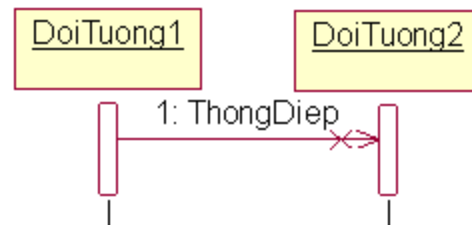
## ❖ Thông điệp phản thân:

- Đối tượng gọi chính thao tác của mình



## ❖ Thông điệp đồng bộ:

- Đối tượng gửi thông điệp và chờ đến khi thông điệp được giải quyết xong

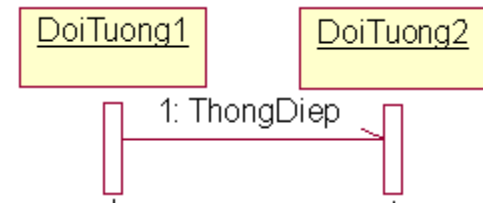




# Sequence diagram

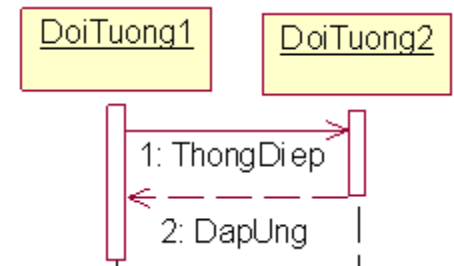
## ❖ Thông điệp bất đồng bộ:

- Đối tượng gửi thông điệp đến đối tượng nhận. Sau đó đối tượng gửi tiếp tục làm công việc khác. Không quan tâm đến thông điệp đó có được xử lý hay không.



## ❖ Thông điệp đáp ứng:

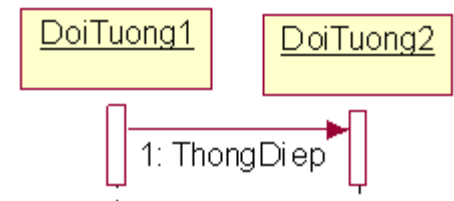
- Sau khi nhận thông điệp từ đối tượng gửi nào đó. Một đối tượng có thể gửi thông điệp đáp ứng cho đối tượng gửi



# Sequence diagram

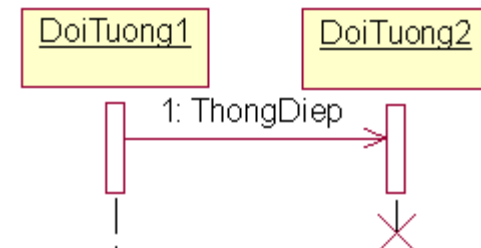
## ❖ Thông điệp thủ tục:

- Đối tượng gửi thông điệp và chờ cho đến khi nó xử lý xong
- Bao gồm việc chờ cho đến khi nào hoàn tất hoạt động được gọi bởi đối tượng nhận trên những đối tượng khác.



## ❖ Thông điệp hủy đối tượng:

- Hủy bỏ đối tượng



# Sequence diagram

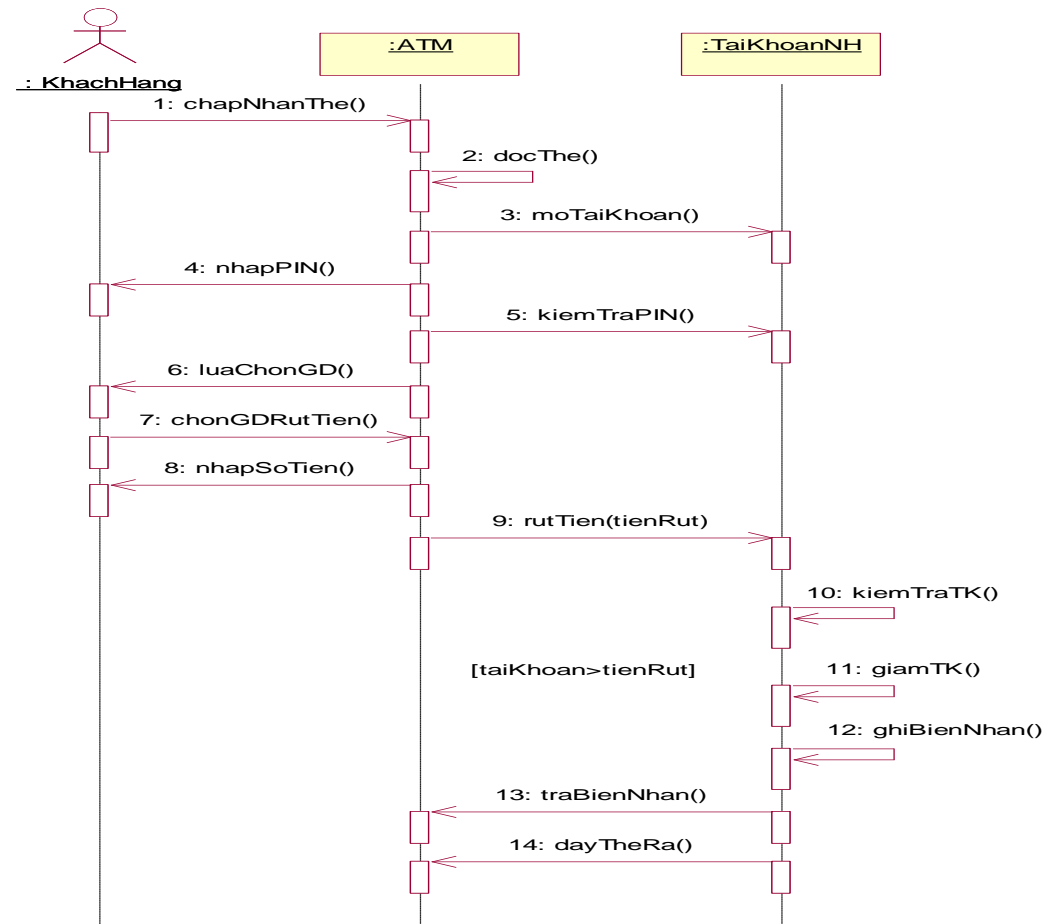
- ❖ Các bước thực hiện biểu đồ tuần tự
  - Xác định ngữ cảnh tương tác: hệ thống, hệ thống con, use case hoặc thao tác.
  - Xem xét các kịch bản thay thế
  - Vẽ các biểu đồ thể hiện
    - Đặt các đối tượng từ trái sang phải
    - Bắt đầu bằng thông điệp khởi đầu tương tác, đặt các thông điệp theo chiều từ trên xuống dưới theo trục thời gian. Biểu diễn các đặc tính của các thông điệp cần thiết để giải thích ngữ nghĩa tương tác.
    - Thêm các vùng hoạt động nếu cần để trực quan hóa hành động
    - Thêm các ràng buộc khác nếu cần (thời gian)
    - Thêm các chú thích vào biểu đồ.

# Sequence diagram

- ❖ Khi xây dựng biểu đồ tương tác, ban đầu chỉ tập trung vào các thông tin liên quan đến khách hàng.
  - Chưa ánh xạ thông điệp thành thao tác
  - Chỉ đặc tả thông điệp không hình thức
  - Chưa ánh xạ đối tượng thành lớp
- ❖ Biểu đồ tạo ra trong giai đoạn này chỉ quan tâm đến luồng tác nghiệp
- ❖ Sau đó biểu đồ sẽ được bổ sung chi tiết hơn như ánh xạ đối tượng thành lớp, thông điệp thành thao tác, bổ sung một số đối tượng khác như giao diện, điều khiển.

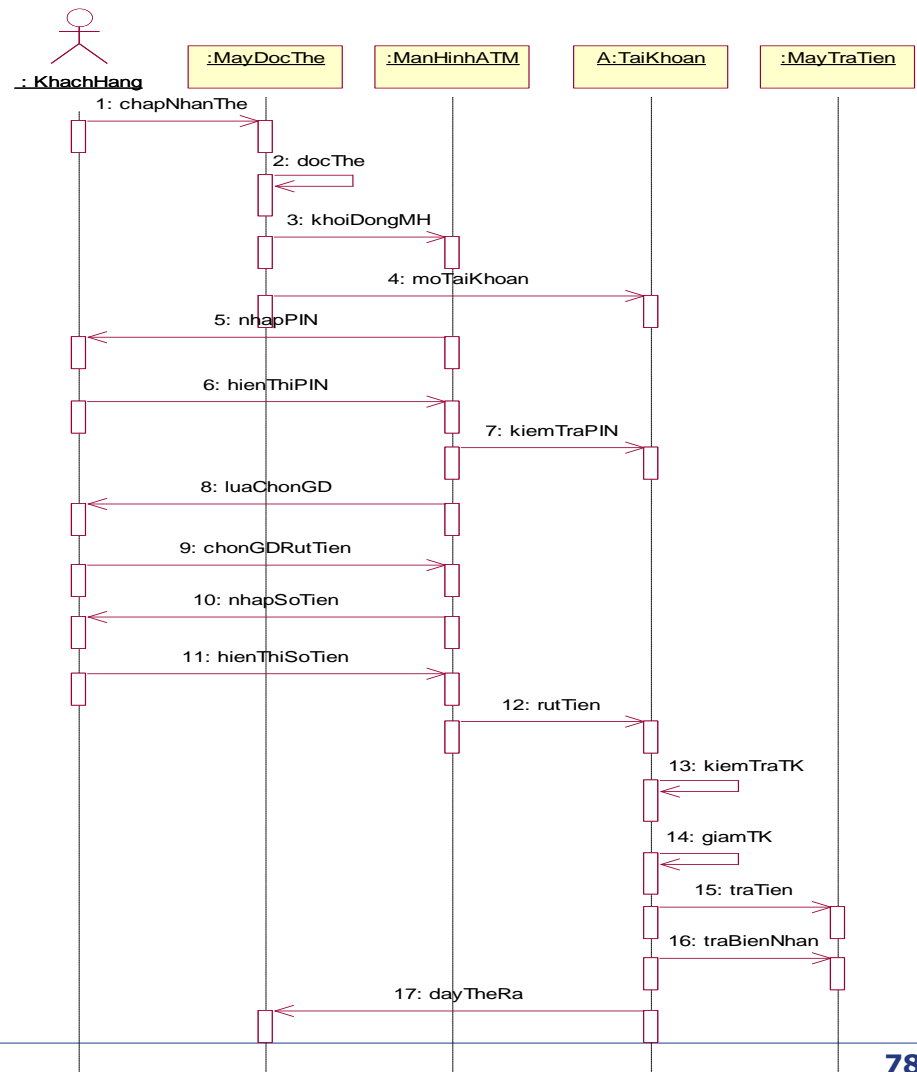
# Sequence diagram

❖ Biểu đồ trình tự của hệ thống ATM ban đầu



# Sequence diagram

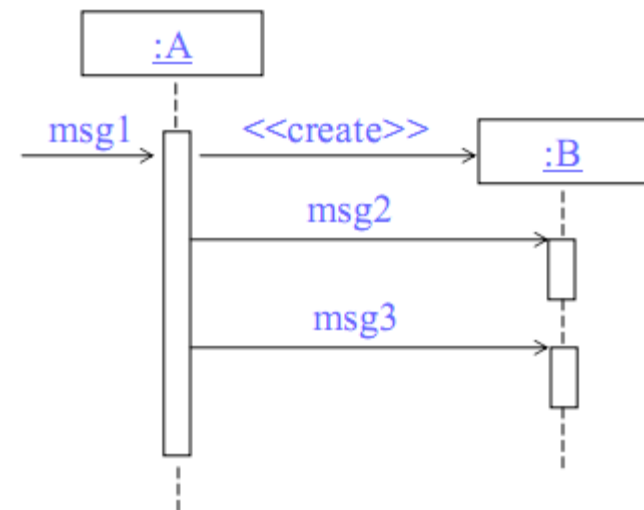
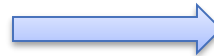
❖ Biểu đồ trình tự ở mức chi tiết hơn



# Sequence diagram

❖ Thể hiện một thao tác của một lớp

```
public class A
{
    private B objB;
    public void msg1()
    {
        objB = new B();
        objB.msg2();
        objB.msg3();
    }
}
```



```
public class B
{
    ...
    public void msg2() { ... }
    public void msg3() { ... }
}
```

# Sequence diagram

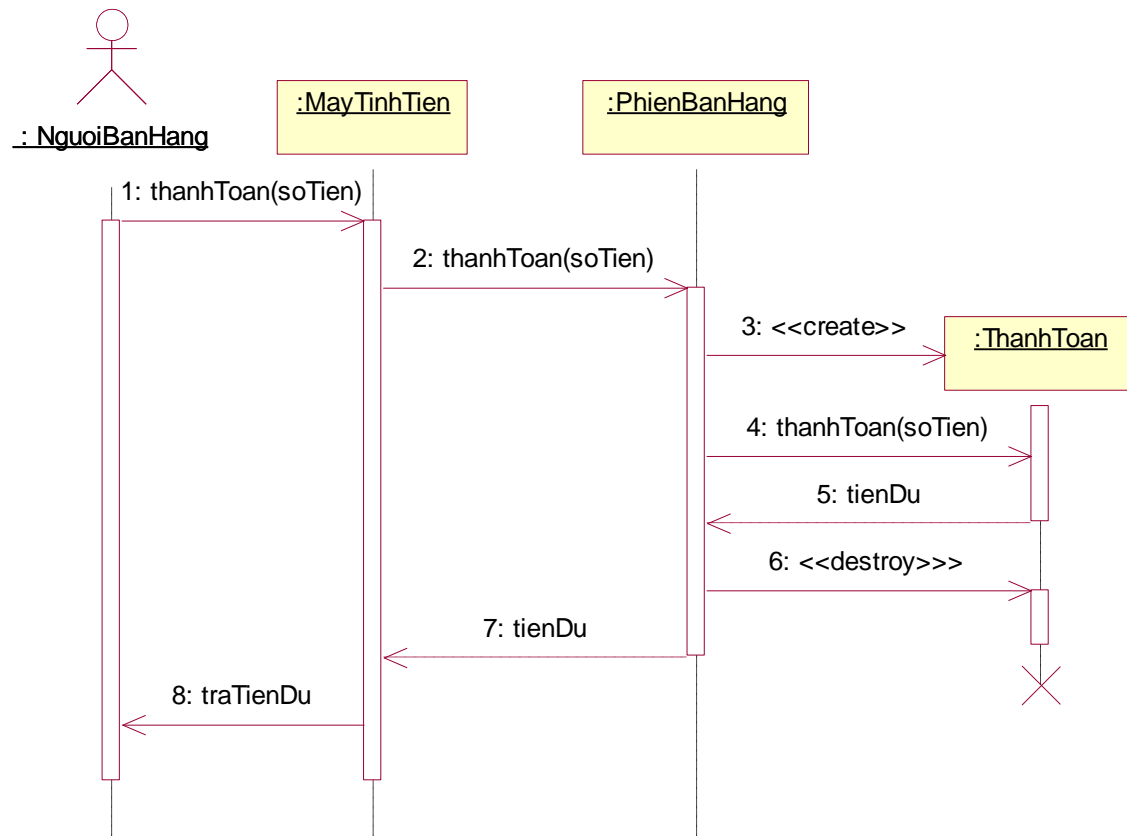
## ❖ Minh họa chức năng tính tiền:

- Các đối tượng:
  - Người bán hàng
  - Máy tính tiền
  - Phiên bán hàng
  - Thanh toán
- Các thao tác (kịch bản)
  - Người bán hàng gửi thông điệp thanh toán với số tiền tương ứng
  - Máy tính tiền yêu cầu phiên bán hàng thanh toán với các thông tin trên
  - Phiên bán hàng thực thi các công việc sau:
    - Tạo đối tượng thanh toán
    - Yêu cầu thanh toán với thông tin vừa nhận
    - Đối tượng thanh toán trả về số tiền dư
    - Thông điệp hủy đối tượng thanh toán
  - Phiên bán hàng trả thông tin số dư về cho máy tính tiền
  - Máy tính tiền thực hiện thông điệp trả tiền dư về cho người bán hàng.



# Sequence diagram

## ❖ Chức năng tính tiền



# Collaboration diagram

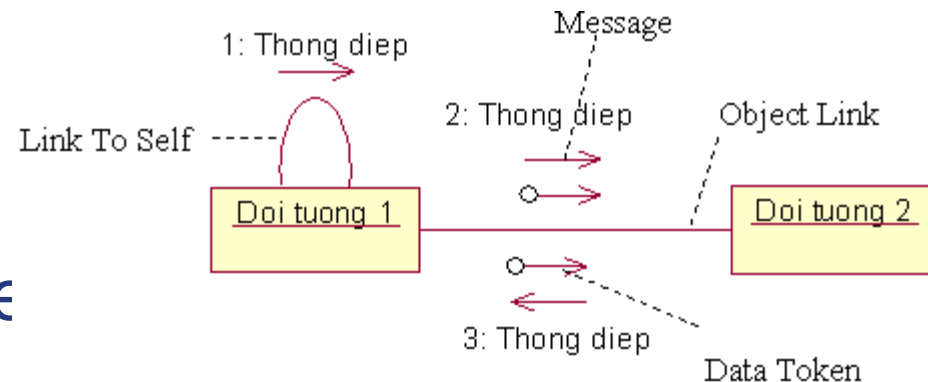
- ❖ Trong hệ thống HĐT, các đối tượng phối hợp hoạt động với nhau tạo ra chức năng mà người sử dụng yêu cầu.
- ❖ Để phối hợp hoạt động, các đối tượng cần giao tiếp với nhau bằng cách gửi thông điệp.
- ❖ Hoạt động như vậy gọi là cộng tác.
- ❖ Biểu đồ cộng tác mô tả sự tương tác giữa các đối tượng bằng việc nhấn mạnh cấu trúc kết hợp giữa các đối tượng và những thông điệp trao đổi của chúng.

# Collaboration diagram

- ❖ Biểu đồ cộng tác hiển thị luồng dữ liệu khác với biểu đồ tuần tự hiển thị luồng điều khiển.
- ❖ Biểu đồ cộng tác là sự mở rộng của biểu đồ đối tượng
- ❖ Biểu đồ cộng tác chỉ ra:
  - Thứ tự gọi các thông điệp: mỗi thông điệp được gán một số tuần tự
  - Điều kiện gọi các thông điệp

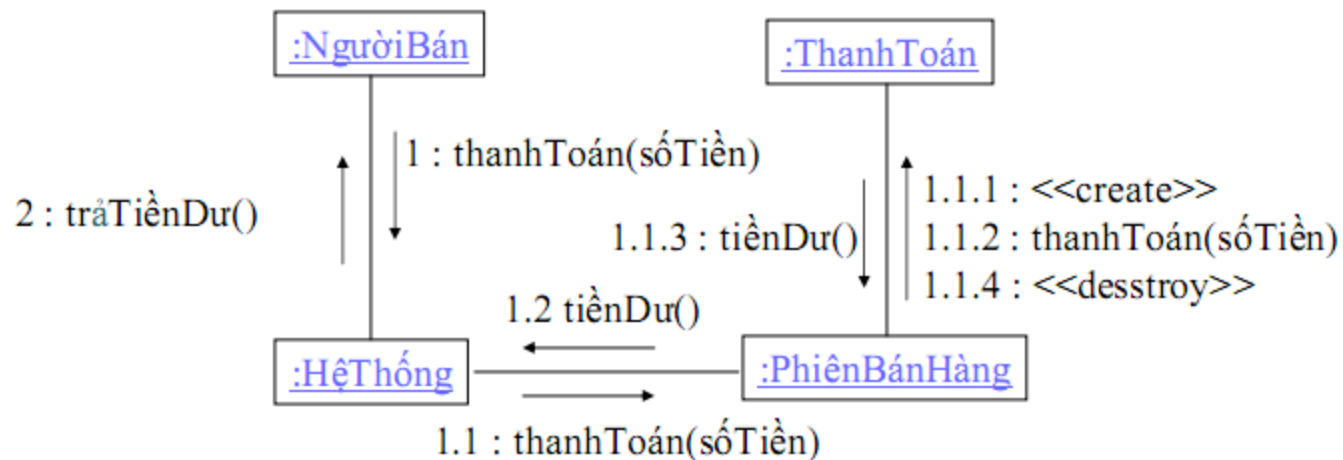
# Collaboration diagram

- ❖ Biểu đồ cộng tác bao gồm các đối tượng cùng các liên kết biểu diễn đường thẳng nối các đối tượng để chỉ ra dòng chảy thông điệp giữa các đối tượng.
- ❖ Các phần tử mô hình biểu diễn trong biểu đồ cộng tác:
  - Đối tượng (object)
  - Liên kết (link)
  - Thông điệp (message)
  - Luồng dữ liệu (data token)

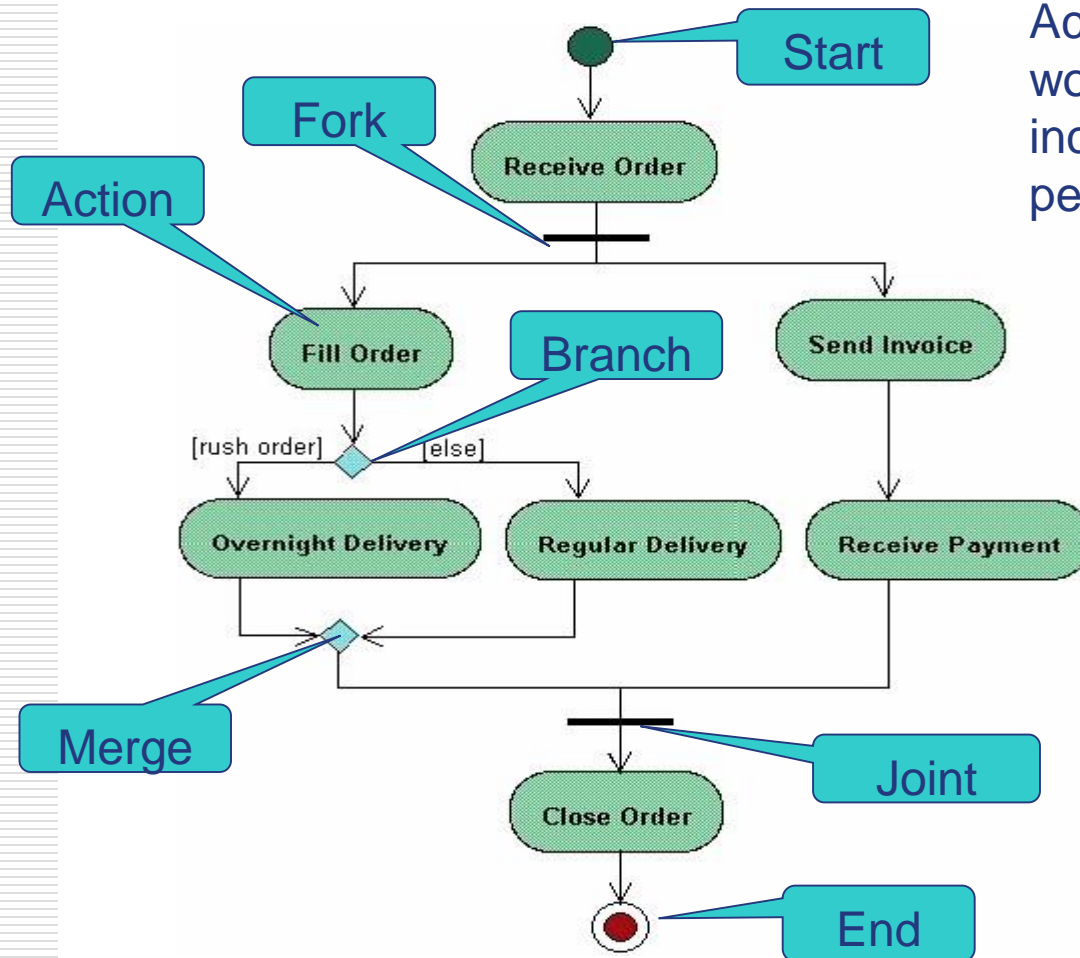


# Collaboration diagram

## ❖ Ví dụ biểu đồ cộng tác



# Modeling Tools - Activities Diagram



Activity diagrams describe the workflow behaviour of a system including a sequence of activities performed from start to finish

Activities could be performed:

- sequential order
- parallel
- conditional transition

*Rhumba!*

