



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

# Nhập môn Công nghệ phần mềm Introduction to Software Engineering (IT3180)

## **Bài tập tuần 07**

Kỹ nghệ yêu cầu phần mềm (Requirement Engineering) (tiếp theo)

Giang vien

# Mục tiêu

- Thực hiện các bài tập (câu hỏi) về phân tích yêu cầu phần mềm
- Thực hiện các bài tập về công cụ đặc tả yêu cầu phần mềm
- Phân tích các yêu cầu cho bài toán (casestudy) của môn học: sử dụng một số biểu đồ của UML
  - Phân rã các usecase thành các lớp phân tích nhận trách nhiệm thực thi các hoạt động trong kịch bản usecase
  - Làm quen với sơ đồ trình tự (sequence diagram)
  - Xây dựng sơ đồ thực thể - quan hệ (ERD): mô hình hoá các dữ liệu cho bài toán

# Đánh giá

- Hoàn thành các bài tập về phân tích và đặc tả yêu cầu phần mềm, nắm được các khái niệm và những hoạt động cần thực hiện trong giai đoạn phân tích
- Vận dụng các công cụ đặc tả yêu cầu phần mềm: Sơ đồ thực thể liên kết – ERD (entity relation diagram) + và một số biểu đồ trong UML như biểu đồ trình tự
- Hoàn thành phân tích các yêu cầu cho bài toán (casestudy) của môn học

# Bài 1.1

**a) Tác nhân ca sử dụng luôn là con người, không phải là các thiết bị hệ thống?**

**1. Sai**

**2. Đúng**

# Bài 1.1

## **b) Use-cases là một kịch bản mà mô tả?**

- 1. Phần mềm thực hiện như thế nào khi được dùng trong một tình huống cho trước**
- 2. Những công cụ CASE sẽ được dùng như thế nào để xây dựng hệ thống**
- 3. Kế hoạch xây dựng cho sản phẩm phần mềm**
- 4. Những test-case cho sản phẩm phần mềm**

# Bài 1.1

c) Phát biểu nào sau đây là đúng về yêu cầu phần mềm? Trả lời bằng cách đánh dấu T (đúng) hoặc F (sai).

- (1) T / F Độ tin cậy và bảo mật là ví dụ về các yêu cầu chức năng.
- (2) T / F Yêu cầu đóng vai trò như một cơ sở cho việc kiểm tra và xác minh.
- (3) T / F Yêu cầu mô tả kiến trúc phần mềm.
- (4) T / F Các yêu cầu về hành vi thường mang tính chủ quan và không thể đo lường được.
- (5) T / F Các trường hợp sử dụng nắm bắt các yêu cầu chức năng.
- (6) T / F Lý do số một mà các dự án thành công là sự tham gia của nhà phát triển.
- (7) T / F Một ca sử dụng đại diện cho một hành vi ví dụ của hệ thống.
- (8) T / F Các tình huống mở rộng của một ca sử dụng thiết lập sự hiểu biết giữa khách hàng và nhà phát triển hệ thống về các yêu cầu.
- (9) T / F Trong hầu hết các trường hợp sử dụng, gần như mọi bước đều có thể bị lỗi.

# Bài 1.1

c) Phát biểu nào sau đây là đúng về yêu cầu phần mềm? Trả lời bằng cách đánh dấu T (đúng) hoặc F (sai). Gợi ý:

- (1) T / F Độ tin cậy và bảo mật là ví dụ về các yêu cầu chức năng.
- (2) T / F Yêu cầu đóng vai trò như một cơ sở cho việc kiểm tra và xác minh.
- (3) T / F Yêu cầu mô tả kiến trúc phần mềm.
- (4) T / F Các yêu cầu về hành vi thường mang tính chủ quan và không thể đo lường được.
- (5) T / F Các trường hợp sử dụng nắm bắt các yêu cầu chức năng.
- (6) T / F Lý do số một mà các dự án thành công là sự tham gia của nhà phát triển.
- (7) T / F Một ca sử dụng đại diện cho một hành vi ví dụ của hệ thống.
- (8) T / F Các tình huống mở rộng của một ca sử dụng thiết lập sự hiểu biết giữa khách hàng và nhà phát triển hệ thống về các yêu cầu.
- (9) T / F Trong hầu hết các trường hợp sử dụng, gần như mọi bước đều có thể bị lỗi.

# Bài 1.2

- a) Hãy trình bày nội dung của hoạt động thẩm định yêu cầu?**
- b) Hãy nêu một số nhược điểm/hạn chế của kỹ thuật mô hình hóa ca sử dụng?**



# Bài 1.2

a) Hãy trình bày nội dung của hoạt động thẩm định yêu cầu?

## Gợi ý:

- Mỗi yêu cầu có phù hợp với dự án tổng thể hoặc mục tiêu của hệ thống không?
- Tất cả các yêu cầu được chỉ định ở mức trừu tượng có phù hợp không?
- Mỗi yêu cầu có cần thiết cho mục tiêu hệ thống hay là một tính năng bổ sung?
- Mỗi yêu cầu có bị ràng buộc và rõ ràng không?
- Nguồn cho từng yêu cầu là gì?
- Các yêu cầu có xung đột với nhau không?
- Yêu cầu có thể đạt được trong môi trường kỹ thuật được đề xuất cho hệ thống hoặc sản phẩm không?
- Mỗi yêu cầu có thể kiểm thử được không?
- Mô hình yêu cầu có phản ánh thông tin, chức năng và hành vi của hệ thống được xây dựng không?
- Mô hình yêu cầu có được tổ chức theo cách cho phép biểu diễn thông tin hệ thống theo hướng chi tiết dần một cách liên tục không?
- Tất cả các mẫu yêu cầu đã được xác thực hợp lệ hay chưa và chúng có phù hợp với yêu cầu của khách hàng không?

# Bài 1.2

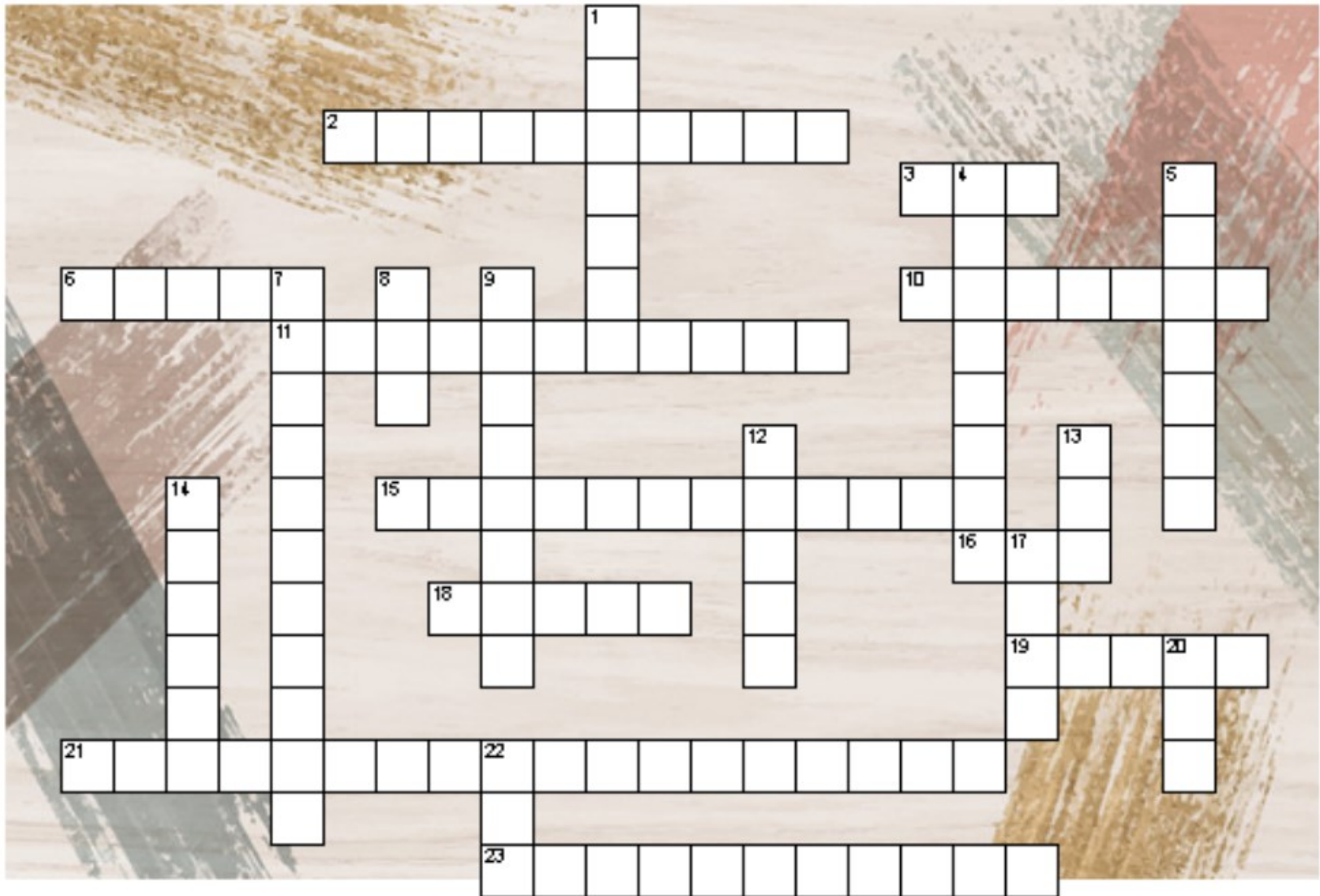
b) Hãy nêu một số nhược điểm/hạn chế của kỹ thuật mô hình hóa ca sử dụng?

## Gợi ý:

- Thiếu đặc tả hình thức trong mô tả ca sử dụng.
- Không phải tất cả các hệ thống đều có các tác nhân được xác định rõ ràng.
- Các ca sử dụng vốn không theo hướng đối tượng.
- Các nhà phát triển có xu hướng tiến hành phân rã chức năng với các ca sử dụng.
- ...

# Bài 1.3

Hãy giải ô chữ tổng hợp kiến thức dưới đây với các gợi ý kèm theo?



# Bài 1.3

## Gợi ý của ô chữ:

### Across

- 2. Kiểm tra để đảm bảo rằng phần mềm tuân thủ các yêu cầu của nó (10)
- 3. Sự phân tách thứ bậc của các hoạt động công việc trong một dự án phần mềm (3)
- 6. Một khái niệm mô tả cho việc đóng gói các thuộc tính và hoạt động trong OOP (5)
- 10. Một chuỗi các giao dịch trong cuộc đối thoại giữa một tác nhân và một thành phần hoặc hệ thống (7)
- 11. Tạo bản mô phỏng nhanh cho ứng dụng (11)
- 15. Hoạt động kiểm tra để đảm bảo tính đúng đắn (12)

### Down

- 1. Sản phẩm đáp ứng các tiêu chí của khách hàng, phù hợp với đặc điểm thiết kế (7)
- 4. Điểm mà một số sản phẩm có thể phân phối được đặt dưới sự kiểm soát thay đổi chính thức (8)
- 5. Một tập các hoạt động nhằm cố gắng phát hiện và loại bỏ lỗi (7)
- 7. Một mô hình phát triển phần mềm tiến hóa với pha xác định và kiểm soát rủi ro (11)
- 8. Lines of code (3)
- 9. Một thỏa thuận ràng buộc lẫn nhau (8)
- 12. Loại biểu đồ được sử dụng phổ biến trong quản lý dự án (5)

# Bài 1.3

## Gợi ý của ô chữ:

### Across

- 16. Loại sơ đồ thể hiện thiết kế cơ sở dữ liệu về thực thể và quan hệ (3)
- 18. Một loại vai trò được thực hiện bởi một thực thể trong tương tác với hệ thống (5)
- 19. Một phương pháp Agile dành cho quản lý tiến trình phần mềm với các vòng lặp ngắn gọi là sprint (5)
- 21. Một phương pháp kỹ thuật phần mềm được sử dụng trong Agile có thể sử dụng lập trình cặp đôi (18)
- 23. Khả năng chuyển phần mềm có thể hoạt động từ môi trường này sang môi trường khác (11)

### Down

- 13. Loại biểu đồ thể hiện luồng dữ liệu trong hệ thống (3)
- 14. Một lỗi hỏng trong một bộ phận hoặc hệ thống gây ra việc không thực hiện được chức năng cần thiết (6)
- 17. Một sự kiện hoặc điều kiện không chắc chắn (4)
- 20. Một ngôn ngữ mô hình hóa mục đích chung được chuẩn hóa (3)
- 22. Khung quy trình phát triển phần mềm hợp nhất có tính lặp có thể thích ứng, thường sử dụng UML kèm theo (3)

# Bài 1.3

## Lời giải:

### Across

- 2. Kiểm tra để đảm bảo rằng phần mềm tuân thủ các yêu cầu của nó (10) [VALIDATION]
- 3. Sự phân tách thứ bậc của các hoạt động công việc trong một dự án phần mềm (3) [WBS]
- 6. Một khái niệm mô tả cho việc đóng gói các thuộc tính và hoạt động trong OOP (5) [CLASS]
- 10. Một chuỗi các giao dịch trong cuộc đối thoại giữa một tác nhân và một thành phần hoặc hệ thống (7) [USECASE]
- 11. Tạo bản mô phỏng nhanh cho ứng dụng (11) [PROTOTYPING]
- 15. Hoạt động kiểm tra để đảm bảo tính đúng đắn (12) [VERIFICATION]

### Down

- 1. Sản phẩm đáp ứng các tiêu chí của khách hàng, phù hợp với đặc điểm thiết kế (7) [QUALITY]
- 4. Điểm mà một số sản phẩm có thể phân phối được đặt dưới sự kiểm soát thay đổi chính thức (8) [BASELINE]
- 5. Một tập các hoạt động nhằm cố gắng phát hiện và loại bỏ lỗi (7) [TESTING]
- 7. Một mô hình phát triển phần mềm tiến hóa với pha xác định và kiểm soát rủi ro (11) [SPIRALMODEL]
- 8. Lines of code (3) [LOC]
- 9. Một thỏa thuận ràng buộc lẫn nhau (8) [CONTRACT]



# Bài 1.3

## Lời giải:

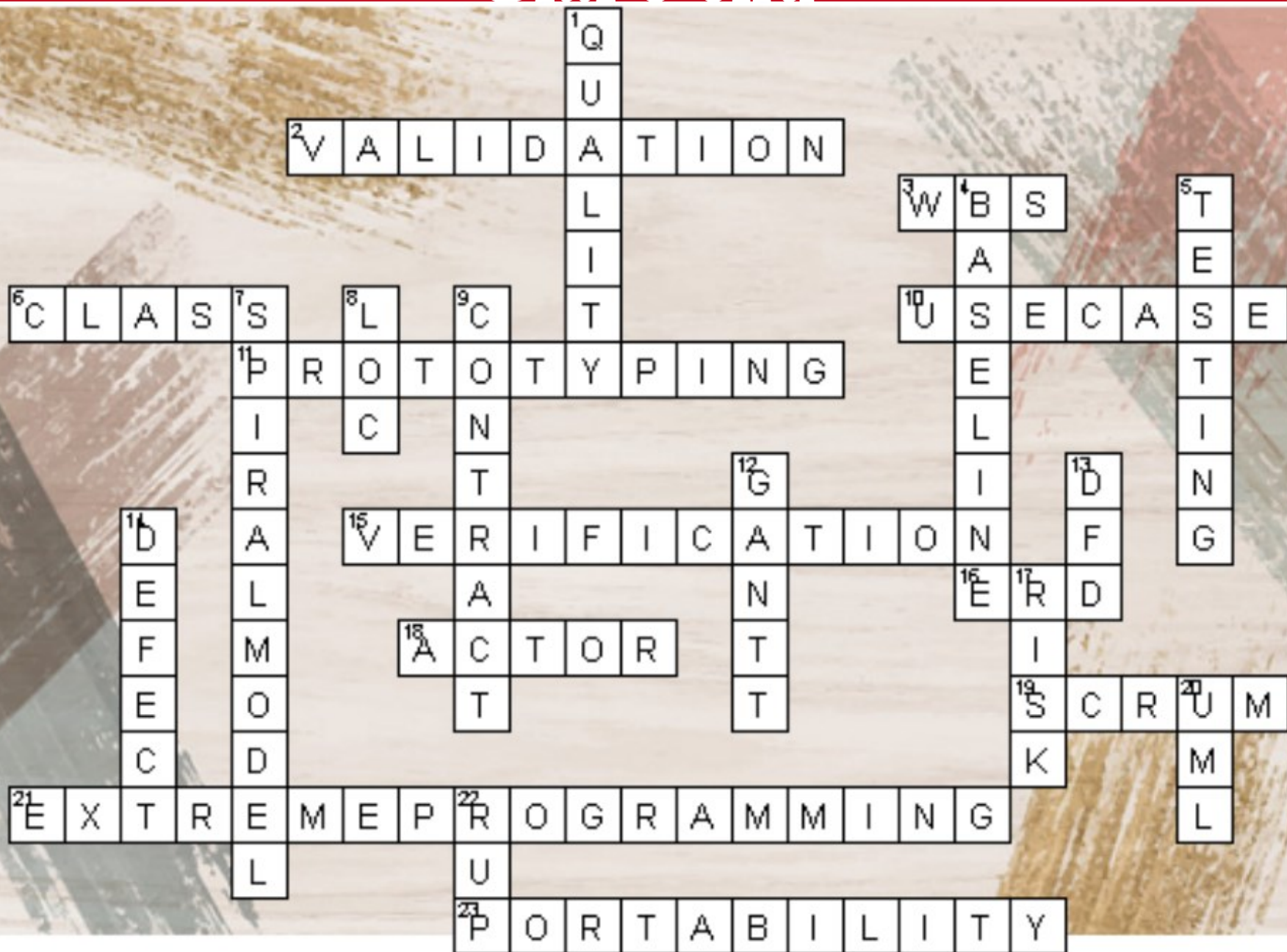
### Across

- 16. Loại sơ đồ thể hiện thiết kế cơ sở dữ liệu về thực thể và quan hệ (3) [ERD]
- 18. Một loại vai trò được thực hiện bởi một thực thể trong tương tác với hệ thống (5) [ACTOR]
- 19. Một phương pháp Agile dành cho quản lý tiến trình phần mềm với các vòng lặp ngắn gọi là sprint (5) [SCRUM]
- 21. Một phương pháp kỹ thuật phần mềm được sử dụng trong Agile có thể sử dụng lập trình cặp đôi (18) [EXTREMEPROGRAMMING]
- 23. Khả năng chuyển phần mềm có thể hoạt động từ môi trường này sang môi trường khác (11) [PORTABILITY]

### Down

- 12. Loại biểu đồ được sử dụng phổ biến trong quản lý dự án (5) [GANTT]
- 13. Loại biểu đồ thể hiện luồng dữ liệu trong hệ thống (3) [DFD]
- 14. Một lỗi hỏng trong một bộ phận hoặc hệ thống gây ra việc không thực hiện được chức năng cần thiết (6) [DEFECT]
- 17. Một sự kiện hoặc điều kiện không chắc chắn (4) [RISK]
- 20. Một ngôn ngữ mô hình hóa mục đích chung được chuẩn hóa (3) [UML]
- 22. Khung quy trình phát triển phần mềm hợp nhất có tính lặp có thể thích ứng, thường sử dụng UML kèm theo (3) [RUP]

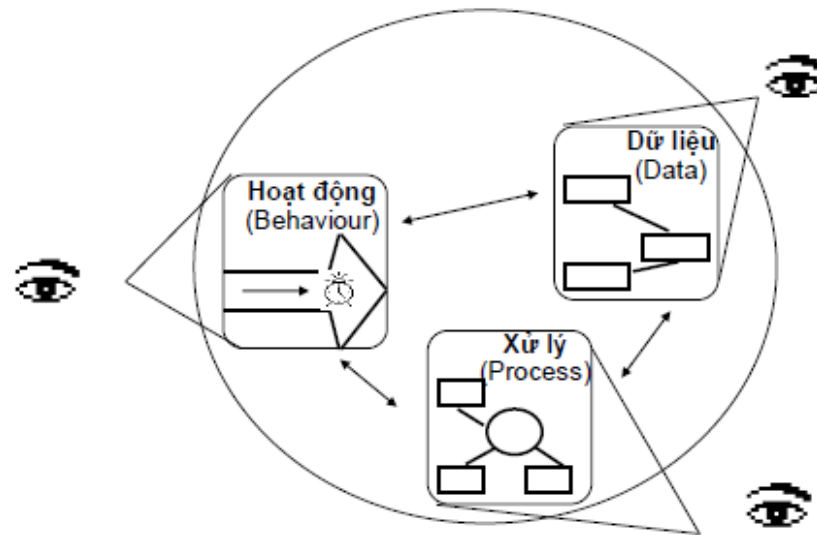
# Giải ô chữ





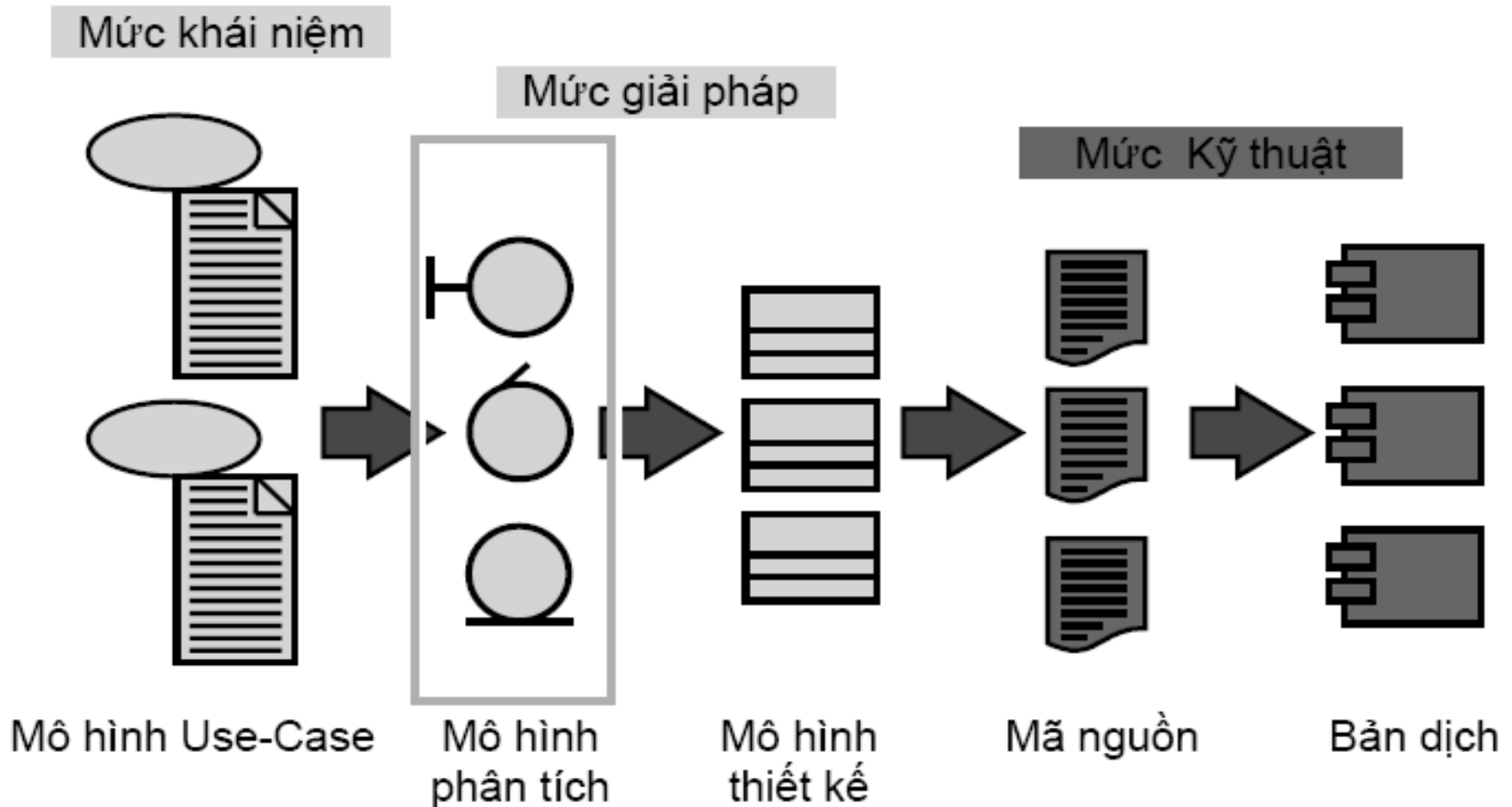
# Phần II: Phân tích usecase

- **Phân tích yêu cầu** là nhằm trả lời câu hỏi hệ thống sẽ làm những gì (what), hơn là chỉ ra cách thức (how) làm những việc đó.
- **Phân rã** các yêu cầu phức tạp được trình bày trong pha xác định yêu cầu thành các nhân tố chính cùng mối quan hệ giữa chúng để làm cơ sở cho giải pháp được trình bày trong pha thiết kế sau này. Kết quả của quá trình phân rã là **các lớp phân tích**. *Tất cả các hoạt động trong kịch bản của Use-Case phải được phản ánh đầy đủ trong các lớp phân tích.*
- Tại sao lại cần phân tích? Mô hình trợ giúp cho người phân tích trong việc hiểu về thông tin, chức năng và hành vi của hệ thống.



# Phần II: Phân tích usecase

- Mô hình phân tích** có vai trò như cầu nối giữa các mô tả hệ thống và mô hình thiết kế:



# Phần II: Phân tích usecase

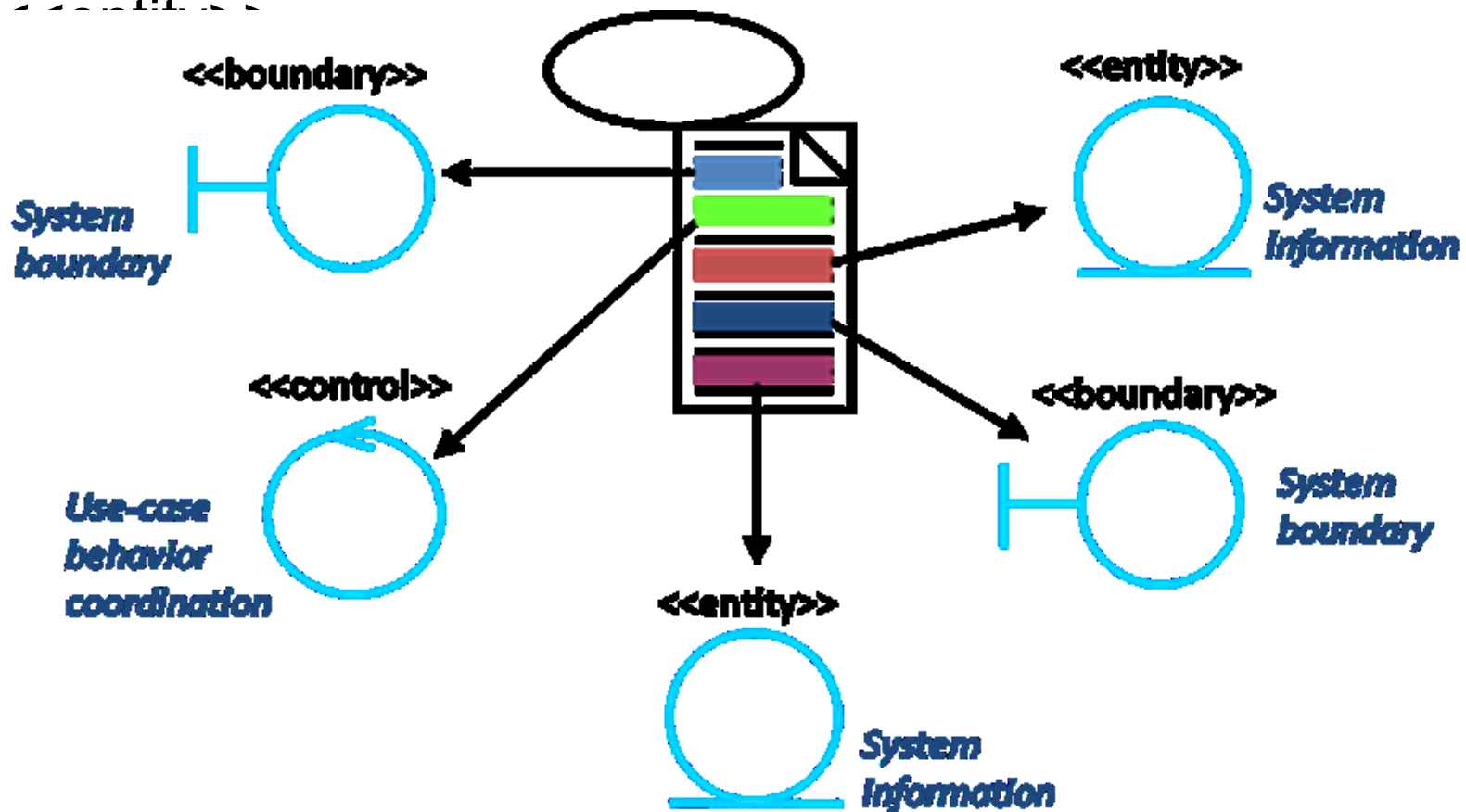
- **Xác định các lớp phân tích như thế nào?** Hầu như không có một công thức chung cho việc phát hiện ra các lớp.
  - Tìm các lớp là một công việc đòi hỏi sự sáng tạo được thực hiện với sự trợ giúp của chuyên gia ứng dụng.
  - Quá trình phân tích và thiết kế có tính lặp: **danh sách các lớp sẽ thay đổi theo thời gian.**
  - Tập hợp của các lớp tìm ra ban đầu chưa chắc đã là tập hợp cuối cùng của các lớp sẽ được thực thi và biến đổi thành code sau này. Nhiều thành phần trong giai đoạn phân tích có thể bị biến đổi hoặc mất đi khi sang pha thiết kế.

# Phần II: Phân tích usecase

- **Ba khía cạnh của hệ thống có thể sẽ thay đổi:**
  - 1. Ranh giới giữa hệ thống và các tác nhân của nó
  - 2. Thông tin hệ thống sử dụng
  - 3. Logic điều khiển của hệ thống
- → Trong nỗ lực cô lập các bộ phận của hệ thống có thể sẽ thay đổi, chúng ta sẽ “đóng hộp” những thay đổi đó vào các lớp. Trên cơ sở đó mỗi usecase được phân rã thành các thành phần thuộc vào một trong ba loại lớp phân tích sau:
  - Lớp biên (ranh giới)
  - Lớp thực thể (chứa thông tin)
  - Lớp điều khiển (logic điều khiển)

# Phần II: Phân tích usecase

- Các loại lớp phân tích khác nhau có thể được biểu diễn bằng các biểu tượng khác nhau hoặc với tên của khuôn mẫu trong cặp dấu (<< >>): <<boundary>>, << control>>, << entity>>



# Phần II: Phân tích usecase

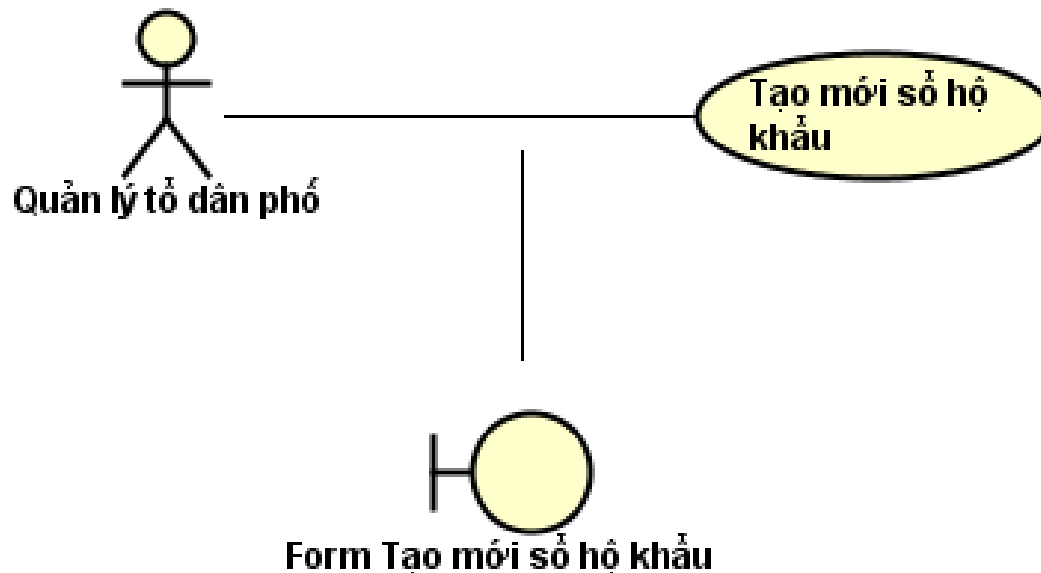
- **Boundary classes**

- Một lớp biên là giao diện trung gian giữa hệ thống và một cái gì đó bên ngoài. Các lớp ranh giới cách ly hệ thống khỏi những thay đổi của môi trường xung quanh (ví dụ, những thay đổi về giao diện), giữ cho những thay đổi này không ảnh hưởng đến phần còn lại của hệ thống.
- Một số hướng dẫn để tìm các lớp biên:
  - Một khuyến nghị cho việc xác định ban đầu các lớp biên là: **mỗi cặp tác nhân / ca sử dụng → xác định ít nhất 1 lớp biên**.
  - Một hệ thống có thể có một số loại lớp biên (theo phân loại của actor):
    - *Các lớp giao diện người dùng*
    - *Các lớp giao diện hệ thống*: ví dụ: các giao diện gọi các API của một hệ thống khác.
    - *Các lớp giao diện thiết bị*

# Phần II: Phân tích usecase

- **Boundary classes**

- Với usecase “Tạo mới sổ hộ khẩu” → Form Tạo mới sổ hộ khẩu



# Phần II: Phân tích usecase

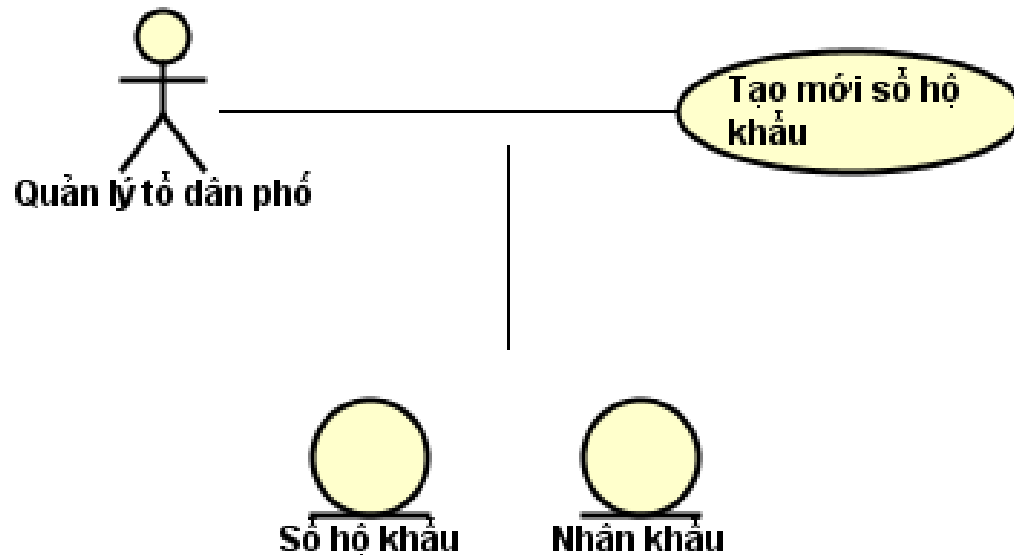
- **Entity classes**

- Các lớp thực thể đại diện cho các kho thông tin trong hệ thống. Chúng thường được sử dụng để đại diện cho các khái niệm chính mà hệ thống quản lý.
- Đối tượng thực thể (các thể hiện của lớp thực thể) được sử dụng để lưu giữ và cập nhật thông tin về một sự kiện, một người hoặc một đối tượng ngoài đời thực.
- Một số hướng dẫn để tìm các lớp thực thể:
  - Đầu vào lấy **luồng sự kiện theo usecase**, gạch dưới **các cụm danh từ** trong luồng sự kiện. Chúng tạo thành danh sách ứng viên ban đầu của các lớp thực thể.
  - Thực hiện lọc bỏ các danh từ: Loại bỏ các ứng viên thừa (trùng lặp), Loại bỏ các ứng viên mơ hồ, Loại bỏ các tác nhân (ngoài phạm vi), Loại bỏ các cấu trúc triển khai, Loại bỏ các thuộc tính (lưu để sử dụng sau), Loại bỏ các hoạt động,... → còn lại là các ứng viên cho lớp thực thể của usecase



# Phần II: Phân tích usecase

- **Entity classes**
- Với usecase “Tạo mới sổ hộ khẩu” → Entity sổ hộ khẩu, Entity nhân khẩu



# Phần II: Phân tích usecase

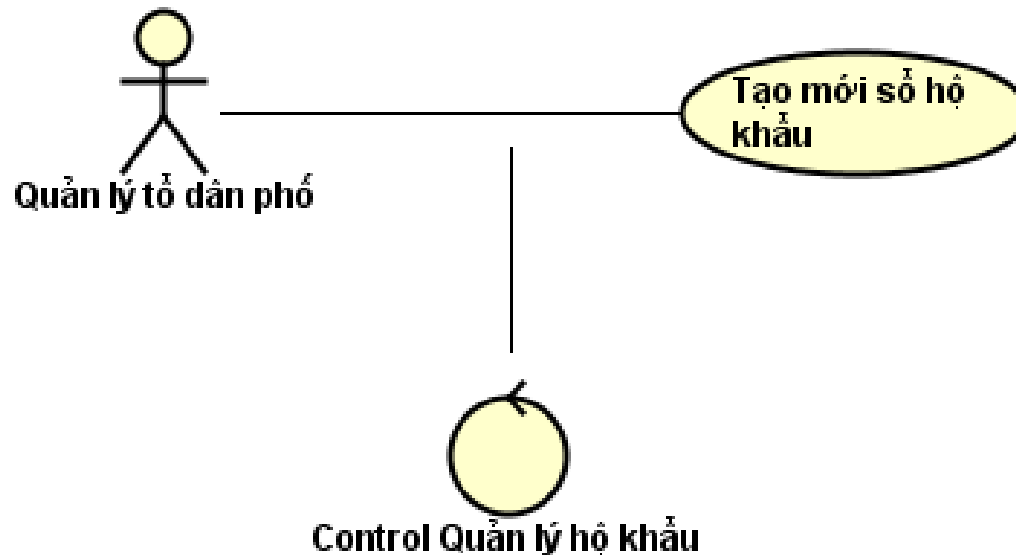
- **Control classes**

- Các lớp điều khiển cung cấp khả năng phối hợp hành vi trong hệ thống.
- Xác định logic điều khiển (thứ tự giữa các sự kiện) và các giao dịch trong một usecase
- Một số hướng dẫn để tìm các lớp điều khiển:
  - **Mỗi usecase xác định ít nhất một lớp điều khiển.**
  - Các usecase chỉ liên quan đến thao tác đơn giản đối với thông tin được lưu trữ hệ thống có thể chỉ sử dụng các lớp thực thể và lớp biên, không dùng lớp điều khiển.
  - Các usecase phức tạp thường yêu cầu **một hoặc nhiều lớp điều khiển** để điều phối hành vi của các đối tượng khác trong hệ thống. Ví dụ về các lớp điều khiển bao gồm quản lý giao dịch, điều phối tài nguyên và xử lý lỗi.

# Phần II: Phân tích usecase

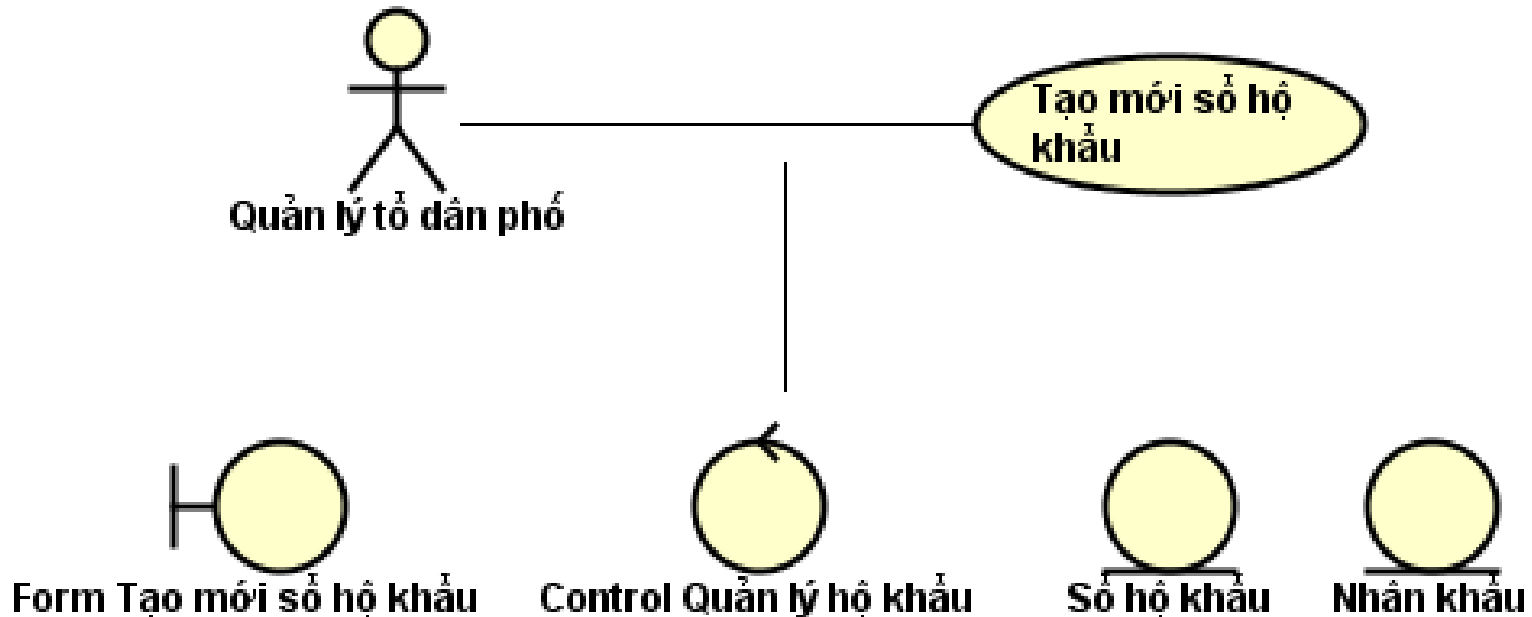
- **Control classes**

- Với usecase “Tạo mới sổ hộ khẩu” → Control Quản lý hộ khẩu



# Phần II: Phân tích usecase

- Kết quả quá trình phân rã bước đầu của usecase “Tạo mới sổ hộ khẩu”:



Thực hiện tương tự với các usecase khác, chúng ta thu được một mô hình phân tích bao gồm các lớp phân tích trong đó. Chú ý: nếu trong các kết quả có các lớp phân tích trùng nhau (ví dụ trong một usecase khác cũng xuất hiện lớp thực thể “Sổ hộ khẩu”) thì cần hợp nhất các lớp phân tích này.

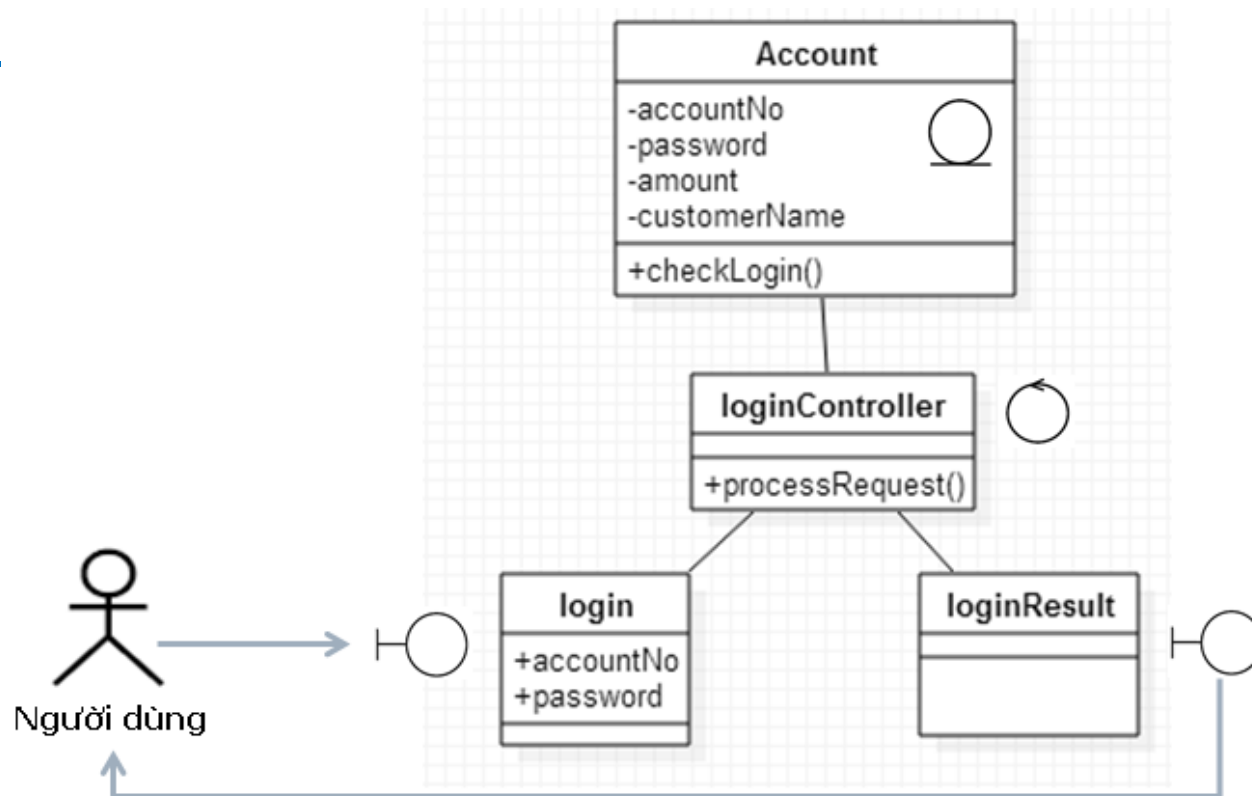
# Phần II: Phân tích usecase

- Bước tiếp theo chúng ta cần xác định các thông tin chi tiết hơn cho các lớp phân tích này:
  - Mỗi lớp: xác định các thuộc tính và phương thức
  - Quan hệ giữa các lớp

# Phần II: Phân tích usecase

- **Bài tập:** Phân rã usecase “Đăng nhập”, xác định các lớp phân tích.

- **Gợi ý:**

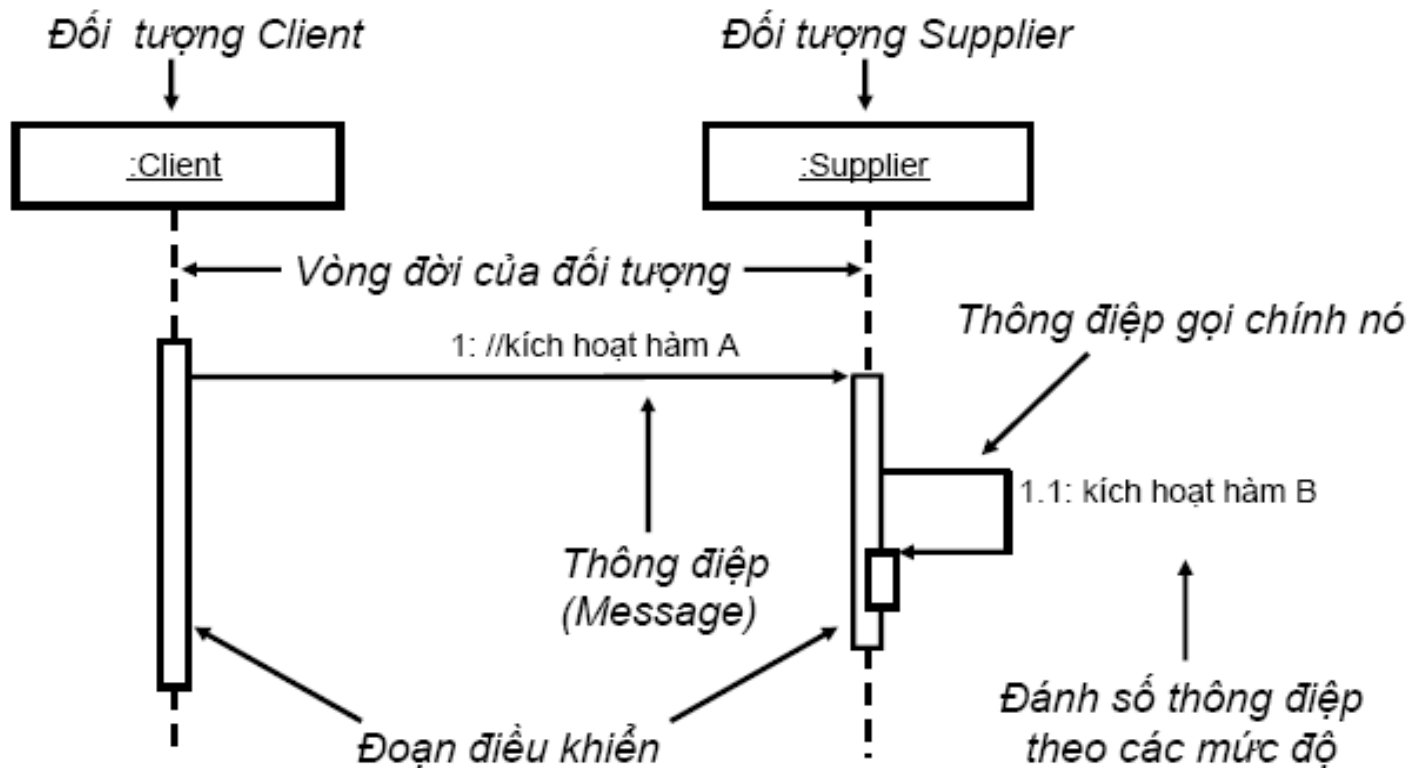


# Phần II: Phân tích usecase

- **Phân bổ trách nhiệm ca sử dụng cho các đối tượng của các lớp phân tích**
  - Với mỗi usecase: chúng ta cần phân bổ trách nhiệm ca sử dụng cho các đối tượng của các lớp phân tích.
  - Đây là một hoạt động quan trọng và đôi khi khó khăn, nó là cơ sở để chúng ta xác định các dữ liệu thành phần (phương thức + thuộc tính) cho mỗi lớp.
  - Kết quả của quá trình này có thể biểu diễn bằng **biểu đồ trình tự (sequence diagram)** hoặc **biểu đồ giao tiếp (communication diagram)** trong UML.

# Phần II: Phân tích usecase

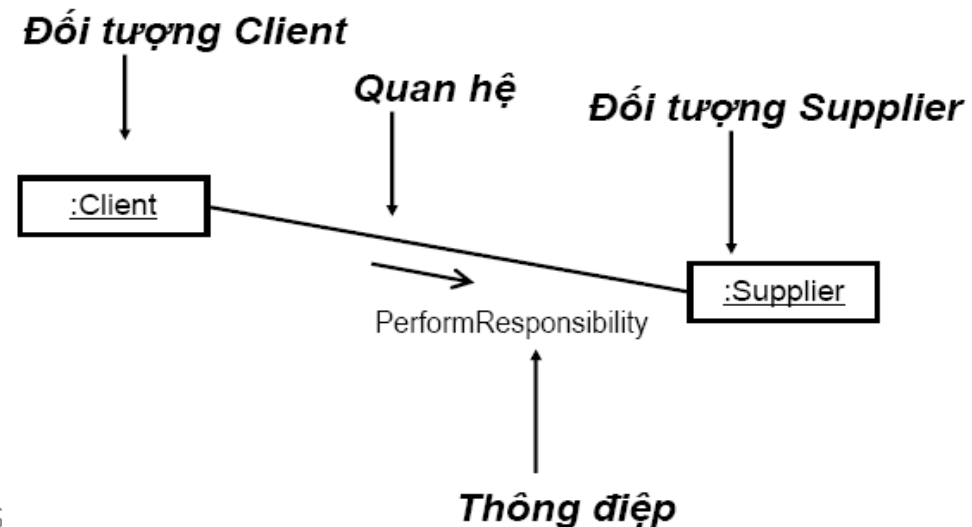
- **Biểu đồ trình tự (sequence diagram):** Là biểu đồ tương tác tập trung vào thứ tự trao đổi các thông điệp theo thời gian.
- **Gồm:** Các đối tượng tham gia tương tác và Trình tự các thông điệp trao đổi với nhau.





# Phần II: Phân tích usecase

- **Biểu đồ giao tiếp (communication diagram):** Là biểu đồ tương tác tập trung vào tổ chức các đối tượng tham gia tương tác.
- **Gồm:** Các đối tượng tham gia tương tác. Đường liên kết giữa các đối tượng. Thông điệp trao chuyển giữa các đối tượng. Tập trung vào sự kiện có nghĩa là chú ý đặc biệt đến mối quan hệ (nối kết) giữa các đối tượng.

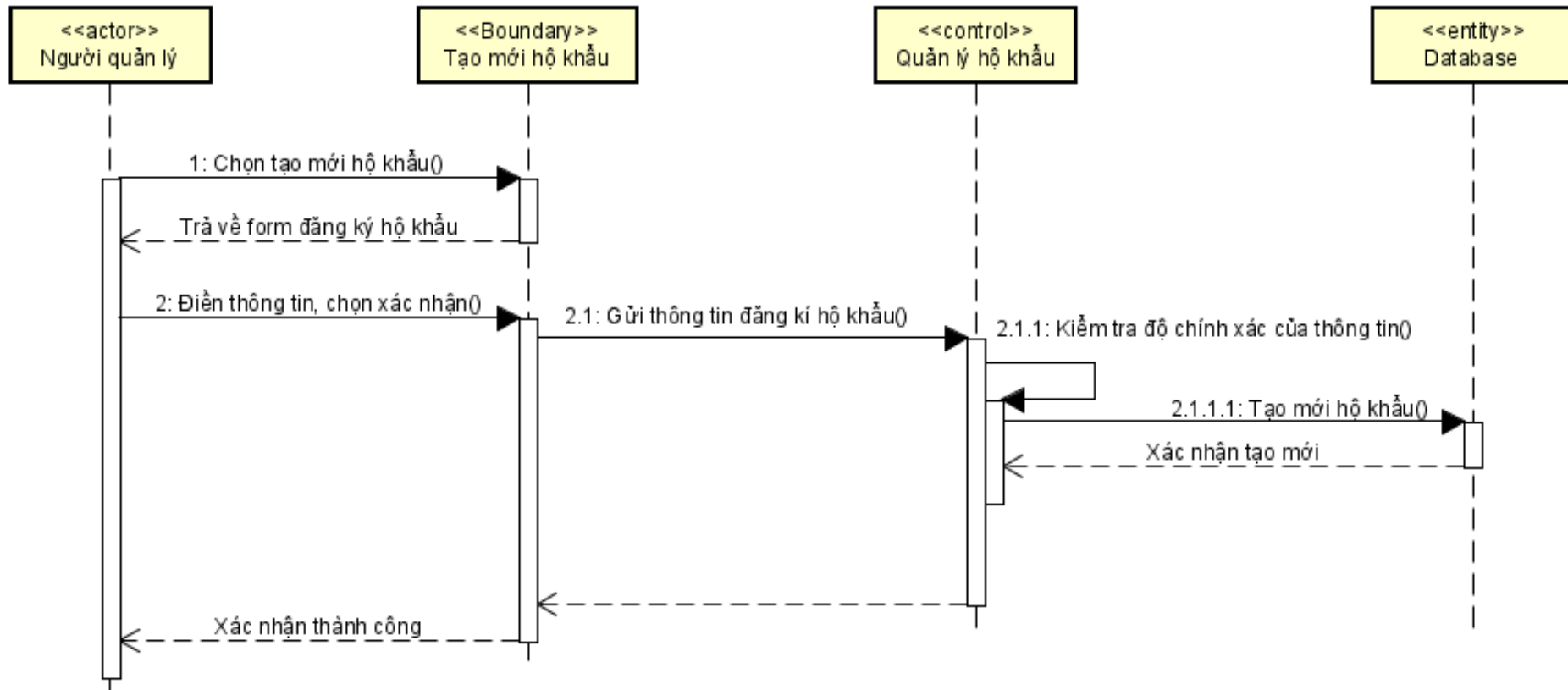


# Phần II: Phân tích usecase

- Bài tập: Từ mô tả chi tiết kịch bản usecase hãy Xây dựng biểu đồ trình tự cho usecase “Tạo mới sổ hộ khẩu”: xác định trình tự các thông điệp tương tác giữa các đối tượng của các lớp phân tích trong usecase này.

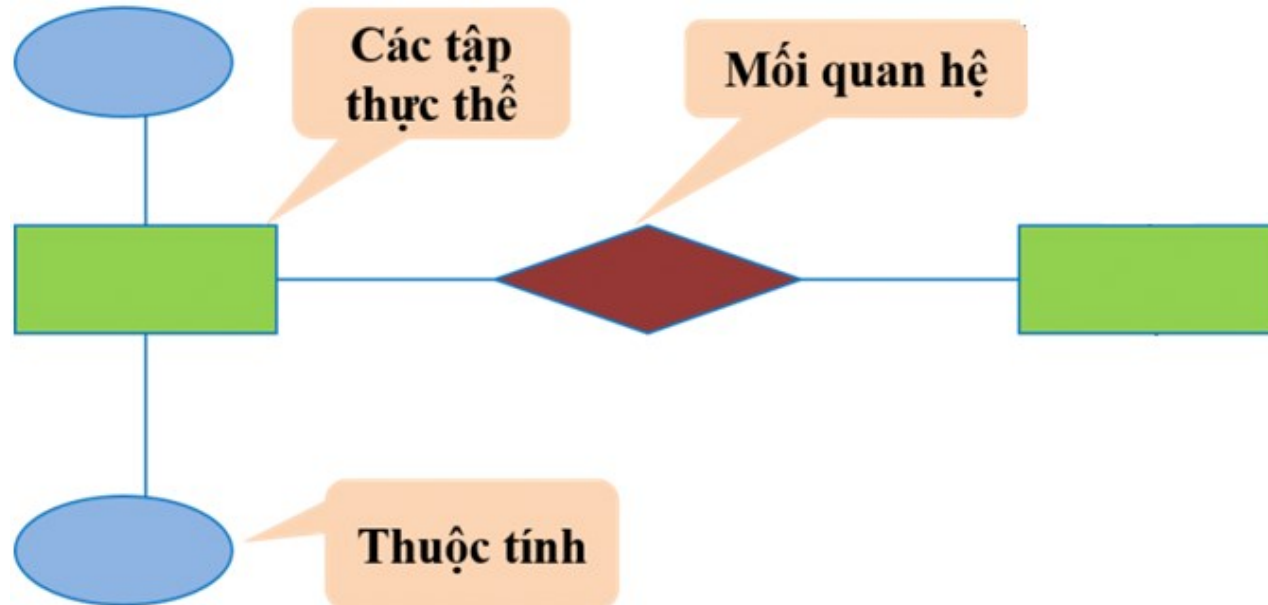
# Phần II: Phân tích usecase

- Gợi ý:



# Phần II: Phân tích usecase

- Phân tích dữ liệu cho hệ thống với biểu đồ thực thể - quan hệ (ERD)
  - Xác định các đối tượng dữ liệu
  - Xác định các đặc tính của các đối tượng dữ liệu
  - Thiết lập các mối quan hệ giữa các đối tượng dữ liệu

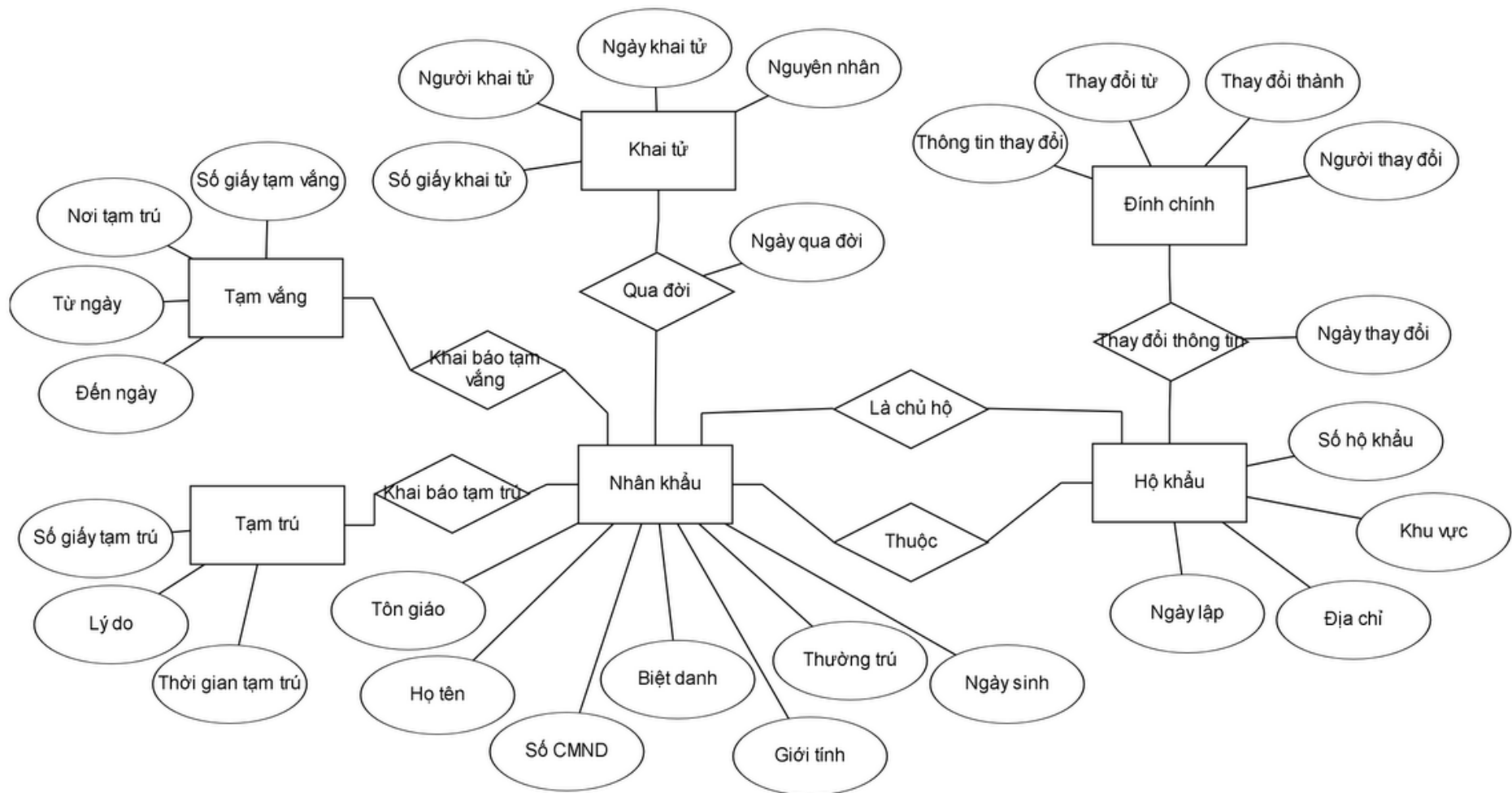


# Phần II: Phân tích usecase

- Bài tập: Xây dựng mô hình dữ liệu ERD cho nhóm chức năng số 1: “1. Quản lý thông tin hộ khẩu, nhân khẩu”.

# Phần II: Phân tích usecase

- Gợi ý:



# Đặc tả các yêu cầu cho bài toán (casestudy) với usecase

- **Yêu cầu:**

- Hoàn thành Phân tích các yêu cầu cho bài toán (casestudy) với các nội dung: phân rã các lớp phân tích, xây dựng biểu đồ trình tự (biểu đồ giao tiếp), mô hình dữ liệu ERD.
- Phần nội dung này các nhóm làm vào trong file .docx (báo cáo)
- Các nhóm chuẩn bị thêm một **slide powerpoint** về nội dung Phân tích các yêu cầu ở trên, buổi học tiếp theo sẽ trình bày.