

## KIẾN TRÚC TÍCH HỢP HỆ THỐNG

### ÔN TẬP CUỐI KÌ

#### Đề cương ôn tập cuối kỳ

#### 1. Nội dung ôn tập:

- **Bảng phân tích yêu cầu người dùng:**
  - **Functional requirements** (Yêu cầu chức năng): Mô tả các tính năng mà hệ thống cần phải thực hiện.
  - **Non-Functional requirements** (Yêu cầu phi chức năng): Mô tả các yếu tố như hiệu suất, bảo mật, khả năng mở rộng, tính dễ sử dụng, v.v.
- **Bảng kiểm thử (Test case):**
  - Viết các test case mô tả cách thức kiểm tra từng yêu cầu của hệ thống, bao gồm các điều kiện đầu vào, kết quả mong đợi và bước thực hiện.
- **Sơ đồ thành phần (Component diagram):**
  - Sử dụng UML 2.0 để mô tả các thành phần của hệ thống và cách chúng giao tiếp với nhau. Các thành phần cần sử dụng <<stereotype>> để xác định loại thành phần.
- **Sơ đồ triển khai (Deployment diagram):**
  - Mô tả cách hệ thống được triển khai trên phần cứng, bao gồm các máy chủ, thiết bị và kết nối giữa chúng.
- **Sơ đồ Use case:**
  - Xác định các chức năng chính trong hệ thống và mối quan hệ với người dùng (actors).
- **Sơ đồ Activity:**
  - Diễn giải các bước trong một quá trình hoặc hành động của người dùng trong hệ thống.
- **Sơ đồ Sequence:**
  - Mô tả các bước và tương tác giữa các thành phần trong hệ thống, thường là tương tác theo thứ tự thời gian.

- **Sơ đồ State:**
    - Mô tả các trạng thái của một đối tượng trong hệ thống và các sự kiện làm thay đổi trạng thái đó.
- 

## 2. Mẹo và gợi ý khi làm bài:

- **Thi bằng tiếng Việt:**
  - Đảm bảo sử dụng ngôn ngữ chính xác và rõ ràng khi giải thích các sơ đồ và mô hình.
- **Vẽ model trước:**
  - Các sơ đồ như use case và test case thường phức tạp hơn, vì vậy bạn nên bắt đầu bằng việc vẽ các mô hình như sơ đồ thành phần và triển khai trước, sơ đồ Activity và State.
- **Đọc kỹ đề bài 3 lần:**
  - Đọc kỹ đề bài để hiểu các yêu cầu và xác định được các yếu tố quan trọng cần có trong sơ đồ và bảng phân tích yêu cầu.
- **Sơ đồ Use case:**
  - Xác định rõ chức năng chính của hệ thống và các đối tượng tương tác với hệ thống (actors). Đọc kỹ đề bài để xác định đúng các chức năng liên quan.
- **Sơ đồ thành phần và triển khai:**
  - Dùng UML 2.0 để biểu diễn các thành phần hệ thống và mối quan hệ giữa chúng. Chú ý rằng các thành phần cần có <<stereotype>> để phân loại.
- **Sơ đồ Activity và State:**
  - Đây là hai sơ đồ dễ lấy điểm nhất, vì vậy cần đảm bảo tính chính xác trong việc thể hiện các bước và trạng thái của hệ thống.
- **Sơ đồ Sequence:**
  - Đây là sơ đồ khó nhất, tương tự như lập trình, bạn cần liệt kê các bước theo trình tự thời gian, từ đó mô hình hóa chúng thành sơ đồ sequence.

## ÔN TẬP

### 1. Nội dung ôn tập:

#### a. Bảng phân tích yêu cầu người dùng:

- **Functional Requirements (Yêu cầu chức năng):**
  - Đây là các yêu cầu mô tả tính năng chính của hệ thống, tức là những gì hệ thống cần phải thực hiện. Ví dụ: “Hệ thống phải cho phép người dùng đăng nhập bằng tài khoản email và mật khẩu”.
- **Non-Functional Requirements (Yêu cầu phi chức năng):**
  - Các yêu cầu này liên quan đến các yếu tố như hiệu suất, bảo mật, khả năng mở rộng, tính dễ sử dụng, khả năng duy trì hệ thống, v.v. Ví dụ: “Hệ thống phải xử lý ít nhất 1000 giao dịch mỗi phút” hoặc “Hệ thống phải bảo mật thông tin người dùng bằng mã hóa SSL”.

#### b. Bảng kiểm thử (Test case):

- **Mục đích:** Mô tả cách thức kiểm tra các yêu cầu của hệ thống.
- **Cấu trúc của một Test case:**
  1. **ID:** Mã định danh của test case.
  2. **Description:** Mô tả ngắn gọn mục đích kiểm tra.
  3. **Pre-conditions:** Điều kiện cần có trước khi thực hiện kiểm thử (ví dụ: người dùng phải đăng nhập).
  4. **Test steps:** Các bước thực hiện kiểm tra.
  5. **Expected Result:** Kết quả mong đợi sau khi thực hiện.
  6. **Actual Result:** Kết quả thực tế khi thực hiện kiểm thử.
  7. **Status:** Trạng thái (Pass/Fail).

#### c. Sơ đồ thành phần (Component Diagram):

- **Mục đích:** Mô tả cấu trúc của hệ thống qua các thành phần và cách chúng tương tác với nhau.
- **Cách vẽ:**
  - Sử dụng **UML 2.0** để vẽ sơ đồ.

- Các thành phần (components) cần có <<stereotype>> để chỉ rõ loại thành phần (ví dụ: <<controller>>, <<service>>).
- Mô tả mối quan hệ giữa các thành phần qua giao diện (interface).

#### d. Sơ đồ triển khai (Deployment Diagram):

- **Mục đích:** Mô tả cách hệ thống được triển khai trên phần cứng, cho thấy các máy chủ, thiết bị và kết nối giữa chúng.
- **Cách vẽ:**
  - Vẽ các **node** (máy chủ, thiết bị), và kết nối chúng với **communication paths**.
  - Chú ý đến các yếu tố như **mạng**, **phần cứng** và **phần mềm** trên mỗi node.

#### e. Sơ đồ Use case:

- **Mục đích:** Mô tả các chức năng hệ thống từ góc nhìn của người dùng (actors).
- **Cách vẽ:**
  - **Actors:** Định nghĩa các đối tượng (người dùng hoặc hệ thống khác) tương tác với hệ thống.
  - **Use Cases:** Các chức năng hệ thống mà người dùng có thể thực hiện.
  - Mối quan hệ giữa actors và use cases là **association** (mối liên kết).

#### f. Sơ đồ Activity:

- **Mục đích:** Diễn giải các bước trong một quá trình hoặc hành động của người dùng trong hệ thống.
- **Cách vẽ:**
  - Sử dụng **activity nodes** để mô tả các bước.
  - Các **decision nodes** để mô tả các lựa chọn.
  - Sử dụng **arrows** để chỉ ra luồng công việc.

#### g. Sơ đồ Sequence:

- **Mục đích:** Mô tả các bước và tương tác giữa các thành phần trong hệ thống, thường là theo thứ tự thời gian.
- **Cách vẽ:**
  - Các đối tượng (objects) sẽ được vẽ theo chiều dọc.

- Các thông điệp (messages) giữa các đối tượng được thể hiện qua các mũi tên ngang.
- Sơ đồ sequence mô tả **dòng thời gian** của tương tác.

#### h. Sơ đồ State:

- **Mục đích:** Mô tả các trạng thái của một đối tượng trong hệ thống và các sự kiện làm thay đổi trạng thái đó.
- **Cách vẽ:**
  - Các **state** là các trạng thái khác nhau mà đối tượng có thể ở.
  - **Transitions** giữa các trạng thái xảy ra khi có sự kiện.
  - **Initial state** và **Final state** đánh dấu điểm bắt đầu và kết thúc trong vòng đời của đối tượng.

## 2. Các mẹo và gợi ý khi làm bài thi:

- **Đọc kỹ đề bài:** Đảm bảo bạn hiểu yêu cầu của đề bài trước khi bắt đầu vẽ các sơ đồ. Đọc lại nhiều lần để nắm rõ các yếu tố cần thiết.
- **Vẽ sơ đồ trước:** Bắt đầu từ các sơ đồ dễ thực hiện như sơ đồ activity hoặc state. Sơ đồ sequence thường phức tạp hơn vì nó yêu cầu mô tả chi tiết các bước và tương tác.
- **Chú ý tới chi tiết:** Đảm bảo rằng bạn sử dụng đúng cú pháp UML cho các sơ đồ. Các thành phần phải có tên chính xác và các mối quan hệ phải được thể hiện rõ ràng.
- **Tập trung vào các sơ đồ dễ lấy điểm:** Những sơ đồ như **Activity** và **State** có thể dễ dàng lấy điểm nếu bạn vẽ đúng quy trình và trạng thái. Chú ý làm rõ các bước và sự kiện.
- **Sequence diagram:** Sơ đồ này khó nhất, nhưng bạn có thể giải quyết bằng cách chia nó thành các bước nhỏ. Liệt kê từng bước một và mô tả tương tác theo thời gian.
- **Giải thích rõ ràng:** Khi mô tả các sơ đồ, đừng chỉ vẽ mà hãy giải thích rõ ràng các yếu tố trong sơ đồ, đặc biệt là các **actors**, **components** và **messages**.

Nhắc lại kiến thức

## I. ÔN TẬP VỀ BẢNG PHÂN TÍCH YÊU CẦU NGƯỜI DÙNG.

### a) Yêu cầu chức năng (Functional Requirements)

Stakeholders	Mô tả yêu cầu	Ý kiến	Độ ưu tiên
1. Quản trị viên	Quản trị trang web	Đồng ý	1
1.1. quản lý tài khoản	Mỗi người dùng có một tài khoản trong hệ thống này	Đồng ý	1
1.2. Đặt lại mật khẩu	Người dùng cần đặt lại mật khẩu sẽ được quản trị viên thông báo qua email, SMS	Đồng ý	1

### b) Yêu cầu phi chức năng (Non-Functional Requirements)

Stakeholders	Mô tả yêu cầu	Ý kiến	Độ ưu tiên
1. Quản trị viên	Quản trị trang web	Đồng ý	1
1.1. Chính sách mật khẩu	Mật khẩu phải có độ phức tạp	Đồng ý	1
1.2. Chính sách tên người dùng	Tên người dùng: chữ cái đầu viết hoa, độ dài tối thiểu	Đồng ý	1

### Functional Requirements (Yêu cầu chức năng)

Stakeholders	Descriptions	Ideas	Priority
1. Administrator	Quản lý hệ thống tài khoản	OK	1
1.1 Quản lý tài khoản			
1.1.1 Thêm tài khoản mới	1. Chọn chức năng tạo tài khoản, hiển thị form. 2. Nhập thông tin tài khoản. 3. Nhấn nút <b>Tạo tài khoản</b> . 4. Hiển thị thông báo:	OK	

Stakeholders	Descriptions	Ideas	Priority
	<ul style="list-style-type: none"> <li>- Nếu thông tin hợp lệ, thông báo thành công.</li> <li>- Nếu thông tin không hợp lệ, đánh dấu đỏ và hiển thị hướng dẫn sửa lỗi.</li> </ul>		
1.1.2 Cập nhật thông tin tài khoản	<ol style="list-style-type: none"> <li>1. Chọn chức năng cập nhật tài khoản, hiển thị danh sách tài khoản.</li> <li>2. Chọn tài khoản cần cập nhật, hiển thị thông tin có thể chỉnh sửa.</li> <li>3. Cập nhật thông tin.</li> <li>4. Nhấn nút <b>OK</b>.</li> <li>5. Hiển thị popup xác nhận với hai nút <b>Yes</b> và <b>No</b>: <ul style="list-style-type: none"> <li>- Nếu chọn <b>Yes</b>: <ul style="list-style-type: none"> <li>+ Nếu thông tin hợp lệ, thông báo thành công.</li> <li>+ Nếu thông tin không hợp lệ, đánh dấu đỏ và hiển thị lỗi.</li> <li>+ Nếu cập nhật thất bại, ghi log hệ thống và thông báo lỗi.</li> </ul> </li> <li>- Nếu chọn <b>No</b>, đóng popup.</li> </ul> </li> </ol>	OK	
1.1.3 Xóa tài khoản	<ol style="list-style-type: none"> <li>1. Chọn tài khoản cần xóa trong danh sách.</li> <li>2. Nhấn nút <b>Xóa</b>.</li> <li>3. Hiển thị xác nhận với nút <b>Yes</b> và <b>No</b>.</li> <li>4. Nếu chọn <b>Yes</b>, thực hiện xóa.</li> <li>5. Thông báo thành công hoặc thất bại.</li> </ol>	OK	
1.1.4 Tìm kiếm tài khoản	<ol style="list-style-type: none"> <li>1. Nhập tên hoặc email để tìm kiếm.</li> <li>2. Hiển thị danh sách kết quả khớp với từ khóa tìm kiếm.</li> </ol>	OK	
1.1.5 Đặt lại mật khẩu tài khoản	<ol style="list-style-type: none"> <li>1. Chọn chức năng đặt lại mật khẩu.</li> <li>2. Nhập email hoặc tên tài khoản cần đặt lại mật khẩu.</li> <li>3. Hệ thống gửi email hoặc mã OTP để xác nhận.</li> <li>4. Người dùng nhập mã OTP hoặc xác nhận qua link trong email.</li> <li>5. Mật khẩu mới được tạo và thông báo.</li> </ol>	OK	

Stakeholders	Descriptions	Ideas	Priority
<b>1.2 Quản lý dữ liệu</b>	Sao lưu và phục hồi dữ liệu hệ thống		
1.2.1 Sao lưu dữ liệu	1. Chọn chức năng sao lưu dữ liệu. 2. Hệ thống tự động tạo file sao lưu và thông báo kết quả.	OK	
1.2.2 Phục hồi dữ liệu	1. Chọn chức năng phục hồi dữ liệu. 2. Chọn file sao lưu để phục hồi. 3. Xác nhận thực hiện và thông báo kết quả.	OK	
<b>1.3 Quản lý phần cứng</b>	Thêm, xóa, cập nhật thiết bị phần cứng		
1.3.1 Thêm thiết bị mới	1. Chọn chức năng thêm thiết bị, hiển thị form nhập thông tin. 2. Nhập thông tin thiết bị. 3. Nhấn nút <b>Thêm</b> và thông báo kết quả.	OK	
1.3.2 Cập nhật thông tin thiết bị	1. Chọn chức năng cập nhật thiết bị, hiển thị danh sách thiết bị. 2. Chọn thiết bị cần cập nhật. 3. Thay đổi thông tin thiết bị và lưu thay đổi. 4. Thông báo kết quả cập nhật.	OK	

### Non-Functional Requirements (Yêu cầu phi chức năng)

Stakeholders	Descriptions	Ideas	Priority
<b>1. Administrator</b>	Quản lý hệ thống với các tiêu chí an toàn và hiệu năng cao	OK	1
<b>1.1 Authentication and Authorization</b>	Người dùng phải đăng nhập trước khi thực hiện bất kỳ tác vụ nào.	OK	1
<b>1.2 Password Policies</b>	1. Mật khẩu phải có độ phức tạp cao (ít nhất 8 ký tự, có số, chữ thường, chữ hoa, ký tự đặc biệt). 2. Mật khẩu phải được thay đổi mỗi tháng. 3. Nhập sai mật khẩu 3 lần, tài khoản bị khóa.	OK	1



Stakeholders	Descriptions	Ideas	Priority
<b>1.3 Username Policies</b>	1. Tên đăng nhập phải dài ít nhất 8 ký tự. 2. Mỗi người dùng chỉ được có một tài khoản với tên đăng nhập duy nhất.	OK	1
<b>1.4 High Availability</b>	Hệ thống phải đảm bảo hoạt động 24/7, không bị gián đoạn.	OK	1
<b>1.5 Data Backup and Recovery</b>	Hệ thống phải hỗ trợ sao lưu dữ liệu tự động và phục hồi nhanh chóng.	OK	1
<b>1.6 Security</b>	Dữ liệu phải được mã hóa khi lưu trữ và truyền tải.	OK	1

### Hướng dẫn tạo bảng yêu cầu người dùng

- Xác định Stakeholders:** Liệt kê các nhóm người dùng hoặc vai trò liên quan (Administrator, User, etc.).
- Liệt kê chức năng (Functional Requirements):**
  - Xác định các tính năng mà hệ thống cần cung cấp.
  - Mô tả chi tiết từng bước, kết quả, và thông báo cần thiết.
- Liệt kê yêu cầu phi chức năng (Non-Functional Requirements):**
  - Đảm bảo yếu tố bảo mật, hiệu suất, tính sẵn sàng của hệ thống.
- Sử dụng bảng mẫu:**
  - Sắp xếp các yêu cầu theo mức độ ưu tiên (**Priority**) từ cao đến thấp.

Ôn thêm

### Functional Requirements (Yêu cầu chức năng)

Stakeholders	Descriptions	Ideas	Priority
<b>1. Administrator</b>	Quản lý hệ thống công ty	OK	1
<b>1.1 Quản lý tài khoản</b>		OK	1

Stakeholders	Descriptions	Ideas Priority
<b>1.1.1 Thêm tài khoản mới</b>	<ol style="list-style-type: none"> <li>1. Chọn chức năng tạo tài khoản, hiển thị form.</li> <li>2. Nhập thông tin tài khoản.</li> <li>3. Nhấn nút "Tạo tài khoản".</li> <li>4. Hiển thị thông báo: <ul style="list-style-type: none"> <li>- Nếu hợp lệ: Popup "Tạo tài khoản thành công".</li> <li>- Nếu không hợp lệ: Hiển thị popup lỗi, đánh dấu đỏ phần thông tin cần sửa.</li> </ul> </li> </ol>	OK
<b>1.1.2 Cập nhật thông tin tài khoản</b>	<ol style="list-style-type: none"> <li>1. Chọn chức năng cập nhật, hiển thị danh sách tài khoản.</li> <li>2. Chọn tài khoản cần cập nhật, form hiển thị thông tin tài khoản.</li> <li>3. Thay đổi thông tin.</li> <li>4. Nhấn "OK".</li> <li>5. Hiển thị popup xác nhận với nút "Yes" hoặc "No": <ul style="list-style-type: none"> <li>- Nếu chọn <b>Yes</b>: <ul style="list-style-type: none"> <li>+ Nếu hợp lệ, thông báo cập nhật thành công.</li> <li>+ Nếu không hợp lệ, đánh dấu lỗi và hiển thị hướng dẫn.</li> <li>+ Nếu cập nhật thất bại, ghi log và hiển thị thông báo lỗi.</li> </ul> </li> <li>- Nếu chọn <b>No</b>, đóng popup.</li> </ul> </li> </ol>	OK
<b>1.1.3 Xóa tài khoản</b>	<ol style="list-style-type: none"> <li>1. Chọn tài khoản cần xóa.</li> <li>2. Nhấn "Xóa".</li> <li>3. Hiển thị popup xác nhận (Yes/No): <ul style="list-style-type: none"> <li>- Nếu chọn Yes, tài khoản sẽ bị xóa và thông báo thành công.</li> <li>- Nếu chọn No, đóng popup và không thực hiện.</li> </ul> </li> </ol>	OK
<b>1.1.4 Tìm kiếm tài khoản</b>	<ol style="list-style-type: none"> <li>1. Nhập từ khóa (tên tài khoản hoặc email) vào ô tìm kiếm.</li> <li>2. Hiển thị danh sách tài khoản phù hợp với từ khóa.</li> </ol>	OK

Stakeholders	Descriptions	Ideas Priority
<b>1.1.5 Đặt lại mật khẩu tài khoản</b>	1. Chọn chức năng "Đặt lại mật khẩu". 2. Nhập email hoặc ID tài khoản. 3. Hệ thống gửi mã xác nhận qua email. 4. Người dùng nhập mã và đặt lại mật khẩu mới. 5. Hiện thị thông báo thành công.	OK
<b>1.1.6 Cấp quyền tài khoản</b>	1. Chọn tài khoản từ danh sách. 2. Chọn quyền cần cấp. 3. Nhấn "Lưu thay đổi". 4. Hiện thị thông báo quyền đã được cấp.	OK
<b>1.2 Sao lưu dữ liệu</b>	1. Chọn chức năng sao lưu. 2. Hệ thống tạo file sao lưu và thông báo kết quả.	OK
<b>1.3 Khôi phục dữ liệu</b>	1. Chọn chức năng khôi phục. 2. Chọn file sao lưu để khôi phục. 3. Hệ thống tiến hành khôi phục và thông báo kết quả.	OK
<b>1.4 Quản lý phần cứng</b>	1. Thêm mới, xóa hoặc cập nhật thông tin phần cứng.	OK

---

### Non-Functional Requirements (Yêu cầu phi chức năng)

Stakeholders	Descriptions	Ideas Priority
<b>1. Administrator</b>	Đảm bảo hệ thống bảo mật và hoạt động ổn định	1
<b>1.1 Authentication and Authorization</b>	Người dùng phải đăng nhập trước khi thực hiện bất kỳ thao tác nào.	1
<b>1.2 Password Policies</b>	- Mật khẩu phải có ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, số và ký tự đặc biệt. - Người dùng phải thay đổi mật khẩu mỗi 30 ngày. - Sai mật khẩu 3 lần, tài khoản sẽ bị khóa.	OK

Stakeholders	Descriptions	Ideas	Priority
1.3 Username Policies	- Tên tài khoản phải là duy nhất và có ít nhất 8 ký tự. - Mỗi người dùng chỉ được tạo một tài khoản.	OK	
1.4 Permission Policies	Quyền truy cập phải được kiểm soát chặt chẽ và cấp phát theo vai trò.	OK	1
1.5 High Availability	Hệ thống phải đảm bảo hoạt động liên tục 24/7, với thời gian ngừng hoạt động tối đa 1 giờ/tháng.	OK	1

## II. ÔN TẬP VỀ USECASE

### 1. Khái niệm cơ bản về Use Case

- **Use Case:** Mô tả các chuỗi hành động mà hệ thống thực hiện nhằm cung cấp kết quả có ý nghĩa cho **actor** (tác nhân).
- **Actor:** Là người dùng hoặc hệ thống khác tương tác với hệ thống.

#### Mục đích của Use Case:

- Mô hình hóa các hành động hệ thống thực hiện.
- Cung cấp cái nhìn tổng quan về hệ thống.
- Là cơ sở để giao tiếp giữa người dùng và nhà phát triển.
- Hỗ trợ đặc tả kiểm tra (test specification).

### 2. Các ký hiệu chính trong Use Case

- **Use Case:** Biểu diễn bằng hình ellipse, tên viết bên trong hoặc bên dưới.
  - Quy tắc đặt tên: Bắt đầu bằng động từ, ví dụ: "Đặt hàng", "Xử lý thanh toán".
- **Actor:** Biểu diễn bằng hình người, tên vai trò đặt bên dưới.
- **Relationship (Quan hệ):** Dùng để kết nối giữa **actor** và **use case**, hoặc giữa các **use case**.

### 3. Các kiểu kết hợp và quan hệ trong Use Case

#### 3.1 Generalization (Tổng quát hóa):

- **Giữa Use Case:**
  - Khi các use case cụ thể có chung một số hành động, nhưng khác nhau trong chi tiết thực hiện.
  - **Ký hiệu:** Mũi tên tam giác rỗng, hướng từ use case cụ thể đến use case tổng quát.
  - **Ví dụ:** "Thêm câu hỏi từ file Word" và "Nhập câu hỏi" là các use case cụ thể của "Thêm ngân hàng câu hỏi".
- **Giữa Actor:**
  - Khi các actor chia sẻ hành vi chung.
  - **Ký hiệu:** Mũi tên tam giác rỗng, hướng từ actor cụ thể đến actor tổng quát.
  - **Ví dụ:** "Nhân viên trả lương tháng" và "Nhân viên làm hợp đồng" thuộc actor tổng quát "Nhân viên".

#### 3.2 Include (Bao gồm):

- **Ý nghĩa:** Khi một số hành động chung được tách thành một use case riêng và được chia sẻ bởi nhiều use case khác.
- **Ký hiệu:** Mũi tên nét đứt, kèm nhãn "include".
- **Ví dụ:** Use case "Mượn sách" và "Trả sách" đều bao gồm use case "Kiểm tra thẻ thư viện".

#### 3.3 Extend (Mở rộng):

- **Ý nghĩa:** Dùng để thêm các hành vi đặc biệt hoặc tùy chọn vào use case chính.
- **Ký hiệu:** Mũi tên nét đứt, kèm nhãn "extend".
- **Ví dụ:** Use case "Bảo mật nội dung" mở rộng từ "Gửi email".

---

### 4. Xác định Actor và Use Case

#### 4.1 Xác định Actor:

- Actor có thể là:
  - Người sử dụng hệ thống.

- Hệ thống khác tương tác với hệ thống.
- Sự kiện thời gian (ví dụ: đến hạn thanh toán).
- **Câu hỏi xác định actor:**
  - Ai nhập/xuất dữ liệu từ hệ thống?
  - Hệ thống cần tương tác với hệ thống nào khác?
  - Ai chịu trách nhiệm quản lý hoặc duy trì hệ thống?

#### 4.2 Xác định Use Case:

- **Use Case:** Thể hiện chức năng hệ thống thực hiện để đạt được kết quả cụ thể.
- **Câu hỏi xác định Use Case:**
  - Nhiệm vụ của mỗi actor là gì?
  - Có actor nào tạo, lưu trữ, thay đổi, hoặc đọc thông tin trong hệ thống không?

### 5. Luồng sự kiện và mô tả Use Case

#### Luồng sự kiện:

- Mô tả chi tiết các bước cần thực hiện để đạt được mục tiêu của Use Case.
- **Các nội dung cần có:**
  - Điều kiện bắt đầu.
  - Luồng chính.
  - Các luồng phụ, luồng rẽ nhánh.

#### Mẫu mô tả Use Case:

1. **Tên Use Case:** Bắt đầu bằng động từ.
2. **Điều kiện bắt đầu:** Điều kiện cần để bắt đầu Use Case.
3. **Luồng sự kiện chính:** Các bước chính diễn ra trong Use Case.
4. **Luồng rẽ nhánh/phụ:** Xử lý các trường hợp ngoại lệ hoặc tùy chọn.

#### Ví dụ:

- Use Case: Rút tiền ATM.

- **Điều kiện bắt đầu:** Người dùng đưa thẻ vào máy ATM.
  - **Luồng chính:**
    1. Máy yêu cầu nhập PIN.
    2. Người dùng nhập PIN.
    3. Máy xác thực PIN.
    4. Người dùng chọn số tiền muốn rút.
    5. Máy rút tiền và trả thẻ.
  - **Luồng rẽ nhánh:** PIN sai hoặc tài khoản không đủ tiền.
- 

## 6. Phân chia Use Case thành các gói (Package)

- Khi hệ thống phức tạp, các Use Case có thể được phân chia thành các gói nhỏ hơn (package).
  - **Ví dụ:**
    - Hệ thống ATM chia thành:
      - Gói **Giao dịch:** Đăng nhập, Giải quyết PIN không hợp lệ.
      - Gói **Rút/Gửi tiền:** Rút tiền, Gửi tiền, Truy vấn tài khoản.
      - Gói **Vận hành:** Khởi động, Đóng máy.
- 

### Lưu ý quan trọng:

- Tên Use Case cần rõ ràng và bắt đầu bằng động từ.
- Actor và Use Case phải được xác định đúng để tránh sai lệch trong thiết kế hệ thống.
- Quan hệ **Include** dùng để tách phần chung; quan hệ **Extend** dùng để mở rộng phần đặc biệt.

### Quan hệ Include và Extend trong Use Case

Quan hệ **Include** và **Extend** là hai kiểu quan hệ quan trọng trong biểu đồ Use Case. Chúng giúp tổ chức và quản lý các **Use Case** một cách logic, tránh trùng lặp và xử lý được các trường hợp đặc biệt.

---

## 1. Quan hệ Include (Bao gồm)

### Ý nghĩa

- **Include** được sử dụng khi một số hành động (hoặc dòng sự kiện) **chung** giữa nhiều Use Case được tách ra thành một Use Case riêng.
- Điều này giúp tránh việc lặp lại nội dung trong các Use Case và tăng tính tái sử dụng.
- Quan hệ này chỉ ra rằng **Use Case chính luôn luôn phải gọi đến Use Case được bao gồm**.

### Ký hiệu

- Mũi tên nét đứt với nhãn <<include>>.
- Mũi tên hướng từ **Use Case chính** đến **Use Case được bao gồm**.

### Ví dụ cụ thể

#### Hệ thống quản lý thư viện:

- **Use Case chính:** "Mượn sách" và "Trả sách".
- Cả hai Use Case này đều cần thực hiện kiểm tra thẻ thư viện của người dùng trước khi thực hiện hành động chính.
- Vì vậy, "Kiểm tra thẻ thư viện" được tách ra thành một Use Case riêng và được **Include** vào hai Use Case trên.

#### Biểu diễn quan hệ Include:

- Use Case "Mượn sách" và "Trả sách" bao gồm "Kiểm tra thẻ thư viện".
- **Luồng thực thi:**
  1. Khi người dùng muốn mượn sách, hệ thống sẽ kiểm tra thẻ thư viện trước.
  2. Nếu thẻ hợp lệ, hệ thống tiếp tục thực hiện Use Case "Mượn sách".
  3. Nếu không hợp lệ, hệ thống dừng quy trình.

### Ký hiệu biểu đồ:



plaintext

Sao chép mã

Mượn sách ----<<include>>----> Kiểm tra thẻ thư viện

Trả sách ----<<include>>----> Kiểm tra thẻ thư viện

---

## 2. Quan hệ Extend (Mở rộng)

### Ý nghĩa

- **Extend** được sử dụng khi một Use Case **chính** có thể có các hành vi đặc biệt, mở rộng thêm trong một số điều kiện cụ thể.
- Quan hệ này chỉ ra rằng Use Case **được mở rộng (extended)** là **tùy chọn**, không phải lúc nào cũng xảy ra.
- Use Case phụ thuộc vào điều kiện hoặc ngữ cảnh để kích hoạt.

### Ký hiệu

- Mũi tên nét đứt với nhãn <<extend>>.
- Mũi tên hướng từ Use Case **mở rộng** đến Use Case **chính**.

### Ví dụ cụ thể

#### Hệ thống ATM:

- Use Case **chính**: "Rút tiền".
- Use Case **mở rộng**: "In biên lai" (hành vi tùy chọn).

#### Luồng thực thi:

1. Người dùng thực hiện Use Case "Rút tiền".
2. Sau khi hoàn tất, hệ thống hỏi người dùng có muốn in biên lai hay không.
3. Nếu người dùng chọn in biên lai, Use Case "In biên lai" sẽ được thực thi (mở rộng).
4. Nếu không, hệ thống kết thúc Use Case "Rút tiền".

### Ký hiệu biểu đồ:

plaintext

Sao chép mã

In biên lai ----<<extend>>----> Rút tiền

---

## So sánh Include và Extend

Đặc điểm	Include (Bao gồm)	Extend (Mở rộng)
Ý nghĩa	Dùng để tách các phần <b>chung</b> trong nhiều Use Case.	Dùng để thêm các hành vi đặc biệt hoặc tùy chọn.
Quan hệ	Luôn luôn được thực hiện bởi Use Case chính.	Chỉ được thực hiện khi có điều kiện cụ thể.
Ký hiệu	<<include>>	<<extend>>
Hướng mũi tên	Từ Use Case chính đến Use Case được bao gồm.	Từ Use Case mở rộng đến Use Case chính.
Ví dụ	"Mượn sách" bao gồm "Kiểm tra thẻ thư viện".	"Rút tiền" mở rộng với "In biên lai".

---

## Mẹo ghi nhớ dễ dàng

### 1. Include (Bao gồm):

- **"Bao gồm là bắt buộc"**: Hành động trong Use Case được bao gồm luôn xảy ra khi thực hiện Use Case chính.
- Hãy nghĩ rằng **Use Case chính cần một phần nội dung chung từ Use Case được bao gồm** để hoạt động.

### 2. Extend (Mở rộng):

- **"Mở rộng là tùy chọn"**: Use Case được mở rộng chỉ xảy ra trong một số trường hợp đặc biệt.
  - Hãy nghĩ rằng **Use Case chính không phụ thuộc hoàn toàn vào Use Case mở rộng**.
- 

## Tóm tắt qua ví dụ:

- **Include:**

- Use Case "Mượn sách" **luôn** kiểm tra thẻ thư viện → Đây là hành động bắt buộc.
- **Extend:**
  - Use Case "Rút tiền" **có thể** in biên lai nếu người dùng yêu cầu → Đây là hành động tùy chọn.