



## Bai tap pvtkht - Bài tập phân tích và thiết kế hệ thống

Hệ thống máy tính (Trường Đại học Công nghiệp Thành phố Hồ Chí Minh)



Scan to open on Studocu

**Trường ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**Khoa Công Nghệ Thông Tin**

-----

**Bài tập thực hành**

**PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

***BỘ MÔN HỆ THỐNG THÔNG TIN***

## MỤC LỤC

Module 1. (3t) Xác định yêu cầu chức năng của hệ thống - mô hình hóa yêu cầu chức năng bằng sơ đồ use case .....	1
Module 2. (6t) Viết đặc tả use case – mô hình hóa bằng Activity .....	11
Module 3. (6t) Phân tích cấu trúc hệ thống - Mô hình hóa bằng Domain .....	15
Module 4. (6t) Thiết kế hệ thống - Sử dụng sơ đồ Sequence – Hiệu chỉnh sơ đồ Domain thành sơ đồ Class .....	20
Module 5. (3t) Thiết kế các thành phần của hệ thống .....	26
Module 6. (3t) Thiết kế hệ thống theo kiến trúc .....	30
Module 7. (3t) Thiết kế cơ sở dữ liệu .....	32

## Module 1. (3t) Xác định yêu cầu chức năng của hệ thống - mô hình hóa yêu cầu chức năng bằng sơ đồ use case

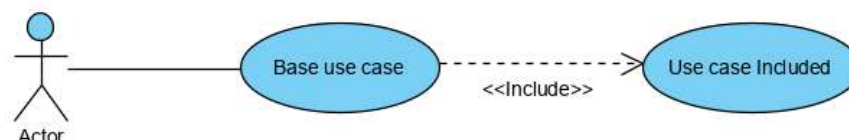
### A) Mục tiêu:

- ✓ Từ đặc tả yêu cầu của khách hàng, sinh viên xác định được các yêu cầu chức năng của hệ thống.
- ✓ Sử dụng sơ đồ **use case** để mô hình hóa các yêu cầu chức năng đã được xác định.

### B) Tóm tắt Kiến thức

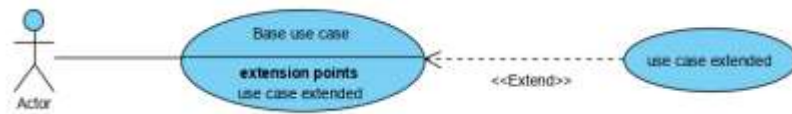
- **Các thành phần trong sơ đồ use case**

- **Actor**: các tác nhân tương tác với hệ thống, actor có thể là người hoặc các hệ thống khác tương tác với hệ thống đang phát triển.
  - + *Actor chính*: tác nhân kích hoạt tương tác với hệ thống
  - + *Actor phụ*: tác nhân tương tác với hệ thống, hỗ trợ actor chính đạt được mục tiêu.
- **Use case**: là danh sách các hành động hoặc các bước sự kiện xác định các tương tác giữa một actor và hệ thống để đạt được mục tiêu xác định.
- **Mối quan hệ**:
  - + **Quan hệ giữa Actor và use case**: là quan hệ **kết hợp (association)**, xác định chức năng mà actor thực hiện trên hệ thống.
    - \* Một Actor phải được liên kết với ít nhất một use case. hoặc nhiều use case.
    - \* Nhiều Actor có thể liên kết với cùng một use case
  - + **Quan hệ giữa use case và use case**:
    - \* **<<include>>**: hành vi của **use case included** là một phần của hành vi của **use case base**, quan hệ **<<include>>** là bắt buộc

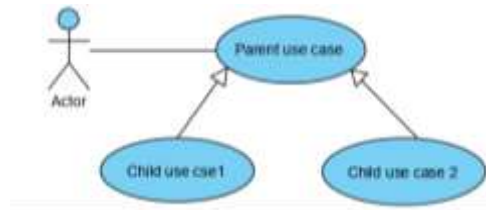


- \* **<<extend>>**: mối quan hệ mở rộng để chỉ định một use case mở rộng hành vi (**extention use case**) của use case khác (**base use case**), mối quan hệ chỉ định *cách thức và thời điểm* hành vi được xác định trong

**extention use case** có thể được chèn vào hành vi được xác định trong **base use case**.

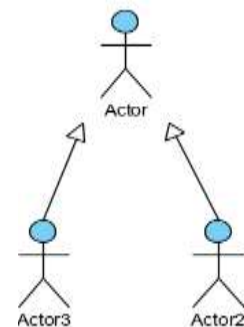


- \* **<<generalization>>**: một use case (child) kế thừa cấu trúc, hành vi và mối quan hệ của một use case khác (parent).



+ **Quan hệ giữa actor và actor:**

- \* **<<generalization>>**: quan hệ tổng quát hóa giữa các actor, một actor có thể kế thừa vai trò của một actor khác.



## C) BÀI TẬP

### Case study 1. Xây dựng hệ thống đăng ký học phần trực tuyến

Việc đăng ký học phần tại trường đại học ABC hiện đang được thực hiện bằng tay. Sinh viên đến trường điền vào các biểu mẫu gồm các thông tin cá nhân và chọn khóa học của họ sau đó nộp lại biểu mẫu cho nhà đăng ký. Thư ký sau đó nhập các lựa chọn vào cơ sở dữ liệu và một quy trình được thực hiện để tạo lịch học cho sinh viên. Quá trình đăng ký mất từ một đến hai tuần để hoàn thành.

Trường đại học quyết định xây dựng một hệ thống **Đăng ký học phần trực tuyến**. Hệ thống này sẽ giúp các giảng viên xem các lớp học mà họ sẽ dạy, các sinh viên chọn các học phần để đăng ký trực tuyến và hoàn tất quá trình đăng ký trong khoảng thời gian ngắn nhất.

**Hệ thống đăng ký học phần trực tuyến** được đặc tả như sau:

Mỗi năm học bao gồm các **học kỳ**, mỗi học kỳ được xác định bằng Mã học kỳ, năm học. Vào đầu mỗi **học kỳ**, hệ thống hiển thị **danh sách các học phần** được mở trong học kỳ đó. Sinh viên có thể xem thông tin chi tiết của mỗi học phần, bằng cách chọn học phần muốn xem, hệ thống hiển thị thông tin chi tiết của học phần được chọn, bao gồm: học phần tiên quyết, số tín chỉ, danh sách các lớp

học phần, ứng với mỗi lớp, hệ thống hiển thị thông tin về ngày, giờ học, **Giảng viên phụ trách**, để giúp sinh viên đưa ra quyết định đúng nhất khi chọn học phần để đăng ký.

- Ngoài ra, mỗi sinh viên phải chọn ra hai học phần thay thế trong trường hợp một học phần bị hủy.
- **Một lớp học phần tối thiểu là 10 sinh viên, và tối đa là 20 sinh viên**, nếu hết thời gian đăng ký mà lớp học phần ít hơn 10 sinh viên thì lớp học phần đó sẽ bị hủy, những sinh viên trong lớp bị hủy sẽ phải chuyển sang học phần thay thế.
- Hệ thống chỉ hiển thị những lớp chưa đủ sĩ số.
- Sau khi sinh viên hoàn tất quá trình đăng ký một học phần thì **Hệ thống đăng ký học phần** sẽ gửi thông tin đến **Hệ thống thanh toán**, sinh viên thực hiện thanh toán và nhận hóa đơn thanh toán học phí sau đó sinh viên có thể xem lịch học của học phần vừa đăng ký.
- Các giảng viên có thể truy cập vào hệ thống để xem danh sách các lớp mà họ giảng dạy, xem danh sách sinh viên của mỗi lớp do họ phụ trách. Cuối mỗi học kỳ, các giảng viên có thể đăng nhập vào hệ thống để nhập điểm cho lớp mà họ phụ trách. Thông tin về **giảng viên bao gồm: Mã giảng viên, học tên, ngày sinh, địa chỉ**.
- Sinh viên có thể đăng nhập vào hệ thống để xem điểm của mình vào cuối mỗi học kỳ. Hệ thống lưu trữ thông tin của **sinh viên bao gồm: Mã sinh viên, họ tên, ngày sinh, địa chỉ**.
- Đối với mỗi học kỳ, sẽ có một khoảng thời gian cho phép sinh viên có thể thay đổi lịch học. Sinh viên truy cập hệ thống trực tuyến trong thời gian này để thêm hoặc hủy các học phần đã đăng ký.
- **Hệ thống thanh toán** sẽ thiết lập trạng thái ghi có cho các sinh viên có học phần bị hủy trong khoảng thời gian này.

### **Yêu cầu:**

1. *Xác định yêu cầu chức năng*
2. *Mô hình hóa yêu cầu chức năng, sử dụng sơ đồ use case.*
3. *Yêu cầu phi chức năng*
4. *Quy tắc nghiệp vụ*
5. *Quy trình nghiệp vụ*

### **Hướng dẫn**

1. **Xác định yêu cầu chức năng của hệ thống:** Dựa vào đặc tả, hệ thống gồm các chức năng sau:

- Sinh viên *xem thông tin chi tiết của các học phần trước khi đăng ký*
- Sinh viên *đăng ký học phần*
- Sinh viên *chọn học phần thay thế*
- Sinh viên *thanh toán học phí*
- Sinh viên *xem lịch học, xem điểm*
- Giảng viên *chọn học phần giảng dạy*
- Giảng viên *xem danh sách lớp giảng dạy*
- Giảng viên *Nhập điểm của lớp giảng dạy*
- Nhân viên *tạo danh mục học phần*
- Nhân viên *quản lý chương trình giảng dạy*
- Nhân viên *Quản lý thông tin Giảng viên*
- Nhân viên *Quản lý thông tin sinh viên.*
- Hệ thống thanh toán thực hiện in hóa đơn khi sinh viên đóng học phí cho học phần đã đăng ký

2. **Mô hình hóa yêu cầu chức năng, sử dụng sơ đồ use case.**

- **Xác định các actor**
  - **Sinh viên:** người đăng ký tham gia các khóa học tại trường Đại học.
  - **Giảng viên:** người tham gia giảng dạy tại trường Đại học.
  - **Nhân viên:** người chịu trách nhiệm duy trì hệ thống đăng ký học phần.
  - **Hệ thống thanh toán:** hệ thống bên ngoài tương tác với hệ thống Đăng ký học phần, thực hiện in hóa đơn cho sinh viên khi sinh viên thanh toán học phí cho học phần đã đăng ký.
- **Xác định các use case:** mỗi use case đại diện cho một chức năng của hệ thống. Để tìm use case trong phần đặc tả hệ thống ta tìm các **động từ** mà nó đại diện cho một chuỗi các hành động tương tác giữa actor và hệ thống để đạt được mục tiêu của actor. Dựa vào đặc tả Hệ thống đăng ký học phần, ta xác định các use case theo từng actor,

#### **Sinh viên**

- + Xem thông tin chi tiết học phần
- + Đăng ký học phần.
- + Chọn học phần thay thế
- + Thanh toán học phí

- + Xem lịch học
- + Xem điểm
- + Đăng nhập: sinh viên phải đăng nhập thành công vào hệ thống để thực hiện các chức năng trên.

### Giảng viên

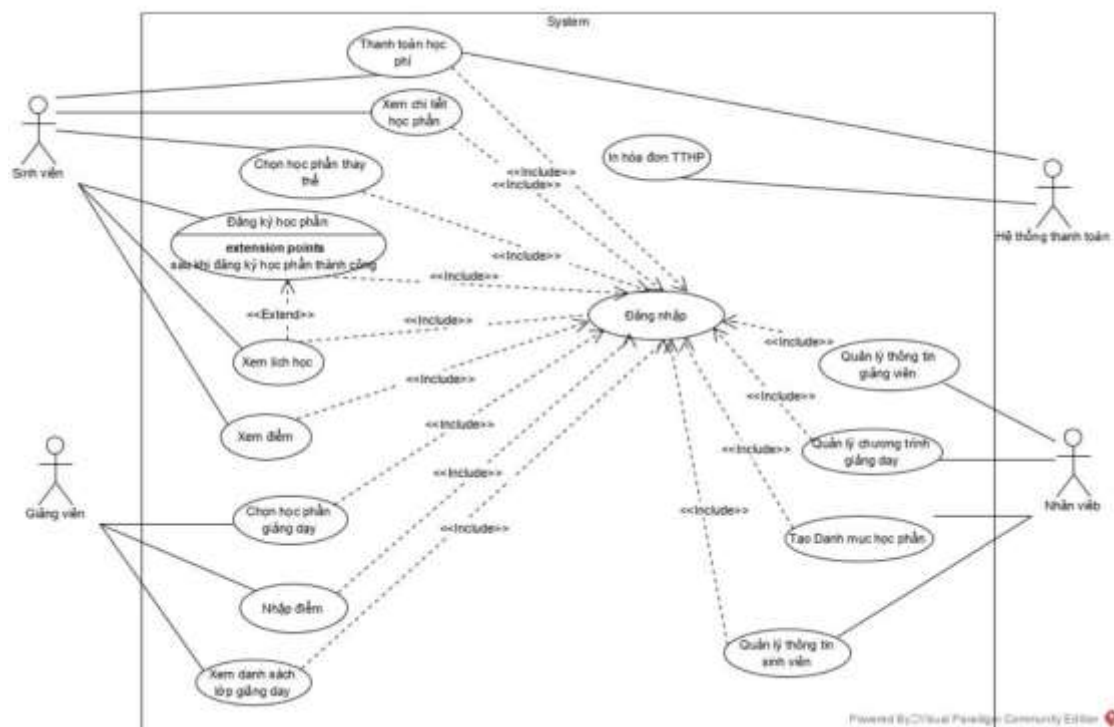
- + Chọn các học phần để giảng dạy.
- + Xem danh sách lớp giảng dạy
- + Nhập điểm của lớp giảng dạy

Giảng viên phải đăng nhập thành công để thực hiện các chức năng trên

### Nhân viên

- + Tạo danh mục học phần.
- + Duy trì chương trình giảng dạy.
- + Quản lý thông tin Giảng viên
- + Quản lý thông tin sinh viên.

### • Sơ đồ use case biểu diễn các chức năng của hệ thống



### 3. Yêu cầu phi chức năng

- Hệ thống phải có giao diện thân thiện, dễ sử dụng
- Có tính bảo mật
- Có tính đúng đắn



- *Độ tin cậy cao*

#### **4. Quy tắc nghiệp vụ**

- *Sinh viên đăng ký một học phần phải đảm bảo đã hoàn tất môn học tiên quyết của học phần sẽ đăng ký.*
- *Mỗi lớp học phần có sĩ số từ 10 đến 20 sinh viên, nếu hết thời gian đăng ký mà sĩ số lớp <10, hệ thống tự động hủy lớp, các sinh viên thuộc lớp bị hủy sẽ được chuyển sang các lớp học phần thay thế đã chọn.*
- *Sinh viên có thể hủy hoặc thay đổi học phần đã đăng ký trong thời gian cho phép.*

#### **5. Quy trình nghiệp vụ**

##### **• Quy trình nghiệp vụ của chức năng đăng ký học phần**

- *Đầu học kỳ, hệ thống sẽ hiển thị những học phần sẽ mở trong học kỳ đó, ứng với mỗi học phần, hệ thống hiển thị danh sách các lớp dự kiến sẽ mở.*
- *Sinh viên cần phải đăng nhập thành công vào hệ thống, sau đó chọn học phần và chọn lớp phù hợp, hệ thống sẽ kiểm tra môn tiên quyết, nếu thỏa điều kiện thì cho phép sinh viên đăng ký.*
- *Nếu lớp đăng ký đã đủ sĩ số, thì hệ thống sẽ thông báo lớp đầy, sinh viên phải chọn lớp khác, nếu lớp còn chỗ thì cho phép đăng ký và thông báo đăng ký thành công. Nếu không thỏa các điều kiện thì sinh viên không thể đăng ký học phần trong học kỳ đó và phải chờ đến học kỳ tiếp theo.*

### **Case study 2. Xây dựng website đặt tour du lịch trực tuyến**

Công ty du lịch **Lucky Tours** mong muốn cung cấp cho khách hàng của họ các dịch vụ tốt nhất về du lịch. Họ cần xây dựng một website để quảng bá các địa điểm du lịch nổi tiếng, cung cấp cho khách hàng nhiều loại tour khác nhau, với nhiều hình thức du lịch đa dạng, phù hợp với nhiều loại đối tượng khách hàng, nhằm giúp cho khách hàng dễ dàng tìm cho mình một tour thích hợp để có thể thư giãn sau những ngày làm việc căng thẳng.

Website **Lucky Tours** hiển thị các địa điểm du lịch theo vùng miền, và thông tin chi tiết về đặc điểm, những điểm tham quan, địa danh nổi tiếng để khách hàng có thể tham khảo.

Website cũng thường xuyên đưa lên những gói **tour** du lịch đa dạng về thời gian, địa điểm, giá cả, dịch vụ và các thông tin khuyến mãi để khách hàng có thể dễ dàng lựa chọn. Với mỗi tour, **website hiển thị các thông tin chi tiết bao gồm: lịch trình, thời gian, địa điểm và thời gian khởi hành, giá tour**. Khi khách hàng chọn một tour cụ thể thì website hiển thị thông tin chi tiết lịch trình mỗi ngày của tour.

Ngoài ra website cũng giúp khách hàng có thể tìm kiếm tour theo từ khóa, hoặc theo các tiêu chí thời gian, địa điểm, **loại tour**, ...bằng cách nhập trực tiếp vào ô **tìm kiếm**.

Khi tìm được một tour phù hợp, khách hàng có thể **đặt tour** trực tuyến, bằng cách điền thông tin đầy đủ vào form đặt tour, gồm các thông tin họ tên người đặt tour, số người lớn và số trẻ em tham gia tour, chọn hình thức thanh toán, hệ thống kiểm tra cú pháp hoặc kiểu dữ liệu của các thông tin nhập, nếu sai thì hệ thống yêu cầu nhập lại, nếu đúng thì hệ thống phải hiển thị thông báo đặt tour thành công, và đơn đặt tour sẽ được lưu vào hệ thống. Một **khách hàng** có thể đặt nhiều tour tại nhiều thời điểm khác nhau, **một tour có thể từ 10 đến 40 khách**.

Sau khi đặt tour thành công, khách hàng có thể thanh toán ngay hoặc thanh toán sau 24g, nếu sau 24g không thanh toán thì đơn đặt tour sẽ bị hủy.

Sau khi đặt tour, nếu vì một lý do nào đó, khách hàng không thể tham gia tour, khách hàng có thể gửi **yêu cầu thay đổi** hoặc **hủy đặt tour**, bằng cách chọn chức năng yêu cầu hủy đặt tour, nếu yêu cầu hủy đặt tour trước ngày khởi hành 24g thì hệ thống chấp nhận và hoàn tiền 70% giá vé, nếu trước ngày khởi hành 12g thì được chấp nhận và hoàn 50% giá vé. Nếu trễ hơn thì không hoàn tiền.

Website **Lucky Tours** cũng hỗ trợ cho **nhân viên** của công ty thực hiện các công việc: tạo tour mới, cập nhật thông tin tour, quản lý đơn đặt tour của khách hàng, một nhân viên có thể quản lý nhiều đơn đặt tour, một đơn đặt tour chỉ thuộc một nhân viên quản lý, thống kê doanh thu, xử lý các yêu cầu thay đổi hoặc hủy đặt tour.

Hệ thống cũng giúp **người quản lý** có thể quản lý nhân viên, phân công hướng dẫn viên cho mỗi tour. **Một tour có thể từ 2 đến 3 hướng dẫn viên, một hướng dẫn viên có thể hướng dẫn nhiều tour.**

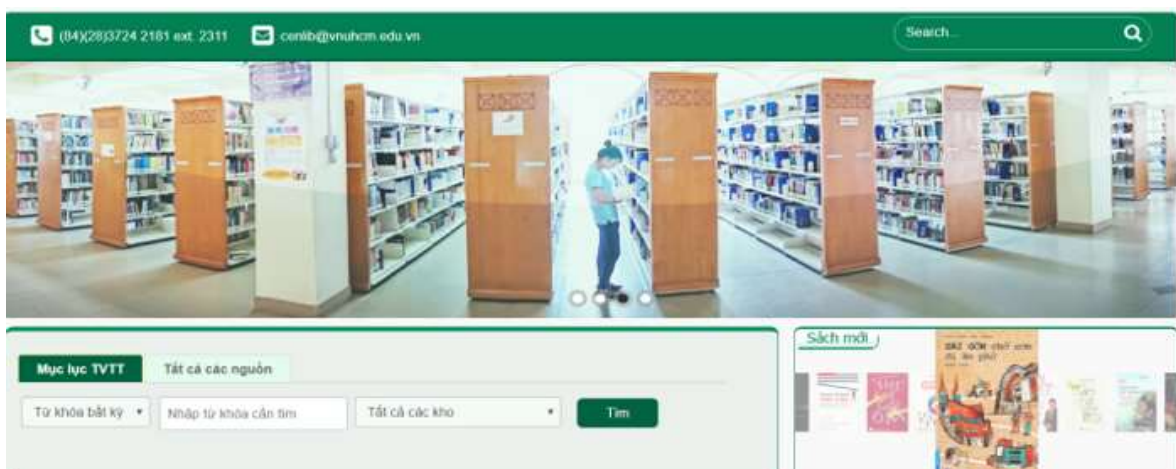
Người **quản trị hệ thống** có nhiệm vụ phân quyền người dùng, cập nhật chức năng của hệ thống.

Ngoài ra, người quản lý công ty du lịch cũng mong muốn website phải có giao diện đẹp, dễ sử dụng, bảo mật, chạy ổn định.

### **Yêu cầu:**

1. *Xác định yêu cầu chức năng*
2. *Mô hình hóa yêu cầu chức năng, sử dụng sơ đồ use case.*
3. *Yêu cầu phi chức năng*
4. *Quy tắc nghiệp vụ*
5. *Quy trình nghiệp vụ*

### **Case study 3. Xây dựng hệ thống quản lý thư viện của một trường đại học**



**Hệ thống quản lý thư viện** là một phần mềm giám sát và kiểm soát các hoạt động trong thư viện, hệ thống cung cấp cho người dùng thông tin đầy đủ về thư viện và các chức năng được thực hiện trong Thư viện. Hệ thống hoạt động trên mạng cục bộ của trường.

Sách trong thư viện được chia làm hai loại, sách in và sách điện tử

- **Đối với sách in:**

- Hệ thống lưu trữ thông tin của các cuốn sách theo từng loại sách (ví dụ: sách kinh tế, sách tin học, ...), với mỗi loại, hệ thống lưu thông tin Mã loại, tên loại, Mã giá sách. Mỗi loại sách gồm nhiều đầu sách, mỗi đầu sách được lưu trữ bởi các thông tin Mã đầu sách, tên đầu sách, tác giả, đơn giá, năm xuất bản, nhà xuất bản, số lượng bản in và thông tin tóm tắt nội dung.

- **Sách điện tử:**

- Hệ thống lưu dưới dạng tập tin, tên tập tin chính là Mã sách.

Hệ thống giúp độc giả có thể **tìm sách** theo nhiều tiêu chí khác nhau như tên loại sách, tên sách, tên tác giả, hoặc theo từ khóa, ...

Độc giả là các **giảng viên** hoặc **sinh viên** của trường, để mượn sách hoặc đọc sách, các độc giả phải đăng ký tài khoản trong hệ thống thư viện, Mã tài khoản chính là Mã giảng viên hoặc Mã sinh viên. Khi cần mượn sách in thì độc giả phải đăng nhập vào hệ thống và thực hiện chức năng **đăng ký mượn sách**, bao gồm các thông tin: Mã sách, tên sách, tác giả, ngày mượn, ngày trả, khi đăng ký thành công thì độc giả đến gặp thủ thư để nhận sách.

Đối với sách điện tử, độc giả cũng phải đăng nhập vào hệ thống để có thể tìm và đọc online trên mạng cục bộ của trường.

Hệ thống quản lý thư viện cung cấp chức năng giúp thủ thư có thể thêm sách mới, quản lý việc mượn và trả sách, thống kê sách đã cho mượn, số sách còn trong kho. Đối với sách điện tử, hệ thống có thể thống kê số lần truy cập, có thể đánh giá sách nào được yêu thích nhất.

Để duy trì hoạt động của thư viện, độc giả phải tuân thủ quy định về việc mượn và trả, nếu trễ hạn sẽ bị phạt, quy định phạt như sau: nếu số ngày trễ hạn <7 ngày thì phạt 5% đơn giá sách. Nếu từ 7 ngày đến 15 ngày thì phạt 10% giá sách, nếu trên 15 ngày thì phạt 20% giá sách.

Ngoài ra, người quản lý thư viện cũng yêu cầu hệ thống phải có giao diện thân thiện để sử dụng, chương trình chạy ổn định.

### **Yêu cầu:**

1. *Xác định yêu cầu chức năng*
2. *Mô hình hóa yêu cầu chức năng, sử dụng sơ đồ use case.*
3. *Yêu cầu phi chức năng*

- 
4. *Quy tắc nghiệp vụ*
  5. *Quy trình nghiệp vụ*

## Module 2. (6t) Viết đặc tả use case – mô hình hóa bằng Activity

### A) Mục tiêu:


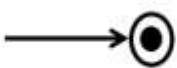
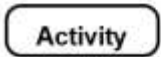

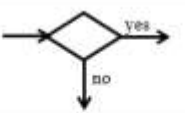
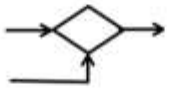
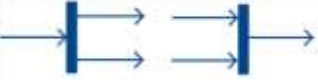
- Giúp sinh viên hiểu được vai trò của **đặc tả use case**
- Biết cách viết đặc tả use case
- Vận dụng sơ đồ **activity** để **mô hình hóa đặc tả use case**

### B) Tóm tắt kiến thức

#### 1. Nội dung của đặc tả use case:

- **Tên use case:** là tên của use case trong sơ đồ use case
- **Mô tả sơ lược:** Mô tả sơ lược chức năng của use case
- **Actor chính:** là actor kết hợp với use case trong sơ đồ use case
- **Actor phụ:** là actor hỗ trợ để thực hiện use case
- **Tiền điều kiện (Pre-condition):** điều kiện tiên quyết của use case, trạng thái của hệ thống phải đáp ứng để bắt đầu thực hiện use case. Điều kiện tiên quyết không được kiểm tra trong use case đang đặc tả.
- **Hậu điều kiện (Post-condition):** liệt kê các trạng thái của hệ thống có thể xảy ra sau khi use case thực thi hoàn tất. Hậu điều kiện được phân loại gồm: Đảm bảo tối thiểu hoặc Bảo đảm thành công
- **Luồng sự kiện chính (main flow):** các bước thực hiện use case, là sự tương tác giữa actor và hệ thống, từ lúc bắt đầu đến kết thúc.
  - o Mỗi bước được đánh số thứ tự.
  - o Actor thực hiện bước bắt đầu
  - o Hệ thống đáp trả.
  - o Actor thực hiện bước tiếp theo ...
- **Luồng sự kiện thay thế (alternate flow):** tại một bước trong luồng sự kiện chính, buộc actor phải chọn nhánh khác để thực hiện cho đến bước cuối hoặc quay lại một bước nào đó trong luồng sự kiện chính.
  - o Luồng sự kiện thay thế được đánh số thứ tự theo bước mà tại đó có luồng rẽ nhánh
- **Luồng sự kiện ngoại lệ (exception flow):** một cách xử lý cho một trường hợp ngoại lệ.

#### 2. Các ký hiệu trong sơ đồ activity

Tên thành phần	Ký hiệu trong UML	Ý nghĩa
Trạng thái bắt đầu (Initial State)		Điểm bắt đầu chuỗi hoạt động
Trạng thái kết thúc (Final State)		Điểm kết thúc chuỗi hoạt động
Hoạt động (Activity/Action State)		Một hoạt động trong chuỗi hoạt động
Luồng hoạt động (Action Flow)		Hướng chuyển từ hoạt động này sang hoạt động khác
Quyết định/Rẽ nhánh (Decisions/Branching)		Quyết định trước khi chuyển sang hoạt động tiếp theo.
Merge Node		Hợp nhất của các luồng, với nhiều đầu vào, và một đầu ra.
Đồng bộ hóa (Synchronization)		Fork: đại diện cho một luồng có thể phân nhánh Merge: nhiều luồng song song, có thể kết hợp thành một luồng.

### C) Bài tập

- Dựa vào **case study 1: xây dựng hệ thống đăng ký học phần trực tuyến**, viết đặc tả cho các use case của hệ thống: **đăng nhập, tạo tài khoản, xem điểm, thay đổi lớp học phần**, ứng với mỗi đặc tả, hãy mô hình hóa bằng sơ đồ activity.

Hướng dẫn:

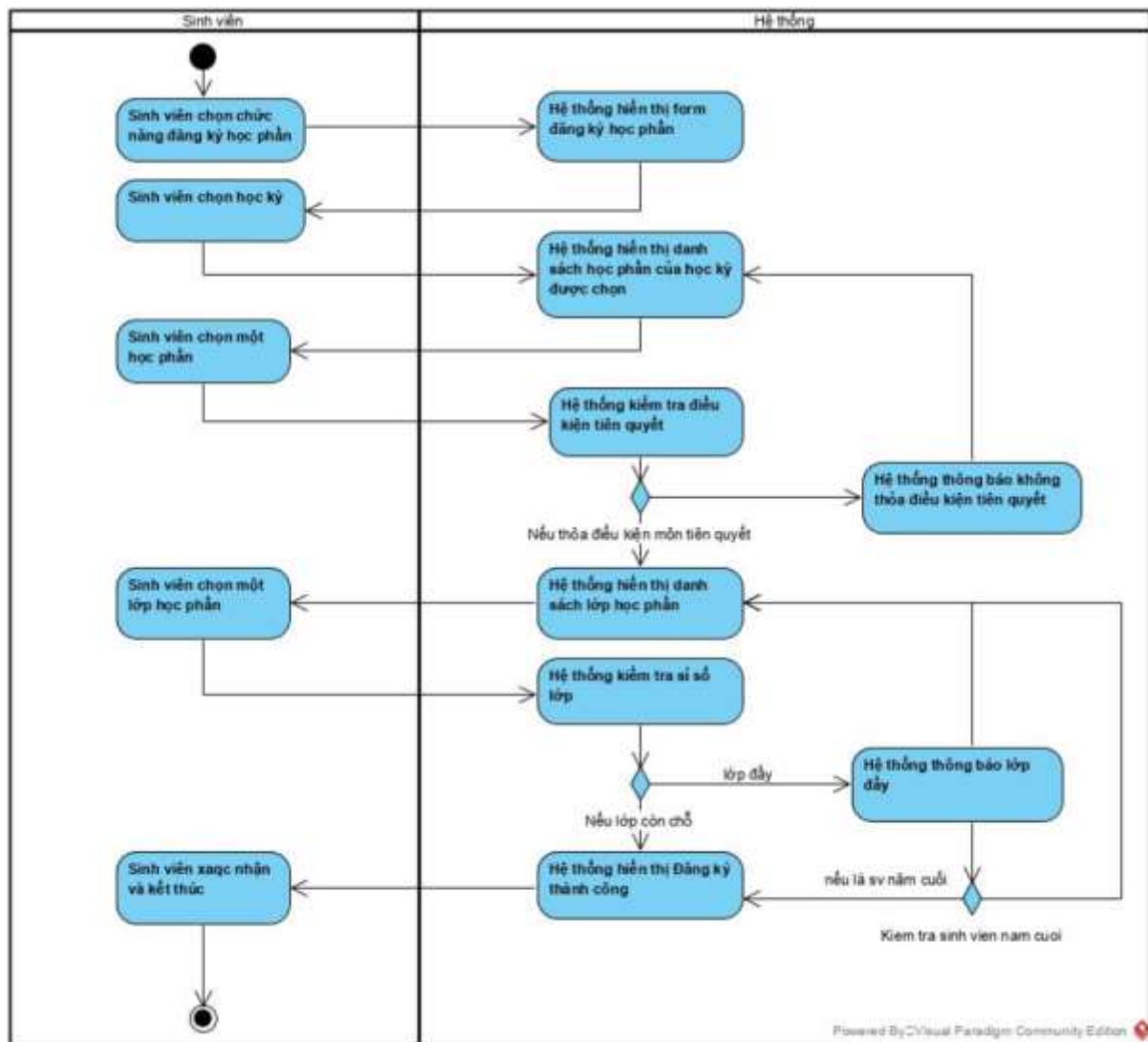
- Đặc tả use case “Đăng ký học phần”

– Tên use case: Đăng ký học phần
– Mô tả sơ lược: chức năng Đăng ký học phần giúp sinh viên có thể đăng ký học phần trực tuyến.
– Actor chính: Sinh viên
– Actor phụ: không
– Tiền điều kiện (Pre-condition): Đăng nhập thành công.
– Hậu điều kiện (Post-condition): nếu đăng ký thành công thì sĩ số lớp được cập nhật và sinh viên sẽ có lịch học.
– Luồng sự kiện chính (main flow):

Actor	System
1. Sinh viên chọn chức năng <b>Đăng ký học phần</b>	2. Hệ thống hiển thị trang đăng ký học phần
3. Sinh viên chọn học kỳ hiện tại	4. Hệ thống hiển thị danh sách học phần trong học kỳ đó
5. Sinh viên chọn học phần muốn đăng ký	6. Hệ thống kiểm tra học phần tiên quyết
	7. Hệ thống hiển thị danh sách lớp học phần
8. Sinh viên chọn 1 lớp học phần	9. Hệ thống kiểm tra sĩ số lớp
	10. Hệ thống thông báo đăng ký thành công.
<b>– Luồng sự kiện thay thế (alternate flow):</b>	
	6.1. Hệ thống hiển thị thông báo môn tiên quyết không thỏa
6.2. Sinh viên xác nhận	6.3. Hệ thống quay lại bước 4
	9.1. Hệ thống hiển thị thông báo lớp đầy
9.2. Sinh viên xác nhận	9.3. Hệ thống quay lại bước 8
<b>– Luồng sự kiện ngoại lệ (exception flow):</b>	
	9.1.1. Hệ thống kiểm tra nếu là sinh viên năm cuối. 9.1.2. Hệ thống hiển thị thông báo đăng ký thành công.
9.1.3. Sinh viên xác nhận và kết thúc	

- **Sơ đồ activity biểu diễn đặc tả use case “Đăng ký học phần”**





2. Dựa vào **case study 2: “Xây dựng website đặt tour du lịch trực tuyến”**, sinh viên hãy viết đặc tả use case:

- “Đặt tour trực tuyến” và vẽ sơ đồ activity biểu diễn đặc tả.
- “Tìm kiếm tour”, và vẽ sơ đồ activity biểu diễn đặc tả

3. Dựa vào **case study 3: “Xây dựng hệ thống quản lý thư viện của một trường đại học”**. sinh viên hãy viết đặc tả use case:

- “Đăng ký mượn sách” và vẽ sơ đồ activity biểu diễn đặc tả.
- “Quản lý việc mượn sách” của actor thủ thư, vẽ sơ đồ activity

## Module 3. (6t) Phân tích cấu trúc hệ thống - Mô hình hóa bằng Domain

### A) Mục tiêu:

- Xác định các lớp khái niệm dựa vào mô tả của hệ thống
- Xác định thuộc tính của các lớp đã xác định ở trên dựa vào mô tả của hệ thống
- Xác định mối quan hệ, lượng số, vai trò của mối quan hệ giữa các lớp
- Vẽ sơ đồ domain

### B) Tóm tắt kiến thức

1. **Lớp khái niệm** là những ý tưởng, sự vật hoặc đối tượng trong phạm vi hệ thống. Lớp khái niệm có thể là **các đối tượng doanh nghiệp** (Business objects), **các đối tượng trong thế giới thực** (Real world objects) hoặc **các sự kiện** (Events that transpire).

#### 2. Cách tìm lớp khái niệm:

##### a. Dựa vào từ điển dữ liệu

- **Xác định các danh từ hoặc cụm danh từ:**
  - Nếu **cụm danh từ lưu thông tin trạng thái hoặc nó có nhiều hành vi**, thì đó là một **lớp**.
  - Nếu **chỉ là một số hoặc một chuỗi**, thì đó có thể là một **thuộc tính**. Tuy nhiên, một số trường hợp, có thể trong ngữ cảnh này thì một khái niệm là **Lớp** nhưng trong ngữ cảnh khác thì nó là **thuộc tính**

##### b. Sử dụng danh mục lớp khái niệm

#### 3. Tìm mối quan hệ giữa các lớp khái niệm

Tất cả các đối tượng trong hệ thống đều liên kết nhau theo một cách trực tiếp hoặc gián tiếp, mạnh hoặc yếu, mối quan hệ giữa các đối tượng thể hiện các thông tin liên quan đến hành vi phụ thuộc lẫn nhau giữa các đối tượng. Quan hệ giữa các lớp gồm 4 loại: **Association, Aggregation, Composition, Reflexive**.

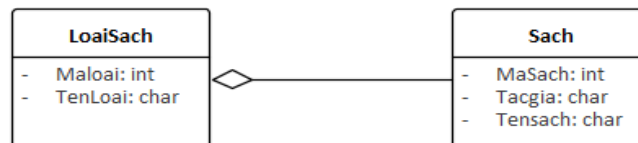
- a. **Association:** Một mối quan hệ được thiết lập khi hai lớp có liên quan với nhau theo bất kỳ cách nào, các đối tượng không phụ thuộc lẫn nhau

**Ví dụ:** **Lớp Sách quan hệ với lớp độc giả**, thể hiện hành vi **Độc giả đọc Sách**



**b. Aggregation: Lớp chứa (Whole)** được tạo ra bằng sự kết hợp của những **lớp thành phần khác (part)**, thời gian sống của lớp thành phần không phụ thuộc nhiều vào vòng đời của lớp chứa.

**Ví dụ:** loại sách được tạo từ nhiều cuốn sách. Nếu loại sách bị hủy thì sách vẫn tồn tại



**c. Composition:** Mỗi quan hệ composition tương tự với mỗi quan hệ aggregation. với sự khác biệt duy nhất là nhấn mạnh **sự phụ thuộc của lớp thành phần vào vòng đời của lớp chứa**.

**Ví dụ:** quan hệ giữa lớp **Sách** và **Chương mục**, nếu sách bị hủy thì chương mục cũng bị hủy

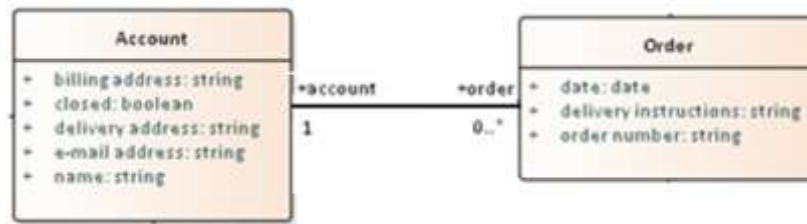


**d. Reflexive:** Khi một lớp có thể có nhiều chức năng hoặc trách nhiệm với chính lớp đó.

#### 4. Xác định vai trò (Roles) của mỗi quan hệ

**Vai trò (Role)** của mỗi quan hệ **thể hiện ý nghĩa của mỗi quan hệ** và số đối tượng của các lớp tham gia vào mỗi quan hệ. **Role** bao gồm các thành phần:

- ✓ **Lượng số tham gia (multiplicity):** là số đối tượng của một lớp tham gia vào mỗi quan hệ. Dựa vào các quy tắc nghiệp vụ và các ràng buộc để xác định lượng số
- ✓ **Tên của của quan hệ:** chỉ ý nghĩa của mỗi quan hệ
- ✓ **Điều hướng (navigability):** để chỉ lớp này có thể thấy lớp kia



## C) Bài tập

- Dựa vào **case study 1: “Xây dựng hệ thống đăng ký học phần trực tuyến”**, thực hiện các yêu cầu sau:
  - Xác định các lớp khái niệm, tìm các thuộc tính cho mỗi lớp.
  - Xác định loại mối quan hệ giữa các lớp, ứng với mỗi quan hệ, xác định lượng số, role của mỗi lớp, giải thích.
  - Vẽ sơ đồ Domain.
- Tương tự với yêu cầu của **câu 1**, thực hiện đối với **case study 2: “Xây dựng website đặt tour du lịch trực tuyến”** và **case study 3: Xây dựng hệ thống quản lý thư viện của một trường đại học”**.

### Hướng dẫn case study 1:

- Xác định các lớp khái niệm, tìm các thuộc tính cho mỗi lớp.**

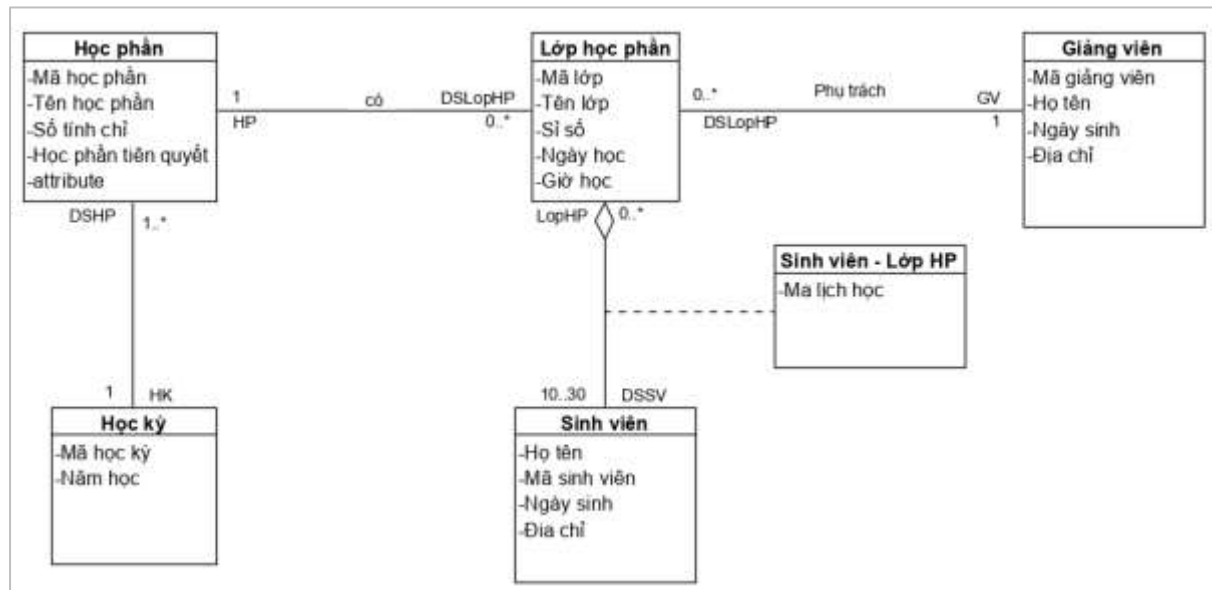
Lớp khái niệm	Thuộc tính
Học kỳ	<ul style="list-style-type: none"> <li>Mã học kỳ</li> <li>Năm học</li> </ul>
Học phần	<ul style="list-style-type: none"> <li>Mã học phần</li> <li>Tên học phần</li> <li>Học phần tiên quyết</li> <li>Số tín chỉ</li> </ul>
Lớp học phần	<ul style="list-style-type: none"> <li>Mã lớp</li> <li>Tên lớp</li> <li>Ngày học</li> <li>Giờ học</li> </ul>
Sinh viên	<ul style="list-style-type: none"> <li>Mã sinh viên</li> <li>Họ tên</li> <li>Ngày sinh</li> </ul>

	<ul style="list-style-type: none"> <li>• Địa chỉ.</li> </ul>
<b>Giảng viên</b>	<ul style="list-style-type: none"> <li>• Mã giảng viên</li> <li>• Họ tên</li> <li>• Ngày sinh</li> <li>• Địa chỉ.</li> </ul>

2. Xác định loại mối quan hệ giữa các lớp, ứng với mỗi quan hệ, xác định lượng số.

Quan hệ	Loại quan hệ	Giải thích
<b>Học kỳ - Học phần</b>	Association	Một học kỳ bao gồm nhiều học phần, mỗi học phần thuộc 1 học kỳ
<b>Học phần - lớp học phần</b>	Association	Một học phần gồm nhiều lớp học phần, một lớp học phần thuộc một học phần.
<b>Lớp học phần – Giảng viên</b>	Association	Một lớp học phần do một giảng viên phụ trách, một Giảng viên phụ trách nhiều lớp học phần
<b>Sinh viên – lớp học phần</b>	Aggregation	Một sinh viên có thể đăng ký nhiều lớp học phần, một lớp học phần có tối thiểu 10 sinh viên và tối đa 30 sinh viên. Mỗi Sinh viên là thành phần của lớp
<b>Lớp kết hợp Sinh viên – Lớp HP</b>		Lớp kết hợp giữa 2 lớp <b>Sinh viên – lớp học phần</b> có quan hệ * và *

### 3. Vẽ sơ đồ Domain







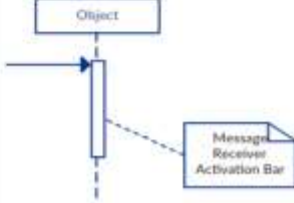
## Module 4. (6t) Thiết kế hệ thống - Sử dụng sơ đồ Sequence – Hiệu chỉnh sơ đồ Domain thành sơ đồ Class

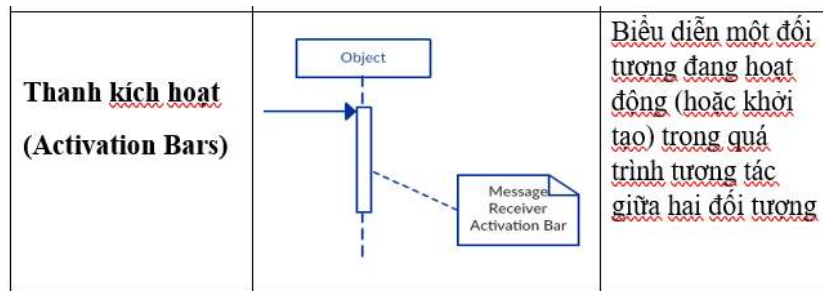
### A) Mục tiêu

- Xác định các đối tượng, các tương tác giữa các đối tượng trong hệ thống để thực hiện chức năng của use case
- Vẽ sơ Sequence mức thiết kế sơ bộ (chưa theo kiến trúc phần mềm cụ thể)
- Phân bổ các phương thức vào các lớp liên quan đến chức năng của use case tương ứng
- Hiệu chỉnh (bổ sung lớp mới, thay đổi mối quan hệ) lại sơ đồ Domain để chuyển sơ đồ Domain thành sơ đồ Class

### B) Tóm tắt Kiến thức




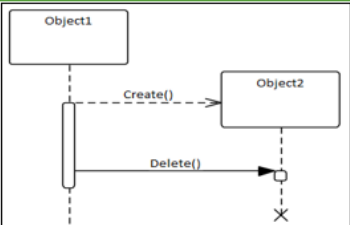
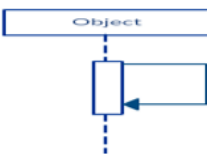
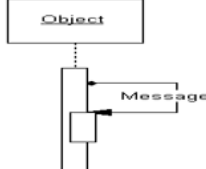
- **Sơ đồ sequence:** biểu diễn chi tiết quá trình tương tác của người dùng và hệ thống để thực hiện chức năng được biểu diễn bằng use case.
- **Các ký hiệu trong sơ đồ sequence và cách vẽ sơ đồ sequence.**

<b>lifeline actor</b>		Đại diện cho đối tượng thực hiện use case
<b>Lifeline Entity</b>		Đại diện cho dữ liệu trong hệ thống
<b>Lifeline boundary</b>		Biểu diễn ranh giới giữa người dùng và phần mềm
<b>Lifeline control</b>		Biểu diễn một đối tượng điều khiển, đóng vai trò trung gian tương tác giữa boundaries và entities
<b>Thanh kích hoạt (Activation Bars)</b>		Biểu diễn một đối tượng đang hoạt động (hoặc khởi tạo) trong quá trình tương tác giữa hai đối tượng



- **Thông điệp (Message):** Mũi tên từ **đối tượng gửi** đến **đối tượng nhận** biểu diễn thông điệp tương tác giữa các đối tượng, nội dung của thông điệp đặt trên mũi tên.

***attribute = message\_name (arguments): return\_type***

Tên loại thông điệp	Ký hiệu trong UML	Ý nghĩa
Thông điệp đồng bộ Synchronous message		đối tượng gửi chờ bên nhận xử lý và trả kết quả trước khi gửi thông điệp tiếp theo.
Thông điệp không đồng bộ (Asynchronous message)		Bên gửi không chờ bên nhận trả kết quả, tiếp tục gửi thông điệp tiếp theo
Return message		bên nhận đã xử lý xong thông điệp yêu cầu và trả lại quyền kiểm soát cho bên gửi.
Tạo/kết thúc đối tượng tham gia (creation/destroy message)		Đối tượng được tạo ngay khi thông điệp được gửi. Thông điệp thường là Create () hoặc New ()
Self message		Thông điệp của một đối tượng gửi cho chính nó
Recursive self message		cuộc gọi đệ quy đến cùng một đối tượng

- **Fragment:** biểu diễn một đoạn tương tác phức tạp giữa các đối tượng trong hệ thống như các luồng thay thế, cấu trúc lựa chọn, vòng lặp.





## C) Bài tập

### 1. Dựa vào **case study 1: Xây dựng hệ thống đăng ký học phần trực tuyến**

#### a. Phân tích Use case “Đăng ký học phần”:

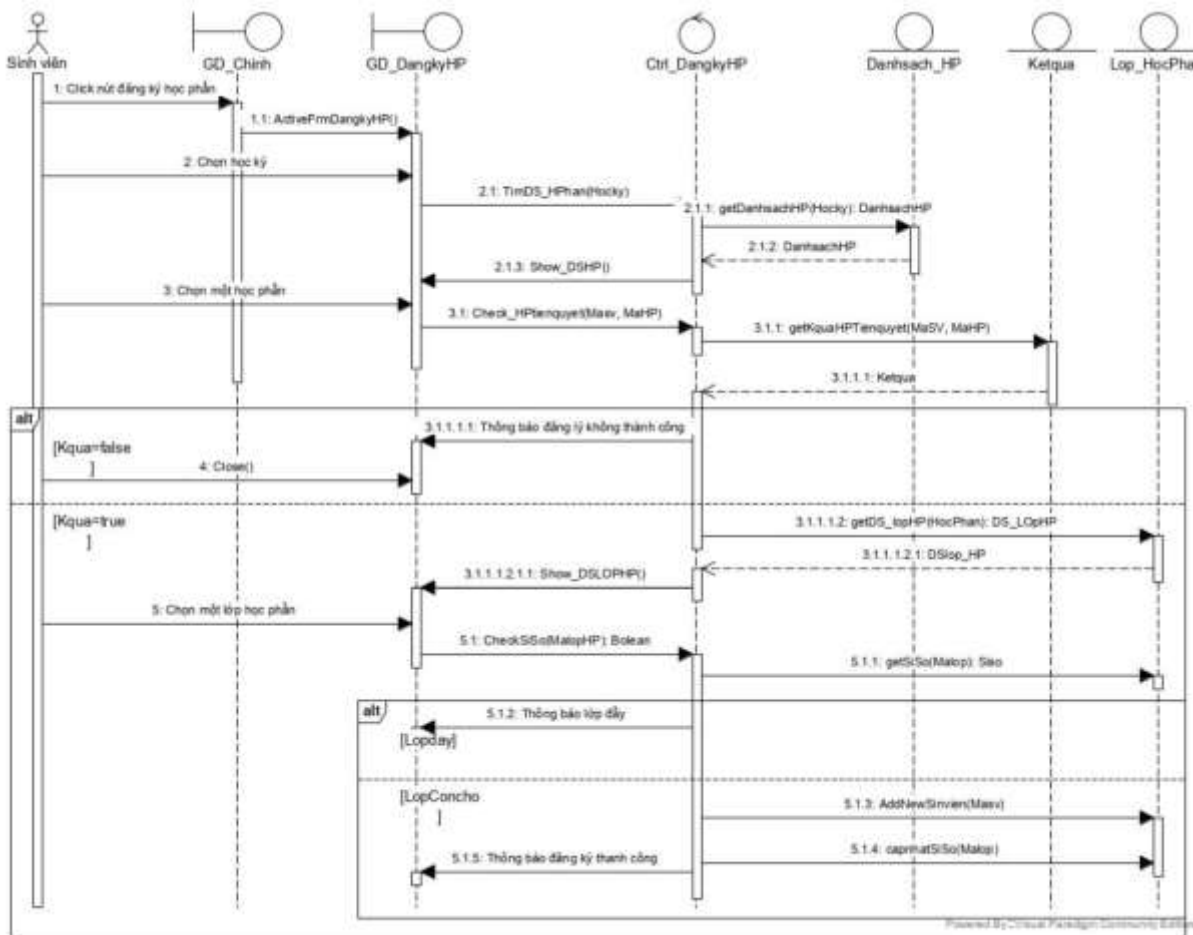
##### • Luồng sự kiện cơ bản

Use case này bắt đầu khi sinh viên đã đăng nhập vào hệ thống thành công. Sinh viên chọn một học kỳ, sau đó chọn học phần, hệ thống sẽ kiểm tra học phần tiên quyết, và hiển thị danh sách lớp học phần. Sinh viên chọn một lớp học phần, hệ thống kiểm tra sĩ số, nếu thỏa các điều kiện, sinh viên xác nhận hoạt động đã hoàn tất. Hệ thống sẽ thông báo đăng ký thành công và in lịch học của sinh viên. Hệ thống gửi thông tin thanh toán cho sinh viên đến hệ thống thanh toán để xử lý.

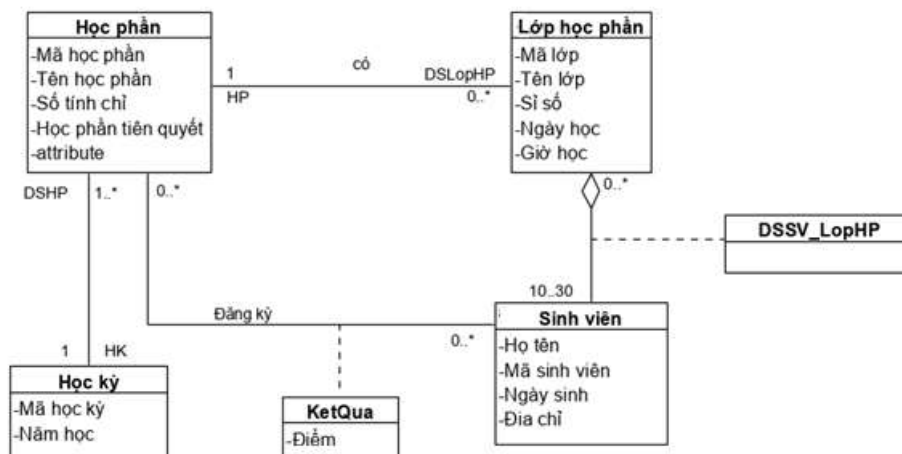
##### • Luồng sự kiện thay thế

- Nếu học phần tiên quyết không thỏa thì hệ thống sẽ hiển thị thông báo và kết thúc quá trình đăng ký.
- Nếu lớp học phần đã đủ, hệ thống yêu cầu chọn lớp khác.

#### b. Vẽ sơ đồ Sequence ở mức thiết kế sơ bộ cho Use case Đăng ký học phần.



c. Bổ sung các Class mới (nếu có) vào trong sơ đồ Domain liên quan đến chức năng **đăng ký học phần**.

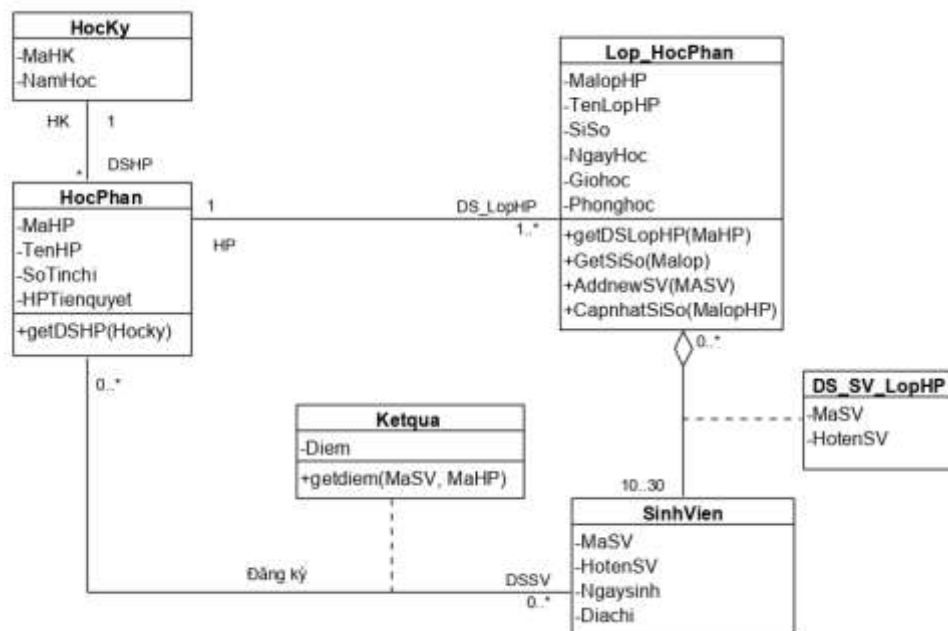


d. Phân bổ các phương thức vào các **Class** tương ứng trong sơ đồ Domain.

Để thực hiện chức năng “Đăng ký học phần” các lớp trong sơ đồ Domain cần phải có các phương thức sau:

- Lớp **Học phần** có các phương thức:

- **getDSHocPhan(HocKy):** trả về danh sách học phần theo học kỳ, với tham số là học kỳ được chọn.
- Lớp **Ketqua** có phương thức:
  - **getKetquaHPTienquyet(MaSV):** trả về kết quả của môn học tiên quyết của sinh viên đăng ký học phần, tham số là **MaSV, MaHP**
- lớp **Lop\_Hocphan** gồm các phương thức sau:
  - **getDS\_LopHP(MaHP):** trả về danh sách lớp học phần của học phần được chọn.
  - **getSiSo(MalopHP):** trả về sĩ số của lớp học phần được chọn.
  - **AddNewSV(MaSV):** thêm một sinh viên mới vào lớp học phần
  - **CapnhatSiSo(Malop):** Cập nhật sĩ số lớp

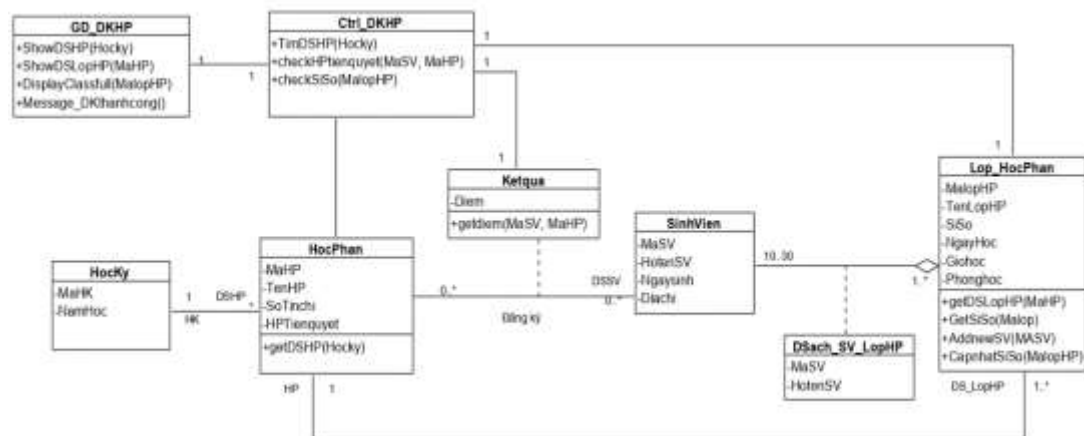


e. Hoàn thiện sơ đồ lớp tương ứng với chức năng “Đăng ký học phần”

Ngoài các lớp **entity**, để thực hiện chức năng “Đăng ký học phần” còn có các lớp **boundary** và **control**

- Lớp **GD\_DangkyHP** (lớp **Boundary**): gồm các phương thức
  - **showDSHP():** Hiển thị danh sách học phần trong học kỳ được chọn
  - **showDSLopHP():** Hiển thị danh sách lớp học phần của học phần được chọn
  - **Display\_Classfull():** Hiển thị thông báo lớp đầy

- **Message\_DKthanhcong()**: hiển thị thông báo đăng ký thành công
- Lớp **Ctrl\_DKHP (lớp Cotrol)** gồm các phương thức:
  - **TimDSHP(Hocky)**: tìm danh sách lớp học phần theo học kỳ được chọn
  - **checkHPtienquyet(MaSV, MaHP)**: Kiểm tra sinh viên có đáp ứng học phần tiên quyết của học phần được chọn
  - **checkSiSo(MaLopHP)**: kiểm tra sĩ số lớp còn cho phép đăng ký hay đã đầy.



2. Dựa vào case **study 1**, Sinh viên thực hiện phân tích chức năng “**xem điểm**” của actor sinh viên, bao gồm các bước tương tự câu 1.
3. Dựa vào case **study 2**, sinh viên phân tích chức năng “**Đặt tour du lịch**”.
4. Dựa vào case **study 3**, sinh viên hãy phân tích chức năng “**Đăng ký mượn sách**”

## Module 5. (3t) Thiết kế các thành phần của hệ thống

### A) Mục tiêu:

- Hiểu các khái niệm Subsystem và Package
- Cách xác định Subsystem và Package của hệ thống
- Vẽ sơ đồ Component

### B) Tóm tắt Kiến thức

**Package:** là một nhóm các **lớp** hoặc các **phần tử mô hình** có quan hệ logic với nhau. Package xác định một không gian tên cho nội dung của nó.

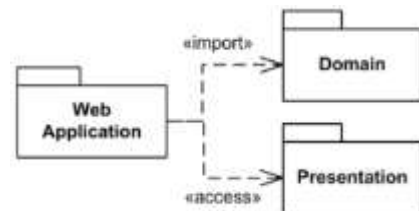


**Package diagram:** biểu diễn cấu trúc và tổ chức của hệ thống trong các dự án quy mô trung bình và lớn, package diagram được sử dụng trong các trường hợp:

- Tạo tổng quan về một tập hợp lớn các phần tử mô hình
- Nhóm các lớp và các phần tử mô hình có quan hệ logic.
- Tách không gian tên của các thành phần trong hệ thống.

**Quan hệ giữa các package** là quan hệ Dependency, gồm 2 loại:

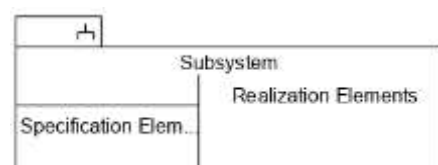
- **<<import>>:** nội dung public của *target package* được thêm vào không gian tên của *source package*
- **<<access>>:** nội dung public của *target package* có sẵn không gian tên của *source package*



**Tiêu chí để phân rã một hệ thống thành các package**

- Dựa vào Chủ sở hữu: Actor thực hiện chức năng trên sơ đồ
- Dựa vào chức năng: mỗi vấn đề có phân vùng rõ ràng
- Các lớp có mối liên kết mạnh được nhóm vào một package
- Dựa vào mẫu kiến trúc: ví dụ kiến trúc MVC
- Giảm sự phụ thuộc giữa các phần tử trong các package khác nhau.

**Subsystem:** Một hệ thống là một tập hợp các phần tử có tổ chức có thể được phân tách đệ quy thành các hệ thống con nhỏ hơn (subsystem)



Ví dụ:

- **Subsystem GUI** chịu trách nhiệm cung cấp giao diện người dùng giúp người dùng có thể tương tác với hệ thống.
- **Subsystem Business processing**: xử lý nghiệp vụ chịu trách nhiệm thực hiện chức năng nghiệp vụ
- **Subsystem Data**: thực hiện chức năng lưu trữ dữ liệu

### Các khía cạnh bên trong và bên ngoài Subsystem

- Một subsystem thể hiện 2 khía cạnh:
  - Bên ngoài: hiển thị các dịch vụ mà subsystem cung cấp
  - Bên trong: thể hiện việc hiện thực hóa của subsystem
  - Một ánh xạ giữa hai khía cạnh

## C) Bài tập

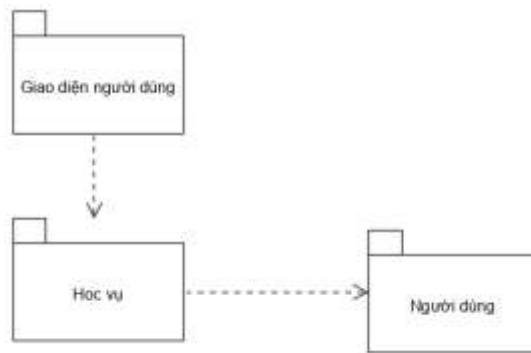
### 1. Dựa vào case study 1: xây dựng hệ thống đăng ký học phần trực tuyến

#### a. Tổ chức các Class trong hệ thống thành các Package, giải thích

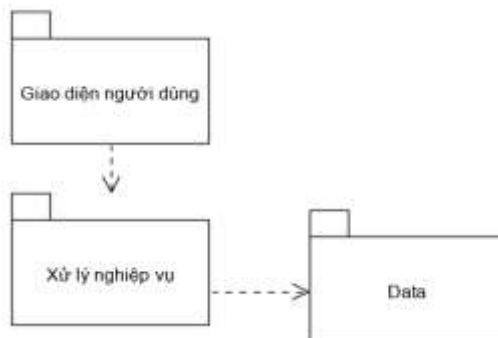
*Phân rã hệ thống theo tiêu chí nhóm các lớp có tính kết dính cao* vào thành một package, **Hệ thống đăng ký học phần** có thể được tổ chức thành các package sau:

- **Package Người dùng**: chứa các lớp liên quan đến người dùng hệ thống
  - Sinh viên
  - Giảng viên
  - Nhân viên
- **Package Học vụ**: chứa các lớp liên quan đến học phần trong các học kỳ
  - Học kỳ
  - Học phần
  - Lớp học phần
  - Lịch học
- **Package Giao diện người dùng**: gồm tất cả các giao diện của hệ thống
  - Form giao diện chính
  - Form giao diện đăng ký học phần
  - Form quản lý học phần
  - Form quản lý thông tin giảng viên
  - Form quản lý thông tin sinh viên

#### b. Vẽ sơ đồ Package của hệ thống



**Phân rã hệ thống theo kiến trúc thì Hệ thống đăng ký học phần** có thể được tổ chức thành các package sau:



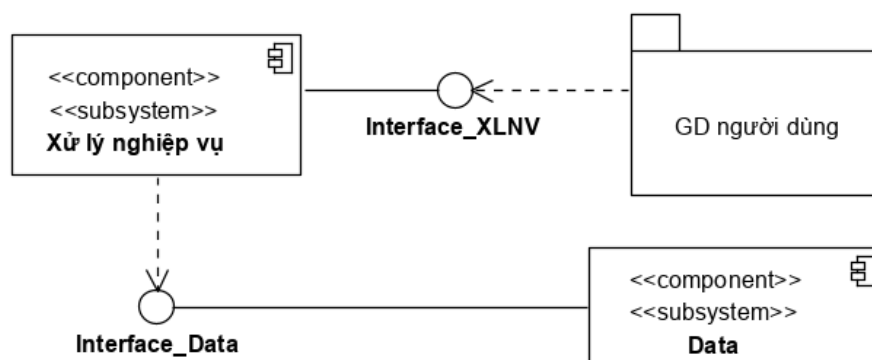
- **Package Giao diện người dùng:** gồm tất cả các giao diện của hệ thống
- **Package Xử lý nghiệp vụ:** chứa các lớp hiện thực các chức năng xử lý nghiệp vụ
- **Package Data:** chứa các lớp hiện thực việc lưu trữ dữ liệu

**c. Phân chia hệ thống đã xây dựng thành các Subsystem, giải thích**

Mỗi Subsystem cung cấp dịch vụ cho các Subsystem khác. **Hệ thống đăng ký học phần** có thể được tổ chức thành các Subsystem sau

- **Subsystem Xử lý nghiệp vụ:** hiện thực các chức năng xử lý nghiệp vụ
- **Subsystem Data:** chịu trách nhiệm thực hiện chức năng lưu trữ dữ liệu

**d. Vẽ sơ đồ subsystem của hệ thống. Giải thích**



- **Subsystem Xử lý nghiệp vụ** cung cấp interface được sử dụng bởi Package Giao diện người dùng.
  - **Subsystem Data:** Cung cấp interface được sử dụng bởi subsystem Xử lý nghiệp vụ.
2. **Dựa vào Case study 2: “Xây dựng website đặt tour du lịch”**, từ sơ đồ lớp được xây dựng trong module3, hãy xây dựng các Package và vẽ sơ đồ Package cho hệ thống.
  3. **Dựa vào Case study 2: “Xây dựng hệ thống quản lý thư viện”**, từ sơ đồ lớp được xây dựng trong module3, hãy xây dựng các Package và vẽ sơ đồ Package cho hệ thống.



## Module 6. (3t) Thiết kế hệ thống theo kiến trúc

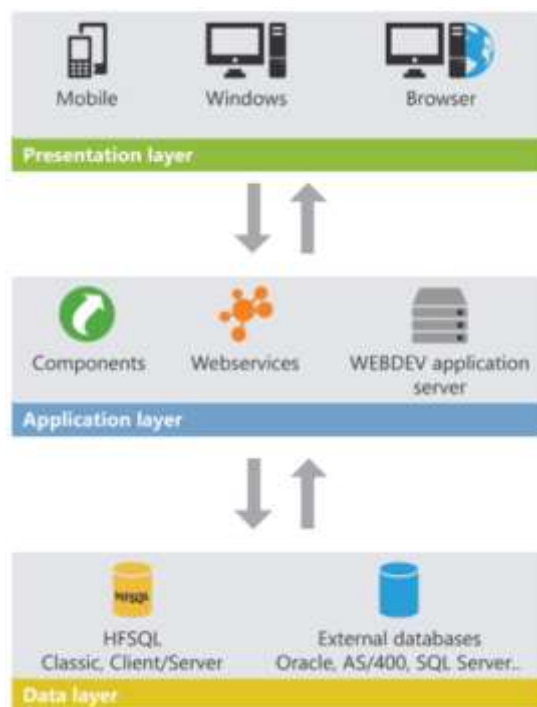
### A) Mục tiêu:

- Hiểu về kiến trúc phần mềm được chọn để áp dụng cho hệ thống (3 tầng)
- Xác định các thành phần của hệ thống ứng với kiến trúc đã chọn
- Vẽ lại các sơ đồ Sequence ở mức thiết kế cho từng Use case
- Thực hiện tinh chỉnh sơ đồ Class của hệ thống

### B) Tóm tắt Kiến trúc

Kiến trúc 3 tầng chia ứng dụng thành 3 độc lập:

- Presentation layer,
- Application (or process) layer,
- Data layer (which manages data access).



### C) Bài tập

Dựa vào **case study 1: xây dựng hệ thống đăng ký học phần trực tuyến**

Module 1. Đối với Use case “Đăng Ký học phần”, vẽ sơ đồ Sequence ở mức thiết kế chi tiết theo kiến trúc đã chọn.

---

Module 2. Sinh viên tiếp tục thực hiện tương tự như câu 1 cho các Use case

- Xem điểm
- .....

Module 3. Thực hiện tinh chỉnh sơ đồ Class của hệ thống

-

## Module 7. (3t)Thiết kế cơ sở dữ liệu

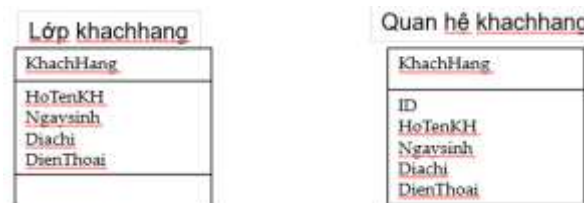
### A) Mục tiêu:

- Chuyển sơ đồ Class sang sơ đồ cơ sở dữ liệu quan hệ

### B) Tóm tắt Kiến thức

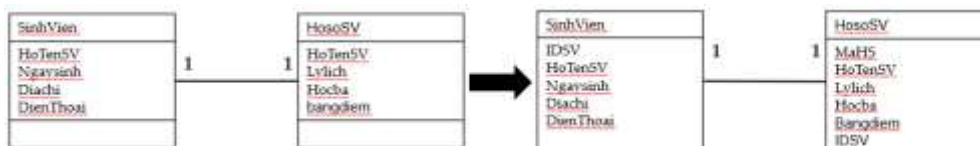
#### 1. Ánh xạ lớp thực thể - lược đồ quan hệ

- Một lớp thực thể ánh xạ thành một quan hệ cùng tên
- Một dòng trong bảng biểu diễn một đối tượng duy nhất trong lớp thực thể.
- Một thuộc tính của thực thể tương ứng với một thuộc tính trong quan hệ.
- Đưa thêm một thuộc tính ID kiểu int làm khóa chính cho quan hệ.

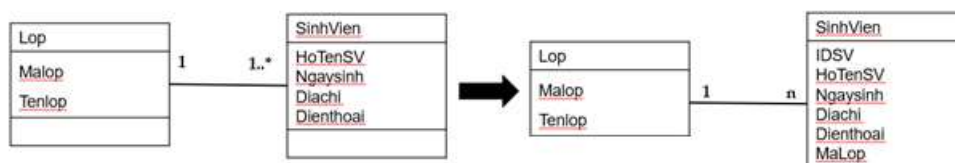


#### 2. Ánh xạ các liên kết 1-1

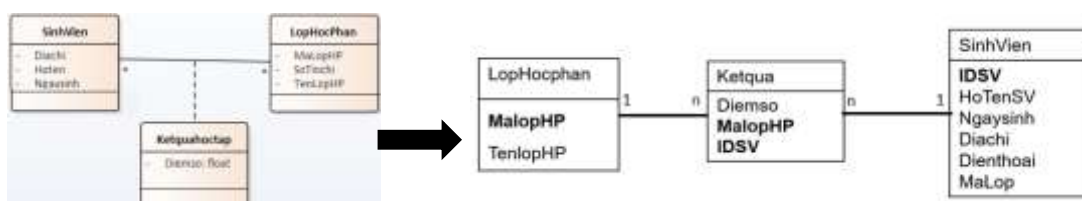
Thêm một khóa ngoại vào một trong các quan hệ. Quan hệ chứa khóa ngoại tham chiếu đến khóa chính của một quan hệ khác. Hoặc gộp 2 bảng thành 1 bảng



#### 3. Ánh xạ các kết 1-n: thêm một khóa ngoại vào bảng “nhiều”



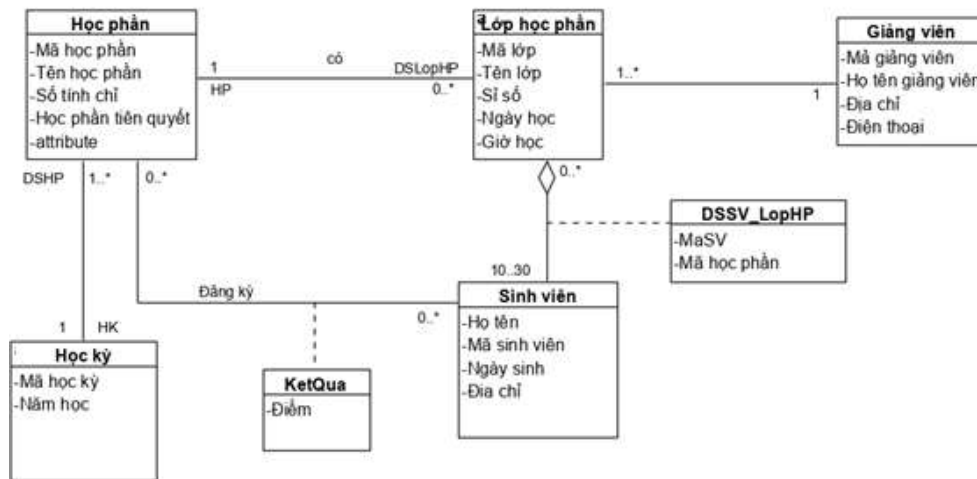
#### 4. Ánh xạ các liên kết n-m: Lớp kết hợp được chuyển thành quan hệ kết hợp, chứa 2 khóa chính của 2 quan hệ tham gia.



## C) Bài tập

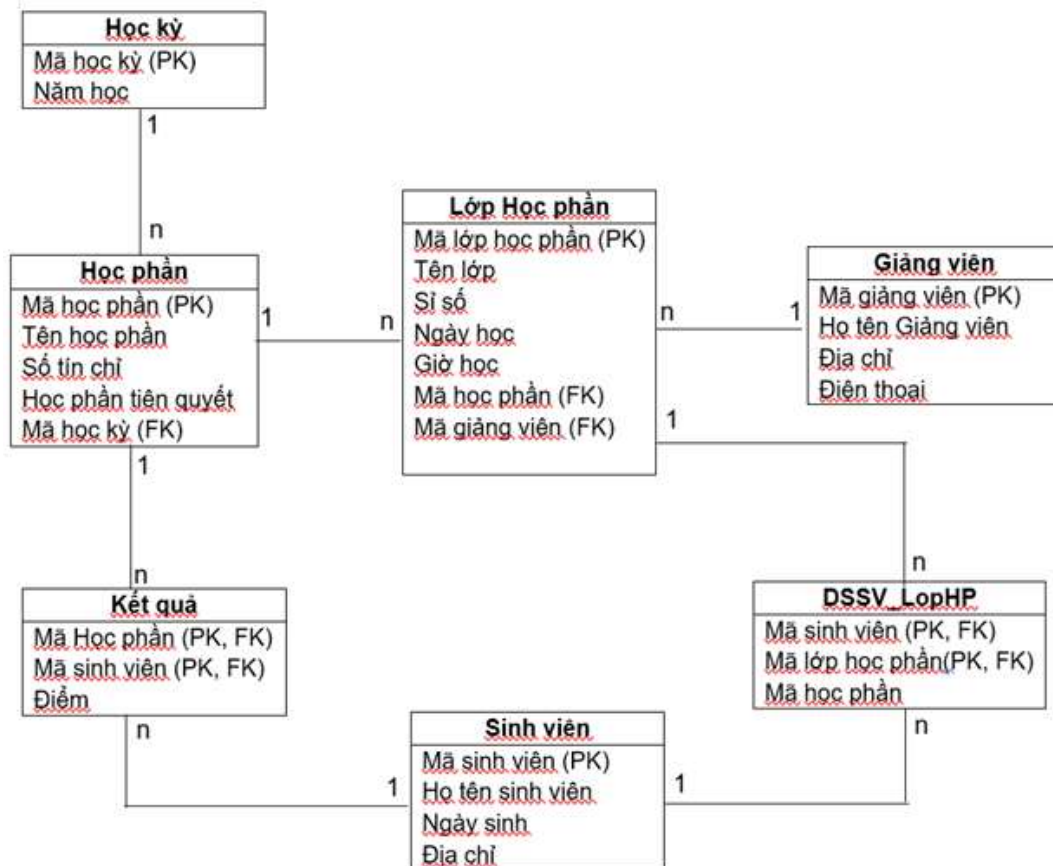
1. **Case study 1:** Dựa vào sơ đồ Class đã thiết kế ở module 5 chuyển sang sơ đồ cơ sở dữ liệu quan hệ.

**Sơ đồ domain**



- **Chuyển các lớp thành các quan hệ:**
  - Các lớp: **Học phần**, **Lớp học phần**, **Giảng viên**, **Sinh viên**, **Học kỳ** sẽ chuyển thành các quan hệ tương ứng cùng tên, thuộc tính của lớp trở thành thuộc tính của quan hệ, chọn một thuộc tính có giá trị duy nhất làm khóa chính.
  - Lớp kết hợp **Kết quả** trở thành **quan hệ Kết quả**, thuộc tính khóa chính của 2 quan hệ tham gia **Mã sinh viên**, **Mã học phần** trở thành khóa ngoại của **quan hệ Kết quả**, và 2 thuộc tính này đồng thời trở thành khóa chính của quan hệ Kết quả
  - Lớp **DSSV\_LopHP** trở thành quan hệ **DSSV\_LopHP**, với 2 thuộc tính khóa ngoại Mã sinh viên và Mã lớp học phần, 2 thuộc tính này trở thành khóa chính của quan hệ **DSSV\_LopHP**
- **Chuyển mỗi quan hệ:**
  - **Học phần - Học kỳ:** Quan hệ Học phần chứa khóa ngoại Mã học kỳ
  - **Học phần - Lớp học phần:** Lớp học phần chứa khóa ngoại Mã học phần
  - **Lớp học phần – Giảng viên:** Lớp học phần chứa khóa ngoại Mã Giảng viên.

Từ phân tích trên ta có **sơ đồ cơ sở dữ liệu quan hệ** của hệ thống Đăng ký học phần như sau:



2. **Case study 2:** Dựa vào sơ đồ Class đã thiết kế ở module 5 chuyển sang sơ đồ cơ sở dữ liệu quan hệ.
3. **Case study 3:** Dựa vào sơ đồ Class đã thiết kế ở module 5 chuyển sang sơ đồ cơ sở dữ liệu quan hệ.