

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



---

**QUẢN LÝ LỚP HỌC TÍN CHỈ**

---

***Nhóm 03 - Sinh viên thực hiện***

1. Nguyễn Xuân Long - 20075751
2. Nguyễn Thuý Tình - 20073231
3. Châu Bích Tuyền - 20067001

MỤC LỤC .....	2
CHƯƠNG 1 GIỚI THIỆU .....	4
1.1 Tổng quan .....	4
1.2 Mục tiêu đề tài .....	4
1.3 Phạm vi đề tài .....	5
1.3.1 Chức năng dành cho người dùng: .....	5
1.3.2 Chức năng dành cho người quản trị: .....	5
1.4 Mô tả yêu cầu chức năng .....	5
1.4.1 Phân tích yêu cầu của hệ thống .....	5
1.4.2 Mô tả hoạt động của hệ thống .....	6
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT .....	7
2.1 Node.js .....	7
2.2 React.js .....	8
2.3 TablePlus .....	9
2.4 Nestjs .....	10
2.5 Docker .....	12
2.6 Redis .....	13
2.7 PostgresSQL .....	14
2.8 Kiến trúc .....	15
CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ .....	18
3.1 Phân tích yêu cầu bằng UML .....	18
3.1.1 Usecase tổng quát .....	18
3.1.2 Danh sách tác nhân và mô tả .....	18
3.1.3 Danh sách các tình huống hoạt động (Use cases) .....	19
3.1.4 Tình huống hoạt động .....	20
3.2 Class diagram .....	33

3.3	ERD diagram .....	34
3.4	Kiến trúc layer .....	34
1.	Tính Modularity (Tính mô-đun): .....	34
2.	Tính Tương tác Thấp (Low Coupling): .....	35
3.	Tính Đa dạng (Abstraction): .....	35
4.	Tính Bảo mật (Security): .....	35
5.	Tính Tái sử dụng (Reusability): .....	36
6.	Tối ưu hóa hiệu năng và chi phí: .....	36
CHƯƠNG 4	: HIỆN THỰC .....	37
4.1	Giao diện của hệ thống .....	37
4.1.1	Giao diện đăng nhập .....	37
4.1.2	Giao diện Xem thông tin cá nhân .....	37
4.1.3	Giao diện đăng kí học phần .....	38
4.1.4	Giao diện xem lịch học .....	38
4.1.5	Giao diện xem chương trình khung .....	39
4.1.6	Giao diện xem kết quả học tập .....	39
CHƯƠNG 5	: KẾT LUẬN .....	40
5.1	Kết quả đạt được .....	40
5.2	Hạn chế của đồ án .....	40
5.3	Hướng phát triển .....	40
TÀI LIỆU THAM KHẢO	.....	41

## CHƯƠNG 1 : GIỚI THIỆU

### 1.1 Tổng quan

Hệ thống quản lý lớp học tín chỉ của trường đại học được thiết kế với mục tiêu tối ưu hóa trải nghiệm học tập cho sinh viên thông qua tính linh hoạt và tiện lợi. Nó không chỉ đơn giản là một công cụ để đăng ký môn học, mà còn là một nền tảng hỗ trợ sinh viên trong việc quản lý lịch trình học tập và theo dõi tiến trình học tập cá nhân. Hệ thống cung cấp các tính năng hỗ trợ cho việc chuẩn bị và xin cấp bằng tốt nghiệp, tạo điều kiện cho sinh viên có thể linh hoạt điều chỉnh lộ trình học tập theo ý muốn và nhu cầu cá nhân. Bằng cách này, sinh viên không chỉ trải qua một quá trình học tập thuận tiện mà còn được khuyến khích và hỗ trợ trong việc phát triển sự nghiệp sau này. Môi trường học tập đa dạng và phong phú được tạo ra, đảm bảo rằng sinh viên có mọi công cụ và nguồn lực cần thiết để thành công trong học tập và sự nghiệp của mình.

### 1.2 Mục tiêu đề tài

- Hệ thống quản lý lớp học tín chỉ được phát triển với mục tiêu rõ ràng là mang lại một trải nghiệm học tập linh hoạt và hiệu quả cho sinh viên. Không chỉ là một công cụ đơn giản để đăng ký các khóa học, hệ thống này còn cung cấp một loạt các tính năng và dịch vụ hỗ trợ để giúp sinh viên tự do quản lý và tối ưu hóa lộ trình học tập của mình.
- Tính linh hoạt của hệ thống cho phép sinh viên dễ dàng đăng ký các khóa học phù hợp với ngành học của mình và tự do điều chỉnh lịch trình học tập một cách thuận tiện. Nó không chỉ giúp sinh viên quản lý việc đăng ký học phần mà còn cung cấp thông tin cá nhân, thời khóa biểu, kết quả học tập và công nợ, tất cả được hiển thị một cách rõ ràng và dễ dàng truy cập qua nền tảng website của hệ thống.
- Sinh viên cũng có khả năng theo dõi tiến trình học tập của mình, xem chương trình khung và tận dụng các cơ hội thực tập và việc làm sau này thông qua hệ thống. Phần mở rộng của hệ thống có thể bao gồm phân tích dữ liệu học tập để cá nhân hóa trải nghiệm học tập, cung cấp thông tin và gợi ý phù hợp cho từng sinh viên.
- Những tính năng chính của hệ thống bao gồm:
  - Đăng ký học phần
  - Xem lịch học
  - Xem kết quả học tập

- Tra cứu công nợ
- Lịch học theo tiến độ
- Tiến trình học tập
- Xem Chương trình khung

### **1.3 Phạm vi đề tài**

#### **1.3.1 Chức năng dành cho người dùng:**

- Sinh viên có thể truy cập các tính năng sau trên hệ thống:
  - Xem thông tin cá nhân và tài khoản.
  - Đăng ký học phần theo nhu cầu học tập của mình.
  - Xem lịch học để tự định kỳ và sắp xếp thời gian học tập.
  - Xem kết quả học tập và theo dõi tiến trình học tập cá nhân.
  - Tra cứu công nợ và quản lý các khoản phí học phí.
  - Xem lịch học theo tiến độ và điều chỉnh lịch trình học tập theo nhu cầu.
  - Xem chương trình khung của ngành học.
- Hệ thống được thiết kế với giao diện trực quan và thân thiện, giúp người dùng dễ dàng tương tác và sử dụng một cách thuận tiện. Đồng thời, sinh viên cũng có khả năng thay đổi thông tin tài khoản và cá nhân theo nhu cầu và thay đổi của họ.

#### **1.3.2 Chức năng dành cho người quản trị:**

Ngoài các yêu cầu giống như của người dùng, thì hệ thống phải đảm bảo những yêu cầu sau:

- Thêm, xóa, chỉnh sửa học phần.
- Thêm, xóa, sửa lịch học
- Tổ chức đăng kí tín chỉ
- Thêm, xóa, sửa điểm số

### **1.4 Mô tả yêu cầu chức năng**

#### **1.4.1 Phân tích yêu cầu của hệ thống**

- Người dùng có thể thay đổi thông tin cá nhân như: tên, giới tính...
- Người dùng thực hiện đăng ký học phần trên web, hệ thống sẽ tiếp nhận và phản hồi cho đối tượng đăng ký học phần.

- Người dùng có thể hủy đăng ký học phần( nếu hệ thống cho phép).
- Từ mô tả trên, có thể đưa ra yêu cầu của hệ thống với hai đối tượng chính tương tác với hệ thống như sau:
- Đối với người dùng:
  - Đăng kí học phần
  - Xem lịch học
  - Xem điểm
  - Xem chương trình khung
  - Xem thông tin cá nhân.

#### **1.4.2 Mô tả hoạt động của hệ thống**

Quy trình quản lý học phần và biên chế sinh viên diễn ra như sau:

1. **Biên chế sinh viên theo ngành học:** Hệ thống tự động phân loại sinh viên vào các khoa và ngành học tương ứng dựa trên thông tin đăng ký ban đầu.
2. **Chọn môn học cho học kỳ:** Nhà trường lựa chọn danh sách môn học cần mở cho mỗi ngành học và nhập vào hệ thống.
3. **Đăng ký học phần:** Sinh viên đăng nhập vào hệ thống, xem danh sách các môn học có sẵn và đăng ký các môn phù hợp cho học kỳ.
4. **Xác nhận đăng ký:** Sinh viên xác nhận đăng ký trước khi hệ thống chấp nhận để tránh đăng ký nhầm lẫn hoặc hủy bỏ sau đó.
5. **Xử lý đăng ký:** Hệ thống xử lý đăng ký, kiểm tra số tín chỉ và quản lý số lượng sinh viên trong từng lớp học.
6. **Quản lý danh sách dự bị:** Sinh viên được đưa vào danh sách dự bị nếu các lớp học đã đầy, và hệ thống cân nhắc mở thêm lớp nếu cần.
7. **Gửi thông báo đăng ký:** Hệ thống gửi email thông báo xác nhận đăng ký và yêu cầu đóng học phí cho sinh viên sau khi đăng ký thành công.
8. **Xem lịch học:** Sinh viên có thể xem lịch học của mình cho từng học kỳ, bao gồm thời gian, địa điểm và thông tin liên quan đến các lớp học
9. **Xem thông tin học tập:** Sinh viên truy cập hệ thống để xem thông tin cá nhân như số tín chỉ đã đạt, số môn đã học, điểm số và điểm trung bình tích lũy.

## **2.1 Node.js**



*Hình 2.1: Node.js development*

Node.js được xây dựng trên nền tảng JavaScript cụ thể là V8 Google Engine của Chrome để xây dựng các ứng dụng mạng nhanh và có khả năng mở rộng.

Mô hình chạy JavaScript này sử dụng một mô hình I/O không chặn và dựa vào sự kiện, giúp nó trở nên hiệu quả và hoàn hảo cho các ứng dụng thời gian thực tốn nhiều dữ liệu chạy trên các thiết bị phân tán.

Hơn nữa, các ứng dụng Node.js được viết bằng JavaScript và có thể chạy trong môi trường Node.js trên Mac OS X, Microsoft Windows và Linux mà không cần thay đổi gì.

### ***Ưu điểm:***

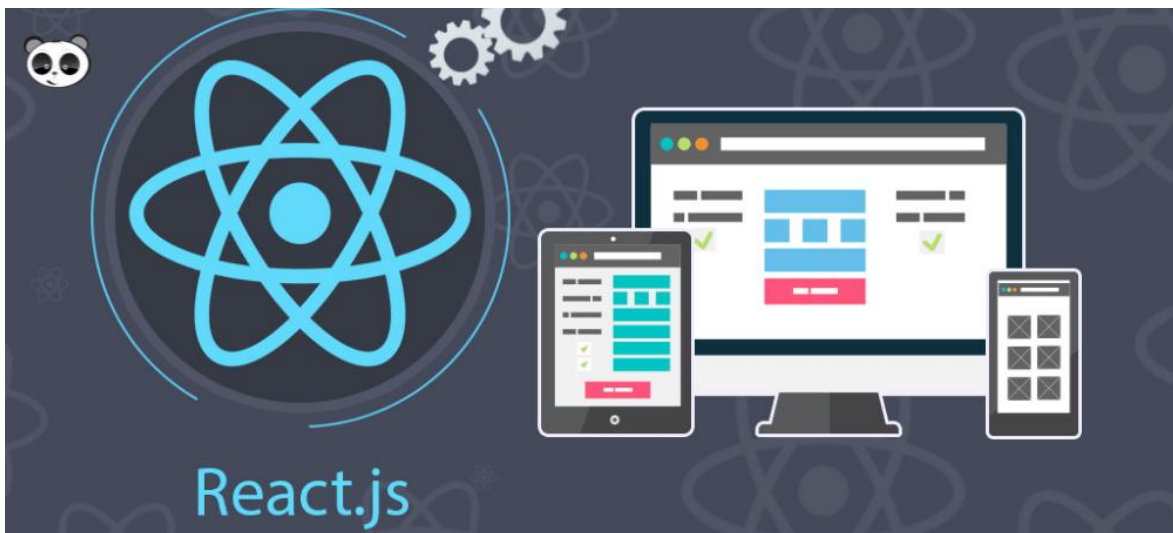
- Cung cấp khả năng mở rộng dễ dàng
- Dễ học
- Phổ biến với hiệu suất cao
- Sự hỗ trợ từ cộng đồng lớn và hoạt động mạnh mẽ
- Ưu điểm của việc caching
- Tiết kiệm chi phí với JavaScript toàn diện
- Ưu tiên cho việc phát triển ứng dụng đa nền tảng
- Sử dụng một ngôn ngữ lập trình duy nhất

- Cộng đồng lớn hỗ trợ (yarn, npm, pnpm) để cài đặt các gói khác nhau có sẵn trên internet trên các máy chủ Node.js một cách dễ dàng và nhanh chóng.

**Nhược điểm:**

- API không ổn định
- Thiếu hệ thống hỗ trợ thư viện mạnh mẽ
- Mô hình lập trình không đồng bộ
- Ít khả năng xử lý các tác vụ liên quan đến CPU.

## 2.2 React.js



*Hình 2.2: React.js development*

ReactJS, một thư viện JavaScript dựa trên các thành phần, được tạo ra ban đầu bởi Facebook và được phát hành chính thức cách đây đúng mười năm. Thư viện này đơn giản hóa việc tạo ra giao diện người dùng tương tác cho các nhà phát triển, đồng thời quản lý trạng thái của các thành phần một cách hiệu quả. Khả năng của nó để kết hợp nhiều thành phần cho các ứng dụng phức tạp mà không mất trạng thái của chúng trong DOM (Mô hình Đối tượng Tài liệu) của trình duyệt là một ưu điểm đáng kể đối với nhiều nhà phát triển.

**Ưu điểm:**

- Sử dụng kiến trúc dựa trên thành phần, giúp tạo ra các thành phần tái sử dụng và dễ bảo trì.



- Linh hoạt trong việc xây dựng các ứng dụng tùy chỉnh.
- Sẵn sàng thích ứng với các xu hướng công nghệ mới.
- Reactjs giúp việc viết các đoạn code js dễ dàng hơn với cú pháp đặc biệt là jsx cho phép tích hợp code html và js trong cùng 1 file

**Nhược điểm:**

- Khái niệm và mẫu phức tạp có thể gây khó khăn cho người mới.
- Phức tạp khi tích hợp với các công nghệ khác.
- Không phải là một framework hoàn chỉnh.
- Kích thước mã lớn và hiệu suất giảm sút trên các thiết bị cũ và mạng yếu.

## **2.3 TablePlus**



*Hình 2.3: TablePlus development*

**Tổng quan về TablePlus:**

TablePlus là một ứng dụng quản lý cơ sở dữ liệu mạnh mẽ và đa nền tảng, cung cấp một giao diện đồ họa thân thiện và nhiều tính năng tiện ích giúp người dùng tương tác với các cơ sở dữ liệu một cách dễ dàng và hiệu quả.

**Ưu điểm của TablePlus:**

1. **Hỗ trợ nhiều loại cơ sở dữ liệu:** TablePlus hỗ trợ nhiều loại cơ sở dữ liệu phổ biến như MySQL, PostgreSQL, SQLite, SQL Server, và nhiều hơn nữa. Điều này giúp cho người dùng có thể làm việc với nhiều loại cơ sở dữ liệu mà không cần phải sử dụng nhiều ứng dụng khác nhau.

2. **Giao diện người dùng thân thiện:** TablePlus cung cấp một giao diện người dùng sáng sủa, trực quan và dễ sử dụng. Các tính năng được tổ chức một cách logic và dễ dàng tiếp cận, giúp cho người dùng dễ dàng thao tác với cơ sở dữ liệu mà không cần phải mất nhiều thời gian tìm hiểu.
3. **Tính linh hoạt và tùy chỉnh:** TablePlus cho phép người dùng tùy chỉnh và điều chỉnh giao diện và tính năng theo nhu cầu của mình. Người dùng có thể thay đổi giao diện, cấu hình kết nối cơ sở dữ liệu, và tinh chỉnh các cài đặt khác để phù hợp với quy trình làm việc của họ.
4. **Tích hợp nhiều công cụ hữu ích:** TablePlus tích hợp nhiều công cụ hữu ích như trình quản lý dự án, trình biên tập SQL, và trình xem dữ liệu. Điều này giúp tối ưu hóa quy trình làm việc và nâng cao hiệu suất làm việc của người dùng.
5. **Đa nền tảng:** TablePlus hỗ trợ đa nền tảng, bao gồm macOS, Windows và Linux. Điều này giúp cho người dùng có thể trải nghiệm ứng dụng trên nhiều hệ điều hành khác nhau mà không cần phải lo lắng về sự tương thích.

Với những ưu điểm trên, TablePlus đã trở thành một trong những công cụ quản lý cơ sở dữ liệu phổ biến và được ưa chuộng trong cộng đồng phát triển phần mềm và quản trị cơ sở dữ liệu.

## 2.4 Nestjs



Hình 2.4: NestJs development

Nest (NestJS) là một framework được thiết kế để xây dựng các ứng dụng máy chủ Node.js hiệu quả và có khả năng mở rộng. Nó sử dụng JavaScript tiến triển, được xây dựng và hoàn toàn hỗ trợ TypeScript (nhưng vẫn cho phép các nhà phát triển viết mã

*Báo cáo đồ án môn học – Môn Kiến trúc phần mềm*  
trong JavaScript thuần) và kết hợp các yếu tố của OOP (Lập trình hướng đối tượng), FP (Lập trình hàm) và FRP (Lập trình hàm phản ứng).

Nest cung cấp một mức độ trừu tượng hóa trên các framework Node.js thông thường này (Express/Fastify), nhưng cũng tiếp tục tiếp cận các API của chúng trực tiếp cho nhà phát triển. Điều này mang lại sự tự do cho nhà phát triển để sử dụng các module của bên thứ ba mà có sẵn cho nền tảng cơ bản.

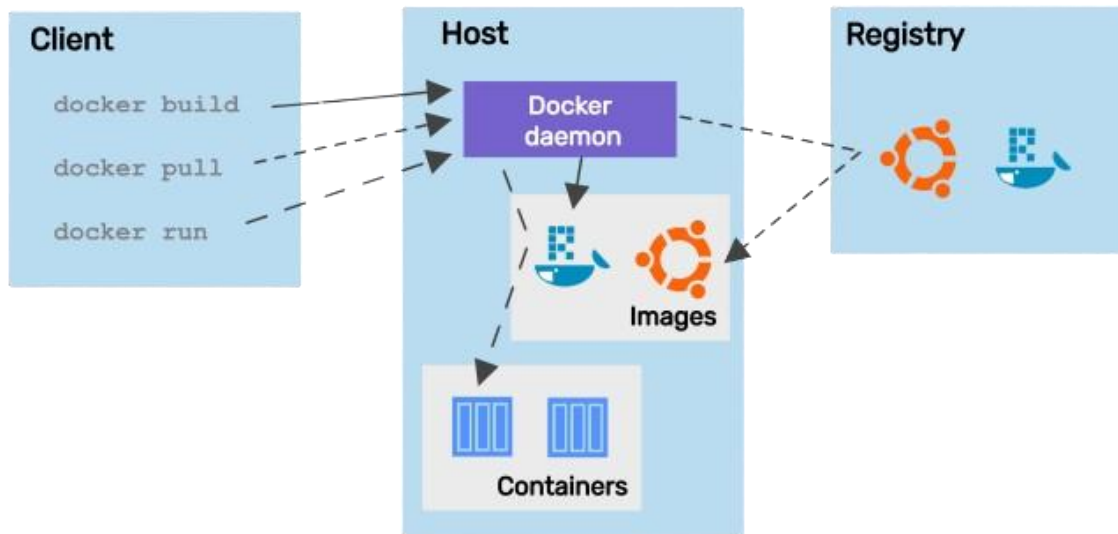
### **Ưu điểm:**

- Khả năng mở rộng và bảo trì: Kiến trúc module giúp dễ dàng tổ chức và bảo trì mã, cho phép mở rộng ứng dụng một cách dễ dàng.
- Hỗ trợ Microservices: Hỗ trợ tự nhiên cho kiến trúc Microservices, gói tiện ích `@nestjs/microservices` hữu ích cho giao tiếp dịch vụ.
- TypeScript là trung tâm: TypeScript giúp mã nguồn mạnh mẽ và phát hiện lỗi sớm.
- Hỗ trợ GraphQL: Module `@nestjs/graphql` giúp tích hợp GraphQL dễ dàng.
- Middleware và Guards: Cho phép can thiệp vào các yêu cầu và áp dụng logic xử lý, hỗ trợ xác thực và phân quyền.
- Cộng đồng năng động và hệ sinh thái phong phú

### **Nhược điểm:**

- Khó học đối với những người mới
- Phức tạp cho dự án nhỏ: Cấu trúc modular có thể quá phức tạp cho các ứng dụng đơn giản, tốn thời gian thiết lập và bảo trì.
- Gánh nặng hiệu suất: Có thể nặng hơn các framework khác, gây lo ngại về hiệu suất cho các ứng dụng yêu cầu tài nguyên thấp.
- Hạn chế trong dự án cổ điển: Không tương thích hoàn toàn với JavaScript hoặc các framework cũ, chuyển đổi sang NestJS có thể tốn nhiều công sức.

## 2.5 Docker



Hình 2.5: Docker development

**Docker** là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa) cho phép bạn dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng.

### Ưu điểm:

- docker hỗ trợ thiết lập nhanh chóng với nhiều lựa chọn điển hình là docker compose
- docker setup có thể quick check vô cùng nhanh. Ví dụ bạn có thể cài đặt redis trên local với thời gian dưới 10s và có thể xóa hoặc tắt nó đi đảm bảo không chạy ngầm
- Hỗ trợ mạng ảo để giao tiếp internal giữa các service
- Hỗ trợ đa kiến trúc, AMD, ARM, x86,...

### Nhược điểm:

- Khó tiếp cận đối với những người lập trình
- Quá nhiều sự lựa chọn triển khai cho cùng một mục đích
- Phải cần am hiểu nhiều về networking, development application, programming language cho việc config đóng gói phần mềm

## 2.6 Redis



*Hình 2.6: Redis development*

Redis là một cơ sở dữ liệu mã nguồn mở và bộ nhớ cache được sử dụng để lưu trữ dữ liệu dạng key-value trong bộ nhớ trong. Nó cũng hoạt động như một trình thông điệp và hỗ trợ các tính năng như sao chép, kịch bản Lua, việc xóa các phần tử ít được truy cập nhất, giao dịch và bền vững trên đĩa. Redis hỗ trợ nhiều loại cấu trúc dữ liệu trừu tượng và cung cấp tính sẵn có cao thông qua Redis Sentinel và Redis Cluster. Nó có thể chạy trên nhiều nền tảng, bao gồm Linux, Windows và macOS.

Ưu điểm:

- Hiệu suất cao: Redis được thiết kế để hoạt động với bộ nhớ trong, giúp cung cấp thời gian đáp ứng nhanh chóng và xử lý dữ liệu nhanh chóng.
- Hỗ trợ nhiều cấu trúc dữ liệu: Redis hỗ trợ nhiều loại cấu trúc dữ liệu trừu tượng như chuỗi, danh sách, bản đồ, tập hợp, và nhiều hơn nữa, giúp phù hợp với nhiều loại ứng dụng khác nhau.
- Hỗ trợ sao chép và bền vững: Redis cung cấp tính năng sao chép dữ liệu và khả năng bền vững trên đĩa, giúp đảm bảo an toàn và sẵn sàng của dữ liệu.
- Linh hoạt và đa nền tảng: Redis có thể chạy trên nhiều nền tảng khác nhau như Linux, Windows và macOS, và có thể tích hợp dễ dàng vào các ứng dụng hiện có.
- Cộng đồng phát triển lớn: Redis có một cộng đồng phát triển mạnh mẽ, cung cấp hỗ trợ và tài liệu phong phú cho người dùng.

Nhược điểm:

- Hạn chế bộ nhớ: Do Redis lưu trữ dữ liệu trong bộ nhớ, nên có thể gặp phải hạn chế về dung lượng bộ nhớ khi xử lý các tập dữ liệu lớn.

- Không phù hợp cho các tác vụ CPU-bound: Redis chủ yếu được thiết kế cho các tác vụ I/O-bound và không phù hợp cho các tác vụ yêu cầu xử lý nhiều CPU-bound.
- Cần quản lý tài nguyên: Vì Redis lưu trữ dữ liệu trong bộ nhớ, nên cần phải quản lý tài nguyên để đảm bảo hiệu suất và sự ổn định của hệ thống.

## 2.7 PostgreSQL



*Hình 2.7: PostgreSQL*

PostgreSQL là một trong những hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng có mục đích chung tiên tiến nhất và là nguồn mở. Là một phần mềm nguồn mở, mã nguồn của nó có sẵn theo giấy phép PostgreSQL, giấy phép nguồn mở tự do. Bất kỳ ai có kỹ năng phù hợp đều có thể tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức. Vì nó có tính ổn định cao nên cần rất ít nỗ lực để duy trì DBMS này.

Ưu điểm:

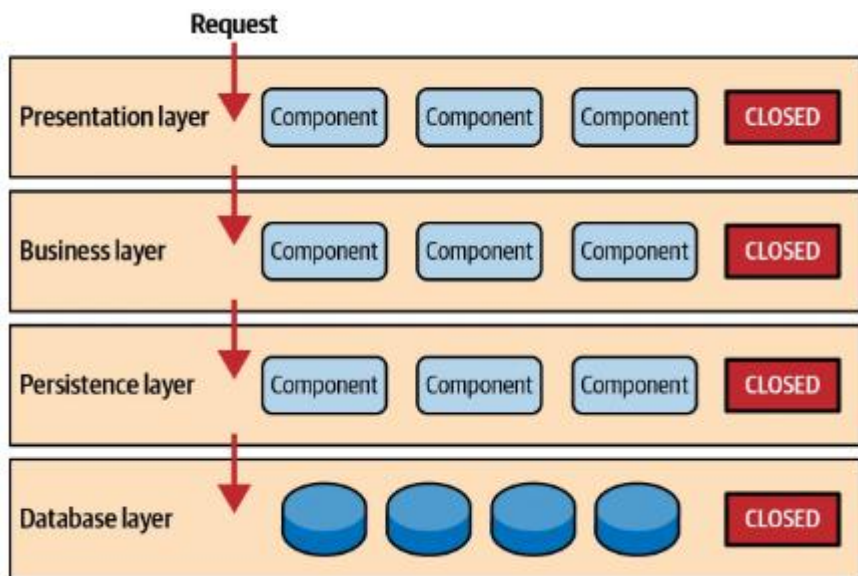
- ACID transactions (Giao dịch ACID): PostgreSQL cung cấp các giao dịch ACID (Nguyên tử, Consistent, Isolated, Durable) đảm bảo tính toàn vẹn của dữ liệu.
- Khả năng mở rộng (Extensibility): PostgreSQL cho phép người dùng mở rộng chức năng của nó bằng cách tạo các hàm, kiểu dữ liệu, ngôn ngữ mới...
- tính năng bảo mật (Security feature): PostgreSQL cung cấp các tính năng bảo mật giúp bảo vệ dữ liệu của bạn.

- Mã nguồn mở (Open source): PostgreSQL là mã nguồn mở và miễn phí.

Nhược điểm:

- Cấu trúc cơ sở dữ liệu (Database structure): PostgreSQL là một cơ sở dữ liệu quan hệ, vì vậy nó có lược đồ (schema) cố định. Điều này có thể gây khó khăn khi bạn cần lưu trữ dữ liệu không khớp với lược đồ.
- Là mã nguồn mở (Open source): Là mã nguồn mở nên PostgreSQL không có sự bảo hành từ một hãng sản xuất nào. Ngoài ra, việc cộng đồng quản lý có thể dẫn đến việc thiếu các giao diện thân thiện với người dùng hoặc các tính năng quen thuộc.

## 2.8 Kiến trúc sử dụng - Layer Architecture



Hình 2.8: Hệ thống sử dụng kiến trúc phân lớp

Mỗi layer đóng vai trò như một lớp trừu tượng, che giấu chi tiết triển khai bên dưới và chỉ cung cấp một giao diện hoặc giao thức cho các layer phía trên. Các layer thường được sắp xếp theo một thứ tự nhất định, từ layer thấp nhất (hay gần giao diện vật lý) đến layer cao nhất (gần với người dùng cuối).

Ví dụ, trong một ứng dụng web, bạn có thể có các layer sau:

1. *Layer Dữ liệu (Data Layer)*: Layer này chịu trách nhiệm lưu trữ và truy cập dữ liệu. Nó có thể bao gồm cơ sở dữ liệu và các dịch vụ liên quan như ORM (Object-Relational Mapping).

2. **Layer Logic Kinh doanh (Business Logic Layer):** Layer này chứa các quy tắc kinh doanh và logic xử lý chính của ứng dụng. Nó có thể xử lý các yêu cầu từ layer giao diện và tương tác với layer dữ liệu để truy xuất hoặc cập nhật dữ liệu.
3. **Layer Giao diện (Presentation Layer):** Layer này chịu trách nhiệm hiển thị giao diện người dùng và tương tác với người dùng cuối. Nó có thể bao gồm các thành phần như HTML/CSS/JavaScript cho ứng dụng web, hoặc các giao diện người dùng khác như ứng dụng di động.

Mỗi layer có thể tương tác với các layer khác thông qua các giao diện hoặc API, giúp tạo ra một hệ thống có tính tái sử dụng cao và dễ dàng mở rộng.

### 1. Ưu điểm:

1. **Tính Modularity (Tính mô-đun):** Kiến trúc layer cho phép hệ thống được phân chia thành các phần nhỏ, gọi là lớp, mỗi lớp đảm nhận một chức năng riêng biệt. Điều này làm cho hệ thống dễ quản lý và dễ mở rộng.
2. **Tính Tương tác Thấp (Low Coupling):** Các lớp không phụ thuộc trực tiếp lẫn nhau, mà thay vào đó chúng chỉ tương tác thông qua giao diện chuẩn hóa. Điều này giúp giảm sự phụ thuộc giữa các thành phần của hệ thống, làm cho việc thay đổi hoặc nâng cấp dễ dàng hơn.
3. **Tính Đa dạng (Abstraction):** Mỗi lớp ẩn đi chi tiết triển khai của các lớp dưới cùng, chỉ cung cấp một giao diện cho các lớp trên cùng. Điều này giúp tạo ra một cách nhìn trừu tượng và dễ hiểu về cách hoạt động của hệ thống.

### 2. Nhược điểm:

1. **Tăng Overhead (Tăng Chi phí):** Mỗi lớp thêm vào đều tăng overhead cho hệ thống, từ việc tạo và duy trì các giao diện đến việc truyền thông qua các lớp. Điều này có thể ảnh hưởng đến hiệu suất của hệ thống, đặc biệt là trong các hệ thống có số lượng lớn các lớp.
2. **Khó Khắc phục Sự cố (Troubleshooting Difficulty):** Khi một sự cố xảy ra, việc xác định lớp nào gây ra sự cố và nơi nó xảy ra có thể trở nên phức tạp. Điều này đặc biệt đúng khi mối quan hệ giữa các lớp phức tạp.

### 3. Một số Khía cạnh Khác:

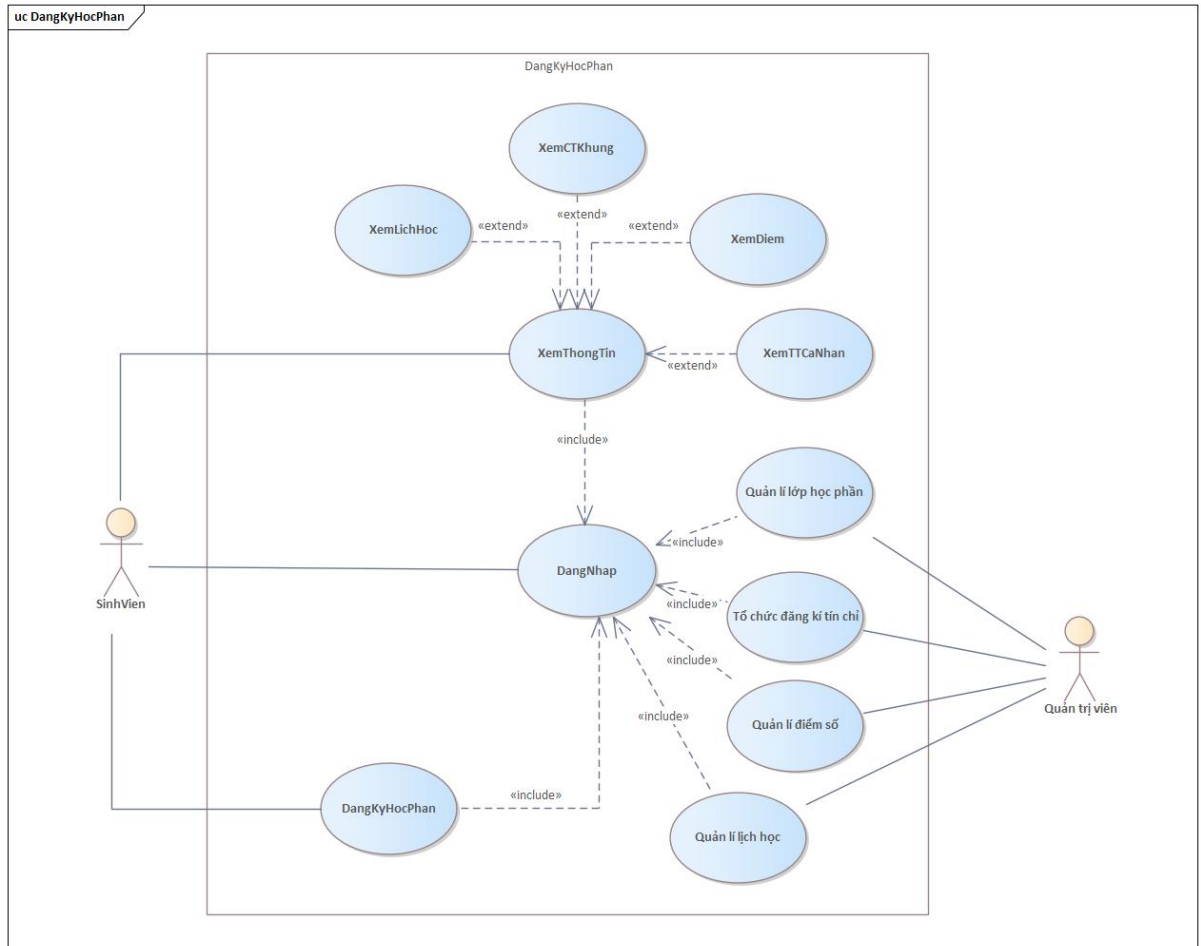


1. **Tính Bảo mật (Security):** Kiến trúc layer có thể được sử dụng để cải thiện bảo mật bằng cách tách biệt các lớp và áp dụng các biện pháp bảo mật tại mỗi lớp.
2. **Tính Tái sử dụng (Reusability):** Các lớp có thể được thiết kế để có thể tái sử dụng, giúp giảm thiểu việc viết lại mã và tăng tốc độ phát triển phần mềm.

## CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ

### 3.1 Phân tích yêu cầu bằng UML

#### 3.1.1 Usecase tổng quát



Hình 3.1: Sơ đồ Use Case tổng quát của hệ thống

#### 3.1.2 Danh sách tác nhân và mô tả

Actor	Mô tả
SinhVien	<p>Sinh viên, là người dùng chính của phần mềm đăng kí học phần này. Có thể thực hiện những chức năng chính của phần mềm:</p> <ul style="list-style-type: none"> <li>- Đăng nhập (bằng tài khoản do trường cấp)</li> <li>- Xem thông tin cá nhân</li> <li>- Xem thông tin lịch học, lịch thi</li> </ul>

	<ul style="list-style-type: none"><li>- Xem thông tin về kết quả học tập</li><li>- Đăng kí học phần</li><li>- Xem chương trình khung</li></ul>
--	--

Bảng 3.1: Hệ thống sử dụng kiến trúc phân lớp

### 3.1.3 Danh sách các tình huống hoạt động (Use cases)

ID use case	Tên Use case
UC01	Đăng nhập
UC02	Xem thông tin cá nhân
UC03	Đăng kí học phần
UC04	Xem lịch học
UC05	Xem chương trình khung
UC06	Xem kết quả học tập

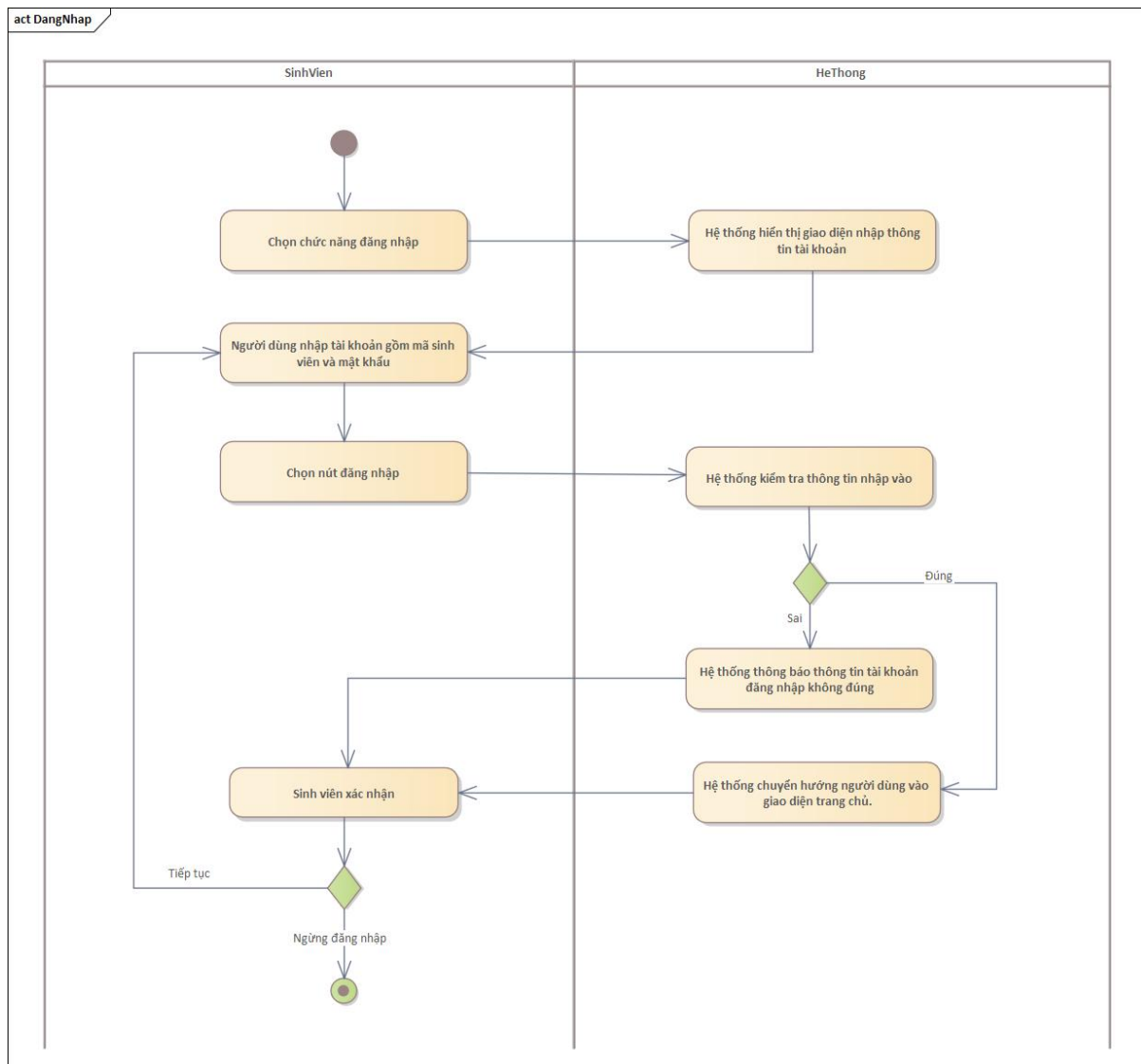
Bảng 3.2: Danh sách các tình huống trong hệ thống

### 3.1.4 Tình huống hoạt động

#### 3.1.4.1 UC01\_Đăng nhập

– <b>Tên use case:</b> Đăng nhập	
– <b>Mô tả sơ lược:</b> Sinh viên đăng nhập tài khoản cá nhân được trường cấp để vào trang web theo dõi lịch học, xem thông tin và các chức năng liên quan khác như đăng kí học phần, xem điểm,...	
– <b>Actor chính:</b> Sinh Viên	
– <b>Actor phụ:</b> Không	
– <b>Tiền điều kiện (Pre-condition):</b> Người dùng đã có tài khoản do trường cấp	
– <b>Hậu điều kiện (Post-condition):</b> Đăng nhập thành công, sinh viên có thể vào trang web để xem thông tin bản thân và thực hiện những chức năng liên quan	
– <b>Luồng sự kiện chính (main flow):</b>	
Actor	System
1. Người dùng chọn chức năng đăng nhập	2. Hệ thống hiển thị giao diện nhập thông tin tài khoản
3. Người dùng nhập tài khoản gồm mã sinh viên và mật khẩu	
4. Chọn nút đăng nhập	5. Hệ thống kiểm tra thông tin nhập vào
	6. Hệ thống chuyển hướng người dùng vào giao diện trang chủ.
7. Sinh viên xác nhận	
– <b>Luồng sự kiện thay thế (alternate flow):</b>	
	6.1. Hệ thống thông báo thông tin tài khoản đăng nhập không đúng
	6.2. Hệ thống quay lại bước 3

Báo cáo đồ án môn học – Môn Kiến trúc phần mềm  
Activity diagram:

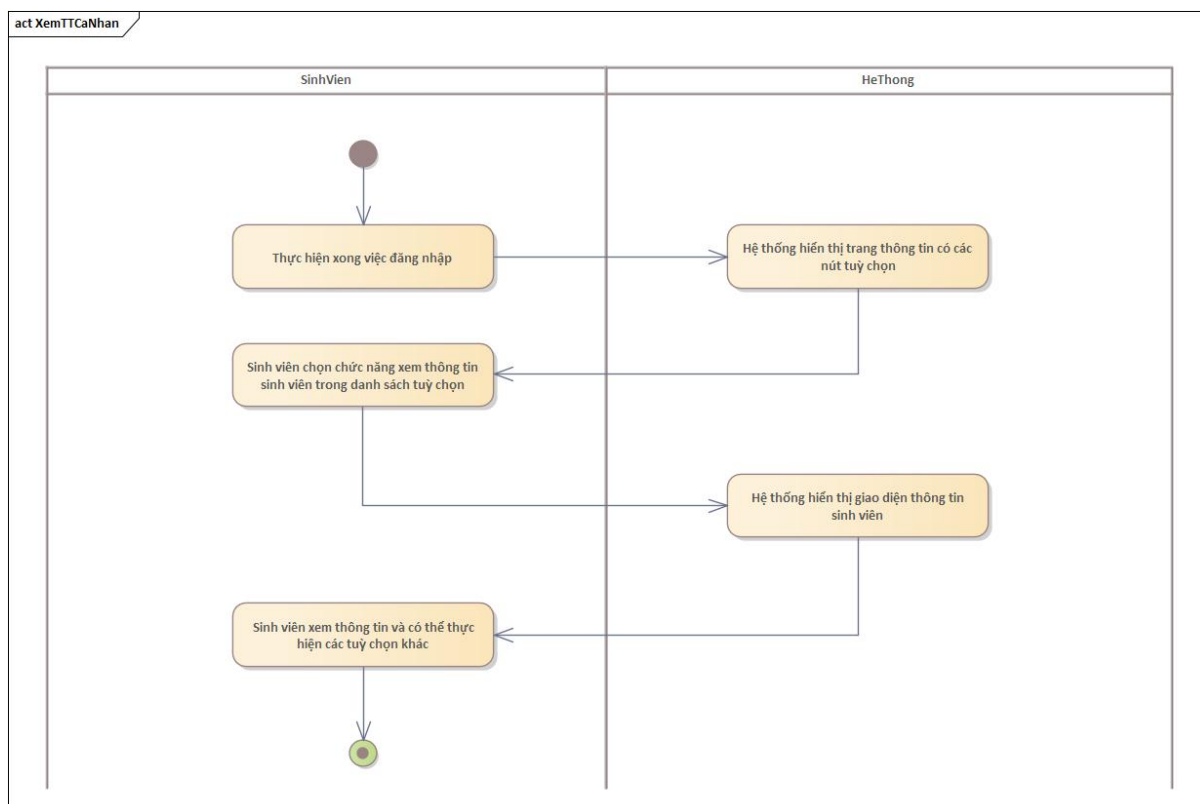


Sequence diagram:

**3.1.4.2UC02\_Xem thông tin cá nhân**

– <b>Tên use case:</b> Xem thông tin cá nhân	
– <b>Mô tả sơ lược:</b> Đăng nhập vào tài khoản cá nhân, sinh viên tự coi thông tin bản thân và có thể tiến hành chỉnh sửa thông tin nếu có sai sót	
– <b>Actor chính:</b> Sinh Viên	
– <b>Actor phụ:</b> Không	
– <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công	
– <b>Hậu điều kiện (Post-condition):</b> Có thể xem, cập nhật thông tin bản thân	
– <b>Luồng sự kiện chính (main flow):</b>	
Actor	System
1. Sinh viên thực hiện chức năng đăng nhập	2. Đăng nhập thành công, hệ thống hiển thị trang thông tin gồm: + Ảnh thẻ sinh viên + Các thông tin cơ bản: tên sinh viên, mã sinh viên, trạng thái học tập + Các nút tùy chọn xem thông tin, đăng kí học phần, xem chương trình khung
3. Sinh viên chọn chức năng xem thông tin sinh viên trong danh sách tùy chọn	4. Hệ thống hiển thị giao diện thông tin sinh viên
5. Sinh viên xem thông tin và có thể thực hiện các tùy chọn khác	
– <b>Luồng sự kiện thay thế (alternate flow):</b>	

--	--



Activity

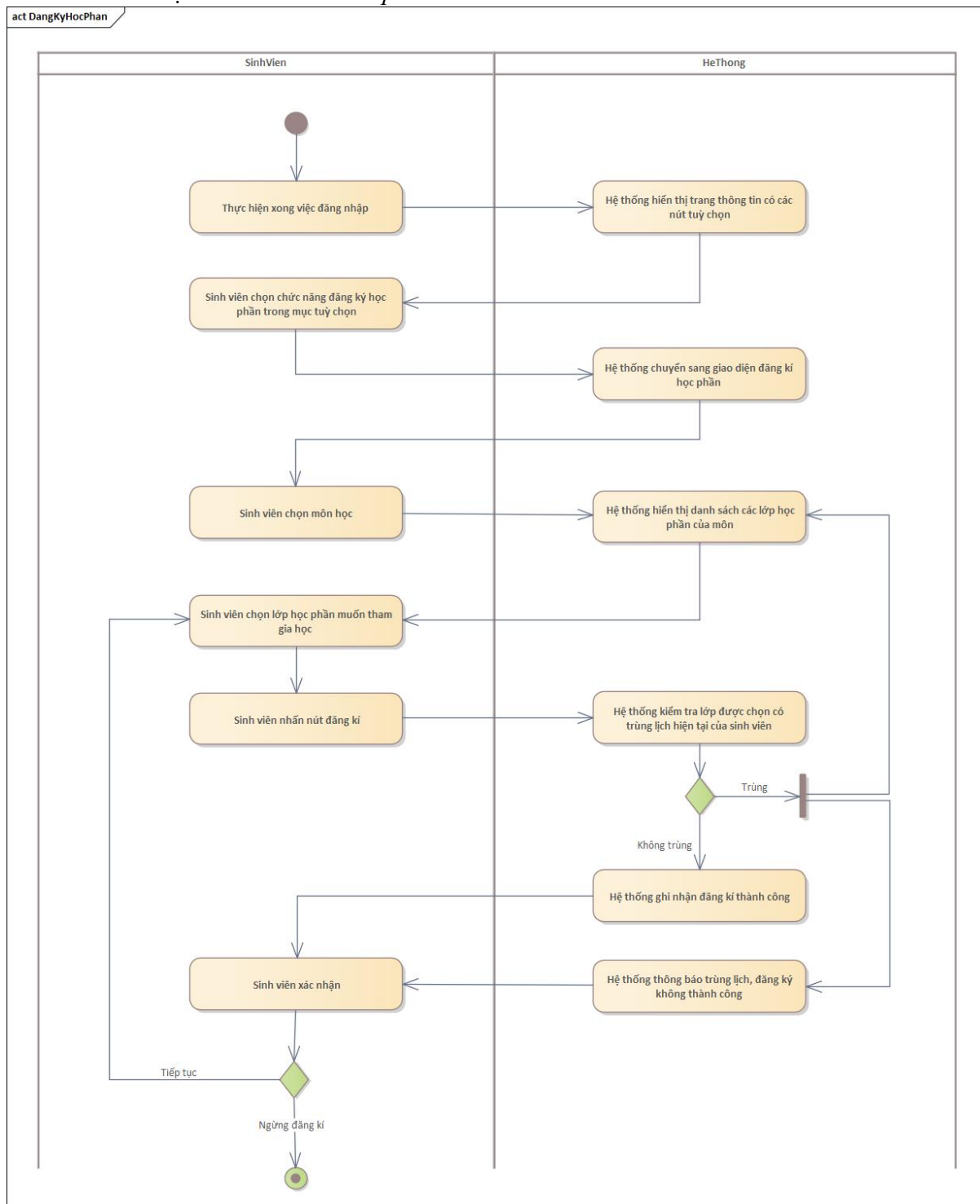
Sequence

### 3.1.4.3UC03\_Đăng kí học phần

– <b>Tên use case:</b> Đăng kí học phần	
– <b>Mô tả sơ lược:</b> Sinh viên đăng kí môn học trong học kì theo quy định của nhà trường, chức năng cho phép sinh viên chủ động chọn môn, lớp, giảng viên, giờ học để đăng ký tham gia lớp học đó.	
– <b>Actor chính:</b> Sinh Viên	
– <b>Actor phụ:</b> Không	
– <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công	
– <b>Hậu điều kiện (Post-condition):</b> Đăng kí thành công, sinh viên có tên trong danh sách lớp học, hiển thị công nợ của môn, hiển thị lịch học của môn đó.	
– <b>Luồng sự kiện chính (main flow):</b>	
Actor	System
1. Sinh viên thực hiện chức năng đăng nhập	2. Đăng nhập thành công, hệ thống hiển thị trang thông tin gồm: <ul style="list-style-type: none"> <li>+ Ảnh thẻ sinh viên</li> <li>+ Các thông tin cơ bản: tên sinh viên, mã sinh viên, trạng thái học tập</li> <li>+ Các nút tùy chọn xem thông tin, đăng kí học phần, xem chương trình khung</li> </ul>
3. Sinh viên chọn chức năng đăng ký học phần trong mục tùy chọn	4. Hệ thống chuyển sang giao diện đăng kí học phần, gồm bảng danh sách môn học của học kì kế tiếp
5. Sinh viên chọn môn học	6. Hệ thống hiển thị danh sách các lớp học phần của môn, thông tin gồm thời gian học, giảng viên, phòng học
7. Sinh viên chọn lớp học phần muốn tham gia học	



8. Sinh viên nhấn nút đồng ý	9. Hệ thống kiểm tra việc trùng lịch học của sinh viên
	10. Hệ thống ghi nhận đăng kí thành công, mục môn học đã đăng kí trong kì tiếp hiện thông tin lớp học phần sinh viên đã đăng kí
11. Sinh viên xác nhận thông tin, kết thúc	
<b>– Luồng sự kiện thay thế (alternate flow):</b>	
	9.1.1 Hệ thống thông báo sinh viên trùng lịch, đăng ký không thành công, quay lại bước 6
9.2 Sinh viên xác nhận, quay lại bước 7 để tiếp tục	
9.2.1 Sinh viên xác nhận, kết thúc không đăng kí nữa	

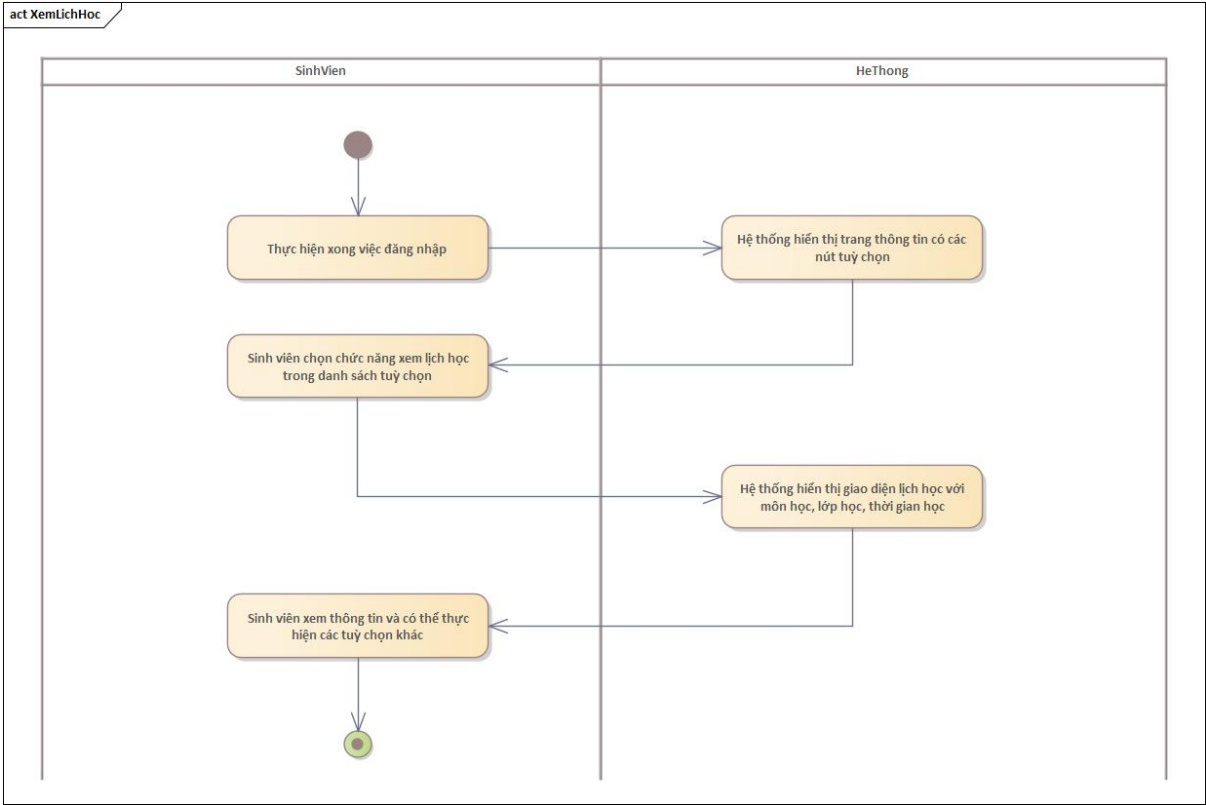


Sequence

**3.1.4.4UC04\_Xem lịch học**

– <b>Tên use case:</b> Xem lịch học	
– <b>Mô tả sơ lược:</b> Sinh viên xem lịch học của bản thân để biết thông tin lớp học, giờ học, môn học, giảng viên để tham gia đúng thời gian học tập	
– <b>Actor chính:</b> Sinh Viên	
– <b>Actor phụ:</b> Không	
– <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công, sinh viên đã đăng kí thành công môn học trước đó	
– <b>Hậu điều kiện (Post-condition):</b> Xem được lịch học, lịch thi của môn để chuẩn bị học tập đạt hiệu quả.	
– <b>Luồng sự kiện chính (main flow):</b>	
Actor	System
1. Sinh viên thực hiện chức năng đăng nhập	2. Đăng nhập thành công, hệ thống hiển thị trang thông tin gồm: <ul style="list-style-type: none"><li>+ Ảnh thẻ sinh viên</li><li>+ Các thông tin cơ bản: tên sinh viên, mã sinh viên, trạng thái học tập</li><li>+ Các nút tùy chọn xem thông tin, đăng kí học phần, xem chương trình khung, xem lịch học</li></ul>
3. Sinh viên chọn chức năng xem lịch học trong danh sách tùy chọn	4. Hệ thống hiển thị giao diện lịch học của sinh viên
5. Sinh viên xem thông tin và có thể thực hiện các tùy chọn khác	
– <b>Luồng sự kiện thay thế (alternate flow):</b>	


Activity

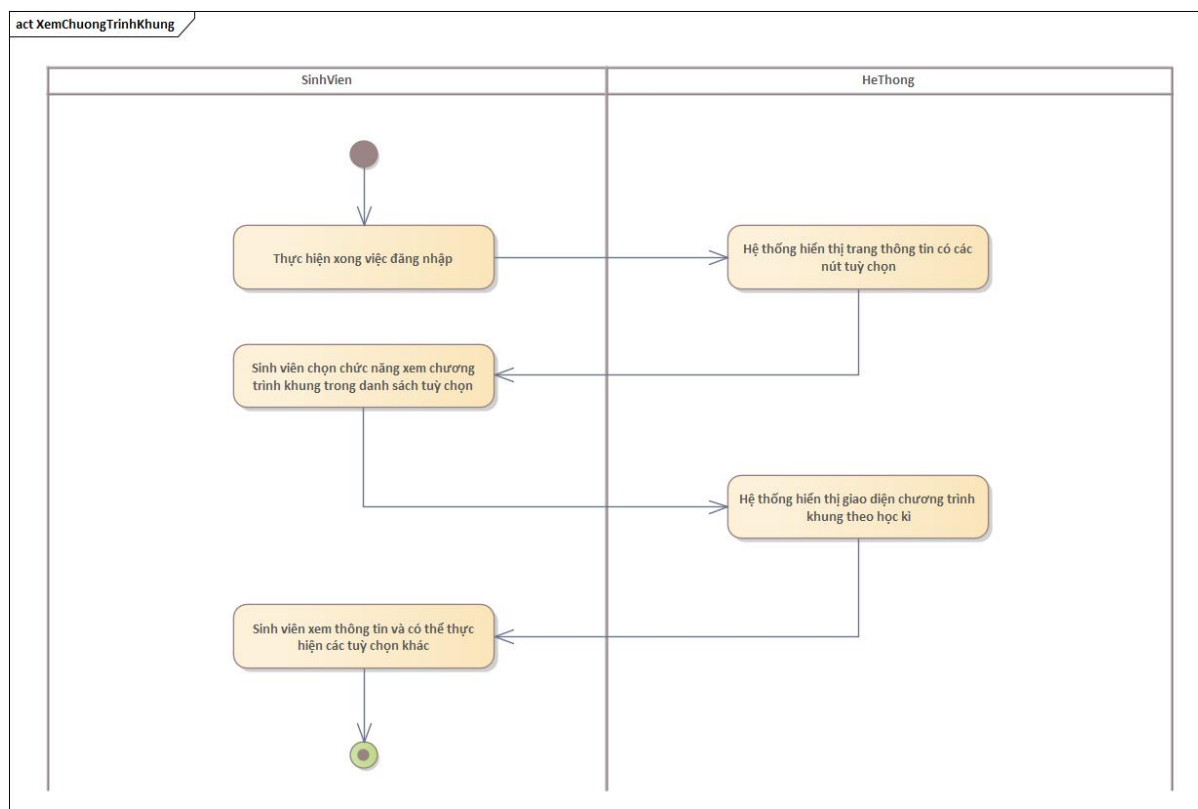


Sequence

**3.1.4.5UC05\_Xem chương trình khung**

– <b>Tên use case:</b> Xem chương trình khung	
– <b>Mô tả sơ lược:</b> Chương trình khung là danh sách các môn học phải trải qua trong suốt khoá học, chia sẵn theo từng kì với tiến độ phù hợp nhất có thể. Sinh viên xem theo đó để có thể tiến hành đăng kí môn cho phù hợp	
– <b>Actor chính:</b> Sinh Viên	
– <b>Actor phụ:</b> Không	
– <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công	
– <b>Hậu điều kiện (Post-condition):</b> Xem chương trình khung của ngành học của bản thân	
– <b>Luồng sự kiện chính (main flow):</b>	
Actor	System
1. Sinh viên thực hiện chức năng đăng nhập	2. Đăng nhập thành công, hệ thống hiển thị trang thông tin gồm: + Ảnh thẻ sinh viên + Các thông tin cơ bản: tên sinh viên, mã sinh viên, trạng thái học tập + Các nút tùy chọn xem thông tin, đăng kí học phần, xem chương trình khung
3. Sinh viên chọn chức năng xem chương trình khung trong danh sách tùy chọn	4. Hệ thống hiển thị giao diện chương trình khung chia theo từng học kì
5. Sinh viên xem thông tin và có thể thực hiện các tùy chọn khác	
– <b>Luồng sự kiện thay thế (alternate flow):</b>	

Activity:

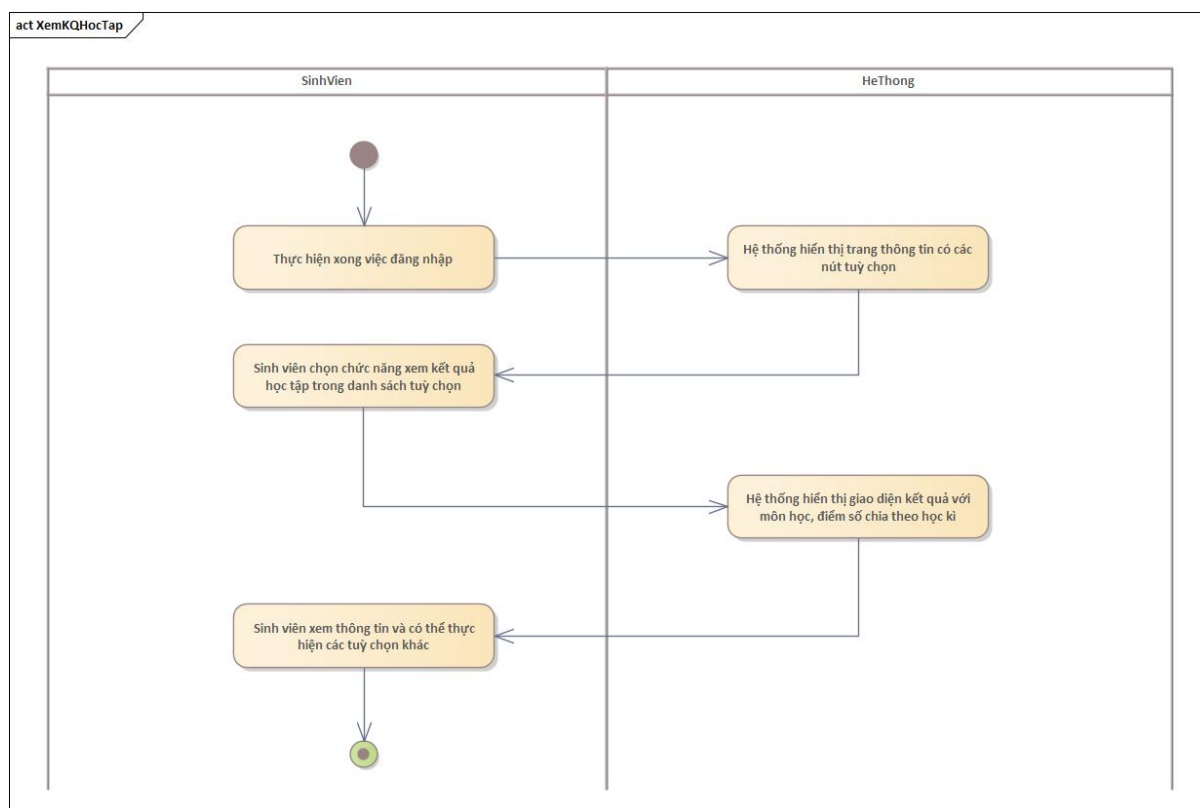


**3.1.4.6UC06\_Xem kết quả học tập**

– <b>Tên use case:</b> Xem kết quả học tập	
– <b>Mô tả sơ lược:</b> Sinh viên xem lại điểm số học tập từng môn trong từng học kỳ của mình tại web, có thể thực hiện các chức năng khác như in bảng điểm.	
– <b>Actor chính:</b> Sinh Viên	
– <b>Actor phụ:</b> Không	
– <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công, sinh viên đã đăng kí môn học và học hoàn tất môn, có tham gia thi và được chấm điểm đầy đủ.	
– <b>Hậu điều kiện (Post-condition):</b> Xem được kết quả từng môn, xem được điểm tích lũy và có thể in bảng điểm ấy ra giấy	
– <b>Luồng sự kiện chính (main flow):</b>	
Actor	System
1. Sinh viên thực hiện chức năng đăng nhập	2. Đăng nhập thành công, hệ thống hiển thị trang thông tin gồm: <ul style="list-style-type: none"><li>+ Ảnh thẻ sinh viên</li><li>+ Các thông tin cơ bản: tên sinh viên, mã sinh viên, trạng thái học tập</li><li>+ Các nút tùy chọn xem thông tin, đăng kí học phần, xem chương trình khung, xem kết quả học tập</li></ul>
3. Sinh viên chọn chức năng xem kết quả học tập trong danh sách tùy chọn	4. Hệ thống hiển thị giao diện kết quả học tập, gồm thông tin môn học theo từng học kỳ, điểm số tương ứng từng môn, điểm trung bình học kỳ và tích lũy
5. Sinh viên xem thông tin và có thể thực hiện các tùy chọn khác	

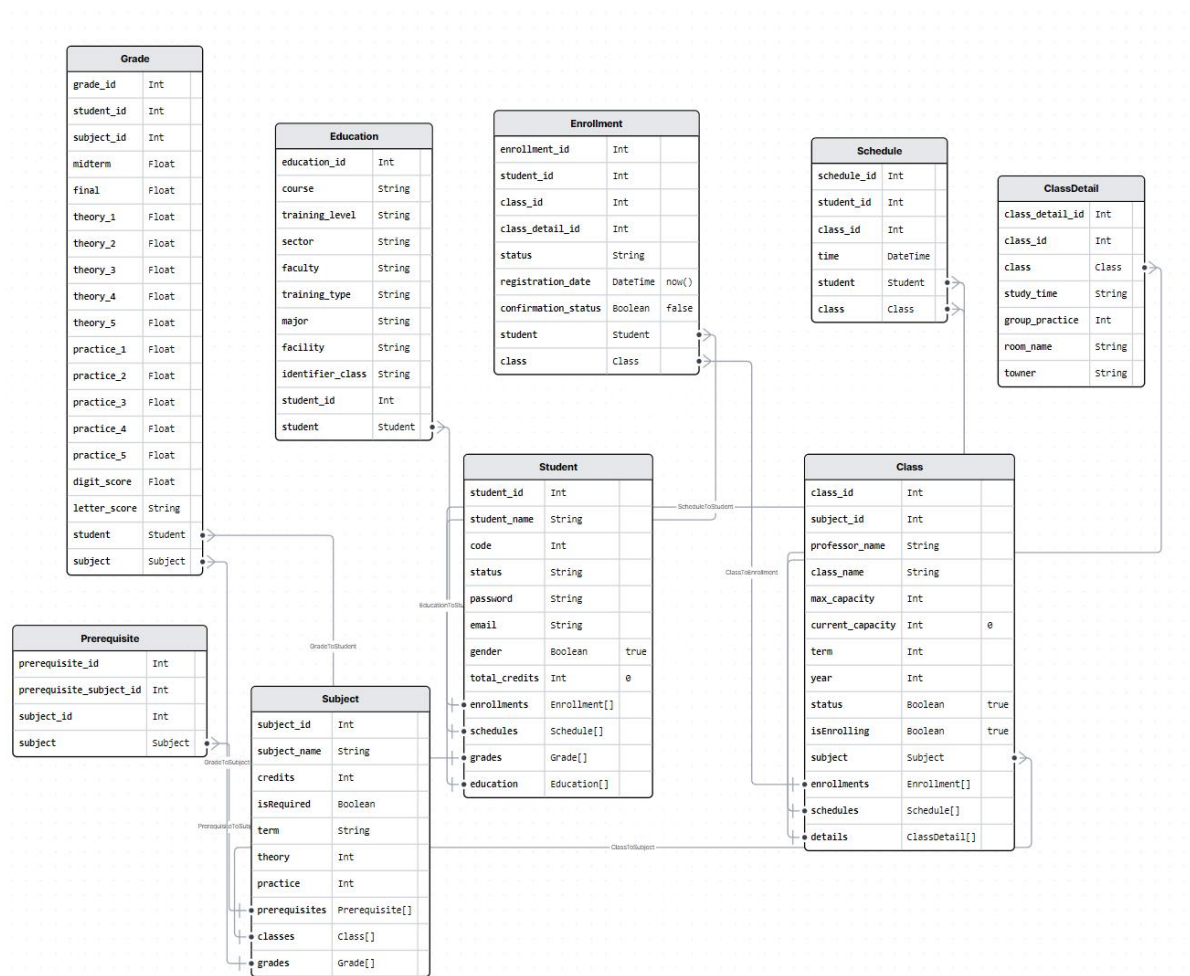
**– Luồng sự kiện thay thế (alternate flow):**


**Acticity**



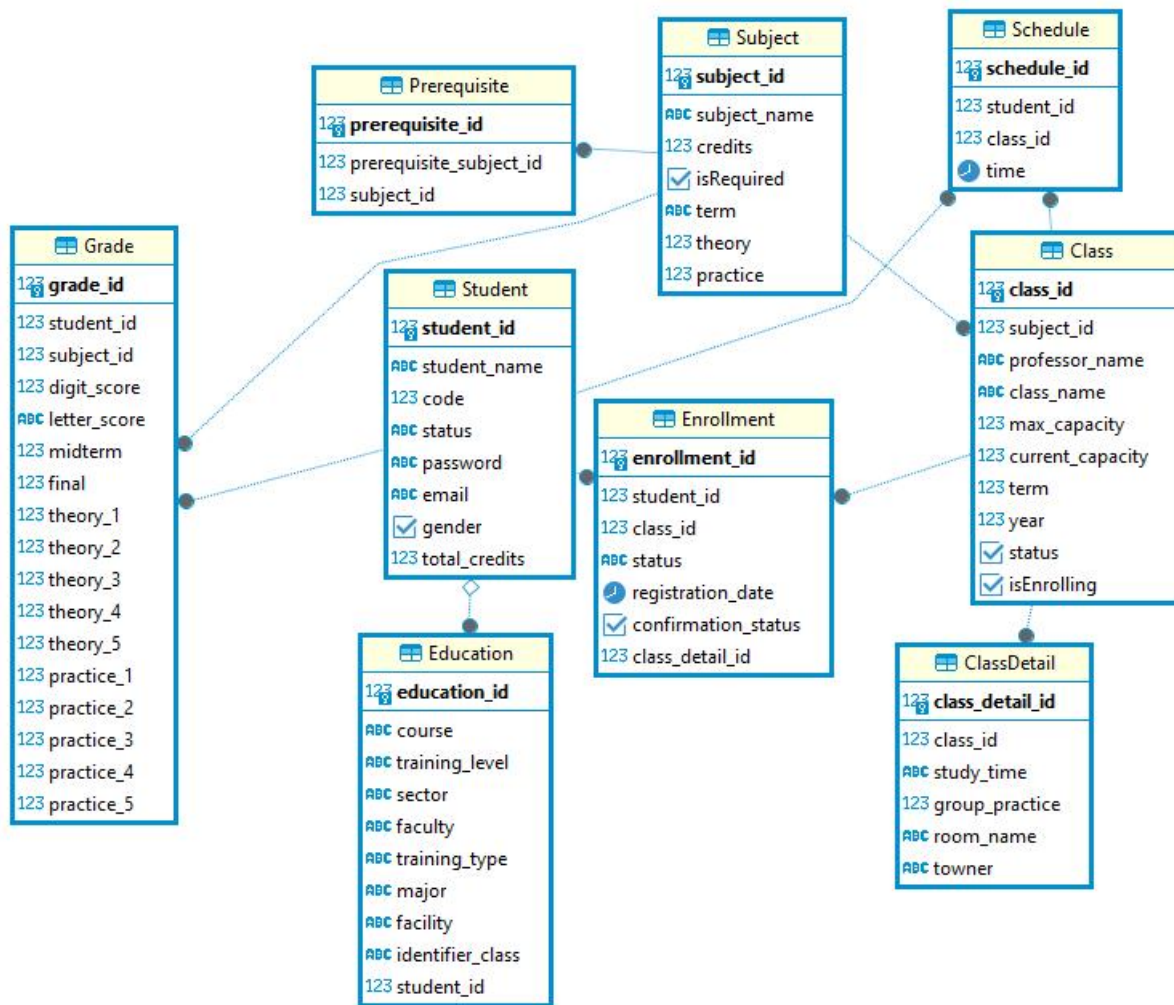


## 3.2 Class diagram



Hình 3.2: Sơ đồ lớp của hệ thống

### 3.3 ERD diagram



Hình 3.3: Sơ đồ mối quan hệ thực thể của hệ thống

### 3.4 Kiến trúc layer

Từ mặt cơ sở lý thuyết ở trên thì dưới đây là một số lí do vì sao chúng ta nên sử dụng kiến trúc Layer cho hệ thống quản lý lớp học tín chỉ:

#### 1. Tính Modularity (Tính mô-đun):

Với tính chất mô-đun, chúng ta có thể tách biệt các thành phần riêng biệt trong hệ thống:

- *Lớp Logic Ứng dụng (Application Logic Layer - Controller)*: Đảm bảo nhiệm vụ phân phối các yêu cầu từ người dùng đến các dịch vụ và xử lý các quy tắc nghiệp vụ của hệ thống.

*Báo cáo đồ án môn học – Môn Kiến trúc phần mềm*

- *Lớp Dịch vụ (Service Layer)*: Cung cấp các dịch vụ như gửi email thông báo, xác nhận đóng học phí, kiểm tra môn học tiên quyết, xác nhận đăng ký, kiểm tra số tín chỉ tối đa và nhiều logic khác.

- *Lớp Truy cập Dữ liệu (Data Access Layer)*: Quản lý việc lưu trữ và truy xuất dữ liệu về sinh viên, môn học, đăng ký môn học, điểm số.

## **2. Tính Tương tác Thấp (Low Coupling):**

Các lớp trong hệ thống không phụ thuộc trực tiếp lẫn nhau mà tương tác thông qua giao diện chuẩn hóa. Điều này giúp giảm sự phụ thuộc giữa các thành phần của hệ thống, làm cho việc thay đổi hoặc nâng cấp dễ dàng hơn. Ví dụ:

- Khi thay đổi quy tắc kiểm tra số tín chỉ tối đa hoặc bổ sung quy tắc mới, chỉ cần thay đổi trong lớp Logic Ứng dụng mà không ảnh hưởng đến các lớp khác.

- Khi thay đổi giao diện người dùng, lớp Logic Ứng dụng và lớp Truy cập Dữ liệu vẫn hoạt động bình thường.

## **3. Tính Đa dạng (Abstraction):**

Mỗi lớp ẩn đi chi tiết triển khai của các lớp dưới cùng, chỉ cung cấp một giao diện cho các lớp trên cùng. Điều này giúp tạo ra một cách nhìn trừu tượng và dễ hiểu về cách hoạt động của hệ thống.

- Sinh viên chỉ cần tương tác với giao diện người dùng mà không cần biết chi tiết về cách hệ thống kiểm tra các quy tắc nghiệp vụ hay lưu trữ dữ liệu.

- Các lớp dưới cung cấp các dịch vụ cần thiết mà không tiết lộ chi tiết về cách chúng được triển khai.

## **4. Tính Bảo mật (Security):**

Kiến trúc Layer có thể được sử dụng để cải thiện bảo mật bằng cách tách biệt các lớp và áp dụng các biện pháp bảo mật tại mỗi lớp. Ví dụ:

- Lớp Giao diện Người dùng có thể thực hiện các biện pháp bảo mật như xác thực và phân quyền người dùng.

- Lớp Logic Ứng dụng có thể kiểm tra các quyền truy cập và thực thi các quy tắc bảo mật.

- Lớp Truy cập Dữ liệu có thể bảo vệ dữ liệu bằng cách sử dụng các cơ chế mã hóa và kiểm tra quyền truy cập dữ liệu.

### **5. Tính Tái sử dụng (Reusability):**

Các lớp trong hệ thống có thể được thiết kế để có thể tái sử dụng. Ví dụ:

- Các dịch vụ gửi email, kiểm tra môn học tiên quyết, và xác nhận đăng ký có thể được sử dụng lại trong nhiều module khác nhau của hệ thống.
- Các thành phần giao diện người dùng có thể được tái sử dụng cho các chức năng tương tự.

### **6. Tối ưu hóa hiệu năng và chi phí:**

Với một ứng dụng để đáp ứng nhu cầu tuyển sinh tại một vùng cụ thể, việc tập trung nguồn tài nguyên tại một nơi sẽ đáp ứng được về mặt hiệu năng cũng như tối ưu về chi phí tại một số thời điểm nhất định. Kiến trúc Layer cho phép hệ thống được triển khai trên một hạ tầng tập trung, giúp giảm chi phí phần cứng và phần mềm, đồng thời tối ưu hóa việc quản lý và vận hành hệ thống. Hơn nữa, việc quản lý tài nguyên tập trung giúp dễ dàng theo dõi và điều chỉnh hiệu suất của hệ thống, đảm bảo đáp ứng kịp thời nhu cầu của sinh viên và nhà trường.

**Tóm lại**, việc chọn kiến trúc Layer cho hệ thống quản lý lớp học tin chỉ là khả thi và hợp lý, vì nó mang lại nhiều lợi ích về quản lý, bảo mật, tái sử dụng, và dễ dàng mở rộng, đồng thời giúp tổ chức hệ thống một cách rõ ràng và hiệu quả. Đồng thời, việc tập trung nguồn tài nguyên tại một nơi giúp tối ưu hóa hiệu năng và chi phí, đảm bảo hệ thống hoạt động hiệu quả trong các giai đoạn cao điểm của tuyển sinh. Mặc dù còn có những bất cập về mặt khó khắc phục sự cố hay tăng chi phí, nhưng với việc triển khai với một hệ thống có lượng người dùng không truy cập thường xuyên ở mức độ cao thì việc chọn kiến trúc Layer là một lựa chọn hợp lý.

## CHƯƠNG 4 : HIỆN THỰC

### 4.1 Giao diện của hệ thống

#### 4.1.1 Giao diện đăng nhập

The screenshot shows the login page of the Industrial University of Ho Chi Minh City (IUH) student portal. At the top, there is a header with the IUH logo and the text "BỘ CÔNG THƯƠNG TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH CÔNG ĐĂNG KÝ HỌC PHẦN SINH VIÊN". Below the header, there is a login form with two input fields: "Mã sinh viên" (Student ID) with the value "20005703" and "Mật khẩu" (Password) with the value "\*\*\*\*\*". A blue button labeled "Đăng nhập" (Login) is positioned below the password field. At the bottom of the page, there is a blue footer containing the university's address, phone number, fax, and email, along with a copyright notice: "Bản quyền của nhóm(2024) - Trường Đại học Công nghiệp TP. Hồ Chí Minh".

Hình 4.1: Giao diện đăng nhập vào hệ thống

#### 4.1.2 Giao diện Xem thông tin cá nhân

The screenshot shows the personal information page of the Industrial University of Ho Chi Minh City (IUH) student portal. At the top, there is a blue header with the text "Xin chào! Nguyễn Xuân Long" and a blue button labeled "Đăng xuất" (Logout). Below the header, there is a table with two columns: "THÔNG TIN SINH VIÊN" (Student Information) and "THÔNG TIN HỌC TẬP" (Study Information). The table contains the following data:

THÔNG TIN SINH VIÊN	THÔNG TIN HỌC TẬP
Giới tính: Nam	ĐĂNG KÝ HỌC PHẦN
MSSV: 20005703	XEM LỊCH HỌC
Trạng thái: Đang học	CHƯƠNG TRÌNH KHUNG

Below the table, there is a section titled "THÔNG TIN SINH VIÊN" (Student Information) with the following details:

- Khóa: 2020-2021
- Bậc đào tạo: Đại học
- Ngành: Kỹ thuật phần mềm
- Khoa: Khoa Công nghệ Thông tin
- Lớp: Đại học Kỹ thuật phần mềm 16A - 7480103
- Loại hình đào tạo: Chính quy
- Chuyên ngành: Kỹ thuật phần mềm - 7480103
- Cơ sở: Cơ sở 1 (Thành phố Hồ Chí Minh)

At the bottom of the page, there is a blue footer containing the university's address, phone number, fax, and email, along with a copyright notice: "Bản quyền của nhóm(2024) - Trường Đại học Công nghiệp TP. Hồ Chí Minh".

Hình 4.2: Giao diện chính sau khi đăng nhập thành công

## Báo cáo đồ án môn học – Môn Kiến trúc phần mềm

### 4.1.3 Giao diện đăng kí học phần

**ĐĂNG KÝ HỌC PHẦN**

Đợt đăng ký: HK2 (2021-2022) ▾ ☒ HỌC MỚI ☐ HỌC LẠI ☐ HỌC CẢI THIẾN

	STT	Mã HP	Tên môn học	Số TC	Bắt buộc	Học phần học trước
<input type="radio"/>	1	526007	Kỹ thuật lập trình	3	✓	227801
<input checked="" type="radio"/>	2	806936	Nhập môn Tin học	2	✓	
<input type="radio"/>	3	172283	Lập trình hướng đối tượng	3	✓	227801
<input type="radio"/>	4	640361	Quản trị doanh nghiệp	3	✗	
<input type="radio"/>	5	669904	Kĩ năng xây dựng kế hoạch	3	✗	
<input type="radio"/>	6	789571	Quản trị học	3	✗	
<input type="radio"/>	7	806092	Kế toán cơ bản	3	✗	
<input type="radio"/>	8	894143	Giao tiếp kinh doanh	3	✗	
<input type="radio"/>	9	966490	Toán cao cấp 2	2	✓	

**Lớp học phần chờ đăng ký**

	STT	Mã LHP	Tên lớp học phần	Sĩ số tối đa	Đã đăng ký	Trạng thái
<input checked="" type="radio"/>	1	281577	Nhập môn Tin học	100	0	Chờ đăng ký
<input type="radio"/>	2	489597	Nhập môn Tin học	100	0	Chờ đăng ký

**Chi tiết lớp học phần**

	STT	Lịch học	Nhóm TH	Phòng	Dãy nhà	Giảng viên
<input type="radio"/>	1	LT - Thứ 5(T10 -> T12)	2	A1-101	A1	Võ Văn Hải
<input type="radio"/>	2	LT - Thứ 6(T10 -> T12)	3	A1-102	A1	Võ Văn Hải
<input checked="" type="radio"/>	3	LT - Thứ 7(T10 -> T12)	4	A1-103	A1	Võ Văn Hải

Đăng ký

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HỒ CHÍ MINH  
 Địa chỉ: Số 12 Nguyễn Văn Bảo, Phường 4, Quận Gò Vấp, TP. Hồ Chí Minh  
 Điện thoại: 0283.8940.390

Hình 4.3: Giao diện đăng kí học phần

### 4.1.4 Giao diện xem lịch học

<
>
today
week

	Sun 5/19	Mon 5/20	Tue 5/21	Wed 5/22	Thu 5/23	Fri 5/24	Sat 5/25
6:30am				Lập trình hướng đối tượng			
7:30am				5979308 - ĐHKTPM16A			
8:30am				Tiết: 1-3			
				Phòng: H3-01			
				Tòa: H3			
				GV: Châu Thị Bảo Hà			
9:30am							
10:30am							
11:30am							
12:30pm							
1:30pm							
2:30pm							
3:30pm				Nhập môn Tin học	Nhập môn Tin học		
				489597 - ĐHKTPM15A	281577 - ĐHKTPM15A		
				Tiết: 10-12	Tiết: 10-12		
				Phòng: A1-101	Phòng: A1-101		
				Tòa: A1	Tòa: A1		
				GV: Võ Văn Hải	GV: Võ Văn Hải		
5:30pm							
6:30pm							
7:30pm							

Hình 4.4: Giao diện xem lịch học

Báo cáo đồ án môn học – Môn Kiến trúc phần mềm  
4.1.5 Giao diện xem chương trình khung

STT	Mã môn học	Tên môn học	Học phần học trước	Số TC/ĐVHT	Số tiết lí thuyết	Số tiết TH	Đạt
Học kì 1 - Năm 1					Tổng số tính chỉ: 14		
Học phần bắt buộc							
1	235767	Kiến trúc và Thiết kế phần mềm		3	45	30	
2	806936	Nhập môn Tin học		2	45	0	✓
3	158095	Kỹ năng làm việc nhóm		2	45	0	
4	239405	Toán cao cấp 1		2	45	0	
5	227801	Nhập môn Lập trình		2	45	0	✓
6	843452	Triết học Mác - Lênin		3	45	0	
Học kì 2 - Năm 1					Tổng số tính chỉ: 23		
Học phần bắt buộc							
1	526007	Kỹ thuật lập trình	227801	3	45	0	✓
2	598015	Kinh tế chính trị Mác - Lênin	843452	2	45	0	
3	304582	Hệ Thống Máy tính		3	45	0	
Học phần tự chọn							
1	481239	Toán ứng dụng		3	45	0	
2	488350	Hàm phức và phép biến đổi Laplace		3	45	0	
3	265362	Phương pháp tính		3	45	0	
4	999745	Vật lý đại cương		3	45	0	
5	879227	Logic học		3	45	0	
Học kì 1 - Năm 2					Tổng số tính chỉ: 27		
Học phần bắt buộc							
1	361512	Hệ cơ sở dữ liệu	806936	4	45	30	
2	966490	Toán cao cấp 2		2	30	0	
3	172283	Lập trình hướng đối tượng	227801	3	30	30	
Học phần tự chọn							
1	894143	Giao tiếp kinh doanh		3	45	0	
2	669904	Kĩ năng xây dựng kế hoạch		3	45	0	
3	640361	Quản trị doanh nghiệp		3	45	0	
4	569598	Môi trường và con người		3	45	0	
5	789571	Quản trị học		3	45	0	
6	806092	Kế toán cơ bản		3	45	0	
Học kì 2 - Năm 2					Tổng số tính chỉ: 12		
Học phần bắt buộc							

Hình 4.5: Giao diện xem chương trình khung

4.1.6 Giao diện xem kết quả học tập

STT	Mã lớp học/phần	Tên môn học/học phần	Giữa kỳ	Thường xuyên					Thực hành					Cuối kì	Điểm tổng	Thang điểm 4	Điểm chữ	Xếp loại
				1	2	3	4	5	1	2	3	4	5					
Học kì 1 (2021-2022)																		
1	961552	Kiến trúc và Thiết kế phần mềm	3	3	3	3								3	3	1.2	F	Kém
2	782584	Nhập môn Tin học	9	9	9	9								9	9	3.6	A+	Xuất sắc
3	942445	Nhập môn Lập trình	9	9	6.5	9								9	8.6	3.44	A	Giỏi
Điểm trung bình học kì hệ 10: 6.87			Điểm trung bình học kì hệ 4: 2.75															
Điểm trung bình tích lũy: 6.87			Điểm trung bình tích lũy hệ 4: 2.75															
Tổng số tín đăng ký: 7			Tổng số tín chỉ tích lũy: 4															
Tổng số tín chỉ đạt: 4			Tổng số tín chỉ nợ tính đến hiện tại: 3															
Xếp loại học lực tích lũy: Khá			Xếp loại học lực học kỳ: Khá															
Học kì 2 (2021-2022)																		
1	612914	Kỹ thuật lập trình	8	8	8	8			8	10	8			8	8	3.2	B+	Khá
2	612914	Kỹ thuật lập trình	10	10	10	10	10	10	10	10	10			10	10	4	F	Xuất sắc
Điểm trung bình học kì hệ 10: 9.00			Điểm trung bình học kì hệ 4: 3.60															
Điểm trung bình tích lũy: 7.94			Điểm trung bình tích lũy hệ 4: 3.17															
Tổng số tín đăng ký: 13			Tổng số tín chỉ tích lũy: 10															
Tổng số tín chỉ đạt: 6			Tổng số tín chỉ nợ tính đến hiện tại: 3															
Xếp loại học lực tích lũy: Khá			Xếp loại học lực học kỳ: Xuất sắc															

Hình 4.6: Giao diện xem kết quả học tập



## **CHƯƠNG 5 : KẾT LUẬN**

### **5.1 Kết quả đạt được**

- Hoàn thiện được những chức năng cơ bản các tính năng đề ra và có kế hoạch xây dựng cụ thể
- Phần mềm có khả năng triển khai thực tế
- Các thành viên đã học được những công nghệ mới và áp dụng được vào sản phẩm
- Thuần thục hơn trong việc làm nhóm, chủ động trao đổi và đề xuất những yêu cầu tính năng phù hợp. Có sự giao tiếp rõ ràng giữa các thành viên trong nhóm để đảm bảo tất cả thành viên có thể nắm bắt được về nghiệp vụ của phần mềm

### **5.2 Hạn chế của đồ án**

- Việc áp dụng nhiều công nghệ mới chưa được sử dụng gây khó khăn khi làm quen và để áp dụng vào thực tế phát triển phần mềm ngay lập tức
- Việc quản lý thời gian và bám sát theo kế hoạch ban đầu vẫn chưa được tốt với nguyện vọng ban đầu

### **5.3 Hướng phát triển**

- Tiếp tục phát triển những tính năng trong backlog còn tồn đọng
- Phát triển những tính năng như công nợ, thanh toán trực tuyến và chỉnh sửa giao diện
- Chuyển hướng kiến trúc sang microservice ở phía Backend và micro Front End ở phía UI
- Phát triển tiếp app mobile cho phần mềm để đồng bộ giữa Web/App.



*Báo cáo đồ án môn học – Môn Kiến trúc phần mềm*  
**TÀI LIỆU THAM KHẢO**