**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**Trang web đăng ký học phần sinh viên**

GVHD: Võ Văn Hải

***Nhóm 26 - Sinh viên thực hiện***

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Tên | Lớp | Vai trò |
| Trương Bình Triết | 20044651 | DHKTPM16A | Nhóm trưởng |
| Văn Quang Phong | 20022731 | DHKTPM16A | Thành viên |
| Lê Thiện Nghĩa | 20038061 | DHKTPM16A | Thành viên |

MỤC LỤC

[CHƯƠNG 1 : GIỚI THIỆU 2](#_Toc167017310)

[1.1 Tổng quan 2](#_Toc167017311)

[1.2 Mục tiêu đề tài 2](#_Toc167017312)

[*1.2.1* Mục tiêu chính 2](#_Toc167017313)

[*1.2.2* Mục tiêu chi tiết 2](#_Toc167017314)

[1.3 Phạm vi đề tài 2](#_Toc167017315)

[1.4 Mô tả yêu cầu chức năng 2](#_Toc167017316)

[1.5 Yêu cầu phi chức năng 3](#_Toc167017317)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 4](#_Toc167017318)

[2.1 Node.js 4](#_Toc167017319)

[2.2 MongoDB 4](#_Toc167017320)

[2.3 TypeScipt 4](#_Toc167017321)

[2.4 Nodemailer 5](#_Toc167017322)

[2.5 Axios 5](#_Toc167017323)

[2.6 Phân tích yêu cầu bằng UML 6](#_Toc167017324)

[*2.6.1* Usecase tổng quát 6](#_Toc167017325)

[*2.6.2* Danh sách tác nhân và mô tả 6](#_Toc167017326)

[*2.6.3* Danh sách các tình huống hoạt động (Use cases) 7](#_Toc167017327)

[*2.6.4* Tình huống hoạt động 7](#_Toc167017328)

[2.7 Class diagram 27](#_Toc167017329)

[2.8 Deployment diagram 27](#_Toc167017330)

[2.9 Giới thiệu và mục đích 28](#_Toc167017331)

[*2.9.1* Giới thiệu 28](#_Toc167017332)

[*2.9.2* Mục tiêu 28](#_Toc167017333)

[2.10 Ràng buộc 28](#_Toc167017334)

[2.11 Bối cảnh và phạm vi 29](#_Toc167017335)

[2.12 Chiến lược giải pháp 29](#_Toc167017336)

[2.13 Quyết định kiến trúc 30](#_Toc167017337)

[*2.13.1* Mô hình 30](#_Toc167017338)

[*2.13.2* Ưu điểm/ nhược điểm 32](#_Toc167017339)

[*2.13.3* Quy trình 33](#_Toc167017340)

[2.14 Khái niệm 34](#_Toc167017341)

[*2.14.1* Kiến trúc của một hệ thống phần mềm (Software Architecture) 34](#_Toc167017342)

[*2.14.2* Thiết kế phần mềm (Software Design) 34](#_Toc167017343)

[*2.14.3* Phong cách kiến trúc (Architecture Style) 34](#_Toc167017344)

[*2.14.4* Kiến trúc phân tầng (Layered Architecture) 35](#_Toc167017345)

[2.15 Tình huống/ yêu cầu 35](#_Toc167017346)

[2.16 Lợi ích/ đánh đổi 36](#_Toc167017347)

[*2.16.1* Presentation Layer: 36](#_Toc167017348)

[*2.16.2* Business Layer: 37](#_Toc167017349)

[*2.16.3* Data Layer: 37](#_Toc167017350)

[CHƯƠNG 3 : HIỆN THỰC 38](#_Toc167017351)

[3.1 Cấu hình phần cứng, phần mềm 38](#_Toc167017352)

[3.2 Giao diện của hệ thống 38](#_Toc167017353)

[*3.2.1* Trang đăng nhập 38](#_Toc167017354)

[*3.2.2* Trang chủ 39](#_Toc167017355)

[*3.2.3* Trang đăng ký học phần 39](#_Toc167017356)

[*3.2.4* Trang hủy học phần 41](#_Toc167017357)

[*3.2.5* Trang lịch học 42](#_Toc167017358)

[*3.2.6* Trang kết quả học tập 42](#_Toc167017359)

[3.3 Kế hoạch và hiện thực kiểm thử hệ thống 43](#_Toc167017360)

[*3.3.1* Kế hoạch kiểm thử 43](#_Toc167017361)

[*3.3.2* Kiểm thử hệ thống 44](#_Toc167017362)

[CHƯƠNG 4 : KẾT LUẬN 46](#_Toc167017363)

[4.1 Kết quả đạt được 46](#_Toc167017364)

[4.2 Hạn chế của đồ án 46](#_Toc167017365)

[4.3 Hướng phát triển 46](#_Toc167017366)

[TÀI LIỆU THAM KHẢO 47](#_Toc167017367)

# : GIỚI THIỆU

## Tổng quan

Hệ thống đăng ký học phần của sinh viên theo theo cơ chế tín chỉ. Các sinh viên sẽ được biên chế ở các khoa theo mỗi ngành học của mình đăng ký.

Trong mỗi học kỳ, nhà trường sẽ chọn một số môn học của từng ngành học cho sinh viên đăng ký học. Sinh viên đăng ký theo học vào một lớp nhất định (lớp tín chỉ) với số sinh viên được đăng ký tối đa được qui định cho từng lớp.

## Mục tiêu đề tài

### Mục tiêu chính

Xây dựng hệ thống quản lý lớp học tín chỉ hiệu quả, đáp ứng nhu cầu quản lý học tập của nhà trường và sinh viên.

### Mục tiêu chi tiết

* Quản lý thông tin sinh viên
* Đăng ký học phần
* Quản lý thông tin học tập
* Quản lý thông báo và học phí
* Cải thiện trãi nghiệm người dùng

## Phạm vi đề tài

- Đề tài này tập trung vào việc thiết kế và triển khai một hệ thống quản lý lớp học tín chỉ cho sinh viên trường Đại học Công nghiệp thành phố Hồ Chí Minh (IUH).

## Mô tả yêu cầu chức năng

- Đăng nhập: sinh viên đăng nhập bằng mã sinh viên và mật khẩu để vào tài khoản sinh viên

- Đăng xuất: sinh viên đăng xuất khỏi tài khoản sinh viên

- Lịch học theo tuần: Sinh viên xem lịch học theo tuần

- Lich thi: Sinh viên xem lịch thi của môn học phần

- Đăng ký học phần: sinh viên đăng ký theo học vào một lớp học phần nhất định (lớp tín chỉ) với số sinh viên được đăng ký tối đa được qui định cho từng lớp học phần (tối đa 30 tín chỉ 1 học kì ).

- Hủy học phần: Sinh viên hủy học phần đã đăng ký

- Xem kết quả học tập: Sinh viên xem thông tin học tập của mình (số tín chỉ đã đạt, số môn đã học, điểm môn học, điểm trung bình tích lũy…)

- Tra cứu công nợ: Sinh viên xem công nợ.

- Xem học phần đã đăng ký: Sinh viên xem học phần đã đăng ký

## Yêu cầu phi chức năng

* Bảo mật và quyền riêng tư: Đảm bảo dữ liệu, thông tin người dùng được bảo vệ an toàn tuân thủ các quy định về quyền riêng tư.
* Giao diện người dùng: Thân thiện, dễ sử dụng, nhất quán và trực quan.
* Độ tin cậy và khả năng sẵn sàng: Đáp ứng được một lượng lớn người dùng cùng một lúc mà vẫn đảm bảo tính ổn định và sẵn sàng.

- Khả năng mở rộng : Dễ dàng thêm, sửa đổi các chức năng khi cần thiết.

# : CƠ SỞ LÝ THUYẾT

## Node.js

Node.js là một nền tảng JavaScript phía máy chủ, được xây dựng trên V8 JavaScript engine của Google Chrome. Nó cho phép phát triển các ứng dụng mạng hiệu năng cao, có khả năng mở rộng lớn với kiến trúc sự kiện bất đồng bộ (asynchronous event-driven architecture).



Hình 2‑1 Node.js development

## MongoDB

DynamoDB là một dịch vụ cơ sở dữ liệu NoSQL mạnh mẽ, linh hoạt và có khả năng mở rộng, phù hợp cho các ứng dụng yêu cầu độ trễ thấp và hiệu suất cao. Với các tính năng như chỉ mục phụ, giao dịch ACID, DynamoDB Streams, và Global Tables, nó cung cấp các công cụ mạnh mẽ để xây dựng các ứng dụng hiện đại, phức tạp. Hiểu rõ các cơ sở lý thuyết và tính năng của DynamoDB sẽ giúp bạn thiết kế và triển khai các giải pháp cơ sở dữ liệu hiệu quả và tối ưu.

## TypeScipt

TypeScript là một ngôn ngữ lập trình mã nguồn mở được phát triển bởi Microsoft, là một siêu ngữ của JavaScript (superset of JavaScript). Nó bổ sung các tính năng như kiểu tĩnh (static typing), các lớp (classes), và các module, giúp phát triển các ứng dụng JavaScript lớn và phức tạp dễ dàng hơn.

## Nodemailer

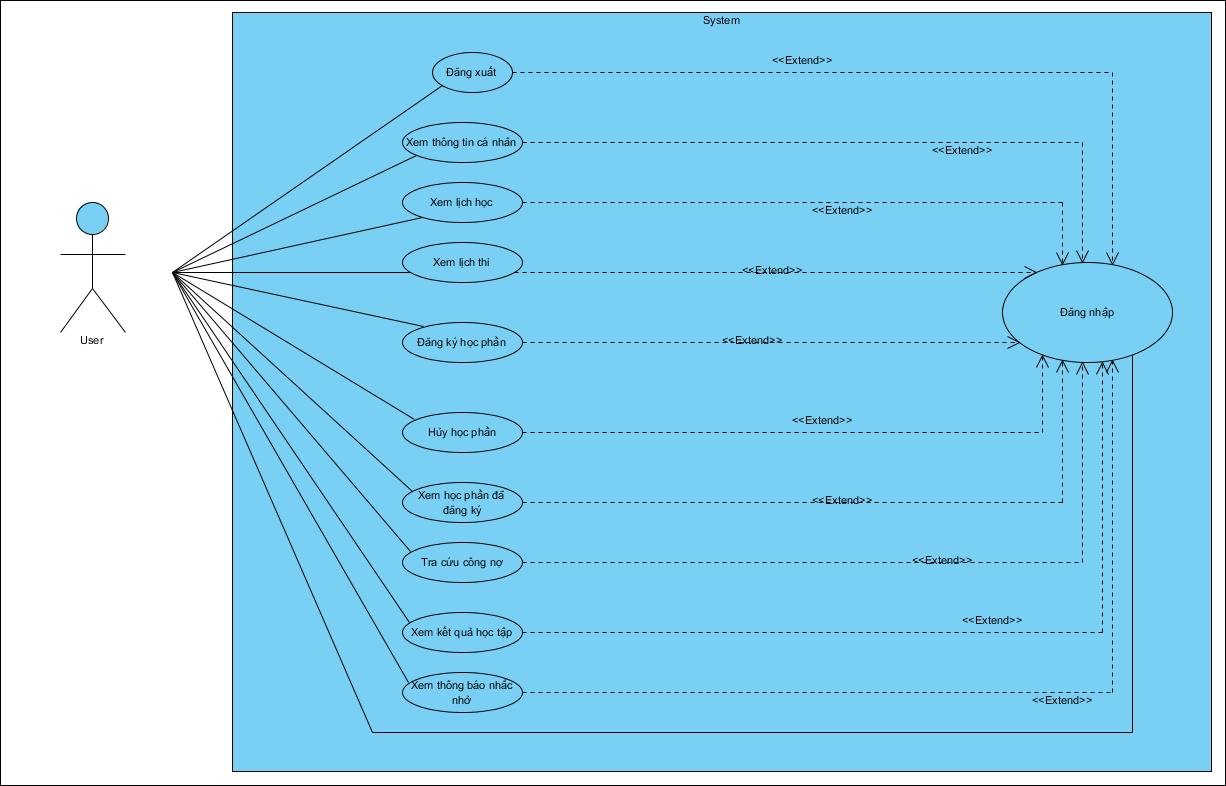
Nodemailer là một công cụ mạnh mẽ và dễ sử dụng cho việc gửi email trong ứng dụng Node.js. Với khả năng hỗ trợ nhiều phương thức vận chuyển và tính năng linh hoạt, Nodemailer là lựa chọn hàng đầu cho việc gửi email trong môi trường Node.js.

## Axios

Axios là thư viện giúp client tương tác với server thông qua giao thức HTTP dựa trên các Promises. Ưu điểm chính của Axios so với XMLHttpRequest truyền thống là giúp cho code của chúng ta gọn gàng và dễ đọc hơn. Ngoài việc cung cấp các phương thức như GET, POST, PUT,... thì Axios còn cung cấp thêm cho chúng ta rất nhiều tính năng bổ sung khác. Chính vì thế Axios đang là thư viện được sử dụng rất phổ biến ở các ứng dụng web SPA hiện đại.: PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát



### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| User | Người dùng (User) là một vai trò quan trọng trong hệ thống phần mềm quản lý học tập của trường học. Đối tượng này bao gồm sinh viên hoặc bất kỳ ai được cấp quyền truy cập vào các tính năng học tập và quản lý cá nhân trên hệ thống. Người dùng có thể thực hiện nhiều tác vụ liên quan đến việc quản lý quá trình học tập của mình một cách tiện lợi và hiệu quả. Nền tảng dễ sử dụng và hiệu quả, giúp người dùng quản lý các khía cạnh quan trọng của quá trình học tập, từ đăng ký học phần đến việc theo dõi lịch học và lịch thi, cũng như quản lý thông tin cá nhân. |

Bảng 3‑1 Danh sách tác nhân và mô tả

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

|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Đăng nhập |
| UC02 | Đăng xuất |
| UC03 | Xem lịch học |
| UC04 | Xem lịch thi |
| UC05 | Đăng ký học phần |
| UC06 | Hủy học phần |
| UC07 | Xem kết quả học tập |
| UC08 | Tra cứu công nợ |
| UC09 | Xem học phần đã đăng ký |
| UC10 | Xem thông báo nhắc nhở |

### Tình huống hoạt động

#### UC01

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Đăng nhập | |
| * **Mô tả sơ lược**: Đăng nhập giúp sinh viên vào tài khoản sinh viên của mình | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Tài khoản tồn tại trong hệ thống | |
| * **Hậu điều kiện (Post-condition):** Đăng nhập vào được hệ thống | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng nhập thông tin đăng nhập   (mã số sinh viên, mật khẩu) |  |
| 1. Nhấn đăng nhập | 1. Hệ thống kiểm tra thông tin đăng nhập |
|  | 1. Hệ thống hiển thị trang chủ |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 3.1 Hệ thống kiểm tra đăng nhập sai |
|  | 3.2 Hiển thị thông báo sai mật khẩu hoặc số điện thoại |
| 3.3 Quay lại bước 1 |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |

* Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

#### UC02

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Đăng xuất | |
| * **Mô tả sơ lược**: Sinh viên đăng xuất khỏi tài khoản sinh viên | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Tài khoản tồn tại trong hệ thống | |
| * **Hậu điều kiện (Post-condition):** Đã đăng nhập vào được hệ thống | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn đăng xuất | 1. Chuyển về trang đăng nhập |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |

* Activity

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa biểu đồ, ảnh chụp màn hình, văn bản, hàng

Mô tả được tạo tự động

#### UC04

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Xem lịch học theo tuần | |
| * **Mô tả sơ lược**: Xem lịch học theo tuần của sinh viên | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Tài khoản tồn tại trong hệ thống, đã đăng nhập | |
| * **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn lịch học | 1. Hệ thống chuyển đến trang lịch học theo tuần |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa hàng, ảnh chụp màn hình, biểu đồ, văn bản

Mô tả được tạo tự động

#### UC05

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Xem lịch thi | |
| * **Mô tả sơ lược**: Xem lịch thi của sinh viên | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Tài khoản tồn tại trong hệ thống, đã đăng nhập | |
| * **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn lịch thi | 1. Hệ thống chuyển đến trang lịch thi |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, biểu đồ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

#### UC06

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Đăng ký học phần | |
| * **Mô tả sơ lược**: Sinh viên đăng ký học phần | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Tài khoản tồn tại trong hệ thống, đã đăng nhập | |
| * **Hậu điều kiện (Post-condition):** Đăng ký thành công học phần | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn đăng ký học phần | 1. Hệ thống chuyển đến trang đăng ký học phần |
| 1. Người dùng chọn học kỳ đăng ký |  |
| 1. Người dùng chọn học phần đăng ký, chọn lớp thực hành |  |
| 1. Nhấn đăng ký môn học | 1. Hệ thống kiểm tra |
|  | 1. Thông báo đăng ký học phần thành công |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 7.1 Hệ thống báo quá số lượng đăng ký |
| 7.2 Quay lại bước 4 |  |
|  | 7.1 Hệ thống thông báo quá số tín chỉ học kỳ |
| 7.2 Quay lại bước 4 |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |

* Activity

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, biểu đồ, hàng, Song song

Mô tả được tạo tự động

#### UC07

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Hủy học phần | |
| * **Mô tả sơ lược**: User có thể hủy học phần mình đã đăng ký khi học phần chưa đổi trạng thái sang chấp nhận mở lớp | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Học phần được xóa ra khỏi danh sách học phần của user | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn hủy học phần | 1. Hiển thị thông báo chấp nhận hủy học phần |
| 1. Chọn hủy | 1. Xóa học phần khỏi dsHocPhan của user, hiển thị thông báo |
| 1. Chọn Ok | 1. Hiển thị lại trang học phần. |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| * 1. Chọn đóng | * 1. Chuyển đến 6. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, hàng, biểu đồ, Song song

Mô tả được tạo tự động

#### UC08

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Xem học phần đã đăng ký | |
| * **Mô tả sơ lược**: User có thể xem tất cả những học phần mình đã từng đăng ký | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Không | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn xem học phần đã đăng ký | 1. Hiển thị giao diện học phần đã đăng ký |
| 1. Xem thông tin những học phần mà user đã đăng ký. |  |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| * 1. Xem thông tin những học phần đã đăng ký theo yêu cầu. | * 1. Hiển thị thông tin những học phần đã đăng ký. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, ảnh chụp màn hình, Song song, hàng

Mô tả được tạo tự động

#### UC09

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Tra cứu công nợ | |
| * **Mô tả sơ lược**: User có thể xem tất cả những công nợ đã qua hay những công cần nợ cần thanh toán cho nhà trường từ lúc nhập học đến khi kết thúc . | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Không | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn tra cứu công nợ | 1. Hiển thị giao diện tra cứu công nợ |
| 1. Xem thông tin công nợ. |  |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| * 1. Tra cứu công nợ theo yêu cầu. | * 1. Hiển thị thông tin công nợ. |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, ảnh chụp màn hình, Song song, số

Mô tả được tạo tự động

#### UC10

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Xem kết quả học tập | |
| * **Mô tả sơ lược**: User có thể xem tất cả các kết quả của từng môn học mình đã tham gia trên hệ thống nhà trường | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Không | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn xem Kết quả học tập | 1. Hiển thị giao diện kết quả học tập |
| 1. Xem thông tin kết quả học tập. |  |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, hàng, biểu đồ, Sơ đồ

Mô tả được tạo tự động

#### UC11

* Đặc tả

|  |  |
| --- | --- |
| * **Tên use case**: Xem thông báo nhắc nhở | |
| * **Mô tả sơ lược**: User có thể xem tất cả các thông báo mới hay cũ mà nhà trường cần truyền đạt đến sinh viên. | |
| * **Actor chính**: User | |
| * **Actor phụ**: Không | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Đổi trạng thái thông báo sang đã xem | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn xem Nhắc nhở | 1. Hiển thị giao diện Nhắc nhở |
| 1. Xem thông tin kết thông báo hoặc nhắc nhở nếu có. | 1. Cập nhật trạng thái thông báo |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

* Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

* Sequence

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

## Class diagram

A screenshot of a computer

Description automatically generated

## Deployment diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Mô tả được tạo tự động: Kiến trúc phần mềm

**(Thiết kế tài liệu theo mẫu arc42)**

## Giới thiệu và mục đích

### Giới thiệu

Hệ thống đăng ký học phần của sinh viên theo theo cơ chế tín chỉ. Các sinh viên sẽ được biên chế ở các khoa theo mỗi ngành học của mình đăng ký.

Trong mỗi học kỳ, nhà trường sẽ chọn một số môn học của từng ngành học cho sinh viên đăng ký học. Sinh viên đăng ký theo học vào một lớp nhất định (lớp tín chỉ) với số sinh viên được đăng ký tối đa được qui định cho từng lớp.

### Mục tiêu

- Tối ưu hóa quá trình đăng ký học phần

- Quản lý thông tin học tập

- Quản lý thông tin sinh viên

## Ràng buộc

- Mỗi học kỳ sinh viên được đăng ký tối đa 30 tín chỉ. Nếu quá số này, hệ thống sẽ không cho đăng ký.

- Sinh viên đăng ký theo học vào một lớp nhất định (lớp tín chỉ) với số sinh viên được đăng ký tối đa được qui định cho từng lớp.

- Trường hợp các lớp đã đầy, sinh viên sẽ được đưa vào một danh sách dự bị để nhà trường cân nhắc có mở thêm lớp hay không. Nếu không mở thêm lớp, sinh viên sẽ bị hủy đăng ký môn đó.

- Sau khi đăng ký thành công, một email thông báo sẽ được gửi cho sinh viên xác nhận việc đăng ký và nhận quyết định đóng học phí.

- Hệ thống cần cho phép sinh viên đăng ký xét tốt nghiệp khi đã đạt đủ số tín chỉ theo yêu cầu của từng ngành học.

- Hệ thống cần cung cấp các chức năng để sinh viên có thể xem thông tin về số tín chỉ đã đạt, số môn đã học, điểm môn học, điểm trung bình tích lũy, và thời khóa biểu.

## Bối cảnh và phạm vi

* Thiết kế Giao diện Thân thiện Người dùng (UI/UX): Giao diện người dùng phải được thiết kế sao cho dễ sử dụng và thân thiện với người dùng cuối, bao gồm cả sinh viên và nhân viên quản lý hệ thống. Các nút chức năng, biểu mẫu và thông tin phải được tổ chức một cách logic để tối ưu trải nghiệm người dùng.
* Đáp ứng (Responsive Design): Giao diện cần có khả năng tự điều chỉnh và thích nghi với kích thước màn hình khác nhau, từ màn hình máy tính lớn đến màn hình điện thoại di động nhỏ.
* Bảo mật Dữ liệu (Data Security): Giao diện cần có các biện pháp bảo mật mạnh mẽ để đảm bảo an toàn cho dữ liệu cá nhân của sinh viên và thông tin quan trọng của hệ thống.
* Hiệu suất và Tốc độ (Performance and Speed): Giao diện phải được tối ưu hóa để tăng hiệu suất và tốc độ tải trang, giúp người dùng trải nghiệm mượt mà và không gặp trở ngại.
* Tích hợp Hệ thống và Dịch vụ Bên Ngoài (Integration with External Systems and Services): Giao diện cần có khả năng tích hợp với các hệ thống và dịch vụ bên ngoài khác chẳng hạn email.
* Tương thích Duyệt Web (Browser Compatibility): Giao diện phải hoạt động tốt trên các trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, Safari và Microsoft Edge để đảm bảo mọi người dùng đều có thể truy cập mà không gặp vấn đề.

## Chiến lược giải pháp

* Sử dụng các công nghệ như React.js xây dựng giao diện người dùng tương tác.
* Sử dụng Node.js để xây dựng các API backend để giao tiếp với Presentation Layer.
* Sử dụng MongoDB để lưu trữ dữ liệu, vì MongoDB phù hợp với dữ liệu không cần quan hệ và có thể mở rộng dễ dàng.
* Sử dụng MongoDB Node.js Driver để kết nối và thao tác với cơ sở dữ liệu MongoDB.
* Sử dụng nodemailer để gửi email từ hệ thống.
* Sử dụng thuật toán hashcode (ví dụ: bcrypt) để bảo mật mật khẩu của người dùng trong cơ sở dữ liệu.
* Sử dụng Git để quản lý mã nguồn của dự án, bao gồm việc theo dõi các phiên bản, quản lý nhánh và hợp nhất mã nguồn.

## Quyết định kiến trúc

### Mô hình

#### Lý do chọn

Dựa vào các yêu cầu chức năng lẫn phi chức năng và nghiệp vụ cơ bản của hệ thống được giải quyết từ 4.1 - 4.5 và Mô hình Layered Architecture có thể đáp ứng tốt các yêu cầu và tình huống cụ thể trên bằng cách chia ứng dụng thành các lớp giao diện người dùng, logic và dữ liệu. Điều này giúp tách biệt các phần của hệ thống và dễ dàng quản lý, mở rộng và bảo trì. Vi vậy chọn mô hình kiến trúc 3-layer architecture cho việc hiện thực ứng dụng:

**Presentation Layer:**

* Đảm nhiệm giao diện người dùng và tương tác với sinh viên.
* Giao diện thân thiện: Phải có giao diện dễ sử dụng và hiệu quả để sinh viên có thể đăng ký môn học một cách nhanh chóng và dễ dàng.

**Business Layer:**

* Xử lý logic nghiệp vụ, bao gồm kiểm tra số lượng sinh viên đăng ký và quản lý danh sách chờ.
* Xử lý đồng thời nhiều yêu cầu: Hệ thống cần xử lý đồng thời hàng ngàn yêu cầu đăng ký môn học từ các sinh viên trong thời gian ngắn.
* Tính nhất quán và đáng tin cậy: Đảm bảo tính nhất quán của dữ liệu và hoạt động của hệ thống, tránh trường hợp trùng lặp đăng ký hoặc mất dữ liệu.
* Quản lý đăng ký và danh sách dự bị: Có khả năng quản lý danh sách dự bị khi lớp học đầy và quyết định mở thêm lớp dựa trên nhu cầu của sinh viên.

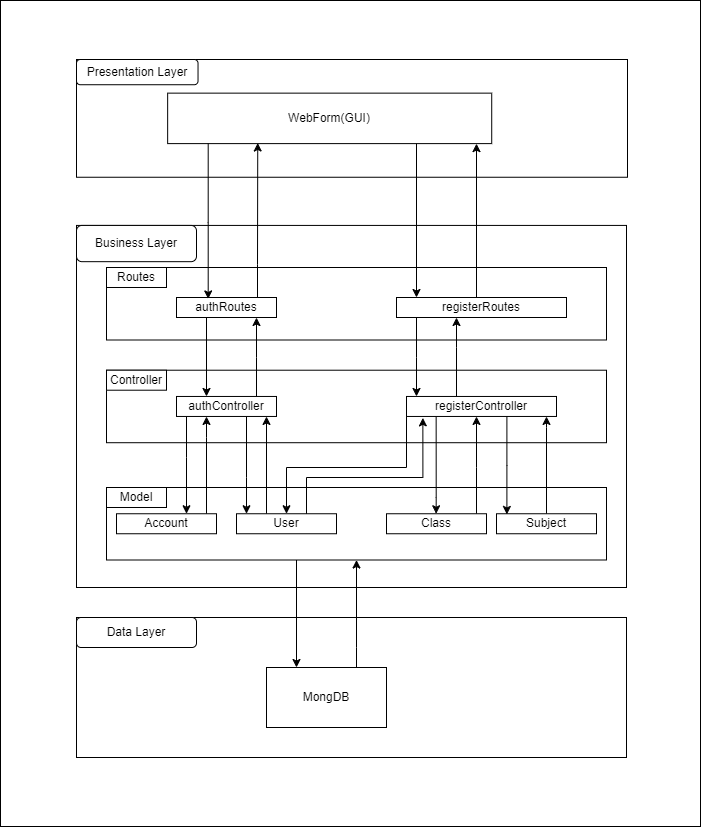
**Data Layer:**

* Lưu trữ thông tin về số lượng sinh viên đăng ký và các thông tin liên quan.
* Hiệu suất cao: Dữ liệu phải được lưu trữ và truy xuất một cách nhanh chóng để đảm bảo trải nghiệm người dùng tốt.
* Bảo mật dữ liệu: Bảo vệ thông tin cá nhân của sinh viên và các dữ liệu quan trọng khác trên cơ sở dữ liệu.
* Dự phòng và phục hồi: Có hệ thống dự phòng và phục hồi dữ liệu để đảm bảo không mất dữ liệu quan trọng trong trường hợp sự cố xảy ra.

**Tổng quan:**

* Khả năng mở rộng: Hệ thống cần có khả năng mở rộng để có thể chứa đựng và xử lý số lượng lớn dữ liệu và người dùng.
* Bảo trì và nâng cấp dễ dàng: Mỗi tầng cần được thiết kế một cách linh hoạt để dễ dàng bảo trì và nâng cấp khi cần thiết.

#### Mô hình



### Ưu điểm/ nhược điểm

#### Ưu điểm

* Tổ chức cấu trúc rõ ràng: Mô hình này chia ứng dụng thành các lớp độc lập, mỗi lớp đảm nhận một chức năng cụ thể, từ đó giúp quản lý dễ dàng và hiểu được cấu trúc của hệ thống.
* Dễ dàng mở rộng: Các lớp có thể được thêm vào hoặc thay đổi mà không ảnh hưởng đến các phần khác của hệ thống, giúp dễ dàng mở rộng hệ thống theo yêu cầu mới.
* Tính tái sử dụng cao: Các lớp có thể được tái sử dụng trong các phần khác của ứng dụng hoặc thậm chí trong các dự án khác, giảm thiểu công sức phát triển và bảo trì.
* Dễ bảo trì và kiểm thử: Do mỗi lớp độc lập với các lớp khác, việc bảo trì và kiểm thử trở nên dễ dàng hơn.
* Hiệu suất cao: Cấu trúc lớp giúp tối ưu hóa hiệu suất vì mỗi lớp chỉ chịu trách nhiệm cho một phần nhất định của logic.

#### Nhược điểm

* Khả năng chồng chéo (Overlap): Có thể xảy ra tình trạng một số logic hoặc tính năng được triển khai ở nhiều lớp, dẫn đến việc phức tạp hóa mã nguồn và khó khăn trong việc bảo trì.
* Khó khăn trong quản lý các phụ thuộc: Các lớp có thể phụ thuộc lẫn nhau một cách phức tạp, đặc biệt khi có các yêu cầu động và thay đổi thường xuyên.
* Khả năng không linh hoạt: Cấu trúc lớp có thể làm hạn chế sự linh hoạt khi cần thay đổi logic hoặc tính năng một cách nhanh chóng.
* Phức tạp trong triển khai ban đầu: Việc phân chia hệ thống thành các lớp có thể đòi hỏi thời gian và kỹ năng trong quá trình triển khai ban đầu.
* Chi phí và tài nguyên: Mô hình layer có thể đòi hỏi chi phí và tài nguyên lớn hơn cho việc phát triển và bảo trì hệ thống so với các mô hình đơn giản hơn.

### Quy trình

1. User sử dụng các chức năng ở WebForm( GUI) trong tầng Presentation . Yêu cầu sẽ được gửi đến Routes (authRoutes, registrationROutes) tương ứng trong tầng Business Layer.
2. Routes định tuyến các yêu cầu từ WebForm đến các controller (authController, registrationController) tương ứng .
3. Controller xử lý yêu cầu từ WebForm, xác định hành động cần thực hiện và gọi các phương thức tương ứng đến Model(Account, User, Class, Subject) để thực hiện logic kinh doanh.
4. Model tương tác với cơ sở dữ liệu (MongoDB) tầng Data để truy xuất và cập nhật dữ liệu ứng với yêu cầu từ Controller.
5. Kết quả được trả về từ MongoDB được chuyển lại qua Controller, Routes và WebForm để hiển thị cho người dùng.

## Khái niệm

### Kiến trúc của một hệ thống phần mềm (Software Architecture)

Kiến trúc của một hệ thống phần mềm (Software Architecture) mô tả các thành phần chính của nó, các mối quan hệ của các thành phần và cách chúng tương tác với nhau.

Kiến trúc đóng vai trò như một **bản thiết kế cho một hệ thống** . Nó cung cấp một sự trừu tượng để quản lý độ phức tạp của hệ thống và thiết lập một cơ chế giao tiếp và phối hợp giữa các thành phần.

### Thiết kế phần mềm (Software Design)

Thiết kế phần mềm cung cấp một **kế hoạch thiết kế** mô tả các phần tử của hệ thống, cách chúng phù hợp và làm việc cùng nhau để đáp ứng yêu cầu của hệ thống. Các mục tiêu của việc có một kế hoạch thiết kế như sau:

* Để thương lượng các yêu cầu hệ thống và đặt kỳ vọng với khách hàng, nhân viên tiếp thị và quản lý.
* Hoạt động như một bản thiết kế trong quá trình phát triển.
* Hướng dẫn các tác vụ triển khai, bao gồm thiết kế chi tiết, mã hóa, tích hợp và thử nghiệm.

### Phong cách kiến trúc (Architecture Style)

Phong cách kiến trúc (Architecture Style) đề cập đến cách tổ chức và thiết kế cấu trúc của một hệ thống phần mềm. Các architecture style định nghĩa các mẫu tổ chức và liên kết các thành phần phần mềm để đạt được các mục tiêu nhất định về hiệu suất, bảo trì, mở rộng, và sự linh hoạt.

### Kiến trúc phân tầng (Layered Architecture)

Kiến trúc Layered (hay còn gọi là kiến trúc lớp) là một mô hình tổ chức các thành phần phần mềm thành các tầng (hoặc lớp) khác nhau, mỗi tầng thực hiện một phần của logic hoặc chức năng cụ thể. Mỗi tầng chỉ tương tác với các tầng lân cận trực tiếp bên dưới và bên trên của nó, giúp tăng tính cô lập và sự độc lập giữa các thành phần, dễ dàng quản lý và bảo trì hệ thống.

Kiến trúc Layered giúp tách biệt các phần của hệ thống và tạo điều kiện cho việc phát triển song song, cũng như cho phép các tầng khác nhau được phát triển bởi các nhóm làm việc độc lập. Tuy nhiên, nếu không thiết kế và quản lý tốt, kiến trúc này có thể dẫn đến sự phức tạp không cần thiết và hiệu suất không tốt.

## Tình huống/ yêu cầu

Tình huống Đăng Ký Môn Học:

* Yêu cầu: Kiểm tra các môn tiên quyết. Kiểm tra số tín chỉ đã đăng ký có vượt quá 30 hay không. Cung cấp thông báo khi lớp đã đầy và thêm vào danh sách chờ.
* Xử lý:
* Lớp Business Logic Layer sẽ thực hiện kiểm tra và cập nhật danh sách chờ.
* Lớp Presentation Layer sẽ cập nhật giao diện tương ứng với thông báo và trạng thái của lớp học.

Tình huống Xác Nhận Đăng Ký:

* Yêu cầu: Yêu cầu sinh viên xác nhận đăng ký. Gửi email thông báo sau khi đăng ký thành công.
* Xử lý:
* Lớp Business Logic Layer sẽ thực hiện xác nhận đăng ký.
* Lớp Presentation Layer sẽ gửi yêu cầu xác nhận qua giao diện và lớp Data Access Layer sẽ cập nhật trạng thái đăng ký trong cơ sở dữ liệu.
* Lớp Presentation Layer sẽ sử dụng dịch vụ gửi email (SMTP, SendGrid) để gửi thông báo cho sinh viên.

Tình huống Quản Lý Thông Tin Học Tập:

* Yêu cầu: Cung cấp thông tin học tập chi tiết và điểm số cho sinh viên.
* Xử lý:
* Lớp Data Access Layer sẽ truy xuất dữ liệu học tập từ cơ sở dữ liệu.
* Lớp Business Logic Layer sẽ xử lý và tính toán các thông tin cần thiết (điểm trung bình, số tín chỉ đạt được).
* Lớp Presentation Layer sẽ hiển thị thông tin này trên giao diện người dùng.

Tình huống Xét Tốt Nghiệp:

* Yêu cầu: Kiểm tra các tiêu chí tốt nghiệp. Cấp bằng và cập nhật trạng thái sinh viên thành cựu sinh viên.
* Xử lý:
* Lớp Business Logic Layer sẽ thực hiện kiểm tra các tiêu chí tốt nghiệp và xử lý yêu cầu đăng ký xét tốt nghiệp.
* Lớp Data Access Layer sẽ cập nhật trạng thái và thông tin của sinh viên trong cơ sở dữ liệu.
* Lớp Presentation Layer sẽ cung cấp thông báo và cập nhật trạng thái trên giao diện người dùng. Với kiến trúc phân lớp như trên, hệ thống sẽ đảm bảo tính linh hoạt, dễ dàng bảo trì và mở rộng trong tương lai, đáp ứng đầy đủ các yêu cầu nghiệp vụ của nhà trường.

## Lợi ích/ đánh đổi

### Presentation Layer:

Đánh đổi:

* Đa dạng giao diện và tính năng: Thiết kế giao diện đa dạng và tính năng phong phú có thể tạo ra một giao diện người dùng phức tạp, đòi hỏi nhiều công việc phát triển và kiểm thử.

Lợi ích:

* Trải nghiệm người dùng tốt hơn: Giao diện đa dạng và tính năng phong phú cung cấp trải nghiệm người dùng tốt hơn, giúp sinh viên và nhân viên dễ dàng tương tác với hệ thống.
* Tương thích với nhiều thiết bị: Sự linh hoạt trong thiết kế giao diện giúp hệ thống tương thích trên nhiều thiết bị khác nhau, từ máy tính đến điện thoại di động.

### Business Layer:

Đánh đổi:

* Độ phức tạp của Business logic: Các chức năng quản lý học phần phức tạp có thể tạo ra một lớp kinh doanh phức tạp và khó bảo trì.

Lợi ích:

* Quản lý học phần hiệu quả: Business logic chính xác và mạnh mẽ giúp quản lý học phần hiệu quả, từ đăng ký môn học đến quản lý điểm số và thời khóa biểu.
* Tính nhất quán của dữ liệu: Business logic đảm bảo tính nhất quán và toàn vẹn của dữ liệu trong toàn bộ hệ thống.

### Data Layer:

Đánh đổi:

* Chi phí và khó khăn khi mở rộng: Quản lý và mở rộng cơ sở dữ liệu có thể đòi hỏi nhiều chi phí và công sức.

Lợi ích:

* Tính nhất quán và toàn vẹn của dữ liệu: Cơ sở dữ liệu đảm bảo tính nhất quán và toàn vẹn của dữ liệu trong toàn bộ hệ thống, đồng thời cung cấp hiệu suất cao cho việc truy vấn dữ liệu.
* Bảo mật dữ liệu: Cơ sở dữ liệu cung cấp bảo mật cho thông tin cá nhân của sinh viên và thông tin quan trọng khác.

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

## Giao diện của hệ thống

### Trang đăng nhập

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

1. Sinh viên nhập mã số sinh viên
2. Sinh viên nhập mật khẩu
3. Sau khi nhập ấn đăng nhập, hệ thống kiểm tra. Nếu đúng sẽ chuyển đến trang chủ.

### Trang chủ

Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, biểu đồ

Mô tả được tạo tự động

* 1. Sinh viên nhấn vào để đến trang đăng ký học phần
  2. Sinh viên nhấn vào để đến trang kết quả học tập
  3. Sinh viên nhấn vào để đến trang lịch học theo tuần
  4. Sinh viên nhấn vào để về trang chủ (nếu ở các trang khác)
  5. Sinh viên nhấn vào để đăng xuất khỏi tài khoản trở về trang đăng nhập
  6. Sinh viên nhấn vào để xem điểm môn các học kì đã đăng ký
  7. Sinh viên nhấn vào để xem các môn học theo học kì đã đăng ký

### Trang đăng ký học phần

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, số, hàng

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Hình chữ nhật, Song song

Mô tả được tạo tự động

* 1. Chọn học kỳ đăng ký
  2. Sinh viện nhấn vào để chọn môn học phần đăng ký
  3. Sinh viên nhấn đăng ký
  4. Sinh viên nhấn x ít để hủy
  5. Sinh viên nhấn x để hủy xác nhận đăng ký
  6. Sinh viên nhấn xác nhận để xác nhận đăng ký học phần

### Trang hủy học phần

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Song song

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Song song, Hình chữ nhật

Mô tả được tạo tự động

1. Sinh viên nhấn để hủy học phần
2. Sinh viên nhấn hủy để học phần
3. Sinh viên nhấn để đóng hủy môn học phần

### Trang lịch học

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, biểu đồ

Mô tả được tạo tự động

1. Sinh viên nhấn để quay về trang chủ
2. Sinh viên nhấn để xem các mục liên quan đến thông tin chung
3. Sinh viên nhấn để xem các mục liên quan đến học tập
4. Sinh viên nhấn để xem các mục liên quan đến đăng ký học phần
5. Sinh viên nhấn để xem các mục liên quan đến học phí

### Trang kết quả học tập

Ảnh có chứa ảnh chụp màn hình, văn bản, hàng, thiết kế

Mô tả được tạo tự động

1. Sinh viên nhấn để yêu cầu tốt nghiệp khi đủ chứng chỉ

## Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

1. Mục đích kiểm thử

* Kiểm tra xử lý các tính năng của hệ thống.
* Đảm bảo tính đúng đắng, ổn định và bảo mật của hệ thống.
* Đảm bảo các chức năng hoạt động chính xác như mong đợi

1. Phạm vi kiểm thử

* Toàn bộ các chức năng được đặc tả:

1. Phương pháp kiểm thử

* Kiểm thử đơn vị, kiểm thử tích hơp và kiểm thử hệ thông.
* Sử dụng các công cụ kiểm thử tự động đa dạng đảm bảo tính khách quan và phủ đủ các trường hợp.

1. Lịch trình kiểm thử

* Xác định lịch trình cụ thể cho việc thực hiện kiểm thử đơn vị, tích hợp và hệ thống, và đảm bảo rằng mọi bước kiểm thử được thực hiện theo kế hoạch.
* Đảm bảo các quá trình kiểm thử tiến hành một cách chặt chẽ.

1. Báo cáo phản hồi

* Ghi lại mọi lỗi được phát hiện trong quá trình kiểm thử, bao gồm cả mô tả chi tiết và bước tái hiện.
* Tạo báo cáo định kỳ về tiến độ kiểm thử, bao gồm số lượng lỗi đã phát hiện, tiến độ hoàn thành và các vấn đề cần được giải quyết.Quản lý kiểm thử
* Báo cáo kết quả cuối cùng của kiểm thử, bao gồm các trường hợp thử nghiệm đã hoàn thành và kết quả của từng bước kiểm thử.

1. Đánh giá và điều chỉnh

* Phân tích kết quả kiểm thử để xác định mức độ đáp ứng của hệ thống đối với các yêu cầu và tiêu chí đã đề ra trong quá trình kiểm thử.
* Đánh giá các lỗi và vấn đề phát hiện được trong quá trình kiểm thử và xác định các biện pháp sửa đổi cần thiết.

1. Kế hoạch kiểm thử lặp lại

* Dựa trên kết quả của kiểm thử ban đầu, xác định các lỗi đã được sửa và các tình huống cần được kiểm thử lại.
* Xác định thời gian và nguồn lực cần thiết để thực hiện kiểm thử lại, bao gồm cả việc xác định các kịch bản kiểm thử cụ thể.

### Kiểm thử hệ thống

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test case** | **Mô tả** | **Thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Ngày kiểm thử** | **Pass/Fail** |
| 1 | Đăng nhập | Đăng nhập vào trang chủ | Nhập mssv: 20044651  Nhập mật khẩu: as | Đăng nhập thành công | Đăng nhập thành công | 15/5/2024 | Pass |
| 2 | Đăng nhập | Đăng nhập vào trang chủ | Nhập mssv: 20044651  Nhập mật khẩu: asaa | Đăng nhập thất bại | Đăng nhập thất bại | 15/5/2024 | Pass |
| 3 | Đăng nhập | Đăng nhập vào trang chủ | Nhập mssv: 20044651  Để trống mật khẩu | Đăng nhập thất bại | Đăng nhập thất bại | 15/5/2024 | Pass |
| 4 | Đăng ký học phần | Đăng ký môn học phần | Chọn học kì 5  Chọn môn Phương pháp luận khoa học  Chọn đăng ký  Nhấn xác nhận | Đăng ký thành công | Đăng ký thành công | 15/5/2024 | Pass |
| 5 | Đăng ký học phần | Đăng ký môn học phần | Chọn học kì 5  Chọn môn Trùng phương pháp luận khoa học  Chọn đăng ký  Nhấn xác nhận | Đăng ký thất bại | Đăng ký thất bại | 15/5/2024 | Pass |
| 6 | Đăng ký học phần | Đăng ký môn học phần | Chọn học kì 5  Chọn môn Trùng phương pháp luận khoa học  Chọn đăng ký  Nhấn hủy | Đăng ký thất bại | Đăng ký thất bại | 15/5/2024 | Pass |
| 7 | Hủy học phần | Hủy học phần đã đăng ký | Chọn hủy với học phần tương ứng  Nhấn hủy | Hủy học phần thành công | Hủy học phần thành công | 15/5/2024 | Pass |
| 8 | Hủy học phần | Hủy học phần đã đăng ký | Chọn hủy với học phần tương ứng  Nhấn đóng | Hủy học phần thất bại | Hủy học phần thất bại | 15/5/2024 | Pass |
| 9 | Đăng xuất | Đăng xuất | Nhấn đăng xuất | Đăng xuất thành công | Đăng xuất thành công | 15/5/2024 | Pass |

# : KẾT LUẬN

## Kết quả đạt được

* Xây dựng hệ thống quản lý người dùng với các chức năng cở bản như đặc tả
* Tạo giao diện người dùng thân thiện và dễ sử dụng.
* Đảm bảo tính bảo mật và quyền riêng tư người dùng (tài khoản).

## Hạn chế của đồ án

* Các chức năng hoàn thành như đặc tả nhưng còn hạn chế nhất định về xử lý
* Gặp phải vài vấn đề hiệu suất khi lượng lớn người dùng.
* Cần cải thiện và điều chỉnh giao diện người dùng hơn dựa vào ý kiến đóng góp người dùng.
* Cần thời gian, tài nguyên, các tính năng, công nghệ mới để theo kịp thị trường.

## Hướng phát triển

* Nâng cáo tính năng và hiệu suát hệ thống đảm bảo được lượng người truy cập đông.
* Phát triển tính năng mới và cải thiện tính năng củ dựa trên đóng góp người dùng.
* Mở rộng trên các nền tảng mới cập khác.
* Liên tục nâng cấp, đảm bảo bảo mật và quyền riêng tư để bảo vệ thông tin người dùng.

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Anh

1. Kevin Faaborg & Sandro Pasquali - Mastering Node.js Second Edition – December 2017
2. David Herron, Diogo Resende & Valentin Bojinov - Node.js Complete Reference Guide – December 2018

Các tài liệu từ Internet

1. <https://nodejs.org/en/docs/>
2. <https://expressjs.com/en/guide/routing.html>
3. [https://radixweb.com/blog/software-architecture-documentation-guide#What](https://radixweb.com/blog/software-architecture-documentation-guide%23What)
4. <https://docs.arc42.org/home/>
5. [https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#References](https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm%23References)