**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC PHENIKAA**



**BÁO CÁO ĐỒ ÁN LIÊN NGÀNH**

**XÂY DỰNG WEBSITE QUẢN LÝ ĐĂNG KÝ VÀ PHÂN CÔNG ĐỒ ÁN CƠ SỞ/ ĐỒ ÁN LIÊN NGÀNH**

**Giảng viên:**  **ThS. Nguyễn Thành Trung**

**Thành viên nhóm:**

**Hoàng Hải Long 20010909**

**Nguyễn Viết Tuấn 20010932**

**HÀ NỘI, THÁNG 11/2023**

# **LỜI CAM KẾT**

Họ và tên nhóm sinh viên:

Hoàng Hải Long

Nguyễn Viết Tuấn

Điện thoại liên lạc:.0967846423.. Email: ..20010909@st.phenikaa-uni.edu.vn..

Lớp: ………..CNTT4…….. …………. Hệ đào tạo: ……....Chính quy….…….

Chúng tôi cam kết Bài tập lớn (BTL) là công trình nghiên cứu của nhóm tôi. Các kết quả nêu trong BTL là trung thực, là thành quả của chúng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong BTL – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Chúng tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm 2023*  Nhóm tác giả BTL  *Hoàng Hải Long*  *Nguyễn Viết Tuấn* |

# **LỜI NÓI ĐẦU**

Để hoàn thành tốt bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến tất cả các Thầy Cô giáo đang giảng dạy tại Trường Đại học Phenikaa đã giảng dạy chúng em trong suốt thời gian qua. Đặc biệt, chúng em xin trân trọng cảm ơn và bày tỏ lòng biết ơn sâu sắc nhất tới thầy **TS.** **Nguyễn Thành Trung** trong quá trình học tập và thực hiện đồ án liên ngành để hoàn thành báo cáo “**Xây dựng ứng dụng Web quản lý đăng ký và phân công đồ án cơ sở/đồ án liên ngành**” thầy đã luôn tạo điều kiện giúp đỡ chúng em hoàn thành báo cáo này.

Chúng em đã cố gắng hoàn thiện bài báo cáo với tất cả sự nỗ lực và cố gắng của cả nhóm. Tuy nhiên, do còn thiếu nhiều kinh nghiệm, chắc chắn bài báo cáo sẽ không tránh khỏi thiếu sót. Vì vậy, chúng em rất mong nhận được sự quan tâm, những ý kiến đóng góp của thầy để bài báo cáo của chúng em có thể hoàn thiện hơn

**Sinh viên thực hiện**

Hoàng Hải Long

Nguyễn Viết Tuấn

**Chúng em xin chân thành cảm ơn !**

**THÔNG TIN CHUNG**

**Tên đề tài:** Xây dựng ứng dụng Web quản lý đăng ký và phân công đồ án cơsở/đồ án liên ngành

**Link Github:** <https://github.com/Nvtuan2002/QuanLy-DangKy-DoAn.git>

**Danh sách thành viên và nhiệm vụ:**

# **BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Người thực hiện** | **Các công việc/đóng góp chính trong BTL** | **Tỷ lệ đóng góp của cá nhân trong toàn bộ project** |
| Hoàng Hải Long | Backend, Design,Thiết kế cơ sở dữ liệu, | 50% |
| Nguyễn Viết Tuấn | Frontend, Kiểm thử | 50% |

# **LỊCH TRÌNH CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Yêu cầu** | **Chi tiết** | **Thời gian** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Đặt Vấn Đề và Nghiên Cứu Ban Đầu | Thu thập thông tin về dự án và yêu cầu của khách hàng. | 2 tuần | 11/09/2023 | 24/9/2023 |
| Xác định phạm vi dự án và mục tiêu chính. |
| Xác định nhóm làm việc và phân công nhiệm vụ. |
| Xây dựng một bản đặc tả yêu cầu sơ bộ. |
| Phân Tích Yêu Cầu Chi Tiết | Xác định yêu cầu chi tiết điều chỉnh bản đặc tả yêu cầu. | 2 tuần | 25/9/2023 | 8/10/2023 |
| Lập kế hoạch cho quá trình phân tích yêu cầu chi tiết. |
| Tiến hành phân tích yêu cầu chi tiết và tạo tài liệu yêu cầu chi tiết. |
| Thiết Kế Hệ Thống | Xác định kiến trúc hệ thống và cơ sở dữ liệu. | 2 tuần | 9/10/2023 | 21/10/2023 |
| Bắt đầu thiết kế giao diện người dùng. |
| Hoàn thiện thiết kế hệ thống và giao diện người dùng. |
| Lên kế hoạch cho giai đoạn lập trình. |
| Lập Trình và Sản Phẩm Cuối Cùng | Bắt đầu quá trình lập trình. | 2 tuần | 23/10/2023 | 4/11/2023 |
| Kiểm tra tích hợp các thành phần của hệ thống |
| Hoàn thiện lập trình và tích hợp hệ thống. |
| Thực hiện kiểm thử và sửa lỗi. |
| Triển khai sản phẩm cuối cùng. |
| Đánh Giá và Báo Cáo |  | 1 tuần | 5/11/2023 | 15/11/2023 |

# **MỤC LỤC**

Contents

[**LỜI CAM KẾT** 2](#_Toc150318044)

[**LỜI NÓI ĐẦU** 3](#_Toc150318045)

[**BẢNG PHÂN CÔNG CÔNG VIỆC** 4](#_Toc150318046)

[**LỊCH TRÌNH CÔNG VIỆC** 5](#_Toc150318047)

[**MỤC LỤC** 7](#_Toc150318048)

[**DANH MỤC HÌNH ẢNH** 9](#_Toc150318049)

[**DANH MỤC BẢNG** 10](#_Toc150318050)

[**DANH SÁCH BIỂU ĐỒ** 11](#_Toc150318051)

[**DANH MỤC CÁC TỪ VIẾT TẮT** 12](#_Toc150318052)

[**DANH MỤC THUẬT NGỮ** 13](#_Toc150318053)

[**Chương 1: GIỚI THIỆU ĐỀ TÀI** 14](#_Toc150318054)

[**1.1 Đặt vấn đề** 14](#_Toc150318055)

[**1.2 Đối tượng và phạm vi dự án** 14](#_Toc150318056)

[**1.3 Định hướng giải pháp** 15](#_Toc150318057)

[**1.4 Bố cục bài tập lớn** 15](#_Toc150318058)

[**CHƯƠNG II: KHẢO SÁT VÀ PHÂN TÍCH NHU CẦU** 17](#_Toc150318059)

[**2.1 Khảo sát hiện trạng** 17](#_Toc150318060)

[**2.2 Tổng quan chức năng** 18](#_Toc150318061)

[**2.2.1 Biểu đồ use case Tổng quan** 18](#_Toc150318062)

[**2.2.2 Đăng Nhập** 20](#_Toc150318063)

[**2.2.3 Đăng ký đồ án** 21](#_Toc150318064)

[**2.2.4 Liên Hệ Giảng Viên** 21](#_Toc150318065)

[**2.2.5 Chỉnh sửa danh sách đăng ký đồ án** 22](#_Toc150318066)

[**2.2.6 Cập nhật đồ án** 22](#_Toc150318067)

[**2.2.7 Cập nhật thông tin cá nhân/ Tiến độ nhóm** 23](#_Toc150318068)

[**2.2.8 Theo dõi tiến trình** 24](#_Toc150318069)

[**2.2.9 Lịch báo cáo** 24](#_Toc150318070)

[**2.3 Đặc tả chức năng** 25](#_Toc150318071)

[**2.3.1 Đăng nhập** 25](#_Toc150318072)

[**2.3.2 Đăng ký đồ án** 26](#_Toc150318073)

[**2.3.3 Liên hệ giảng viên** 27](#_Toc150318074)

[**2.3.4 Chỉnh sửa danh sách đăng ký đồ án** 28](#_Toc150318075)

[**2.3.5 Cập nhật đồ án** 29](#_Toc150318076)

[**2.3.6 Cập nhật thông tin cá nhân/ tiến độ nhóm** 30](#_Toc150318077)

[**2.3.7 Theo dõi tiến trình** 32](#_Toc150318078)

[**2.3.8 Lịch báo cáo** 33](#_Toc150318079)

[**2.4 Yêu cầu phi chức năng** 34](#_Toc150318080)

[**Chương III CÔNG NGHỆ SỬ DỤNG** 35](#_Toc150318081)

[**3.1 FRONT- END** 35](#_Toc150318082)

[**3.2 BACK-END** 36](#_Toc150318083)

[**3.3 DATABASE** 38](#_Toc150318084)

[**3.4 FRAMEWORK** 40](#_Toc150318085)

[**3.4.1 Bootstrap** 40](#_Toc150318086)

[**3.4.2 Laravel** 41](#_Toc150318087)

[**Chương IV: PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG** 45](#_Toc150318088)

[**4.1 Thiết kế tổng quan** 45](#_Toc150318089)

[**4.2 Thiết kế chi tiết** 46](#_Toc150318090)

[**4.2.1 Thiết kế giao diện** 46](#_Toc150318091)

[**4.2.2 Thiết kế cơ sở dữ liệu** 48](#_Toc150318092)

[**4.3 Xây dựng ứng dụng** 48](#_Toc150318093)

[**4.3.1 Thư viện và công cụ sử dụng** 48](#_Toc150318094)

[**4.3.2 Kết quả đạt được** 49](#_Toc150318095)

[**4.4 Kiểm thử** 49](#_Toc150318096)

[**CHƯƠNG V: CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT** 50](#_Toc150318097)

[**CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 51](#_Toc150318098)

[**6.1 Kết luận** 51](#_Toc150318099)

[**6.2 Hướng phát triển** 51](#_Toc150318100)

[**TÀI LIỆU THAM KHẢO** 52](#_Toc150318101)

# **DANH MỤC HÌNH ẢNH**

Hình 1: Front-End

Hình 2: Back-End

Hình 3: MySql

Hình 4: Framework Laravel 10.x

Hình 5: Hoạt động của Web Laravel

Hình 6: Minh họa giao diện trang Đăng nhập

Hình 7: Minh họa giao diện trang chủ Sinh Viên

Hình 8: Minh họa giao diện trang chủ Giảng Viên

# **DANH MỤC BẢNG**

Bảng 1: Danh sách các Actor

Bảng 9:Danh sách thư viện và công cụ sử dụng

# **DANH SÁCH BIỂU ĐỒ**

Biểu đồ 1: use case Tổng quan…………………………………………………….

Biểu đồ 2: use case Đăng nhập………………………………………………….....

Biểu đồ 3: use case Đăng ký đồ án……………………………………………......

Biểu đồ 4: use case Liên hệ giảng viên……………………………………………

Biểu đồ 5: use case Chỉnh sửa danh sách đăng ký đồ án………………………

Biểu đồ 6: use case Cập nhật đồ án………………………………......................

Biểu đồ 7: use case Cập nhật thông tin cá nhân / Tiến độ nhóm……………....

Biểu đồ 8: use case Theo dõi tiến trình…………………………………..............

Biểu đồ 9: use case Lịch báo cáo………………………………………………….

# **DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **RDBMS** | Relational Database Management System  Hệ quản trị cơ sở dữ liệu quan hệ |
| **PHP** | Hypertext Preprocessor  Ngôn ngữ lập trình PHP |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **JS** | JavaScript  Ngôn ngữ lập trình tạo ra các trang web tương tác |
| **CSS** | Cascading Style Sheets  Ngôn ngữ tìm và định dạng, miêu tả lại các phần tử |

# **DANH MỤC THUẬT NGỮ**

|  |  |
| --- | --- |
| **Developer** | Lập trình viên |
| **Cache** | Bộ nhớ đệm |
| **World Wide Web Consortium** | Tổ chức tiêu chuẩn quốc tế |
| **Java** | Ngôn ngữ lập trình hướng đối tượng |
| **Client** | Máy trạm – nơi gửi yêu cầu đến Server |
| **MySQL** | Hệ thống quản trị cơ sở dữ liệu |
| **Actor** | Người hay hệ thống bên ngoài tương tác, trao đổi thông tin với phần mềm |

# **Chương 1: GIỚI THIỆU ĐỀ TÀI**

### **1.1 Đặt vấn đề**

Như chúng ta đã biết hiện nay số lượng sinh viên lựa chọn học đại học đang là rất lớn với con số 660.258 vào năm 2023 vì con số rất lớn nên là dẫn đến việc quản lý và phân công đăng ký đồ án rất là khó có thể dẫn đến nhầm lẫn người thì đăng ký được người thì bị bỏ qua.

Vì vậy nhóm chúng em muốn tìm hiểu Xây dựng Website quản lý đăng ký và phân công đồ án cơ sở/ đồ án liên ngành cho môi trường đại học để mang lại lợi ích và tối ưu quá trình phân công quản lý đồ án. Tính năng đăng ký dự án cho phép sinh viên đơn giản quá trình đăng ký giúp giảng viên dễ dàng quản lý đăng ký từ phía sinh viên đăng ký đồ án. Sinh viên và giảng viên có thể theo dõi danh sách đăng ký, tiến độ, đánh giá và cung cấp phản hồi, tạo cho môi trường học tập hiệu quả hơn.

### **1.2 Đối tượng và phạm vi dự án**

* **Đối tượng**

- Sinh viên: Sinh viên đại học là một trong những đối tượng chính của dự án. Họ sử dụng ứng dụng web để đăng ký dự án, theo dõi tiến độ, cập nhật thông tin, tham gia vào các khóa học đồ án và đánh giá kết quả công việc của họ.

- Giảng viên (Hướng dẫn đồ án): Giảng viên hoặc người hướng dẫn đồ án là những người quan trọng trong quá trình quản lý dự án. Họ sử dụng ứng dụng để duyệt đăng ký đồ án, thực hiện phân công dự án, theo dõi tiến độ, đánh giá hiệu suất của sinh viên và cung cấp hướng dẫn cho họ.

* **Phạm vi dự án**

- Tìm hiểu thu thập thông tin và phân tích sự cần thiết và yêu cầu của khách hàng

- Thiết kế giao diện Người dùng

- Thiết kế giao diện hệ thống Quản lý

- Các chức năng chính:

+ Đăng ký đồ án: Sinh viên có thể đăng ký dự án cơ sở hoặc đồ án liên ngành thông qua giao diện web. Họ cung cấp thông tin về đề tài, giảng viên hướng dẫn, và thành viên trong nhóm nếu có.

+ Quản lý đăng ký: Cung cấp cho giảng viên hoặc người quản lý dự án một giao diện để xem và duyệt các đăng ký đồ án từ sinh viên.

+ Phân công đồ án: Hệ thống tự động phân công sinh viên vào các đề tài dự án dựa trên thông tin đăng ký và sẽ hướng dẫn sinh viên về các bước tiếp theo trong quá trình thực hiện dự án.

+ Theo dõi tiến độ đồ án: Sinh viên có thể cập nhật tiến độ dự án, thêm báo cáo và tài liệu, và ghi chú công việc đã thực hiện thông qua ứng dụng web.

+ Đánh giá và phản hồi: Giảng viên và người quản lý dự án có thể đánh giá tiến độ và hiệu suất của sinh viên trong dự án và cung cấp phản hồi.

### **1.3 Định hướng giải pháp**

**Khảo sát hiện trạng, thị trường xã hội:**

Sự cạnh tranh của các website

Tâm lý, nhu cầu của khách hàng

**Thiết kế xây dựng website:**

Tìm hiểu cách thức xây dựng

Xây dựng phần mềm quản lý đăng ký đồ án

Tìm hiểu ngôn ngữ đáp ứng nhu cầu của website

Tìm hiểu kết nối mở rộng quy mô của website

### **1.4 Bố cục bài tập lớn**

Phần còn lại của báo cáo Bài tập lớn này được tổ chức như sau.

Chương 2 trình bày nội dung về khảo sát và phân tích yêu cầu hệ thống bao gồm khảo sát hiện trạng, tổng quan các chức năng, yêu cầu chức năng và phi chức năng, biểu đồ use case.

Chương 3, nhóm em tìm hiểu về cơ sở lý thuyết, công nghệ sử dụng khi xây dựng website.

Chương 4 sẽ trình bày chi tiết quy trình phát triển và triển khai ứng dụng, cách thức xây dựng.

Chương 5 là các giải pháp và đóng góp nổi bật.

Chương 6, trình bày giải pháp, đưa ra kết luận và hướng phát triển.

# **CHƯƠNG II: KHẢO SÁT VÀ PHÂN TÍCH NHU CẦU**

## **2.1 Khảo sát hiện trạng**

Thực tế, số lượng sinh viên hiện nay mỗi năm mỗi tăng nhiều số lượng sinh viên như năm 2023 có tới 660.000 Học sinh nhập học Đại học. Vì số lượng quá đông nên việc kiểm soát đăng ký xét duyệt môn học rất là phức tạp dẫn đến nhầm lẫn khi đăng ký môn học. Vì vậy ta có:

**Kết quả khảo sát:**

Câu hỏi 1: Bạn có gặp khó khăn khi đăng ký môn học tại trường của mình không?

* Có: 85%
* Không: 15%

Câu hỏi 2: Những khó khăn cụ thể mà bạn gặp phải khi đăng ký môn học là gì?

* Quy trình đăng ký phức tạp, khó hiểu: 55%
* Lịch đăng ký môn học không phù hợp: 25%
* Số lượng môn học đăng ký tối đa hạn chế: 10%
* Thông tin về môn học không đầy đủ, chính xác: 10%

Câu hỏi 3: Bạn đánh giá mức độ hài lòng của mình về quy trình đăng ký môn học tại trường của mình như thế nào?

* Không hài lòng: 60%
* Hài lòng: 30%
* Rất hài lòng: 10%

Kết quả khảo sát cho thấy, hầu hết sinh viên đều gặp khó khăn khi đăng ký môn học tại trường của mình. Những khó khăn cụ thể bao gồm quy trình đăng ký phức tạp, khó hiểu, lịch đăng ký môn học không phù hợp, số lượng môn học đăng ký tối đa hạn chế và thông tin về môn học không đầy đủ, chính xác.

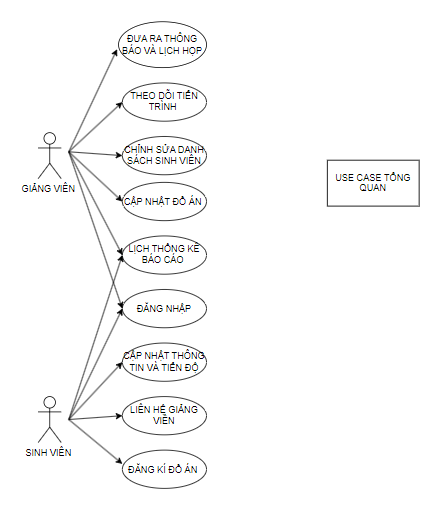
## **2.2 Tổng quan chức năng**

* **Đăng nhập** **(sinh viên / giảng viên)**: Website cung cấp tính năng đăng nhập cho 3 vai trò trên nhằm đáp ứng tính bảo mật và quản lý.
* **Cập nhật thông tin cá nhân/ tiến độ nhóm(sinh viên):** Sinh viên sau khi đăng ký đồ án và được phân vào nhóm hợp lý sinh viên có thể cập nhật thông tin cá nhân và tiến độ báo cáo của nhóm
* **Đăng ký đồ án (sinh viên):** Sinh viên có thể đăng ký các đồ án khác nhau của các giảng viên khác nhau
* **Liên hệ giảng viên( sinh viên):** Mỗi sinh viên hoặc nhóm sinh viên có thể liên hệ với giảng viên để trao đổi các thắc mắc giúp phát triển đồ án
* **Chỉnh sửa danh sách sinh viên đăng kí ( giảng viên):** Khi sinh viên muốn chỉnh sửa về thông tin hay đổi giảng viên hoặc là đổi nhóm giảng viên có thể thực hiện chức năng này
* **Cập nhật đồ án( giảng viên)**: Giảng viên có thể đưa ra những đề tài và những tiêu chí giúp sinh viên có thể dễ dàng chọn lựa
* **Theo dõi tiến trình báo cáo( giảng viên):** Sau khi sinh viên bắt đầu đồ án giảng viên có thể theo dõi tiến độ nhằm đánh giá chất lượng báo cáo cuối kỳ.
* **Lịch thống kê báo cáo( giảng viên/ sinh viên):** Với lịch báo cáo sẽ giúp cho sinh viên và giảng viên có thể nắm bắt dễ dàng lịch báo cáo và được phân công vào nhóm/ giảng viên nào.

### **2.2.1 Biểu đồ use case Tổng quan**

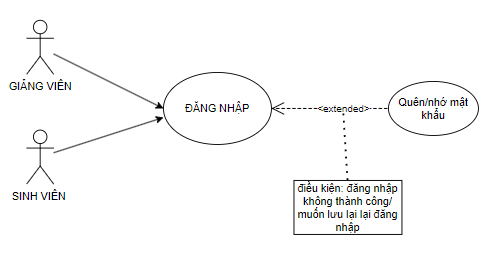
Bảng 1: Danh sách các Actor

|  |  |  |
| --- | --- | --- |
| STT | Actor | Mô tả |
| 1 | Giảng viên | Là người có toàn quyền tương tác với hệ thống, thực hiện các thao tác: Đăng nhập, Chỉnh sửa danh sách đăng ký, Cập nhật đồ án, Theo dõi tiến trình báo cáo, Lịch thống kê báo cáo |
| 2 | Sinh viên | Là người có quyền tương tác với hệ thống, thực hiện các thao tác: đăng nhập, Cập nhật thông tin cá nhân/ Tiến độ nhóm , Đăng ký đồ án , Liên hệ giảng viên, Lịch thống kê báo cáo |



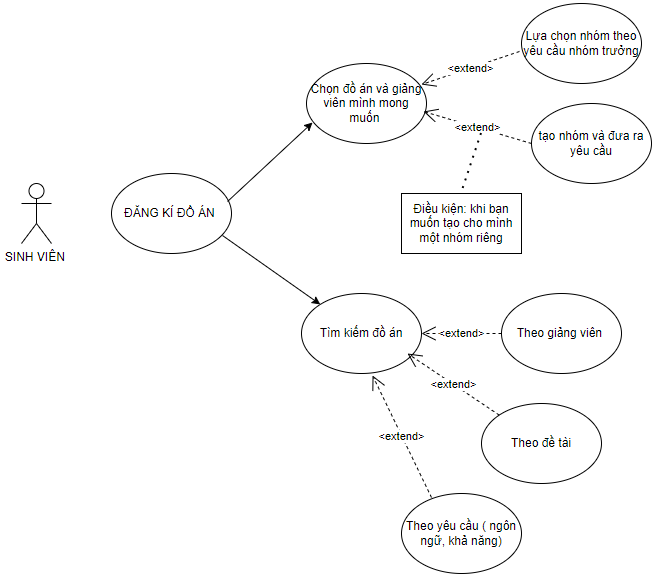
Biểu đồ 1: Use case Tổng quan

### **2.2.2 Đăng Nhập**

****

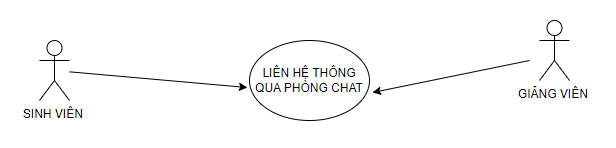
*Biểu đồ 2: use case Đăng nhập*

### **2.2.3 Đăng ký đồ án**

****

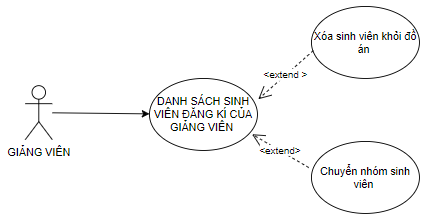
Biểu đồ 3: use case Đăng ký đồ án

### **2.2.4 Liên Hệ Giảng Viên**

****

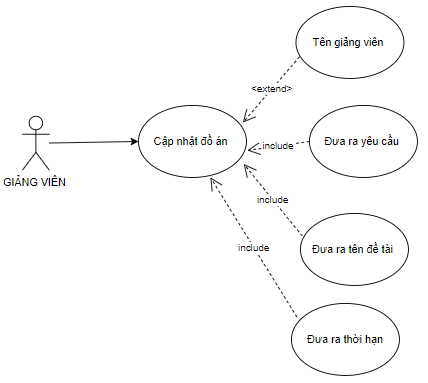
Biểu đồ 4: use case Liên hệ Giảng Viên

### **2.2.5 Chỉnh sửa danh sách đăng ký đồ án**

****

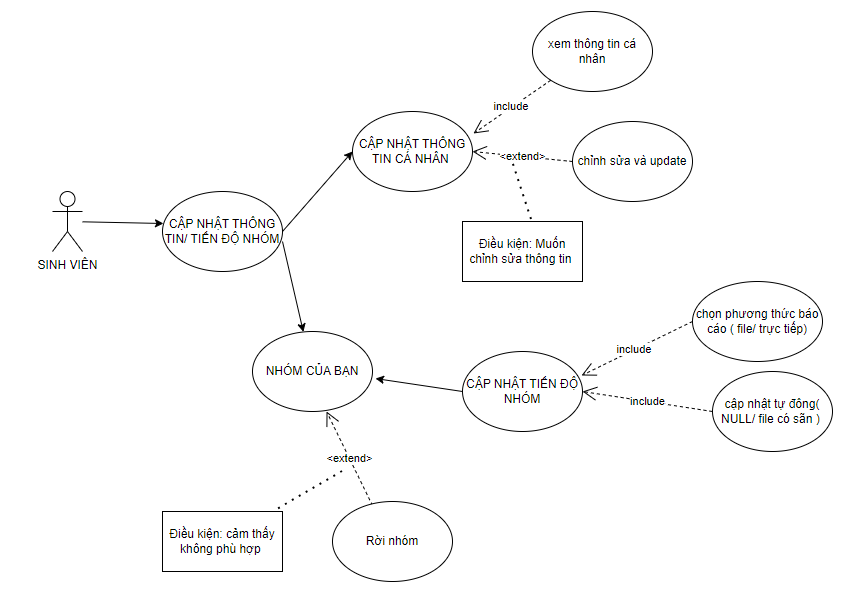
Biểu đồ 5: Chỉnh sửa danh sách đăng ký đồ án

### **2.2.6 Cập nhật đồ án**

****

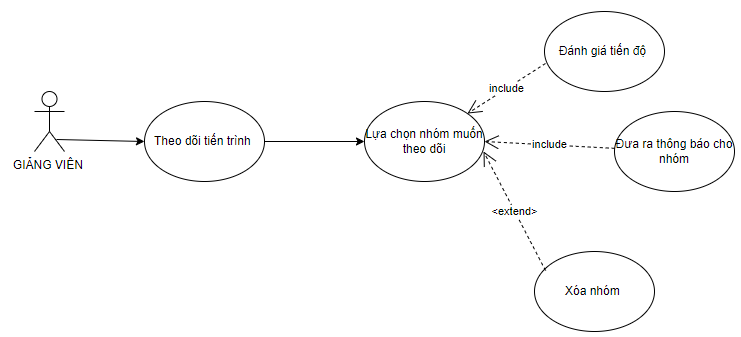
Biểu đồ 6: cập nhật đồ án

### **2.2.7 Cập nhật thông tin cá nhân/ Tiến độ nhóm**

****

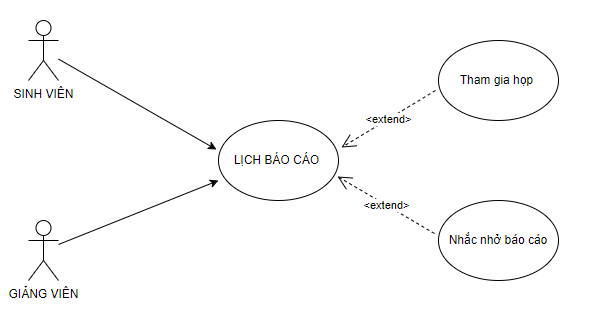
Biểu đồ 7: Cập nhật thông tín cá nhân / Tiến độ nhóm

### **2.2.8 Theo dõi tiến trình**

****

Biểu đồ 8: theo dõi tiến trình

### **2.2.9 Lịch báo cáo**

****

Biểu đồ 9: Lịch báo cáo

## **2.3 Đặc tả chức năng**

### **2.3.1 Đăng nhập**

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng nhập |
| **Mô tả** | Use Case này mô tả quá trình đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu. Nó bao gồm ba tác nhân chính: Giảng viên và Sinh viên. Mục đích chính là cho phép các tác nhân này truy cập vào hệ thống và thực hiện các nhiệm vụ phù hợp với vai trò của họ |
| **Tác nhân** | Giảng viên, Sinh Viên |
| **Tiền điều kiện** | * Tài khoản người dùng đã được tạo sẵn * Tài khoản người dùng đã được phân quyền * Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập |
| **Hậu điều kiện** | Người dùng đã đăng nhập thành công hoặc không thành công. |
| **Luồng cơ bản** | 1. Người dùng truy cập màn hình đăng nhập của hệ thống 2. Người dùng nhập tên đăng nhập và mật khẩu của họ 3. Người dùng nhấn nút "Đăng nhập" 4. Hệ thống kiểm tra tên đăng nhập và mật khẩu 5. Nếu thông tin đăng nhập chính xác, hệ thống cho phép người dùng truy cập vào tài khoản của họ và chuyển tới màn hình chính của vai trò của họ ( Giảng viên, hoặc Sinh viên. |
| **Luồng ngoại lệ** | * Nếu thông tin đăng nhập không chính xác (tên đăng nhập hoặc mật khẩu sai), hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập. * Nếu người dùng quên mật khẩu, họ có thể nhấn vào liên kết "Quên mật khẩu." * Hệ thống chuyển họ tới màn hình khôi phục mật khẩu. * Người dùng cung cấp địa chỉ email đã đăng ký và yêu cầu hệ thống gửi một liên kết để đặt lại mật khẩu. * Hệ thống gửi email chứa mật khẩu mới tới địa chỉ email đã cung cấp. * Người dùng sau đó có thể sử dụng mật khẩu mới để đăng nhập lại vào hệ thống. |

### **2.3.2 Đăng ký đồ án**

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng ký đồ án |
| **Mô tả** | Use Case này mô tả quá trình đăng ký đồ án của Sinh viên trong hệ thống quản lý đăng ký và phân công đồ án cơ sở/đồ án liên ngành. |
| **Tác nhân** | Sinh viên |
| **Tiền điều kiện** | * Sinh viên đã đăng nhập vào hệ thống. * Danh sách đồ án và thông tin liên quan đã được cập nhật trong hệ thống. |
| **Hậu điều kiện** | Sinh viên đã đăng ký cho đồ án và thông tin liên quan đã được cập nhật trong hệ thống. |
| **Luồng cơ bản** | 1. Sinh viên đăng nhập vào hệ thống. 2. Sinh viên truy cập màn hình đăng ký đồ án. 3. Hệ thống hiển thị danh sách các đồ án có sẵn bao gồm các thông tin sau:  * Tên đồ án. * Mô tả đồ án. * Tên giảng viên hướng dẫn (nếu có).  1. Sinh viên chọn một đồ án mà họ muốn đăng ký bằng cách nhấp vào đồ án đó. 2. Hệ thống hiển thị thông tin chi tiết về đồ án, bao gồm mô tả, yêu cầu đặc biệt, và tên giảng viên hướng dẫn (nếu có). 3. Sinh viên nhấp vào nút "Đăng ký" để xác nhận đăng ký cho đồ án. 4. Hệ thống ghi nhận đăng ký của sinh viên cho đồ án. |
| **Luồng ngoại lệ** | * Nếu đồ án đã đạt số lượng sinh viên tối đa hoặc đã đóng đăng ký, hệ thống hiển thị thông báo rằng đăng ký cho đồ án này không còn khả dụng * Nếu Sinh viên đã đăng ký cho đồ án nhưng muốn hủy đăng ký, họ có thể thực hiện thao tác này trên màn hình đăng ký đồ án. Hệ thống sẽ ghi nhận việc hủy đăng ký và cập nhật thông tin liên quan |

### **2.3.3 Liên hệ giảng viên**

|  |  |
| --- | --- |
| **Tên Use Case** | Liên hệ giảng viên |
| **Mô tả** | Use Case này mô tả quá trình Sinh viên liên hệ với giảng viên trong phòng chat hệ thống quản lý đăng ký và phân công đồ án cơ sở/đồ án liên ngành. |
| **Tác nhân** | Giảng viên, Sinh viên |
| **Tiền điều kiện** | * Sinh viên đã đăng nhập vào hệ thống. * Giảng viên của Sinh viên đã được phân cho đồ án của họ. |
| **Hậu điều kiện** | Sinh viên đã gửi yêu cầu liên hệ và giảng viên có thể phản hồi. |
| **Luồng cơ bản** | 1. Sinh viên đăng nhập vào hệ thống. 2. Sinh viên truy cập màn hình liên hệ giảng viên 3. Hệ thống hiển thị danh sách giảng viên liên quan đến đồ án của sinh viên. 4. Sinh viên chọn giảng viên mà họ muốn liên hệ bằng cách nhấp vào tên giảng viên 5. Sinh viên nhập nội dung liên hệ, bao gồm chủ đề và nội dung hỗ trợ 6. Sinh viên nhấn nút "Gửi" để gửi yêu cầu liên hệ tới giảng viên. 7. Hệ thống ghi nhận yêu cầu liên hệ và thông báo cho giảng viên về yêu cầu từ sinh viên. |
| **Luồng ngoại lệ** | Nếu Sinh viên không nhập nội dung liên hệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập đầy đủ chủ đề và nội dung để tiếp tục. |

### **2.3.4 Chỉnh sửa danh sách đăng ký đồ án**

|  |  |
| --- | --- |
| **Tên Use Case** | Chỉnh sửa danh sách đăng ký đồ án |
| **Mô tả** | Use Case này mô tả quá trình cho Giảng viên muốn chỉnh sửa danh sách đăng ký đồ án, bao gồm thêm, sửa, hoặc xóa đăng ký của Sinh viên cho một đồ án cụ thể. |
| **Tác nhân** | Giảng viên |
| **Tiền điều kiện** | * Giảng viên đã đăng nhập vào hệ thống. * Danh sách đăng ký đồ án và thông tin liên quan đã được cập nhật trong hệ thống |
| **Hậu điều kiện** | Danh sách đăng ký đồ án đã được chỉnh sửa và cập nhật trong hệ thống |
| **Luồng cơ bản** | 1. Giảng viên đăng nhập vào hệ thống. 2. Giảng viên truy cập màn hình quản lý danh sách đăng ký đồ án. 3. Hệ thống hiển thị danh sách các đăng ký đồ án hiện có, bao gồm thông tin về Sinh viên, đồ án, trạng thái đăng ký, và các thông tin khác 4. Giảng viên chọn một đăng ký đồ án cụ thể để chỉnh sửa 5. Hệ thống hiển thị thông tin chi tiết về đăng ký đồ án, bao gồm thông tin về Sinh viên, đồ án, và trạng thái đăng ký 6. Giảng viên có thể thực hiện các hoạt động chỉnh sửa như thêm, sửa, hoặc xóa đăng ký đồ án của Sinh viên. 7. Sau khi hoàn thành chỉnh sửa Giảng viên nhấn nút "Lưu" để cập nhật danh sách đăng ký đồ án |
| **Luồng ngoại lệ** | * Nếu Giảng viên thực hiện thao tác không hợp lệ hoặc có lỗi trong quá trình chỉnh sửa, hệ thống hiển thị thông báo lỗi và yêu cầu thực hiện lại thao tác.. * Nếu Giảng viên xóa một đăng ký đồ án, hệ thống sẽ ghi nhận thao tác này và cập nhật danh sách đăng ký đồ án. |

### **2.3.5 Cập nhật đồ án**

|  |  |
| --- | --- |
| **Tên Use Case** | Cập nhật đồ án |
| **Mô tả** | Use Case này mô tả quá trình cho Giảng viên cập nhật thông tin về đồ án, bao gồm đề tài và tiêu chí, để giúp Sinh viên có thể dễ dàng chọn đồ án. |
| **Tác nhân** | Giảng viên |
| **Tiền điều kiện** | * Giảng viên đã đăng nhập vào hệ thống. * Danh sách các đồ án và thông tin liên quan đã được cập nhật trong hệ thống. |
| **Hậu điều kiện** | Thông tin về đồ án đã được cập nhật trong hệ thống và sẵn sàng để Sinh viên xem xét và chọn. |
| **Luồng cơ bản** | 1. Giảng viên đăng nhập vào hệ thống. 2. Giảng viên truy cập màn hình cập nhật thông tin đồ án 3. Hệ thống hiển thị danh sách các đồ án hiện có, bao gồm thông tin về tên đồ án, mô tả đồ án, và tiêu chí đánh giá 4. Giảng viên chọn một đồ án cụ thể để cập nhật 5. Hệ thống hiển thị màn hình cập nhật thông tin về đồ án, cho phép Giảng viên thay đổi thông tin như đề tài, mô tả đồ án, và tiêu chí đánh giá. 6. Sau khi cập nhật thông tin, Giảng viên nhấn nút "Lưu" để cập nhật thông tin đồ án. |
| **Luồng ngoại lệ** | Nếu Giảng viên thực hiện thiếu đầu vào hoặc có lỗi trong quá trình cập nhật, hệ thống hiển thị thông báo lỗi và yêu cầu thực hiện lại thao tác. |

### **2.3.6 Cập nhật thông tin cá nhân/ tiến độ nhóm**

|  |  |
| --- | --- |
| **Tên Use Case** | Cập nhật thông tin cá nhân và tiến độ nhóm |
| **Mô tả** | Use Case này mô tả quá trình cho Sinh viên sau khi đăng ký đồ án và được phân vào một nhóm hợp lý, họ có quyền cập nhật thông tin cá nhân và tiến độ báo cáo của nhóm trong hệ thống. |
| **Tác nhân** | Sinh viên |
| **Tiền điều kiện** | * Sinh viên đã đăng nhập vào hệ thống. * Sinh viên đã đăng ký vào một đồ án và được phân vào một nhóm. |
| **Hậu điều kiện** | Thông tin cá nhân và tiến độ báo cáo của sinh viên và nhóm đã được cập nhật trong hệ thống. |
| **Luồng cơ bản** | 1. Sinh viên đăng nhập vào hệ thống. 2. Sinh viên truy cập màn hình quản lý thông tin cá nhân và tiến độ nhóm 3. Hệ thống hiển thị thông tin cá nhân của sinh viên, bao gồm tên, email, số điện thoại, và các thông tin cá nhân khác. 4. Sinh viên có thể cập nhật các thông tin cá nhân nếu cần thiết, sau đó nhấn nút "Lưu" để cập nhật thông tin 5. Hệ thống hiển thị thông tin về tiến độ báo cáo của nhóm đồ án của sinh viên, bao gồm các tiêu đề báo cáo, trạng thái tiến độ, và thời hạn hoàn thành 6. Sinh viên có thể cập nhật tiến độ của báo cáo, thay đổi trạng thái (đang thực hiện, đã hoàn thành, chưa bắt đầu, v.v.), và cung cấp các thông tin khác liên quan đến tiến độ báo cáo của nhóm 7. Sau khi cập nhật, sinh viên nhấn nút "Lưu" để cập nhật tiến độ báo cáo. |
| **Luồng ngoại lệ** | Nếu Sinh viên cung cấp thông tin cá nhân không hợp lệ hoặc có lỗi, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin hợp lệ. |

### **2.3.7 Theo dõi tiến trình**

|  |  |
| --- | --- |
| **Tên Use Case** | Theo dõi tiến trình báo cáo |
| **Mô tả** | Use Case này mô tả quá trình cho Giảng viên theo dõi tiến độ của báo cáo đồ án của Sinh viên để đánh giá chất lượng báo cáo cuối kỳ. |
| **Tác nhân** | Giảng viên |
| **Tiền điều kiện** | * Giảng viên đã đăng nhập vào hệ thống. * Sinh viên đã đăng ký vào đồ án mà Giảng viên hướng dẫn. * Sinh viên đã bắt đầu thực hiện báo cáo đồ án. |
| **Hậu điều kiện** | Giảng viên đã theo dõi tiến độ của báo cáo đồ án và có thông tin để đánh giá chất lượng. |
| **Luồng cơ bản** | 1. Giảng viên đăng nhập vào hệ thống 2. Giảng viên truy cập màn hình theo dõi tiến trình báo cáo. 3. Hệ thống hiển thị danh sách các Sinh viên đã đăng ký vào đồ án mà Giảng viên hướng dẫn 4. Giảng viên chọn Sinh viên cụ thể để theo dõi tiến trình. 5. Hệ thống hiển thị thông tin về tiến độ báo cáo của Sinh viên, bao gồm tiêu đề báo cáo, trạng thái tiến độ, và thời hạn hoàn thành 6. Giảng viên theo dõi tiến độ báo cáo, kiểm tra các tài liệu trung gian, ghi chú, và bất kỳ thông tin liên quan khác từ Sinh viên. 7. Giảng viên có thể cung cấp phản hồi hoặc hỗ trợ cho Sinh viên dựa trên tiến độ hiện tại và nội dung báo cáo |
| **Luồng ngoại lệ** | * Nếu Sinh viên không cập nhật tiến độ hoặc không cung cấp thông tin cần thiết, Giảng viên có thể thông báo cho Sinh viên về việc cập nhật thông tin hoặc cung cấp thêm thông tin. * Nếu có sự cố hoặc vấn đề về tiến độ báo cáo, Giảng viên có thể theo dõi và cung cấp hỗ trợ hoặc thực hiện các biện pháp cần thiết để giải quyết vấn đề. |

### **2.3.8 Lịch báo cáo**

|  |  |
| --- | --- |
| **Tên Use Case** | Lịch thống kê báo cáo |
| **Mô tả** | Use Case này mô tả quá trình cho Giảng viên và Sinh viên xem và sử dụng lịch thống kê báo cáo để nắm bắt lịch trình báo cáo. |
| **Tác nhân** | Giảng viên, Sinh viên |
| **Tiền điều kiện** | * Giảng viên hoặc Sinh viên đã đăng nhập vào hệ thống. * Danh sách lịch thống kê báo cáo đã được cập nhật trong hệ thống. |
| **Hậu điều kiện** | Giảng viên hoặc Sinh viên đã xem thông tin về lịch thống kê báo cáo |
| **Luồng cơ bản** | 1. Giảng viên hoặc Sinh viên đăng nhập vào hệ thống 2. Giảng viên hoặc Sinh viên truy cập màn hình lịch thống kê báo cáo 3. Hệ thống hiển thị lịch thống kê báo cáo, bao gồm các thông tin sau:  * Thời gian và ngày báo cáo * Đề tài báo cáo * Tên giảng viên hướng dẫn hoặc tên thành viên trong nhóm (nếu có) * Địa điểm báo cáo (nếu có)  1. Giảng viên hoặc Sinh viên có thể xem thông tin về lịch thống kê báo cáo để nắm bắt thời gian và địa điểm báo cáo của họ hoặc của những người khác. |
| **Luồng ngoại lệ** | Nếu không có lịch thống kê báo cáo hoặc lịch chưa được cập nhật, hệ thống có thể thông báo cho Giảng viên hoặc Sinh viên rằng không có thông tin lịch thống kê báo cáo nào có sẵn. |

## **2.4 Yêu cầu phi chức năng**

* Website thao tác dễ hiểu, dễ sử dụng.
* Giao diện dễ dàng, trực quan, gần gũi đối với người sử dụng
* Đảm bảo về bảo mật tuân thủ theo các nguyên tắc mã hóa
* Đảm bảo về hiệu suất website có khả năng truy cập lớn ( >1000 persons).

# **Chương III CÔNG NGHỆ SỬ DỤNG**

## **3.1 FRONT- END**

* **Khái niệm:**

Front-end development là quá trình xây dựng giao diện người dùng (UI) của một trang web hoặc ứng dụng web. Nó bao gồm sử dụng các ngôn ngữ và công cụ như HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), và JavaScript để tạo và tùy chỉnh giao diện mà người dùng thấy và tương tác trên trình duyệt web.



Hình 1: Front-End

* **Ưu điểm:**

Khả năng tương tác cao: Sử dụng JavaScript, front-end developers có thể tạo các tính năng tương tác mạnh mẽ như biểu đồ động, trò chơi, và hình ảnh phản ánh dữ liệu thời gian thực, cải thiện trải nghiệm người dùng.

Độ thân thiện với người dùng: Là phần giao diện của ứng dụng hoặc trang web, front-end đóng vai trò quan trọng trong việc tạo ra giao diện thân thiện và dễ sử dụng, làm tăng sự hấp dẫn và tương tác của người dùng.

Tiết kiệm thời gian và nguồn lực: Sử dụng thư viện và khung làm việc phổ biến như React, Angular, hoặc Vue.js giúp giảm thiểu việc viết mã từ đầu và tăng hiệu suất phát triển.

Tích hợp dễ dàng: Front-end có thể dễ dàng tích hợp với các back-end frameworks và APIs, cho phép ứng dụng hoạt động với dữ liệu và tương tác với máy chủ.

* **Nhược điểm:**

Khả năng bảo mật thấp: Front-end code thường được thực thi trên máy khách, và do đó, có thể bị dễ dàng truy cập và sửa đổi. Điều này đặt ra thách thức về bảo mật cho các ứng dụng quan trọng.

Đa dạng hóa trình duyệt: Mỗi trình duyệt web có cách hiển thị và hỗ trợ tiêu chuẩn HTML, CSS và JavaScript khác nhau. Điều này có thể gây ra các vấn đề tương thích và đòi hỏi kiến thức chi tiết để xử lý chúng.

Khả năng mở rộng khó khăn: Đối với các dự án lớn, quản lý mã nguồn và duy trì sự nhất quán trong front-end codebase có thể trở nên khó khăn.

Phức tạp khi xây dựng ứng dụng lớn: Xây dựng ứng dụng web lớn và phức tạp có thể gây ra sự phức tạp trong việc quản lý codebase, điều này đòi hỏi kỹ năng và kế hoạch phát triển cẩn thận.

Tóm lại, front-end development là một phần quan trọng của quá trình phát triển ứng dụng web và trang web. Nó giúp cải thiện trải nghiệm người dùng và tạo sự tương tác, nhưng cũng đặt ra một số thách thức về bảo mật và tương thích trình duyệt.

## **3.2 BACK-END**

**Khái niệm:**

Back-end development là quá trình phát triển phần mềm và hệ thống xử lý dữ liệu bên trong một ứng dụng web hoặc trang web. Nhiệm vụ chính của back-end là quản lý và xử lý dữ liệu, đảm bảo các tính năng và dịch vụ hoạt động một cách mượt mà và bảo mật.



Hình 2: Back-End Php

**Ưu điểm:**

Quản lý dữ liệu: Back-end là nơi lưu trữ và quản lý dữ liệu của ứng dụng, bao gồm cơ sở dữ liệu, tệp tin, và thông tin khách hàng. Điều này cho phép ứng dụng hoạt động hiệu quả và duyệt qua các giao diện người dùng.

Bảo mật: Back-end là nơi kiểm soát truy cập và xác thực người dùng, giúp đảm bảo an toàn cho dữ liệu và chống lại các tấn công bảo mật như SQL injection.

Tích hợp: Nó cho phép tích hợp ứng dụng với các dịch vụ và hệ thống bên ngoài như thanh toán trực tuyến, hệ thống email, và các API của bên thứ ba.

Hiệu suất: Back-end có thể được tối ưu hóa để đáp ứng nhu cầu của ứng dụng và tối ưu hóa hiệu suất, đảm bảo rằng ứng dụng hoạt động nhanh chóng và ổn định.

**Nhược điểm:**

Phát triển tốn thời gian: Phát triển back-end thường đòi hỏi nhiều thời gian hơn so với phát triển front-end do nó đòi hỏi kiến thức sâu về lập trình và quản lý dữ liệu.

Khó tùy chỉnh: Một số ngôn ngữ back-end, bao gồm PHP, có thể cảm thấy khó để tùy chỉnh và bảo trì nếu không được thiết kế cẩn thận.

Chi phí phát triển: Phát triển và duy trì back-end có thể tốn kém về thời gian và nguồn lực, bao gồm việc thuê nhân viên chuyên nghiệp và cơ sở dữ liệu.

Tương thích hệ thống: Đôi khi, tích hợp các hệ thống cũ hoặc bên thứ ba có thể gây khó khăn về tương thích và đòi hỏi sự phức tạp trong việc quản lý và xử lý dữ liệu.

Tóm lại, back-end development là phần quan trọng của quá trình phát triển ứng dụng web và đóng vai trò quan trọng trong việc quản lý và bảo mật dữ liệu. Tuy nhiên, nó cũng đòi hỏi sự đầu tư thời gian và nguồn lực để phát triển và duy trì một cách hiệu quả.

## **3.3 DATABASE**

**Khái niệm:**

MySQL là hệ thống quản lý cơ sở dữ liệu mã nguồn mở (RDBMS - Relational Database Management System) phổ biến, được sử dụng rộng rãi trên toàn thế giới để lưu trữ và quản lý dữ liệu. phpMyAdmin, trong trường hợp này, là một giao diện người dùng đồ họa dựa trên web cho MySQL, giúp người dùng dễ dàng quản lý cơ sở dữ liệu MySQL thông qua trình duyệt web.



Hình 3: MySQl

**Ưu điểm:**

Tích hợp dễ dàng: MySQL và phpMyAdmin thường được tích hợp dễ dàng với các ứng dụng web và hệ thống khác. Nó cung cấp các giao tiếp API và thư viện cho nhiều ngôn ngữ lập trình, giúp ứng dụng tương tác với cơ sở dữ liệu một cách thuận tiện.

Phổ biến và ổn định: MySQL là một RDBMS phổ biến được sử dụng rộng rãi trong cộng đồng phát triển. Điều này đảm bảo sự ổn định và khả năng tìm kiếm hỗ trợ và tài liệu dễ dàng.

Hiệu suất: MySQL được thiết kế để hoạt động nhanh chóng và hiệu quả, đặc biệt là cho các ứng dụng có dữ liệu lớn. Nó hỗ trợ các công cụ tối ưu hóa và chỉ mục để cải thiện hiệu suất truy vấn.

Bảo mật: MySQL cung cấp các tính năng bảo mật mạnh mẽ, bao gồm quản lý quyền truy cập và mã hóa dữ liệu, giúp bảo vệ cơ sở dữ liệu khỏi các mối đe dọa bảo mật.

**Nhược điểm:**

Phức tạp trong việc quản lý: MySQL có thể phức tạp trong việc quản lý và cấu hình, đặc biệt là đối với người mới sử dụng. Cần có kiến thức về cơ sở dữ liệu để tận dụng hết tiềm năng của nó.

Khả năng mở rộng: Mặc dù MySQL có thể mở rộng theo cách dọc (vertical scaling), tuy nhiên, mở rộng theo cách ngang (horizontal scaling) có thể phức tạp hơn và đòi hỏi kiến thức về việc thiết kế cơ sở dữ liệu phân phối.

Hạn chế trong một số tính năng: MySQL có thể hạn chế trong việc hỗ trợ một số tính năng cao cấp như xử lý dữ liệu không cấu trúc (unstructured data) hoặc dữ liệu đa phương tiện một cách hiệu quả.

Thiếu tính năng mới: Một số tính năng mới và xu hướng trong quản lý cơ sở dữ liệu như NoSQL không được MySQL hỗ trợ một cách tốt.

Tóm lại, MySQL và phpMyAdmin là một cơ sở dữ liệu mạnh mẽ và phổ biến, thích hợp cho nhiều dự án phát triển web và ứng dụng. Tuy nhiên, việc quản lý và mở rộng có thể đòi hỏi kiến thức kỹ thuật và sự hiểu biết về các tính năng và giới hạn của hệ thống này.

## **3.4 FRAMEWORK**

### **3.4.1 Bootstrap**

* **Khái niệm**

Bootstrap là một framework CSS và JavaScript mã nguồn mở được phát triển bởi Mark Otto và Jacob Thornton. Nó được sử dụng để tạo giao diện web đáp ứng, linh hoạt và đẹp mắt. Bootstrap bao gồm các CSS, JavaScript và các tài nguyên khác được thiết kế để giúp các nhà phát triển web tạo giao diện web nhanh chóng và dễ dàng.

* **Ưu điểm**

**Giao diện đáp ứng:** Bootstrap được thiết kế để đáp ứng, có nghĩa là nó sẽ hiển thị tốt trên mọi thiết bị, từ máy tính để bàn đến điện thoại thông minh.

**Linh hoạt**: Bootstrap có thể được sử dụng để tạo nhiều loại giao diện web, từ trang web đơn giản đến trang web phức tạp.

**Tích hợp sẵn**: Bootstrap bao gồm nhiều tính năng tích hợp sẵn, chẳng hạn như thanh menu, thanh trượt và hộp thoại.

**Cộng đồng lớn**: Bootstrap có một cộng đồng lớn và tích cực, cung cấp hỗ trợ và tài nguyên cho các nhà phát triển.

* **Nhược điểm**

**Kích thước lớn**: Bootstrap có thể khá lớn, có thể gây ra vấn đề về hiệu suất trên các trang web có lưu lượng truy cập cao.

**Hạn chế tùy chỉnh**: Bootstrap cung cấp một số tùy chỉnh, nhưng nó có thể không đủ cho các dự án có yêu cầu phức tạp.

### **3.4.2 Laravel**

* **Khái Niệm:**

Laravel là một framework PHP mã nguồn mở và miễn phí, được xây dựng nhằm hỗ trợ phát triển các phần mềm, ứng dụng, theo kiến trúc MVC. Hiện nay, Laravel đang là PHP framework phổ biến nhất và tốt nhất.



Hình 4: FrameWork Laravel 10.x

* **Mô hình MVC (Model-View-Controller)**:

**Controller**: Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng.

**Model**: Là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý…

**View**: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images…

Laravel sử dụng kiến trúc MVC để giúp lập trình viên phát triển các ứng dụng một cách dễ dàng và hiệu qu

* **Ưu điểm:**

Kiến trúc MVC mạnh mẽ: Laravel sử dụng kiến trúc MVC để giúp lập trình viên phát triển các ứng dụng một cách dễ dàng và hiệu quả.

Cơ sở hạ tầng mạnh mẽ: Laravel cung cấp một cơ sở hạ tầng mạnh mẽ để hỗ trợ phát triển các ứng dụng web và API.

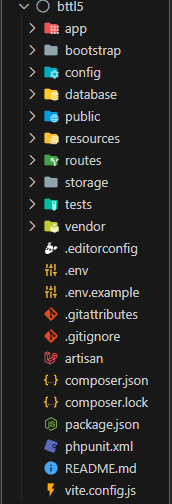
Tính năng bảo mật cao: Laravel cung cấp một số tính năng bảo mật cao để giúp bảo vệ ứng dụng khỏi các cuộc tấn công.

Cộng đồng lớn: Laravel có một cộng đồng lớn và tích cực để hỗ trợ lập trình viên.

* **Nhược điểm:**

Có thể phức tạp đối với người mới bắt đầu: Laravel có một số tính năng và khái niệm phức tạp, có thể khiến người mới bắt đầu gặp khó khăn.

Không có sẵn cho tất cả các nền tảng: Laravel chỉ khả dụng cho nền tảng PHP, do đó nó không thể được sử dụng để phát triển ứng dụng cho các nền tảng khác.

* **Bố cục Folder:**

**vendor**: Nơi lưu các package của hệ thống.

**Controllers**: Chứa các class xử lý các yêu cầu HTTP từ người dùng.

**Middleware**: Chứa các class xử lý điều kiện bắt buộc trước khi đi vào Controller.

**Models**: Chứa các class biểu diễn dữ liệu, truy cập cơ sở dữ liệu, ...

**public**: Chứa các file công khai, mọi người đều có thể truy cập vào bằng địa chỉ web và tên file.

**resources**: Chứa các file gốc trước khi biên dịch và bỏ sang public để sử dụng. Ví dụ: Sass, JS, Lass

**views**: Chứa các file giao diện của web.

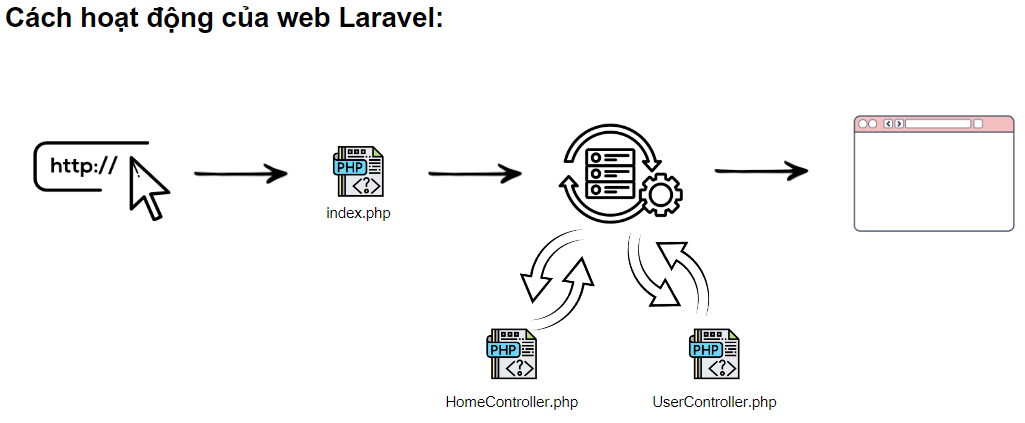
**routes**: Chứa các file định tuyến URL của trang web.

**storage**: Nơi lưu dữ liệu tạm thời, dữ liệu người dùng tải lên, ... và được bảo mật để không bị truy cập file trái phép.

**migrations**: Chứa các class thực hiện khởi tạo, cập nhật, xóa, thao tác với các bảng trong cơ sở dữ liệu, nhằm phiên bản hóa cơ sở dữ liệu.

**factories**: Chứa các class thực hiện tạo dữ liệu mẫu cho trang web, thường được sử dụng khi làm test.

**seeders**: Chứa các class thực hiện thêm dữ liệu ban đầu, tạo dữ liệu mẫu vào cơ sở dữ liệu khi kết hợp với factories và migrations.

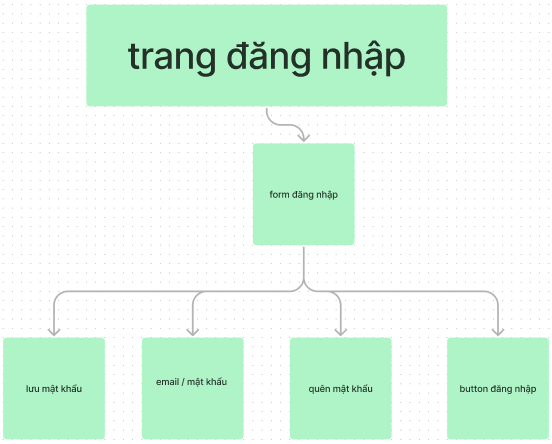


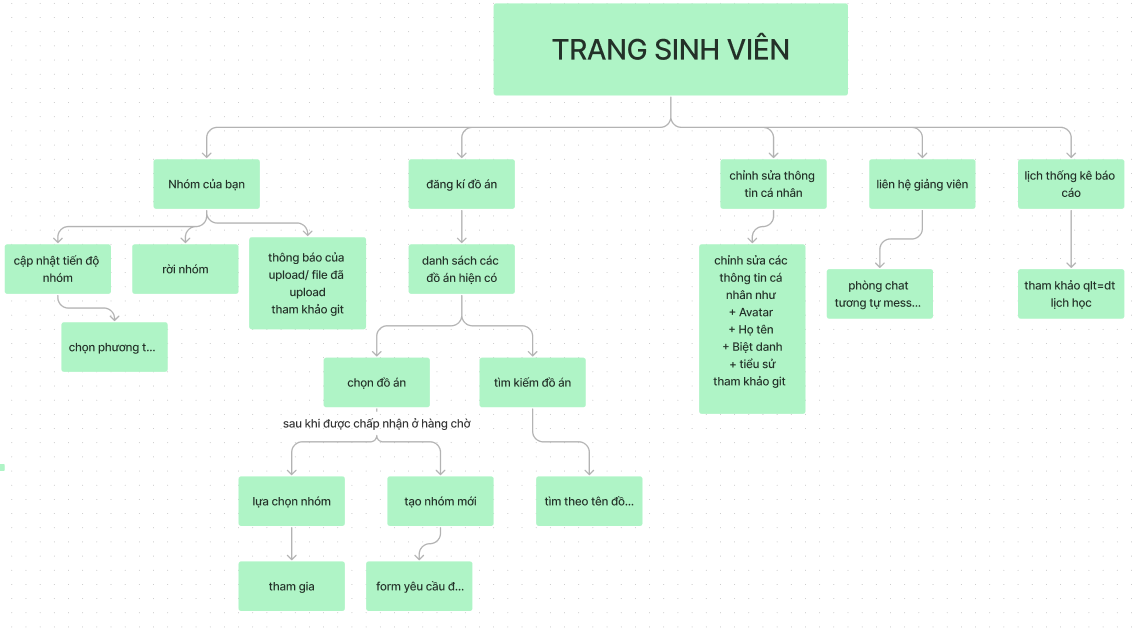
Hình 5: Luồng hoạt động của web Laravel

# 

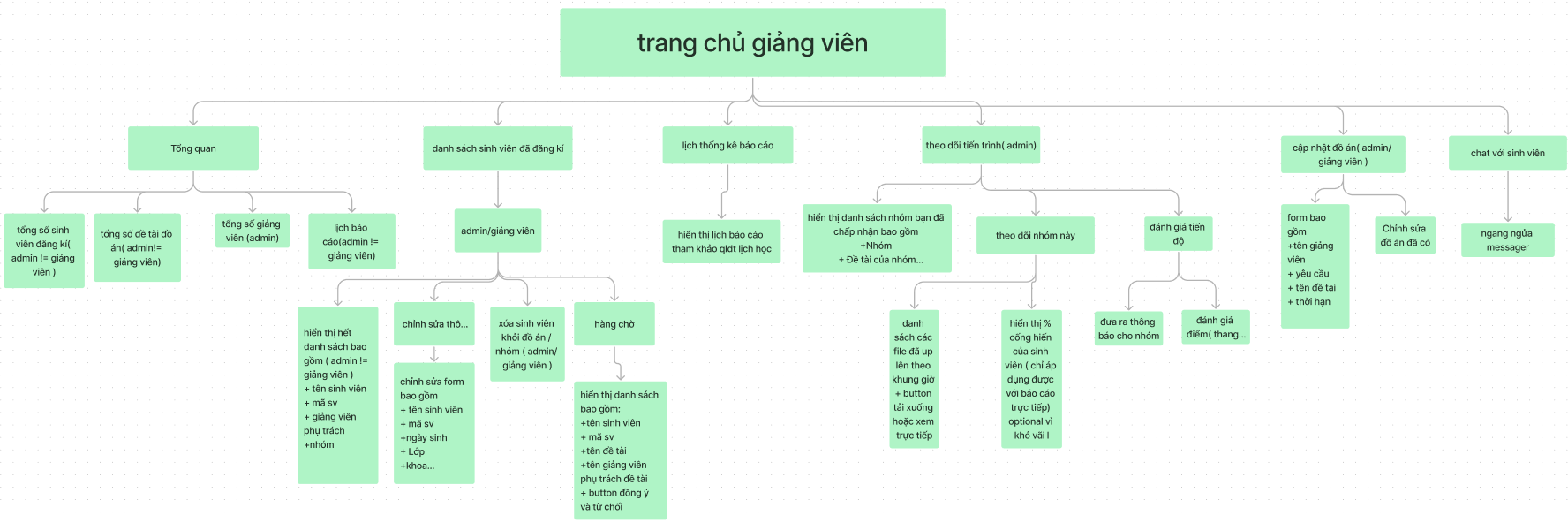
# **Chương IV: PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG**

## **4.1 Thiết kế tổng quan**

****Trang đăng nhập



Trang sinh viên

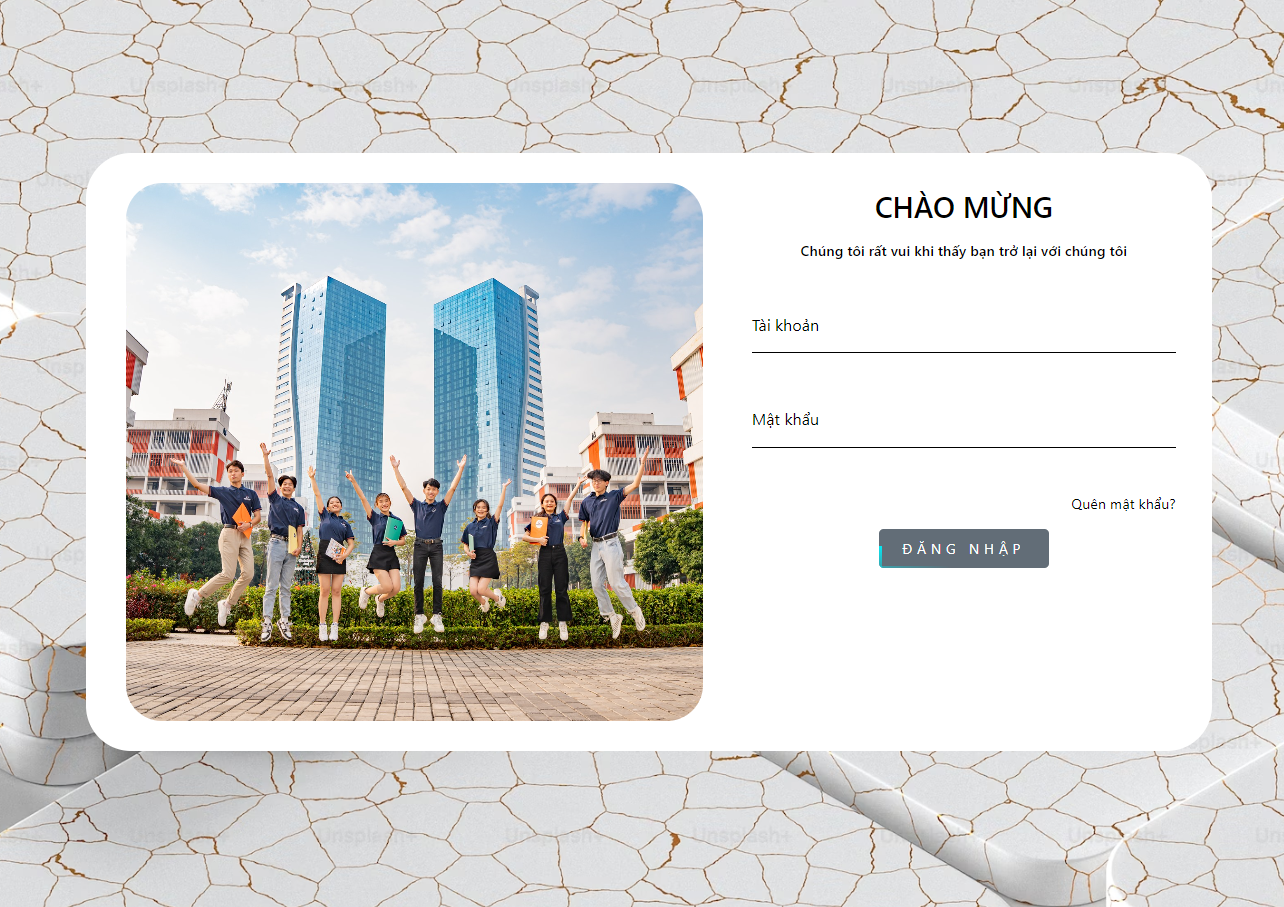


Trang giảng viên

## **4.2 Thiết kế chi tiết**

### **4.2.1 Thiết kế giao diện**

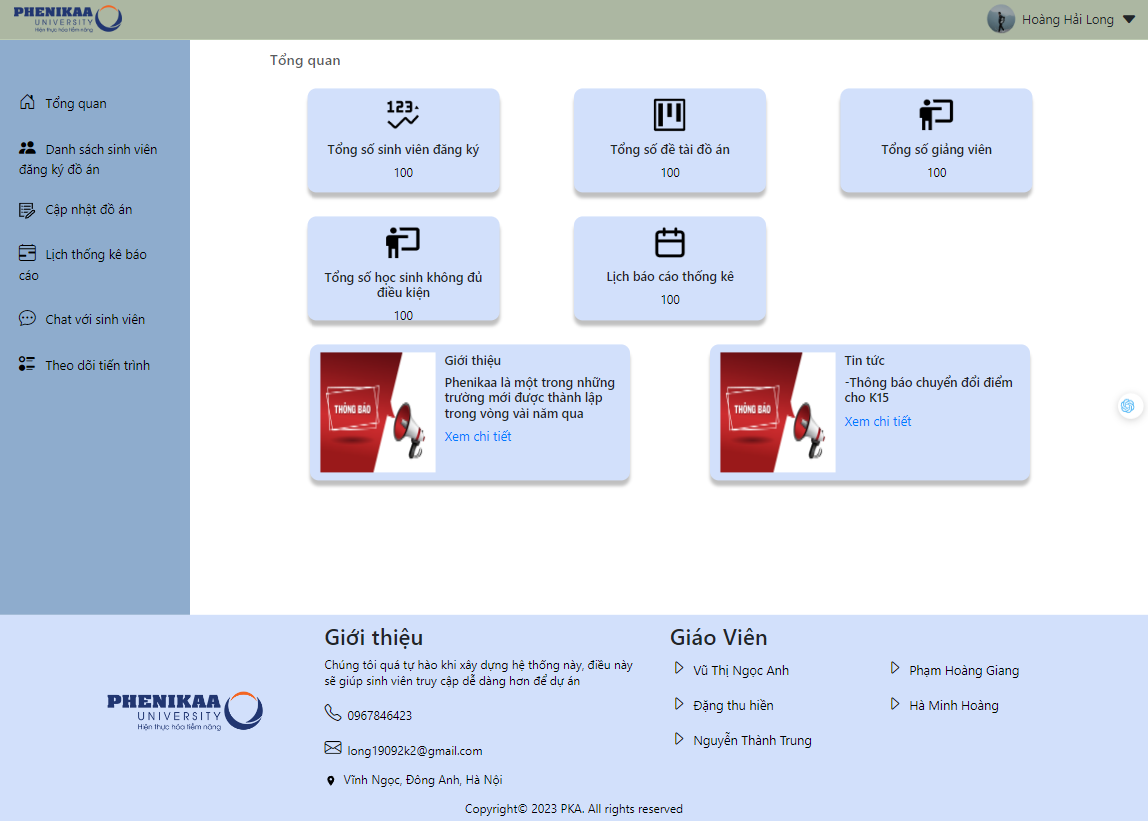
Thiết kế giao diện với Figma: <https://www.figma.com/file/BWrVRdLCWe890umPf3hsDF/wireframe-qu%E1%BA%A3n-l%C3%AD-%C4%91%C4%83ng-k%C3%AD-%C4%91%E1%BB%93-%C3%A1n?type=design&node-id=95%3A2907&mode=dev>



Hình 6. Minh họa giao diện trang Đăng nhập



Hình 7. Minh họa trang chủ giao diện Sinh viên



Hình 8. Minh họa Trang chủ giao diện Giảng viên

### **4.2.2 Thiết kế cơ sở dữ liệu**

Các bảng

## **4.3 Xây dựng ứng dụng**

### **4.3.1 Thư viện và công cụ sử dụng**

**Bảng 9** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Visual Studio Code | <https://code.visualstudio.com/> |
| Chạy localhost | Xampp | <https://www.apachefriends.org/download.html> |
| FrameWork Laravel 10.x | Composer | <https://laravel.com/docs/10.x/installation> |
| Lưu trữ code | Github | <https://github.com/> |
| Thiết kế giao diện | Figma | <https://www.figma.com/> |
| Thiết kế use case | Diagram | <https://app.diagrams.net/> |

### **4.3.2 Kết quả đạt được**

**Ưu điểm:**

* **Có thể dễ dàng giúp sinh viên và giảng viên thuận lợi trong việc đăng kí đồ án**

**Hạn chế:**

* **Còn nhiều chức năng nhỏ cần phát triển thêm để đáp ứng được nhu cầu của người sử dụng**

## 

# **CHƯƠNG V: CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT**

Sau một thời gian tập trung và triển khai, nhóm em đã hoàn thành được đề tài “**Hệ thống quản lý đăng ký và phân công đồ án cơ sở / đồ án liên ngành**” với giao diện thân thiện gần gũi, thao tác sử dụng dễ dàng. Đồng thời, trong suốt quá trình thực hiện đề tài, nhóm em đã có thêm những kỹ năng nền tảng để xây dựng được một website với Framework Laravel 10.x, ngôn ngữ PHP và MySQL, áp dụng được các kiến thức về HTML, CSS và Javascript đã học vào việc phát triển ứng dụng web. Bên cạnh đó, chúng em cũng được mở rộng thêm các kiến thức chuyên ngành mới nhờ việc tìm hiểu một số kiến thức lập trình, nhờ vậy mà kỹ năng tự học, tìm hiểu tài liệu, vận dụng các kỹ năng mới được nâng cao hơn. Những kỹ năng này sẽ là nền tảng, giúp chúng em nâng cao trình độ bản thân, tích lũy được nhiều kinh nghiệm quý giá trong thực tế để có thể làm việc với các doanh nghiệp sau này. Bên cạnh những kết quả khả quan đạt được,

# 

# **CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **6.1 Kết luận**

Để làm được bài tập lớn chúng em đã vận dụng các kiến thức được thầy cô giảng dạy qua các môn học cùng với các tài liệu trên mạng để tìm hiểu thêm. Nhóm cũng đã nắm bắt các quá trình nghiệp vụ của quá trình phân tích thiết kế một **hệ thống quản lý website đăng ký và phân công đồ án cơ sở/ đồ án liên ngành** và có thể nhìn nhận các thành phần của hệ thống một cách chi tiết. Nhóm em đã rút ra một số kinh nghiệm khi xây dựng các biểu đồ biểu diễn hoạt động của hệ thống, có thể hiểu một cách khái quát về cơ sở dữ liệu của hệ thống. Tuy nhiên do trình độ và thời gian hạn chế nhóm chúng em vẫn còn nhiều vấn đề của hệ thống chưa giải quyết được kính mong nhận được sự góp ý của thầy và các bạn. Chúng em xin chân thành cảm ơn!

## **6.2 Hướng phát triển**

Sau khi hoàn thành xây dựng đề tài **“Thiết kế website quản lý đăng ký và phân công đồ án cơ sở / đồ án liên ngành**”, nhóm em sẽ tiếp tục phát triển để đưa website vào ứng dụng thực tế:

· Xây dựng cơ sở dữ liệu phong phú.

· Xây dựng website đáp ứng đầy đủ yêu cầu nghiệp vụ của hệ thống.

## 

# **TÀI LIỆU THAM KHẢO**

1. <https://laravel.com/docs/10.x/installation>
2. <https://topdev.vn/blog/css-la-gi/>
3. <https://bizflycloud.vn/tin-tuc/html-la-gi-20211119162558126.htm>
4. <https://aptechvietnam.com.vn/san-pham-hoc-vien/website-quan-ly-dao-tao/>