

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên : Nguyễn Thị Phụng

Giảng viên hướng dẫn: Ths. Đỗ Văn Tuyên

HẢI PHÒNG – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NGHIÊN CỨU VÀ ĐỀ XUẤT XÂY DỰNG PHẦN MỀM
QUẢN LÝ HÀNH CHÍNH ONLINE DÀNH CHO
SINH VIÊN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên : Nguyễn Thị Phụng

Giảng viên hướng dẫn: Ths. Đỗ Văn Tuyên

HẢI PHÒNG – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Nguyễn Thị Phụng

Mã SV: 2112111039

Lớp : CT2501C

Ngành : Công nghệ thông tin

Tên đề tài: Nghiên cứu và đề xuất xây dựng phần mềm quản lý hành chính online dành cho sinh viên.

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Mô tả tóm tắt đề tài

Đề tài tập trung vào việc nghiên cứu, phân tích và xây dựng một hệ thống website nhằm hỗ trợ công tác quản lý hành chính online dành cho sinh viên tại các trường đại học. Hệ thống cho phép sinh viên thực hiện các thủ tục hành chính như xin xác nhận, nộp biểu mẫu, tra cứu kết quả, và nhận phản hồi một cách nhanh chóng và tiện lợi thông qua môi trường trực tuyến.

b. Nội dung hướng dẫn

- Tìm hiểu hiện trạng quy trình xử lý hồ sơ hành chính cho sinh viên tại các trường đại học.
- Phân tích yêu cầu người dùng và hệ thống.
- Thiết kế cơ sở dữ liệu và kiến trúc hệ thống website.
- Xây dựng và lập trình giao diện người dùng, chức năng đăng nhập, nộp biểu mẫu, thông báo xử lý và phản hồi
- Kiểm thử và đánh giá hệ thống.
- Hoàn thiện tài liệu báo cáo và trình bày kết quả.

c. Kết quả cần đạt được

- Một website hoàn chỉnh phục vụ cho việc quản lý và xử lý các thủ tục hành chính của sinh viên.
- Hệ thống cho phép:
 - o Đăng nhập theo vai trò (sinh viên, cán bộ, quản trị viên),
 - o Nộp và quản lý biểu mẫu hành chính online,
 - o Phản hồi và theo dõi trạng thái xử lý hồ sơ,
 - o Gửi thông báo tự động qua app (đã triển khai cơ bản, lưu thông báo vào cơ sở dữ liệu). Tính năng gửi email đang được phát triển và sẽ hoàn thiện trong giai đoạn tiếp theo.

- Tài liệu báo cáo phân tích thiết kế hệ thống và mô tả chi tiết quá trình xây dựng website.

2. Các tài liệu, số liệu cần thiết

- Tài liệu tham khảo về:
 - o Hệ quản trị cơ sở dữ liệu (MySQL),
 - o Ngôn ngữ lập trình web (PHP, HTML/CSS, JavaScript),
 - o Phân tích và thiết kế hệ thống thông tin.
- Tham khảo từ các phần mềm/hệ thống quản lý hành chính hiện có tại một số trường đại học.
- Yêu cầu thực tế từ sinh viên và cán bộ quản lý hành chính tại nhà trường.

3. Địa điểm thực tập tốt nghiệp

- Trường Đại học Quản lý và Công nghệ Hải Phòng

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : ThS. Đỗ Văn Tuyên

Cơ quan công tác : Trường Đại học Quản lý và Công nghệ Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu hiện trạng quy trình xử lý hồ sơ hành chính cho sinh viên tại các trường đại học.
- Phân tích yêu cầu người dùng và hệ thống.
- Thiết kế cơ sở dữ liệu và kiến trúc hệ thống website.
- Xây dựng và lập trình giao diện người dùng, chức năng đăng nhập, nộp biểu mẫu, thông báo xử lý và phản hồi
- Kiểm thử và đánh giá hệ thống.
- Hoàn thiện tài liệu báo cáo và trình bày kết quả.

Kết quả cần đạt được:

- Một hệ thống quản lý hành chính online hoàn chỉnh, tích hợp các chức năng cần thiết để hỗ trợ sinh viên thực hiện các thủ tục hành chính.
- Giảm thiểu thời gian và công sức của sinh viên khi thực hiện các thủ tục hành chính.
- Nâng cao hiệu quả quản lý và xử lý thủ tục hành chính của các phòng ban trong trường.
- Tài liệu hướng dẫn sử dụng hệ thống và báo cáo kết quả thử nghiệm.

Đề tài tốt nghiệp được giao ngày tháng năm 2025

Yêu cầu phải hoàn thành xong trước ngày tháng năm 2025

Đã nhận nhiệm vụ ĐTTN
Sinh viên

Đã giao nhiệm vụ ĐTTN
Giảng viên hướng dẫn

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Đỗ Văn Tuyên

Đơn vị công tác: Khoa Công Nghệ Thông Tin - Trường Đại học Quản lý và Công nghệ Hải Phòng.

Họ và tên sinh viên: Nguyễn Thị Phụng

Ngành: Công nghệ Thông tin

Nội dung hướng dẫn:

- Tìm hiểu về bài toán
- Thực hiện phân tích thiết kế hệ thống
- Nghiên cứu lựa chọn công nghệ
- Tìm hiểu hệ quản trị cơ sở dữ liệu, ngôn ngữ lập trình web để xây dựng phần mềm quản lý Hành chính trực tuyến dành cho sinh viên tại trường Đại học Quản Lý và Công Nghệ Hải Phòng.
- Nhận xét, đánh giá và kết luận

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

- Sinh viên có ý thức tốt, có tinh thần cố gắng trong quá trình làm đề án tốt nghiệp. Từ việc sưu tập, tổng hợp tài liệu và tìm hiểu bài toán, sinh viên đã vận dụng các kiến thức đã học để phân tích thiết kế hệ thống và xây dựng phần mềm quản lý Hành chính trực tuyến dành cho sinh viên.
- Trong quá trình thực hiện đề án tốt nghiệp, sinh viên luôn cố gắng để đảm bảo đúng tiến độ thực hiện theo quy định của Nhà trường và hướng dẫn của giáo viên hướng dẫn.

2. Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đó đề ra trong nhiệm vụ Đ.T.T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...)

- Đề án tốt nghiệp của sinh viên đã đáp ứng được các yêu cầu của đề cương đề án tốt nghiệp đã đặt ra.
- Phần lý thuyết đã cơ bản đáp ứng được yêu cầu tổng quan kiến thức chung và tìm hiểu chi tiết về bài toán thực tế cần giải quyết.
- Đề xuất cho phép chạy thực tế phần mềm quản lý hành chính trực tuyến trường Đại học Quản Lý và Công Nghệ Hải Phòng.

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Đạt ☒

Không đạt ☐

Điểm: 8.50 (Tám phẩy năm)

Hải Phòng, ngày tháng năm 2025.

Giảng viên hướng dẫn

ThS. Đỗ Văn Tuyên

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: Nguyễn Thị Phương Ngành: Công nghệ thông tin

Đề tài tốt nghiệp: Xây dựng ứng dụng nền tảng web quản lý hành chính online
dành cho sinh viên.

1. Phần nhận xét của giảng viên chấm phản biện

.....
.....
.....
.....
.....
.....

2. Những mặt còn hạn chế

.....
.....
.....
.....
.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm:.....

Hải Phòng, ngày.....tháng năm 2024

Giảng viên chấm phản biện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Với lòng kính trọng và biết ơn chân thành, em xin được gửi lời cảm ơn sâu sắc tới thầy Đỗ Văn Tuyên đã định hướng và hướng dẫn em tận tình trong suốt quá trình nghiên cứu để em hoàn thành được bài tiểu luận tốt nghiệp này.

Em xin chân thành cảm ơn các thầy cô trong Khoa Công nghệ thông tin của Trường Đại học Quản Lý và Công Nghệ Hải Phòng đã hết lòng giúp đỡ, tạo điều kiện cho em được học hỏi và trao dồi kiến thức, kỹ năng để hoàn thành bài tiểu luận tốt nghiệp.

Trong quá trình thực hiện tiểu luận tốt nghiệp, mặc dù đã cố gắng nhưng khó tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp của các thầy cô giáo để bài tiểu luận tốt nghiệp của em thêm hoàn thiện.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2024.

Sinh viên

Nguyễn Thị Phụng

LỜI CAM ĐOAN

Em xin cam đoan công trình nghiên cứu của riêng em và được sự hướng dẫn của thầy Đỗ Văn Tuyên. Các nội dung nghiên cứu và kết quả trong đề tài nay trung thực và chưa công bố dưới bất kì hình thức nào trước đây.

Nếu phát hiện bất kỳ gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung bài đồ án của mình. Trường Đại học Quản Lý và Công Nghệ Hải Phòng không liên quan đến các vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện nếu có.

Hải Phòng, ngày tháng năm 2024.

Sinh viên

Nguyễn Thị Phụng

MỤC LỤC

LỜI CẢM ƠN.....	i
LỜI CAM ĐOAN.....	ii
MỤC LỤC.....	iii
DANH SÁCH HÌNH VẼ.....	v
MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	3
1. Hiện trạng hệ thống quản lý hành chính tại các trường đại học.....	3
2. Phát biểu bài toán	3
3. Giải pháp	3
4. Yêu cầu đạt được của hệ thống	4
5. Giới thiệu các phần mềm sử dụng.....	4
a. ReactJS.....	4
b. Node.js	4
c. Axios	5
d. MySQL.....	5
e. Visual Studio Code	5
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	6
1. Khảo sát hệ thống.....	6
1.1. Cách thức khảo sát:.....	6
1.2. Kết quả:.....	6
2. Phân tích yêu cầu hệ thống.....	6
2.1. Tranh đăng nhập/ đăng ký:	6
2.2. Trang chủ (sinh viên):.....	6
2.3. Trang xử lý hồ sơ (sinh viên):	6
2.4. Trang quản lý (cán bộ).....	6
2.5. Trang quản trị (admin).....	6
3. Quy trình nghiệp vụ.....	7
3.1. Quy trình nghiệp vụ chung	7
3.2. Quy trình nghiệp vụ chi tiết.....	11
4. BIỂU ĐỒ USE CASE	19
4.1. Use Case tổng quát	19
4.2. Use Case chi tiết	21
5. BIỂU ĐỒ TUẦN TỰ	28

5.1. Mô tả tổng quát.....	28
5.2. Biểu đồ tuần tự chi tiết	30
6. BIỂU ĐỒ CỘNG TÁC THỰC THI.....	37
7. BIỂU ĐỒ LỚP	41
7.1. Mô hình lớp “Đăng nhập”	41
7.2. Mô hình lớp “Nộp đơn hành chính”	42
7.3. Mô hình lớp “Theo dõi trạng thái hồ sơ”	46
7.4. Mô hình lớp “Xử lý hồ sơ”	47
7.5. Mô hình lớp “Quản lý người dùng”	49
7.6. Mô hình lớp “Quản lý phòng ban”	51
7.7. Mô hình lớp “Quản lý mẫu biểu”	53
8. BIỂU ĐỒ LỚP THIẾT KẾ UML	57
9. CHUYỂN ĐỔI LỚP THIẾT KẾ UML SANG ER.....	59
10. CƠ SỞ DỮ LIỆU	60
10.1. Bảng users	60
10.2. Bảng departments.....	60
10.3. Bảng formtemplates	60
10.4. Bảng submittedforms	60
CHƯƠNG 3. CÀI ĐẶT CHƯƠNG TRÌNH VÀ THỬ NGHIỆM.....	62
1. Môi trường triển khai	62
1.1. Phần cứng	62
1.2. Phần mềm	62
1.3. Công cụ phát triển.....	62
2. Giao diện chương trình.....	63
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	70
1. Kết luận	70
2. Hướng phát triển.....	71
TÀI LIỆU THAM KHẢO	72

DANH SÁCH HÌNH VẼ

Hình 2. 1: Sơ đồ tiến trình nghiệp vụ chung Sinh viên	8
Hình 2. 2: Sơ đồ tiến trình nghiệp vụ chung Cán bộ (staff)	9
Hình 2. 3: Sơ đồ tiến trình nghiệp vụ chung Admin	10
Hình 2. 4: Sơ đồ Use Case tổng quát.....	19
Hình 2. 5: Biểu đồ tuần tự thực thi “Đăng nhập”	30
Hình 2. 6: Biểu đồ tuần tự thực thi “Nộp đơn”	31
Hình 2. 7: Biểu đồ tuần tự thực thi “Theo dõi trạng thái hồ sơ”	32
Hình 2. 8: Biểu đồ tuần tự thực thi “Xử lý hồ sơ”	33
Hình 2. 9: Biểu đồ tuần tự thực thi “Quản lý tài khoản người dùng”	34
Hình 2. 10: Biểu đồ tuần tự thực thi “Quản lý phòng ban”	35
Hình 2. 11: Biểu đồ tuần tự thực thi “Quản lý mẫu biểu”	36
Hình 2. 12: Biểu đồ công tác thực thi Use Case “Đăng nhập”	37
Hình 2. 13: Biểu đồ công tác thực thi Use Case “Nộp đơn”	37
Hình 2. 14: Biểu đồ công tác thực thi Use Case “Theo dõi trạng thái hồ sơ”	38
Hình 2. 15: Biểu đồ công tác thực thi Use Case “Xử lý hồ sơ”	38
Hình 2. 16: Biểu đồ công tác thực thi Use Case “Quản lý tài khoản người dùng”	39
Hình 2. 17: Biểu đồ công tác thực thi Use Case “Quản lý phòng ban”	39
Hình 2. 18: Biểu đồ công tác thực thi Use Case “Quản lý mẫu biểu”	40
Hình 2. 19: Mô hình lớp “Đăng nhập”	41
Hình 2. 20: Mô hình lớp “Nộp đơn hành chính”	43
Hình 2. 21: Mô hình lớp “Theo dõi trạng thái hồ sơ”	46
Hình 2. 22: Mô hình lớp “Xử lý hồ sơ”	48
Hình 2. 23: Mô hình lớp “Quản lý người dùng”	50
Hình 2. 24: Mô hình lớp “Quản lý phòng ban”	52
Hình 2. 25: Mô hình lớp “Quản lý mẫu biểu”	54
Hình 2. 26: Biểu đồ lớp thiết kế UML	57
Hình 2. 27: Mô hình thực thể ER	59
Hình 3. 1: Giao diện trang đăng nhập.....	63
Hình 3. 2: Giao diện quản trị admin	63
Hình 3. 3: Giao diện admin trang “Quản lý Khoa/Phòng ban”	64
Hình 3. 4: Giao diện admin trang “Thêm mới phòng ban”	64
Hình 3. 5: Giao diện admin trang “Quản lý mẫu biểu”	65
Hình 3. 6: Giao diện admin trang “Thêm mẫu biểu mới”	65
Hình 3. 7: Giao diện admin trang “Quản lý người dùng”	66
Hình 3. 8: Giao diện admin trang “Thêm người dùng mới”	66
Hình 3. 9: Giao diện trang quản lý	67

Hình 3. 10: Giao diện quản lý trang “Quản lý yêu cầu”	67
Hình 3. 11: Giao diện quản lý trang “Quản lý thủ tục”	68
Hình 3. 12: Giao diện trang chủ sinh viên.....	68
Hình 3. 13: Giao diện sinh viên trang “Theo dõi tình trạng xử lý hồ sơ”	69
Hình 3. 14: Giao diện sinh viên trang “Thông tin đăng ký”	69

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, trong các trường đại học và cơ sở giáo dục, công tác quản lý hành chính đối với sinh viên vẫn còn nhiều hạn chế do quy trình xử lý thủ công, yêu cầu sinh viên phải nộp hồ sơ giấy và chờ đợi xử lý trực tiếp tại các phòng ban. Điều này gây mất nhiều thời gian, công sức và tiềm ẩn nguy cơ thất lạc hoặc xử lý chậm trễ hồ sơ.

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong mọi lĩnh vực, đặc biệt là giáo dục, việc xây dựng một hệ thống quản lý hành chính online là giải pháp cấp thiết nhằm tự động hóa quy trình xử lý hồ sơ, tạo điều kiện thuận lợi cho sinh viên, đồng thời giúp cán bộ quản lý giảm tải công việc và nâng cao hiệu suất.

Hơn nữa, đại dịch Covid-19 đã tạo ra bước ngoặt trong nhận thức của nhà trường và sinh viên về việc cần thiết phải chuyển sang mô hình làm việc từ xa, trực tuyến, đặc biệt trong các quy trình hành chính. Từ những lý do nêu trên, em quyết định chọn đề tài “Xây dựng website quản lý hành chính online dành cho sinh viên” làm đề án tốt nghiệp.

2. Nội dung nghiên cứu

- Tìm hiểu quy trình xử lý thủ tục hành chính tại các trường đại học.
- Phân tích yêu cầu người dùng và hệ thống.
- Thiết kế cơ sở dữ liệu và kiến trúc hệ thống website.
- Xây dựng giao diện người dùng và các chức năng: đăng nhập, nộp biểu mẫu, theo dõi trạng thái hồ sơ, quản lý phòng ban, người dùng, mẫu biểu.
- Kiểm thử và đánh giá hệ thống

3. Mục đích chọn đề tài

Đề tài nhằm xây dựng một website quản lý hành chính online, giúp:

- Sinh viên dễ dàng thực hiện các thủ tục hành chính trực tuyến.
- Cán bộ hành chính quản lý và xử lý hồ sơ hiệu quả.
- Quản trị viên kiểm soát toàn bộ hệ thống (phòng ban, người dùng, mẫu biểu).
- Tăng cường ứng dụng công nghệ thông tin trong giáo dục.

4. Phương pháp nghiên cứu

- **Lý thuyết:** Tìm hiểu về mô hình Client-Server, hệ quản trị cơ sở dữ liệu (MySQL), và các công nghệ web (ReactJS, Node.js, Axios).
- **Thực hành:** Sử dụng ReactJS và Ant Design để phát triển giao diện, Node.js để xây dựng API, MySQL để quản lý dữ liệu.
- **Hoạt động:** Đảm bảo hệ thống đáp ứng các chức năng theo yêu cầu, cung cấp kết quả xử lý chính xác.

5. Phạm vi, đối tượng nghiên cứu

- **Phạm vi:** Hệ thống được thiết kế cho sinh viên, cán bộ hành chính, và quản trị viên tại các trường đại học, với các chức năng như đăng nhập, nộp đơn, theo dõi trạng thái, và quản lý hệ thống.
- **Đối tượng:** Sinh viên, cán bộ hành chính, và quản trị viên có nhu cầu sử dụng dịch vụ hành chính trực tuyến.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1. Hiện trạng hệ thống quản lý hành chính tại các trường đại học

Hiện nay, nhiều trường đại học vẫn sử dụng quy trình xử lý thủ tục hành chính thủ công, bao gồm:

- Sinh viên phải đến trực tiếp các phòng ban để nộp hồ sơ, xin xác nhận, hoặc tra cứu kết quả.
- Cán bộ hành chính xử lý hồ sơ bằng giấy tờ, dễ xảy ra sai sót hoặc mất mát.
- Quy trình thiếu tính minh bạch, sinh viên khó theo dõi trạng thái hồ sơ.

Những hạn chế này làm giảm hiệu quả quản lý và gây bất tiện cho sinh viên. Một số trường đã triển khai hệ thống trực tuyến, nhưng vẫn thiếu các tính năng như thông báo tự động hoặc quản lý mẫu biểu linh hoạt.

2. Phát biểu bài toán

Nhu cầu thực hiện thủ tục hành chính trực tuyến ngày càng tăng, nhưng các hệ thống hiện tại chưa đáp ứng đầy đủ. Đề tài tập trung xây dựng một website:

- Cho phép sinh viên đăng ký, đăng nhập, nộp biểu mẫu, và theo dõi trạng thái hồ sơ.
- Hỗ trợ cán bộ hành chính xem và xử lý hồ sơ, quản lý mẫu biểu.
- Cung cấp giao diện quản trị viên để quản lý phòng ban, người dùng, và mẫu biểu.
- Tích hợp thông báo qua email hoặc ứng dụng (tính năng chưa triển khai đầy đủ).

3. Giải pháp

- **Cơ sở dữ liệu:** Sử dụng MySQL để lưu trữ thông tin về người dùng, phòng ban, mẫu biểu, và hồ sơ.
- **Giao diện người dùng:** Thiết kế giao diện thân thiện với Ant Design và ReactJS, hỗ trợ responsive.
- **Chức năng:** Xây dựng các module đăng nhập/đăng ký, nộp/xử lý hồ sơ, theo dõi trạng thái, và quản lý hệ thống.
- **API:** Sử dụng Node.js và Axios để kết nối frontend và backend.

4. Yêu cầu đạt được của hệ thống

- **Yêu cầu chung:**
 - Đăng nhập/đăng ký theo vai trò (sinh viên, cán bộ, quản trị viên).
 - Quản lý mẫu biểu, phòng ban, người dùng.
 - Theo dõi và xử lý hồ sơ.
- **Yêu cầu chức năng:**
 - Sinh viên: Xem danh sách mẫu biểu, điền đơn, nộp hồ sơ, theo dõi trạng thái.
 - Cán bộ: Xem danh sách hồ sơ, xử lý (phê duyệt/từ chối), phản hồi.
 - Quản trị viên: Quản lý phòng ban, người dùng, mẫu biểu.
 - Giao diện: Thân thiện, dễ sử dụng, hỗ trợ tìm kiếm và lọc dữ liệu.

5. Giới thiệu các phần mềm sử dụng

Trong quá trình xây dựng hệ thống website quản lý hành chính online dành cho sinh viên, nhóm phần mềm và công cụ được lựa chọn đóng vai trò rất quan trọng, giúp đảm bảo hiệu quả trong lập trình, quản lý mã nguồn, thiết kế giao diện và vận hành hệ thống. Dưới đây là các phần mềm, thư viện và môi trường phát triển chính được sử dụng trong dự án:

a. ReactJS

ReactJs là một thư viện JavaScript mã nguồn mở do Facebook phát triển, được sử dụng rộng rãi trong việc xây dựng giao diện người dùng (UI) cho các ứng dụng web một trang (Single Page Application - SPA). React sử dụng khái niệm component-based architecture, tức là chia giao diện thành các thành phần độc lập, giúp lập trình viên dễ dàng tái sử dụng, bảo trì và mở rộng hệ thống.

Trong dự án này, ReactJS được dùng để phát triển toàn bộ phần giao diện người dùng. Mỗi chức năng như gửi biểu mẫu, xem hồ sơ, quản lý trạng thái,... đều được tổ chức dưới dạng component, giúp việc quản lý giao diện trở nên khoa học và hiệu quả hơn.

b. Node.js

Node.js là một môi trường chạy JavaScript phía máy chủ, giúp xây dựng các ứng dụng backend hiệu quả, nhẹ và linh hoạt. Trong hệ thống này, Node.js (kết hợp với ExpressJS) được sử dụng để xây dựng API backend, nơi xử lý các yêu cầu HTTP từ frontend (ReactJS) và tương tác với cơ sở dữ liệu (MySQL).

Việc sử dụng Node.js giúp đảm bảo phản hồi nhanh, dễ mở rộng, phù hợp với kiến trúc RESTful hiện đại. Backend NodeJS xử lý các chức năng như: xác

thực người dùng, lưu trữ biểu mẫu, cập nhật trạng thái hồ sơ và gửi thông báo phản hồi.

c. Axios

Axios là một thư viện JavaScript dùng để gửi các yêu cầu HTTP từ frontend đến backend. Axios hỗ trợ các phương thức như GET, POST, PUT, DELETE,... với cú pháp đơn giản, dễ sử dụng và hỗ trợ Promise/async-await.

Trong dự án, Axios là cầu nối giữa ReactJS và Node.js. Mỗi lần người dùng gửi biểu mẫu, truy vấn danh sách, hoặc cập nhật trạng thái hồ sơ, React sẽ gọi API thông qua Axios để trao đổi dữ liệu với backend. Thư viện này giúp đảm bảo việc giao tiếp giữa frontend và backend diễn ra hiệu quả và bảo mật.

d. MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ phổ biến, được sử dụng để lưu trữ toàn bộ thông tin người dùng, biểu mẫu, hồ sơ nộp, phản hồi và trạng thái xử lý. Với cấu trúc bảng rõ ràng, hỗ trợ mối quan hệ giữa các bảng (quan hệ 1-n, n-n), MySQL rất phù hợp với các hệ thống quản lý dữ liệu như đề tài này.

Cơ sở dữ liệu được thiết kế với các bảng chính như: users, departments, form_templates, submitted_forms, notifications, feedbacks,... nhằm đảm bảo truy vấn nhanh, dễ mở rộng và quản lý logic dữ liệu hiệu quả.

e. Visual Studio Code

Visual Studio Code (VSCode) là trình soạn thảo mã nguồn đa nền tảng, nhẹ, mạnh mẽ và miễn phí do Microsoft phát triển. VSCode hỗ trợ nhiều tiện ích (extensions) như ESLint, Prettier, GitLens, hỗ trợ terminal tích hợp và đặc biệt phù hợp với phát triển ứng dụng ReactJS + Node.js.

Trong quá trình phát triển đồ án, VSCode là công cụ chính được sử dụng để viết mã frontend (ReactJS), backend (Node.js) cũng như thiết kế cấu trúc thư mục dự án. Tính năng tự động gợi ý, quản lý file, và tích hợp Git giúp quá trình phát triển trở nên nhanh chóng và hiệu quả hơn.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1. Khảo sát hệ thống

1.1. Cách thức khảo sát:

- Phỏng vấn sinh viên và cán bộ hành chính để hiểu nhu cầu.
- Tham khảo các hệ thống quản lý hành chính tại các trường đại học.
- Phân tích quy trình xử lý hồ sơ truyền thống.

1.2. Kết quả:

- Sinh viên cần hệ thống trực tuyến để nộp đơn và theo dõi trạng thái.
- Cán bộ cần giao diện quản lý hồ sơ hiệu quả.
- Quản trị viên cần công cụ quản lý toàn diện.

2. Phân tích yêu cầu hệ thống

2.1. Tranh đăng nhập/ đăng ký:

- Hỗ trợ xác thực theo vai trò (admin, staff, student).
- Lưu token và thông tin người dùng vào localStorage.

2.2. Trang chủ (sinh viên):

- Hiện thị danh sách mẫu biểu.
- Mở modal để điền và nộp đơn.

2.3. Trang xử lý hồ sơ (sinh viên):

- Lọc hồ sơ theo trạng thái (pending, processing, approved, rejected).
- Xem chi tiết hồ sơ.

2.4. Trang quản lý (cán bộ)

- Xem danh sách hồ sơ, phê duyệt/từ chối, phản hồi.
- Quản lý mẫu biểu.

2.5. Trang quản trị (admin)

- Quản lý phòng ban, người dùng, mẫu biểu.

3. Quy trình nghiệp vụ

Quy trình nghiệp vụ mô tả các bước mà các tác nhân (sinh viên, cán bộ, quản trị viên) thực hiện khi tương tác với hệ thống quản lý hành chính online dành cho sinh viên. Hệ thống không hỗ trợ người dùng tự đăng ký tài khoản; thay vào đó, tài khoản được quản trị viên (admin) tạo và cấp cho người dùng. Phần này được chia thành quy trình nghiệp vụ chung (tổng quan cho từng vai trò) và quy trình nghiệp vụ chi tiết (mô tả từng bước cho các chức năng cụ thể: đăng nhập, nộp đơn, theo dõi trạng thái, xử lý hồ sơ, quản lý hệ thống).

3.1. Quy trình nghiệp vụ chung

Quy trình nghiệp vụ chung mô tả luồng hoạt động tổng thể của các vai trò trong hệ thống:

3.1.1. Sinh viên:

- Sử dụng tài khoản do quản trị viên cấp để đăng nhập.
- Xem danh sách mẫu biểu, chọn mẫu, điền thông tin, nộp hồ sơ.
- Theo dõi trạng thái hồ sơ (chờ xử lý, đang xử lý, được phê duyệt, bị từ chối).
- Xem chi tiết hồ sơ và phản hồi từ cán bộ (nếu có).

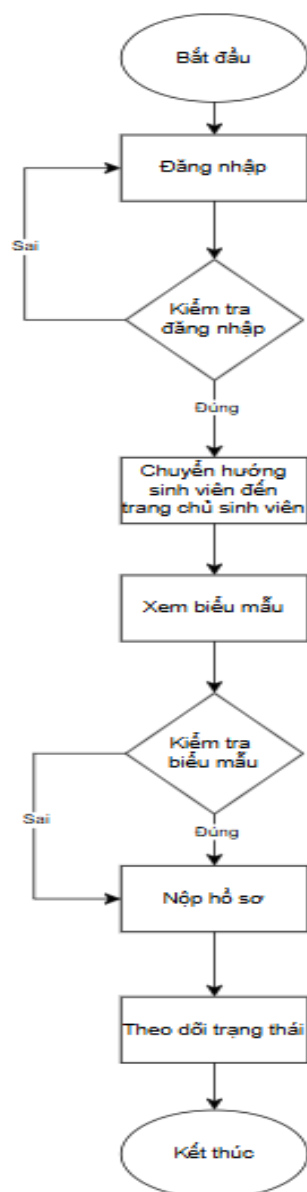
3.1.2. Cán bộ (staff)

- Sử dụng tài khoản do quản trị viên cấp để đăng nhập.
- Xem danh sách hồ sơ được gửi đến phòng ban của mình.
- Xử lý hồ sơ (phê duyệt, từ chối, gửi phản hồi).
- Quản lý mẫu biểu (xem, thêm, sửa, xóa).

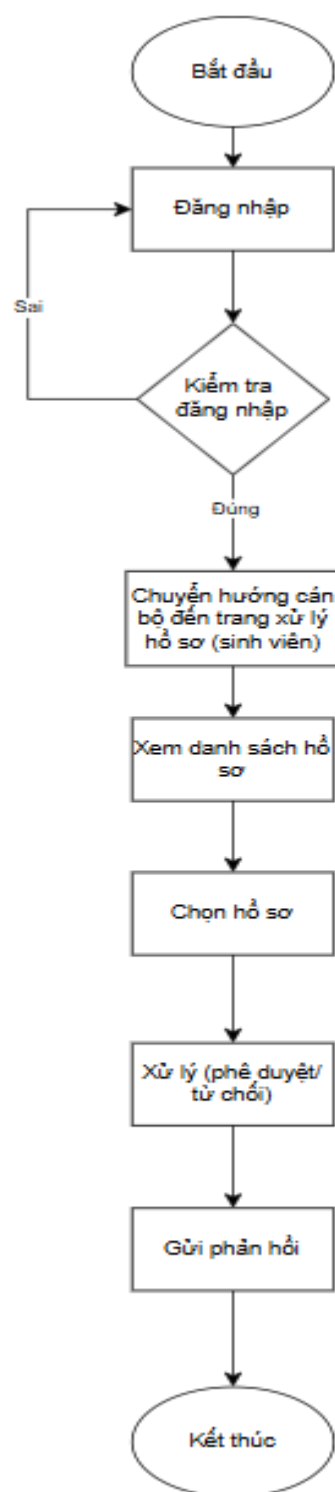
3.1.3. Quản trị viên (admin)

- Sử dụng tài khoản admin để đăng nhập.
- Tạo và quản lý tài khoản cho sinh viên và cán bộ (thêm, sửa, xóa).
- Quản lý phòng ban (thêm, sửa, xóa).
- Quản lý mẫu biểu (thêm, sửa, xóa, xem trước).

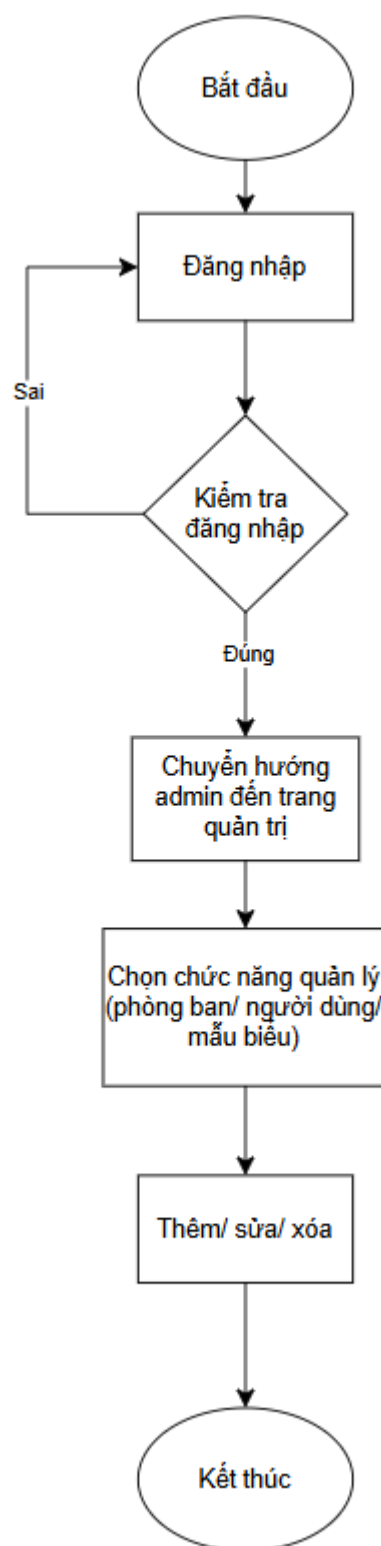
❖ Quy trình nghiệp vụ chung



Hình 2. 1: Sơ đồ tiến trình nghiệp vụ chung Sinh viên



Hình 2. 2: Sơ đồ tiến trình nghiệp vụ chung Cán bộ (staff)



Hình 2. 3: Sơ đồ tiến trình nghiệp vụ chung Admin

3.2. Quy trình nghiệp vụ chi tiết

3.2.1. Quy trình “Đăng nhập”

- **Tác nhân:** Người dùng (sinh viên, cán bộ, quản trị viên)
- **Mục đích:** Xác thực và truy cập hệ thống theo vai trò.
- **Điều kiện trước:** Người dùng đã được quản trị viên cấp tài khoản và truy cập trang đăng nhập (/login).
- **Các bước thực hiện:**
 - Người dùng truy cập website và vào trang đăng nhập.
 - Hệ thống hiển thị giao diện đăng nhập với các trường:
 - Username (tên đăng nhập)
 - Password (mật khẩu).
 - Người dùng nhập thông tin tài khoản và nhấp nút “Đăng nhập”
 - Hệ thống gửi yêu cầu POST đến API **/auth/login** với dữ liệu { username, password }.
 - Backend xác thực:
 - Kiểm tra username và password có khớp với dữ liệu trong database không.
 - Nếu khớp, trả về token JWT và thông tin người dùng (id, username, role, departmentId).
 - Hệ thống lưu token và thông tin người dùng vào localStorage.
 - Dựa trên vai trò (role) của người dùng, hệ thống chuyển hướng:
 - Admin: **/admin**
 - Staff: **/staff**
 - Student: **/home**
 - Hiển thị thông báo “Đăng nhập thành công” (nếu được cấu hình trong giao diện).
- **Ngoại lệ:**
 - Nếu username hoặc password sai: Hiển thị thông báo “Tên đăng nhập hoặc mật khẩu không đúng”.
 - Nếu token hết hạn hoặc lỗi 401 (Unauthorized): Xóa token cũ trong localStorage và chuyển hướng về **/login** với thông báo “Phiên đăng nhập hết hạn”
 - Nếu mất kết nối với API: Hiển thị thông báo “Không thể kết nối, vui lòng thử lại”.

3.2.2. Quy trình “Nộp đơn”

- **Tác nhân:** Sinh viên.
- **Mục đích:** Nộp hồ sơ hành chính dựa trên mẫu biểu.
- **Điều kiện trước:** Sinh viên đã đăng nhập và truy cập trang chủ (*/home*).
- **Các bước thực hiện:**
 - Sinh viên truy cập trang chủ và nhấp vào mục “Danh sách mẫu biểu”.
 - Hệ thống gửi yêu cầu GET đến API */form-templates* để lấy danh sách mẫu biểu.
 - Hệ thống hiển thị danh sách mẫu biểu dưới dạng bảng hoặc danh sách, bao gồm:
 - Tiêu đề mẫu biểu.
 - Phòng ban liên quan.
 - Mô tả (nếu có).
 - Sinh viên chọn một mẫu biểu và nhấp nút “Nộp đơn”.
 - Hệ thống mở modal hiển thị nội dung HTML của mẫu biểu (lấy từ trường *html_content* trong dữ liệu API)
 - Sinh viên điền thông tin vào các trường trong mẫu (ví dụ: thông tin cá nhân, lý do nộp đơn).
 - Sinh viên nhấp nút “Gửi” trong modal.
 - Hệ thống gửi yêu cầu POST đến API */submitted-forms* với dữ liệu:
 - *formTemplateId*: ID của mẫu biểu.
 - *userId*: ID của sinh viên (lấy từ *localStorage*).
 - *html_content*: Nội dung đã điền.
 - *Status*: Mặc định là “pending”.
 - Nếu API trả về thành công, hệ thống hiển thị thông báo “Nộp đơn thành công” và đóng modal.
 - Sinh viên được chuyển hướng đến trang theo dõi trạng thái (*/home/xuly*) để xem hồ sơ vừa nộp.
- **Ngoại lệ:**
 - Nếu mẫu biểu không hợp lệ hoặc không tồn tại: Hiển thị thông báo “Mẫu biểu không khả dụng”.

- Nếu thông tin điều thiếu hoặc không hợp lệ: Hiển thị thông báo “Vui lòng điền đầy đủ thông tin:.”
- Nếu APU trả về lỗi: Hiển thị thông báo “Nộp đơn thất bại, vui lòng thử lại”.

3.2.3. Quy trình “Theo dõi trạng thái hồ sơ”

- **Tác nhân:** Sinh viên.
- **Mục đích:** Xem trạng thái và chi tiết các hồ sơ đã nộp.
- **Điều kiện trước:** Sinh viên đã đăng nhập và truy cập trang theo dõi trạng thái (*/home/xuly*).
- **Các bước thực hiện:**
 - Sinh viên nhấp vào mục “Theo dõi trạng thái” từ menu hoặc trang chủ.
 - Hệ thống gửi yêu cầu GET đến API */submitted-forms/my-submissions* để lấy danh sách hồ sơ của sinh viên (dựa trên `userId` từ `localStorage`).
 - Hệ thống hiển thị danh sách hồ sơ dưới dạng bảng với các cột:
 - Tên biểu mẫu.
 - Phòng ban.
 - Ngày nộp (`submitted_at`).
 - Trạng thái (`pending`, `processing`, `approved`, `rejected`).
 - Người xử lý (nếu có).
 - Sinh viên sử dụng bộ lọc (`dropdown` hoặc `checkbox`) để lọc hồ sơ theo trạng thái.
 - Sinh viên nhấp vào một hồ sơ để xem chi tiết.
 - Hệ thống chuyển hướng đến trang chi tiết hồ sơ (*submissions/:id*) và gửi yêu cầu GET đến API */submitted-forms/:id* để lấy thông tin chi tiết:
 - Nội dung HTML của hồ sơ.
 - Phản hồi từ cán bộ (`feedback`).
 - Thời gian xử lý (`processed_at`).
 - Hệ thống hiển thị chi tiết hồ sơ, bao gồm trạng thái, nội dung đã điền, và phản hồi (nếu có).
- **Ngoại lệ:**

- Nếu không có hồ sơ: Hiển thị thông báo “Chưa có hồ sơ nào”.
- Nếu API trả về lỗi: Hiển thị thông báo “Không thể tải danh sách hồ sơ, vui lòng thử lại”.
- Nếu hồ sơ không tồn tại: Chuyển hướng đến trang lỗi (*/not-found*).

3.2.4. Quy trình “Xử lý hồ sơ”

- **Tác nhân:** Cán bộ (staff).
- **Mục đích:** Xem, phê duyệt, từ chối, hoặc gửi phản hồi cho hồ sơ.
- **Điều kiện trước:** Cán bộ đã đăng nhập và truy cập trang quản lý yêu cầu (*/staff/requests*).
- **Các bước thực hiện:**
 - Cán bộ nhấp vào mục “Yêu cầu” từ menu.
 - Hệ thống gửi yêu cầu GET đến API */submitted-forms/department/:departmentId* để lấy danh sách hồ sơ thuộc phòng ban của cán bộ (dựa trên *departmentId* từ *localStorage*).
 - Hệ thống hiển thị danh sách hồ sơ dưới dạng bảng với các cột:
 - Tên mẫu biểu.
 - Sinh viên nộp.
 - Ngày nộp.
 - Trạng thái.
 - Cán bộ nhấp vào một hồ sơ để xem chi tiết.
 - Hệ thống hiển thị nội dung HTML của hồ sơ và các tùy chọn xử lý:
 - Phê duyệt (Approved).
 - Từ chối (Rejected).
 - Gửi phản hồi (textarea để nhập nội dung phản hồi).
 - Cán bộ chọn hành động (phê duyệt hoặc từ chối) và nhập phản hồi (tùy chọn).
 - Hệ thống gửi yêu cầu PUT đến API */submitted-forms/:id* với dữ liệu:
 - status: Approved hoặc Rejected.

- feedback: Nội dung phản hồi (nếu có).
- processed_at: Thời gian xử lý hiện tại.
- Nếu API trả về thành công, hệ thống hiển thị thông báo “Xử lý hồ sơ thành công” và làm mới danh sách hồ sơ.
- **Ngoại lệ:**
 - Nếu hồ sơ không thuộc phòng ban của cán bộ: Chuyển hướng đến trang lỗi (**/unauthorized**).
 - Nếu API trả về lỗi: Hiển thị thông báo “Xử lý thất bại, vui lòng thử lại”.
 - Nếu hồ sơ đã được xử lý trước đó: Hiển thị thông báo “Hồ sơ đã được xử lý”.

3.2.5. Quy trình “Quản lý tài khoản người dùng”

- **Tác nhân:** Quản trị viên (admin).
- **Mục đích:** Tạo, sửa, xóa tài khoản cho sinh viên và cán bộ.
- **Điều kiện trước:** Quản trị viên đã đăng nhập và truy cập trang quản lý người dùng (**/admin/users**).
- **Các bước thực hiện:**
 - Quản trị viên nhấp vào mục “Người dùng” từ menu admin.
 - Hệ thống gửi yêu cầu GET đến API **/users** để lấy danh sách người dùng và hiển thị dưới dạng bảng với các cột:
 - Username.
 - Email.
 - Vai trò (admin, staff, student).
 - Phòng ban.
 - **Để tạo tài khoản:**
 - Quản trị viên nhấp nút “Thêm mới”.
 - Hệ thống hiển thị form với các trường:
 - Username.
 - Password (mật khẩu mặc định).
 - Email.
 - Role (admin, staff, student).
 - Department (chọn từ danh sách phòng ban).
 - Quản trị viên nhập thông tin và nhấp “Lưu”.
 - Hệ thống gửi yêu cầu POST đến API **/users**.

- **Để sửa tài khoản:**
 - Quản trị viên nhấp nút “Sửa” trên một người dùng.
 - Hệ thống hiển thị form với thông tin hiện tại.
 - Quản trị viên chỉnh sửa (ví dụ: thay đổi email, vai trò, phòng ban) và nhấp “Lưu”.
 - Hệ thống gửi yêu cầu PUT đến API */users/:id*.
- **Để xóa tài khoản:**
 - Quản trị viên nhấp nút “Xóa” trên một người dùng.
 - Hệ thống hiển thị thông báo xác nhận.
 - Nếu xác nhận, hệ thống gửi yêu cầu DELETE đến API */users/:id*.
- Nếu API trả về thành công, hệ thống làm mới danh sách và hiển thị thông báo tương ứng (“Thêm thành công”, “Sửa thành công”, “Xóa thành công”).
- **Ngoại lệ:**
 - Nếu username hoặc email đã tồn tại: Hiển thị thông báo “Người dùng đã tồn tại”.
 - Nếu người dùng đang có hồ sơ: Hiển thị thông báo “Không thể xóa người dùng có hồ sơ”.
 - Nếu API trả về lỗi: Hiển thị thông báo “Thao tác thất bại, vui lòng thử lại”.

3.2.6. Quy trình “Quản lý phòng ban”

- **Tác nhân:** Quản trị viên (admin).
- **Mục đích:** Thêm, sửa, xóa thông tin phòng ban.
- **Điều kiện trước:** Quản trị viên đã đăng nhập và truy cập trang quản lý phòng ban (*/admin/departments*).
- **Các bước thực hiện:**
 - Quản trị viên nhấp vào mục “Phòng ban” từ menu admin.
 - Hệ thống gửi yêu cầu GET đến API */departments* để lấy danh sách phòng ban và hiển thị dưới dạng bảng với các cột:
 - Tên phòng ban.
 - Số lượng người dùng.
 - Số lượng mẫu biểu.
 - **Để thêm phòng ban:**

- Quản trị viên nhấp nút “Thêm mới”.
- Hệ thống hiển thị form với trường “Tên phòng ban”.
- Quản trị viên nhập tên và nhấp “Lưu”.
- Hệ thống gửi yêu cầu POST đến API */departments*.
- **Để sửa phòng ban:**
 - Quản trị viên nhấp nút “Sửa” trên một phòng ban.
 - Hệ thống hiển thị form với tên phòng ban hiện tại.
 - Quản trị viên chỉnh sửa tên và nhấp “Lưu”.
 - Hệ thống gửi yêu cầu PUT đến API */departments/:id*.
- **Để xóa phòng ban:**
 - Quản trị viên nhấp nút “Xóa” trên một phòng ban.
 - Hệ thống hiển thị thông báo xác nhận.
 - xác nhận, hệ thống gửi yêu cầu DELETE đến API */departments/:id*.
- Nếu API trả về thành công, hệ thống làm mới danh sách và hiển thị thông báo tương ứng.
- **Ngoại lệ:**
 - Nếu tên phòng ban đã tồn tại: Hiển thị thông báo “Phòng ban đã tồn tại”.
 - Nếu phòng ban đang có người dùng hoặc mẫu biểu: Hiển thị thông báo “Không thể xóa phòng ban đang được sử dụng”.
 - Nếu API trả về lỗi: Hiển thị thông báo “Thao tác thất bại, vui lòng thử lại”.

3.2.7. Quy trình “Quản lý mẫu biểu”

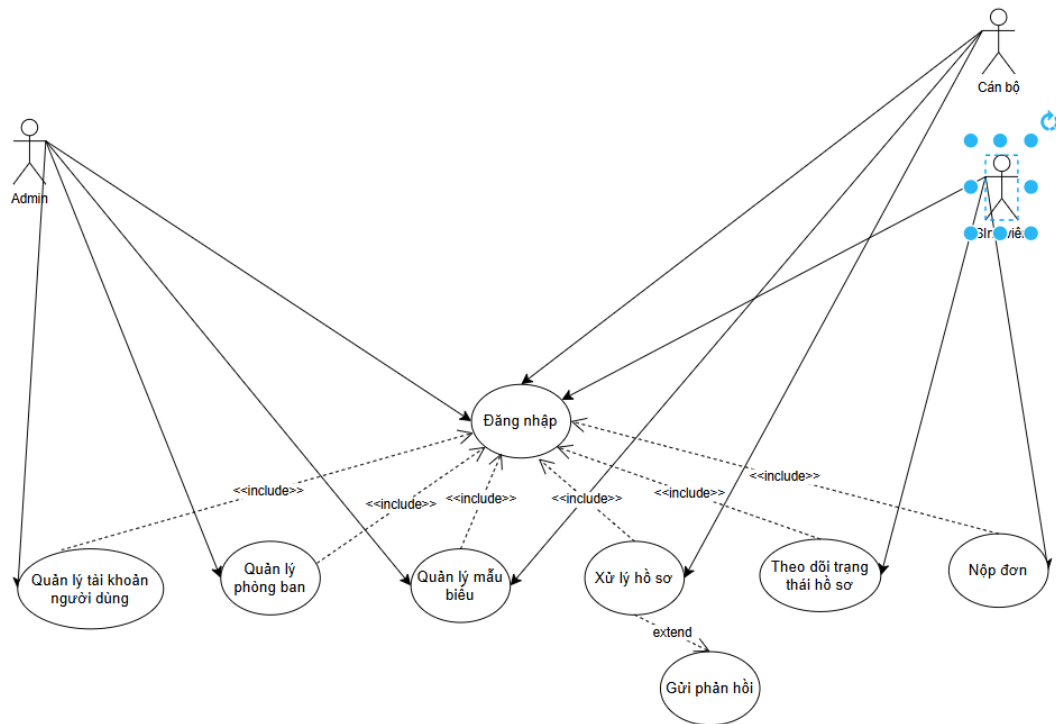
- **Tác nhân:** Quản trị viên hoặc cán bộ.
- **Mục đích:** Thêm, sửa, xóa, xem trước mẫu biểu.
- **Điều kiện trước:** Đã đăng nhập và truy cập trang quản lý mẫu biểu (*/admin/form-templates* hoặc */staff/form*).
- **Các bước thực hiện:**
 - Người dùng nhấp vào mục “Mẫu biểu” từ menu.
 - Hệ thống gửi yêu cầu GET đến:
 - API */form-templates* (cho admin).
 - API */form-templates/department/:departmentId* (cho staff)

- Hệ thống hiển thị danh sách mẫu biểu với các cột:
 - Tiêu đề.
 - Phòng ban.
 - Ngày tạo.
- **Để thêm mẫu biểu:**
 - Người dùng nhấp nút “Thêm mới”.
 - Hệ thống hiển thị form với các trường:
 - Tiêu đề.
 - Phòng ban (chọn từ danh sách, chỉ admin có quyền chọn).
 - File HTML (tải lên file hoặc nhập trực tiếp).
 - Người dùng nhập thông tin và nhấp “Lưu”.
 - Hệ thống gửi yêu cầu POST đến API */form-templates* với Content-Type: *multipart/form-data*.
- **Để sửa mẫu biểu:**
 - Người dùng nhấp nút “Sửa” trên một mẫu biểu.
 - Hệ thống hiển thị form với thông tin hiện tại.
 - Người dùng chỉnh sửa và nhấp “Lưu”.
 - Hệ thống gửi yêu cầu PUT đến API */form-templates/:id*.
- **Để xem trước mẫu biểu:**
 - Người dùng nhấp nút “Xem trước”.
 - Hệ thống hiển thị nội dung HTML trong trang preview.
- **Để xóa mẫu biểu:**
 - Người dùng nhấp nút “Xóa” và xác nhận.
 - Hệ thống gửi yêu cầu DELETE đến API */form-templates/:id*.
- Nếu API trả về thành công, hệ thống hiển thị thông báo tương ứng (“Thêm thành công”, “Sửa thành công”, “Xóa thành công”).
- **Ngoại lệ:**
 - Nếu file HTML không hợp lệ: Hiển thị thông báo “File không đúng định dạng”.

- Nếu mẫu biểu đang có hồ sơ: Hiển thị thông báo “Không thể xóa mẫu biểu đang được sử dụng”.
- Nếu API trả về lỗi: Hiển thị thông báo “Thao tác thất bại, vui lòng thử lại”.

4. BIỂU ĐỒ USE CASE

4.1. Use Case tổng quát



Hình 2. 4: Sơ đồ Use Case tổng quát

Mô tả:

- Tác nhân:

- **Sinh viên:** Người dùng có vai trò student, sử dụng tài khoản do quản trị viên cấp để nộp hồ sơ hành chính và theo dõi trạng thái.
- **Cán bộ (Staff):** Người dùng có vai trò staff, sử dụng tài khoản do quản trị viên cấp để xem và xử lý hồ sơ, quản lý mẫu biểu trong phạm vi phòng ban.
- **Quản trị viên (Admin):** Người dùng có vai trò admin, có toàn quyền quản lý tài khoản người dùng, phòng ban, và mẫu biểu.

- **Chức năng chính:**
 - **Đăng nhập:** Tất cả tác nhân đăng nhập vào hệ thống bằng tài khoản được cấp.
 - **Nộp đơn:** Sinh viên chọn mẫu biểu, điền thông tin, và nộp hồ sơ.
 - **Theo dõi trạng thái hồ sơ:** Sinh viên xem danh sách hồ sơ và chi tiết trạng thái.
 - **Xử lý hồ sơ:** Cán bộ xem, phê duyệt, từ chối, hoặc gửi phản hồi cho hồ sơ.
 - **Quản lý tài khoản người dùng:** Quản trị viên tạo, sửa, xóa tài khoản cho sinh viên và cán bộ.
 - **Quản lý phòng ban:** Quản trị viên thêm, sửa, xóa phòng ban.
 - **Quản lý mẫu biểu:** Quản trị viên và cán bộ thêm, sửa, xóa, xem trước mẫu biểu (cán bộ chỉ quản lý trong phòng ban của mình).
- **Mối quan hệ:**
 - **Include:** Các chức năng như “Nộp đơn” hoặc “Xử lý hồ sơ” yêu cầu “Đăng nhập” trước (đăng nhập là điều kiện tiên quyết).
 - **Extend:** Một số chức năng có thể mở rộng, ví dụ: “Xử lý hồ sơ” có thể bao gồm “Gửi phản hồi” như một tùy chọn.

4.2. Use Case chi tiết

4.2.1. Use Case 1: Đăng nhập

Tên Use Case	Đăng nhập
Tác nhân	Sinh viên, Cán bộ, Quản trị viên.
Mục đích	Xác thực người dùng để truy cập hệ thống theo vai trò.
Điều kiện trước	- Người dùng đã được quản trị viên cấp tài khoản (username, password). - Người dùng truy cập trang đăng nhập (/login).
Luồng chính: - Người dùng mở trình duyệt và truy cập trang đăng nhập. - Hệ thống hiển thị giao diện đăng nhập với các trường: Username, Password. - Người dùng nhập username và password, sau đó nhấp nút “Đăng nhập”. - Hệ thống gửi yêu cầu POST đến API /auth/login với dữ liệu {username, password}. - Backend kiểm tra: ▪ Username và password có khớp với dữ liệu trong bảng Users không. ▪ Nếu khớp, trả về token JWT và thông tin người dùng (id, username, role, departmentId). - Hệ thống lưu token và thông tin người dùng vào localStorage. - Hệ thống chuyển hướng người dùng dựa trên vai trò: ▪ Sinh viên (student): /home. ▪ Cán bộ (staff): /staff. ▪ Quản trị viên (admin): /admin. - Hiển thị thông báo “Đăng nhập thành công” (tùy thuộc cấu hình giao diện).	
Luồng phụ: - Nếu người dùng quên mật khẩu, nhấp liên kết “Quên mật khẩu” (chưa triển khai, chuyển hướng đến trang liên hệ admin).	

Ngoại lệ	- Username hoặc password sai: Hiện thị thông báo “Tên đăng nhập hoặc mật khẩu không đúng”. - Token hết hạn hoặc lỗi 401: Xóa token cũ và hiện thị thông báo “Phiên đăng nhập hết hạn”. - Mất kết nối API: Hiện thị thông báo “Không thể kết nối, vui lòng thử lại”.
----------	---

4.2.2. Use Case 2: Nộp đơn

Tên Use Case	Nộp đơn
Tác nhân	Sinh viên.
Mục đích	Nộp hồ sơ hành chính dựa trên mẫu biểu.
Điều kiện trước	- Sinh viên đã đăng nhập (vai trò student). - Sinh viên truy cập trang chủ (<i>/home</i>).
Luồng chính: - Sinh viên nhấp vào mục “Danh sách mẫu biểu” trên trang chủ. - Hệ thống gửi yêu cầu GET đến API <i>/form-templates</i> để lấy danh sách mẫu biểu. - Hệ thống hiện thị danh sách mẫu biểu với thông tin: tiêu đề, phòng ban, mô tả. - Sinh viên chọn một mẫu biểu và nhấp nút “Nộp đơn”. - Hệ thống mở modal (FormModal) hiện thị nội dung HTML của mẫu biểu (<i>html_content</i>). - Sinh viên điền thông tin vào các trường trong mẫu (ví dụ: thông tin cá nhân, lý do nộp đơn). - Sinh viên nhấp nút “Gửi”. - Hệ thống gửi yêu cầu POST đến API <i>/submitted-forms</i> với dữ liệu: - o <i>formTemplateId</i>: ID mẫu biểu. - o <i>userId</i>: ID sinh viên (từ <i>localStorage</i>). - o <i>html_content</i>: Nội dung đã điền. - o <i>status</i>: “pending”. - Nếu API trả về thành công, hệ thống hiện thị thông báo “Nộp đơn thành công” và đóng modal. - Sinh viên được chuyển hướng đến trang theo dõi trạng thái (<i>/home/xuly</i>).	
Luồng phụ: Không có.	

Ngoại lệ	- Mẫu biểu không tồn tại: Hiện thị thông báo “Mẫu biểu không khả dụng”. - Thông tin điền thiếu: Hiện thị thông báo “Vui lòng điền đầy đủ thông tin”. - Lỗi API: Hiện thị thông báo “Nộp đơn thất bại, vui lòng thử lại”.
----------	---

4.2.3. Use Case 3: Theo dõi trạng thái hồ sơ

Tên Use Case	Theo dõi trạng thái hồ sơ
Tác nhân	Sinh viên.
Mục đích	Xem danh sách và chi tiết trạng thái các hồ sơ đã nộp.
Điều kiện trước	- Sinh viên đã đăng nhập (vai trò student). - Sinh viên truy cập trang theo dõi trạng thái (<i>/home/xuly</i>).
Luồng chính: - Sinh viên nhấp vào mục “Theo dõi trạng thái. - Hệ thống gửi yêu cầu GET đến API <i>/submitted-forms/my-submissions</i> để lấy danh sách hồ sơ của sinh viên (dựa trên userId). - Hệ thống hiển thị bảng hồ sơ với các cột: - o Tên mẫu biểu. - o Phòng ban. - o Ngày nộp. - o Trạng thái (pending, processing, approved, rejected). - Sinh viên lọc hồ sơ theo trạng thái bằng dropdown hoặc checkbox. - Sinh viên nhấp vào một hồ sơ để xem chi tiết. - Hệ thống chuyển hướng đến trang chi tiết (<i>/submissions/:id</i>) và gửi yêu cầu GET đến API <i>/submitted-forms/:id</i>. - Hệ thống hiển thị: - o Nội dung HTML của hồ sơ. - o Trạng thái. - o Phản hồi từ cán bộ (feedback). - o Thời gian xử lý (processed_at).	
Luồng phụ: Không có.	
Ngoại lệ	- Không có hồ sơ: Hiện thị thông báo “Chưa có hồ sơ nào”.

	- Lỗi API: Hiện thị thông báo “Không thể tải danh sách hồ sơ”. - Hồ sơ không tồn tại: Chuyển hướng đến /not-found.
--	---

4.2.4. Use Case 4: Xử lý hồ sơ

Tên Use Case	Xử lý hồ sơ
Tác nhân	Cán bộ (staff)
Mục đích	Xem, phê duyệt, từ chối, hoặc gửi phản hồi cho hồ sơ.
Điều kiện trước	- Cán bộ đã đăng nhập (vai trò staff). - Cán bộ truy cập trang quản lý yêu cầu (/staff/requests).

Luồng chính:

- Cán bộ nhấp vào mục “Yêu cầu”.
- Hệ thống gửi yêu cầu GET đến API **/submitted-forms/department/:departmentId** để lấy danh sách hồ sơ của phòng ban.
- Hệ thống hiển thị bảng hồ sơ với các cột:
 - o Tên mẫu biểu.
 - o Sinh viên nộp.
 - o Ngày nộp.
 - o Trạng thái.
- Cán bộ nhấp vào một hồ sơ để xem chi tiết.
- Hệ thống hiển thị nội dung HTML của hồ sơ và các tùy chọn:
 - o Phê duyệt (Approved).
 - o Từ chối (Rejected).
 - o Nhập phản hồi (textarea).
- Cán bộ chọn hành động (phê duyệt/từ chối) và nhập phản hồi (tùy chọn).
- Hệ thống gửi yêu cầu PUT đến API **/submitted-forms/:id** với dữ liệu:
 - o status: Approved hoặc Rejected.
 - o feedback: Nội dung phản hồi.
 - o processed_at: Thời gian hiện tại.
- Nếu API trả về thành công, hiển thị thông báo “**Xử lý hồ sơ thành công**” và làm mới danh sách.

Luồng phụ:

- Cán bộ có thể chỉ gửi phản hồi mà không thay đổi trạng thái (chưa triển khai đầy đủ).	
Ngoại lệ	- Hồ sơ không thuộc phòng ban: Chuyển hướng đến /unauthorized. - Lỗi API: Hiển thị thông báo “Xử lý thất bại”. - Hồ sơ đã xử lý: Hiển thị thông báo “Hồ sơ đã được xử lý”.

4.2.5. Use Case 5: Quản lý tài khoản người dùng

Tên Use Case	Quản lý tài khoản người dùng
Tác nhân	Quản trị viên (admin)
Mục đích	Tạo, sửa, xóa tài khoản cho sinh viên và cán bộ.
Điều kiện trước	- Quản trị viên đã đăng nhập (vai trò admin). - Quản trị viên truy cập trang quản lý người dùng (/admin/users).

Luồng chính:

- Quản trị viên nhấp vào mục “Người dùng”
- Hệ thống gửi yêu cầu GET đến API **/users** để lấy danh sách người dùng.
- Hệ thống hiển thị bảng người dùng với các cột:
 - o Username.
 - o Email.
 - o Vai trò.
 - o Phòng ban.
- Để tạo tài khoản:
 - o Nhấp nút “**Thêm mới**”
 - o Hiển thị form với các trường: Username, Password, Email, Role, Department.
 - o Quản trị viên nhập thông tin và nhấp “**Lưu**”.
 - o Hệ thống gửi yêu cầu POST đến API **/users**.
- Để sửa tài khoản:
 - o Nhấp nút “**Sửa**” trên một người dùng.
 - o Hiển thị form với thông tin hiện tại.
 - o Quản trị viên chỉnh sửa và nhấp “**Lưu**”.
 - o Hệ thống gửi yêu cầu PUT đến API **/users/:id**.
- Để xóa tài khoản:

○ Nhấp nút “Xóa” và xác nhận. ○ Hệ thống gửi yêu cầu DELETE đến API <i>/users/:id</i>.	
- Nếu API trả về thành công, hiển thị thông báo và làm mới danh sách.	
Luồng phụ: Không có.	
Ngoại lệ	- Username/email trùng: Hiển thị thông báo “Người dùng đã tồn tại”. - Người dùng có hồ sơ: Hiển thị thông báo “Không thể xóa người dùng có hồ sơ”. - Lỗi API: Hiển thị thông báo “Thao tác thất bại”.

4.2.6. Use Case 6: Quản lý phòng ban

Tên Use Case	Quản lý phòng ban
Tác nhân	Quản trị viên (admin)
Mục đích	Thêm, sửa, xóa phòng ban.
Điều kiện trước	- Quản trị viên đã đăng nhập (vai trò admin). - Quản trị viên truy cập trang quản lý phòng ban (<i>/admin/departments</i>).
Luồng chính: - Quản trị viên nhấp vào mục “Phòng ban” - Hệ thống gửi yêu cầu GET đến API <i>/departments</i> để lấy danh sách phòng ban. - Hệ thống hiển thị bảng phòng ban với các cột: ○ Tên phòng ban. ○ Số lượng người dùng. ○ Số lượng mẫu biểu. - Để thêm phòng ban: ○ Nhấp nút “Thêm mới” ○ Hiển thị form với trường “Tên phòng ban”. ○ Quản trị viên nhập tên và nhấp “Lưu”. ○ Hệ thống gửi yêu cầu POST đến API <i>/departments</i>. - Để sửa phòng ban: ○ Nhấp nút “Sửa” trên một phòng ban. ○ Hiển thị form với tên hiện tại.	

○ Quản trị viên chỉnh sửa và nhấp “Lưu”. ○ Hệ thống gửi yêu cầu PUT đến API <i>/departments/:id</i>. - Để xóa phòng ban: ○ Nhấp nút “Xóa” và xác nhận. ○ Hệ thống gửi yêu cầu DELETE đến API <i>/departments/:id</i>. - Nếu API trả về thành công, hiển thị thông báo và làm mới danh sách.	
Luồng phụ: Không có.	
Ngoại lệ	- Tên phòng ban trùng: Hiển thị thông báo “Phòng ban đã tồn tại”. - Phòng ban đang sử dụng: Hiển thị thông báo “Không thể xóa phòng ban đang được sử dụng”. - Lỗi API: Hiển thị thông báo “Thao tác thất bại”.

4.2.7. Use Case 7: Quản lý mẫu biểu

Tên Use Case	Quản lý mẫu biểu
Tác nhân	Quản trị viên (admin), Cán bộ (staff)
Mục đích	Thêm, sửa, xóa, xem trước mẫu biểu.
Điều kiện trước	- Người dùng đã đăng nhập (vai trò admin hoặc staff). - Truy cập trang quản lý mẫu biểu (<i>/admin/form-templates</i> hoặc <i>/staff/form</i>).
Luồng chính: - Người dùng nhấp vào mục “Mẫu biểu” - Hệ thống gửi yêu cầu GET đến: ○ <i>/form-templates</i> (admin). ○ <i>/form-templates/department/:departmentId</i> (staff). - Hệ thống hiển thị bảng mẫu biểu với các cột: ○ Tiêu đề. ○ Phòng ban. ○ Ngày tạo. - Để thêm mẫu biểu: ○ Nhấp nút “Thêm mới” Hiển thị form với các trường: Tiêu đề,	

Phòng ban (admin chọn, staff mặc định), File HTML. ○ Người dùng nhập thông tin và nhấp “Lưu”. ○ Hệ thống gửi yêu cầu POST đến API <i>/form-templates</i>.	
- Để sửa mẫu biểu: ○ Nhấp nút “Sửa” trên một mẫu biểu. ○ Hiện thị form với thông tin hiện tại. ○ Người dùng chỉnh sửa và nhấp “Lưu”. ○ Hệ thống gửi yêu cầu PUT đến API <i>/form-templates/:id</i>.	
- Để xem trước: ○ Nhấp nút “Xem trước”. ○ Hiện thị nội dung HTML.	
- Để xóa: ○ Nhấp nút “Xóa” và xác nhận. ○ Hệ thống gửi yêu cầu DELETE đến API <i>/form-templates/:id</i>.	
- Nếu API trả về thành công, hiện thị thông báo và làm mới danh sách.	
Luồng phụ: Không có.	
Ngoại lệ	- File HTML không hợp lệ: Hiện thị thông báo “File không đúng định dạng”. - Mẫu biểu đang sử dụng: Hiện thị thông báo “Không thể xóa mẫu biểu có hồ sơ”. - Lỗi API: Hiện thị thông báo “Thao tác thất bại”.

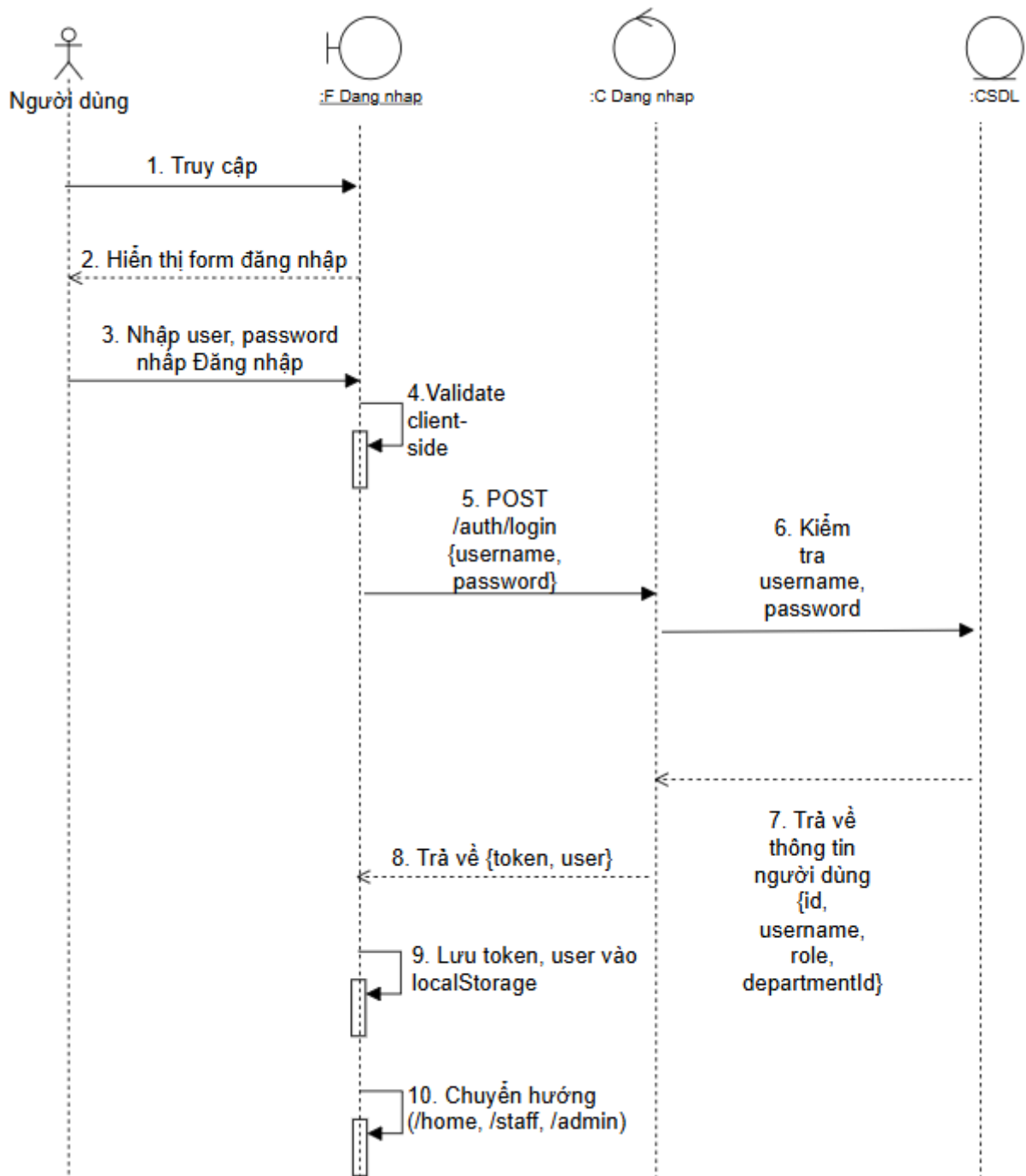
5. BIỂU ĐỒ TUẦN TỰ

5.1. Mô tả tổng quát

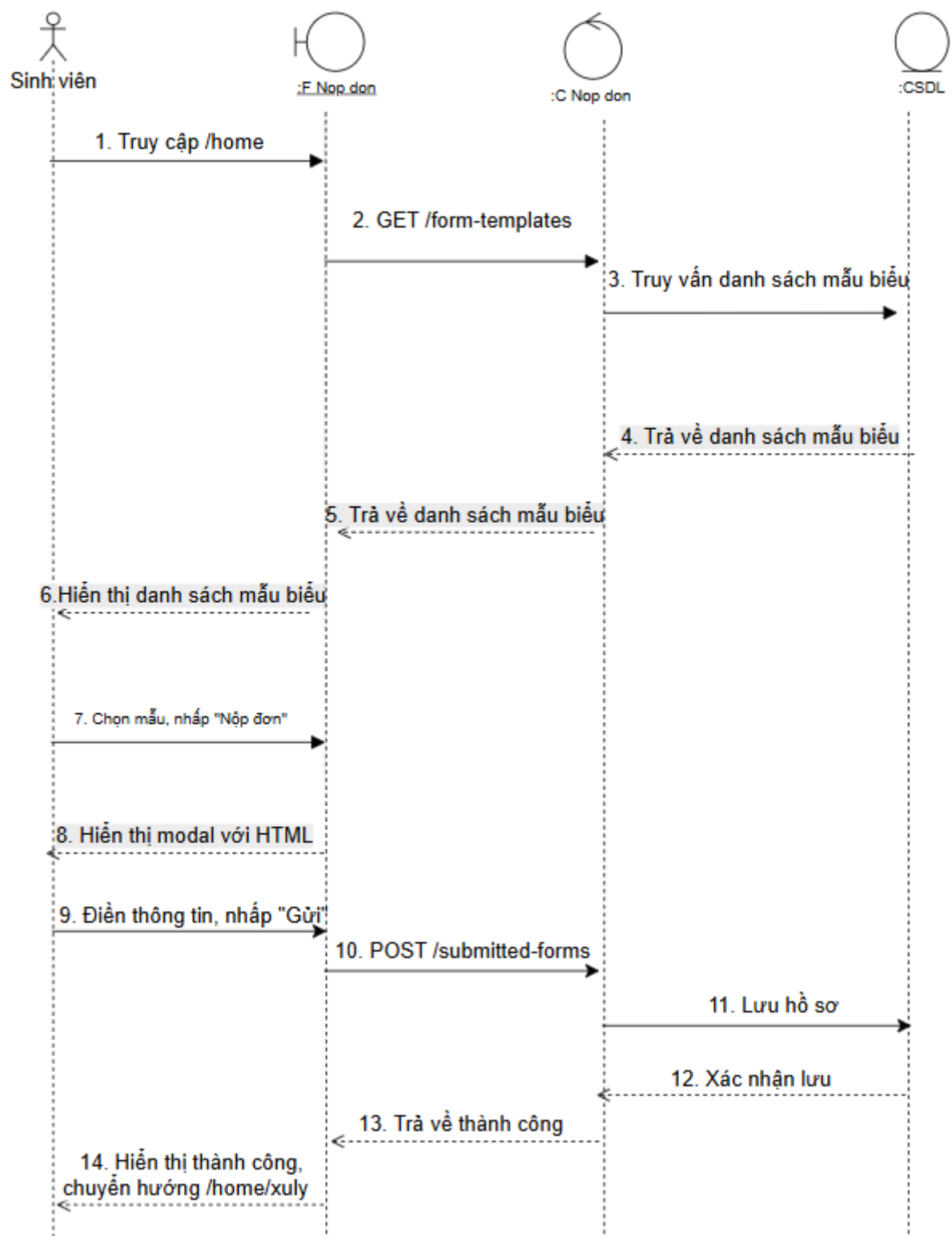
- **Mục đích:** Biểu đồ tuần tự thể hiện thứ tự các thông điệp (message) được gửi giữa các đối tượng trong hệ thống để thực hiện một chức năng cụ thể, giúp minh họa rõ ràng luồng xử lý từ khi người dùng bắt đầu tương tác đến khi nhận được kết quả.
- Thành phần tham gia:
 - Tác nhân (Actor): Sinh viên, Cán bộ, Quản trị viên.
 - Giao diện (Frontend): Các thành phần giao diện người dùng được xây dựng bằng ReactJS và Ant Design.
 - API (Backend): Các dịch vụ API xử lý yêu cầu từ frontend và tương tác với cơ sở dữ liệu.

- Cơ sở dữ liệu (Database): MySQL, lưu trữ thông tin về người dùng (Users), phòng ban (Departments), mẫu biểu (FormTemplates), và hồ sơ (SubmittedForms).
- **Chức năng chính:**
 - Đăng nhập.
 - Nộp đơn.
 - Theo dõi trạng thái hồ sơ.
 - Xử lý hồ sơ.
 - Quản lý tài khoản người dùng.
 - Quản lý phòng ban.
 - Quản lý mẫu biểu.
- **Nguyên tắc xây dựng:**
 - Mỗi biểu đồ tuần tự mô tả một chức năng cụ thể, từ khi tác nhân gửi yêu cầu đến khi hệ thống trả về kết quả.
 - Các thông điệp được biểu diễn bằng mũi tên:
 - Gọi hàm (Synchronous): Mũi tên đặc (→) cho các yêu cầu chờ phản hồi (như gọi API).
 - Phản hồi (Return): Mũi tên đứt đoạn (-->), trả về kết quả.
 - Tự gọi (Self-call): Nếu một đối tượng tự xử lý nội bộ.

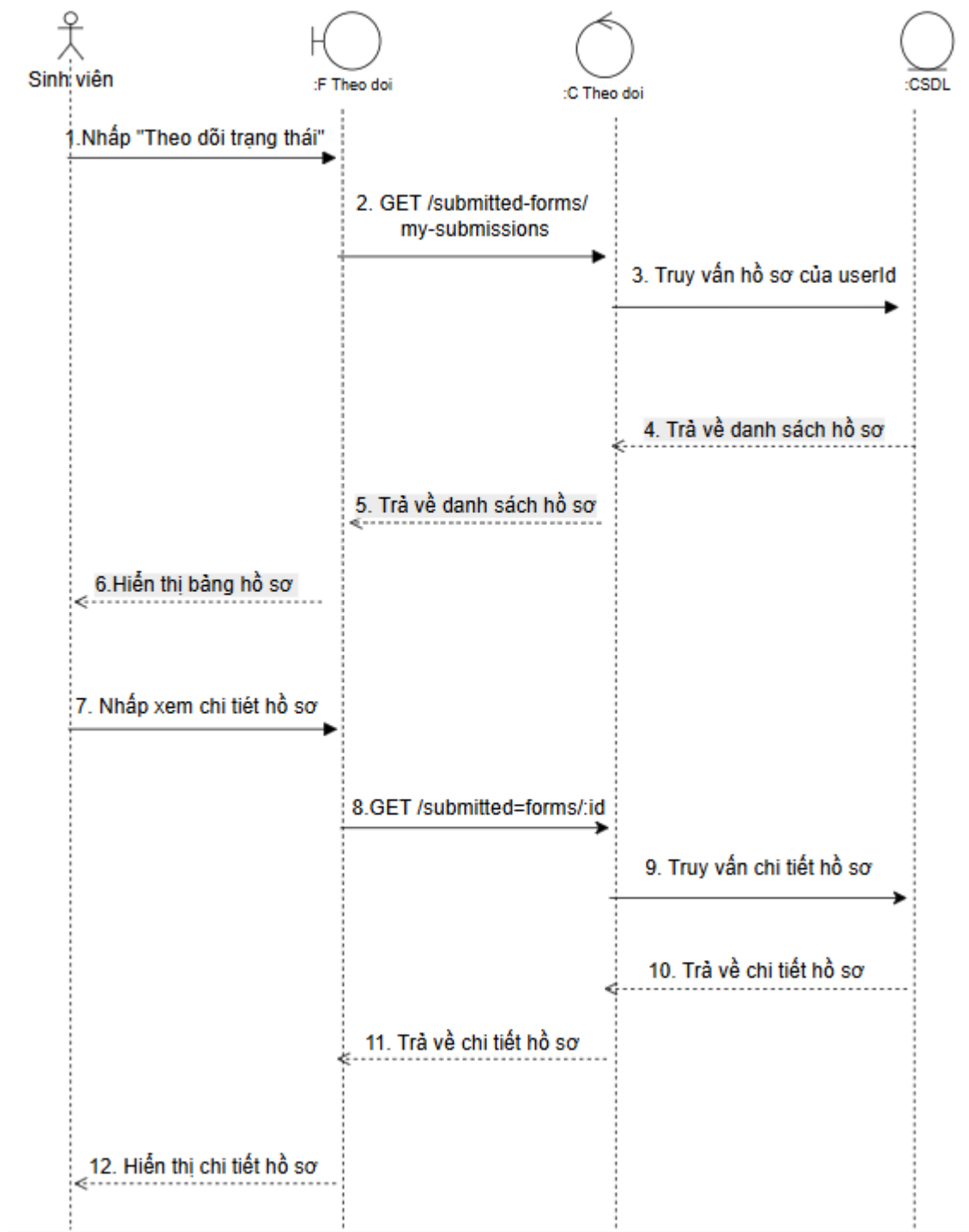
5.2. Biểu đồ tuần tự chi tiết



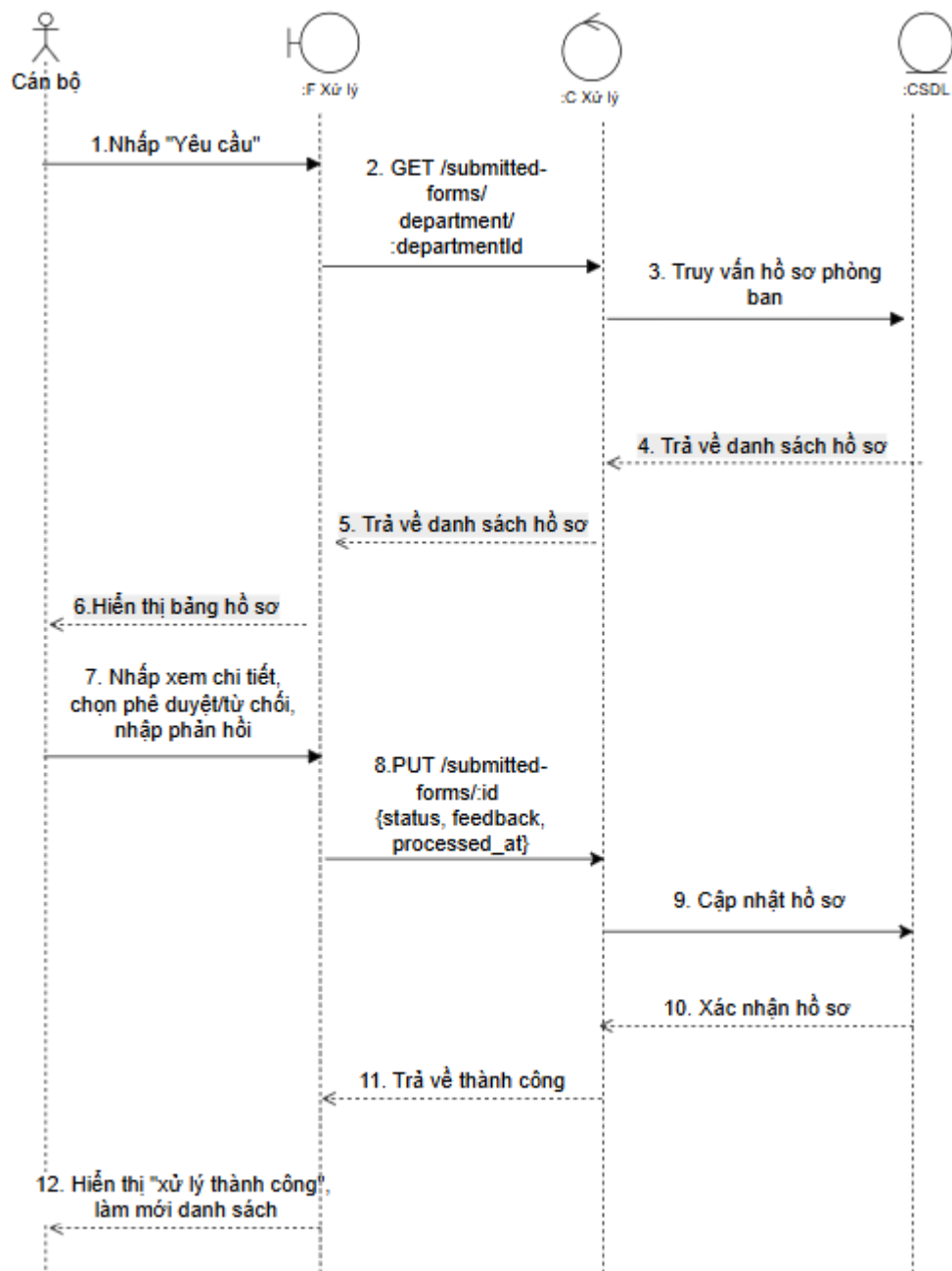
Hình 2. 5: Biểu đồ tuần tự thực thi “Đăng nhập”



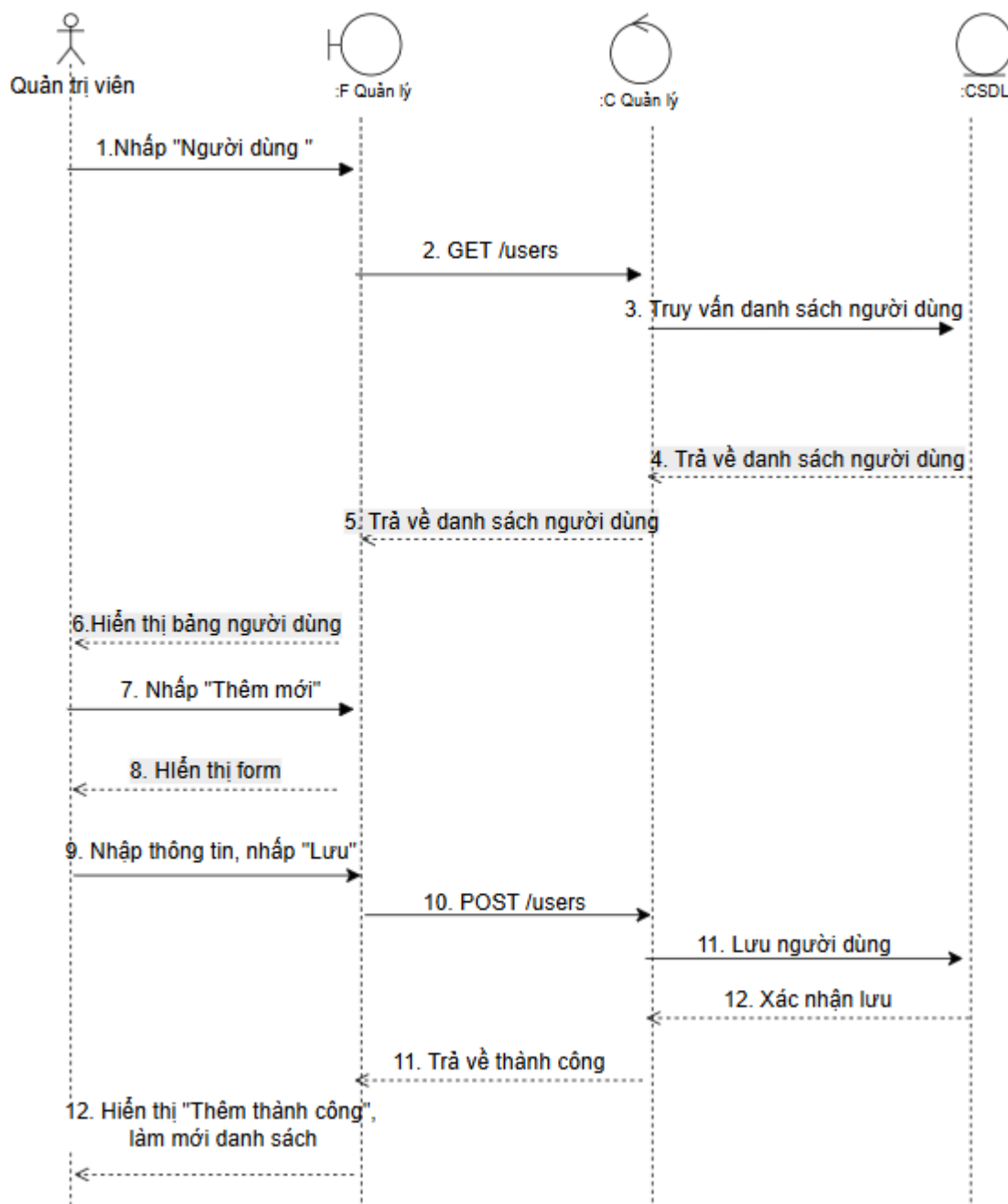
Hình 2. 6: Biểu đồ tuần tự thực thi “Nộp đơn”



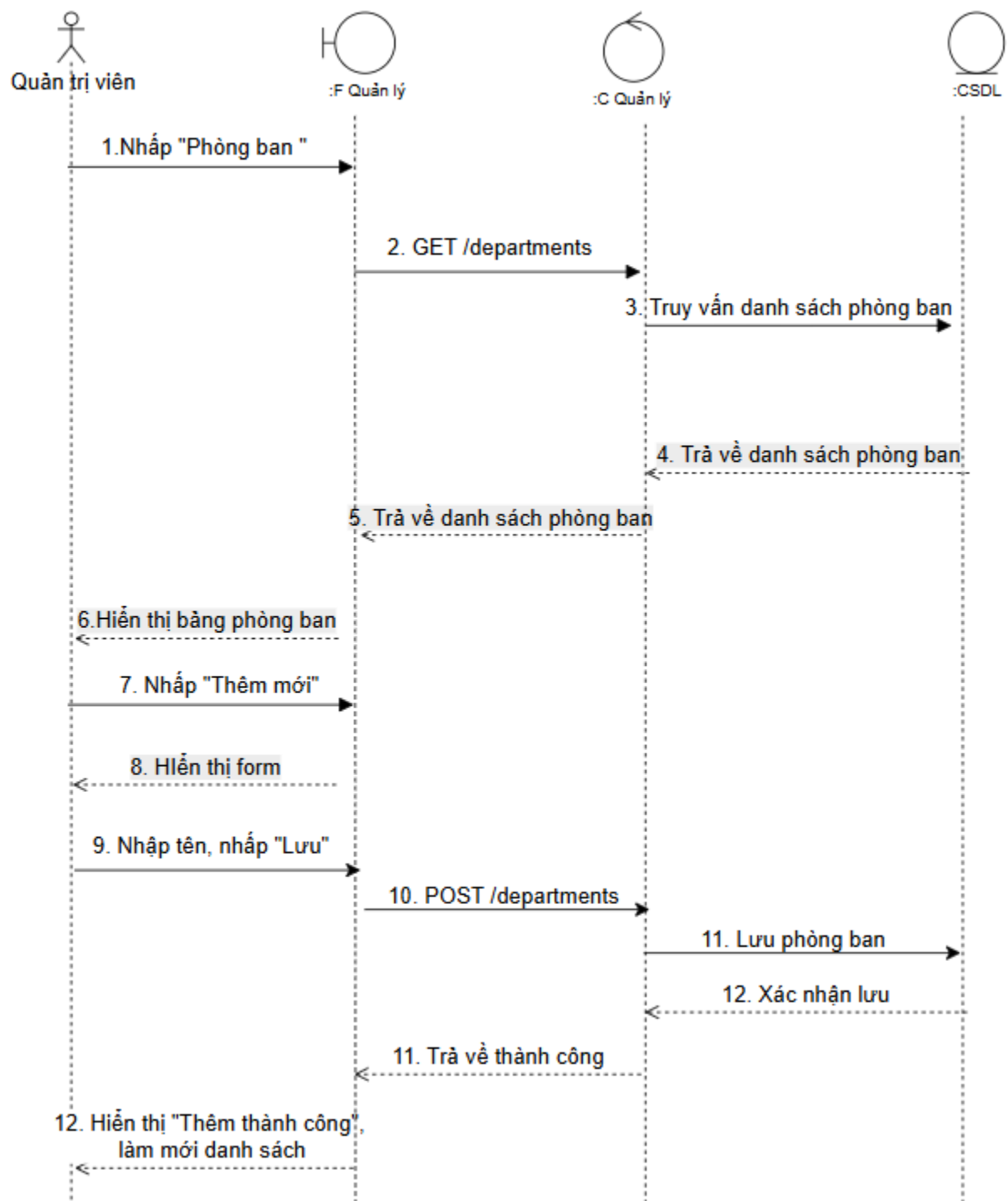
Hình 2. 7: Biểu đồ tuần tự thực thi “Theo dõi trạng thái hồ sơ”



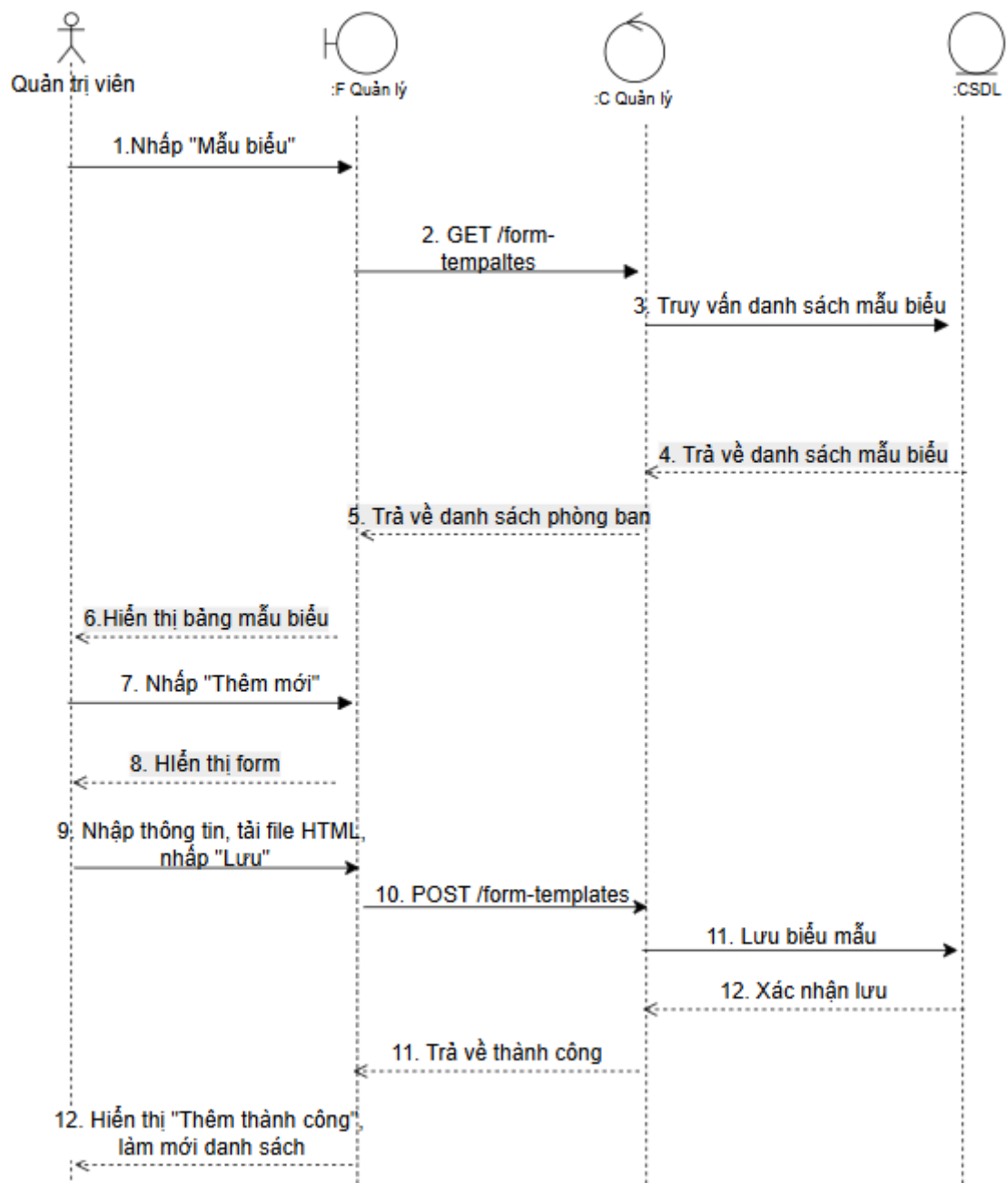
Hình 2. 8: Biểu đồ tuần tự thực thi “Xử lý hồ sơ”



Hình 2. 9: Biểu đồ tuần tự thực thi “Quản lý tài khoản người dùng”

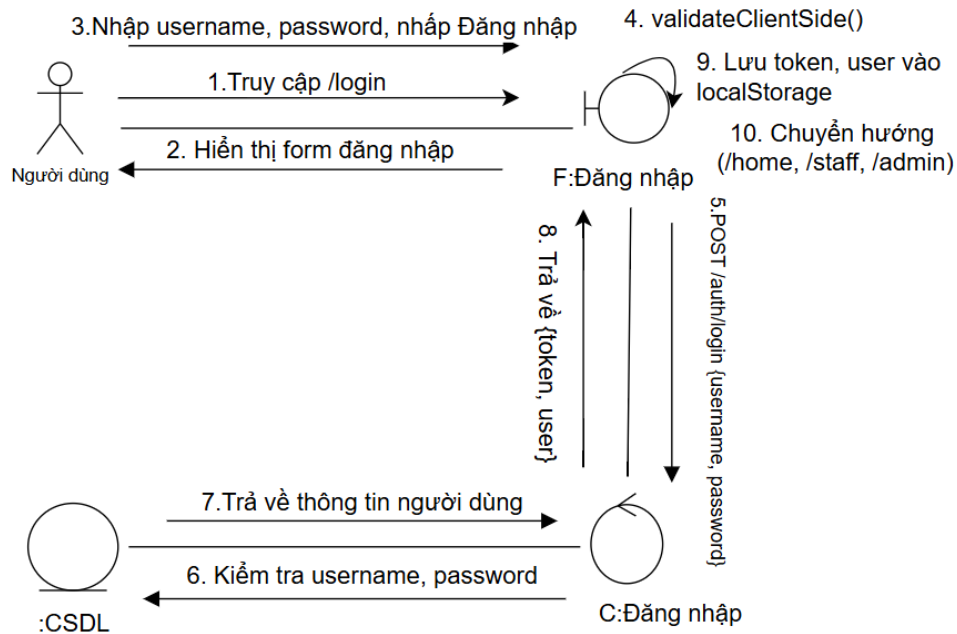


Hình 2. 10: Biểu đồ tuần tự thực thi “Quản lý phòng ban”

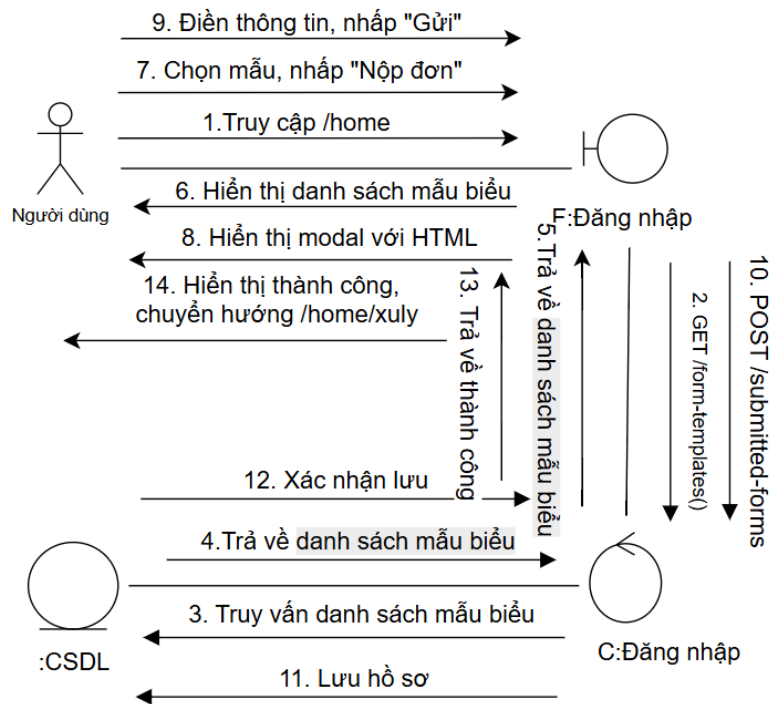


Hình 2. 11: Biểu đồ tuần tự thực thi “Quản lý mẫu biểu”

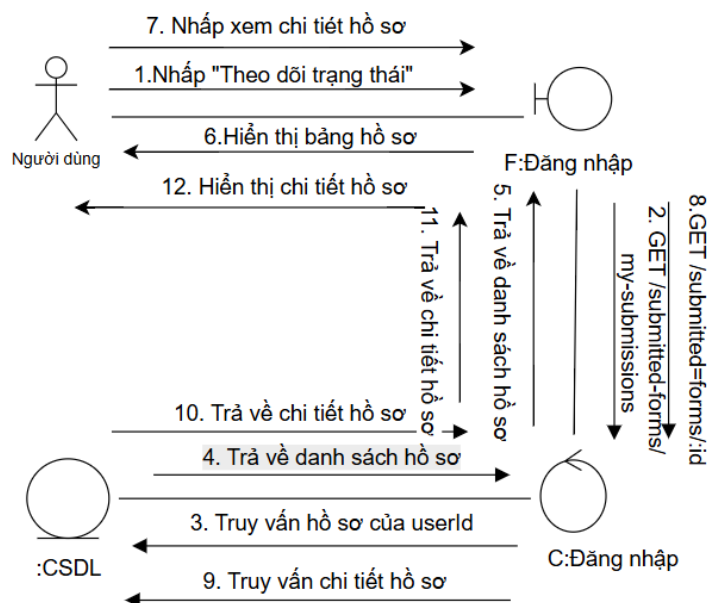
6. BIỂU ĐỒ CÔNG TÁC THỰC THI



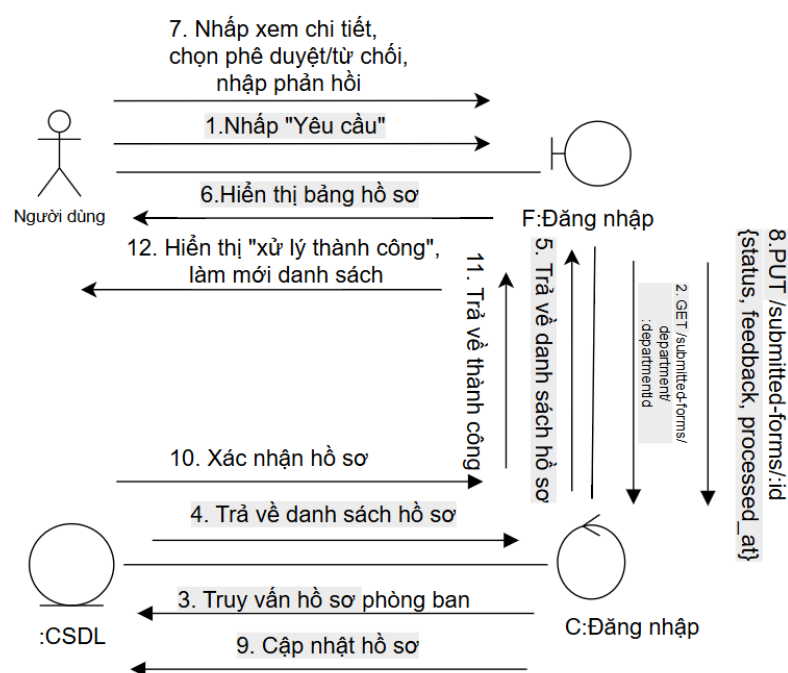
Hình 2. 12: Biểu đồ công tác thực thi Use Case “Đăng nhập”



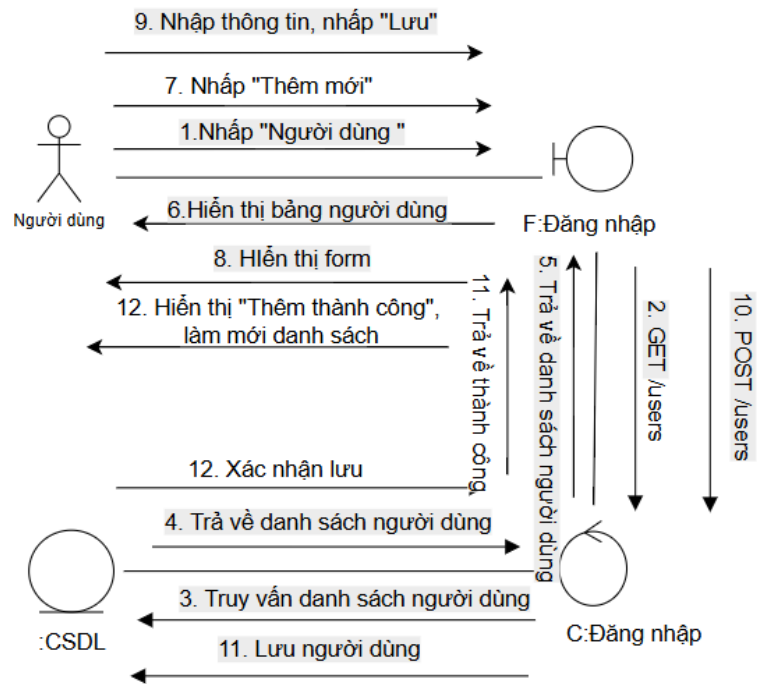
Hình 2. 13: Biểu đồ công tác thực thi Use Case “Nộp đơn”



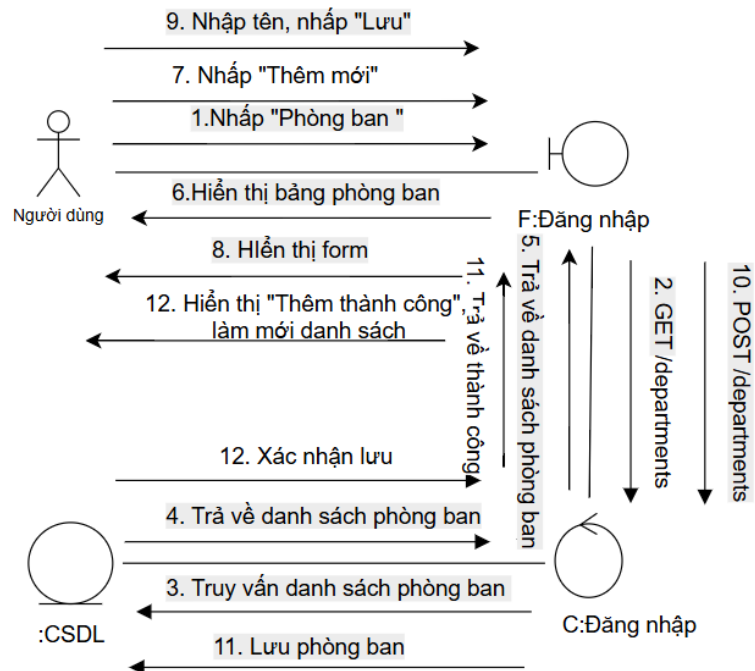
Hình 2. 14: Biểu đồ công tác thực thi Use Case “Theo dõi trạng thái hồ sơ”



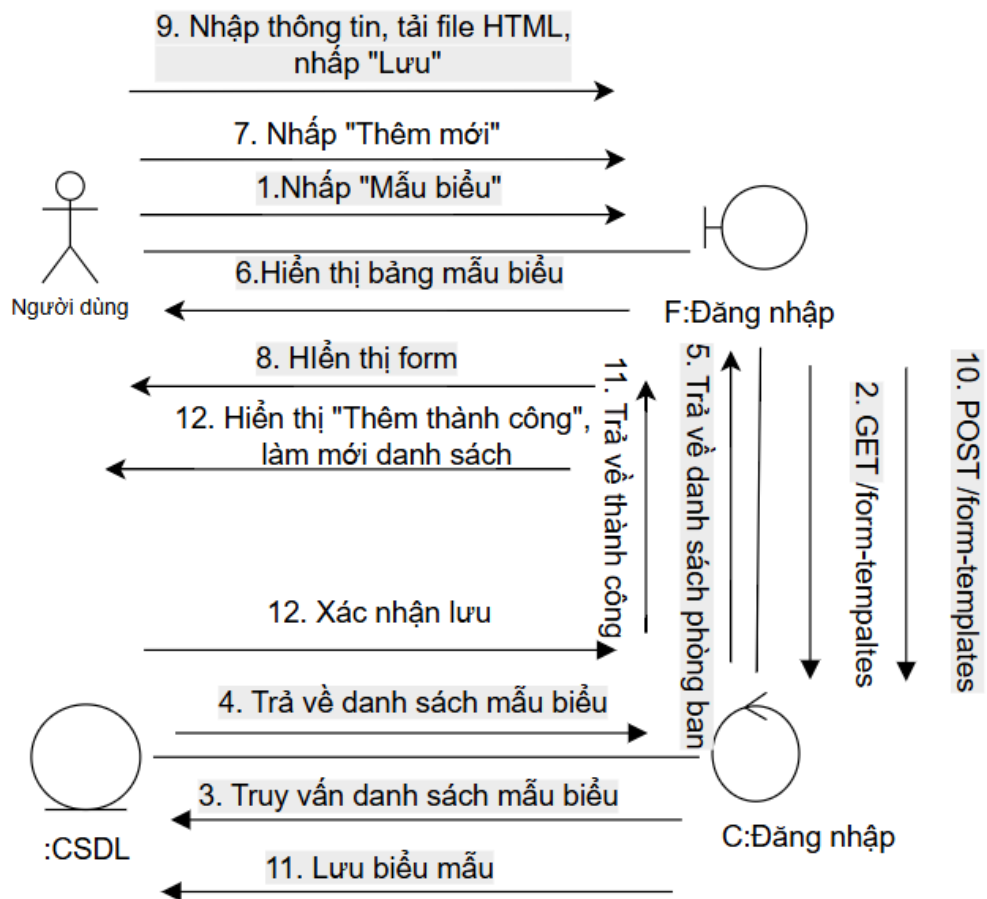
Hình 2. 15: Biểu đồ công tác thực thi Use Case “Xử lý hồ sơ”



Hình 2. 16: Biểu đồ công tác thực thi Use Case “Quản lý tài khoản người dùng”



Hình 2. 17: Biểu đồ công tác thực thi Use Case “Quản lý phòng ban”

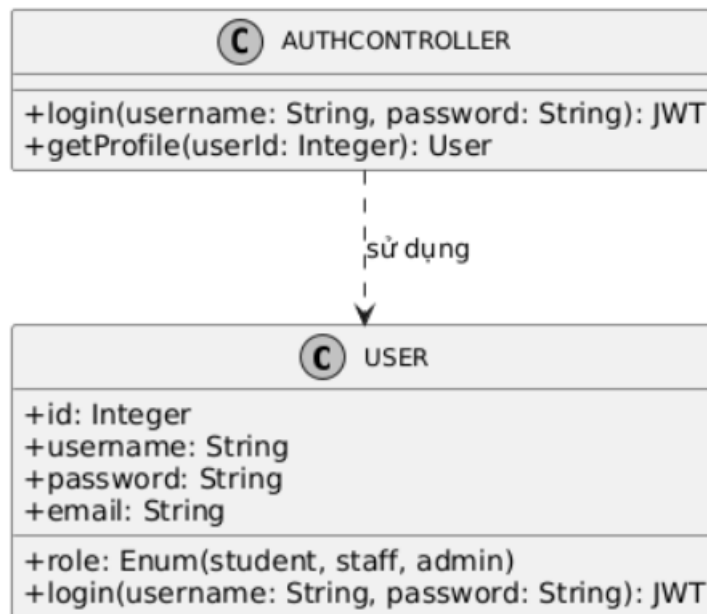


Hình 2. 18: Biểu đồ công tác thực thi Use Case “Quản lý mẫu biểu”

7. BIỂU ĐỒ LỚP

Biểu đồ lớp (Class Diagram) là một công cụ thiết kế hướng đối tượng, mô tả cấu trúc tĩnh của hệ thống thông qua các lớp, thuộc tính, phương thức, và mối quan hệ giữa chúng. Trong hệ thống quản lý hành chính online dành cho sinh viên, biểu đồ lớp đóng vai trò cầu nối giữa các biểu đồ Use Case, tuần tự, cộng tác (mô tả luồng tương tác) và mô hình quan hệ (mô tả cơ sở dữ liệu), giúp xác định các thực thể chính và cách chúng phối hợp.

7.1. Mô hình lớp “Đăng nhập”



Hình 2. 19: Mô hình lớp “Đăng nhập”

Mô hình lớp “Đăng nhập” mô tả các thành phần liên quan đến chức năng xác thực người dùng (sinh viên, cán bộ, quản trị viên) trong hệ thống. Chức năng này cho phép người dùng nhập thông tin đăng nhập (tên đăng nhập, mật khẩu) để truy cập hệ thống với vai trò tương ứng.

Mô tả chi tiết

Các lớp chính:

a. User:

- Thuộc tính:

- + id: Integer: Mã định danh duy nhất.
- + username: String: Tên đăng nhập của người dùng.
- + password: String: Mật khẩu (lưu ý: trong thực tế, mật khẩu thường được băm trước khi lưu).

- + email: String: Địa chỉ email của người dùng.
- + role: Enum(student, staff, admin): Vai trò người dùng (sinh viên, nhân viên, quản trị viên).
- **Phương thức:**
 - + login(username: String, password: String): JWT: Xác thực người dùng và trả về mã JWT (JSON Web Token) để sử dụng trong các yêu cầu bảo mật.

b. AuthController

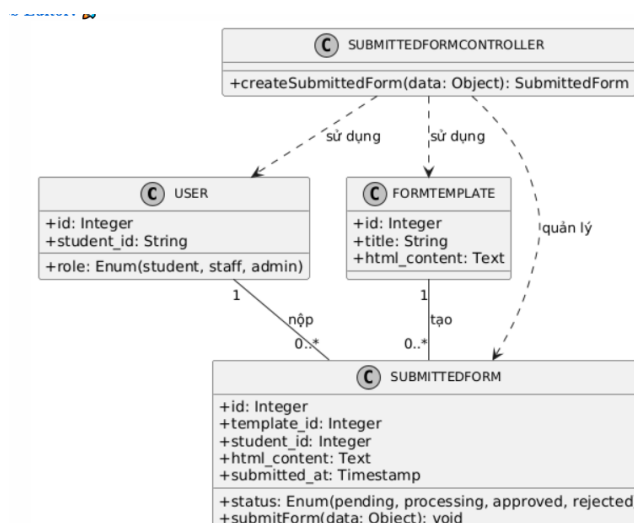
- **Mô tả:** Lớp điều khiển quản lý các thao tác xác thực và truy xuất thông tin hồ sơ người dùng.
- **Phương thức:**
 - + login(username: String, password: String): JWT: Xác thực người dùng và trả về mã JWT.
 - + getProfile(userId: Integer): User: Truy xuất thông tin hồ sơ người dùng theo userId.

❖ Mối quan hệ

- **AuthController ..> User : sử dụng**
 - + Ý nghĩa: AuthController sử dụng lớp User (mối quan hệ phụ thuộc), để xác thực (login) hoặc truy xuất thông tin (getProfile).
 - + Ký hiệu: ..> (đường nét đứt) biểu thị sự phụ thuộc.

7.2. Mô hình lớp “Nộp đơn hành chính”

Mô hình lớp Nộp đơn hành chính mô tả các thành phần liên quan đến chức năng sinh viên nộp hồ sơ hành chính qua hệ thống. Sinh viên chọn mẫu biểu, điền thông tin, và nộp hồ sơ, được mô tả trong Use Case 2.



Hình 2. 20: Mô hình lớp “Nộp đơn hành chính”

a. User

- **Mô tả:** Lớp User đại diện cho người dùng trong hệ thống, có thể là sinh viên, nhân viên hoặc quản trị viên.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất cho mỗi người dùng.
 - + role: Enum(student, staff, admin): Vai trò của người dùng, sử dụng kiểu liệt kê (enum) với ba giá trị: student (sinh viên), staff (nhân viên), admin (quản trị viên).
 - + student_id: String: Mã số sinh viên, có thể chỉ áp dụng cho người dùng có vai trò student. (Lưu ý: Thuộc tính này có thể dư thừa nếu không phải tất cả người dùng đều cần student_id.)

b. FormTemplate

- **Mô tả:** Lớp FormTemplate đại diện cho một mẫu biểu mẫu được định nghĩa sẵn, có thể được sử dụng để tạo ra các biểu mẫu nộp bởi người dùng.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất cho mỗi mẫu biểu mẫu.
 - + title: String: Tiêu đề của biểu mẫu (ví dụ: "Đơn xin nghỉ học").
 - + html_content: Text: Nội dung HTML định nghĩa cấu trúc và giao diện của biểu mẫu.

c. SubmittedForm

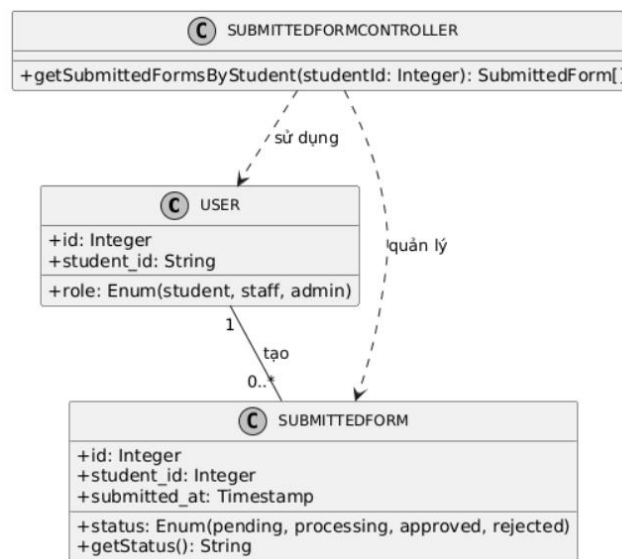
- **Mô tả:** Lớp SubmittedForm đại diện cho một biểu mẫu mà người dùng (chủ yếu là sinh viên) đã nộp dựa trên một mẫu biểu mẫu.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất cho biểu mẫu đã nộp.
 - + template_id: Integer: Khóa ngoại liên kết với FormTemplate, chỉ ra mẫu biểu mẫu được sử dụng.
 - + student_id: Integer: Khóa ngoại liên kết với User, chỉ ra người dùng (sinh viên) đã nộp biểu mẫu.
 - + html_content: Text: Nội dung HTML của biểu mẫu đã nộp, có thể chứa dữ liệu người dùng nhập vào (dựa trên html_content của FormTemplate).
 - + status: Enum(pending, processing, approved, rejected): Trạng thái của biểu mẫu, với bốn giá trị:
 - + pending: Đang chờ xử lý.
 - + processing: Đang được xử lý.

- + approved: Đã được phê duyệt.
- + rejected: Bị từ chối.
- + submitted_at: Timestamp: Thời gian biểu mẫu được nộp.
- **Phương thức:**
 - + submitForm(data: Object): void: Phương thức để nộp biểu mẫu, nhận dữ liệu (dưới dạng một đối tượng) và không trả về giá trị.
- d. SubmittedFormController**
- **Mô tả:** Lớp SubmittedFormController là một lớp điều khiển, chịu trách nhiệm quản lý việc tạo và xử lý các biểu mẫu đã nộp.
- **Phương thức:**
 - + createSubmittedForm(data: Object): SubmittedForm: Tạo một biểu mẫu đã nộp mới dựa trên dữ liệu đầu vào và trả về đối tượng SubmittedForm.
- ❖ **Mối quan hệ**
- **User "1" -- "0..*" SubmittedForm : nộp**
 - + Ý nghĩa: Một người dùng (User) có thể nộp 0 hoặc nhiều biểu mẫu (SubmittedForm). Mối quan hệ này là một-nhiều (1-to-many).
 - + Ký hiệu: "1" -- "0..*", trong đó 1 là một người dùng, 0..* là số lượng biểu mẫu (từ 0 đến nhiều).
- **FormTemplate "1" -- "0..*" SubmittedForm : tạo**
 - + Ý nghĩa: Một mẫu biểu mẫu (FormTemplate) có thể được sử dụng để tạo ra 0 hoặc nhiều biểu mẫu đã nộp (SubmittedForm). Đây cũng là mối quan hệ một-nhiều.
 - + Ký hiệu: Tương tự như trên.
- **SubmittedFormController ..> SubmittedForm : quản lý**
 - + Ý nghĩa: Bộ điều khiển (SubmittedFormController) quản lý các biểu mẫu đã nộp (SubmittedForm). Đây là mối quan hệ phụ thuộc (dependency), nghĩa là bộ điều khiển sử dụng hoặc thao tác trên các đối tượng SubmittedForm.
 - + Ký hiệu: ..> là đường nét đứt, biểu thị sự phụ thuộc.
- **SubmittedFormController ..> User : uses**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin từ lớp User, có thể để xác minh người nộp biểu mẫu hoặc lấy thông tin liên quan.
 - + Ký hiệu: Cũng là mối quan hệ phụ thuộc.
- **SubmittedFormController ..> FormTemplate : uses**

- + Ý nghĩa: Bộ điều khiển sử dụng các mẫu biểu mẫu (FormTemplate), ví dụ để lấy cấu trúc HTML khi tạo biểu mẫu đã nộp.
- + Ký hiệu: Mỗi quan hệ phụ thuộc.

7.3. Mô hình lớp “Theo dõi trạng thái hồ sơ”

Mô hình lớp này mô tả chức năng sinh viên theo dõi trạng thái hồ sơ đã nộp (Use Case 3). Các lớp liên quan bao gồm User (sinh viên), SubmittedForm (hồ sơ), và SubmittedFormController (xử lý truy vấn trạng thái).



Hình 2. 21: Mô hình lớp “Theo dõi trạng thái hồ sơ”

a. User

- Thuộc tính:

- + id: Integer: Mã định danh duy nhất cho người dùng.
- + role: Enum(student, staff, admin): Vai trò của người dùng, với ba giá trị: student, staff, admin.
- + student_id: String: Mã số sinh viên, có thể chỉ áp dụng cho người dùng có vai trò student.

b. SubmittedForm

- **Mô tả:** Đại diện cho một biểu mẫu đã nộp bởi người dùng (sinh viên).
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất cho biểu mẫu.
 - + student_id: Integer: Khóa ngoại liên kết với User, chỉ ra sinh viên nộp biểu mẫu.

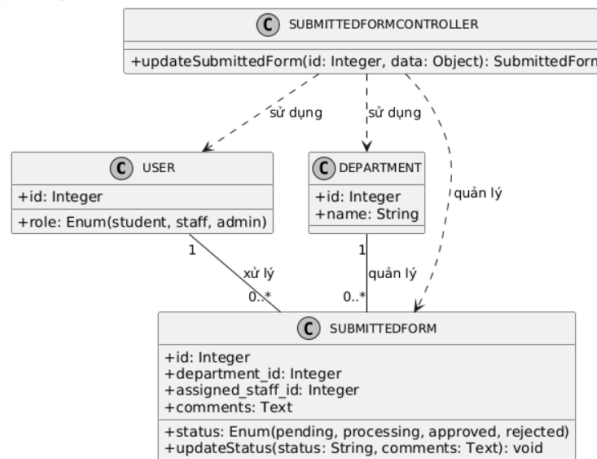
- + status: Enum(pending, processing, approved, rejected): Trạng thái của biểu mẫu (chờ xử lý, đang xử lý, được phê duyệt, bị từ chối).
- + submitted_at: Timestamp: Thời gian nộp biểu mẫu.
- **Phương thức:**
 - + getStatus(): String: Trả về trạng thái của biểu mẫu dưới dạng chuỗi (có thể để hiển thị trạng thái trong giao diện).
- c. SubmittedFormController**
- **Mô tả:** Lớp điều khiển chịu trách nhiệm quản lý và truy xuất các biểu mẫu đã nộp.
- **Phương thức:**
 - + getSubmittedFormsByStudent(studentId: Integer): SubmittedForm[]: Truy xuất danh sách các biểu mẫu đã nộp dựa trên studentId, trả về một mảng các đối tượng SubmittedForm.

❖ **Mối quan hệ giữa các lớp**

- **User "1" -- "0..*" SubmittedForm : tạo**
 - + Ý nghĩa: Một người dùng (User) có thể tạo 0 hoặc nhiều biểu mẫu đã nộp (SubmittedForm). Đây là mối quan hệ một-nhiều.
 - + Ký hiệu: "1" -- "0..*", với 1 là một người dùng và 0..* là số lượng biểu mẫu.
- **SubmittedFormController ..> SubmittedForm : quản lý**
 - + Ý nghĩa: Bộ điều khiển quản lý các biểu mẫu đã nộp (mối quan hệ phụ thuộc).
 - + Ký hiệu: ..> (đường nét đứt) biểu thị sự phụ thuộc.
- **SubmittedFormController ..> User : sử dụng**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin từ User, có thể để kiểm tra hoặc lấy dữ liệu người dùng.
 - + Ký hiệu: Mối quan hệ phụ thuộc.

7.4. Mô hình lớp “Xử lý hồ sơ”

Mô hình lớp này mô tả chức năng cán bộ xử lý hồ sơ (phê duyệt, từ chối, thêm ghi chú) (Use Case 4). Các lớp liên quan bao gồm User (cán bộ), SubmittedForm (hồ sơ), Department (phòng ban), và SubmittedFormController (xử lý cập nhật).



Hình 2. 22: Mô hình lớp “Xử lý hồ sơ”

a. User

- **Mô tả:** Đại diện cho người dùng trong hệ thống (sinh viên, nhân viên, hoặc quản trị viên).
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + role: Enum(student, staff, admin): Vai trò người dùng.

b. Department (Phòng ban)

- **Mô tả:** Đại diện cho một phòng ban trong hệ thống, ví dụ: phòng đào tạo, phòng nhân sự.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + name: String: Tên phòng ban.

c. SubmittedForm (Biểu mẫu đã nộp)

- **Mô tả:** Đại diện cho biểu mẫu đã nộp, với thông tin về trạng thái, phòng ban, nhân viên được gán, và bình luận.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + status: Enum(pending, processing, approved, rejected): Trạng thái biểu mẫu (chờ xử lý, đang xử lý, được phê duyệt, bị từ chối).
 - + department_id: Integer: Khóa ngoại liên kết với Department, chỉ phòng ban quản lý biểu mẫu.
 - + assigned_staff_id: Integer: Khóa ngoại liên kết với User (nhân viên hoặc quản trị viên) được gán xử lý biểu mẫu.
 - + comments: Text: Bình luận hoặc ghi chú liên quan đến biểu mẫu.
- **Phương thức:**

- + `updateStatus(status: String, comments: Text): void`: Cập nhật trạng thái và bình luận của biểu mẫu.

d. SubmittedFormController (Bộ điều khiển biểu mẫu đã nộp)

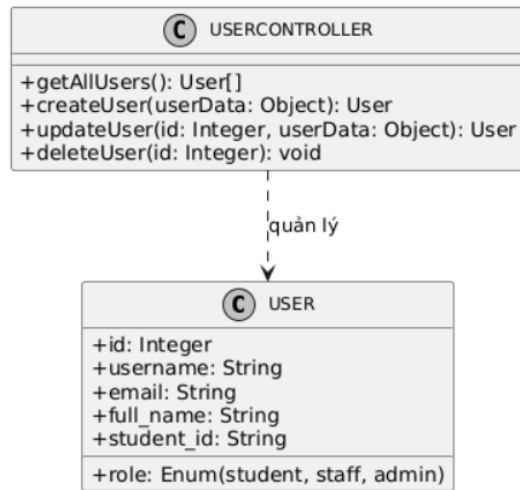
- Mô tả: Lớp điều khiển quản lý việc cập nhật biểu mẫu đã nộp.
- Phương thức:
 - + `updateSubmittedForm(id: Integer, data: Object): SubmittedForm`: Cập nhật thông tin biểu mẫu theo id, trả về đối tượng SubmittedForm đã cập nhật.

❖ Mọi quan hệ

- **User "1" -- "0..*" SubmittedForm : xử lý**
 - + Ý nghĩa: Một người dùng (có thể là nhân viên hoặc quản trị viên) xử lý 0 hoặc nhiều biểu mẫu (mọi quan hệ một-nhiều).
 - + Ký hiệu: "1" -- "0..*", với 1 là một người dùng, 0..* là số lượng biểu mẫu.
- **Department "1" -- "0..*" SubmittedForm : quản lý**
 - + Ý nghĩa: Một phòng ban quản lý 0 hoặc nhiều biểu mẫu (mọi quan hệ một-nhiều).
 - + Ký hiệu: "1" -- "0..*", với 1 là một phòng ban, 0..* là số lượng biểu mẫu.
- **SubmittedFormController ..> SubmittedForm : quản lý**
 - + Ý nghĩa: Bộ điều khiển quản lý các biểu mẫu đã nộp (mọi quan hệ phụ thuộc).
 - + Ký hiệu: ..> (đường nét đứt) biểu thị sự phụ thuộc.
- **SubmittedFormController ..> User : sử dụng**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin từ User, ví dụ để xác định nhân viên được gán (`assigned_staff_id`).
 - + Ký hiệu: Mọi quan hệ phụ thuộc.
- **SubmittedFormController ..> Department : sử dụng**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin từ Department, ví dụ để gán biểu mẫu cho phòng ban (`department_id`).
 - + Ký hiệu: Mọi quan hệ phụ thuộc.

7.5. Mô hình lớp “Quản lý người dùng”

Mô hình lớp này mô tả chức năng quản trị viên quản lý thông tin người dùng (thêm, sửa, xóa) (Use Case 5). Các lớp liên quan bao gồm User và UserController.



Hình 2. 23: Mô hình lớp “Quản lý người dùng”

a. User

- **Mô tả:** Đại diện cho người dùng trong hệ thống, có thể là sinh viên, nhân viên hoặc quản trị viên.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất cho người dùng.
 - + username: String: Tên đăng nhập của người dùng.
 - + email: String: Địa chỉ email của người dùng.
 - + role: Enum(student, staff, admin): Vai trò của người dùng (sinh viên, nhân viên, quản trị viên).
 - + full_name: String: Họ và tên đầy đủ của người dùng.
 - + student_id: String: Mã số sinh viên, áp dụng cho người dùng có vai trò student.

b. UserController

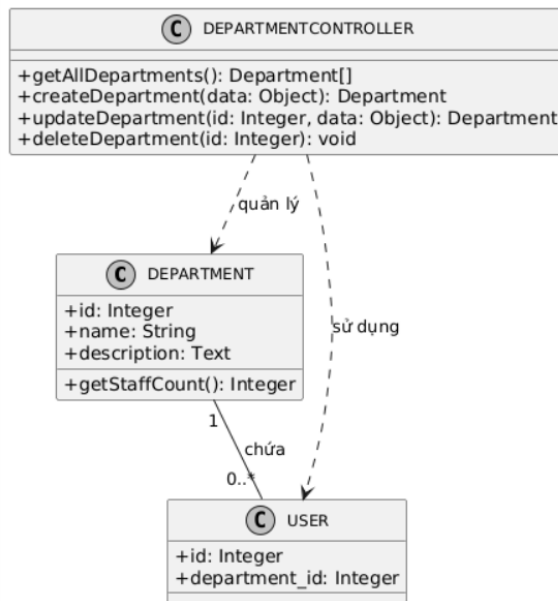
- **Mô tả:** Lớp điều khiển chịu trách nhiệm quản lý các thao tác CRUD đối với người dùng.
- **Phương thức:**
 - + getAllUsers(): User[]: Truy xuất danh sách tất cả người dùng, trả về mảng các đối tượng User.
 - + createUser(userData: Object): User: Tạo một người dùng mới dựa trên dữ liệu đầu vào, trả về đối tượng User.
 - + updateUser(id: Integer, userData: Object): User: Cập nhật thông tin người dùng theo id, trả về đối tượng User đã cập nhật.
 - + deleteUser(id: Integer): void: Xóa người dùng theo id, không trả về giá trị.

❖ **Mối quan hệ**

- UserController ..> User : quản lý
 - + Ý nghĩa: UserController quản lý lớp User, thể hiện mối quan hệ phụ thuộc (dependency). Bộ điều khiển sử dụng hoặc thao tác trên các đối tượng User để thực hiện các chức năng CRUD.
 - + Ký hiệu: ..> (đường nét đứt) biểu thị sự phụ thuộc.

7.6. Mô hình lớp “Quản lý phòng ban”

Mô hình lớp này mô tả chức năng quản trị viên quản lý thông tin phòng ban (thêm, sửa, xóa) (Use Case 6). Các lớp liên quan bao gồm Department, User (nhân viên thuộc phòng ban), và DepartmentController.



Hình 2. 24: Mô hình lớp “Quản lý phòng ban”

a. Department (Phòng ban)

- **Mô tả:** Đại diện cho một phòng ban trong tổ chức (ví dụ: phòng đào tạo, phòng nhân sự).
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + name: String: Tên phòng ban.
 - + description: Text: Mô tả chi tiết về phòng ban.
- **Phương thức:**
 - + getStaffCount(): Integer: Trả về số lượng nhân viên trong phòng ban.

b. User (Người dùng)

- **Mô tả:** Đại diện cho người dùng thuộc một phòng ban.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + department_id: Integer: Khóa ngoại liên kết với Department, chỉ ra phòng ban của người dùng.

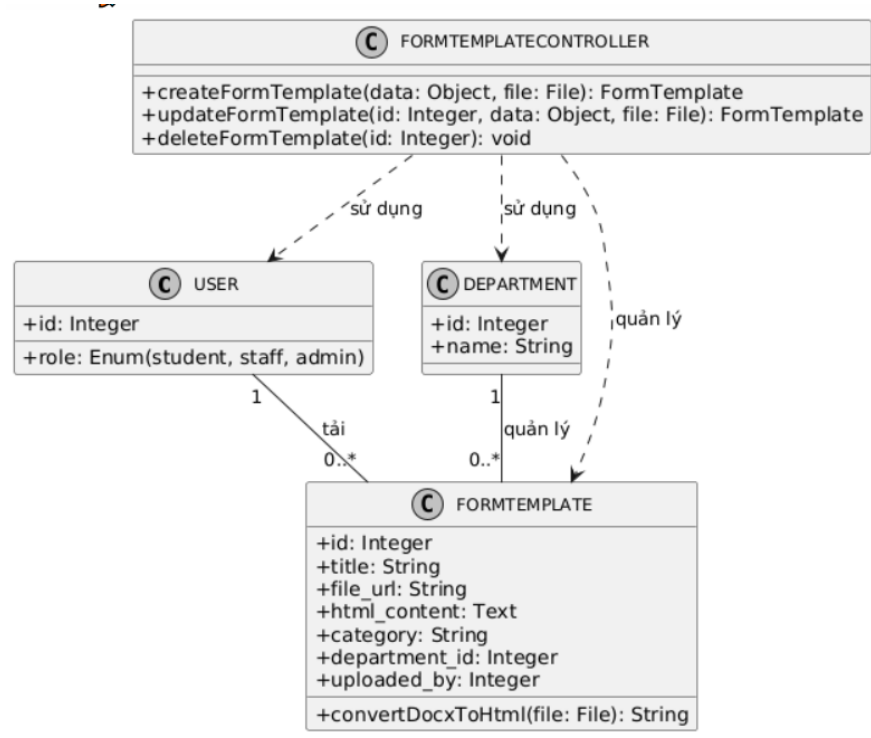
c. DepartmentController (Bộ điều khiển phòng ban)

- **Mô tả:** Lớp điều khiển quản lý các thao tác CRUD cho phòng ban.
- **Phương thức:**
 - + getAllDepartments(): Department[]: Truy xuất danh sách tất cả phòng ban.
 - + createDepartment(data: Object): Department: Tạo phòng ban mới.

- + updateDepartment(id: Integer, data: Object): Department: Cập nhật thông tin phòng ban theo id.
- + deleteDepartment(id: Integer): void: Xóa phòng ban theo id.
- ❖ **Mối quan hệ**
- **Department "1" -- "0..*" User : chứa**
 - + Ý nghĩa: Một phòng ban chứa 0 hoặc nhiều người dùng (mối quan hệ một-nhiều).
 - + Ký hiệu: "1" -- "0..*", với 1 là một phòng ban, 0..* là số lượng người dùng.
 - + Nhãn: contains (chứa), chỉ ra người dùng thuộc về phòng ban.
- **DepartmentController ..> Department : quản lý**
 - + Ý nghĩa: Bộ điều khiển quản lý các phòng ban (mối quan hệ phụ thuộc).
 - + Ký hiệu: ..> (đường nét đứt) biểu thị sự phụ thuộc.
- **DepartmentController ..> User : sử dụng**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin người dùng, có thể để kiểm tra hoặc gán người dùng vào phòng ban.
 - + Ký hiệu: Mối quan hệ phụ thuộc.

7.7. Mô hình lớp “Quản lý mẫu biểu”

Mô hình lớp này mô tả chức năng quản trị viên hoặc cán bộ quản lý mẫu biểu hành chính (tạo, sửa, xóa) (Use Case 7). Các lớp liên quan bao gồm User, Department, FormTemplate, và FormTemplateController.



Hình 2. 25: Mô hình lớp “Quản lý mẫu biểu”

a. User (Người dùng)

- **Mô tả:** Đại diện cho người dùng (sinh viên, nhân viên, hoặc quản trị viên).
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + role: Enum(student, staff, admin): Vai trò người dùng.

b. Department (Phòng ban)

- **Mô tả:** Đại diện cho phòng ban (ví dụ: phòng đào tạo, nhân sự).
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + name: String: Tên phòng ban.

c. FormTemplate (Mẫu biểu mẫu)

- **Mô tả:** Đại diện cho mẫu biểu mẫu được tải lên và quản lý.
- **Thuộc tính:**
 - + id: Integer: Mã định danh duy nhất.
 - + title: String: Tiêu đề biểu mẫu.
 - + file_url: String: URL của tệp biểu mẫu (ví dụ: tệp DOCX).
 - + html_content: Text: Nội dung HTML của biểu mẫu.
 - + category: String: Danh mục của biểu mẫu (ví dụ: đăng ký học, xin nghỉ).
 - + department_id: Integer: Khóa ngoại liên kết với Department.
 - + uploaded_by: Integer: Khóa ngoại liên kết với User (người tải lên).
- **Phương thức:**
 - + convertDocxToHtml(file: File): String: Chuyển đổi tệp DOCX thành HTML.

d. FormTemplateController (Bộ điều khiển mẫu biểu mẫu)

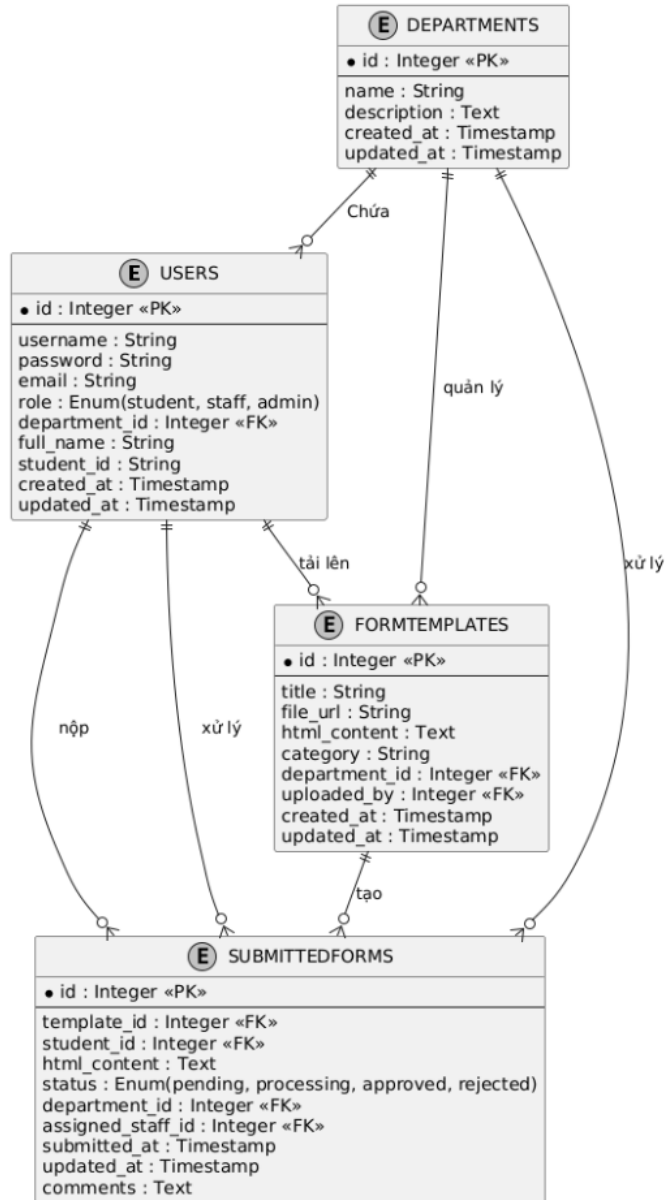
- **Mô tả:** Quản lý các thao tác CRUD cho mẫu biểu mẫu.
- **Phương thức:**
 - + createFormTemplate(data: Object, file: File): FormTemplate: Tạo mẫu biểu mẫu mới.
 - + updateFormTemplate(id: Integer, data: Object, file: File): FormTemplate: Cập nhật mẫu biểu mẫu.
 - + deleteFormTemplate(id: Integer): void: Xóa mẫu biểu mẫu.

❖ Mối quan hệ

- **User "1" -- "0..*" FormTemplate : tải**

- + Ý nghĩa: Một người dùng tải lên 0 hoặc nhiều mẫu biểu mẫu (mối quan hệ một-nhiều).
- + Ký hiệu: "1" -- "0..*".
- + Nhãn: tải (tải lên), liên kết với uploaded_by.
- **Department "1" -- "0..*" FormTemplate : quản lý**
 - + Ý nghĩa: Một phòng ban quản lý 0 hoặc nhiều mẫu biểu mẫu (mối quan hệ một-nhiều).
 - + Ký hiệu: "1" -- "0..*".
 - + Nhãn: quản lý, liên kết với department_id.
- **FormTemplateController ..> FormTemplate : quản lý**
 - + Ý nghĩa: Bộ điều khiển quản lý mẫu biểu mẫu (mối quan hệ phụ thuộc).
 - + Ký hiệu: ..> (đường nét đứt).
- **FormTemplateController ..> User : sử dụng**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin người dùng (ví dụ: kiểm tra uploaded_by).
 - + Ký hiệu: Mối quan hệ phụ thuộc.
- **FormTemplateController ..> Department : sử dụng**
 - + Ý nghĩa: Bộ điều khiển sử dụng thông tin phòng ban (ví dụ: gán department_id).
 - + Ký hiệu: Mối quan hệ phụ thuộc.

8. BIỂU ĐỒ LỚP THIẾT KẾ UML

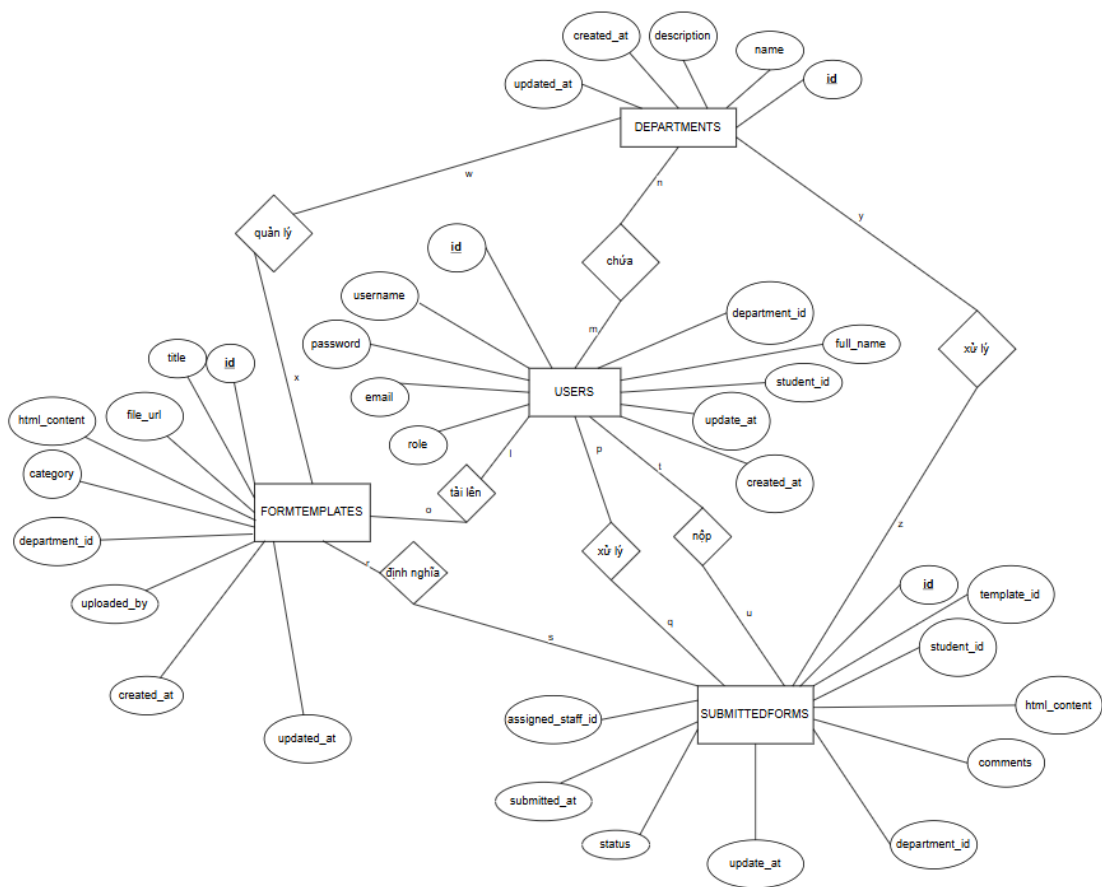


Hình 2. 26: Biểu đồ lớp thiết kế UML

- **departments** ||--o{ **users** : Một phòng ban chứa 0 hoặc nhiều người dùng (department_id).
- **departments** ||--o{ **formtemplates** : Một phòng ban quản lý 0 hoặc nhiều mẫu biểu mẫu (department_id).
- **departments** ||--o{ **submittedforms** : Một phòng ban xử lý 0 hoặc nhiều biểu mẫu đã nộp (department_id).
- **users** ||--o{ **formtemplates** : Một người dùng tải lên 0 hoặc nhiều mẫu biểu mẫu (uploaded_by).
- **users** ||--o{ **submittedforms** : Một người dùng (sinh viên) nộp 0 hoặc nhiều biểu mẫu (student_id).
- **users** ||--o{ **submittedforms** : Một người dùng (nhân viên) xử lý 0 hoặc nhiều biểu mẫu (assigned_staff_id).

- **users ||--o{ notifications** : Một người dùng nhận 0 hoặc nhiều thông báo (user_id).
- **users ||--o{ feedbacks** : Một người dùng (sinh viên) cung cấp 0 hoặc nhiều phản hồi (student_id).
- **formtemplates ||--o{ submittedforms** : Một mẫu biểu mẫu định nghĩa 0 hoặc nhiều biểu mẫu đã nộp (template_id).
- **submittedforms ||--o{ notifications** : Một biểu mẫu đã nộp kích hoạt 0 hoặc nhiều thông báo (submission_id).
- **submittedforms ||--o{ feedbacks** : Một biểu mẫu đã nộp nhận 0 hoặc nhiều phản hồi (submission_id).

9. CHUYỂN ĐỔI LỚP THIẾT KẾ UML SANG ER



Hình 2. 27: Mô hình thực thể ER

10. CƠ SỞ DỮ LIỆU

10.1. Bảng users

Quản lý thông tin tài khoản người dùng (sinh viên, cán bộ phòng ban).

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
username	VARCHAR(50)	Tên đăng nhập
password	VARCHAR(255)	Mật khẩu
email	VARCHAR(255)	Email liên hệ
role	ENUM(...)	Vai trò (student, staff, admin)
department_id	INT	Phòng ban công tác
full_name	VARCHAR(255)	Họ tên đầy đủ
student_id	VARCHAR(50)	Mã sinh viên (nếu là sinh viên)
created_at	TIMESTAMP	Thời gian tạo
updated_at	TIMESTAMP	Cập nhật gần nhất

10.2. Bảng departments

Lưu thông tin về các phòng ban thuộc trường.

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
name	VARCHAR(255)	Tên phòng ban
description	TEXT	Mô tả phòng ban
created_at	TIMESTAMP	Ngày tạo
updated_at	TIMESTAMP	Cập nhật gần nhất

10.3. Bảng formtemplates

Lưu trữ các biểu mẫu hành chính được các phòng ban cung cấp

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
title	VARCHAR(255)	Tiêu đề biểu mẫu
file_url	VARCHAR(255)	Đường dẫn file đính kèm (PDF)
html_content	TEXT	Nội dung HTML của biểu mẫu
category	VARCHAR(100)	Loại biểu mẫu (học vụ, tài chính...)
department_id	INT	Phòng ban tạo biểu mẫu
uploaded_by	INT	ID người tải lên (staff/admin)
created_at	TIMESTAMP	Ngày tạo
updated_at	TIMESTAMP	Ngày cập nhật

10.4. Bảng submittedforms

Lưu các biểu mẫu đã được sinh viên điền và nộp.

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
template_id	INT	Liên kết đến biểu mẫu gốc
student_id	INT	Người nộp biểu mẫu
html_content	TEXT	Nội dung đã điền của biểu mẫu
status	ENUM(...)	Trạng thái xử lý (tiếp nhận, pending, approved, rejected)
department_id	INT	Phòng ban tiếp nhận
assigned_staff_id	INT	Cán bộ được phân công xử lý
submitted_at	TIMESTAMP	Ngày nộp
updated_at	TIMESTAMP	Cập nhật gần nhất
comments	TEXT	Ghi chú thêm

CHƯƠNG 3. CÀI ĐẶT CHƯƠNG TRÌNH VÀ THỬ NGHIỆM

1. Môi trường triển khai

1.1. Phần cứng

❖ Máy chủ (Server):

- CPU: Intel Core i5 hoặc tương đương (tối thiểu 4 lõi).
- RAM: 8GB.
- Dung lượng lưu trữ: SSD 120GB trở lên (dành cho hệ điều hành, ứng dụng, và cơ sở dữ liệu).
- Hệ điều hành: Windows Server 2019.

❖ Máy trạm (Client):

- CPU: Intel Core i3 hoặc tương đương.
- RAM: 4GB.
- Trình duyệt: Google Chrome, Mozilla Firefox, hoặc Microsoft Edge (phiên bản mới nhất).
- Kết nối mạng: Tốc độ tối thiểu 10Mbps.

1.2. Phần mềm

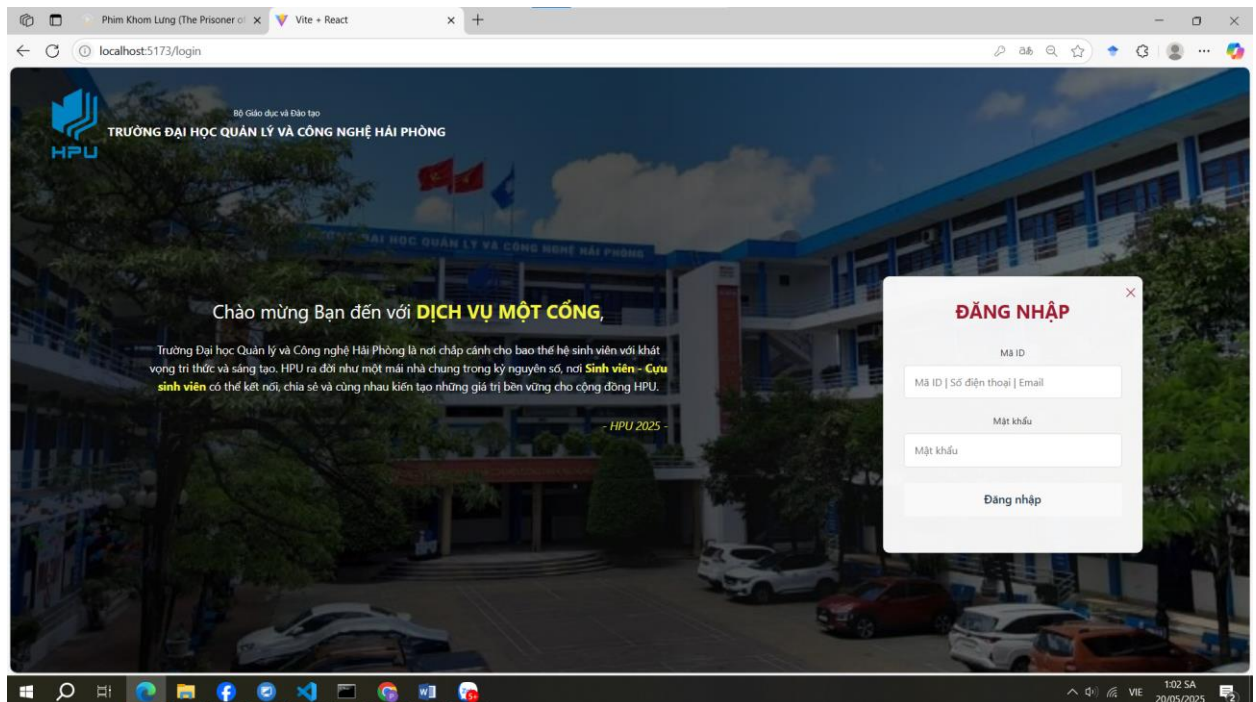
- Hệ điều hành máy chủ: Windows Server 2019.
- Môi trường runtime:
 - o Node.js: Phiên bản 16.x hoặc cao hơn (cho backend).
 - o npm: Phiên bản 8.x hoặc cao hơn (quản lý gói Node.js).
- Cơ sở dữ liệu: MySQL 8.0 hoặc PostgreSQL 13 (MySQL được sử dụng trong hệ thống này).
- Công cụ phát triển:
 - o Visual Studio Code: Soạn thảo mã nguồn (hỗ trợ JavaScript, ReactJS).
 - o Git: Quản lý phiên bản mã nguồn.
 - o Postman: Kiểm tra API.
- Công cụ triển khai:
 - o Nginx hoặc Apache: Máy chủ web để phục vụ ứng dụng frontend.
 - o PM2: Quản lý tiến trình Node.js cho backend.

1.3. Công cụ phát triển

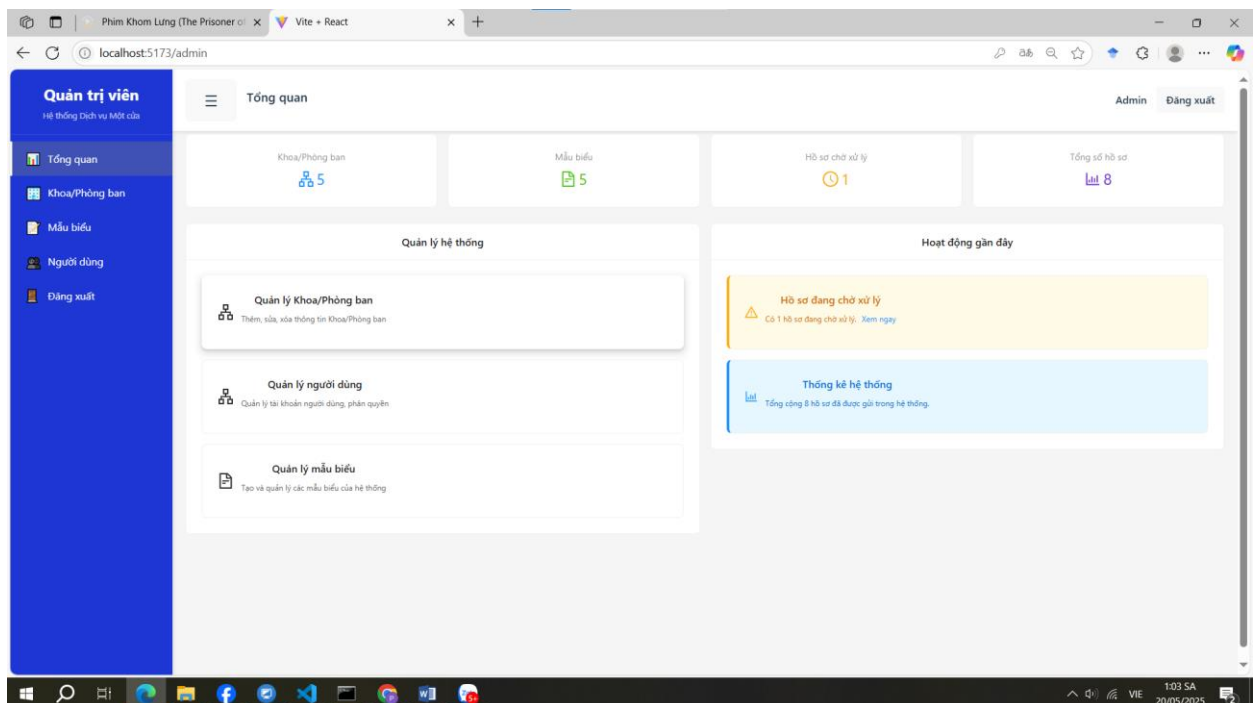
- Frontend: ReactJS (với Ant Design cho giao diện).
- Backend: Node.js với Express.js (framework xử lý API).
- Cơ sở dữ liệu: MySQL Workbench (quản lý và thiết kế database).
- Kiểm thử: Jest (unit test cho backend), React Testing Library (test frontend).

- Quản lý mã nguồn: GitHub hoặc GitLab (lưu trữ và quản lý mã nguồn).

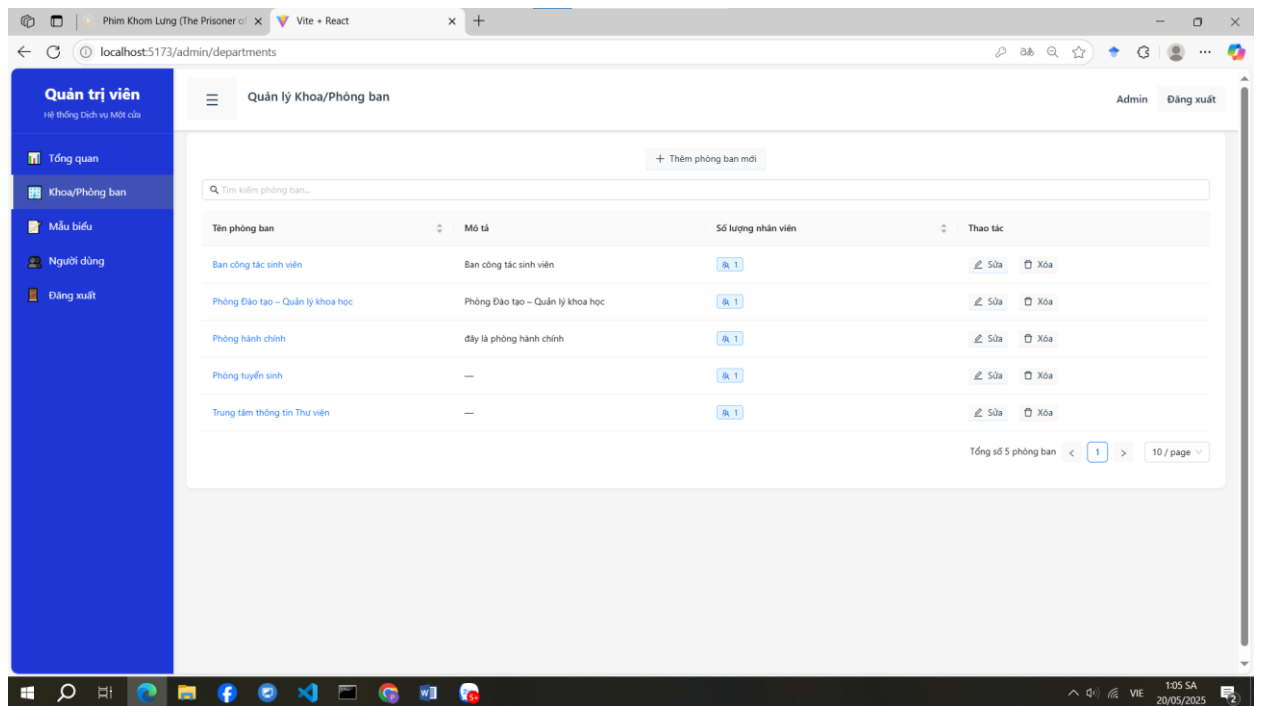
2. Giao diện chương trình



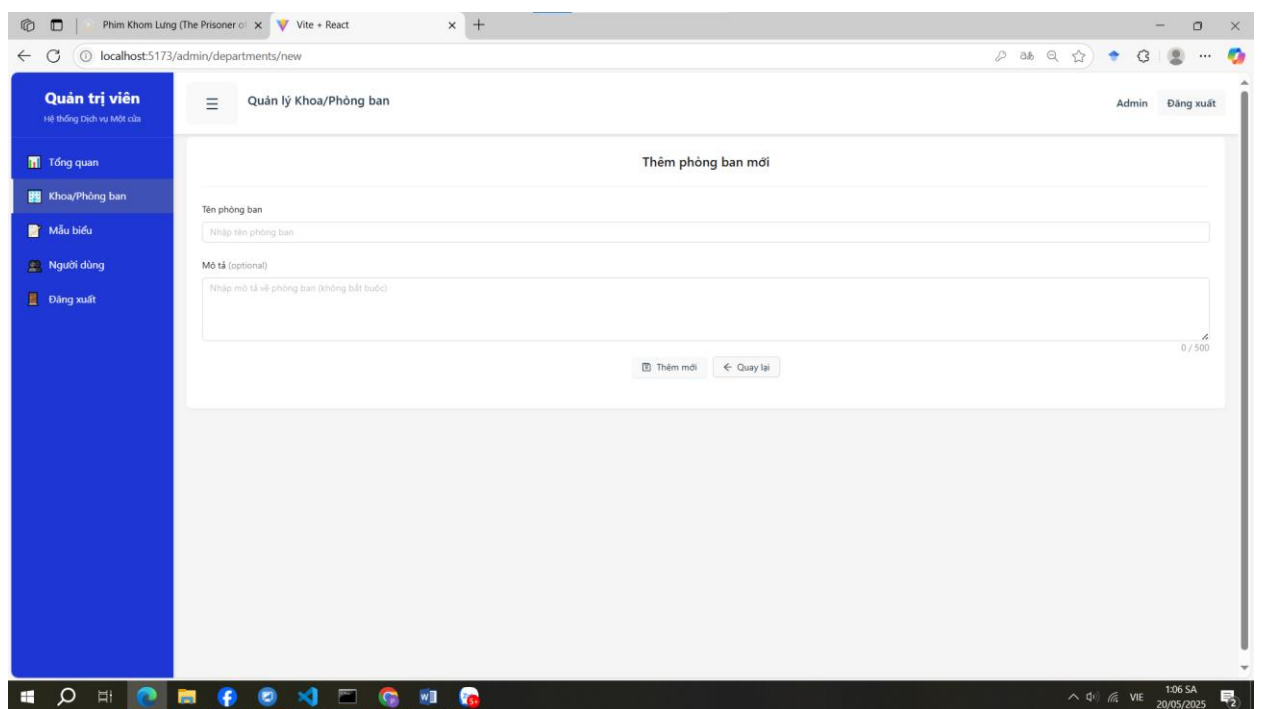
Hình 3. 1: Giao diện trang đăng nhập



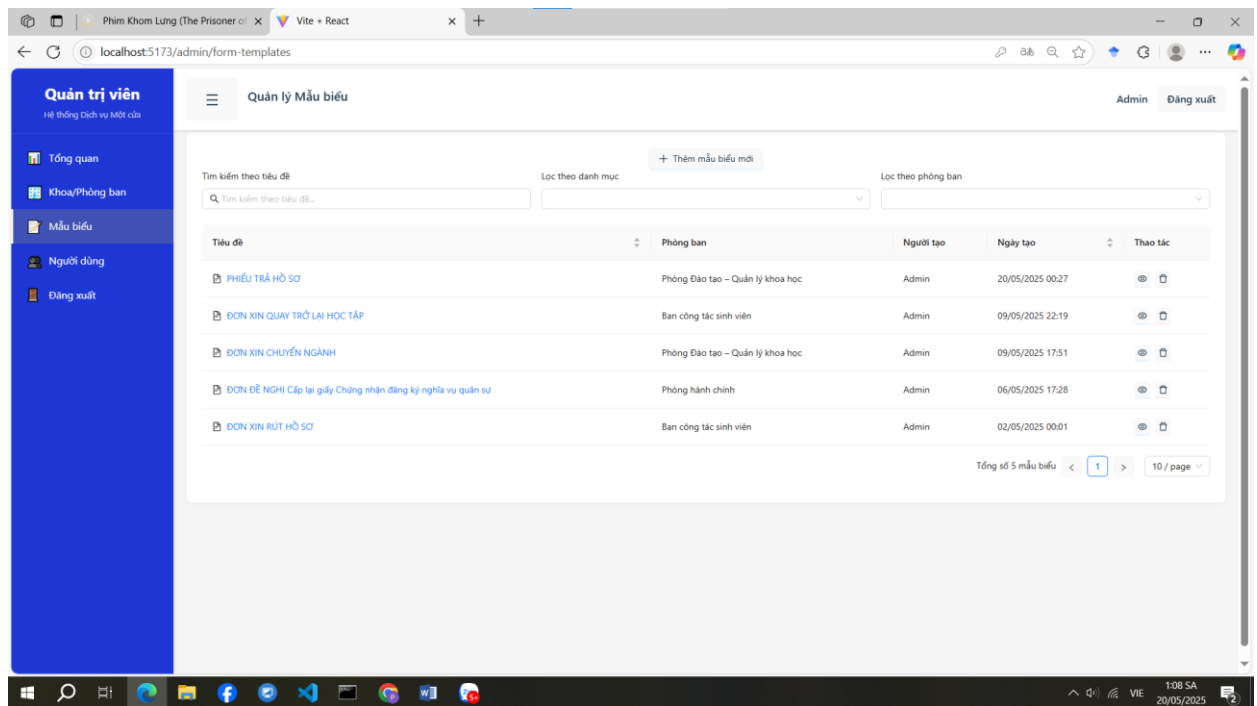
Hình 3. 2: Giao diện quản trị admin



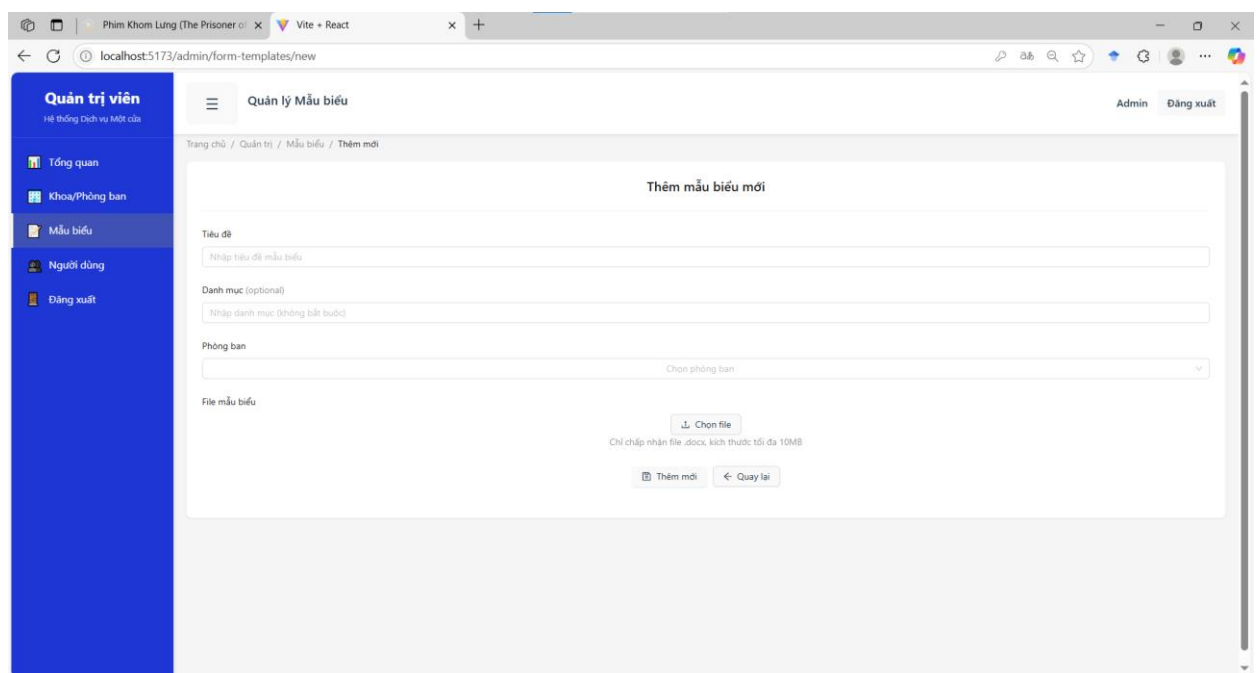
Hình 3. 3: Giao diện admin trang “Quản lý Khoa/Phòng ban”



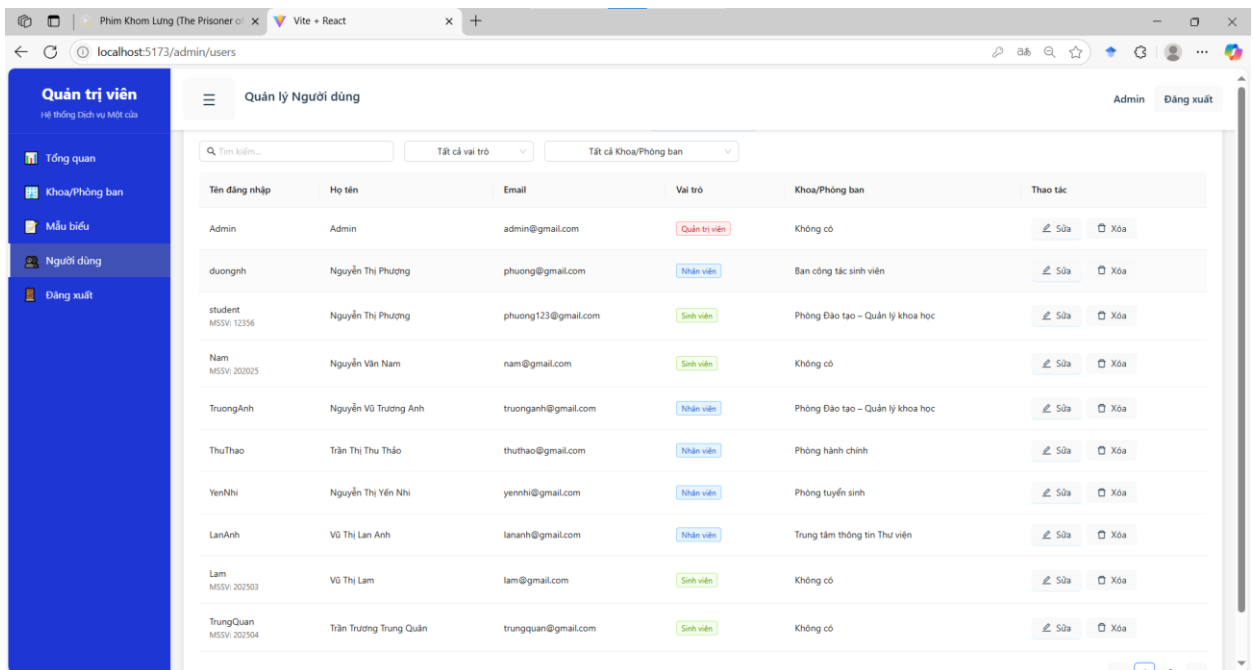
Hình 3. 4: Giao diện admin trang “Thêm mới phòng ban”



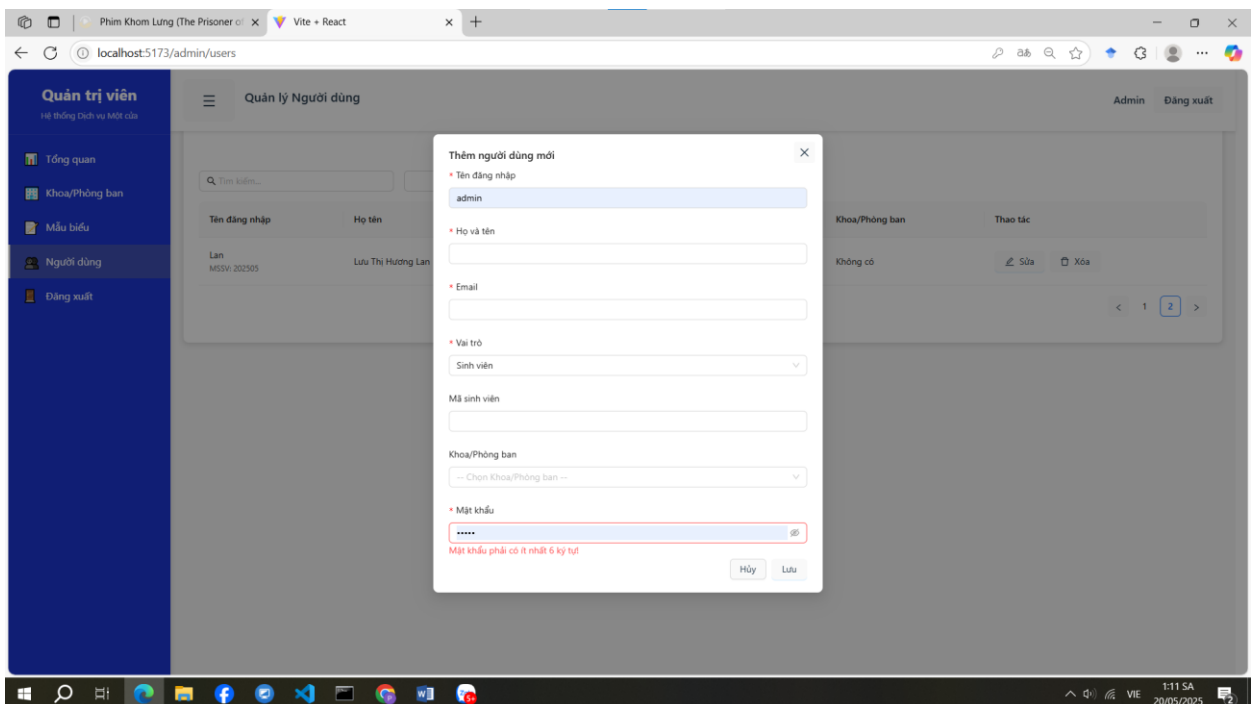
Hình 3. 5: Giao diện admin trang “Quản lý mẫu biểu”



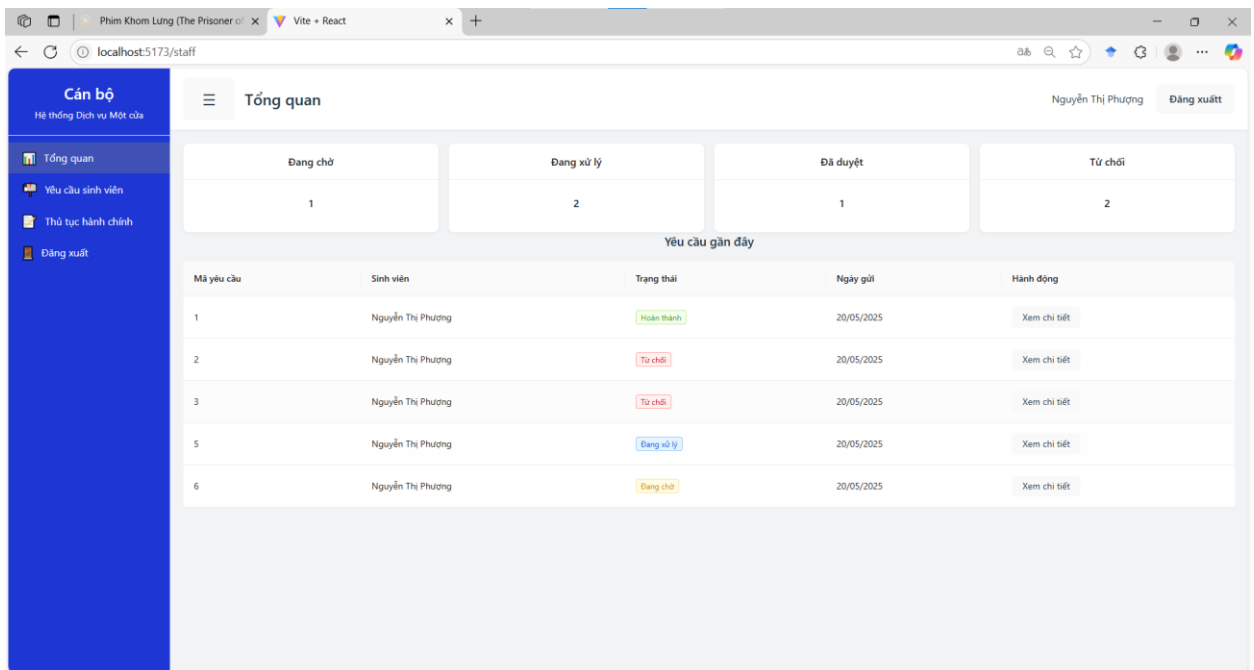
Hình 3. 6: Giao diện admin trang “Thêm mẫu biểu mới”



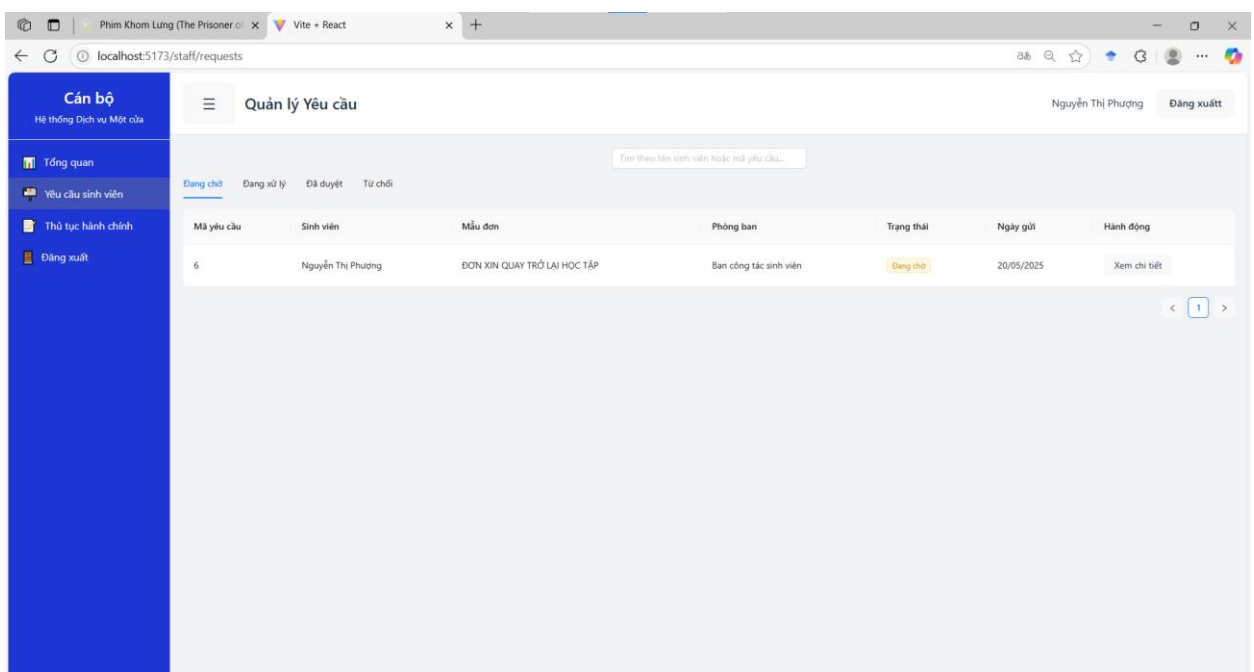
Hình 3. 7: Giao diện admin trang “Quản lý người dùng”



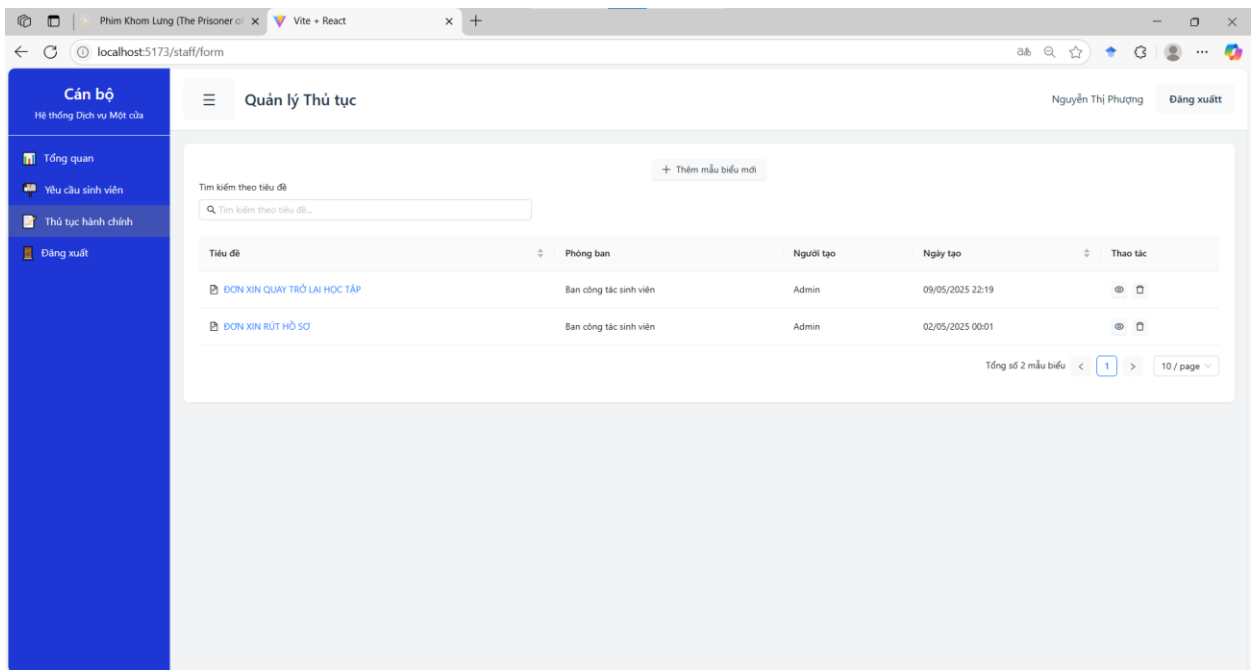
Hình 3. 8: Giao diện admin trang “Thêm người dùng mới”



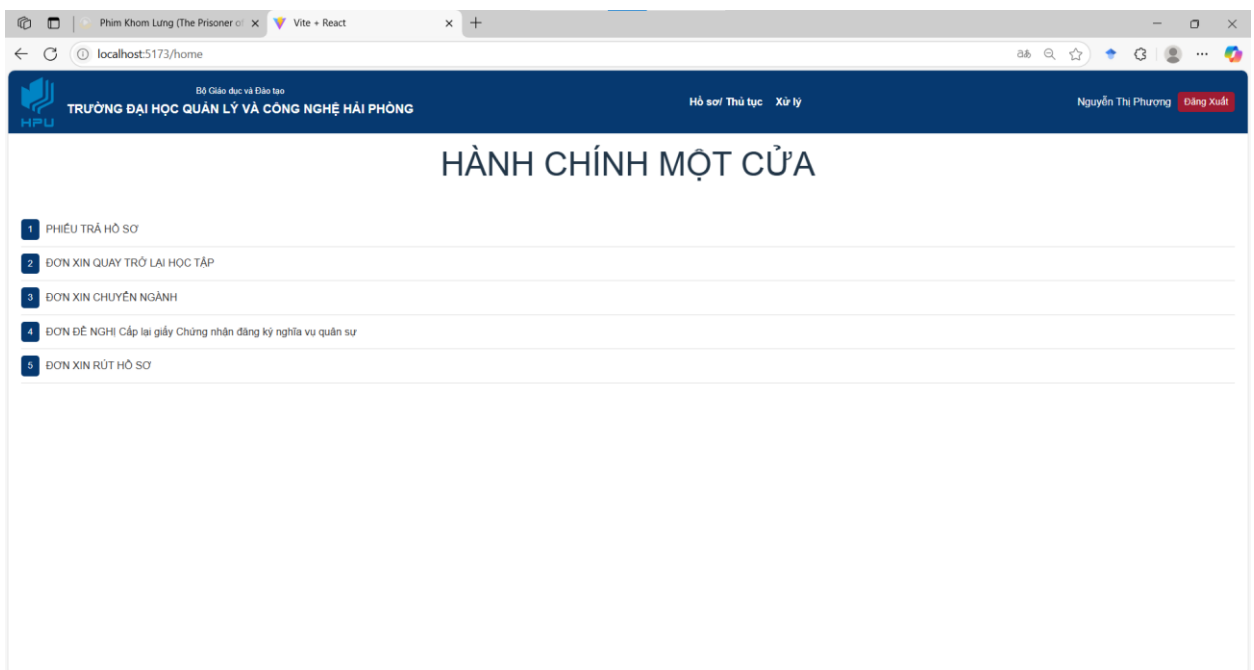
Hình 3. 9: Giao diện trang quản lý



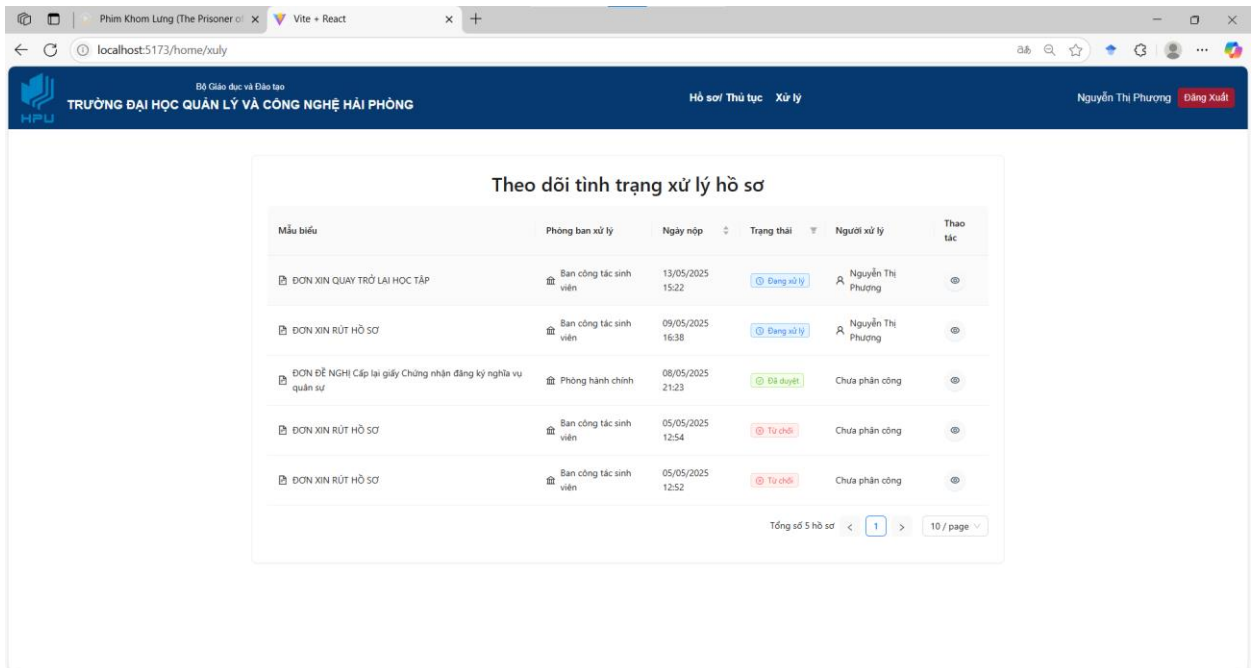
Hình 3. 10: Giao diện quản lý trang “Quản lý yêu cầu”



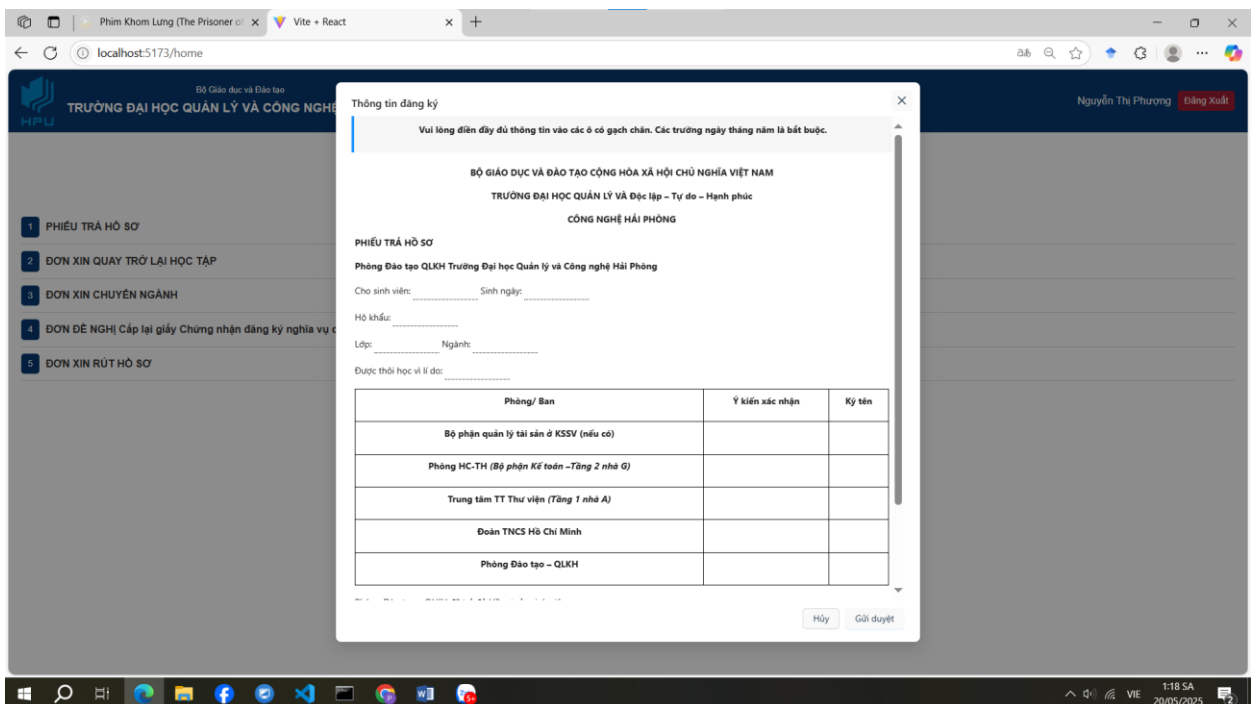
Hình 3. 11: Giao diện quản lý trang “Quản lý thủ tục”



Hình 3. 12: Giao diện trang chủ sinh viên



Hình 3. 13: Giao diện sinh viên trang “Theo dõi tình trạng xử lý hồ sơ”



Hình 3. 14: Giao diện sinh viên trang “Thông tin đăng ký”

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Đồ án tốt nghiệp với đề tài “Nghiên cứu và đề xuất xây dựng phần mềm quản lý hành chính online dành cho sinh viên” đã được thực hiện thành công, đáp ứng các mục tiêu đề ra ban đầu. Hệ thống website được phát triển cho phép tự động hóa các quy trình hành chính tại trường đại học, giúp giảm thiểu thời gian và công sức cho cả sinh viên và cán bộ hành chính. Các chức năng chính của hệ thống bao gồm:

- Đăng nhập và phân quyền: Hỗ trợ xác thực người dùng theo vai trò (sinh viên, cán bộ, quản trị viên), đảm bảo tính bảo mật và phân quyền truy cập phù hợp.
- Nộp và theo dõi hồ sơ: Sinh viên có thể dễ dàng nộp biểu mẫu trực tuyến và theo dõi trạng thái hồ sơ (chờ xử lý, đang xử lý, được phê duyệt, bị từ chối) thông qua giao diện thân thiện.
- Xử lý hồ sơ: Cán bộ hành chính có thể xem, phê duyệt, từ chối và gửi phản hồi cho các hồ sơ một cách hiệu quả.
- Quản lý hệ thống: Quản trị viên có khả năng quản lý tài khoản người dùng, phòng ban, và mẫu biểu, đảm bảo hệ thống hoạt động trơn tru và linh hoạt.

Trong quá trình thực hiện, đồ án đã áp dụng các công nghệ hiện đại như ReactJS, Node.js, MySQL để xây dựng một hệ thống đáp ứng tốt các yêu cầu về giao diện, hiệu năng, và khả năng mở rộng. Kết quả kiểm thử cho thấy hệ thống hoạt động ổn định, đáp ứng được nhu cầu thực tế của sinh viên và cán bộ tại Trường Đại học Quản lý và Công nghệ Hải Phòng. Đồ án không chỉ mang lại giá trị thực tiễn mà còn thể hiện khả năng vận dụng kiến thức công nghệ thông tin của sinh viên vào việc giải quyết các bài toán thực tế trong lĩnh vực giáo dục.

Mặc dù đã đạt được nhiều kết quả tích cực, hệ thống vẫn còn một số hạn chế, như việc chưa triển khai đầy đủ tính năng thông báo tự động qua email hoặc ứng dụng, cũng như chưa tích hợp các tính năng nâng cao như phân tích dữ liệu hay hỗ trợ đa ngôn ngữ. Những hạn chế này mở ra cơ hội để tiếp tục cải tiến và phát triển hệ thống trong tương lai.

2. Hướng phát triển

Để nâng cao hiệu quả và tính ứng dụng của hệ thống quản lý hành chính online, một số hướng phát triển trong tương lai có thể được xem xét:

- Tích hợp thông báo tự động: Hoàn thiện tính năng gửi thông báo qua email hoặc ứng dụng di động (push notification) để cập nhật trạng thái hồ sơ cho sinh viên và cán bộ một cách tức thời, giúp tăng tính tương tác và tiện lợi.
- Phát triển ứng dụng di động: Xây dựng ứng dụng di động (iOS và Android) dựa trên hệ thống web hiện tại, sử dụng các framework như React Native hoặc Flutter, nhằm hỗ trợ sinh viên truy cập dịch vụ mọi lúc, mọi nơi.
- Tăng cường bảo mật: Áp dụng các biện pháp bảo mật nâng cao như mã hóa dữ liệu end-to-end, xác thực hai yếu tố (2FA), và giám sát các hoạt động đăng nhập để bảo vệ thông tin người dùng.
- Hỗ trợ đa ngôn ngữ: Tích hợp tính năng đa ngôn ngữ (ví dụ: tiếng Anh, tiếng Việt) để hệ thống có thể phục vụ sinh viên quốc tế hoặc được triển khai tại các trường đại học có chương trình quốc tế.
- Phân tích và báo cáo: Thêm các công cụ phân tích dữ liệu (ví dụ: thống kê số lượng hồ sơ theo phòng ban, thời gian xử lý trung bình) và tạo báo cáo tự động để hỗ trợ quản trị viên trong việc đánh giá hiệu quả hoạt động của các phòng ban.
- Tích hợp trí tuệ nhân tạo (AI): Ứng dụng AI để tự động phân loại hồ sơ, gợi ý mẫu biểu phù hợp, hoặc cung cấp chatbot hỗ trợ sinh viên trong quá trình sử dụng hệ thống.
- Mở rộng chức năng: Tích hợp thêm các tính năng như lịch sử chỉnh sửa hồ sơ, hỗ trợ nộp hồ sơ theo nhóm, hoặc liên kết với các hệ thống khác của trường (ví dụ: hệ thống quản lý học tập, tài chính) để tạo ra một hệ sinh thái số toàn diện.

Những hướng phát triển này không chỉ giúp nâng cao trải nghiệm người dùng mà còn góp phần đưa hệ thống trở thành một giải pháp quản lý hành chính toàn diện, đáp ứng xu hướng chuyển đổi số trong giáo dục. Với sự hỗ trợ của công nghệ và sự đóng góp ý kiến từ người dùng, hệ thống có tiềm năng được triển khai rộng rãi tại nhiều trường đại học trong tương lai.

TÀI LIỆU THAM KHẢO

- [1]. Ramez A. Elmasri, Shamkant Navathe, Fundamentals of Database Systems, 7th edition, John Wiley & Sons, Inc., 2016.
- [2]. Đoàn Thanh Nghị, Giáo trình Lập trình Web, Đại học Quốc gia TP.HCM, 2022.
- [3]. Nguyễn Tuệ, Giáo trình nhập môn hệ cơ sở dữ liệu , Nhà xuất bản giáo dục (tái bản lần thứ nhất), 2009.
- [4]. Trương Ninh Thuận, Đặng Đức Hạnh, Giáo trình Phân tích thiết kế hướng đối tượng, Đại học Quốc gia Hà Nội, 2016.