

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 : Đoàn Duy Linh

Giảng viên hướng dẫn: Th.S Vũ Phạm Minh Hiền

HẢI PHÒNG - 2025

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

**PHÁT TRIỂN TRANG QUẢN LÝ KHO DỮ LIỆU BÀI
TẬP LỚN SINH VIÊN HPU NGÀNH CNTT**

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

Sinh viên : Đoàn Duy Linh

Giảng viên hướng dẫn: Th.S Vũ Phạm Minh Hiến

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: Đoàn Duy Linh

Mã SV: 2112111014

Lớp : CT2501C

Ngành : Công nghệ thông tin

Tên đề tài: *“Phát triển trang quản lý kho dữ liệu bài tập lớn sinh viên HPU ngành CNTT”*

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Vũ Phạm Minh Hiền

Cơ quan công tác : Khoa Công nghệ thông tin

Nội dung hướng dẫn:

- Ngôn ngữ lập trình
- Platform
- Các use case

Đề tài tốt nghiệp được giao ngày 23 tháng 08 năm 2025

Yêu cầu phải hoàn thành xong trước ngày 17 tháng 11 năm 2025

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Ths Vũ Phạm Minh Hiền

Hải Phòng, ngày tháng năm 2025

TRƯỞNG KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Vũ Phạm Minh Hiền

Đơn vị công tác: Khoa Công nghệ thông tin - Trường Đại học Quản Lý và Công Nghệ Hải Phòng.

Họ và tên sinh viên : Đoàn Duy Linh

Ngành: Công nghệ Thông tin

Nội dung hướng dẫn : Toàn bộ đề tài

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

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

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

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

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ

Không được bảo vệ

Điểm hướng dẫn

Hải Phòng, ngày tháng năm 2025

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong quá trình làm đồ án vừa qua vì được sự chỉ dẫn nhiệt tình của thầy ThS. Vũ Phạm Minh Hiền – Trường Đại học Quản lý và Công nghệ Hải Phòng, em đã hoàn thành đồ án của mình. Mặc dù em đã cố gắng với sự tận tâm của thầy, nhưng vì thời gian và khả năng nên đồ án của em vẫn còn không tránh được những điều thiếu sót.

Em xin chân thành và bày tỏ lòng biết ơn sâu sắc đến thầy Vũ Phạm Minh Hiền vì đã tận tình chỉ bảo, hướng dẫn và giành thời gian quý báu của mình cho em trong thời gian qua để em có thể hoàn thành đồ án của mình đúng thời hạn.

Em xin cảm ơn tất cả thầy cô giáo trong khoa Công nghệ thông tin vì đã truyền đạt cho em rất nhiều các kiến thức nền tảng, chuyên ngành, chuyên môn và chuyên sâu cực kì vững chắc trong những năm qua để em có thể hoàn thành được đồ án này.

Em xin cảm ơn Trường Đại học Quản lý và Công nghệ Hải Phòng vì không ngừng hỗ trợ và đào tạo những điều kiện tốt nhất trong những năm vừa qua để em có thể học và thực hiện tốt đồ án.

Em xin chân thành cảm ơn !

LỜI CAM ĐOAN

Em xin cam đoan bài đồ án tốt nghiệp với đề tài “PHÁT TRIỂN TRANG QUẢN LÝ KHO DỮ LIỆU BÀI TẬP LỚN SINH VIÊN HPU NGÀNH CNTT” là công trình nghiên cứu độc lập dưới sự hướng dẫn của TS. Vũ Phạm Minh Hiền-Giảng viên Khoa Công nghệ thông tin Trường Đại Học Quản Lý và Công Nghệ Hải Phòng. Những số liệu và kết quả trong đề tài là trung thực, các kết quả là do em thực hiện

Các tài liệu tham khảo đã được trích dẫn đầy đủ.

Em xin chịu hoàn toàn trách nhiệm về mọi nội dung của đề tài làm đồ án tốt nghiệp này

MỤC LỤC

LỜI CẢM ƠN	7
LỜI CAM ĐOAN.....	8
DANH MỤC HÌNH ẢNH.....	Error! Bookmark not defined.
CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI.....	14
1.1. Lý do chọn đề tài	14
1.2. Mục tiêu của hệ thống	14
1.3. Đối tượng sử dụng hệ thống	15
1.4. Phạm vi và giới hạn đề tài	16
1.4.1. Phạm vi đề tài	16
1.4.2. Giới hạn của đề tài	17
1.5. Phạm vi và giới hạn đề tài	17
CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ PHÂN TÍCH YÊU CẦU.....	19
2.1. Tổng quan về quản lý dữ liệu bài tập lớn.....	19
2.1.1. Khái niệm "Hệ thống quản lý tài liệu" trong Dự án.....	19
2.1.2. Tầm quan trọng của việc quản lý, lưu trữ và tra cứu tài sản trí tuệ trong môi trường học thuật.....	19
2.1.3. So sánh với các giải pháp hiện có và nêu hạn chế.....	20
2.2. Công nghệ sử dụng.....	22
2.2.1. Công nghệ phía Client (Frontend).....	22
2.2.2. Công nghệ phía Server (Backend).....	24
2.3. Yêu cầu chức năng	27
2.3.1. Chức năng cho Sinh viên.....	27
2.3.2. Chức năng cho Giảng viên	27
2.3.3. Chức năng cho Quản trị viên (Admin).....	27
2.3.4. Chức năng chung (Dành cho cả 3 vai trò).....	28
2.4. Yêu cầu phi chức năng	28
CHƯƠNG III: THIẾT KẾ HỆ THỐNG	30
3.1. Sơ đồ Use Case.....	30
3.1.1. Use Case tổng quát	30

3.1.2.	Biểu đồ usecase phân rã.....	32
3.2.	Đặc tả chức năng	42
3.2.1.	Use case Đăng nhập.....	42
3.2.2.	Use case Đăng ký.....	44
3.2.3.	Use case Đăng xuất.....	46
3.2.4.	Use case Thay đổi mật khẩu cá nhân.....	46
3.2.5.	Use case Tìm kiếm tài liệu đã được duyệt.....	48
3.2.6.	Use case Tải tài liệu về máy	50
3.2.7.	Use case Nộp bài tập hoặc tài liệu mới cho môn học.....	52
3.2.8.	Use case Xem lịch sử các bài đã nộp.....	54
3.2.9.	Use case Xem trang thống kê cá nhân.....	55
3.2.10.	Use case Xem danh sách bài nộp đang chờ duyệt.....	57
3.2.11.	Use case Xem tất cả bài nộp trong các môn phụ trách.....	58
3.2.12.	Use case Duyệt bài nộp của sinh viên.	60
3.2.13.	Use case Từ chối bài nộp của sinh viên.	61
3.2.14.	Use case Xem trang thống kê giảng viên	63
3.2.15.	Use case Quản lý tài khoản người dùng	65
3.2.16.	Use case Quản lý danh mục môn học.....	68
3.2.17.	Use case Gán giảng viên vào môn học	70
3.2.18.	Use case Quản lý tài khoản giảng viên đang chờ phê duyệt	72
3.2.19.	Use case Xem thống kê tổng quan toàn hệ thống.....	74
3.3.	Biểu đồ tuần tự	76
3.3.1.	Biểu đồ tuần tự usecase đăng nhập.....	76
3.3.2.	Biểu đồ tuần tự usecase đăng ký của sinh viên	77
3.3.3.	Biểu đồ tuần tự usecase đăng xuất.....	77
3.3.4.	Biểu đồ tuần tự usecase thay đổi mật khẩu	78
3.3.5.	Biểu đồ tuần tự usecase tìm kiếm tài liệu.....	78
3.3.6.	Biểu đồ tuần tự usecase tìm kiếm tài liệu.....	78
3.3.7.	Biểu đồ tuần tự usecase nộp bài tập	79
3.3.8.	Biểu đồ tuần tự usecase xem lịch sử các bài đã nộp	80

3.3.9.	Biểu đồ tuần tự usecase xem trang thống kê cá nhân.....	80
3.3.10.	Biểu đồ tuần tự usecase Giảng viên xem danh sách bài nộp.....	80
3.3.11.	Biểu đồ tuần tự usecase Giảng viên duyệt bài nộp.....	81
3.3.12.	Biểu đồ tuần tự usecase Quản lý tài khoản người dùng	82
3.3.13.	Biểu đồ tuần tự usecase Quản lý môn học	83
3.3.14.	Biểu đồ tuần tự usecase Gán giảng viên vào môn học.....	85
3.1.15.	Biểu đồ tuần tự usecase các tài khoản giảng viên đang chờ phê duyệt	85
3.4.	Thiết kế cơ sở dữ liệu	86
3.4.1.	Sơ đồ cơ sở dữ liệu	86
CHƯƠNG IV: CÀI ĐẶT VÀ XÂY DỰNG HỆ THỐNG		87
4.1.	Môi trường phát triển.....	87
4.2.	Cấu trúc dự án.....	89
4.2.1.	Cấu trúc dự án Backend.....	89
4.2.2.	Cấu trúc dự án Frontend	89
4.3.	Mô tả các chức năng chính.....	90
4.3.1.	Chức năng chung	90
4.3.2.	Chức năng của Sinh viên	92
4.3.3.	Chức năng của Giảng viên.....	93
4.3.4.	Chức năng của Admin	94
CHƯƠNG V: KẾT LUẬN.....		96
5.1.	Kết quả đạt được.....	96
5.2.	Hạn chế	97
5.3.	Hướng phát triển.....	98
Tài liệu tham khảo		99

DANH MỤC HÌNH ẢNH

Hình 2.1. Công nghệ React.....	23
Hình 2.2. Công nghệ Nodejs Express	24
Hình 2.3. Công nghệ MySQL	26
Hình 3.1. Sơ đồ Usecase tổng quát.	31
Hình 3.2. Biểu đồ phân rã UseCase đăng nhập.....	32
Hình 3.3. Biểu đồ phân rã UseCase đăng xuất.....	32
Hình 3.3. Biểu đồ phân rã UseCase đổi mật khẩu.	33
Hình 3.4. Biểu đồ phân rã UseCase tìm kiếm tài liệu.....	34
Hình 3.5. Biểu đồ phân rã UseCase tải tài liệu về máy.....	34
Hình 3.6. Biểu đồ phân rã UseCase đăng ký.	35
Hình 3.7. Biểu đồ phân rã UseCase nộp tài liệu.	36
Hình 3.8. Biểu đồ phân rã UseCase xem lịch sử bài nộp.....	37
Hình 3.9. Biểu đồ phân rã UseCase xem trang thống kê cá nhân.....	37
Hình 3.10. Biểu đồ phân rã UseCase xem danh sách bài nộp chờ duyệt.....	38
Hình 3.11. Biểu đồ phân rã UseCase xem danh sách tất cả bài nộp.....	38
Hình 3.12. Biểu đồ phân rã UseCase xem trang thống kê giảng viên.	39
Hình 3.13. Biểu đồ phân rã UseCase quản lý người dùng.....	39
Hình 3.14. Biểu đồ phân rã UseCase quản lý danh mục môn học.....	40
Hình 3.15. Biểu đồ phân rã UseCase gán giảng viên vào môn học.....	40
Hình 3.16. Biểu đồ phân rã UseCase Quản lý giảng viên đang chờ phê duyệt.....	41
Hình 3.17. Biểu đồ phân rã UseCase xem thông báo toàn hệ thống.....	42
Ảnh 3.18. Biểu đồ tuần tự chức năng Đăng nhập	76
Ảnh 3.19. Biểu đồ tuần tự chức năng Đăng ký của sinh viên.....	77
Ảnh 3.20. Biểu đồ tuần tự chức năng Đăng xuất	77
Ảnh 3.21. Biểu đồ tuần tự chức năng Đổi mật khẩu.....	78
Ảnh 3.22. Biểu đồ tuần tự chức năng tìm kiếm tài liệu	78
Ảnh 3.23. Biểu đồ tuần tự chức năng tải tài liệu.....	79
Ảnh 3.24. Biểu đồ tuần tự chức năng nộp bài tập.....	79

Ảnh 3.25. Biểu đồ tuần tự chức năng xem lịch sử các bài đã nộp.....	80
Ảnh 3.26. Biểu đồ tuần tự chức năng xem trang thống kê sinh viên.....	80
Ảnh 3.27. Biểu đồ tuần tự chức năng quản lý bài nộp.....	81
Ảnh 3.28. Biểu đồ tuần tự chức năng duyệt bài nộp.....	81
Ảnh 3.30. Biểu đồ tuần tự chức năng quản lý môn học.....	84
Ảnh 3.31. Biểu đồ tuần tự chức năng gán giảng viên vào môn học	85
Ảnh 3.32. Biểu đồ tuần tự chức năng phê duyệt giảng viên	85
Ảnh 3.33. Sơ đồ cơ sở dữ liệu.....	86
Ảnh 4.1. Cấu trúc thư mục backend.....	89
Ảnh 4.2. Cấu trúc thư mục frontend.....	90
Ảnh 4.3. Giao diện Trang đăng nhập	90
Ảnh 4.4. Giao diện trang đăng ký	91
Ảnh 4.5. Giao diện trang tìm kiếm và tải tài liệu.....	91
Ảnh 4.6. Giao diện trang nộp bài	92
Ảnh 4.7. Giao diện trang lịch sử nộp bài	92
Ảnh 4.7. Giao diện trang duyệt bài của giảng viên.....	93
Ảnh 4.8. Giao diện trang quản lý tất cả bài nộp của Giảng viên	93
Ảnh 4.9. Giao diện trang báo cáo của Giảng viên	94
Ảnh 4.9. Giao diện trang quản lý user của admin.....	94
Ảnh 4.10. Giao diện trang quản lý môn học của admin.....	95
Ảnh 4.11. Giao diện trang thống kê của admin	95

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0, chuyển đổi số đang diễn ra mạnh mẽ trong mọi lĩnh vực, đặc biệt là giáo dục. Việc quản lý, lưu trữ và chia sẻ tài liệu học tập, bài giảng, và các thông tin liên quan đến môn học tại các cơ sở giáo dục vẫn còn tồn tại nhiều bất cập. Phần lớn quy trình còn thực hiện thủ công qua email, mạng xã hội hoặc các phương tiện lưu trữ cá nhân, dẫn đến tình trạng tài liệu phân tán, khó tìm kiếm, và thiếu tính hệ thống.

Mặt khác, việc giao và nộp bài tập, theo dõi tiến độ học tập của sinh viên và quản lý thông tin người dùng (giảng viên, sinh viên) cũng gặp nhiều khó khăn. Các quy trình thủ công không chỉ gây lãng phí thời gian, công sức mà còn dễ phát sinh sai sót, ảnh hưởng đến chất lượng dạy và học.

Do đó, đề tài “Phát triển hệ thống quản lý dữ liệu học tập” được lựa chọn với mục tiêu xây dựng một nền tảng tập trung, giúp giải quyết các vấn đề trên. Hệ thống sẽ hỗ trợ giảng viên, sinh viên và quản trị viên trong việc quản lý môn học, tài liệu và bài tập một cách hiệu quả, góp phần nâng cao trải nghiệm và chất lượng giáo dục.

1.2. Mục tiêu của hệ thống

Mục tiêu chính của đề tài là phát triển một hệ thống quản lý dữ liệu học tập trên nền tảng web, giúp tối ưu hóa các quy trình liên quan đến giảng dạy và học tập.

Các mục tiêu cụ thể bao gồm:

- Xây dựng website quản lý dữ liệu học tập với giao diện hiện đại, thân thiện và dễ sử dụng.
- Phát triển hệ thống đa người dùng với các vai trò được phân quyền rõ ràng:

- + Sinh viên (Student): Đăng ký, đăng nhập, tìm kiếm môn học, xem và tải tài liệu, nộp bài tập.
 - + Giảng viên (Lecturer): Quản lý các môn học mình phụ trách, đăng tải, cập nhật tài liệu, quản lý bài nộp của sinh viên.
 - + Quản trị viên (Admin): Quản lý tài khoản người dùng (giảng viên, sinh viên), quản lý danh mục môn học và giám sát hoạt động của toàn bộ hệ thống.
- Thiết kế cơ sở dữ liệu có khả năng mở rộng, đảm bảo tính nhất quán và toàn vẹn dữ liệu.
 - Xây dựng backend bằng Node.js và Express.js theo kiến trúc RESTful API để cung cấp các dịch vụ cho phía frontend.
 - Phát triển frontend bằng ReactJS nhằm tạo ra một giao diện người dùng linh hoạt, hiệu năng cao và dễ dàng cho việc bảo trì, nâng cấp.

1.3. Đối tượng sử dụng hệ thống

Đối tượng	Mô tả chức năng và quyền hạn
Sinh viên (Student)	Đăng ký, đăng nhập. Tìm kiếm và xem danh sách các môn học. Tải tài liệu từ các môn học. Nộp bài tập và theo dõi trạng thái bài nộp của mình.
Giảng viên (Lecturer)	Quản lý các môn học mình được phân công. Đăng tải, cập nhật, và xóa tài liệu, bài giảng. Xem và quản lý các bài nộp của sinh viên trong môn học của mình.
Quản trị viên (Admin)	Quản lý toàn bộ hệ thống: tạo, sửa, xóa tài khoản (sinh viên, giảng viên). Quản lý danh

	mục môn học. Giám sát các hoạt động chung và phân quyền cho người dùng.
--	---

Bảng 1.1. Bảng đối tượng sử dụng.

1.4. Phạm vi và giới hạn đề tài

1.4.1. Phạm vi đề tài

Đề tài tập trung vào việc xây dựng một Hệ thống Quản lý Dữ liệu Học tập hoàn chỉnh trên nền tảng web, bao gồm các thành phần chính:

- Frontend: Sử dụng ReactJS kết hợp với thư viện Material-UI (MUI) để xây dựng giao diện người dùng hiện đại, trực quan và có tính tương tác cao.
- Backend: Sử dụng Node.js và Express.js để xây dựng hệ thống RESTful API, xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu.
- Cơ sở dữ liệu: Sử dụng hệ quản trị cơ sở dữ liệu MySQL để lưu trữ toàn bộ dữ liệu của hệ thống như thông tin người dùng, môn học, tài liệu và các bài nộp.

Các chức năng chính đã được triển khai trong hệ thống:

- Quản lý người dùng: Đăng ký, đăng nhập, phân quyền chi tiết cho 3 vai trò (Admin, Giảng viên, Sinh viên). Giảng viên đăng ký phải được Admin duyệt.
- Chức năng cho Sinh viên: Tìm kiếm môn học, tải tài liệu, nộp bài tập/tài liệu theo từng môn, xem lại lịch sử và trạng thái (chờ duyệt, đã duyệt, bị từ chối) của các bài đã nộp.
- Chức năng cho Giảng viên: Quản lý các môn học được phân công, xem danh sách bài nộp từ sinh viên, thực hiện duyệt hoặc từ chối bài nộp kèm theo phản hồi.
- Chức năng cho Admin: Quản lý toàn bộ tài khoản người dùng và danh mục môn học trong hệ thống (thêm, sửa, xóa).

- Thống kê và Báo cáo: Cung cấp các trang Dashboard trực quan với biểu đồ và số liệu thống kê cho từng vai trò. Admin và Giảng viên có chức năng xuất báo cáo ra file CSV hoặc PDF.
- Hệ thống thông báo: Tự động gửi thông báo cho người dùng khi có các sự kiện quan trọng (ví dụ: bài nộp được duyệt).

1.4.2. Giới hạn của đề tài

Mặc dù đã hoàn thành các mục tiêu chính, đề tài vẫn còn một số giới hạn cần được cải thiện trong tương lai:

- Tương tác thời gian thực: Hệ thống thông báo hiện tại chưa hoạt động theo thời gian thực (real-time), người dùng cần tải lại trang để cập nhật thông báo mới.
- Quản lý phiên bản: Chức năng quản lý tài liệu chưa hỗ trợ version control (quản lý phiên bản), việc cập nhật tài liệu được thực hiện bằng cách xóa và tải lên lại.
- Tối ưu giao diện: Giao diện được thiết kế và tối ưu chủ yếu cho trải nghiệm trên máy tính (Desktop), chưa hoàn toàn tương thích và tối ưu cho các thiết bị di động (Mobile).
- Kiểm thử tự động: Việc kiểm thử chủ yếu được thực hiện thủ công (manual testing). Hệ thống chưa xây dựng một bộ kiểm thử tự động (automated tests) hoàn chỉnh cho Unit Test và Integration Test.

1.5. Phạm vi và giới hạn đề tài

Trong quá trình thực hiện đề tài, các phương pháp luận sau đã được áp dụng:

- Phương pháp khảo sát thực tiễn: Nghiên cứu, phân tích các hệ thống quản lý học tập (LMS - Learning Management System) phổ biến hiện nay như Google Classroom, Moodle, Blackboard và các cổng thông tin sinh viên của các

- trường đại học. Qua đó, học hỏi về các luồng chức năng, trải nghiệm người dùng và rút ra các yêu cầu cần thiết cho hệ thống.
- Phương pháp phân tích và thiết kế hệ thống: Sử dụng ngôn ngữ mô hình hóa thống nhất UML (Unified Modeling Language) để phân tích và thiết kế hệ thống, bao gồm:
 - + Biểu đồ Use Case: Xác định các tác nhân và các trường hợp sử dụng của hệ thống.
 - + Biểu đồ Lớp (Class Diagram): Thiết kế cấu trúc dữ liệu và mối quan hệ giữa các thực thể.
 - + Biểu đồ Trình tự (Sequence Diagram): Mô tả luồng tương tác giữa các đối tượng trong các chức năng chính.
 - Phương pháp phát triển phần mềm: Áp dụng mô hình phát triển lặp và tăng trưởng (Iterative and Incremental Development), kết hợp với các nguyên tắc của Agile. Hệ thống được chia thành các giai đoạn nhỏ, mỗi giai đoạn tập trung phát triển một nhóm chức năng hoàn chỉnh, cho phép kiểm thử, nhận phản hồi và cải tiến liên tục.
 - Phương pháp kiểm thử: Áp dụng các cấp độ kiểm thử khác nhau để đảm bảo chất lượng phần mềm:
 - + Kiểm thử đơn vị (Unit Test): Kiểm tra các hàm và thành phần nhỏ, độc lập.
 - + Kiểm thử tích hợp (Integration Test): Đảm bảo sự tương tác chính xác giữa các module (ví dụ: Frontend gọi API Backend).
 - + Kiểm thử chức năng (Functional Test): Kiểm tra các chức năng hoàn chỉnh của hệ thống từ góc độ người dùng cuối.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ PHÂN TÍCH YÊU CẦU

2.1. Tổng quan về quản lý dữ liệu bài tập lớn

2.1.1. Khái niệm "Hệ thống quản lý tài liệu" trong Dự án

Định nghĩa:

Hệ thống quản lý tài liệu (DMS) ở đây là một nền tảng số hóa để quản lý vòng đời của các tài liệu học thuật. Nó bao gồm việc tải lên, lưu trữ, duyệt xét, tìm kiếm và thống kê tài liệu, với vai trò phân quyền rõ ràng (sinh viên, giảng viên, quản trị viên).

Chức năng cốt lõi:

- Tải lên và lưu trữ tài liệu: Sinh viên có thể tải lên tài liệu (PDF, v.v.) kèm thông tin như tiêu đề, mô tả, thẻ tag, và liên kết với môn học cụ thể.
- Duyệt xét tài liệu: Giảng viên duyệt hoặc từ chối bài nộp, cung cấp phản hồi (feedback). Trạng thái tài liệu bao gồm "pending" (chờ duyệt), "approved" (duyet), "rejected" (từ chối).
- Tìm kiếm và lọc: Hệ thống cho phép tìm kiếm tài liệu theo từ khóa, môn học, trạng thái, thẻ tag, và khoảng thời gian.
- Thông báo: Tự động gửi thông báo cho sinh viên khi bài nộp được xử lý.
- Thống kê và báo cáo: Cung cấp số liệu về số lượng bài nộp, tỷ lệ duyệt/từ chối theo môn học, giảng viên, hoặc thời gian (tháng).
- Quản lý người dùng: Phân quyền dựa trên vai trò (student, lecturer, admin), với xác thực JWT.

2.1.2. Tầm quan trọng của việc quản lý, lưu trữ và tra cứu tài sản trí tuệ trong môi trường học thuật

Trong môi trường học thuật, việc quản lý, lưu trữ và tra cứu tài sản trí tuệ như bài tập lớn, đề án, luận văn hay nghiên cứu khoa học có vai trò hết sức quan trọng. Thứ nhất, nó giúp bảo tồn và phát huy giá trị tri thức của sinh viên, giảng viên, cũng như

của toàn bộ cơ sở đào tạo, tránh tình trạng thất lạc hoặc trùng lặp đề tài. Thứ hai, việc lưu trữ khoa học giúp cho quá trình tra cứu, tham khảo tài liệu trở nên nhanh chóng, thuận tiện, hỗ trợ tốt cho công tác học tập, giảng dạy và nghiên cứu. Thứ ba, quản lý tốt tài sản trí tuệ góp phần đảm bảo tính minh bạch, công bằng trong đánh giá học thuật, đồng thời giúp xác định quyền sở hữu và tránh vi phạm bản quyền. Cuối cùng, hệ thống lưu trữ hiện đại còn tạo điều kiện cho việc chia sẻ tri thức, khuyến khích sáng tạo và đổi mới trong cộng đồng học thuật.

2.1.3. So sánh với các giải pháp hiện có và nêu hạn chế

Dự án được xây dựng như một hệ thống quản lý tài liệu học thuật tùy chỉnh, tập trung vào việc nộp bài, duyệt bài và lưu trữ tài liệu trong môi trường giáo dục. Dưới đây là phần so sánh với một số giải pháp phổ biến hiện nay như Google Classroom, Moodle, và Microsoft Teams, cùng với những hạn chế còn tồn tại của hệ thống.

- So sánh với Google Classroom: Điểm tương đồng: Cả hai hệ thống đều hỗ trợ sinh viên nộp bài, giảng viên duyệt và phản hồi. Google Classroom cho phép tải lên file và nhận nhận xét, tương tự với các chức năng nộp bài và `updateDocumentStatus` trong dự án. Điểm khác biệt: Hệ thống của dự án được thiết kế chuyên biệt hơn cho quản lý tài liệu số hóa, với các tính năng như tìm kiếm nâng cao (theo tag, môn học, thời gian), thống kê chi tiết, và thông báo tự động. Trong khi đó, Google Classroom tích hợp sâu với Google Drive và Google Docs, nhưng chưa tập trung nhiều vào phần thống kê, báo cáo như chức năng `getLecturerStatsBySubject`. Ưu thế: Hệ thống của dự án linh hoạt hơn trong môi trường giáo dục Việt Nam, có thể tùy chỉnh ngôn ngữ, quy trình nghiệp vụ, đồng thời chi phí thấp do tự triển khai, không phụ thuộc vào nền tảng đám mây của Google.
- So sánh với Moodle: Điểm tương đồng: Moodle là một hệ thống LMS (Learning Management System) toàn diện, hỗ trợ nộp bài, duyệt bài và quản

lý môn học, tương tự như luồng hoạt động từ SubmitPage đến LecturerPage trong dự án. Cả hai đều có cơ chế phân quyền người dùng (student/lecturer/admin). Điểm khác biệt: Moodle có cấu trúc phức tạp, tích hợp nhiều module như quiz, forum, trong khi dự án tập trung thuần túy vào quản lý tài liệu học thuật. Hệ thống sử dụng Node.js và React, gọn nhẹ và dễ triển khai hơn so với Moodle (dựa trên PHP và yêu cầu tài nguyên máy chủ lớn).

Ưu thế: Dự án đơn giản, dễ sử dụng cho quy mô nhỏ hoặc các trường đại học, đồng thời hỗ trợ API RESTful giúp dễ dàng tích hợp với các hệ thống khác. Ngược lại, Moodle phù hợp cho tổ chức lớn nhưng quá phức tạp với nhu cầu quản lý tài liệu cơ bản.

- So sánh với Microsoft Teams: Điểm tương đồng: Microsoft Teams hỗ trợ chia sẻ file, giao và nộp bài qua module Assignments, tương tự các chức năng upload file và notifications trong dự án. Điểm khác biệt: Teams tập trung mạnh vào giao tiếp thời gian thực (chat, họp trực tuyến), trong khi hệ thống của dự án nhấn mạnh lưu trữ lâu dài, tìm kiếm chuyên sâu (searchDocuments) và quản lý trạng thái tài liệu chi tiết (getDocumentStats).
Ưu thế: Hệ thống hoạt động độc lập, không phụ thuộc vào hệ sinh thái Microsoft (Office 365), phù hợp với các đơn vị không sử dụng dịch vụ bản quyền cao, giúp tiết kiệm chi phí triển khai và vận hành.
- Hạn chế của hệ thống: Mặc dù đáp ứng tốt mục tiêu quản lý tài liệu học thuật, dự án vẫn còn một số hạn chế cần cải thiện trong tương lai:
 - + Khả năng mở rộng: Hệ thống sử dụng MySQL connection pool cơ bản, chưa tối ưu cho tải lớn (thiếu caching như Redis). Khi số lượng tài liệu tăng cao, hiệu suất truy xuất có thể giảm.
 - + Bảo mật: Cơ chế upload file mới chỉ kiểm tra định dạng cơ bản, chưa có virus scanning hoặc mã hóa dữ liệu cho tài liệu nhạy cảm. Hệ thống dùng

JWT authentication nhưng chưa hỗ trợ xác thực hai lớp (2FA) hay rate limiting.

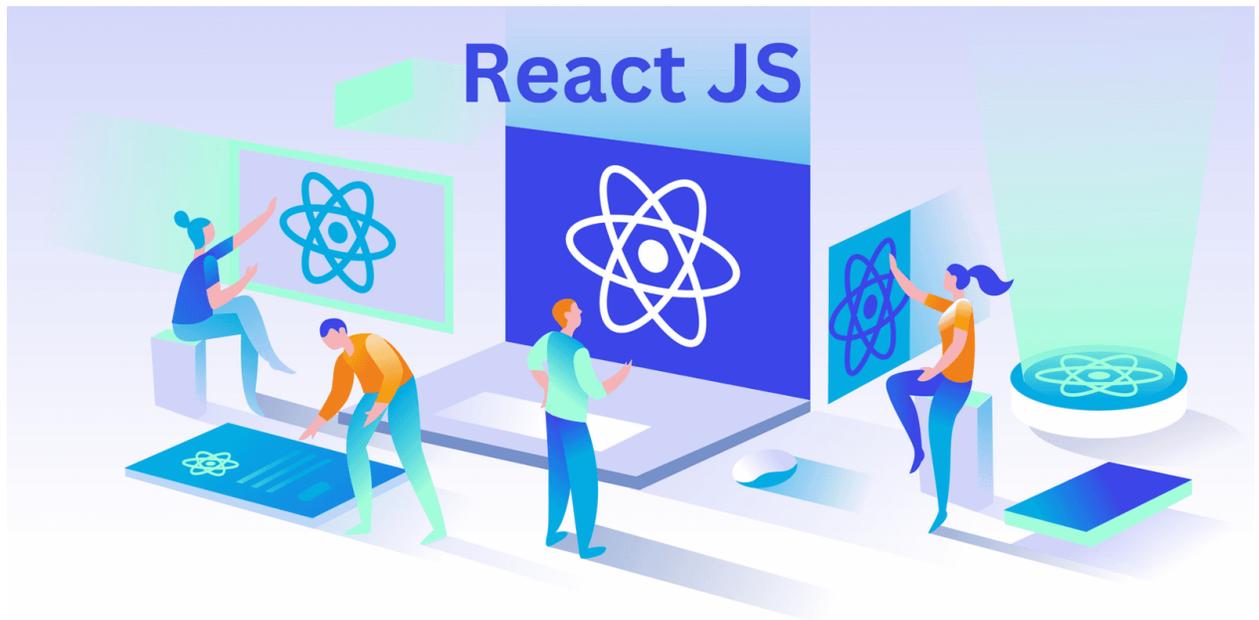
- + Giao diện người dùng: Giao diện React còn đơn giản, chưa tối ưu hoàn toàn cho thiết bị di động, và chưa có tính năng xem trước (preview) tài liệu trực tiếp.
- + Khả năng tích hợp: Chưa hỗ trợ kết nối với các dịch vụ bên thứ ba như Google Drive, OneDrive, hoặc email notification nâng cao. Các thống kê hiện tại dựa trên truy vấn cơ sở dữ liệu, chưa có biểu đồ trực quan hóa dữ liệu.
- + Quản lý phiên bản: Hệ thống chưa hỗ trợ versioning cho tài liệu, dẫn đến nguy cơ mất dữ liệu khi ghi đè file cũ.
- + Tuân thủ pháp lý: Thiếu audit logs chi tiết về hoạt động truy cập và chỉnh sửa, có thể chưa đáp ứng các yêu cầu của GDPR hoặc quy định bảo vệ dữ liệu cá nhân tại Việt Nam.

2.2. Công nghệ sử dụng

2.2.1. Công nghệ phía Client (Frontend)

Giao diện người dùng của hệ thống được xây dựng dưới dạng một ứng dụng trang đơn (Single-Page Application - SPA) nhằm mang lại trải nghiệm nhanh, mượt mà và hiện đại. Các công nghệ chính được sử dụng bao gồm:

- ReactJS: Là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng, được chọn làm nền tảng chính cho dự án. ReactJS cho phép xây dựng các thành phần UI (components) độc lập, có thể tái sử dụng, giúp mã nguồn trở nên có tổ chức, dễ quản lý và bảo trì. Kiến trúc dựa trên component cũng giúp tăng tốc độ phát triển.



Hình 2.1. Công nghệ React.

- Material-UI (MUI): Đây là bộ thư viện UI component toàn diện, cung cấp các thành phần giao diện theo ngôn ngữ thiết kế Material Design của Google. Việc sử dụng MUI giúp xây dựng một giao diện người dùng chuyên nghiệp, nhất quán và đẹp mắt một cách nhanh chóng. Các thành phần như Card, Table, Button, Dialog, Tabs và hệ thống Grid đã được tận dụng triệt để trong toàn bộ ứng dụng.
- React Router: Thư viện này được sử dụng để quản lý việc điều hướng (routing) phía client. Nó cho phép tạo ra các URL khác nhau cho từng trang (ví dụ: /dashboard, /submit, /admin) mà không cần tải lại toàn bộ trang web, tạo ra trải nghiệm liền mạch đặc trưng của một SPA.
- React Hooks & Context API: Dự án hoàn toàn sử dụng các thành phần chức năng (functional components) kết hợp với React Hooks (useState, useEffect, useContext, v.v.). Đặc biệt, Context API được dùng để quản lý trạng thái toàn cục của ứng dụng, cụ thể là thông tin xác thực người dùng (AuthContext) và chủ đề giao diện (ThemeContext), giúp chia sẻ dữ liệu

giữa các component một cách hiệu quả mà không cần đến các thư viện quản lý trạng thái phức tạp.

- Recharts: Là một thư viện biểu đồ được xây dựng trên React, được sử dụng để trực quan hóa dữ liệu trong các trang Dashboard. Các biểu đồ cột, tròn giúp người dùng (đặc biệt là Admin và Giảng viên) dễ dàng nắm bắt các số liệu thống kê về hoạt động của hệ thống.
- Framer Motion: Thư viện này được tích hợp để tạo ra các hiệu ứng chuyển động (animation) tinh tế cho các thành-phần giao diện, giúp tăng tính tương tác và mang lại trải nghiệm người dùng sống động, hấp dẫn hơn.
- jsPDF & jspdf-autotable: Được sử dụng để triển khai chức năng "Xuất Báo cáo" cho Giảng viên. Các thư viện này cho phép tạo file PDF chứa dữ liệu thống kê trực tiếp từ trình duyệt, cung cấp một tính năng hữu ích và chuyên nghiệp.

2.2.2. Công nghệ phía Server (Backend)



Hình 2.2. Công nghệ Nodejs Express

Phía server của hệ thống được xây dựng theo kiến trúc RESTful API, có nhiệm vụ xử lý logic nghiệp vụ, xác thực người dùng và cung cấp dữ liệu cho client. Các công nghệ và thư viện chính được sử dụng bao gồm:

- Node.js: Được chọn làm môi trường thực thi phía server. Node.js với mô hình I/O không đồng bộ, hướng sự kiện (event-driven, non-blocking I/O) cho phép xây dựng các ứng dụng web có hiệu năng cao, khả năng mở rộng tốt và xử lý được nhiều kết nối đồng thời.
- Express.js: Là một framework web tối giản và linh hoạt cho Node.js, được sử dụng để xây dựng bộ khung cho RESTful API. Express.js giúp đơn giản hóa việc định tuyến (routing), quản lý các request/response và tích hợp các middleware một cách có cấu trúc.
- MySQL và thư viện mysql2: Hệ quản trị cơ sở dữ liệu quan hệ MySQL được lựa chọn để lưu trữ dữ liệu nhờ tính ổn định, bền vững và được cộng đồng hỗ trợ mạnh mẽ. Thư viện mysql2 được sử dụng để kết nối ứng dụng Node.js với cơ sở dữ liệu MySQL. Thư viện này hỗ trợ Promises và connection pools, giúp tối ưu hóa hiệu suất truy vấn và quản lý kết nối cơ sở dữ liệu một cách hiệu quả.



Hình 2.3. Công nghệ MySQL

- Xác thực và Bảo mật:
 - + JSON Web Token (JWT): Hệ thống sử dụng JWT để xác thực người dùng. Sau khi đăng nhập thành công, server sẽ tạo ra một token chứa thông tin người dùng (ID, vai trò) và gửi về cho client. Client sẽ đính kèm token này trong header của mỗi request tiếp theo để xác thực quyền truy cập mà không cần duy trì phiên (session) ở phía server.
 - + Bcrypt.js: Thư viện này được dùng để băm (hash) mật khẩu của người dùng trước khi lưu vào cơ sở dữ liệu. Việc này đảm bảo rằng mật khẩu không được lưu dưới dạng văn bản thuần, giúp tăng cường bảo mật cho thông tin tài khoản.
- Middleware: Hệ thống sử dụng nhiều middleware để xử lý các tác vụ khác nhau:
 - + cors: Cho phép client (chạy trên domain khác) có thể gửi request đến server, giải quyết vấn đề Cross-Origin Resource Sharing.
 - + helmet: Tự động thiết lập các HTTP header cần thiết để bảo vệ ứng dụng khỏi các lỗ hổng bảo mật web phổ biến.
 - + morgan: Ghi lại (log) thông tin về các HTTP request, rất hữu ích trong quá trình phát triển và gỡ lỗi.
 - + multer: Xử lý dữ liệu multipart/form-data, được sử dụng chủ yếu cho chức năng tải file (upload) của người dùng.
 - + Middleware tùy chỉnh: Hệ thống tự xây dựng các middleware để kiểm tra token (authenticateToken) và phân quyền truy cập dựa trên vai trò người dùng (requireRole), đảm bảo chỉ những người dùng hợp lệ mới có thể truy cập các tài nguyên được bảo vệ.

- dotenv: Thư viện này được dùng để quản lý các biến môi trường (environment variables) như thông tin kết nối cơ sở dữ liệu, khóa bí mật của JWT, v.v. Việc tách các thông tin cấu hình nhạy cảm ra khỏi mã nguồn giúp tăng tính bảo mật và linh hoạt cho ứng dụng.

2.3. Yêu cầu chức năng

2.3.1. Chức năng cho Sinh viên

- Quản lý bài tập: Sinh viên đăng nhập vào hệ thống, chọn môn học và tiến hành nộp (upload) file bài tập lớn của mình. Hệ thống ghi nhận bài nộp ở trạng thái "Chờ duyệt".
- Thống kê cá nhân: Sinh viên có thể xem lại lịch sử các bài đã nộp và trạng thái của chúng (ví dụ: Đã duyệt, Bị từ chối, Chờ duyệt).

2.3.2. Chức năng cho Giảng viên

- Quản lý bài nộp: Giảng viên xem danh sách các bài tập sinh viên đã nộp thuộc môn học mình quản lý. Giảng viên có quyền xem nội dung và thực hiện hành động Duyệt (cho phép hiển thị công khai) hoặc Từ chối (kèm lý do).
- Thống kê môn học: Giảng viên xem được các báo cáo thống kê về môn học của mình (ví dụ: tổng số bài nộp, tỷ lệ sinh viên đã nộp/chưa nộp).

2.3.3. Chức năng cho Quản trị viên (Admin)

- Quản lý người dùng: Admin có toàn quyền quản lý tài khoản trên hệ thống, bao gồm: tạo, sửa, xóa tài khoản; phân quyền (thăng cấp/giáng cấp) cho người dùng (ví dụ: cấp quyền Giảng viên cho một tài khoản).
- Quản lý môn học: Admin quản lý danh mục các môn học trên toàn hệ thống (thêm môn học mới, sửa tên, xóa môn học).
- Thống kê hệ thống: Admin xem được các số liệu tổng quan (dashboard) về toàn bộ hệ thống (tổng số người dùng, tổng số tài liệu, dung lượng lưu trữ...).

2.3.4. Chức năng chung (Dành cho cả 3 vai trò)

- Tìm kiếm và Tải xuống: Tất cả các vai trò sau khi đăng nhập đều có quyền truy cập kho tài liệu chung. Người dùng có thể tìm kiếm tài liệu (bài tập lớn, đồ án đã được duyệt) theo tên, môn học, hoặc từ khóa. Khi tìm thấy, người dùng có quyền xem và tải file về máy để tham khảo.

2.4. Yêu cầu phi chức năng

Bên cạnh các yêu cầu về chức năng, hệ thống cũng cần đáp ứng các tiêu chí về chất lượng, hiệu suất và bảo mật trong quá trình vận hành.

- Hiệu năng (Performance):
 - + Thời gian phản hồi: Hệ thống phải phản hồi các tương tác của người dùng (tải trang, tìm kiếm, nộp bài) trong thời gian ngắn, dưới 3 giây cho hầu hết các tác vụ thông thường.
 - + Tải trang phía Client: Ứng dụng Frontend (React) phải được tối ưu để có tốc độ tải ban đầu nhanh, mang lại trải nghiệm mượt mà cho người dùng.
 - + Khả năng chịu tải: Backend phải có khả năng xử lý nhiều yêu cầu đồng thời từ người dùng mà không làm giảm hiệu suất chung của hệ thống.
- Bảo mật (Security):
 - + Xác thực: Mọi quyền truy cập vào các tài nguyên, chức năng quan trọng đều phải được xác thực thông qua cơ chế JSON Web Token (JWT).
 - + Phân quyền: Hệ thống phải đảm bảo người dùng chỉ có thể thực hiện các chức năng và truy cập dữ liệu đúng với vai trò (role) đã được cấp phát.
 - + Mã hóa mật khẩu: Mật khẩu người dùng phải được băm (hash) bằng thuật toán an toàn (Bcrypt) trước khi lưu vào cơ sở dữ liệu.

- + Bảo vệ API: Các API endpoint phía backend phải được bảo vệ, chống lại các hình thức tấn công phổ biến như Cross-Site Scripting (XSS), SQL Injection. Việc sử dụng thư viện helmet là một phần của yêu cầu này.
- Tính dễ sử dụng (Usability):
 - + Giao diện thân thiện: Giao diện người dùng phải được thiết kế nhất quán, trực quan và dễ thao tác cho cả 3 nhóm người dùng, kể cả những người không có nhiều kinh nghiệm về công nghệ.
 - + Phản hồi rõ ràng: Hệ thống cần cung cấp các thông báo, chỉ dẫn rõ ràng cho người dùng trong quá trình sử dụng (ví dụ: thông báo thành công, cảnh báo lỗi, trạng thái tải dữ liệu).
 - + Tính nhất quán: Thiết kế, bố cục và luồng hoạt động phải nhất quán trên toàn bộ các trang của ứng dụng.
- Khả năng bảo trì và mở rộng (Maintainability & Scalability):
 - + Mã nguồn có cấu trúc: Mã nguồn cả frontend và backend phải được tổ chức thành các module, component rõ ràng, tuân thủ các quy ước đặt tên và định dạng, giúp việc bảo trì và nâng cấp trở nên dễ dàng hơn.
 - + Kiến trúc linh hoạt: Kiến trúc RESTful API và Single-Page Application (SPA) cho phép hệ thống có thể được mở rộng trong tương lai (thêm chức năng mới, tích hợp dịch vụ bên thứ ba) một cách độc lập giữa backend và frontend.

CHƯƠNG III: THIẾT KẾ HỆ THỐNG

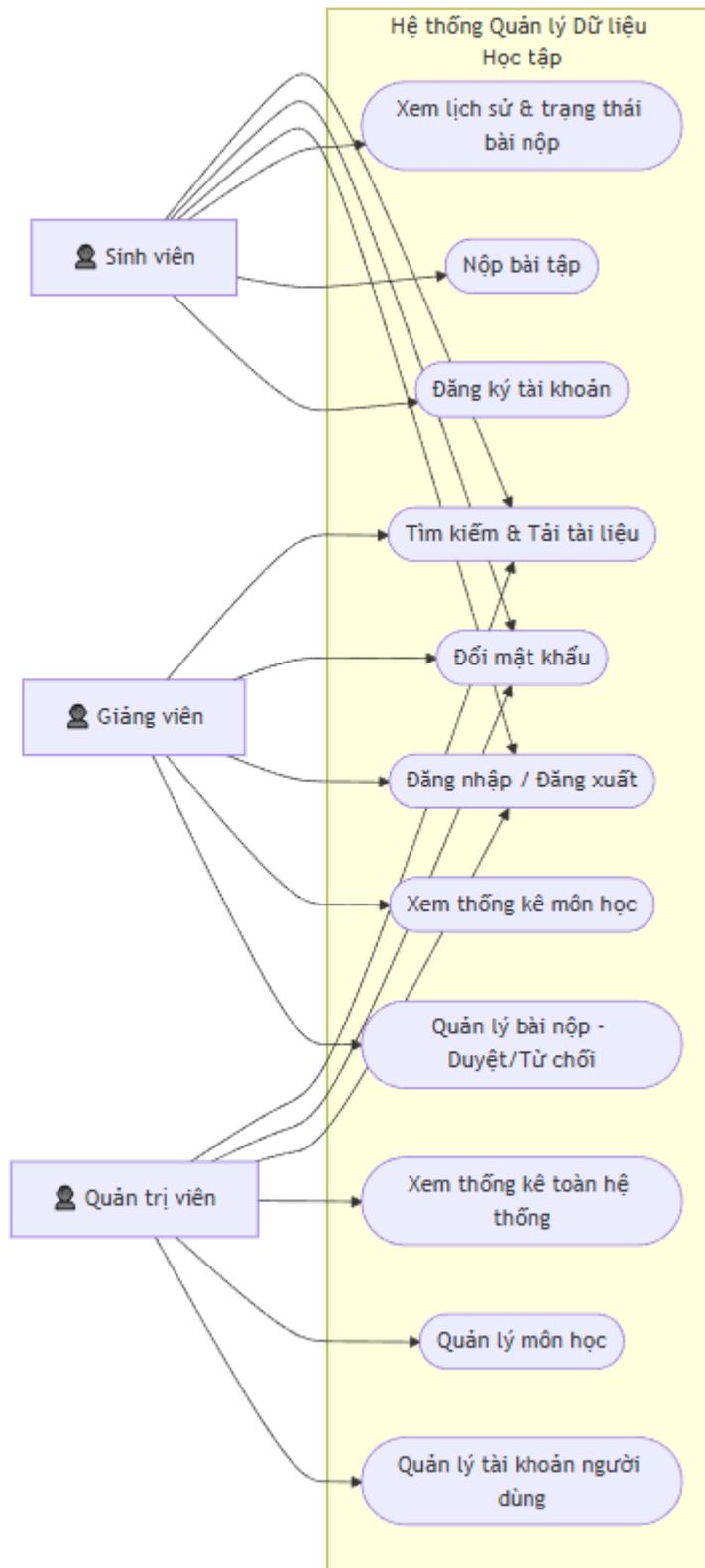
3.1. Sơ đồ Use Case

3.1.1. Use Case tổng quát

Biểu đồ Use Case tổng quát mô tả các chức năng chính của hệ thống và sự tương tác của các tác nhân (người dùng) với các chức năng đó. Hệ thống có ba tác nhân chính: Sinh viên, Giảng viên, và Quản trị viên (Admin).

- Sinh viên: Là người dùng cơ bản, có thể đăng ký tài khoản, nộp bài tập/tài liệu, xem lại lịch sử và trạng thái các bài đã nộp.
- Giảng viên: Có vai trò quản lý chuyên môn, chịu trách nhiệm xem xét, duyệt hoặc từ chối các bài nộp của sinh viên trong môn học mình phụ trách, cũng như xem các thống kê liên quan.
- Quản trị viên (Admin): Là người có quyền cao nhất, chịu trách nhiệm quản lý toàn bộ hệ thống, bao gồm quản lý tài khoản người dùng, quản lý danh mục môn học và xem các thống kê tổng quan.

Ngoài ra, tất cả các tác nhân sau khi đăng nhập đều có thể thực hiện các chức năng chung như thay đổi mật khẩu, tìm kiếm và tải các tài liệu đã được duyệt.

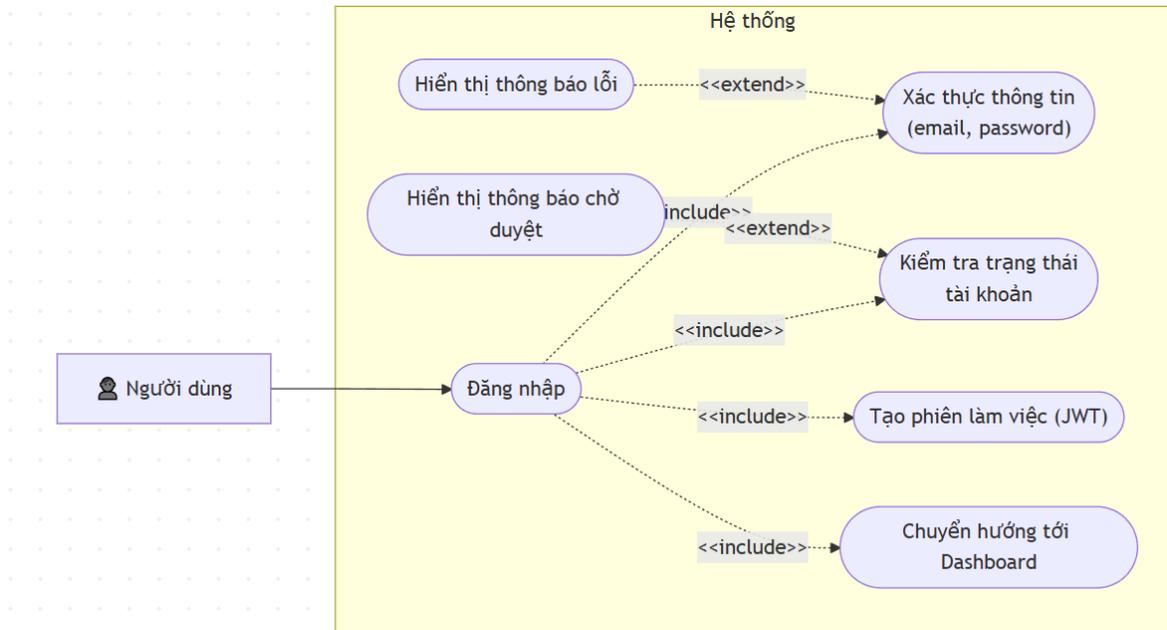


Hình 3.1. Sơ đồ Usecase tổng quát.

3.1.2. Biểu đồ usecase phân rã.

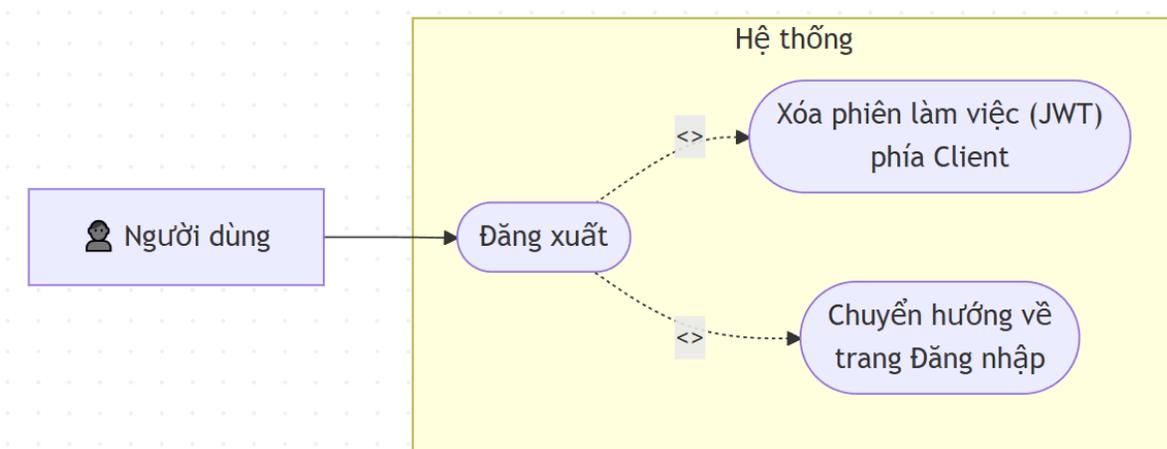
A. Chức năng chung (Dành cho tất cả người dùng đã đăng nhập)

Usecase đăng nhập



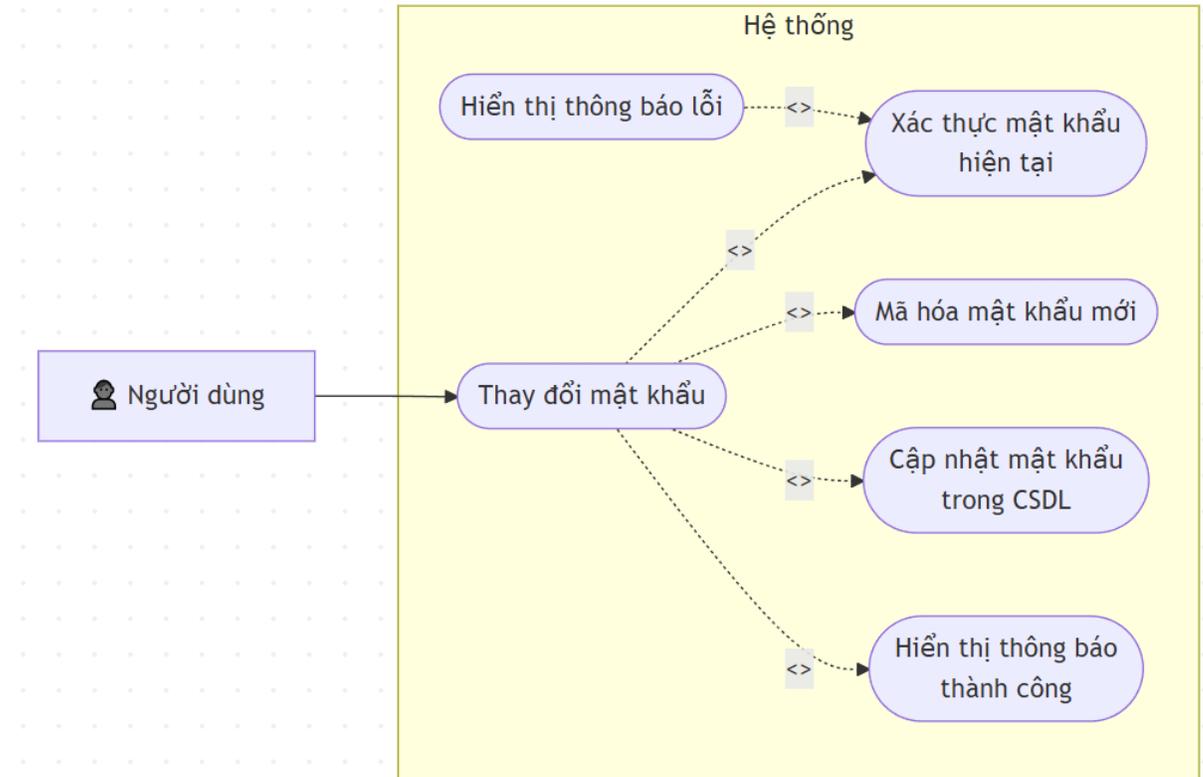
Hình 3.2. Biểu đồ phân rã UseCase đăng nhập.

Usecase đăng xuất



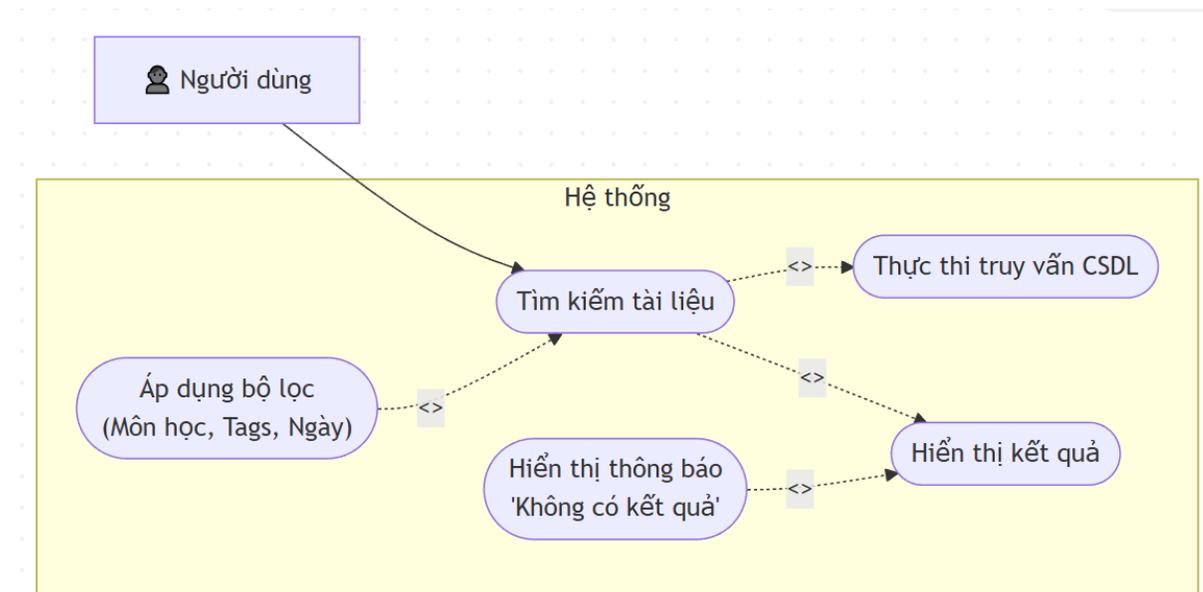
Hình 3.3. Biểu đồ phân rã UseCase đăng xuất.

Usecase thay đổi mật khẩu



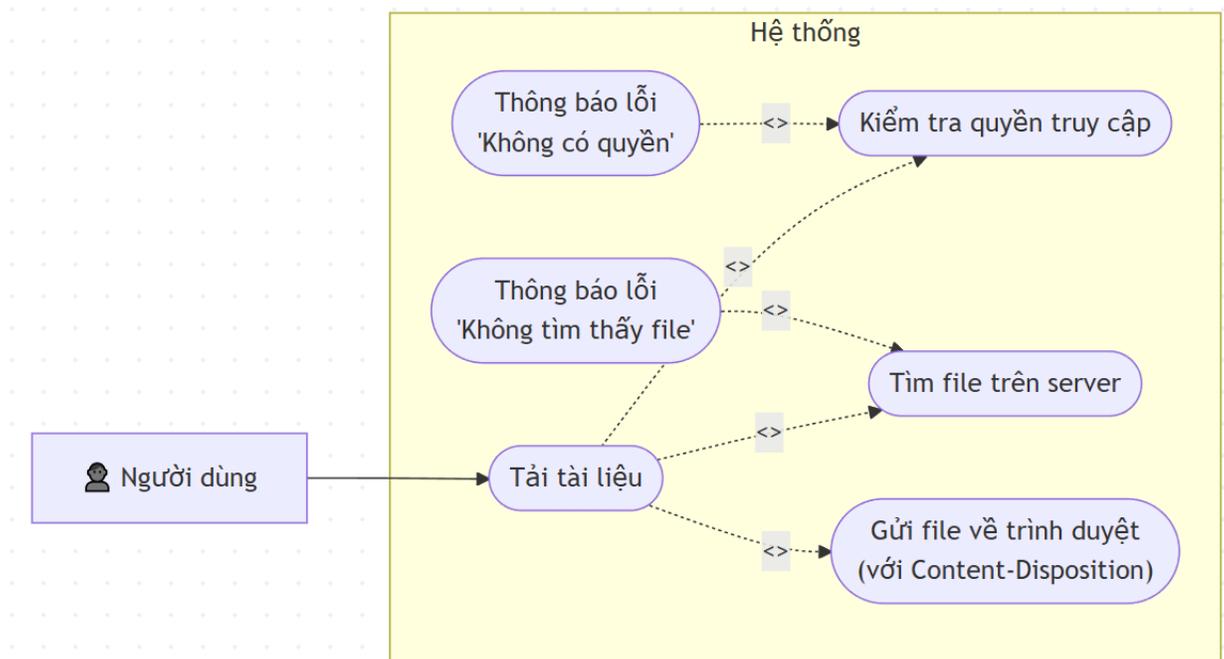
Hình 3.3. Biểu đồ phân rã UseCase đổi mật khẩu.

Usecase Tìm tài liệu



Hình 3.4. Biểu đồ phân rã UseCase tìm kiếm tài liệu.

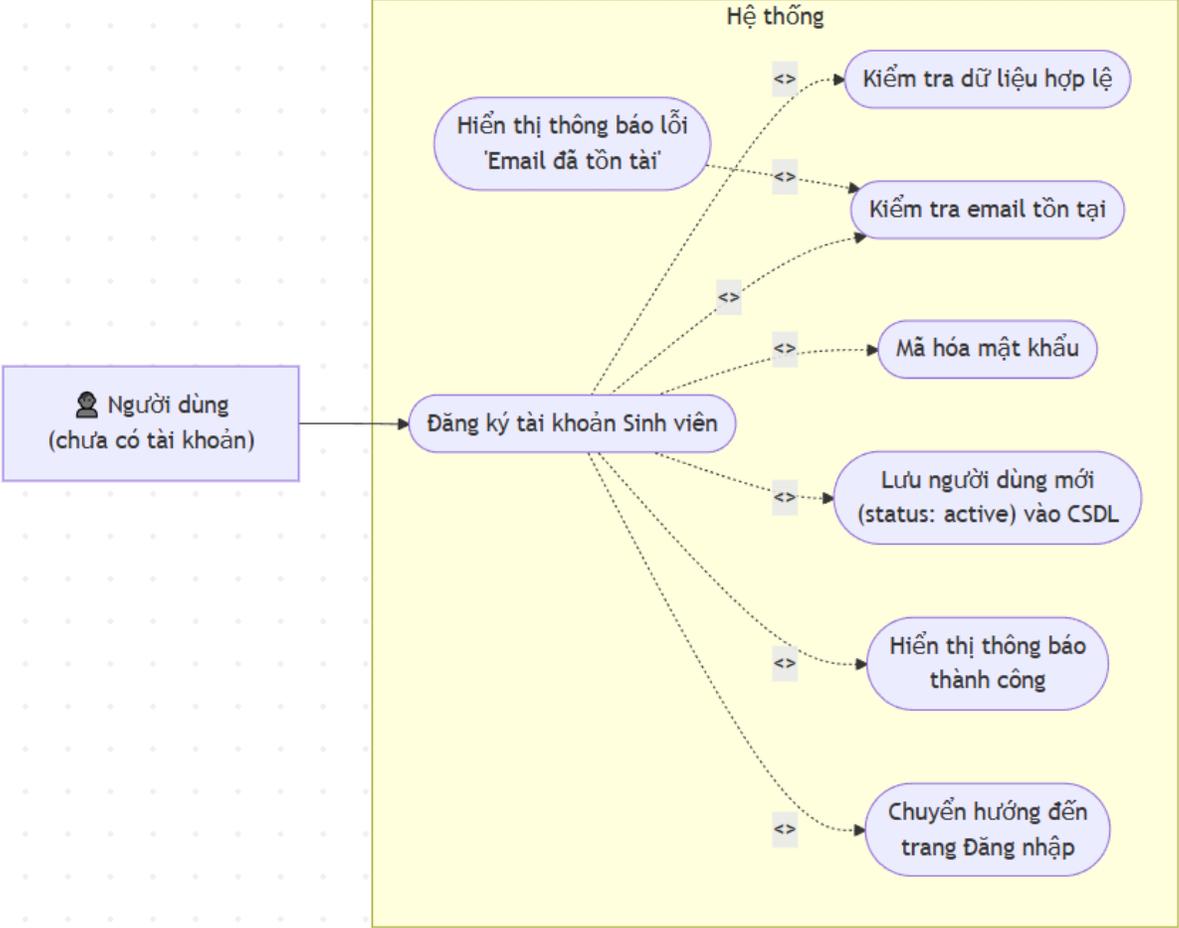
Usecase tải tài liệu về máy



Hình 3.5. Biểu đồ phân rã UseCase tải tài liệu về máy.

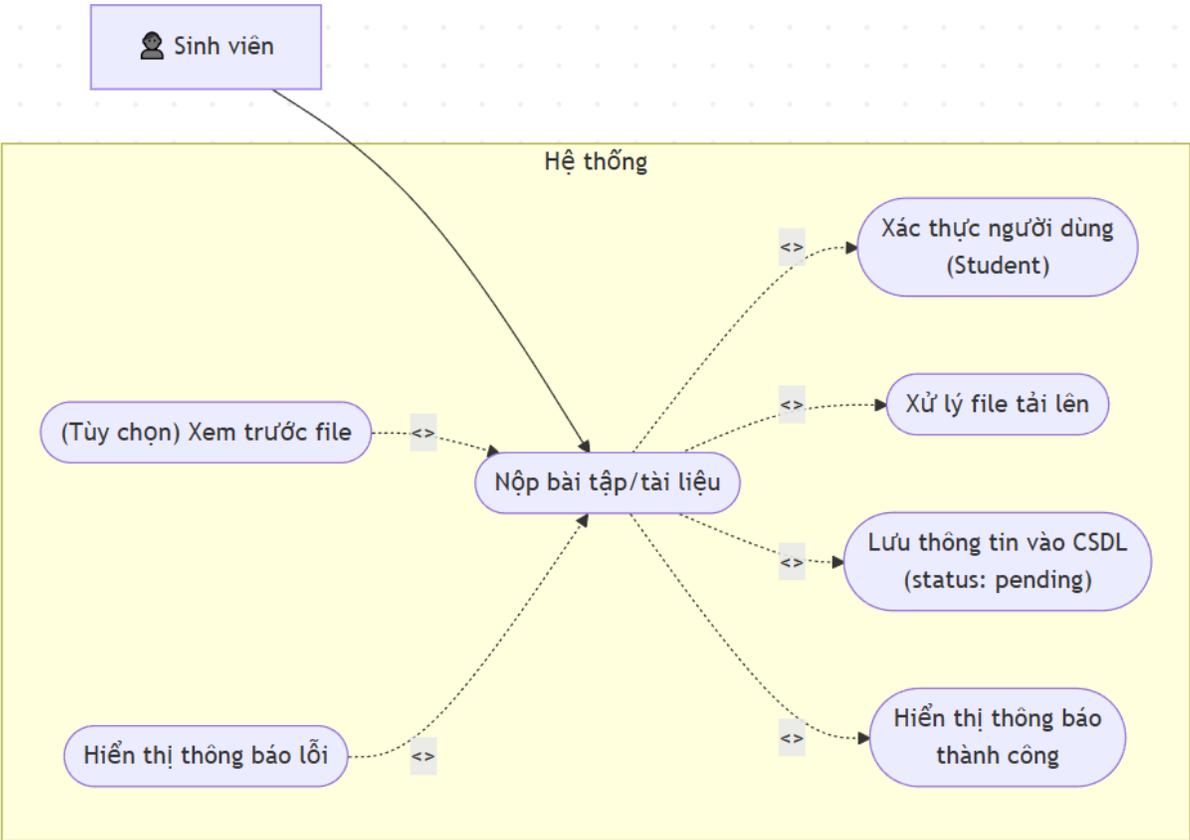
B. Chức năng dành cho Sinh viên.

Usecase đăng ký



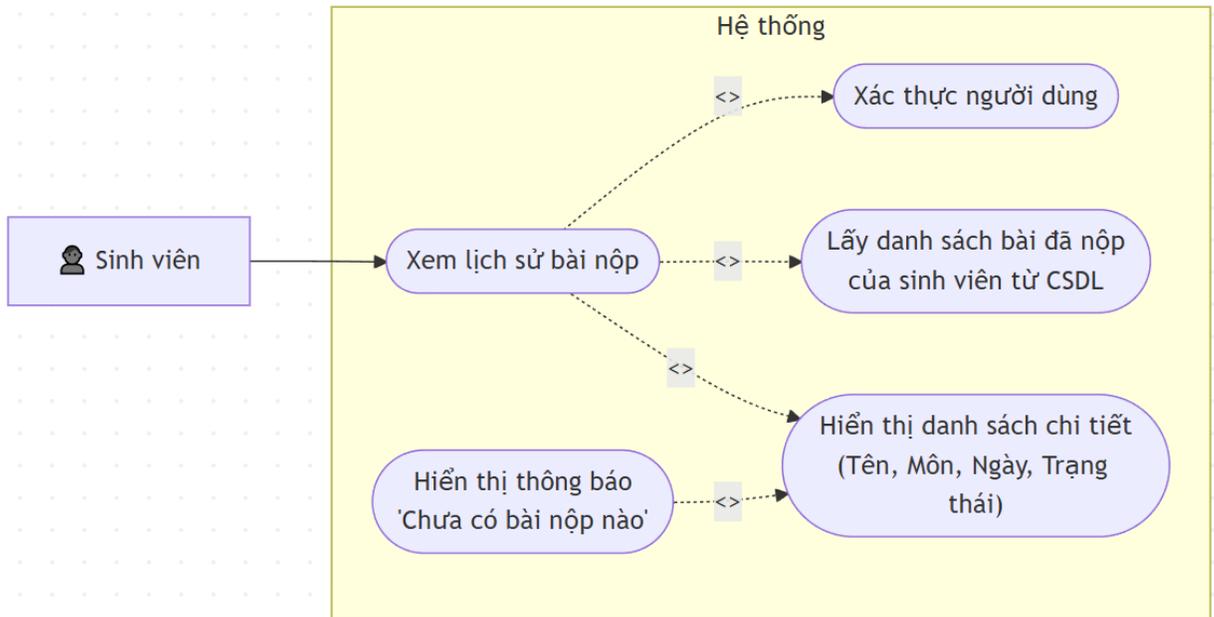
Hình 3.6. Biểu đồ phân rã UseCase đăng ký.

Usecase nộp tài liệu



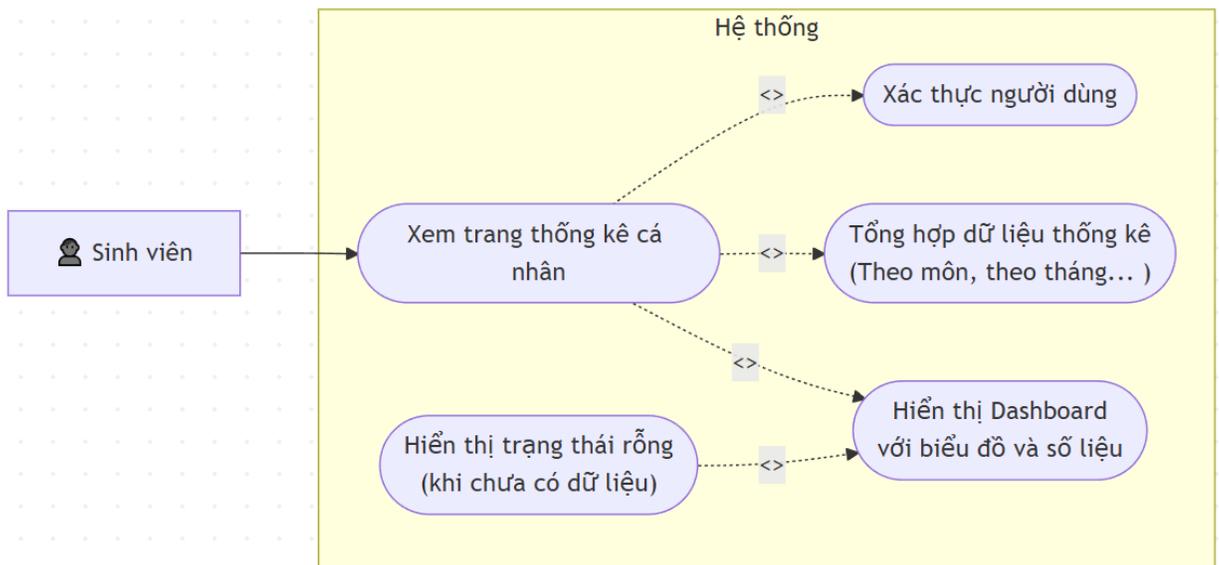
Hình 3.7. Biểu đồ phân rã UseCase nộp tài liệu.

Usecase xem lịch sử bài nộp



Hình 3.8. Biểu đồ phân rã UseCase xem lịch sử bài nộp.

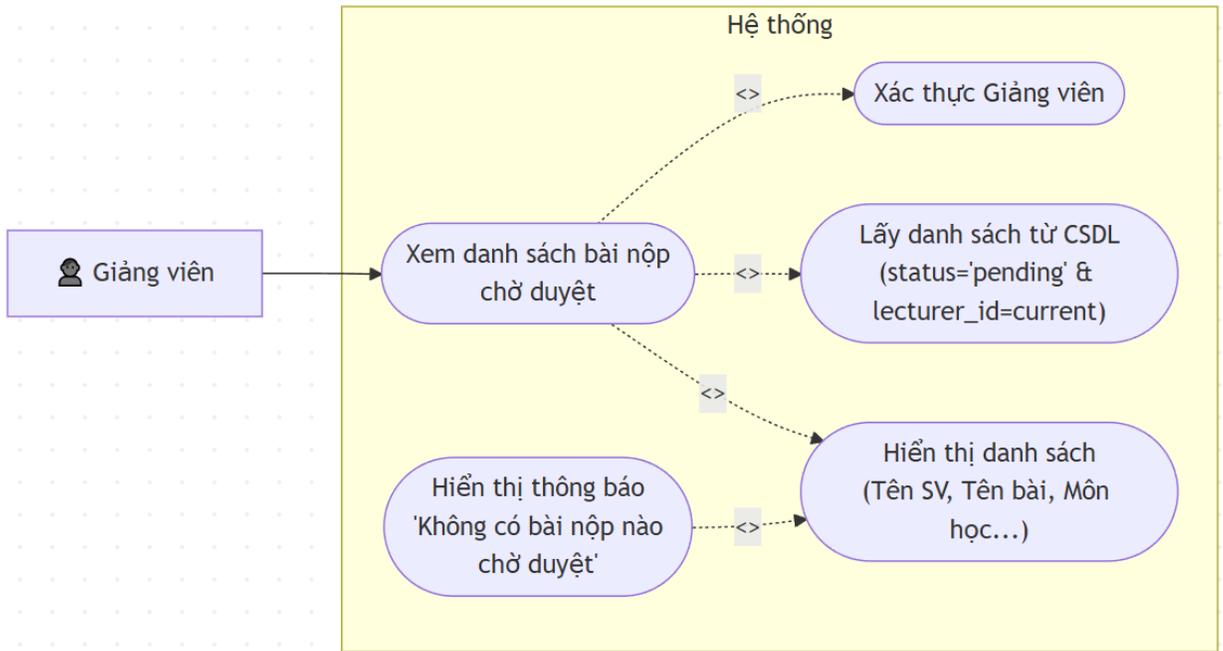
UseCase xem trang thống kê cá nhân



Hình 3.9. Biểu đồ phân rã UseCase xem trang thống kê cá nhân.

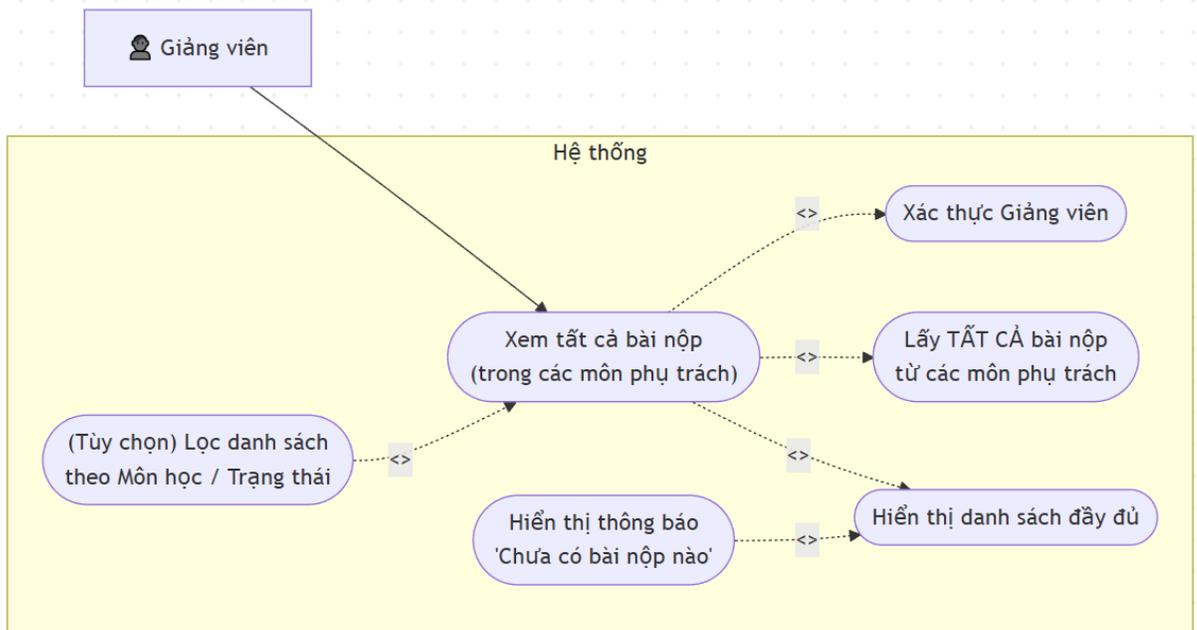
C. Chức năng dành cho Giảng viên

UseCase xem danh sách bài nộp chờ duyệt



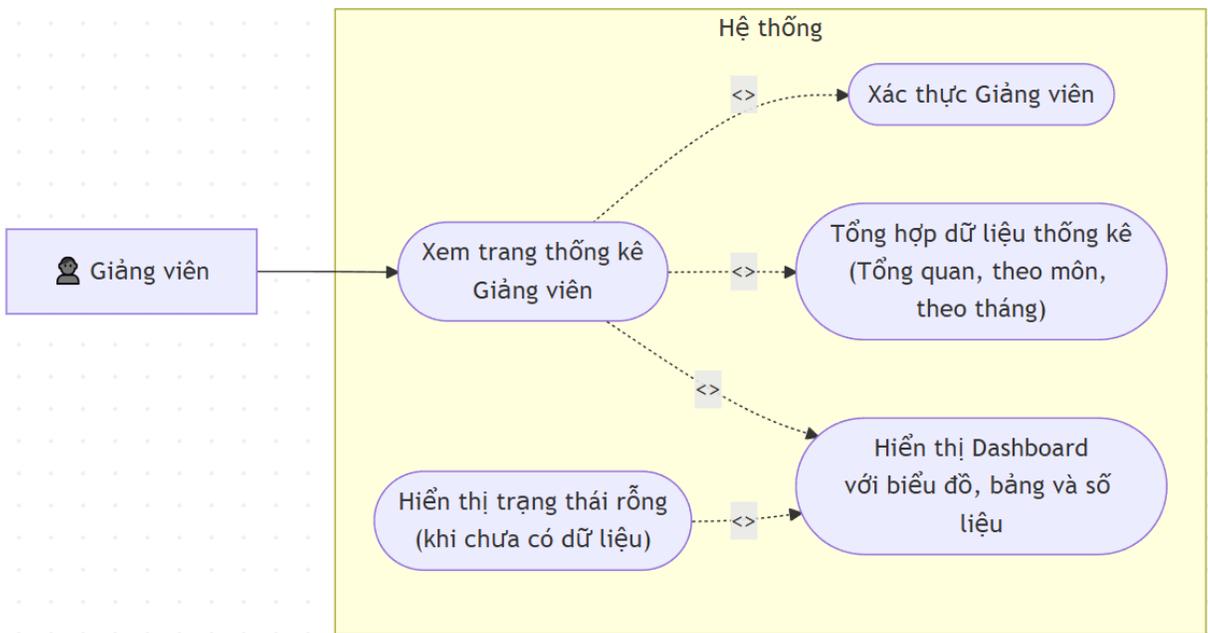
Hình 3.10. Biểu đồ phân rã UseCase xem danh sách bài nộp chờ duyệt.

UseCase xem danh sách tất cả bài nộp



Hình 3.11. Biểu đồ phân rã UseCase xem danh sách tất cả bài nộp.

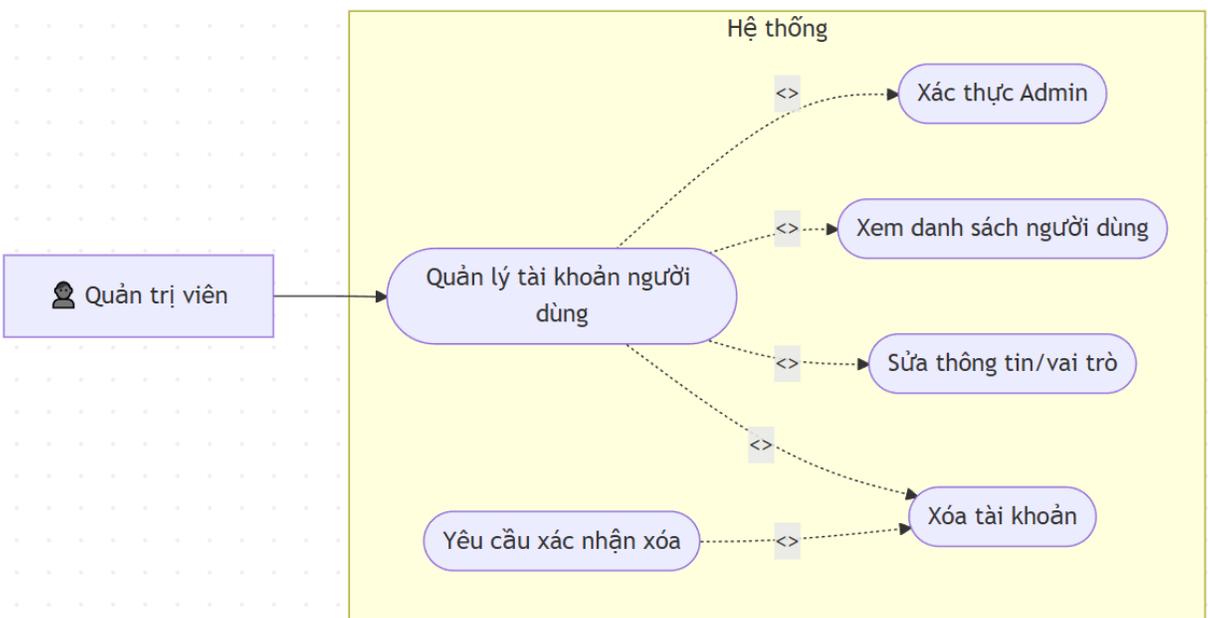
UseCase xem trang thống kê Giảng viên



Hình 3.12. Biểu đồ phân rã UseCase xem trang thống kê giảng viên.

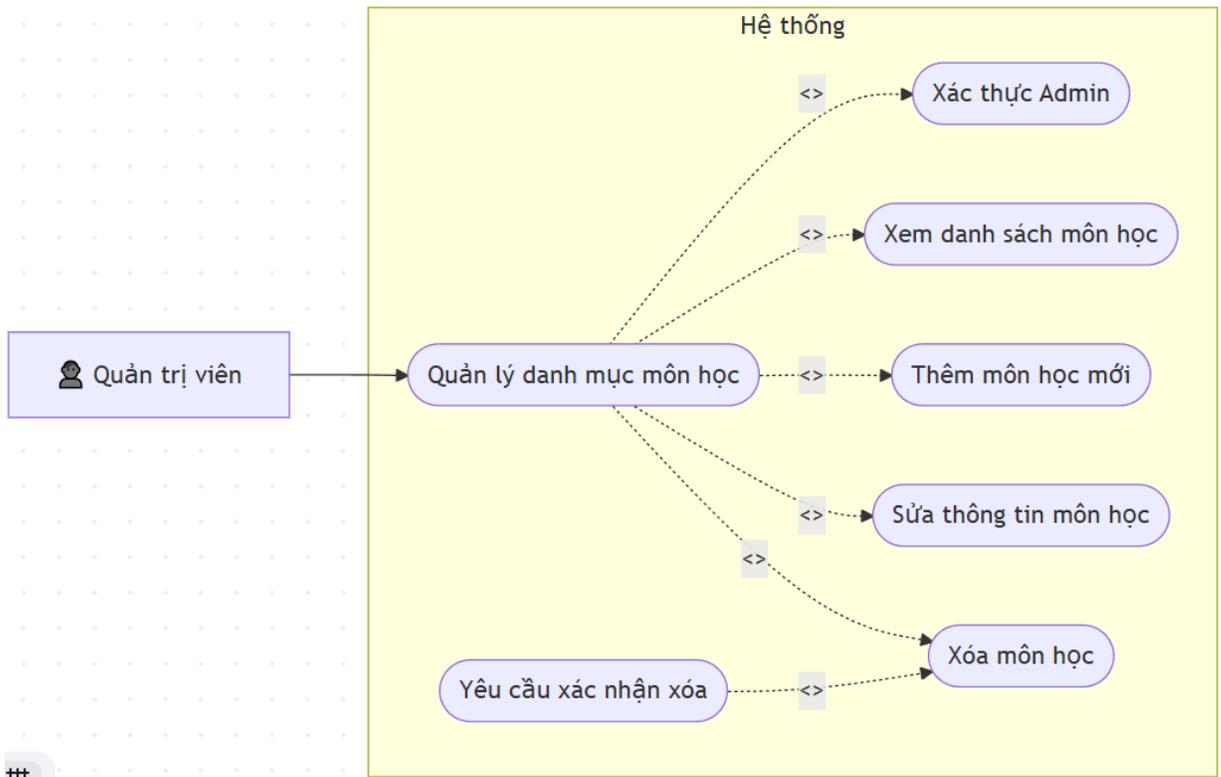
D. Chức năng của Quản trị viên (Admin)

Usecase quản lý tài khoản người dùng



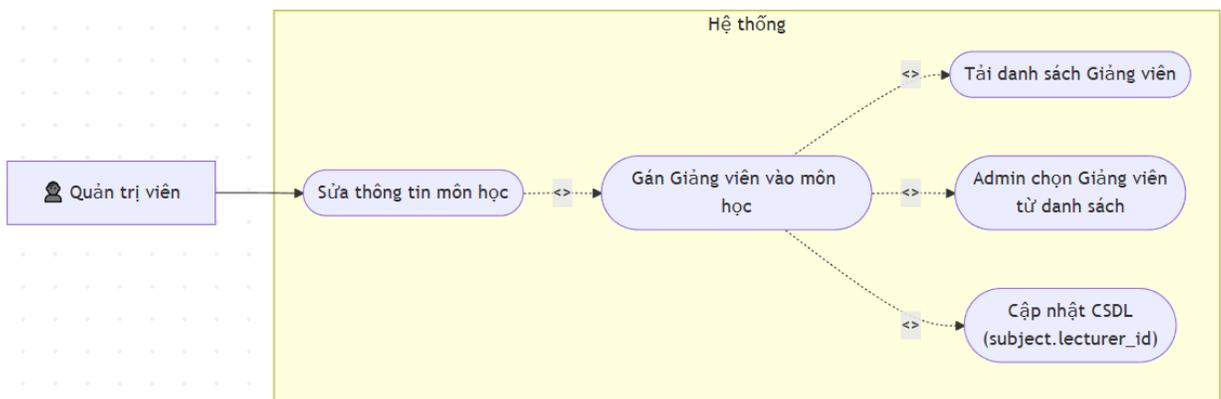
Hình 3.13. Biểu đồ phân rã UseCase quản lý người dùng.

Usecase quản lý danh mục môn học



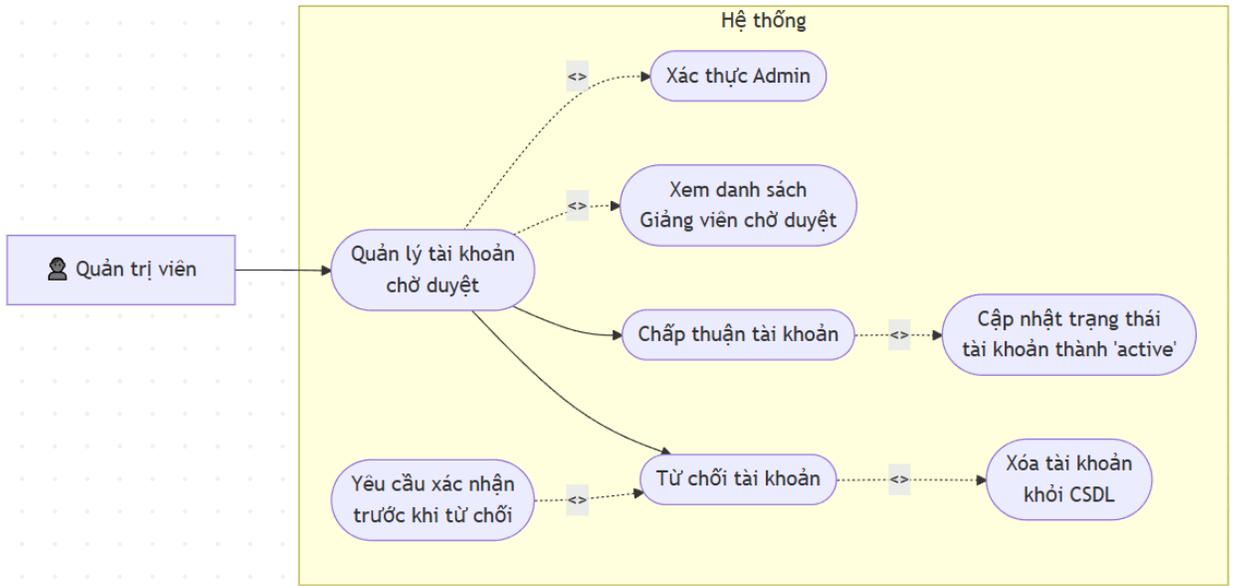
Hình 3.14. Biểu đồ phân rã UseCase quản lý danh mục môn học.

Usecase gán giảng viên vào môn học



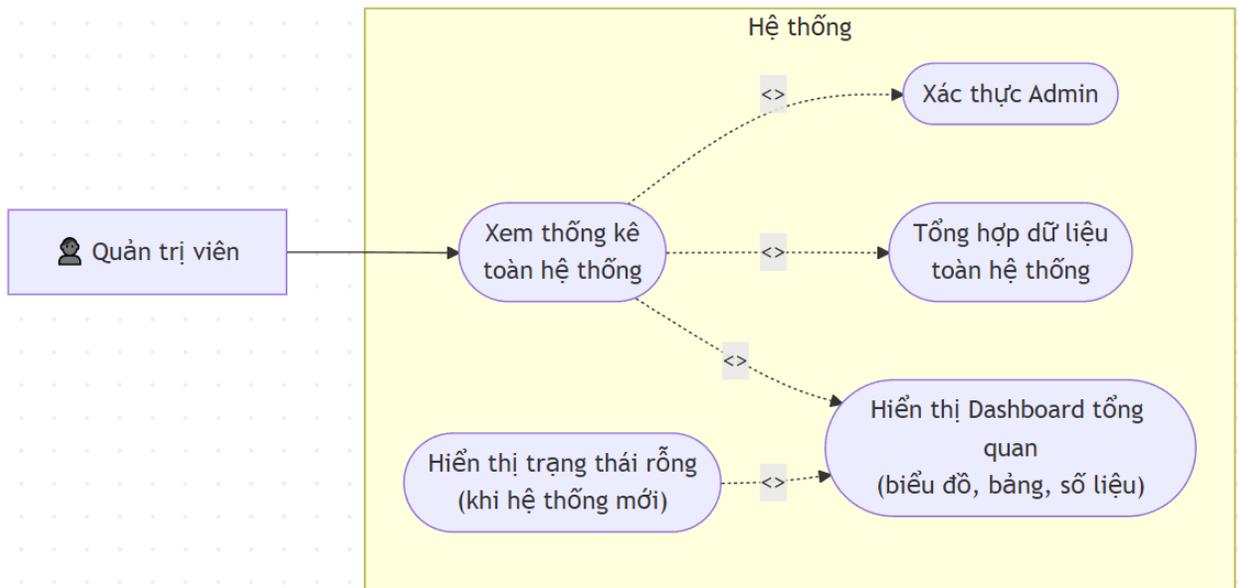
Hình 3.15. Biểu đồ phân rã UseCase gán giảng viên vào môn học.

Usecase Quản lý tài khoản Giảng viên đang chờ phê duyệt



Hình 3.16. Biểu đồ phân rã UseCase Quản lý giảng viên đang chờ phê duyệt.

UseCase Xem thống kê toàn hệ thống



Hình 3.17. Biểu đồ phân rã UseCase xem thông báo toàn hệ thống.

3.2. Đặc tả chức năng

3.2.1. Use case Đăng nhập

Tên Use Case		Đăng nhập	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng (Sinh viên, Giảng viên, Admin) đã có tài khoản truy cập vào hệ thống bằng email và mật khẩu.	
Tác nhân		Người dùng (chưa đăng nhập)	
Tiền điều kiện		Người dùng đã truy cập trang web và đã có một tài khoản được đăng ký trước đó trong hệ thống.	
Hậu điều kiện	Thành công	Người dùng được xác thực, hệ thống tạo phiên làm việc (token), chuyển hướng người dùng đến trang chức năng chính tương ứng với vai trò của họ.	
	Lỗi	Hệ thống hiển thị thông báo lỗi, người dùng vẫn ở lại trang Đăng nhập.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Người dùng truy cập vào trang Đăng nhập của hệ thống. 2. Hệ thống hiển thị một biểu mẫu (form) yêu cầu Người dùng nhập "Email" và "Mật khẩu". 3. Người dùng nhập email và mật khẩu đã đăng ký, sau đó nhấn nút "Đăng nhập". 4. Hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào (ví dụ: định dạng email). Nếu không hợp lệ, chuyển đến Luồng A. 			

<ol style="list-style-type: none"> 5. Hệ thống gửi thông tin xác thực đến máy chủ (backend). 6. Máy chủ tìm kiếm tài khoản người dùng trong cơ sở dữ liệu dựa trên email. Nếu không tìm thấy hoặc mật khẩu không khớp, chuyển đến Luồng B. 7. Máy chủ kiểm tra trạng thái của tài khoản. Nếu là tài khoản Giảng viên đang chờ phê duyệt, chuyển đến Luồng C. 8. Xác thực thành công. Hệ thống tạo một JSON Web Token (JWT) chứa thông tin định danh người dùng (ID, vai trò). 9. Hệ thống trả token về cho trình duyệt. Trình duyệt lưu lại và tự động chuyển hướng người dùng đến trang Dashboard tương ứng với vai trò của họ. Use case kết thúc thành công.
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Dữ liệu nhập không hợp lệ
<ul style="list-style-type: none"> - Tại bước 4 của luồng chính, nếu Người dùng không nhập email/mật khẩu hoặc nhập sai định dạng email. - Hệ thống hiển thị thông báo lỗi ngay bên dưới trường nhập liệu tương ứng (ví dụ: "Vui lòng nhập email", "Mật khẩu không được để trống"). - Use case quay trở lại bước 3 của luồng chính.
Luồng B: Thông tin đăng nhập không chính xác
<ul style="list-style-type: none"> - Tại bước 6 của luồng chính, nếu email không tồn tại trong hệ thống hoặc mật khẩu không khớp. - Hệ thống hiển thị một thông báo lỗi chung: "Email hoặc mật khẩu không chính xác". - Use case quay trở lại bước 3 của luồng chính.
Luồng C: Tài khoản Giảng viên đang chờ phê duyệt
<ul style="list-style-type: none"> - Tại bước 7 của luồng chính, nếu tài khoản đăng nhập là của Giảng viên và có trạng thái là "chờ duyệt".

- Hệ thống hiển thị một thông báo: "Tài khoản của bạn đang chờ Quản trị viên phê duyệt."
- Use case kết thúc. Người dùng vẫn ở lại trang đăng nhập.

3.2.2. Use case Đăng ký

Tên Use Case		Đăng ký tài khoản sinh viên	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng mới (chưa có tài khoản) tự đăng ký một tài khoản trên hệ thống với vai trò mặc định là "Sinh viên".	
Tác nhân		Người dùng (chưa đăng nhập)	
Tiền điều kiện		Người dùng đã truy cập vào trang web và đang ở trang Đăng ký.	
Hậu điều kiện	Thành công	Một tài khoản sinh viên mới được tạo và lưu vào cơ sở dữ liệu. Hệ thống chuyển hướng người dùng đến trang Đăng nhập kèm theo thông báo thành công.	
	Lỗi	Tài khoản không được tạo, hệ thống hiển thị thông báo lỗi và người dùng vẫn ở lại trang Đăng ký.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Người dùng chọn chức năng "Đăng ký" trên giao diện. 2. Hệ thống hiển thị trang Đăng ký, yêu cầu Người dùng nhập các thông tin: "Họ và tên", "Email", "Mật khẩu", và "Xác nhận mật khẩu". 3. Người dùng điền đầy đủ thông tin vào biểu mẫu và nhấn nút "Đăng ký". 			

4. Hệ thống kiểm tra tính hợp lệ của dữ liệu ở phía client (ví dụ: các trường không được bỏ trống, mật khẩu và mật khẩu xác nhận phải khớp). Nếu không hợp lệ, chuyển đến Luồng A.
5. Hệ thống gửi thông tin đăng ký đến máy chủ (backend).
6. Máy chủ kiểm tra xem email người dùng cung cấp đã tồn tại trong cơ sở dữ liệu hay chưa. Nếu đã tồn tại, chuyển đến Luồng B.
7. Máy chủ tiến hành mã hóa mật khẩu để bảo mật.
8. Máy chủ lưu thông tin tài khoản mới (họ tên, email, mật khẩu đã mã hóa, vai trò mặc định là 'student') vào cơ sở dữ liệu.
9. Hệ thống trả về thông báo đăng ký thành công cho client.
10. Client hiển thị thông báo "Đăng ký thành công! Bây giờ bạn có thể đăng nhập." và chuyển hướng Người dùng đến trang Đăng nhập. Use case kết thúc thành công.

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng A: Dữ liệu nhập không hợp lệ

- Tại bước 4 của luồng chính, nếu Người dùng bỏ trống một trong các trường bắt buộc hoặc mật khẩu xác nhận không khớp.
- Hệ thống hiển thị một thông báo lỗi cụ thể ngay bên dưới trường nhập liệu có vấn đề (ví dụ: "Vui lòng nhập họ và tên", "Mật khẩu xác nhận không khớp").
- Use case quay trở lại bước 3 của luồng chính.

Luồng B: Email đã tồn tại

- Tại bước 6 của luồng chính, khi máy chủ phát hiện email đã được sử dụng bởi một tài khoản khác.
- Hệ thống hiển thị một thông báo lỗi: "Email này đã được sử dụng. Vui lòng chọn một email khác."
- Use case quay trở lại bước 3 của luồng chính.

3.2.3. Use case Đăng xuất

Tên Use Case		Đăng xuất khỏi hệ thống	Độ phức tạp: Thấp
Mô tả		Cho phép người dùng đang đăng nhập (Sinh viên, Giảng viên, Admin) kết thúc phiên làm việc hiện tại và thoát khỏi hệ thống một cách an toàn.	
Tác nhân		Người dùng (đã đăng nhập)	
Tiền điều kiện		Người dùng đã đăng nhập thành công vào hệ thống và có một phiên làm việc (token) hợp lệ đang hoạt động.	
Hậu điều kiện	Thành công	Phiên làm việc của người dùng bị hủy ở phía client (token bị xóa). Hệ thống chuyển hướng người dùng về trang Đăng nhập và giao diện được cập nhật sang trạng thái chưa đăng nhập.	
	Lỗi	(Hiếm khi xảy ra) Người dùng không đăng xuất được và vẫn ở lại trang hiện tại.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Người dùng đã đăng nhập nhấn vào nút "Đăng xuất" trên giao diện (thường nằm ở thanh điều hướng hoặc menu cá nhân). 2. Client (trình duyệt) xóa token xác thực (JWT) đã được lưu trữ (ví dụ: xóa khỏi localStorage). 3. Client ngay lập tức chuyển hướng người dùng về trang Đăng nhập. 4. Giao diện hệ thống được cập nhật về trạng thái mặc định cho khách (ví dụ: hiển thị lại nút "Đăng nhập", "Đăng ký"). Use case kết thúc thành công. 			
Luồng sự kiện phát sinh/Kịch bản phát sinh			

3.2.4. Use case Thay đổi mật khẩu cá nhân

Tên Use Case		Thay đổi mật khẩu cá nhân	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng đã đăng nhập (Sinh viên, Giảng viên, Admin) tự thay đổi mật khẩu tài khoản của mình để bảo vệ tài khoản.	
Tác nhân		Người dùng (đã đăng nhập)	
Tiền điều kiện		Người dùng đã đăng nhập thành công vào hệ thống và đang ở trang "Thay đổi mật khẩu".	
Hậu điều kiện	Thành công	Mật khẩu của người dùng trong cơ sở dữ liệu được cập nhật bằng mật khẩu mới. Hệ thống hiển thị thông báo thành công.	
	Lỗi	Mật khẩu không được thay đổi, hệ thống hiển thị thông báo lỗi và người dùng vẫn ở lại trang thay đổi mật khẩu.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Người dùng chọn chức năng "Đổi mật khẩu" trong trang cá nhân của mình. 2. Hệ thống hiển thị một biểu mẫu yêu cầu Người dùng nhập: "Mật khẩu hiện tại", "Mật khẩu mới", và "Xác nhận mật khẩu mới". 3. Người dùng điền đầy đủ thông tin và nhấn nút "Cập nhật". 4. Hệ thống kiểm tra tính hợp lệ của dữ liệu phía client (ví dụ: các trường không được trống, mật khẩu mới và mật khẩu xác nhận phải khớp, mật khẩu mới phải đủ độ phức tạp tối thiểu). Nếu không hợp lệ, chuyển đến Luồng A. 5. Hệ thống gửi toàn bộ dữ liệu (mật khẩu hiện tại, mật khẩu mới) đến máy chủ. 			

<p>6. Máy chủ kiểm tra xem "Mật khẩu hiện tại" mà người dùng nhập có khớp với mật khẩu đang được lưu trong cơ sở dữ liệu hay không. Nếu không khớp, chuyển đến Luồng B.</p> <p>7. Máy chủ tiến hành mã hóa "Mật khẩu mới".</p> <p>8. Máy chủ cập nhật mật khẩu mới đã được mã hóa vào cơ sở dữ liệu cho tài khoản của người dùng.</p> <p>9. Hệ thống trả về thông báo thành công cho client.</p> <p>10. Client hiển thị thông báo "Đổi mật khẩu thành công!" và xóa trống các trường trong biểu mẫu. Use case kết thúc thành công.</p>
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Dữ liệu nhập không hợp lệ
<ul style="list-style-type: none"> - Tại bước 4 của luồng chính, nếu một trong các trường bị bỏ trống, hoặc mật khẩu mới không khớp với mật khẩu xác nhận. - Hệ thống hiển thị thông báo lỗi cụ thể ngay bên dưới trường nhập liệu có vấn đề. - Use case quay trở lại bước 3 của luồng chính.
Luồng B: Mật khẩu hiện tại không chính xác
<ul style="list-style-type: none"> - Tại bước 6 của luồng chính, nếu mật khẩu hiện tại không đúng. - Hệ thống hiển thị một thông báo lỗi: "Mật khẩu hiện tại không chính xác." - Use case quay trở lại bước 3 của luồng chính.

3.2.5. Use case Tìm kiếm tài liệu đã được duyệt

Tên Use Case	Tìm kiếm tài liệu đã được duyệt	Độ phức tạp:
--------------	---------------------------------	--------------

		Trung bình
Mô tả	Cho phép người dùng đã đăng nhập (Sinh viên, Giảng viên, Admin) tìm kiếm và xem danh sách các tài liệu (bài tập lớn, đề án...) đã được phê duyệt và công khai trong kho tài liệu chung của hệ thống.	
Tác nhân	Người dùng (đã đăng nhập)	
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Hệ thống hiển thị danh sách các tài liệu phù hợp với tiêu chí tìm kiếm của người dùng.
	Lỗi	Hệ thống hiển thị thông báo không tìm thấy kết quả.
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/Kịch bản chính		
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Người dùng truy cập vào trang "Kho tài liệu" hoặc "Tìm kiếm". 2. Hệ thống hiển thị một giao diện tìm kiếm, bao gồm một ô nhập từ khóa và có thể có các bộ lọc (ví dụ: lọc theo Môn học). 3. Người dùng nhập từ khóa (ví dụ: tên tài liệu, tên tác giả) và/hoặc chọn Môn học từ danh sách, sau đó nhấn nút "Tìm kiếm". 4. Hệ thống gửi các tiêu chí tìm kiếm (từ khóa, môn học đã chọn) đến máy chủ. 5. Máy chủ thực hiện một truy vấn đến cơ sở dữ liệu để lấy danh sách các tài liệu thỏa mãn đồng thời các điều kiện: <ol style="list-style-type: none"> a. Có trạng thái là "đã duyệt" (approved). b. Tên tài liệu hoặc mô tả chứa từ khóa tìm kiếm. c. Thuộc môn học đã được chọn trong bộ lọc (nếu có). 		

<p>6. Máy chủ trả về một danh sách các tài liệu tìm thấy (kết quả có thể được phân trang). Nếu không có tài liệu nào phù hợp, chuyển đến Luồng A.</p> <p>7. Client nhận dữ liệu và hiển thị danh sách kết quả cho Người dùng. Mỗi kết quả bao gồm các thông tin cơ bản như: Tên tài liệu, Môn học, Ngày tải lên.</p> <p>8. Use case kết thúc thành công.</p>
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Không tìm thấy tài liệu phù hợp
<ul style="list-style-type: none"> - Tại bước 6 của luồng chính, nếu truy vấn không trả về bất kỳ tài liệu nào. - Hệ thống hiển thị một thông báo cho người dùng, ví dụ: "Không tìm thấy tài liệu nào phù hợp." - Use case kết thúc.

3.2.6. Use case Tải tài liệu về máy

Tên Use Case		Tải tài liệu về máy	Độ phức tạp: Thấp
Mô tả		Cho phép người dùng đã đăng nhập tải một tệp tài liệu (đã được duyệt) từ hệ thống về máy tính cá nhân để tham khảo.	
Tác nhân		Người dùng (đã đăng nhập)	
Tiền điều kiện		Người dùng đã tìm thấy một tài liệu cụ thể muốn tải. Người dùng có quyền truy cập tài liệu đó.	
Hậu điều kiện	Thành công	Tệp tài liệu được tải xuống và lưu trên thiết bị của người dùng.	
	Lỗi	Tệp không được tải xuống và hệ thống hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			

1. Use case bắt đầu khi Người dùng nhấn vào nút hoặc liên kết "Tải xuống" tương ứng với một tài liệu trong danh sách.
2. Trình duyệt gửi một yêu cầu đến máy chủ để truy cập vào đường dẫn tải tệp, kèm theo ID của tài liệu.
3. Máy chủ nhận yêu cầu, xác thực phiên làm việc của người dùng (đảm bảo người dùng đã đăng nhập).
4. Máy chủ truy xuất thông tin tài liệu từ cơ sở dữ liệu để lấy đường dẫn vật lý của tệp trên máy chủ. Nếu không tìm thấy thông tin hoặc tệp, chuyển đến **Luồng A**.
5. Máy chủ đọc tệp từ hệ thống lưu trữ và gửi nội dung tệp về cho trình duyệt, kèm theo header Content-Disposition: attachment để yêu cầu trình duyệt mở hộp thoại lưu tệp thay vì hiển thị trực tiếp.
6. Trình duyệt của người dùng hiển thị hộp thoại "Lưu tệp", cho phép người dùng chọn vị trí lưu.
7. Người dùng xác nhận và quá trình tải tệp bắt đầu. Use case kết thúc thành công.

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng A: Tệp không tồn tại hoặc đã bị xóa

- Tại bước 4 của luồng chính, nếu thông tin tài liệu không có trong cơ sở dữ liệu hoặc tệp vật lý không còn tồn tại trên máy chủ.
- Máy chủ trả về một mã lỗi (ví dụ: 404 Not Found).
- Client hiển thị một thông báo cho người dùng: "Lỗi: Không tìm thấy tệp hoặc tệp đã bị xóa."
- Use case kết thúc.

3.2.7. Use case Nộp bài tập hoặc tài liệu mới cho môn học

Tên Use Case		Nộp bài tập hoặc tài liệu mới cho môn học	Độ phức tạp: Trung bình
Mô tả		Cho phép Sinh viên tải lên (upload) một tệp tài liệu (bài tập lớn, đề án...) cho một môn học cụ thể. Tài liệu sau khi nộp sẽ có trạng thái "Chờ duyệt".	
Tác nhân		Sinh Viên	
Tiền điều kiện		Người dùng đã tìm thấy một tài liệu cụ thể muốn tải. Người dùng có quyền truy cập tài liệu đó.	
Hậu điều kiện	Thành công	Tệp tài liệu được tải lên máy chủ. Một bản ghi mới về bài nộp được tạo trong cơ sở dữ liệu với trạng thái "Chờ duyệt". Hệ thống hiển thị thông báo thành công.	
	Lỗi	Tệp không được tải lên, không có bản ghi nào được tạo trong CSDL. Hệ thống hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Sinh viên điều hướng đến trang "Nộp bài". 2. Hệ thống hiển thị một biểu mẫu (form) yêu cầu Sinh viên cung cấp các thông tin: <ol style="list-style-type: none"> a. Chọn Môn học (từ danh sách các môn học có trong hệ thống). b. Nhập "Tên tài liệu". c. Nhập "Mô tả" (tùy chọn). d. Chọn một tệp từ máy tính để tải lên. 3. Sinh viên điền đầy đủ thông tin, chọn tệp cần nộp và nhấn nút "Nộp bài". 			

4. Hệ thống kiểm tra tính hợp lệ của dữ liệu phía client (các trường bắt buộc đã được điền, tệp đã được chọn, kích thước và định dạng tệp hợp lệ). Nếu không hợp lệ, chuyển đến **Luồng A**.
5. Hệ thống tiến hành tải tệp lên máy chủ. Quá trình này có thể hiển thị thanh tiến trình. Nếu tải tệp thất bại, chuyển đến **Luồng B**.
6. Sau khi tệp được tải lên thành công, client gửi thông tin chi tiết về bài nộp (tên tài liệu, môn học, đường dẫn tệp trên server...) đến máy chủ.
7. Máy chủ tạo một bản ghi mới trong cơ sở dữ liệu, lưu lại thông tin bài nộp cùng với ID của sinh viên và gán trạng thái mặc định là "chờ duyệt" (pending).
8. Máy chủ trả về thông báo thành công.
9. Client hiển thị thông báo "Nộp bài thành công! Bài nộp của bạn đang chờ phê duyệt." và làm mới lại biểu mẫu. Use case kết thúc thành công.

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng A: Dữ liệu hoặc tệp không hợp lệ

- Tại bước 4 của luồng chính, nếu Sinh viên chưa điền đủ thông tin bắt buộc hoặc tệp tải lên không hợp lệ (ví dụ: kích thước quá lớn, sai định dạng cho phép).
- Hệ thống hiển thị một thông báo lỗi cụ thể (ví dụ: "Vui lòng chọn môn học", "Định dạng tệp không được hỗ trợ").
- Use case quay trở lại bước 3.

Luồng B: Lỗi tải tệp lên máy chủ

- Tại bước 5 của luồng chính, nếu có lỗi xảy ra khi đang tải tệp (ví dụ: mất kết nối mạng, lỗi server).
- Hệ thống hiển thị thông báo lỗi: "Đã có lỗi xảy ra khi tải tệp. Vui lòng thử lại."
- Use case quay trở lại bước 3.

3.2.8. Use case Xem lịch sử các bài đã nộp

Tên Use Case		Xem lịch sử các bài đã nộp	Độ phức tạp: Thấp
Mô tả		Cho phép Sinh viên xem lại danh sách tất cả các tài liệu mà mình đã nộp, cùng với trạng thái hiện tại của từng bài (ví dụ: Chờ duyệt, Đã duyệt, Bị từ chối).	
Tác nhân		Sinh Viên	
Tiền điều kiện		Sinh viên đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Hệ thống hiển thị đầy đủ danh sách các bài đã nộp của sinh viên đó.	
	Lỗi	(Hiếm khi xảy ra) Hệ thống không tải được danh sách và hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Sinh viên điều hướng đến trang "Lịch sử nộp bài" hoặc "Quản lý bài nộp" trong khu vực cá nhân. 2. Client tự động gửi một yêu cầu đến máy chủ để lấy danh sách các bài nộp của chính sinh viên đó. 3. Máy chủ nhận yêu cầu, xác thực sinh viên và thực hiện truy vấn cơ sở dữ liệu để lấy tất cả các bài nộp gắn với ID của sinh viên đang đăng nhập. 4. Máy chủ trả về một danh sách các bản ghi. Mỗi bản ghi chứa thông tin: Tên tài liệu, Môn học, Ngày nộp, và Trạng thái (pending, approved, rejected). Nếu sinh viên chưa từng nộp bài, chuyển đến Luồng A. 5. Client nhận dữ liệu và hiển thị cho sinh viên dưới dạng một bảng hoặc danh sách. 			

<p>6. Trạng thái của mỗi bài nộp được hiển thị rõ ràng bằng nhãn và màu sắc để dễ phân biệt. Nếu bài nộp bị từ chối, lý do từ chối (nếu có) cũng sẽ được hiển thị.</p> <p>7. Use case kết thúc thành công.</p>
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Chưa có bài nộp nào
<ul style="list-style-type: none"> - Tại bước 4 của luồng chính, nếu truy vấn không trả về kết quả nào. - Hệ thống hiển thị một thông báo ngay trên trang: "Bạn chưa nộp tài liệu nào." - Use case kết thúc.

3.2.9. Use case Xem trang thống kê cá nhân

Tên Use Case		Xem trang thống kê cá nhân	Độ phức tạp: Trung bình
Mô tả		Cho phép Sinh viên xem một trang tổng quan (dashboard) thống kê về hoạt động của chính mình, bao gồm tổng số bài đã nộp, số lượng bài theo từng trạng thái (đã duyệt, chờ duyệt, bị từ chối), và tỷ lệ trực quan.	
Tác nhân		Sinh Viên	
Tiền điều kiện		Sinh viên đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Hệ thống hiển thị trang thống kê với các số liệu và biểu đồ trực quan về hoạt động của sinh viên.	
	Lỗi	Hệ thống không thể tải được dữ liệu và hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			

1. Use case bắt đầu khi Sinh viên điều hướng đến trang "Thống kê cá nhân" hoặc "Dashboard".
2. Client tự động gửi một yêu cầu đến máy chủ để lấy dữ liệu thống kê của sinh viên đang đăng nhập.
3. Máy chủ nhận yêu cầu và thực hiện các truy vấn tổng hợp (aggregation) trên cơ sở dữ liệu để tính toán các số liệu sau (dựa trên ID của sinh viên):
 - a. Tổng số bài đã nộp.
 - b. Số lượng bài đang ở trạng thái "Chờ duyệt".
 - c. Số lượng bài đã được "Đã duyệt".
 - d. Số lượng bài đã "Bị từ chối".
4. Máy chủ xử lý dữ liệu và trả về một đối tượng (object) chứa các số liệu đã tính toán.
5. Client nhận dữ liệu và hiển thị lên giao diện một cách trực quan.
6. Các số liệu được trình bày dưới dạng:
 - a. Các thẻ (cards) hiển thị những con số quan trọng (ví dụ: "Tổng số bài nộp: 15").
 - b. Một biểu đồ (ví dụ: biểu đồ tròn) để minh họa tỷ lệ phần trăm của các bài nộp theo từng trạng thái.
7. Use case kết thúc thành công.

Luồng sự kiện phát sinh/Kịch bản phát sinh

3.2.10. Use case Xem danh sách bài nộp đang chờ duyệt

Tên Use Case		Xem danh sách bài nộp đang chờ duyệt	Độ phức tạp: Trung bình
Mô tả		Cho phép Giảng viên xem một danh sách các bài nộp của sinh viên, thuộc các môn học mà giảng viên đó phụ trách và đang ở trạng thái "Chờ duyệt", để thực hiện hành động tiếp theo (duyet hoặc từ chối).	
Tác nhân		Giảng Viên	
Tiền điều kiện		Giảng viên đã đăng nhập thành công vào hệ thống. Giảng viên đã được phân công phụ trách ít nhất một môn học.	
Hậu điều kiện	Thành công	Hệ thống hiển thị danh sách các bài nộp đang chờ duyệt thuộc các môn học mà giảng viên phụ trách.	
	Lỗi	Hệ thống không tải được danh sách và hiển thị lỗi, hoặc hiển thị danh sách trống nếu không có bài nộp nào.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Giảng viên truy cập vào trang "Dashboard" hoặc trang "Quản lý bài nộp". 2. Hệ thống mặc định hiển thị tab hoặc khu vực dành cho các bài nộp "Chờ duyệt". 3. Client tự động gửi yêu cầu đến máy chủ để lấy danh sách các bài nộp cần duyệt. 4. Máy chủ xác thực Giảng viên, sau đó lấy danh sách các ID môn học mà giảng viên này phụ trách. 5. Máy chủ thực hiện truy vấn cơ sở dữ liệu để tìm tất cả các bài nộp thỏa mãn đồng thời các điều kiện sau: <ol style="list-style-type: none"> a. Thuộc một trong các môn học mà giảng viên phụ trách. 			

<p>b. Có trạng thái là "chờ duyệt" (pending).</p> <p>6. Máy chủ trả về một danh sách các bài nộp tìm thấy. Mỗi bài nộp bao gồm các thông tin: Tên tài liệu, Tên sinh viên nộp, Tên môn học, Ngày nộp. Nếu không có bài nộp nào, chuyển đến Luồng A.</p> <p>7. Client nhận dữ liệu và hiển thị dưới dạng một bảng hoặc danh sách. Mỗi hàng trong bảng là một bài nộp đang chờ xử lý.</p> <p>8. Mỗi bài nộp trong danh sách sẽ có các nút hành động đi kèm như "Xem chi tiết", "Duyệt", "Từ chối".</p> <p>9. Use case kết thúc thành công.</p>
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Không có bài nộp nào chờ duyệt
<ul style="list-style-type: none"> - Tại bước 6 của luồng chính, nếu truy vấn không trả về kết quả nào. - Hệ thống hiển thị một thông báo rõ ràng: "Không có bài nộp nào đang chờ duyệt." - Use case kết thúc.

3.2.11. Use case Xem tất cả bài nộp trong các môn phụ trách.

Tên Use Case	Xem tất cả bài nộp trong các môn phụ trách	Độ phức tạp: Trung bình
Mô tả	Cho phép Giảng viên xem một danh sách tổng hợp toàn bộ các bài nộp (bao gồm tất cả các trạng thái) của sinh viên trong những môn học mà giảng viên đó được phân công phụ trách.	
Tác nhân	Giảng Viên	
Tiền điều kiện	Giảng viên đã đăng nhập thành công vào hệ thống.	

Hậu điều kiện	Thành công	Hệ thống hiển thị một danh sách đầy đủ (có thể được phân trang) các bài nộp thuộc môn học của giảng viên.
	Lỗi	Hệ thống không tải được danh sách và hiển thị thông báo lỗi.
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/Kịch bản chính		
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Giảng viên truy cập trang "Quản lý bài nộp" và chọn xem tab "Tất cả bài nộp". 2. Client gửi yêu cầu đến máy chủ để lấy toàn bộ bài nộp thuộc các môn học do giảng viên phụ trách. 3. Máy chủ xác thực Giảng viên và lấy danh sách các môn học mà giảng viên này quản lý. 4. Máy chủ thực hiện truy vấn cơ sở dữ liệu để tìm tất cả các bài nộp thuộc các môn học đó, không phân biệt trạng thái (pending, approved, rejected). 5. Máy chủ trả về một danh sách các bài nộp tìm thấy, sắp xếp theo ngày nộp gần nhất và có phân trang để tối ưu hiệu suất. Nếu không có bài nộp nào, chuyển đến Luồng A. 6. Client nhận dữ liệu và hiển thị dưới dạng một bảng. 7. Mỗi hàng của bảng hiển thị thông tin chi tiết về một bài nộp: Tên tài liệu, Sinh viên nộp, Môn học, Ngày nộp, và Trạng thái (được biểu thị bằng nhãn và màu sắc). 8. Use case kết thúc thành công. 		
Luồng sự kiện phát sinh/Kịch bản phát sinh		
Luồng A: Không có bài nộp nào chờ duyệt		
<ul style="list-style-type: none"> - Tại bước 5 của luồng chính, nếu truy vấn không tìm thấy bất kỳ bài nộp nào trong các môn học giảng viên phụ trách. - Hệ thống hiển thị một thông báo: "Chưa có bài nộp nào trong các môn học của bạn." - Use case kết thúc. 		

3.2.12. Use case **Duyệt bài nộp của sinh viên.**

Tên Use Case		Duyệt bài nộp của sinh viên	Độ phức tạp: Thấp
Mô tả		Cho phép Giảng viên thay đổi trạng thái của một bài nộp từ "Chờ duyệt" sang "Đã duyệt". Sau khi được duyệt, bài nộp sẽ trở nên công khai và có thể được tìm thấy trong kho tài liệu chung.	
Tác nhân		Giảng Viên	
Tiền điều kiện		Giảng viên đã đăng nhập và đang xem danh sách các bài nộp đang chờ duyệt.	
Hậu điều kiện	Thành công	Trạng thái của bài nộp trong CSDL được cập nhật thành "đã duyệt" (approved). Một thông báo được tạo và gửi đến sinh viên. Bài nộp được đưa vào kho tài liệu chung.	
	Lỗi	Trạng thái của bài nộp không thay đổi. Hệ thống hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Giảng viên nhấn nút "Duyệt" trên một bài nộp cụ thể trong danh sách đang chờ. 2. Hệ thống hiển thị một hộp thoại xác nhận, hỏi "Bạn có chắc chắn muốn duyệt bài nộp này không?". 3. Giảng viên chọn "Đồng ý". Nếu chọn "Hủy", use case kết thúc. 4. Client gửi yêu cầu "Duyệt" đến máy chủ, kèm theo ID của bài nộp. 5. Máy chủ xác thực quyền của Giảng viên (đảm bảo giảng viên có quyền duyệt bài nộp thuộc môn học này). Nếu không có quyền, chuyển đến Luồng A. 			

<p>6. Máy chủ cập nhật trạng thái của bài nộp trong cơ sở dữ liệu từ pending thành approved.</p> <p>7. Hệ thống tự động tạo một bản ghi thông báo mới trong cơ sở dữ liệu cho sinh viên đã nộp bài, với nội dung ví dụ: "Bài nộp '[Tên bài nộp]' của bạn đã được duyệt."</p> <p>8. Máy chủ trả về thông báo duyệt thành công cho client của Giảng viên.</p> <p>9. Giao diện của Giảng viên cập nhật lại, xóa bài nộp đó khỏi danh sách "Chờ duyệt" và hiển thị một thông báo ngắn "Duyệt thành công". Use case kết thúc.</p>
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Không có quyền thực hiện hành động
<ul style="list-style-type: none"> - Tại bước 5 của luồng chính, nếu hệ thống phát hiện Giảng viên không phụ trách môn học của bài nộp này. - Máy chủ từ chối yêu cầu và trả về một mã lỗi (ví dụ: 403 Forbidden). - Client hiển thị thông báo lỗi: "Bạn không có quyền duyệt bài nộp này." - Use case kết thúc.

3.2.13. Use case Từ chối bài nộp của sinh viên.

Tên Use Case	Từ chối bài nộp của sinh viên	Độ phức tạp: Trung bình
Mô tả	Cho phép Giảng viên thay đổi trạng thái của một bài nộp từ "Chờ duyệt" sang "Bị từ chối" và cung cấp lý do hoặc phản hồi cụ thể cho sinh viên.	
Tác nhân	Giảng Viên	
Tiền điều kiện	Giảng viên đã đăng nhập và đang xem danh sách các bài nộp đang chờ duyệt.	

Hậu điều kiện	Thành công	Trạng thái của bài nộp trong CSDL được cập nhật thành "bị từ chối" (rejected) và lý do từ chối được lưu lại. Một thông báo (kèm lý do) được tạo và gửi đến sinh viên.
	Lỗi	Trạng thái của bài nộp không thay đổi. Hệ thống hiển thị thông báo lỗi.
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/Kịch bản chính		
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Giảng viên nhấn nút "Từ chối" trên một bài nộp cụ thể trong danh sách đang chờ. 2. Hệ thống hiển thị một hộp thoại (modal) yêu cầu Giảng viên nhập "Lý do từ chối" vào một vùng văn bản (textarea). 3. Giảng viên nhập lý do và nhấn nút "Xác nhận từ chối". Nếu Giảng viên không nhập lý do (và lý do là bắt buộc), chuyển đến Luồng A. 4. Client gửi yêu cầu "Từ chối" đến máy chủ, kèm theo ID của bài nộp và nội dung lý do. 5. Máy chủ xác thực quyền của Giảng viên đối với bài nộp này. Nếu không có quyền, chuyển đến Luồng B. 6. Máy chủ cập nhật trạng thái của bài nộp trong cơ sở dữ liệu từ pending thành rejected và lưu lại nội dung "Lý do từ chối". 7. Hệ thống tự động tạo một bản ghi thông báo mới cho sinh viên, với nội dung ví dụ: "Bài nộp '[Tên bài nộp]' của bạn đã bị từ chối. Lý do: [Nội dung lý do]". 8. Máy chủ trả về thông báo từ chối thành công cho client của Giảng viên. 9. Giao diện của Giảng viên cập nhật lại, xóa bài nộp đó khỏi danh sách "Chờ duyệt" và hiển thị một thông báo ngắn "Đã từ chối bài nộp". Use case kết thúc thành công. 		
Luồng sự kiện phát sinh/Kịch bản phát sinh		
Luồng A: Không có quyền thực hiện hành động		

<ul style="list-style-type: none"> - Tại bước 3 của luồng chính, nếu Giảng viên nhấn "Xác nhận từ chối" nhưng không điền lý do (trường hợp lý do là bắt buộc). - Hệ thống hiển thị một thông báo lỗi ngay trên hộp thoại: "Vui lòng nhập lý do từ chối." - Use case quay trở lại bước 3.
Luồng B: Không có quyền thực hiện hành động
<ul style="list-style-type: none"> - Tại bước 5 của luồng chính, nếu hệ thống phát hiện Giảng viên không phụ trách môn học của bài nộp này. - Máy chủ từ chối yêu cầu và trả về một mã lỗi (ví dụ: 403 Forbidden). - Client hiển thị thông báo lỗi: "Bạn không có quyền từ chối bài nộp này." - Use case kết thúc.

3.2.14. Use case Xem trang thống kê giảng viên

Tên Use Case	Xem trang thống kê giảng viên	Độ phức tạp: Cao
Mô tả	Cho phép Giảng viên xem một trang tổng quan (dashboard) với các số liệu và biểu đồ thống kê chi tiết về các bài nộp trong những môn học mình phụ trách. Giảng viên có thể lọc dữ liệu theo từng môn học và theo các khoảng thời gian khác nhau.	
Tác nhân	Giảng Viên	
Tiền điều kiện	Giảng viên đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Hệ thống hiển thị trang thống kê với các số liệu và biểu đồ trực quan, phản ánh chính xác dữ liệu theo các bộ lọc được áp dụng.

	Lỗi	Hệ thống không thể tải và xử lý dữ liệu, và sẽ hiển thị thông báo lỗi.
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/Kịch bản chính		
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Giảng viên điều hướng đến trang "Thống kê". 2. Hệ thống hiển thị giao diện thống kê, bao gồm các bộ lọc chính: <ol style="list-style-type: none"> a. Một danh sách thả xuống (dropdown) để chọn Môn học (mặc định là "Tất cả các môn phụ trách"). b. Một bộ chọn khoảng thời gian (date range picker), ví dụ: "7 ngày qua", "Tháng này", "Năm nay", hoặc một khoảng tùy chỉnh. 3. Ban đầu, hệ thống tự động tải dữ liệu thống kê tổng quan cho tất cả các môn giảng viên phụ trách trong một khoảng thời gian mặc định (ví dụ: "30 ngày qua"). 4. Client gửi yêu cầu lấy dữ liệu đến máy chủ với các bộ lọc mặc định. 5. Máy chủ thực hiện các truy vấn tổng hợp (aggregation) phức tạp để tính toán các số liệu dựa trên ID của giảng viên và các bộ lọc: <ol style="list-style-type: none"> a. Tổng số bài nộp. b. Phân bố bài nộp theo từng trạng thái (chờ duyệt, đã duyệt, bị từ chối). c. Phân bố bài nộp theo từng môn học. d. Xu hướng nộp bài theo thời gian (ví dụ: số lượng bài nộp mỗi ngày). 6. Máy chủ trả về một đối tượng JSON chứa tất cả các dữ liệu đã được tính toán và cấu trúc lại. 7. Client nhận dữ liệu và hiển thị lên giao diện một cách trực quan, bao gồm: <ol style="list-style-type: none"> a. Các thẻ (cards) hiển thị các số liệu tổng quan chính. 		

<p>b. Một biểu đồ tròn thể hiện tỷ lệ bài nộp theo trạng thái.</p> <p>c. Một biểu đồ cột so sánh số lượng bài nộp giữa các môn học.</p> <p>d. Một biểu đồ đường thể hiện xu hướng nộp bài theo thời gian.</p> <p>8. Use case kết thúc thành công.</p>
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Giảng viên thay đổi bộ lọc
<ul style="list-style-type: none"> - Tại một thời điểm bất kỳ, Giảng viên thay đổi giá trị của bộ lọc (ví dụ: chọn một môn học cụ thể hoặc một khoảng thời gian khác). - Hệ thống quay trở lại bước 4 của luồng chính, gửi yêu cầu mới đến máy chủ với các tham số lọc đã được cập nhật. - Giao diện được vẽ lại với dữ liệu thống kê mới.

3.2.15. Use case Quản lý tài khoản người dùng

Tên Use Case	Quản lý tài khoản người dùng		Độ phức tạp: Cao
Mô tả	Cho phép Quản trị viên (Admin) xem, tìm kiếm, lọc, chỉnh sửa thông tin (bao gồm cả việc thăng/giáng cấp vai trò), và xóa tài khoản người dùng khỏi hệ thống.		
Tác nhân	Admin		
Tiền điều kiện	Admin đã đăng nhập thành công vào hệ thống với quyền quản trị cao nhất.		
Hậu điều kiện	Thành công	Các thay đổi (chỉnh sửa, xóa) được áp dụng thành công trong cơ sở dữ liệu và giao diện được cập nhật để phản ánh thay đổi.	
	Lỗi	Thao tác không thành công, dữ liệu không thay đổi và hệ thống hiển thị thông báo lỗi.	

ĐẶC TẢ CHỨC NĂNG

Luồng sự kiện chính/Kịch bản chính

1. Use case bắt đầu khi Admin truy cập vào trang "Quản lý Người dùng".
2. Hệ thống hiển thị giao diện quản lý, bao gồm:
 - a. Một ô tìm kiếm (theo tên hoặc email).
 - b. Một bộ lọc theo vai trò (Tất cả, Admin, Giảng viên, Sinh viên).
 - c. Một bảng danh sách người dùng (được phân trang).
3. Client tự động gửi yêu cầu đến máy chủ để lấy trang đầu tiên của danh sách người dùng.
4. Máy chủ truy vấn CSDL để lấy danh sách người dùng (có phân trang), bao gồm các thông tin: Họ tên, Email, Vai trò, Trạng thái, Ngày tham gia.
5. Máy chủ trả về danh sách. Client nhận và hiển thị dữ liệu trong bảng. Mỗi hàng có các nút hành động "Sửa" và "Xóa".
6. **(Phân nhánh)** Admin có thể thực hiện các hành động **Chỉnh sửa (Luồng B)** hoặc **Xóa (Luồng C)**.
7. **(Phân nhánh)** Admin có thể nhập từ khóa tìm kiếm hoặc thay đổi bộ lọc. Khi đó, hệ thống lặp lại từ bước 3 với các tham số mới.

Luồng B: Chỉnh sửa thông tin và vai trò người dùng

1. Từ danh sách người dùng, Admin nhấn nút "Sửa" trên một tài khoản cụ thể.
2. Hệ thống hiển thị một hộp thoại (modal) chứa biểu mẫu với thông tin hiện tại của người dùng đó. Các trường có thể sửa bao gồm: Họ tên, Vai trò (Role).
3. Admin thay đổi thông tin cần thiết. Ví dụ: thay đổi vai trò của một "Sinh viên" thành "Giảng viên".
4. Admin nhấn nút "Lưu thay đổi".

5. Client gửi yêu cầu cập nhật đến máy chủ, kèm theo ID người dùng và các thông tin đã thay đổi.
6. Máy chủ xác thực và cập nhật thông tin người dùng trong CSDL.
7. Máy chủ trả về thông báo thành công. Giao diện cập nhật lại thông tin của người dùng vừa sửa trong bảng và hiển thị thông báo "Cập nhật thành công". Luồng B kết thúc.

Luồng C: Xóa tài khoản người dùng

1. Từ danh sách người dùng, Admin nhấn nút "Xóa" trên một tài khoản cụ thể.
2. Hệ thống hiển thị một hộp thoại xác nhận: "Bạn có chắc chắn muốn xóa tài khoản [Tên người dùng]? Hành động này không thể hoàn tác."
3. Admin chọn "Đồng ý".
4. Client gửi yêu cầu xóa đến máy chủ, kèm theo ID của người dùng.
5. Máy chủ kiểm tra xem Admin có đang cố gắng xóa tài khoản của chính mình không. Nếu có, chuyển đến **Luồng C1**.
6. Máy chủ thực hiện xóa tài khoản khỏi CSDL (có thể là xóa mềm - soft delete).
7. Máy chủ trả về thông báo thành công. Giao diện xóa người dùng đó khỏi bảng và hiển thị thông báo "Đã xóa tài khoản thành công". Luồng C kết thúc.

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng C1: Admin tự xóa tài khoản của mình

- Tại bước 5 của Luồng C, nếu ID người dùng cần xóa trùng với ID của Admin đang đăng nhập.
- Máy chủ từ chối yêu cầu và trả về lỗi.
- Client hiển thị thông báo lỗi: "Bạn không thể xóa tài khoản của chính mình."
- Luồng C kết thúc.

3.2.16. Use case Quản lý danh mục môn học

Tên Use Case		Quản lý danh mục môn học	Độ phức tạp: Trung bình
Mô tả		Cho phép Quản trị viên (Admin) thực hiện đầy đủ các thao tác quản lý (CRUD) đối với danh mục môn học của toàn hệ thống.	
Tác nhân		Admin	
Tiền điều kiện		Admin đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Các thay đổi (thêm, sửa, xóa) đối với danh mục môn học được áp dụng thành công trong CSDL và giao diện được cập nhật.	
	Lỗi	Thao tác không thành công, dữ liệu không thay đổi và hệ thống hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none"> 1. Use case bắt đầu khi Admin truy cập vào trang "Quản lý Môn học". 2. Hệ thống hiển thị giao diện quản lý, bao gồm: <ol style="list-style-type: none"> a. Một biểu mẫu để "Thêm môn học mới" với các trường: Mã môn học, Tên môn học. b. Một bảng danh sách các môn học hiện có. 3. Client gửi yêu cầu đến máy chủ để lấy danh sách môn học. 4. Máy chủ truy vấn và trả về danh sách các môn học hiện có. Client hiển thị dữ liệu trong bảng, mỗi hàng có nút "Sửa" và "Xóa". 5. (Phân nhánh - Thêm mới) Admin điền thông tin vào biểu mẫu "Thêm môn học mới" và nhấn nút "Thêm". 6. Client gửi yêu cầu tạo môn học mới đến máy chủ. 			

7. Máy chủ kiểm tra xem mã hoặc tên môn học đã tồn tại chưa. Nếu có, chuyển đến **Luồng A**.
8. Máy chủ tạo bản ghi môn học mới trong CSDL và trả về thông báo thành công.
9. Giao diện được cập nhật, thêm môn học mới vào danh sách và hiển thị thông báo "Thêm môn học thành công".

Luồng B: Chỉnh sửa môn học

1. Từ danh sách, Admin nhấn nút "Sửa" trên một môn học.
2. Hệ thống hiển thị một hộp thoại (modal) với thông tin hiện tại của môn học đó.
3. Admin thay đổi Tên môn học hoặc Mã môn học và nhấn "Lưu thay đổi".
4. Client gửi yêu cầu cập nhật đến máy chủ. Máy chủ kiểm tra tính duy nhất của thông tin mới. Nếu trùng, chuyển đến **Luồng A**.
5. Máy chủ cập nhật thông tin môn học trong CSDL và trả về thông báo thành công.
6. Giao diện cập nhật lại thông tin của môn học trong bảng và hiển thị thông báo "Cập nhật thành công". Luồng B kết thúc.

Luồng C: Xóa môn học

1. Từ danh sách, Admin nhấn nút "Xóa" trên một môn học.
2. Client gửi yêu cầu kiểm tra ràng buộc đến máy chủ. Máy chủ kiểm tra xem môn học này đã có tài liệu hoặc đã được gán cho giảng viên nào chưa. Nếu có, chuyển đến **Luồng C1**.
3. Nếu không có ràng buộc, hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn xóa môn học [Tên môn học]?".
4. Admin chọn "Đồng ý". Client gửi yêu cầu xóa đến máy chủ.
5. Máy chủ thực hiện xóa môn học khỏi CSDL và trả về thông báo thành công.

6. Giao diện xóa môn học khỏi bảng và hiển thị thông báo "Đã xóa môn học thành công". Luồng C kết thúc.
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Trùng mã hoặc tên môn học
<ul style="list-style-type: none"> - Tại bước 7 (Thêm) hoặc bước 4 (Sửa), nếu mã hoặc tên môn học được cung cấp đã tồn tại trong hệ thống. - Máy chủ trả về lỗi. Client hiển thị thông báo: "Mã hoặc tên môn học đã tồn tại." - Use case quay lại bước trước đó.
Luồng C1: Không thể xóa môn học do có dữ liệu liên quan
<ul style="list-style-type: none"> - Tại bước 2 của Luồng C, nếu môn học đã có tài liệu được nộp hoặc đã được gán cho giảng viên. - Hệ thống từ chối cho phép xóa và hiển thị thông báo: "Không thể xóa môn học này vì đã có dữ liệu liên quan. Vui lòng gỡ bỏ các ràng buộc trước." - Luồng C kết thúc.

3.2.17. Use case Gán giảng viên vào môn học

Tên Use Case	Gán giảng viên vào môn học	Độ phức tạp: Trung bình
Mô tả	Cho phép Quản trị viên (Admin) thiết lập mối quan hệ giữa Giảng viên và Môn học, chỉ định giảng viên nào sẽ phụ trách môn học nào.	
Tác nhân	Admin	
Tiền điều kiện	Admin đã đăng nhập thành công. Hệ thống đã có sẵn dữ liệu về các tài khoản Giảng viên và các Môn học.	

Hậu điều kiện	Thành công	Mỗi quan hệ phân công giữa giảng viên và môn học được lưu lại trong cơ sở dữ liệu. Giảng viên được gán sẽ có quyền quản lý các bài nộp thuộc môn học đó.
	Lỗi	Thao tác gán không thành công, dữ liệu không thay đổi và hệ thống hiển thị thông báo lỗi.
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/Kịch bản chính		
<ol style="list-style-type: none"> 1. Use case có thể bắt đầu theo hai hướng: <ol style="list-style-type: none"> a. Hướng 1: Từ trang "Quản lý Môn học", Admin chọn một môn học và nhấn vào tùy chọn "Phân công giảng viên". b. Hướng 2: Từ trang "Quản lý Người dùng", Admin chọn một tài khoản Giảng viên và nhấn vào tùy chọn "Phân công môn học". 2. Hệ thống hiển thị một giao diện phân công. Ví dụ, nếu theo Hướng 1, giao diện sẽ hiển thị tên môn học và một danh sách tất cả các Giảng viên trong hệ thống. 3. Mỗi giảng viên trong danh sách có một ô đánh dấu (checkbox). Các giảng viên hiện đang phụ trách môn học này sẽ được đánh dấu sẵn. 4. Admin tiến hành chọn (tick) hoặc bỏ chọn (untick) các giảng viên để gán hoặc hủy gán khỏi môn học. 5. Sau khi hoàn tất lựa chọn, Admin nhấn nút "Lưu thay đổi". 6. Client gửi một yêu cầu đến máy chủ, bao gồm ID của môn học và một danh sách (array) các ID của tất cả giảng viên đã được chọn. 7. Máy chủ nhận yêu cầu. Dựa trên ID môn học, máy chủ sẽ xóa tất cả các phân công cũ liên quan đến môn học đó và tạo ra các bản ghi phân công mới dựa trên danh sách ID giảng viên được gửi lên. 8. Máy chủ trả về thông báo thành công. 9. Client hiển thị thông báo "Cập nhật phân công thành công" và đóng giao diện phân công. Use case kết thúc thành công. 		

Luồng sự kiện phát sinh/Kịch bản phát sinh

3.2.18. Use case Quản lý tài khoản giảng viên đang chờ phê duyệt

Tên Use Case		Quản lý tài khoản giảng viên đang chờ phê duyệt	Độ phức tạp: Trung bình
Mô tả		Cho phép Quản trị viên (Admin) xem danh sách các tài khoản có vai trò "Giảng viên" mới được tạo và đang ở trạng thái "Chờ phê duyệt", sau đó ra quyết định chấp thuận (kích hoạt) hoặc từ chối tài khoản đó.	
Tác nhân		Admin	
Tiền điều kiện		Admin đã đăng nhập thành công. Có ít nhất một tài khoản Giảng viên được tạo nhưng chưa được kích hoạt (đang ở trạng thái chờ).	
Hậu điều kiện	Thành công	Trạng thái tài khoản Giảng viên được cập nhật thành "hoạt động" (active), cho phép họ đăng nhập. Từ chối: Yêu cầu về tài khoản bị xóa hoặc chuyển sang trạng thái "đã từ chối".	
	Lỗi	Trạng thái tài khoản không thay đổi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<ol style="list-style-type: none">1. Use case bắt đầu khi Admin truy cập trang "Quản lý Người dùng" và chọn tab/bộ lọc "Chờ phê duyệt".2. Hệ thống hiển thị một danh sách các tài khoản Giảng viên đang có trạng thái là "chờ phê duyệt" (pending_approval).3. Mỗi hàng trong danh sách hiển thị thông tin cơ bản (Họ tên, Email, Ngày tạo) và hai nút hành động: "Chấp thuận" và "Từ chối".4. Nếu không có tài khoản nào đang chờ, hệ thống sẽ hiển thị thông báo "Không có tài khoản nào đang chờ phê duyệt."			

5. **(Phân nhánh)** Admin có thể thực hiện hành động **Chấp thuận (Luồng B)** hoặc **Từ chối (Luồng C)**.

Luồng B: Chấp thuận tài khoản

1. Từ danh sách chờ, Admin nhấn nút "Chấp thuận" trên một tài khoản cụ thể.
2. Hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn kích hoạt tài khoản cho [Tên giảng viên]?".
3. Admin chọn "Đồng ý".
4. Client gửi yêu cầu "Chấp thuận" đến máy chủ, kèm theo ID của tài khoản.
5. Máy chủ cập nhật trạng thái của tài khoản trong CSDL từ pending_approval thành active.
6. (Tùy chọn) Hệ thống gửi một email thông báo đến giảng viên rằng tài khoản của họ đã được kích hoạt.
7. Máy chủ trả về thông báo thành công. Giao diện xóa tài khoản khỏi danh sách "Chờ phê duyệt" và hiển thị thông báo "Tài khoản đã được kích hoạt thành công". Luồng B kết thúc.

Luồng C: Từ chối tài khoản

1. Từ danh sách chờ, Admin nhấn nút "Từ chối" trên một tài khoản.
2. Hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn từ chối yêu cầu tạo tài khoản này?".
3. Admin chọn "Đồng ý".
4. Client gửi yêu cầu "Từ chối" đến máy chủ, kèm theo ID của tài khoản.
5. Máy chủ xóa bản ghi tài khoản đang chờ duyệt khỏi CSDL (hoặc cập nhật trạng thái thành rejected).
6. (Tùy chọn) Hệ thống gửi một email thông báo cho người đăng ký rằng yêu cầu của họ đã bị từ chối.

7. Máy chủ trả về thông báo thành công. Giao diện xóa tài khoản khỏi danh sách "Chờ phê duyệt" và hiển thị thông báo "Đã từ chối yêu cầu tạo tài khoản". Luồng C kết thúc.

3.2.19. Use case Xem thống kê tổng quan toàn hệ thống

Tên Use Case		Xem thống kê tổng quan toàn hệ thống	Độ phức tạp: Cao
Mô tả		Cho phép Quản trị viên (Admin) xem một trang tổng quan (dashboard) với các số liệu, chỉ số và biểu đồ thống kê cốt lõi về toàn bộ hoạt động của hệ thống.	
Tác nhân		Admin	
Tiền điều kiện		Admin đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Hệ thống hiển thị trang dashboard với các dữ liệu thống kê tổng quan chính xác và trực quan.	
	Lỗi	Hệ thống không thể tải và xử lý dữ liệu, và sẽ hiển thị thông báo lỗi.	
ĐẶC TẢ CHỨC NĂNG			

Luồng sự kiện chính/Kịch bản chính

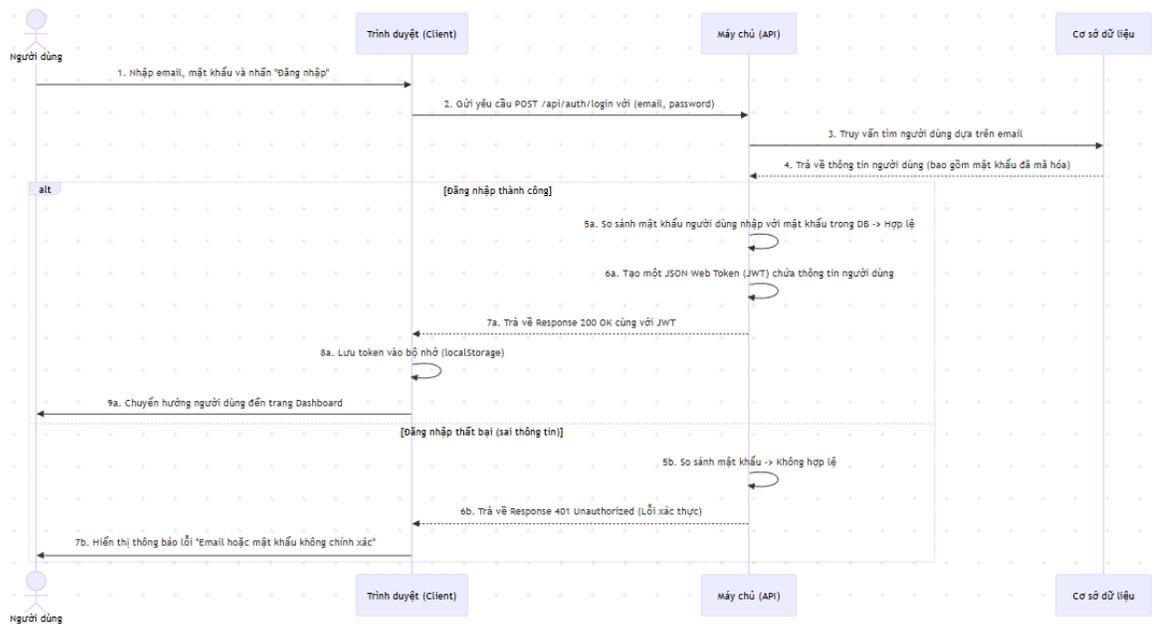
1. Use case bắt đầu khi Admin truy cập vào trang "Dashboard" hoặc "Tổng quan hệ thống".
2. Client tự động gửi một yêu cầu đến máy chủ để lấy dữ liệu thống kê toàn hệ thống.
3. Máy chủ thực hiện nhiều truy vấn tổng hợp (aggregation) trên các bảng khác nhau của cơ sở dữ liệu để tính toán các chỉ số hiệu suất chính (KPIs):
 - a. Về người dùng:
 - + Tổng số người dùng.
 - + Phân bố số lượng người dùng theo từng vai trò (Admin, Giảng viên, Sinh viên).
 - + Số lượng tài khoản mới trong tháng.
 - b. Về tài liệu:
 - + Tổng số bài nộp trong hệ thống.
 - + Phân bố số lượng bài nộp theo từng trạng thái (Chờ duyệt, Đã duyệt, Bị từ chối).
 - c. Về môn học:
 - + Tổng số môn học đang có.
 - d. Về hệ thống:
 - + Tổng dung lượng lưu trữ mà các tệp tải lên đang chiếm dụng.
4. Máy chủ tập hợp tất cả dữ liệu đã tính toán vào một đối tượng JSON và trả về cho client.
5. Client nhận dữ liệu và hiển thị lên giao diện một cách trực quan, thường bao gồm:

- a. Một dãy các thẻ (Cards) hiển thị các KPI quan trọng nhất (ví dụ: Tổng người dùng, Tổng tài liệu).
- b. Một biểu đồ cột hoặc tròn để thể hiện sự phân bố người dùng theo vai trò.
- c. Một biểu đồ tròn khác để thể hiện tỷ lệ các bài nộp theo trạng thái.
- d. Một biểu đồ đường (line chart) có thể hiển thị xu hướng tăng trưởng người dùng hoặc số lượng bài nộp theo thời gian.
- e. Use case kết thúc thành công.

Luồng sự kiện phát sinh/Kịch bản phát sinh

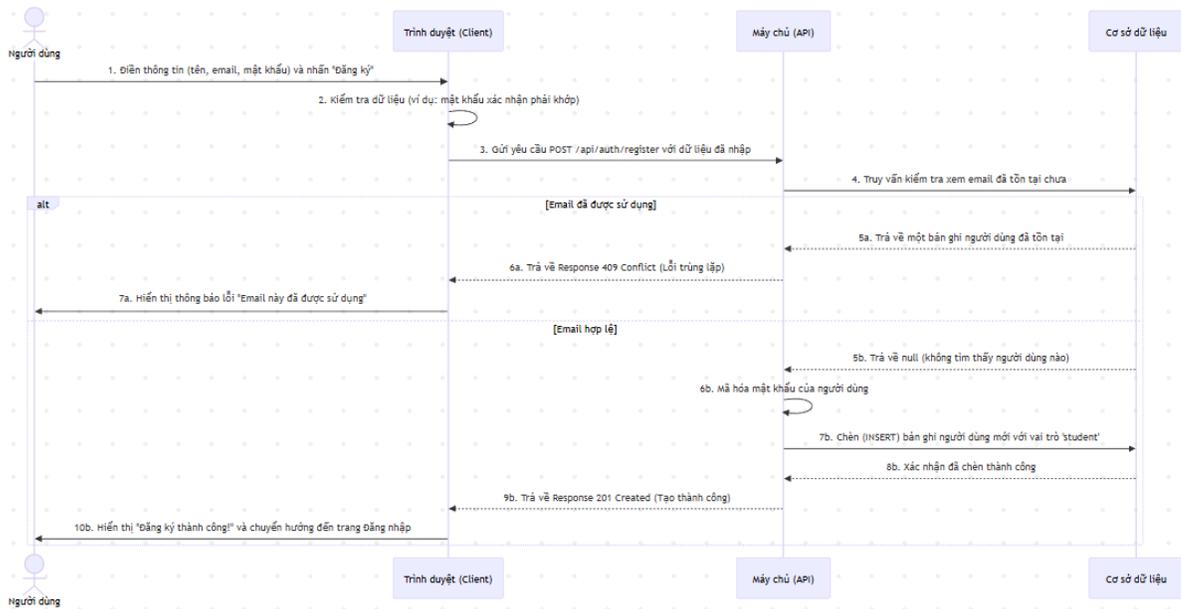
3.3. Biểu đồ tuần tự

3.3.1. Biểu đồ tuần tự usecase đăng nhập



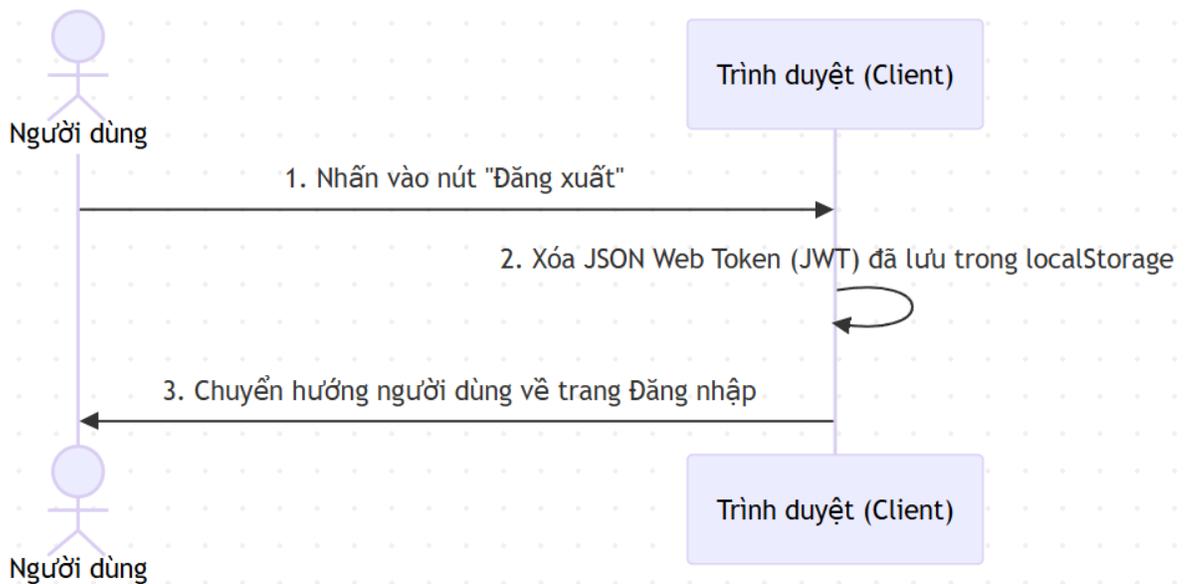
Ảnh 0.18. Biểu đồ tuần tự chức năng Đăng nhập

3.3.2. Biểu đồ tuần tự usecase đăng ký của sinh viên



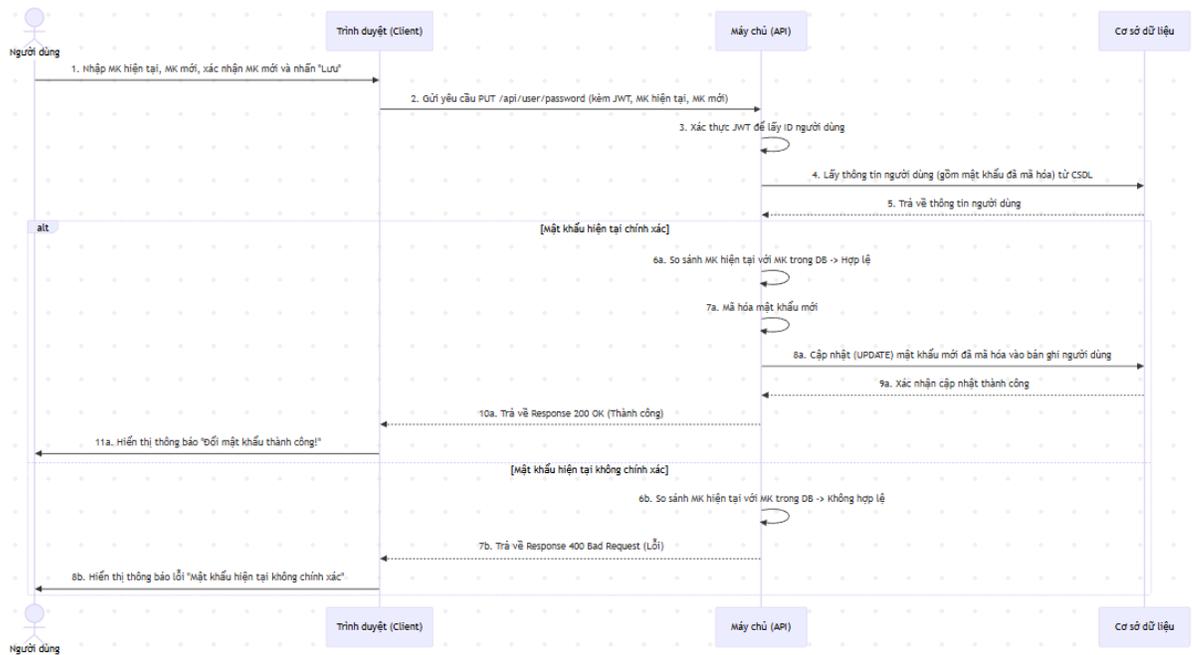
Ảnh 0.19. Biểu đồ tuần tự chức năng Đăng ký của sinh viên

3.3.3. Biểu đồ tuần tự usecase đăng xuất



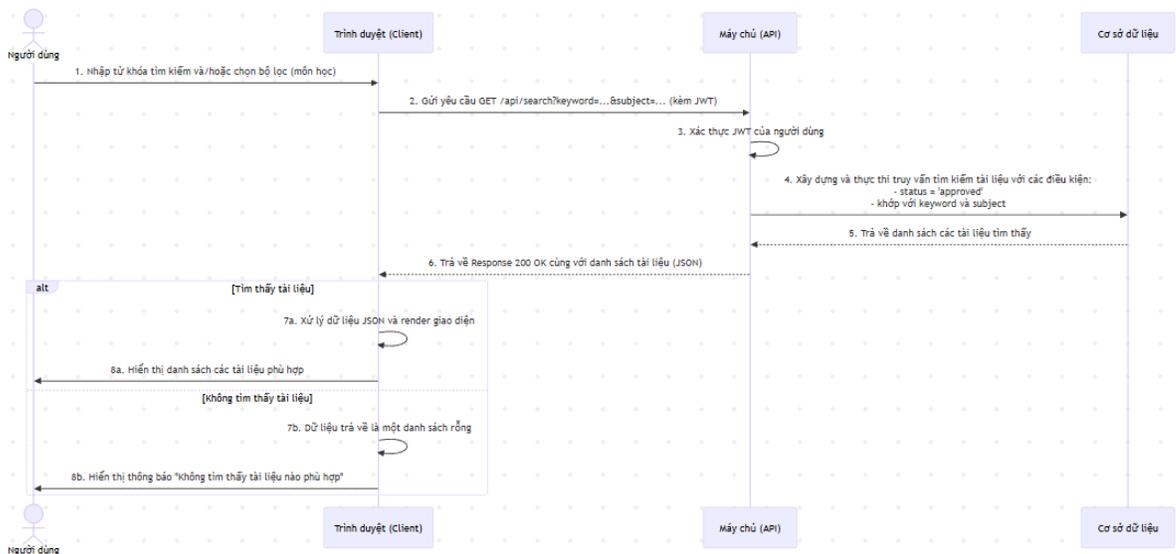
Ảnh 0.20. Biểu đồ tuần tự chức năng Đăng xuất

3.3.4. Biểu đồ tuần tự usecase thay đổi mật khẩu



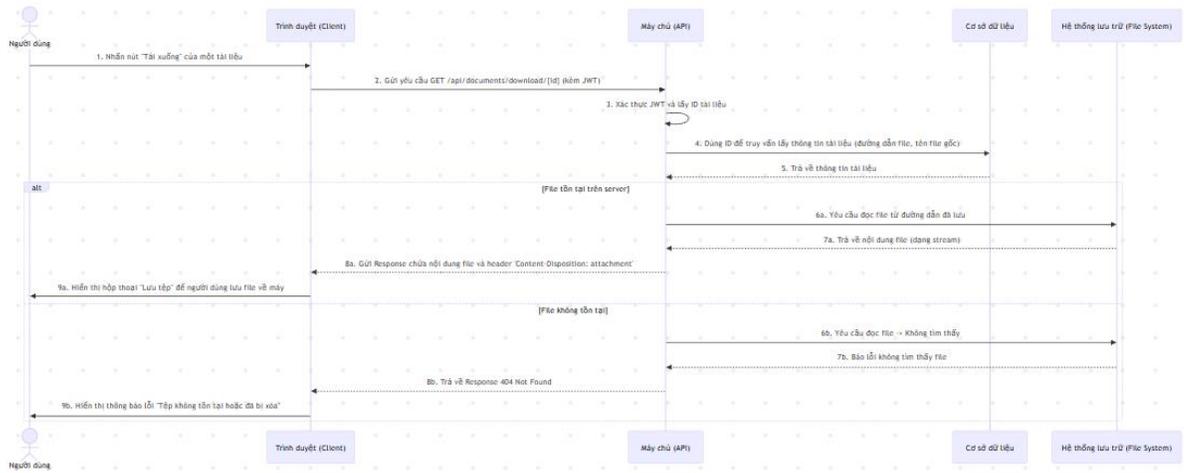
Ảnh 0.21. Biểu đồ tuần tự chức năng Đổi mật khẩu

3.3.5. Biểu đồ tuần tự usecase tìm kiếm tài liệu



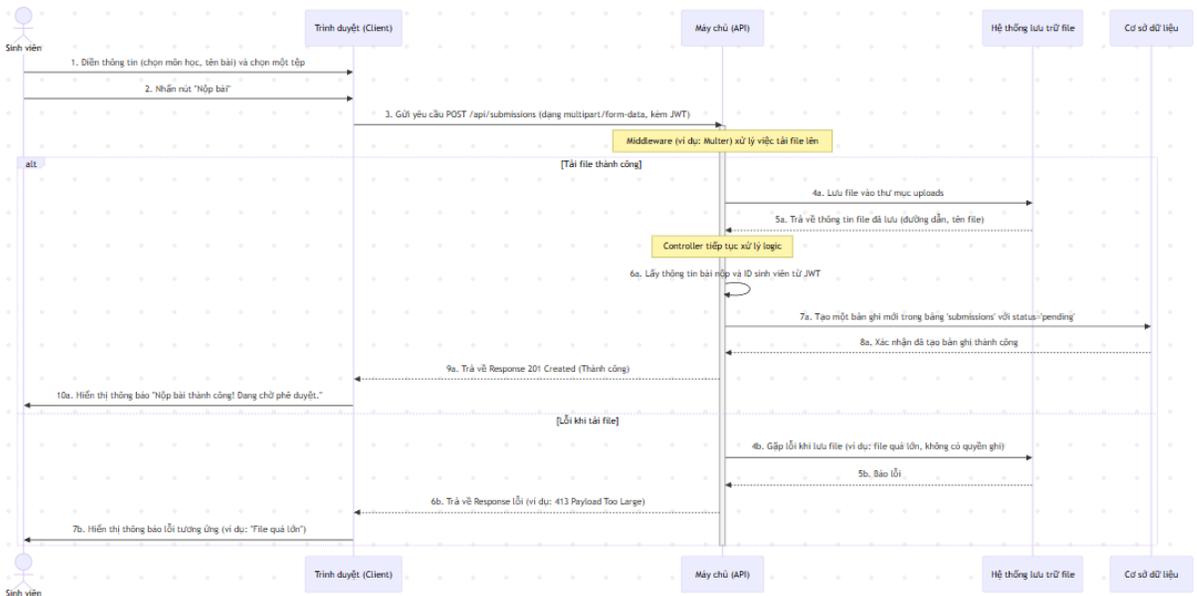
Ảnh 0.22. Biểu đồ tuần tự chức năng tìm kiếm tài liệu

3.3.6. Biểu đồ tuần tự usecase tìm kiếm tài liệu



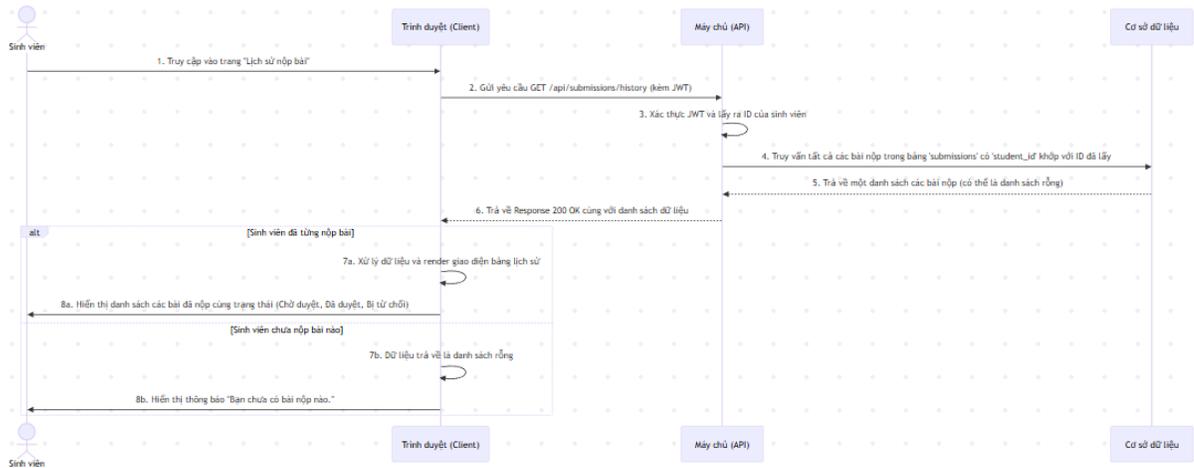
Ảnh 0.23. Biểu đồ tuần tự chức năng tải tài liệu

3.3.7. Biểu đồ tuần tự usecase nộp bài tập



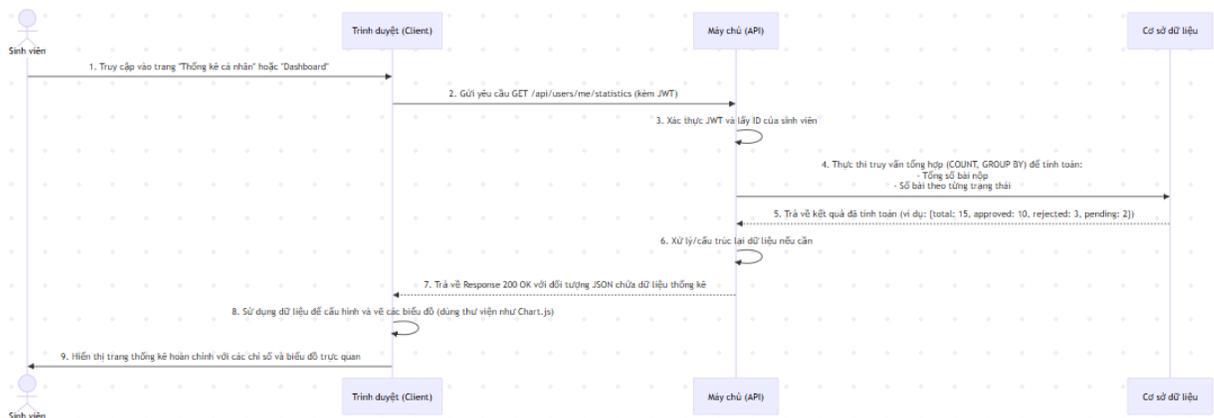
Ảnh 0.24. Biểu đồ tuần tự chức năng nộp bài tập

3.3.8. Biểu đồ tuần tự usecase xem lịch sử các bài đã nộp



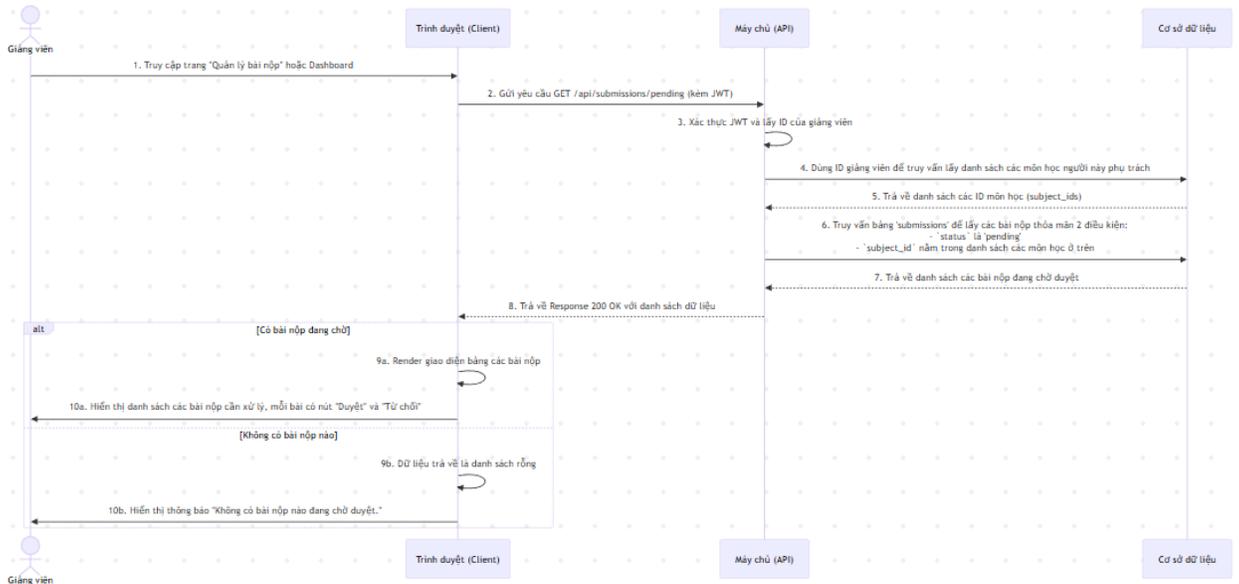
Ảnh 0.25. Biểu đồ tuần tự chức năng xem lịch sử các bài đã nộp

3.3.9. Biểu đồ tuần tự usecase xem trang thống kê cá nhân



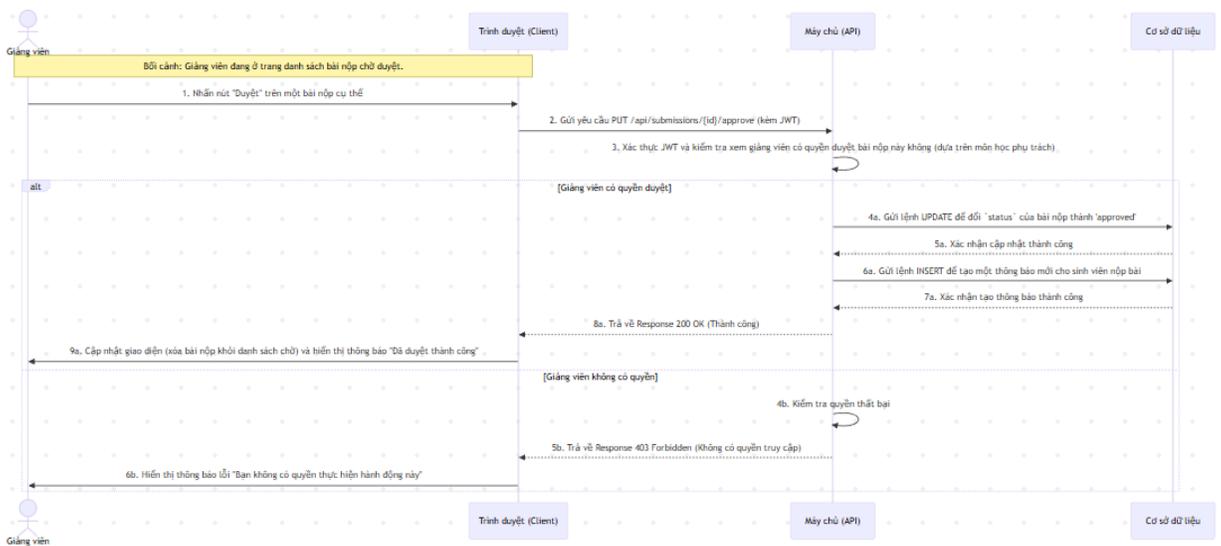
Ảnh 0.26. Biểu đồ tuần tự chức năng xem trang thống kê sinh viên

3.3.10. Biểu đồ tuần tự usecase Giảng viên xem danh sách bài nộp



Ảnh 0.27. Biểu đồ tuần tự chức năng quản lý bài nộp

3.3.11. Biểu đồ tuần tự usecase Giảng viên duyệt bài nộp



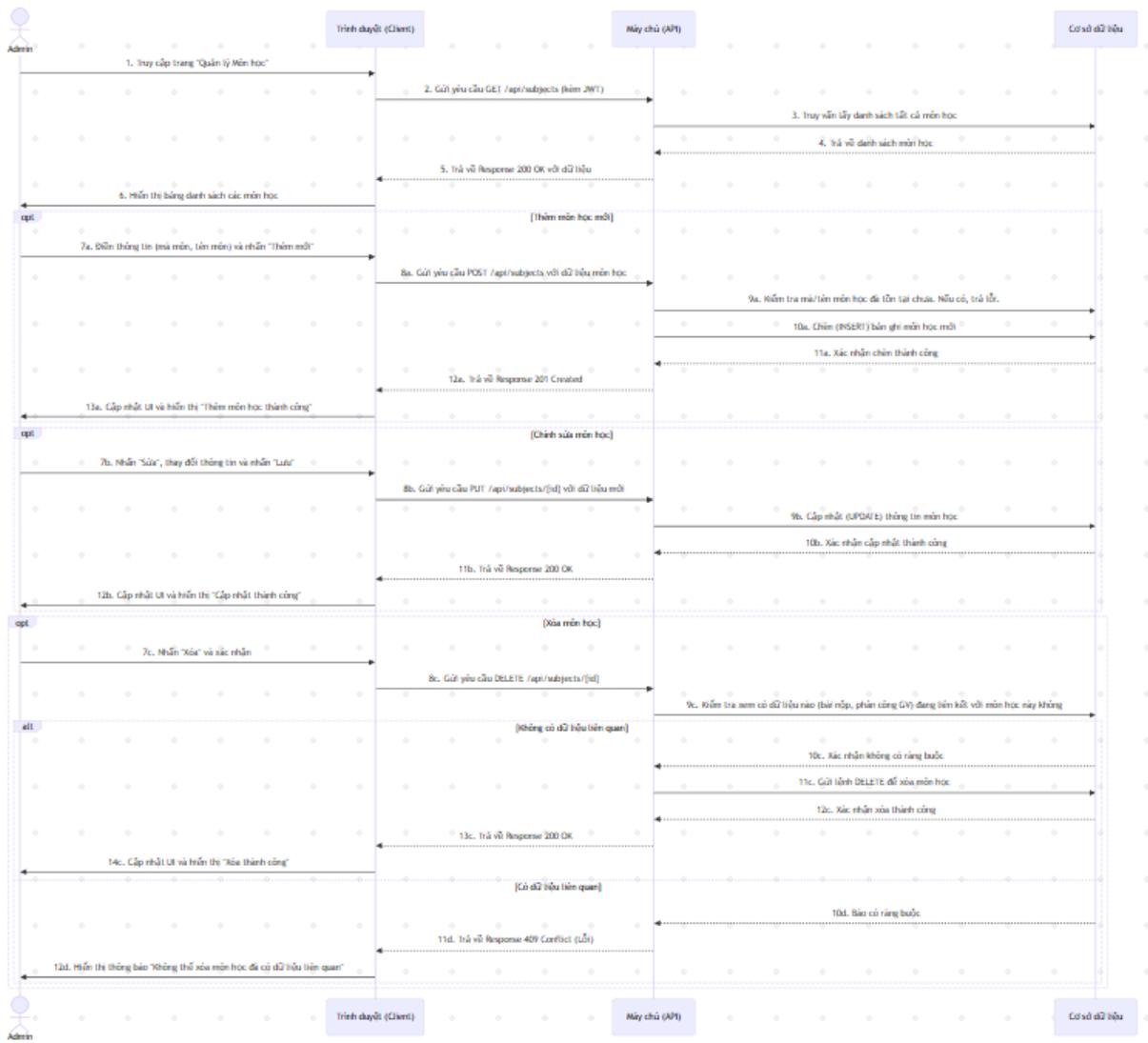
Ảnh 0.28. Biểu đồ tuần tự chức năng duyệt bài nộp

3.3.12. Biểu đồ tuần tự usecase Quản lý tài khoản người dùng



Ảnh 0.29. Biểu đồ tuần tự chức năng quản lý tài khoản

3.3.13. Biểu đồ tuần tự usecase Quản lý môn học



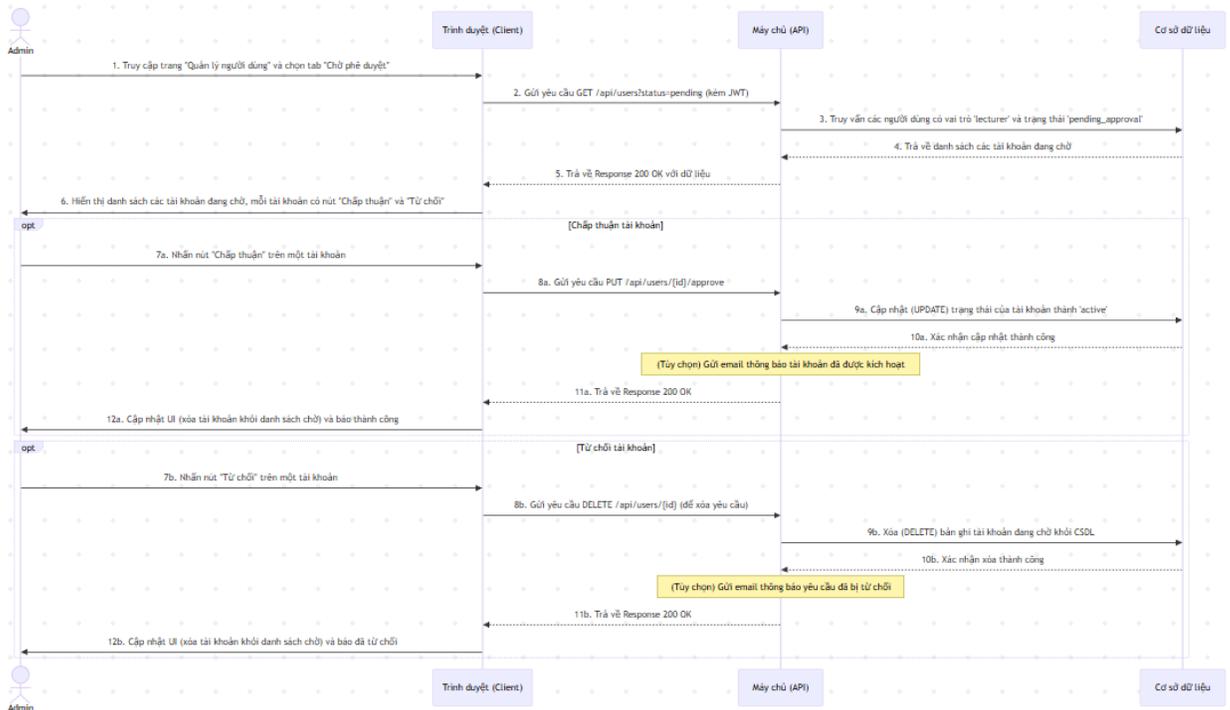
Ảnh 0.30. Biểu đồ tuần tự chức năng quản lý môn học

3.3.14. Biểu đồ tuần tự usecase Gán giảng viên vào môn học



Ảnh 0.31. Biểu đồ tuần tự chức năng gán giảng viên vào môn học

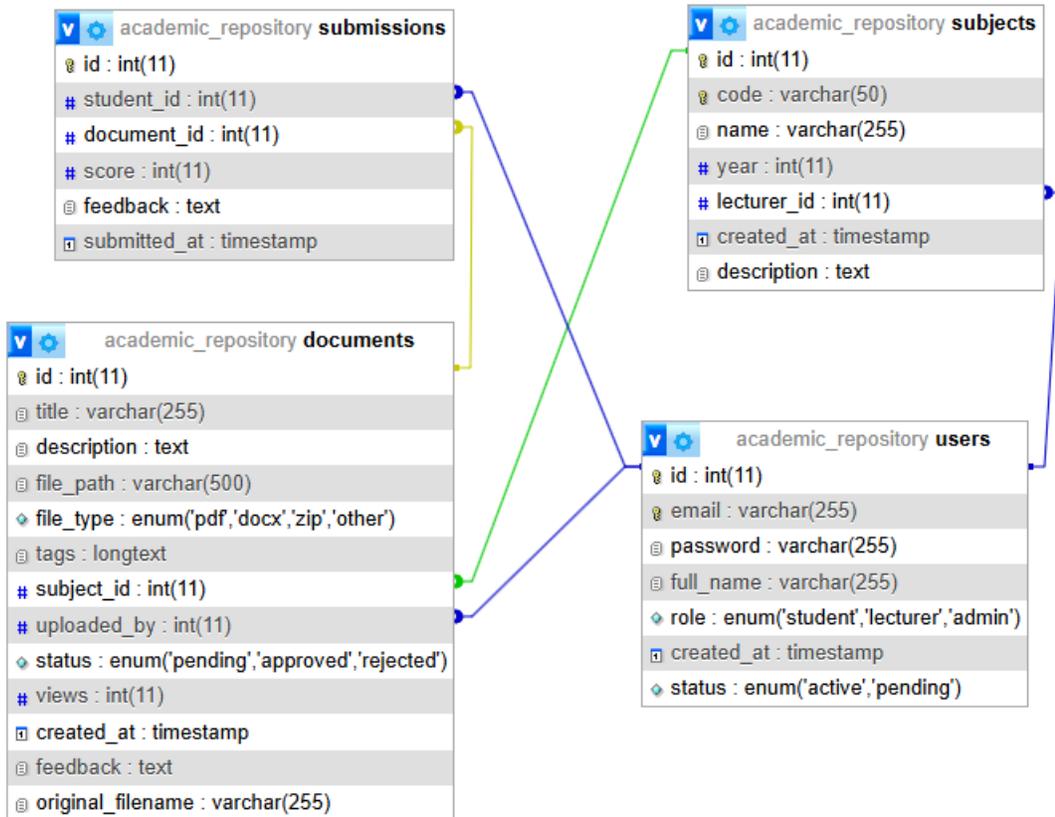
3.1.15. Biểu đồ tuần tự usecase các tài khoản giảng viên đang chờ phê duyệt



Ảnh 0.32. Biểu đồ tuần tự chức năng phê duyệt giảng viên

3.4. Thiết kế cơ sở dữ liệu

3.4.1. Sơ đồ cơ sở dữ liệu



Ảnh 0.33. Sơ đồ cơ sở dữ liệu

CHƯƠNG IV: CÀI ĐẶT VÀ XÂY DỰNG HỆ THỐNG

4.1. Môi trường phát triển

Để xây dựng hệ thống, các công nghệ và công cụ sau đã được lựa chọn và sử dụng:

Hạng mục	Công nghệ/Công cụ	Mô tả
Backend	Node.js	Nền tảng JavaScript phía máy chủ, giúp xây dựng các ứng dụng mạng nhanh và có khả năng mở rộng.
Backend	Express.js	Framework web cho Node.js, cung cấp một bộ tính năng mạnh mẽ để phát triển API RESTful một cách tối giản và linh hoạt.
Backend	MySQL	Hệ quản trị cơ sở dữ liệu quan hệ (SQL) phổ biến, nổi tiếng về sự ổn định và hiệu năng.
Backend	mysql2	Thư viện (driver) được tối ưu hóa cho Node.js để kết nối và tương tác với cơ sở dữ liệu MySQL.
Backend	JSON Web Token (JWT)	Tiêu chuẩn mở để tạo các token truy cập, được sử dụng

		để xác thực và phân quyền người dùng một cách an toàn.
Frontend	ReactJS	Thư viện JavaScript phổ biến để xây dựng giao diện người dùng (UI) dựa trên các thành phần (components).
Frontend	React Router	Thư viện định tuyến (routing) cho các ứng dụng React, cho phép điều hướng giữa các trang khác nhau.
Frontend	Axios	Thư viện HTTP client dựa trên Promise, được sử dụng để thực hiện các yêu cầu API từ trình duyệt đến máy chủ.
Frontend	Bootstrap/CSS	Framework CSS được sử dụng để xây dựng giao diện responsive, thân thiện với người dùng một cách nhanh chóng.
Công cụ	Visual Studio Code	Trình soạn thảo mã nguồn chính được sử dụng trong quá trình phát triển.
Công cụ	Postman	Công cụ để kiểm thử và gỡ lỗi các điểm cuối (endpoints) của API phía backend.

Công cụ	Git & GitHub	Hệ thống quản lý phiên bản phân tán, theo dõi các thay đổi trong mã nguồn và hỗ trợ cộng tác.
Công cụ	Node Package Manager (npm)	Trình quản lý gói cho JavaScript, dùng để cài đặt và quản lý các thư viện của dự án.

4.2. Cấu trúc dự án

Dự án được chia thành hai phần riêng biệt: academic-backend (máy chủ) và frontend (máy khách).

4.2.1. Cấu trúc dự án Backend

Mã nguồn backend được tổ chức theo mô hình MVC (Model-View-Controller) biến thể, giúp mã nguồn rõ ràng, dễ bảo trì và mở rộng.

```

academic-backend/
├── middleware/ # Chứa các hàm trung gian (xác thực, xử lý file upload)
│   ├── auth.js
│   └── upload.js
├── models/ # Chứa các lớp/hàm để tương tác với các bảng trong CSDL MySQL
│   ├── Document.js
│   ├── Subject.js
│   └── User.js
├── routes/ # Định nghĩa các điểm cuối (endpoints) của API
│   ├── admin.js
│   ├── auth.js
│   └── ...
├── uploads/ # Thư mục lưu trữ các file do người dùng tải lên
├── utils/ # Chứa các hàm tiện ích dùng chung
├── .env # File cấu hình môi trường (biến môi trường)
├── db.js # Cấu hình và tạo connection pool để kết nối đến CSDL MySQL
└── server.js # File khởi động chính của máy chủ

```

Ảnh 4.1. Cấu trúc thư mục backend

4.2.2. Cấu trúc dự án Frontend

Mã nguồn frontend được xây dựng bằng Create React App và có cấu trúc thư mục được tổ chức theo chức năng.

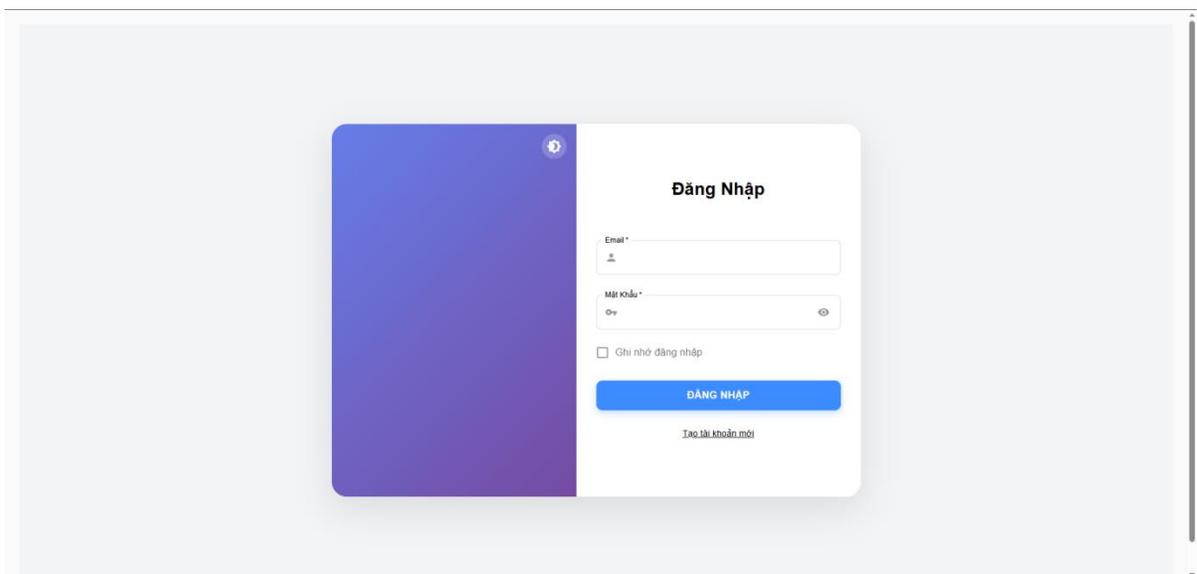
```
frontend/
├── public/      # Chứa các file tĩnh (index.html, favicon)
├── src/        # Thư mục chứa mã nguồn chính của ứng dụng
│   ├── components/ # Chứa các thành phần UI tái sử dụng (Header, Sidebar)
│   ├── contexts/  # Quản lý trạng thái toàn cục (AuthContext)
│   ├── pages/    # Chứa các trang chính của ứng dụng (Login, Dashboard)
│   ├── App.js    # Thành phần gốc, nơi định tuyến các trang
│   ├── index.js  # Điểm vào của ứng dụng React
│   └── setupProxy.js # Cấu hình proxy để gọi API từ backend
└── package.json # Chứa thông tin và các gói phụ thuộc của dự án
```

Ảnh 4.2. Cấu trúc thư mục frontend

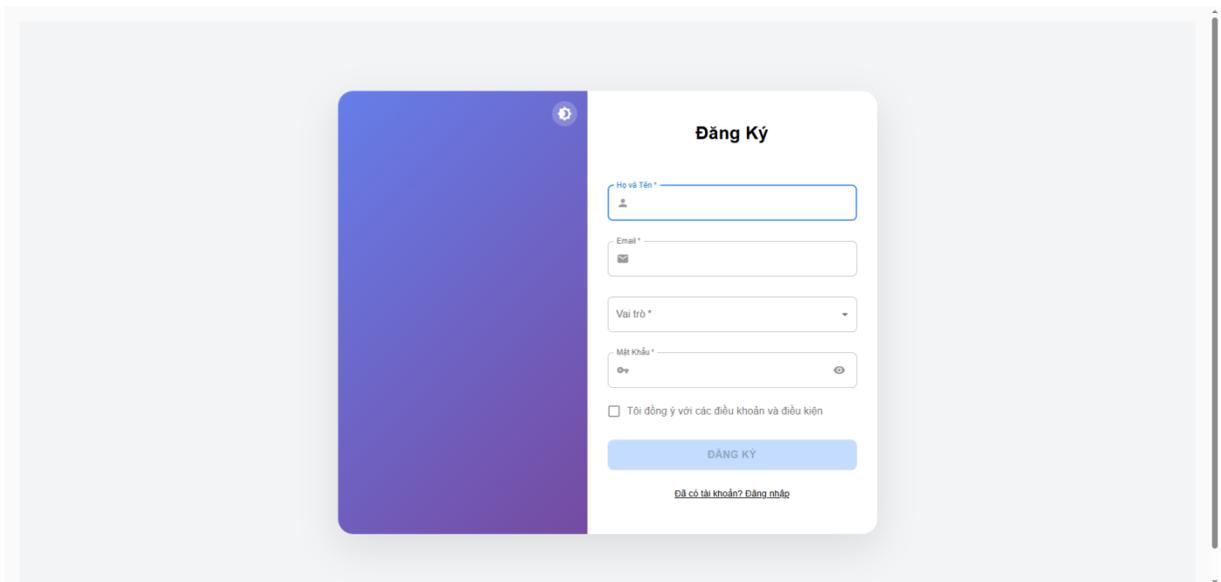
4.3. Mô tả các chức năng chính

4.3.1. Chức năng chung

Đăng nhập và Đăng ký: Hệ thống cho phép người dùng mới đăng ký tài khoản (với vai trò sinh viên) và người dùng đã có tài khoản đăng nhập để truy cập các chức năng tương ứng với vai trò của mình.

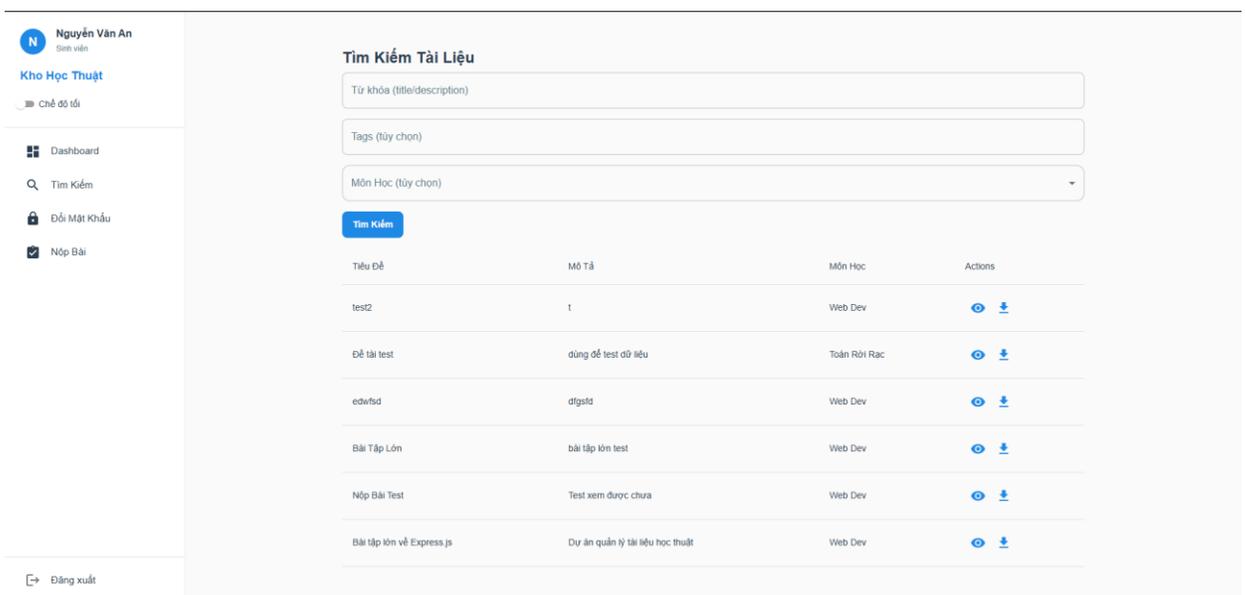


Ảnh 4.3. Giao diện Trang đăng nhập



Ảnh 4.4. Giao diện trang đăng ký

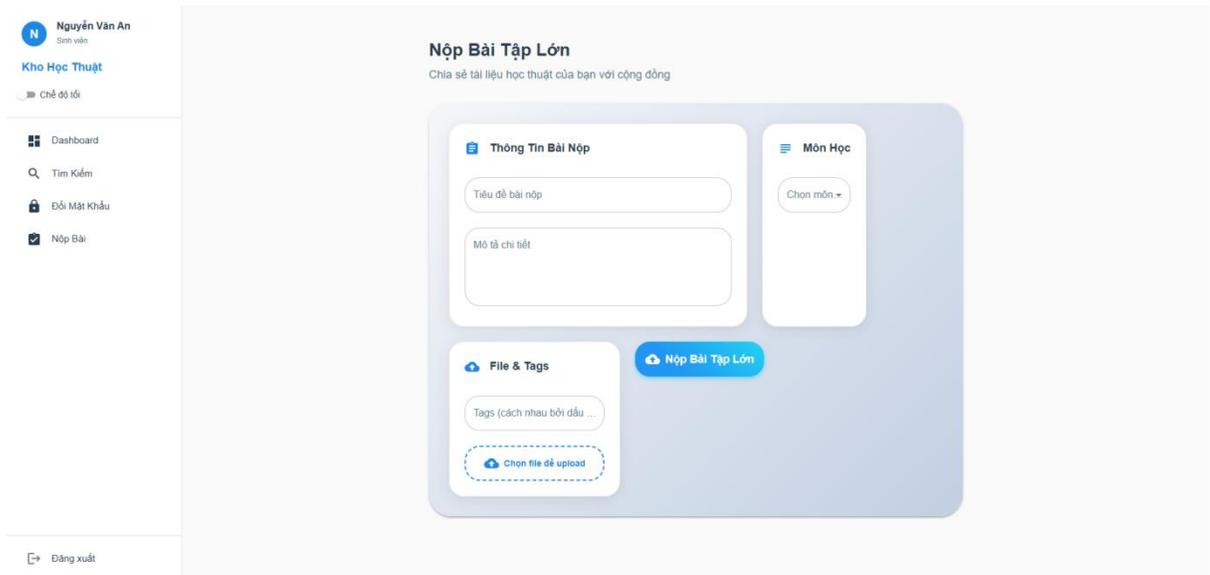
Tìm kiếm và Tải tài liệu: Sau khi đăng nhập, tất cả người dùng đều có thể truy cập kho tài liệu chung, tìm kiếm các bài nộp đã được duyệt theo từ khóa hoặc môn học và tải về tham khảo.



Ảnh 4.5. Giao diện trang tìm kiếm và tải tài liệu

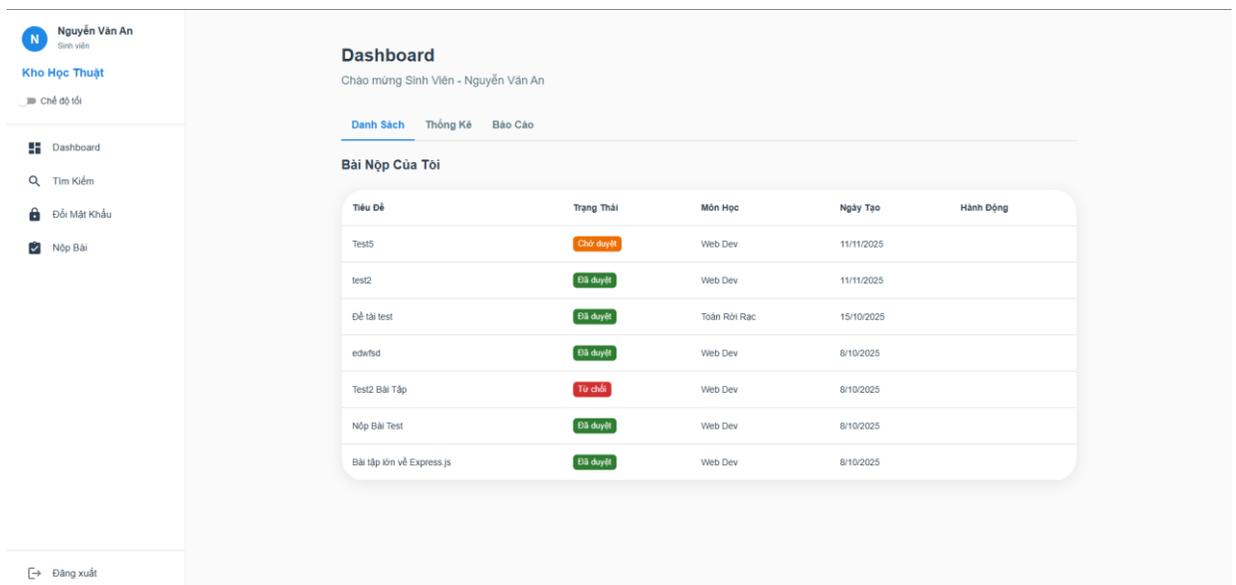
4.3.2. Chức năng của Sinh viên

Nộp bài tập/tài liệu: Sinh viên có thể nộp bài tập lớn hoặc đồ án của mình cho một môn học cụ thể. Bài nộp sẽ ở trạng thái "Chờ duyệt".



Ảnh 4.6. Giao diện trang nộp bài

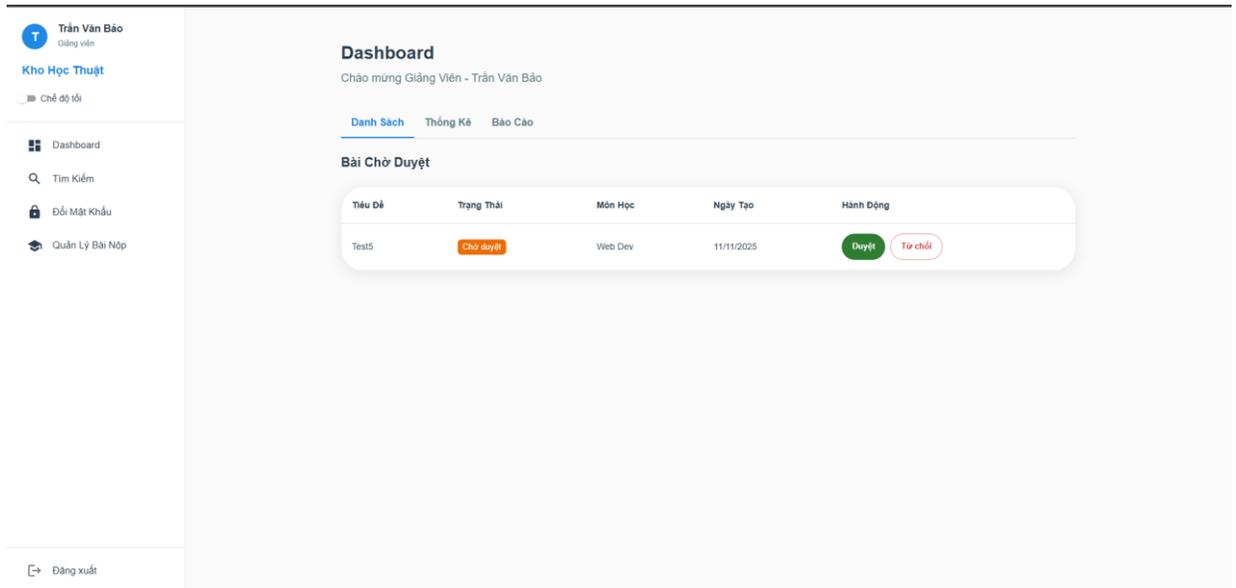
Theo dõi lịch sử và trạng thái bài nộp: Sinh viên có thể xem lại danh sách các bài mình đã nộp, theo dõi trạng thái (Chờ duyệt, Đã duyệt, Bị từ chối) và xem phản hồi từ giảng viên nếu bài bị từ chối.



Ảnh 4.7. Giao diện trang lịch sử nộp bài

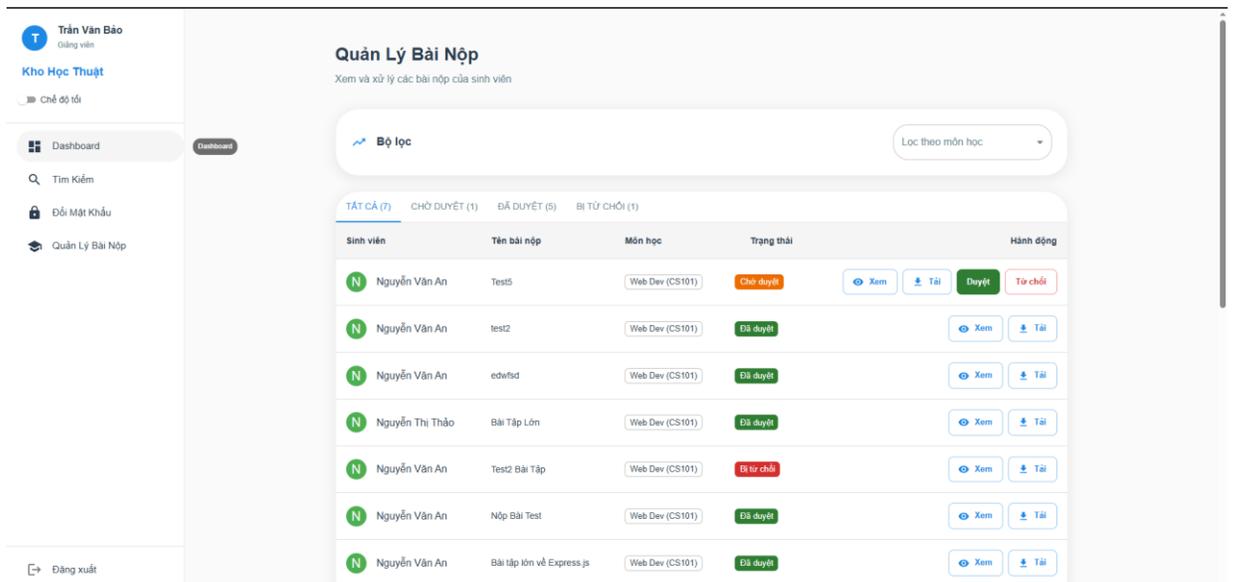
4.3.3. Chức năng của Giảng viên

Duyệt bài nộp: Giảng viên xem danh sách các bài nộp đang chờ duyệt thuộc các môn mình phụ trách và ra quyết định "Duyệt" hoặc "Từ chối" (kèm phản hồi).



Ảnh 4.7. Giao diện trang duyệt bài của giảng viên

Quản lý bài nộp: Xem lại tất cả các bài nộp (ở mọi trạng thái) trong các môn học mình quản lý.



Ảnh 4.8. Giao diện trang quản lý tất cả bài nộp của Giảng viên



Ảnh 4.9. Giao diện trang báo cáo của Giảng viên

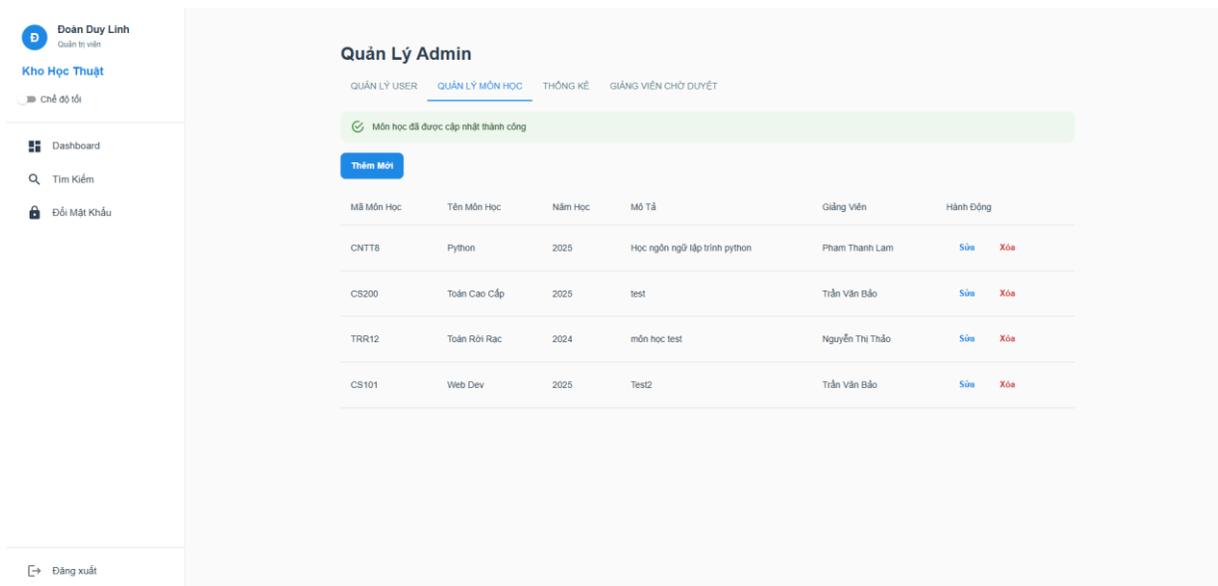
4.3.4. Chức năng của Admin

Dashboard tổng quan: Xem các số liệu thống kê toàn hệ thống như tổng số người dùng, tổng số tài liệu, dung lượng đã sử dụng.

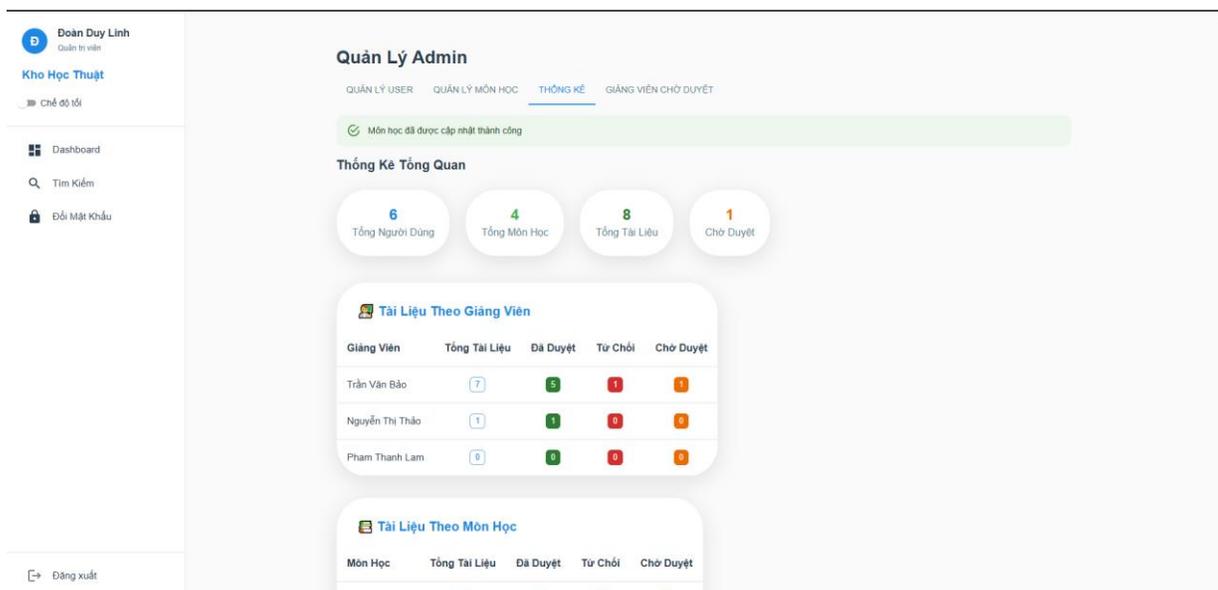
The 'Quản Lý Admin' interface shows the following user list:

Email	Họ Tên	Vai Trò	Ngày Tạo	Hành Động
student3@example.com	Phạm Hữu Biên	student	16/10/2025	Sửa Xóa
admin1@example.com	Đoàn Duy Linh	admin	08/10/2025	
student2@example.com	Nguyễn Thị Thảo	lecturer	08/10/2025	Sửa Xóa
lecturer1@example.com	Trần Văn Bảo	lecturer	08/10/2025	Sửa Xóa
student1@example.com	Nguyễn Văn An	student	08/10/2025	Sửa Xóa

Ảnh 4.9. Giao diện trang quản lý user của admin



Ảnh 4.10. Giao diện trang quản lý môn học của admin



Ảnh 4.11. Giao diện trang thống kê của admin

CHƯƠNG V: KẾT LUẬN

Sau một thời gian nghiên cứu, phân tích và triển khai, dự án "Xây dựng hệ thống quản lý dữ liệu học tập" đã hoàn thành các mục tiêu cơ bản đề ra. Chương này sẽ tổng kết lại những kết quả đã đạt được, chỉ ra những mặt còn hạn chế và đề xuất các hướng phát triển tiềm năng cho dự án trong tương lai.

5.1. Kết quả đạt được

Trong quá trình triển khai và phát triển dự án, em đã đạt được những kết quả quan trọng, bao gồm:

- Hoàn thiện tài liệu: Đã phân tích và đặc tả chi tiết các yêu cầu chức năng của hệ thống thông qua các biểu đồ Use Case, cũng như thiết kế chi tiết luồng hoạt động cho từng chức năng qua các biểu đồ tuần tự (Sequence Diagram).
- Xây dựng cơ sở dữ liệu: Đã thiết kế và triển khai thành công cơ sở dữ liệu quan hệ bằng MySQL, đảm bảo cấu trúc để lưu trữ và quản lý hiệu quả dữ liệu cho các đối tượng chính: Người dùng (với 3 vai trò), Môn học, Bài nộp, và Thông báo.
- Xây dựng Backend: Đã phát triển thành công hệ thống API phía máy chủ bằng Node.js và Express.js, xử lý toàn bộ logic nghiệp vụ cho các chức năng cốt lõi như xác thực, phân quyền, quản lý dữ liệu và tải tệp lên.
- Xây dựng Frontend: Đã tạo dựng giao diện người dùng hoàn chỉnh bằng React.js, cung cấp trải nghiệm người dùng riêng biệt và phù hợp cho cả ba vai trò: Sinh viên, Giảng viên và Quản trị viên.
- Hoàn thiện các chức năng chính:
 - + Sinh viên: Đăng ký, đăng nhập, đổi mật khẩu, nộp bài tập cho môn học, xem lịch sử và trạng thái bài nộp, xem phản hồi của giảng viên, và xem trang thống kê cá nhân.

- + Giảng viên: Quản lý các bài nộp đang chờ duyệt thuộc môn mình phụ trách, xem lại tất cả bài nộp, duyệt hoặc từ chối bài nộp (kèm phản hồi), và xem trang thống kê của các môn học được phân công.
- + Quản trị viên (Admin): Quản lý toàn diện tài khoản người dùng (xem, sửa, xóa, phân quyền), quản lý danh mục môn học (thêm, sửa, xóa), gán giảng viên vào môn học, phê duyệt tài khoản giảng viên mới, và theo dõi dashboard thống kê toàn hệ thống.

5.2. Hạn chế

Bên cạnh những kết quả đạt được, dự án vẫn còn một số hạn chế cần được cải thiện trong các phiên bản sau:

- Hệ thống thông báo còn đơn giản: Chức năng thông báo hiện tại chỉ hiển thị trên giao diện web, chưa có thông báo real-time (thời gian thực) hoặc tích hợp gửi thông báo tự động qua email cho người dùng.
- Thiếu tính năng tương tác trực tiếp: Chưa có hệ thống bình luận hay thảo luận trực tiếp trên từng bài nộp, khiến việc trao đổi giữa giảng viên và sinh viên về một vấn đề cụ thể còn hạn chế.
- Chức năng tìm kiếm cơ bản: Chức năng tìm kiếm tài liệu mới chỉ hỗ trợ tìm theo từ khóa, chưa có các bộ lọc nâng cao như tìm theo tác giả, theo ngày tháng, hoặc sắp xếp kết quả.
- Luồng nộp bài chưa linh hoạt: Hệ thống chưa hỗ trợ sinh viên nộp lại (resubmit) phiên bản mới cho một bài tập đã bị từ chối, mà yêu cầu phải tạo một bài nộp hoàn toàn mới.
- Bảo mật ở mức cơ bản: Hệ thống đã có xác thực và phân quyền nhưng chưa áp dụng các biện pháp bảo mật nâng cao như giới hạn số lần yêu cầu (rate limiting) để chống tấn công brute-force, hay xác thực hai yếu tố (2FA).

5.3. Hướng phát triển

Dựa trên nền tảng đã xây dựng và các hạn chế đã nêu, dự án có thể được mở rộng và phát triển trong tương lai với các hướng sau:

- Nâng cấp hệ thống thông báo: Tích hợp công nghệ WebSocket để cung cấp thông báo real-time ngay trên giao diện, đồng thời xây dựng chức năng gửi email tự động khi có các sự kiện quan trọng (bài được duyệt/từ chối, có tài khoản mới chờ duyệt).
- Xây dựng tính năng thảo luận: Thêm một mục bình luận/trao đổi cho mỗi bài nộp, cho phép sinh viên và giảng viên có thể thảo luận trực tiếp, giúp quá trình phản hồi và sửa lỗi hiệu quả hơn.
- Cải thiện luồng nộp bài: Cho phép sinh viên nộp lại phiên bản mới cho các bài đã bị từ chối. Xa hơn là xây dựng một hệ thống quản lý phiên bản (version control) đơn giản cho tài liệu.
- Nâng cấp tìm kiếm và thống kê: Phát triển bộ lọc nâng cao cho chức năng tìm kiếm. Bổ sung các biểu đồ và báo cáo chi tiết hơn cho Admin và Giảng viên (ví dụ: thời gian trung bình để duyệt bài, tỷ lệ nộp bài theo từng lớp/môn).
- Tích hợp kiểm tra đạo văn: Nghiên cứu và tích hợp các API kiểm tra đạo văn của bên thứ ba để tự động quét các bài nộp của sinh viên, hỗ trợ giảng viên trong quá trình đánh giá tính trung thực.
- Tăng cường bảo mật: Áp dụng các biện pháp bảo mật nâng cao như Rate Limiting, bảo vệ chống tấn công CSRF, và xem xét tích hợp tùy chọn xác thực hai yếu tố (2FA) cho các tài khoản quan trọng như Admin và Giảng viên.

Tài liệu tham khảo

- [1] Mozilla Developer Network (2024). *Client-Server overview*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview
- [2] React Team. *React Official Documentation*. [Online]. Available: <https://react.dev/>
- [3] React Router Team. *React Router Documentation*. [Online]. Available: <https://reactrouter.com/>
- [4] Node.js Foundation. *Node.js Documentation*. [Online]. Available: <https://nodejs.org/en/docs/>
- [5] Express Team. *Express.js - Fast, unopinionated, minimalist web framework for Node.js*. [Online]. Available: <https://expressjs.com/>
- [6] W3Schools. *MySQL Tutorial*. [Online]. Available: <https://www.w3schools.com/mysql/>
- [7] JWT.IO. *JSON Web Token Introduction*. [Online]. Available: <https://jwt.io/introduction>
- [8] OpenJS Foundation. *mysql2 - npm*. [Online]. Available: <https://www.npmjs.com/package/mysql2>
- [9] OpenJS Foundation. *bcrypt.js - npm*. [Online]. Available: <https://www.npmjs.com/package/bcryptjs>
- [10] Express Team. *Multer - npm*. [Online]. Available: <https://www.npmjs.com/package/multer>

- [11] Trần Đức Tài (2023). *Sử dụng Middleware trong Express.js để bảo vệ Website API*, Đại học Công nghệ Thông tin - ĐHQG TP.HCM.
- [12] Nguyễn Văn Minh (2022). *Xây dựng và kết nối frontend-backend trong ứng dụng thương mại điện tử*, Trường Đại học Bách Khoa Hà Nội.