

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO
DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

HỆ THỐNG QUẢN LÝ ĐĂNG KÝ ĐỀ TÀI

Giảng viên hướng dẫn: **ThS. Mai Văn Mạnh**

Sinh viên thực hiện: **Tô Vĩnh Khang - 51800408**

Bùi Quang Khải – 51800785

Lớp: **18050203**

Khóa: **22**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO
DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

HỆ THỐNG QUẢN LÝ **ĐĂNG KÝ ĐỀ TÀI**

Giảng viên hướng dẫn: **ThS. Mai Văn Mạnh**

Sinh viên thực hiện: **Tô Vĩnh Khang - 51800408**

Bùi Quang Khải – 51800785

Lớp: **18050203**

Khóa: **22**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Chúng em chân thành cảm ơn Khoa Công nghệ thông tin và Trường Đại học Tôn Đức Thắng đã cho chúng em được học tập trong suốt thời gian qua. Chân thành cảm ơn Thầy Mai Văn Mạnh đã hướng dẫn và tạo dựng những điều kiện tốt nhất để chúng em hoàn thành Dự án Công nghệ thông tin 2 với đề tài “Hệ thống Quản lý Đăng ký Đề tài”. Nhờ những nhận xét, đánh giá bổ sung từ thầy đã giúp chúng em ngày một hoàn thiện hơn, có thêm kiến thức rộng mở hơn từ mọi giai đoạn của quy trình phát triển phần mềm. Hiểu biết khái quát hơn về cách ứng dụng các công nghệ tiên tiến, mới nhất hiện nay để áp dụng vào việc giải quyết các vấn đề phát sinh của thực tiễn.

Trong quá trình thực hiện, nhóm vẫn khó tránh khỏi những sai sót không mong muốn, kính mong thầy có thể góp ý và giúp đỡ chúng em. Nhóm xin chân thành cảm ơn thầy.

DỰ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là sản phẩm của riêng chúng tôi và được sự hướng dẫn của thầy Mai Văn Mạnh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung dự án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 08 tháng 08 năm 2022

Tác giả

(ký tên và ghi rõ họ tên)

Tô Vĩnh Khang

Bùi Quang Khải

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

TÓM TẮT

Xã hội ngày càng phát triển, nhu cầu về việc số hóa các quy trình thủ công càng được chú trọng. Trước thực trạng đăng ký đề tài thông qua google form hiện nay, Nhóm đã triển khai thực hiện dự án thiết kế phần mềm “Hệ thống Quản lý Đăng ký đề tài”. Phần mềm được triển khai với mục tiêu giúp cho việc quản lý, đăng ký đề tài được tinh gọn, giảm thiểu được tối đa các quy trình thủ công cho văn phòng Khoa, giảng viên, sinh viên. Đáp ứng được nhu cầu về tổ chức đăng ký theo học kỳ, chia theo đợt đăng ký, rõ ràng trong từng giai đoạn thông qua các phân quyền cụ thể.

Hiện nay, cơ chế lập trình tách biệt server và client ra thành hai phần riêng biệt. Phía server có nhiệm vụ nhận yêu cầu, xử lý dữ liệu và trả về cho phía client, phía client có nhiệm vụ hiển thị giao diện và yêu cầu dữ liệu từ phía server để hiển thị lên cho người dùng. Với xu hướng lập trình web theo cơ chế client-side rendering đang ngày càng phổ biến, nhiều công nghệ, phương pháp được ra đời. Nổi bật nhất có thể kể đến là MERN Stack. Chính vì lý do này, công nghệ chính mà nhóm lựa chọn gồm MongoDB, ExpressJS, NodeJS và ReactJS. Các phần phụ trợ để việc hiện thực hệ thống tối ưu nhất gồm thư viện MaterialUI (tài nguyên giao diện), Redux (quản lý trạng thái lightweight cho React), TypeScript (giúp thêm type vào JavaScript, dễ quản lý và bảo trì), PNPM (nhanh hơn npm/yarn, có hiển thị thống kê các package đã thêm mới/đã xóa), ViteJS (nhanh hơn create-react-app), Firebase Storage (dùng để upload lưu trữ image, video and document) và theo mô hình RESTful API. Về hosting, nhóm sử dụng Railway, MondoDB Atlas cho phía backend và Netlify cho phía frontend. Phần mềm được triển khai với giao diện thân thiện với người dùng, dễ sử dụng. Thông tin được bảo mật an toàn, cập nhật nhất quán toàn vẹn. Thông qua sơ đồ usecase và ERD đã minh họa được các chức năng cần thiết và cốt lõi của website, cách thức hoạt động của từng chức năng và một số ràng buộc quan trọng.

Với đề tài “Hệ thống Quản lý Đăng ký đề tài”, nhóm mong muốn áp dụng những kiến thức đã học trong trường cùng với việc tìm hiểu nghiên cứu, phân tích hệ thống, những kiến thức bên ngoài xã hội để xây dựng hệ thống quản lý đăng ký đề tài cho Khoa đạt được hiệu quả cao nhất và có thể sử dụng được trong thời gian tới.

MỤC LỤC

LỜI CẢM ƠN	2
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	4
TÓM TẮT	5
MỤC LỤC	6
DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT	9
DANH MỤC CÁC BẢNG BIỂU VÀ HÌNH	10
Danh mục Hình	10
Danh mục Bảng	12
CHƯƠNG I: GIỚI THIỆU	13
1.1 Giới thiệu đề tài	13
1.2 Phương pháp sử dụng	13
1.3 Lý do và mục đích chọn đề tài	14
CHƯƠNG II: CƠ SỞ LÝ THUYẾT	15
2.1 NodeJS	15
2.2 ExpressJS	15
2.3 PNPM	16
2.4 MongoDB	16
2.5 ReactJS	17
2.6 Redux	18
2.7 TypeScript	19
2.8 ViteJS	20
2.9 MaterialUI	20
2.10 Firebase	20
2.11 Netlify	21
2.12 Railway	21
CHƯƠNG III: PHÂN TÍCH YÊU CẦU	22
3.1 Yêu cầu cơ bản	22
3.2 Yêu cầu chức năng	22

3.2.1 Mô tả	22
3.2.2 Xác định các tác nhân	23
3.2.3 Xác định các use case	23
3.2.4 Phiên bản công nghệ sử dụng	26
3.3 Yêu cầu phi chức năng	26
3.3.1 Yêu cầu về Hiệu suất	26
3.3.2 Yêu cầu về Bảo mật	26
3.3.3 Yêu cầu về Khả năng mở rộng	26
CHƯƠNG IV: THIẾT KẾ HỆ THỐNG	27
4.1 Sơ đồ use case Tổng quát hệ thống	27
4.2 Đặc tả use case	28
4.2.1 Đặc tả use case Đăng nhập	28
4.2.2 Đặc tả use case Đăng xuất	29
4.2.3 Đặc tả use case Xem danh sách đề tài	30
4.2.4 Đặc tả use case Đăng ký/Đề xuất đề tài	31
4.2.5 Đặc tả use case Xem lịch sử đề tài đăng ký cho GVHD	32
4.2.6 Đặc tả use case Xem lịch sử đề tài đăng ký cho GVKD	33
4.2.7 Đặc tả use case Quản lý đề tài hướng dẫn	34
4.2.8 Đặc tả use case Duyệt/Góp ý đề tài	35
4.2.9 Đặc tả use case Xem góp ý đề tài	36
4.2.10 Đặc tả use case Đăng ký nhóm	37
4.2.11 Đặc tả use case Quản lý học kỳ	37
4.2.12 Đặc tả use case Quản lý giảng viên	39
4.2.13 Đặc tả use case Quản lý sinh viên	40
4.2.14 Đặc tả use case Quản lý thiết lập	42
4.3 Sơ đồ hoạt động	44
4.3.1 Sơ đồ hoạt động Đăng nhập	44
4.3.2 Sơ đồ hoạt động Đăng xuất	45
4.3.3 Sơ đồ hoạt động Xem danh sách đề tài	45
4.3.4 Sơ đồ hoạt động Đăng ký/Đề xuất đề tài	46
4.3.5 Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVHD	47

4.3.6 Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVKD	47
4.3.7 Sơ đồ hoạt động Quản lý đề tài hướng dẫn	48
4.3.8 Sơ đồ hoạt động Duyệt/Góp ý đề tài	49
4.3.9 Sơ đồ hoạt động Xem góp ý đề tài	50
4.3.10 Sơ đồ hoạt động Đăng ký nhóm	50
4.3.11 Sơ đồ hoạt động Quản lý học kỳ	51
4.3.12 Sơ đồ hoạt động Quản lý giảng viên	52
4.3.13 Sơ đồ hoạt động Quản lý sinh viên	53
4.3.14 Sơ đồ hoạt động Quản lý thiết lập	54
4.4 Sơ đồ ERD	54
4.5 Bản vẽ Wireframe	55
CHƯƠNG V: HIỆN THỰC HỆ THỐNG VÀ KẾT QUẢ	56
5.1 Danh sách API	56
5.2 Kết quả đạt được	57
5.2.1 Màn hình Đăng nhập	58
5.2.2 Màn hình Xem danh sách đề tài	59
5.2.3 Màn hình Đăng ký đề tài	59
5.2.4 Màn hình Đề xuất đề tài	60
5.2.5 Màn hình Xem lịch sử đề tài đăng ký	60
5.2.6 Màn hình Quản lý đề tài hướng dẫn	61
5.2.7 Màn hình Đề tài được đề xuất	63
5.2.8 Màn hình Xem lịch sử đề tài SV đã đăng ký	64
5.2.9 Màn hình Quản lý học kỳ	64
5.2.10 Màn hình Quản lý giảng viên	66
5.2.11 Màn hình Quản lý sinh viên	67
CHƯƠNG VI: TỔNG KẾT	69
TÀI LIỆU THAM KHẢO	70

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

ERD	:	Entity Relationship Diagram
SQL	:	Structured Query Language
DB	:	Database
DOM	:	Document Object Model
API	:	Application Programming Interface
HTTP	:	Hyper Text Transfer Protocol
JS	:	JavaScript
CSS	:	Cascading Style Sheets
JSON	:	JavaScript Object Notation
ES	:	ECMAScript
RAM	:	Random Access Memory
SV	:	Sinh Viên
GVHD	:	Giảng Viên Hướng Dẫn
GVKD	:	Giảng Viên Kiểm Duyệt
ALL	:	Tất cả
HT	:	Hệ Thống

DANH MỤC CÁC BẢNG BIỂU VÀ HÌNH

Danh mục Hình

Hình 1. Hình ảnh về logo NodeJS	15
Hình 2. Hình ảnh về logo ExpressJS	16
Hình 3. Hình ảnh về logo PNPM	16
Hình 4. Hình ảnh về logo MongoDB	17
Hình 5. Hình ảnh về logo ReactJS	17
Hình 6. Hình ảnh về logo Redux	19
Hình 7. Hình ảnh về logo TypeScript	19
Hình 8. Hình ảnh về logo ViteJS	20
Hình 9. Hình ảnh về logo MaterialUI	20
Hình 10. Hình ảnh về logo Firebase	21
Hình 11. Hình ảnh về logo Netlify	21
Hình 12. Hình ảnh về logo Railway	21
Hình 13. Sơ đồ use case Tổng quát hệ thống	27
Hình 14. Sơ đồ hoạt động Đăng nhập	44
Hình 15. Sơ đồ hoạt động Đăng xuất	45
Hình 16. Sơ đồ hoạt động Xem danh sách đề tài	45
Hình 17. Sơ đồ hoạt động Đăng ký/Đề xuất đề tài	46
Hình 18. Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVHD	47
Hình 19. Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVKD	47
Hình 20. Sơ đồ hoạt động Quản lý đề tài hướng dẫn	48
Hình 21. Sơ đồ hoạt động Duyệt/Góp ý đề tài	49
Hình 22. Sơ đồ hoạt động Xem góp ý đề tài	50
Hình 23. Sơ đồ hoạt động Đăng ký nhóm	50
Hình 24. Sơ đồ hoạt động Quản lý học kỳ	51
Hình 25. Sơ đồ hoạt động Quản lý giảng viên	52
Hình 26. Sơ đồ hoạt động Quản lý sinh viên	53

Hình 27. Sơ đồ hoạt động Quản lý thiết lập	54
Hình 28. Sơ đồ ERD	55
Hình 29. Màn hình Đăng nhập	58
Hình 30. Màn hình Xem danh sách đề tài	59
Hình 31. Màn hình Đăng ký đề tài	59
Hình 32. Màn hình Đề xuất đề tài	60
Hình 33. Màn hình Xem lịch sử đề tài đăng ký	60
Hình 34. Màn hình Quản lý đề tài hướng dẫn	62
Hình 35. Màn hình Quản lý đề tài hướng dẫn	63
Hình 36. Màn hình Xem lịch sử đề tài SV đã đăng ký	64
Hình 37. Màn hình Quản lý học kỳ	65
Hình 38. Màn hình Quản lý giảng viên	67
Hình 39. Màn hình Quản lý sinh viên	68

Danh mục Bảng

Bảng 1. Danh sách các Tác nhân	23
Bảng 2. Danh sách các Use case	26
Bảng 3. Đặc tả use case Đăng nhập	29
Bảng 4. Đặc tả use case Đăng xuất	30
Bảng 5. Đặc tả use case Xem danh sách đề tài	31
Bảng 6. Đặc tả use case Đăng ký/Đề xuất đề tài	32
Bảng 7. Đặc tả use case Xem lịch sử đề tài đăng ký cho GVHD	33
Bảng 8. Đặc tả use case Xem lịch sử đề tài đăng ký cho GVKD	34
Bảng 9. Đặc tả use case Quản lý đề tài hướng dẫn	35
Bảng 10. Đặc tả use case Duyệt/Góp ý đề tài	36
Bảng 11. Đặc tả use case Xem góp ý đề tài	37
Bảng 12. Đặc tả use case Đăng ký nhóm	37
Bảng 13. Đặc tả use case Quản lý học kỳ	39
Bảng 14. Đặc tả use case Quản lý giảng viên	40
Bảng 15. Đặc tả use case Quản lý sinh viên	42
Bảng 16. Đặc tả use case Quản lý thiết lập	43
Bảng 17. Danh sách API	57

CHƯƠNG I: GIỚI THIỆU

1.1 Giới thiệu đề tài

Số hóa là mục tiêu, là nhu cầu cần thiết trong xã hội ngày nay. Để hiện đại hóa đất nước thì việc giảm thiểu quy trình thủ công càng được chú trọng. Ta không thể một sớm một chiều thay đổi tất cả, nhưng nó góp phần to lớn vào việc cải thiện và hướng mọi người đến với mục tiêu chung. Và đề tài này là một phần nhỏ để đóng góp cho mục tiêu chung đó.

Trước thực trạng đăng ký đề tài thông qua google form hiện nay ta cần triển khai một phần mềm giúp cho việc quản lý, đăng ký đề tài được tinh gọn, giảm thiểu được tối đa các quy trình thủ công cho văn phòng Khoa, giảng viên, sinh viên. Đáp ứng được nhu cầu về tổ chức đăng ký theo học kỳ, chia theo đợt đăng ký, rõ ràng trong từng giai đoạn thông qua các phân quyền cụ thể.

1.2 Phương pháp sử dụng

Hiện nay, cơ chế lập trình tách biệt server và client ra thành hai phần riêng biệt. Phía server có nhiệm vụ nhận yêu cầu, xử lý dữ liệu và trả về cho phía client, phía client có nhiệm vụ hiển thị giao diện và yêu cầu dữ liệu từ phía server để hiển thị lên cho người dùng. Với xu hướng lập trình web theo cơ chế client-side rendering đang ngày càng phổ biến, nhiều công nghệ, phương pháp được ra đời. Nổi bật nhất có thể kể đến là MERN Stack. Chính vì lý do này, công nghệ chính mà nhóm lựa chọn gồm MongoDB, ExpressJS, NodeJS và ReactJS.

Các phần phụ trợ để việc hiện thực hệ thống tối ưu nhất gồm thư viện MaterialUI (tài nguyên giao diện), Redux (quản lý trạng thái lightway cho React), TypeScript (giúp thêm type vào JavaScript, dễ quản lý và bảo trì), PNPM (nhanh hơn npm/yarn, có hiển thị thống kê các package đã thêm mới/đã xóa), ViteJS (nhanh hơn create-react-app), Firebase Storage (dùng để upload lưu trữ image, video and document) và theo mô hình RESTful API. Về hosting, nhóm sử dụng Railway, MondoDB Atlas cho phía backend và Netlify cho phía frontend.

1.3 Lý do và mục đích chọn đề tài

Thứ nhất, chúng em chọn đề tài này là vì muốn có thêm kiến thức về quy trình quản lý đăng ký đề tài của Khoa, cách thức quản lý thông tin của việc đăng ký đề xuất đề tài, học kỳ, giáo viên, sinh viên. Hiểu thêm về cách triển khai một hệ thống thực tế.

Thứ hai, giải quyết được các điểm đau của cả phía Khoa, giảng viên và sinh viên. Cũng như loại hình dịch vụ này là vô cùng cần thiết cho môi trường học đường khi mà nhu cầu của người đăng ký học ngày một tăng cao.

Với đề tài “Quản lý Đăng ký đề tài”, nhóm mong muốn áp dụng những kiến thức đã học trong trường cùng với việc tìm hiểu nghiên cứu, phân tích hệ thống, những kiến thức bên ngoài xã hội để xây dựng hệ thống quản lý đăng ký đề tài cho Khoa đạt được hiệu quả cao nhất và có thể sử dụng được trong thời gian tới.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1 NodeJS

NodeJS là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, chạy theo hướng sự kiện bất đồng bộ, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối thiểu tổng chi phí và tối đa khả năng mở rộng. NodeJS đã có một cộng đồng lập trình to lớn và nguồn thư viện khổng lồ được cung cấp để hỗ trợ tối đa cho lập trình viên trong việc phát triển. Việc dùng NodeJS sẽ giúp thống nhất về mặt ngôn ngữ lập trình ở cả 2 phía client và server. Nó là một thành phần chính trong MERN Stack và được dùng để xây dựng một server theo chuẩn kiến trúc RESTful API.



Hình 1. Hình ảnh về logo NodeJS

2.2 ExpressJS

ExpressJS là một framework được xây dựng trên nền tảng của NodeJS. Được phát hành dưới dạng phần mềm mã nguồn mở. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Nó hỗ trợ các method HTTP, middleware, route để tạo ra API vô cùng mạnh mẽ và dễ sử dụng. Gọn nhẹ và gần gũi với lập trình viên nên thường được ưu tiên sử dụng để tạo ra các tính năng đáp ứng nhu cầu nhanh chóng.



Hình 2. Hình ảnh về logo ExpressJS

2.3 PNPM

PNPM là package manager cho Javascript tương tự như npm, yarn, cung cấp những cải tiến đáng kể về tốc độ (nhanh hơn gấp 2 lần) và sử dụng dung lượng ổ đĩa hơn npm và yarn. Cũng sử dụng package.json để quản lý các dependencies và sử dụng file lock để đảm bảo cài đặt nhất quán trên nhiều máy.



Hình 3. Hình ảnh về logo PNPM

2.4 MongoDB

MongoDB là một chương trình cơ sở dữ liệu hướng tài liệu (Document) đa nền tảng có sẵn nguồn. Được phân loại là một chương trình cơ sở dữ liệu NoSQL. Mỗi bộ sưu tập (Collection) sẽ có các kích cỡ và các tài liệu khác nhau. Mỗi Document sẽ bao gồm các key và giá trị của key đó. Còn Collection thì là một nơi để chứa các Document, có thể hiểu nó tương tự với bảng trong SQL nhưng khác ở chỗ không cần phải định nghĩa trước các cột, các hàng hay kiểu dữ liệu. Trong Document còn có Field, có thể hiểu nó tương tự với thuật ngữ cột trong SQL nhưng giá trị này có thể đa dạng nhiều kiểu dữ liệu khác nhau. Một thuật ngữ quan trọng không kém đó là `_id`, là một trường bắt buộc phải có đối với mỗi Document, nó được tự động tạo ra khi Document được tạo.



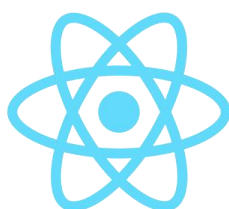
Hình 4. Hình ảnh về logo MongoDB

Nó cũng được tự động đánh index để tăng tốc độ truy vấn. Thông thường, nó được xem như một khóa chính hay một đại diện của Document vì thông qua nó có thể phân biệt được giữa các Document khác. Bên cạnh các kiểu dữ liệu quen thuộc như double, string, boolean,... MongoDB còn hỗ trợ kiểu dữ liệu array và json object.

Các dữ liệu được lưu trữ trong tài liệu kiểu JSON nên việc truy vấn rất nhanh. Dữ liệu lưu trữ phi cấu trúc, tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ, dữ liệu được ghi đệm lên RAM. Tuy nhiên, không có tính ràng buộc nên không được ứng dụng cho các mô hình giao dịch nào có yêu cầu độ chính xác cao, khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất cũng như yêu cầu một bộ nhớ RAM phải lớn.

2.5 ReactJS

ReactJS là một thư viện mã nguồn mở và miễn phí để xây dựng giao diện người dùng hoặc các thành phần UI. Nó chỉ quan tâm đến việc quản lý trạng thái và hiển thị trạng thái đó cho DOM, vì vậy việc tạo các ứng dụng React thường yêu cầu sử dụng các thư viện bổ sung để định tuyến, cũng như một số chức năng phía máy khách nhất định. Điều này cho phép lựa chọn bất kỳ thư viện nào mà nhà phát triển thích để hoàn thành các tác vụ như thực hiện truy cập mạng hoặc lưu trữ dữ liệu cục bộ.



Hình 5. Hình ảnh về logo ReactJS

Nói đến ReactJS thì không thể bỏ qua khái niệm vô cùng quan trọng đó là State. Trong mỗi Component của ReactJS, thường sẽ có một số dữ liệu cần được thay đổi theo sự tác động của người dùng lên Component. Các dữ liệu này sẽ được lưu vào trong biến trạng thái là State. Khi các dữ liệu được lưu ở các biến được render ra

chung với giao diện thì các dữ liệu này sẽ trở thành các giá trị tĩnh, nên dù cho có thể thay đổi giá trị của biến lưu trữ thì DOM sẽ không diễn ra sự cập nhật nào hay nói cách khác là trên giao diện không được thay đổi. Để khắc phục, đảm bảo phần giao diện người dùng luôn được cập nhật theo sự thay đổi của dữ liệu, các dữ liệu sẽ được gán vào State của Component và lấy dữ liệu đó ra thông qua State. Khi có sự tương tác nào đó của người dùng thì dữ liệu trong State sẽ thay đổi, kéo theo là giao diện sẽ được render lại để hiển thị phù hợp với dữ liệu. Việc thay đổi dữ liệu trong State không thể thực hiện bằng gán trực tiếp mà phải thông qua một vài cách đặc biệt đó là dùng useState Hook. Ngoài useState thì chúng tôi cũng có thể dùng useReducer để quản lý state cho một Component. Trong trường hợp muốn chia sẻ State với các Component khác thì dùng Global State hay còn được biết đến là State của toàn ứng dụng. Để dùng Global State, cần sử dụng một số công cụ quản lý State như Redux. Trong ReactJS, vẫn có thể sử dụng được CSS để tiến hành style cho các Component, có thể phân chia file CSS thành nhiều file riêng lẻ cho từng Component hay có thể chuyển đổi đoạn code CSS thành một Component của ReactJS.

Hiện nay, ReactJS được đánh giá là một trong những thư viện phổ biến nhất để xây dựng giao diện người dùng. Với sự phổ biến của ReactJS thì cũng có rất nhiều các module, các thư viện được tạo ra để sử dụng được trong ReactJS.

2.6 Redux

Redux là một thư viện nhỏ với một API giới hạn, mã nguồn mở dùng để quản lý và được thiết kế đơn giản để trở thành một vùng chứa có thể đoán trước cho trạng thái ứng dụng. Hiểu đơn giản đó là một store và có thể truy cập bởi các Component khác nhau. Sau đó, nếu ở Component nào nhận được sự tương tác của người dùng có liên quan đến việc thay đổi dữ liệu trạng thái trong store thì một hành động tương ứng sẽ được gửi đến store. Lúc này, những Component đang sử dụng dữ liệu State từ store sẽ lập tức nhận được sự thay đổi về dữ liệu và cập nhật sự thay đổi này lên giao diện.



Hình 6. Hình ảnh về logo Redux

Nó dùng để quản lý các State chính trong ứng dụng và quản lý các API tương ứng cho từng trạng thái bởi ko phải thông qua từng Component, một giá trị State mà khi muốn lấy chỉ việc gọi ra. Ở các dự án lớn, số lượng Component nhiều và có một sự liên kết với nhau về mặt dữ liệu thì với mỗi thao tác của người dùng thì dữ liệu sẽ được thay đổi, cập nhật ở nhiều Component.

2.7 TypeScript

TypeScript là một ngôn ngữ lập trình, một tập hợp siêu cú pháp nghiêm ngặt của JavaScript và thêm tính năng kiểu tĩnh tùy chọn vào ngôn ngữ. Nó được thiết kế để phát triển các ứng dụng lớn. Vì TypeScript là một siêu tập hợp JavaScript, các chương trình JavaScript hiện có cũng là các chương trình TypeScript hợp lệ.



Hình 7. Hình ảnh về logo TypeScript

Nó có thể được sử dụng để phát triển các ứng dụng cho cả thực thi phía máy khách và phía máy chủ. Những thách thức đối với việc xử lý mã JavaScript phức tạp đã dẫn đến nhu cầu về công cụ tùy chỉnh để dễ dàng phát triển các thành phần trong ngôn ngữ này.

2.8 ViteJS

ViteJS là công cụ xây dựng giao diện người dùng mới giúp cải thiện đáng kể trải nghiệm phát triển giao diện người dùng. Nó tập trung vào tốc độ và hiệu suất, sử dụng các bản nhập ES của trình duyệt gốc để cho phép hỗ trợ các trình duyệt hiện đại mà không cần quá trình xây dựng.



Hình 8. Hình ảnh về logo ViteJS

2.9 MaterialUI

MaterialUI là một thư viện cho phép nhập và sử dụng các thành phần khác nhau để tạo giao diện người dùng trong các ứng dụng React. Điều này giúp tiết kiệm đáng kể thời gian vì các nhà phát triển không cần phải viết mọi thứ từ đầu.



Hình 9. Hình ảnh về logo MaterialUI

2.10 Firebase

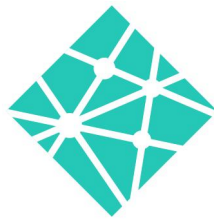
Firebase Storage là một giải pháp độc lập cho việc tải lên nội dung của người dùng, như hình ảnh và video. Nó sử dụng hệ thống tệp/ thư mục đơn giản để cấu trúc dữ liệu. Và được thiết kế đặc biệt để mở rộng ứng dụng, cung cấp bảo mật, đảm bảo khả năng phục hồi mạng.



Hình 10. Hình ảnh về logo Firebase

2.11 Netlify

Netlify là một nền tảng deploy tất cả trong một để tự động hóa cho việc triển khai dự án web với vài thao tác vô cùng đơn giản. Nó thường xuyên được dùng trong việc hosting nhanh một website, kết hợp các công cụ và API yêu thích để tạo ứng dụng nhanh nhất cho web.



Hình 11. Hình ảnh về logo Netlify

2.12 Railway

Railway là một nền tảng deploy nơi có thể cung cấp cơ sở hạ tầng, phát triển với cơ sở hạ tầng đó tại địa phương và sau đó triển khai lên cloud. Từ lần deploy đầu tiên đã vô cùng nhanh chóng bằng cách loại bỏ các rào cản phổ biến của nhà phát triển. Nó cũng sẽ giữ cho các ứng dụng chạy mượt mà ngay cả khi bị đẩy đến giới hạn.



Hình 12. Hình ảnh về logo Railway

CHƯƠNG III: PHÂN TÍCH YÊU CẦU

3.1 Yêu cầu cơ bản

Bài toán quản lý đăng ký đề tài đặt ra các nhiệm vụ cơ bản như sau:

Quản lý được các đợt đăng ký theo học kỳ, Quản lý được danh sách các giảng viên, danh sách sinh viên và nhóm sinh viên theo đợt đăng ký, Quản lý được danh sách đề tài đăng ký hoặc đề xuất, Đăng ký và đề xuất đề tài, Tra cứu lịch sử và kết quả đăng ký, Duyệt và góp ý đề tài, Đăng nhập bằng tài khoản HT hoặc bằng Google theo phân quyền cụ thể.

3.2 Yêu cầu chức năng

3.2.1 Mô tả

- Người trực tiếp thực hiện đề tài (được gọi là Sinh viên): Đăng nhập bằng Google, đăng ký nhóm, xem danh sách đề tài chung theo học kỳ, đăng ký đề tài sẵn có hoặc đề xuất đề tài riêng theo học kỳ, xem lịch sử đăng ký đề xuất và lời góp ý của giảng viên hướng dẫn.
- Người hướng dẫn/cố vấn đề tài (được gọi là Giảng viên hướng dẫn): Đăng nhập bằng Google, xem danh sách đề tài chung theo học kỳ, quản lý đề tài của chính mình theo học kỳ, duyệt và góp ý đề tài của sinh viên, xem lịch sử kết quả đề tài trình lên cho giảng viên kiểm duyệt và lời góp ý của giảng viên kiểm duyệt.
- Người hướng dẫn/cố vấn đề tài và có quyền Trưởng Khoa/Trưởng bộ môn (được gọi là Giảng viên kiểm duyệt): Đăng nhập bằng Google, tương tự giảng viên hướng dẫn, có thêm duyệt và góp ý đề tài của giảng viên hướng dẫn.
- Người tổ chức các đợt đăng ký đề tài/Cán bộ trực Văn phòng Khoa (được gọi là Admin): Đăng nhập bằng tài khoản HT, quản lý học kỳ, quản lý giảng viên, quản lý sinh viên, xem danh sách đề tài.

3.2.2 Xác định các tác nhân

Sau khi đã tìm hiểu, nhóm tổng hợp được các tác nhân của hệ thống như sau:

STT	Tác nhân	Mô tả
1	Sinh viên (SV)	Đăng nhập bằng Google, đăng ký nhóm, xem danh sách đề tài chung theo học kỳ, đăng ký đề tài sẵn có hoặc đề xuất đề tài riêng theo học kỳ, xem lịch sử đăng ký đề xuất và lời góp ý của giảng viên hướng dẫn.
2	Giảng viên hướng dẫn (GVHD)	Đăng nhập bằng Google, xem danh sách đề tài chung theo học kỳ, quản lý đề tài của chính mình theo học kỳ, duyệt và góp ý đề tài của sinh viên, xem lịch sử kết quả đề tài trình lên cho giảng viên kiểm duyệt và lời góp ý của giảng viên kiểm duyệt.
3	Giảng viên kiểm duyệt (GVKD)	Đăng nhập bằng Google, tương tự giảng viên hướng dẫn, có thêm duyệt và góp ý đề tài của giảng viên hướng dẫn.
4	VP Khoa (ADMIN)	Đăng nhập bằng tài khoản HT, quản lý học kỳ, quản lý giảng viên, quản lý sinh viên, xem danh sách đề tài.

Bảng 1. Danh sách các Tác nhân

3.2.3 Xác định các use case

Dựa vào yêu cầu đặc ra và với các tác nhân tương ứng nhóm xác định được các use case cho hệ thống gồm:

ID	Use case	Mô tả	Tác nhân tương ứng
----	----------	-------	--------------------

1	Đăng nhập	Người dùng đăng nhập vào hệ thống bằng tài khoản của Google hoặc tài khoản HT.	ALL
2	Đăng xuất	Người dùng đăng xuất khỏi hệ thống.	ALL
3	Xem danh sách đề tài	Người dùng xem danh sách đề tài theo học kỳ được chọn. Có thể nhận xem chi tiết đề tài.	ALL
4	Đăng ký nhóm	Sinh viên đăng ký nhóm, tối đa 2 người.	SV
5	Đăng ký/Đề xuất đề tài	Sinh viên đăng ký đề tài có sẵn trong danh sách đề tài hoặc đề xuất đề tài, tối đa 10 đề tài.	SV
6	Xem lịch sử đề tài đăng ký cho GVHD	Sinh viên xem danh sách kết quả các đề tài đã đăng ký.	SV
7	Xem góp ý đề tài từ GVHD	Sinh viên xem lời góp ý của giảng viên hướng dẫn.	SV
8	Duyệt đề tài cho SV	Giảng viên hướng dẫn duyệt đề tài của sinh viên, nếu đồng ý duyệt thì các đề tài khác đã đăng ký sẽ đc kết quả là từ chối, các nhóm sinh viên còn lại có đăng ký cùng đề tài đc duyệt cũng sẽ bị từ chối.	GVHD GVKD
9	Góp ý đề tài cho SV	Giảng viên hướng dẫn gửi lời góp ý nhận xét đến cho sinh viên.	GVHD GVKD
10	Quản lý đề tài hướng dẫn	Giảng viên hướng dẫn được phép Xem/Thêm/Xóa/Sửa danh sách đề tài của chính mình.	GVHD GVKD

11	Xem lịch sử đề tài đăng ký cho GVKD	Giảng viên hướng dẫn xem kết quả các đề tài của chính mình trình lên cho giảng viên kiểm duyệt.	GVHD GVKD
12	Xem góp ý đề tài từ GVKD	Giảng viên hướng dẫn xem lời góp ý của giảng viên kiểm duyệt.	GVHD GVKD
13	Duyệt đề tài cho GVHD	Giảng viên kiểm duyệt duyệt đề tài của giảng viên hướng dẫn, nếu đồng ý duyệt thì giảng viên hướng dẫn có thể bắt đầu hướng dẫn cho sinh viên được ngay.	GVKD
14	Góp ý đề tài cho GVHD	Giảng viên kiểm duyệt gửi lời góp ý nhận xét đề tài đến cho giảng viên hướng dẫn.	GVKD
15	Quản lý học kỳ	Để khởi tạo đợt đăng ký cho Khoa, Admin được Xem/Thêm/Xóa/Sửa danh sách học kỳ để quản lý theo từng đợt đăng ký cụ thể.	ADMIN
16	Quản lý giảng viên	Để giảng viên thao tác được trong học kỳ nào thì Admin đưa danh sách giảng viên vào trong học kỳ được chọn. Admin được Xem/Thêm/Xóa/Sửa hoặc import file danh sách giảng viên. Giảng viên được đánh dấu kiểm duyệt gọi tắt là Giảng viên kiểm duyệt.	ADMIN
17	Quản lý sinh viên	Để sinh viên thao tác được trong học kỳ nào thì Admin đưa danh sách sinh viên vào trong học kỳ được chọn. Admin được Xem/Thêm/Xóa/Sửa hoặc import file danh sách sinh viên.	ADMIN
18	Quản lý	Để giới hạn số lượng đề tài được đăng ký, Admin được Xem/Thêm/Xóa/Sửa giới hạn	ADMIN

	thiết lập	đăng ký của sinh viên.	
--	-----------	------------------------	--

Bảng 2. Danh sách các Use case

3.2.4 Phiên bản công nghệ sử dụng

NodeJS (Phiên bản 16.15.0), ExpressJS (Phiên bản 4.18.1), MongoDB (Phiên bản 6.3.4), PNPM (Phiên bản 7.1.0), ReactJS (Phiên bản 17.0.2), Redux (Phiên bản 7.2.6), TypeScript (Phiên bản 4.6.3), ViteJS (Phiên bản 2.9.7), MaterialUI (Phiên bản 5.5.1)

3.3 Yêu cầu phi chức năng

3.3.1 Yêu cầu về Hiệu suất

Dữ liệu được cập nhật nhanh chóng, nhất quán toàn vẹn.

Thông tin tiếp nhận được nhanh chóng hiệu quả thông qua website.

Giao diện thân thiện, dễ sử dụng, tốc độ phản hồi nhanh.

Có khả năng phục vụ một lượng sinh viên nhất định truy cập cùng một lúc.

3.3.2 Yêu cầu về Bảo mật

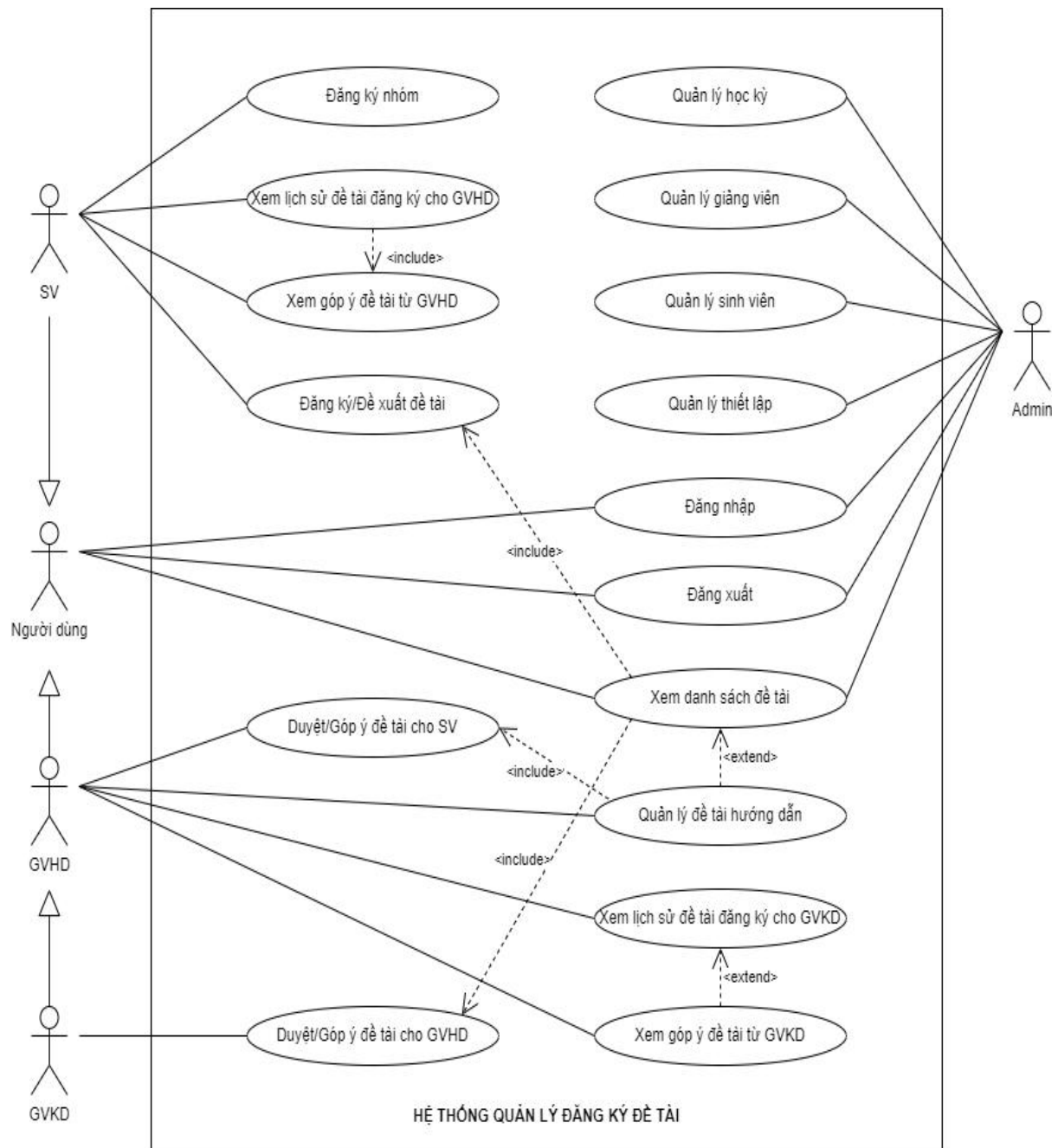
Phân quyền chặt chẽ. Các chức năng Xem, Thêm, Sửa, Xóa được thiết kế độc lập để linh hoạt hơn trong việc phân quyền, kiểm soát dữ liệu, an toàn và bảo mật.

3.3.3 Yêu cầu về Khả năng mở rộng

Xây dựng trên nền tảng web và tương thích được với nhiều loại trình duyệt để sử dụng mọi lúc mọi nơi. Khả năng tái sử dụng phần mềm cho một số mục đích quản lý thông tin khác rất mạnh mẽ. Dễ dàng bảo trì, nâng cấp, phát triển phần mềm sau khi đã triển khai.

CHƯƠNG IV: THIẾT KẾ HỆ THỐNG

4.1 Sơ đồ use case Tổng quát hệ thống



Hình 13. Sơ đồ use case Tổng quát hệ thống

4.2 Đặc tả use case

4.2.1 Đặc tả use case Đăng nhập

Use case	Đăng nhập	
ID	1	
Mô tả chi tiết	Khi người dùng truy cập vào hệ thống, hệ thống sẽ kiểm tra token, nếu không có nghĩa là người dùng chưa đăng nhập thì người dùng sẽ được chuyển đến trang đăng nhập để thực hiện việc đăng nhập rồi mới có thể vào sử dụng hệ thống, nếu đã có thì người dùng được vào thẳng hệ thống.	
Sự kiện kích hoạt	Khi người dùng truy cập vào hệ thống mà chưa đăng nhập.	
Tác nhân	Tất cả	
Điều kiện trước	Người dùng chưa đăng nhập. Nếu người dùng đăng nhập bằng Google thì tài khoản phải được đăng ký trước trên Google. Nếu người dùng đăng nhập bằng tài khoản HT thì tài khoản phải tồn tại trong hệ thống.	
Điều kiện sau	Người dùng đăng nhập thành công sẽ được điều hướng đến trang dashboard theo phân quyền của tài khoản.	
Luồng sự kiện	Người dùng	Hệ thống
	1.1 Người dùng chọn hình thức đăng nhập bằng Google. 1.2 Người dùng nhập thông tin theo các bước trên modal.	1.1 Hệ thống hiển thị modal xác thực của Google. 1.2 Hệ thống xác nhận tài khoản là hợp lệ khi Google trả về hợp lệ kèm điều kiện đuôi mail @student.tdtu.edu.vn hoặc @tdtu.edu.vn. Khi được hệ thống xác nhận hợp lệ sẽ điều hướng người dùng đến trang dashboard theo phân

	<p>2.1 Người dùng chọn hình thức đăng nhập bằng tài khoản HT.</p> <p>2.2 Người dùng nhập thông tin tài khoản hệ thống và nhấn nút Đăng nhập</p>	<p>quyền.</p> <p>2.1 Kiểm tra tính hợp lệ của các thông tin trong form. Nếu hợp lệ về đuôi mail @tdtu.edu.vn thì gửi request đăng nhập kèm email và password đã được hash. Nếu không hợp lệ, thông báo yêu cầu điền lại form.</p> <p>2.2 Kiểm tra email và password có tồn tại trong CSDL hay không. Nếu có tồn tại thì tạo token cho người dùng và điều hướng người dùng đến trang dashboard theo phân quyền, nếu không tồn tại thì thông báo đăng nhập không thành công.</p>
Ngoại lệ (nếu có)	<p>1.1 Thông tin tài khoản không hợp lệ sẽ thông báo lỗi.</p> <p>1.2 Phiên đăng nhập hết hạn thì cần đăng nhập lại để tạo mới token.</p>	

Bảng 3. Đặc tả use case Đăng nhập

4.2.2 Đặc tả use case Đăng xuất

Use case	Đăng xuất
ID	2
Mô tả chi tiết	Người dùng đăng xuất tài khoản khỏi hệ thống. Hệ thống tiến hành xóa token của người dùng và điều hướng về trang đăng nhập.

Sự kiện kích hoạt	Người dùng nhấn biểu tượng đăng xuất trong thanh điều hướng bên trái.	
Tác nhân	Tất cả	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Đăng xuất thành công.	
Luồng sự kiện	Người dùng	Hệ thống
	1.1 Người dùng nhấn vào biểu tượng đăng xuất. 1.2 Người dùng xác nhận đồng ý đăng xuất khỏi hệ thống.	1.1 Hệ thống hiển thị popup xác nhận đăng xuất. 1.2 Hệ thống sẽ xóa token và điều hướng về trang đăng nhập.
Ngoại lệ (nếu có)	-	

Bảng 4. Đặc tả use case Đăng xuất

4.2.3 Đặc tả use case Xem danh sách đề tài

Use case	Xem danh sách đề tài	
ID	3	
Mô tả chi tiết	Người dùng vào mục đăng ký đề tài, hệ thống sẽ lấy ra danh sách đề tài với học kỳ được chọn. Mặc định ban đầu sẽ là học kỳ mới nhất được tạo. Cho phép xem chi tiết đề tài được chọn.	
Sự kiện kích hoạt	Người dùng chọn Đăng ký đề tài trong thanh điều hướng.	
Tác nhân	Tất cả	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem được danh sách học kỳ theo học kỳ được chọn. Xem được chi tiết đề tài được chọn.	
Luồng sự kiện	Người dùng	Hệ thống

	<p>1.1 Người dùng vào mục đăng ký đề tài trong thanh điều hướng.</p> <p>1.2 Người dùng nhấn vào đề tài để xem chi tiết đề tài.</p>	<p>1.1 Hệ thống lấy ra danh sách đề tài với học kỳ được chọn, mặc định học kỳ mới nhất, sau đó sẽ căn cứ vào chọn lựa của người dùng và lấy ra đề tài theo học kỳ được chọn.</p> <p>1.2 Hệ thống hiển thị modal chi tiết đề tài được chọn.</p>
Ngoại lệ (nếu có)	-	

Bảng 5. Đặc tả use case Xem danh sách đề tài

4.2.4 Đặc tả use case Đăng ký/Đề xuất đề tài

Use case	Đăng ký/Đề xuất đề tài	
ID	4	
Mô tả chi tiết	Sinh viên đăng ký đề tài từ danh sách đề tài hoặc tự đề xuất đề tài theo nguyện vọng của bản thân.	
Sự kiện kích hoạt	Sinh viên nhấn vào biểu tượng đăng ký đề tài hoặc nút đề xuất đề tài trong dashboard đăng ký đề tài.	
Tác nhân	SV	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Đăng ký đề tài hoặc đề xuất đề tài thành công.	
Luồng sự kiện	Sinh viên	Hệ thống
	1.1 Sinh viên nhấn biểu tượng đăng ký đề tài.	1.1 Hệ thống hiển thị modal điền thông tin form đăng ký đề tài, với các thông tin có sẵn từ danh sách đề tài sẽ được hiển thị ngay trong modal này.
	1.2 Sinh viên điền thông tin	1.2 Hệ thống kiểm tra điều

	sinh viên và nhấn submit.	kiện các trường thông tin bắt buộc hợp lệ sẽ tiến hành ghi nhận thông tin đăng ký đề tài và thông báo đăng ký đề tài thành công.
	2.1 Sinh viên nhấn nút đề xuất đề tài.	2.1 Hệ thống hiển thị modal điền thông tin form đề xuất đề tài.
	2.2 Sinh viên điền thông tin sinh viên và nhấn submit.	2.2 Hệ thống kiểm tra điều kiện các trường thông tin bắt buộc hợp lệ sẽ tiến hành ghi nhận thông tin đề xuất đề tài và thông báo đề xuất đề tài thành công.
Ngoại lệ (nếu có)	-	

Bảng 6. Đặc tả use case Đăng ký/Đề xuất đề tài

4.2.5 Đặc tả use case Xem lịch sử đề tài đăng ký cho GVHD

Use case	Xem lịch sử đề tài đăng ký cho GVHD	
ID	5	
Mô tả chi tiết	Khi sinh viên đăng ký hoặc đề xuất đề tài sẽ vào xem lịch sử kết quả đề tài của chính mình.	
Sự kiện kích hoạt	Sinh viên nhấn vào mục xem lịch sử đề tài đăng ký.	
Tác nhân	SV	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem được danh sách lịch sử các đề tài đã đăng ký/đề xuất.	
Luồng sự kiện	Sinh viên	Hệ thống

	<p>1.1 Sinh viên nhấn vào mục xem lịch sử đề tài đăng ký.</p> <p>1.2 Sinh viên xem lịch sử danh sách đề tài đã đăng ký/đề xuất để nắm được trạng thái của đề tài.</p>	<p>1.1 Hệ thống sẽ lấy danh sách các đăng ký/đề xuất theo id sinh viên và hiển thị.</p> <p>1.2 Hệ thống sẽ cho biết trạng thái khi đăng ký/đề xuất thành công hay thất bại. Những đề tài có trạng thái thành công sẽ chờ GVHD/GVKD duyệt đề tài. Nếu đồng ý duyệt, trạng thái được đổi thành đã duyệt, ngược lại là đã từ chối duyệt.</p>
Ngoại lệ (nếu có)	-	

Bảng 7. Đặc tả use case Xem lịch sử đề tài đăng ký cho GVHD

4.2.6 Đặc tả use case Xem lịch sử đề tài đăng ký cho GVKD

Use case	Xem lịch sử đề tài đăng ký cho GVKD	
ID	6	
Mô tả chi tiết	Xem lịch sử đề tài đăng ký của GVHD cho GVKD để nhằm mục đích xem trạng thái đề tài của GVHD khi tạo đề tài có được chấp thuận để đưa vào danh sách đề tài sẽ công bố khi mở đợt đăng ký.	
Sự kiện kích hoạt	GV nhấn vào mục Xem lịch sử đề tài đăng ký cho GVKD.	
Tác nhân	GVHD, GVKD	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem được danh sách lịch sử các đề tài đã tạo.	
Luồng sự kiện	Giảng viên	Hệ thống
	1.1 GV nhấn vào mục xem lịch sử đề tài đăng ký.	1.1 Hệ thống sẽ lấy danh sách các đề tài được tạo theo id GV.

	1.2 GV xem lịch sử danh sách đề tài đã tạo để nắm được trạng thái của đề tài.	1.2 Hệ thống sẽ cho biết trạng thái đề tài được đồng ý hay bị từ chối duyệt.
Ngoại lệ (nếu có)	-	

Bảng 8. Đặc tả use case Xem lịch sử đề tài đăng ký cho GVKD

4.2.7 Đặc tả use case Quản lý đề tài hướng dẫn

Use case	Quản lý đề tài hướng dẫn	
ID	7	
Mô tả chi tiết	Trước khi công bố danh sách đề tài cho SV đăng ký, GVHD sẽ tạo đề tài. Khi tạo xong sẽ chờ GVKD duyệt đề tài được tạo để đưa vào danh sách đề tài. Có thể sửa hoặc xóa đề tài của chính mình.	
Sự kiện kích hoạt	Xem danh sách đề tài đăng ký	
Tác nhân	GVHD, GVKD	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Tạo mới, sửa, xóa, xem chi tiết đề tài của bản thân.	
Luồng sự kiện	Giảng viên	Hệ thống
	1.1 GV điền thông tin đề tài và nhấn tạo đề tài.	1.1 Hệ thống ghi nhận thông tin đề tài được tạo mới và thông báo tạo mới đề tài thành công. Nếu là GVHD thì đề tài sẽ vào trạng thái chờ GVKD duyệt. là GVKD thì đề tài được tự động duyệt ngay.
	1.2 GV chỉnh sửa đề tài và nhấn cập nhật đề tài.	1.2 Hệ thống ghi nhận thông tin đề tài cập nhật theo id và thông báo cập nhật đề tài thành

	1.3 GV nhấn xóa đề tài và nhấn xác nhận xóa.	công. 1.3 Hệ thống ghi nhận thông tin đề tài được xóa theo id và thông báo xóa đề tài thành công.
Ngoại lệ (nếu có)	-	

Bảng 9. Đặc tả use case Quản lý đề tài hướng dẫn

4.2.8 Đặc tả use case Duyệt/Góp ý đề tài

Use case	Duyệt/Góp ý đề tài	
ID	8-9	
Mô tả chi tiết	<p>Khi đề tài được duyệt bởi GVKD thì đề tài đó được đưa vào danh sách đề tài. Khi đó, đề tài có sinh viên đăng ký thì GVHD có quyền duyệt đề tài hoặc góp ý đề tài đến cho SV.</p> <p>Muốn duyệt đề tài, GV cần vào chi tiết đề tài đó và nhấn Đồng ý hoặc từ chối duyệt. Muốn góp ý đề tài, GV nhấn vào biểu tượng góp ý và nhập nội dung cần góp ý.</p>	
Sự kiện kích hoạt	<p>Nhấn biểu tượng góp ý trong danh sách đề tài.</p> <p>Nhấn biểu tượng Đồng ý/Từ chối duyệt trong chi tiết đề tài.</p>	
Tác nhân	GVHD, GVKD	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Duyệt/Góp ý được đề tài cho SV.	
Luồng sự kiện	Giảng viên	Hệ thống
	<p>1.1 GV nhấn xem chi tiết đề tài.</p> <p>1.2 GV nhấn nút đồng ý/từ chối duyệt.</p>	<p>1.1 Hệ thống hiển thị modal chi tiết đề tài được chọn.</p> <p>1.2 Hệ thống ghi nhận thông tin người duyệt đề tài, đánh</p>

	<p>2.1 GV nhấn biểu tượng góp ý</p> <p>2.2 GV điền nội dung góp ý và nhấn submit.</p>	<p>dấu đề tài đó là đồng ý/từ chối duyệt và thông báo trạng thái duyệt tương ứng.</p> <p>2.1 Hệ thống hiển thị modal nhập nội dung góp ý.</p> <p>2.2 Hệ thống ghi nhận nội dung góp ý và thông báo góp ý thành công.</p>
Ngoại lệ (nếu có)	Đợt đăng ký đang mở thì chức năng của GVKD sẽ đóng.	

Bảng 10. Đặc tả use case Duyệt/Góp ý đề tài

4.2.9 Đặc tả use case Xem góp ý đề tài

Use case	Xem góp ý đề tài	
ID	10-11	
Mô tả chi tiết	SV xem góp ý từ GVHD, GVHD xem góp ý từ GVKD.	
Sự kiện kích hoạt	Nhấn vào đề tài trong lịch sử đề tài.	
Tác nhân	SV, GVHD	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem được nội dung góp ý.	
Luồng sự kiện	Sinh viên/Giảng viên	Hệ thống
	1.1 SV nhấn vào đề tài trong lịch sử đề tài đăng ký.	1.1 Hệ thống hiển thị modal chứa nội dung góp ý từ GVHD.
	1.2 GVHD nhấn vào đề tài trong lịch sử đề tài đăng ký.	1.2 Hệ thống hiển thị modal chứa nội dung góp ý từ GVKD.
Ngoại lệ (nếu có)	Đợt đăng ký đang mở thì chức năng của GVKD sẽ đóng.	

Bảng 11. Đặc tả use case Xem góp ý đề tài

4.2.10 Đặc tả use case Đăng ký nhóm

Use case	Đăng ký nhóm	
ID	12	
Mô tả chi tiết	Để tạo mối liên kết ràng buộc, sinh viên cần đăng ký nhóm, có thể đăng ký tối đa nhóm 2 sinh viên. Sinh viên sẽ không được đăng ký sang nhóm khác nếu chưa thoát khỏi nhóm hiện tại.	
Sự kiện kích hoạt	Sinh viên nhấn vào mục Đăng ký nhóm	
Tác nhân	SV	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Tạo nhóm sinh viên thành công.	
Luồng sự kiện	Sinh viên	Hệ thống
	1.1 Sinh viên nhấn vào mục Đăng ký nhóm. 1.2 Sinh viên điền thông tin của bản thân, thông tin của thành viên nhóm nếu có và nhấn submit.	1.1 Hệ thống hiển thị modal điền thông tin nhóm sinh viên. 1.2 Hệ thống ghi nhận thông tin nhóm sinh viên được tạo và thông báo tạo nhóm sinh viên thành công.
Ngoại lệ (nếu có)	-	

Bảng 12. Đặc tả use case Đăng ký nhóm

4.2.11 Đặc tả use case Quản lý học kỳ

Use case	Quản lý học kỳ
ID	13
Mô tả chi tiết	Cán bộ văn phòng Khoa sẽ quản lý học kỳ cho các đợt đăng ký đề tài. Có quyền tạo mới, sửa, xóa học kỳ. Học kỳ sẽ như cây

	phân nhánh, danh sách đề tài, sinh viên sẽ căn cứ theo học kỳ để hiển thị thông tin phù hợp.	
Sự kiện kích hoạt	Admin nhấn vào mục Quản lý học kỳ.	
Tác nhân	ADMIN	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem, tạo mới, sửa và xóa được học kỳ.	
Luồng sự kiện	Admin	Hệ thống
	1.1 Admin truy cập vào mục Quản lý học kỳ.	1.1 Hệ thống lấy danh sách học kỳ và hiển thị.
	2.1 Admin nhấn nút tạo mới học kỳ.	2.1 Hệ thống hiển thị modal điền thông tin tạo mới học kỳ.
	2.2 Admin điền thông tin tạo mới học kỳ và nhấn submit.	2.2 Hệ thống ghi nhận thông tin học kỳ được tạo mới và thông báo tạo mới học kỳ thành công.
	3.1 Admin nhấn nút sửa thông tin học kỳ.	3.1 Hệ thống lấy ra thông tin học kỳ được chọn và hiển thị modal sửa thông tin học kỳ chứa thông tin sẵn có.
	3.2 Admin điền thông tin cập nhật học kỳ và nhấn submit.	3.2 Hệ thống ghi nhận thông tin học kỳ được cập nhật theo id và thông báo cập nhật học kỳ thành công.
	4.1 Admin nhấn nút xóa học kỳ.	4.1 Hệ thống lấy ra thông tin học kỳ được chọn và hiển thị modal xác nhận xóa học kỳ.
	4.2 Admin xác nhận thông tin học kỳ cần xóa và nhấn submit.	4.2 Hệ thống xóa học kỳ theo id và thông báo xóa học kỳ

		thành công.
Ngoại lệ (nếu có)	-	

Bảng 13. Đặc tả use case Quản lý học kỳ

4.2.12 Đặc tả use case Quản lý giảng viên

Use case	Quản lý giảng viên	
ID	14	
Mô tả chi tiết	Cán bộ văn phòng Khoa sẽ quản lý danh sách giảng viên. Có quyền tạo mới, sửa, xóa giảng viên. Import danh sách giảng viên theo file excel. Đánh dấu giảng viên nào là trưởng khoa/trưởng bộ môn.	
Sự kiện kích hoạt	Admin nhấn vào mục Quản lý giảng viên.	
Tác nhân	ADMIN	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem, tạo mới, sửa, xóa được giảng viên và import theo file danh sách giảng viên.	
Luồng sự kiện	Admin	Hệ thống
	1.1 Admin truy cập vào mục Quản lý giảng viên.	1.1 Hệ thống lấy danh sách giảng viên và hiển thị.
	2.1 Admin nhấn nút tạo mới giảng viên.	2.1 Hệ thống hiển thị modal điền thông tin tạo mới giảng viên.
	2.2 Admin điền thông tin tạo mới giảng viên và nhấn submit.	2.2 Hệ thống ghi nhận thông tin giảng viên được tạo mới và thông báo tạo mới giảng viên thành công.
	3.1 Admin nhấn nút sửa thông	3.1 Hệ thống lấy ra thông tin

	<p>tin giảng viên.</p> <p>3.2 Admin điền thông tin cập nhật giảng viên và nhấn submit.</p> <p>4.1 Admin nhấn nút xóa giảng viên.</p> <p>4.2 Admin xác nhận thông tin giảng viên cần xóa và nhấn submit.</p> <p>5.1 Admin nhấn nút import, chọn file cần import và nhấn submit.</p>	<p>giảng viên được chọn và hiển thị modal sửa thông tin giảng viên chứa thông tin sẵn có.</p> <p>3.2 Hệ thống ghi nhận thông tin giảng viên được cập nhật theo id và thông báo cập nhật giảng viên thành công.</p> <p>4.1 Hệ thống lấy ra thông tin giảng viên được chọn và hiển thị modal xác nhận xóa giảng viên.</p> <p>4.2 Hệ thống xóa giảng viên theo id và thông báo xóa giảng viên thành công.</p> <p>5.1 Hệ thống đọc file và ghi nhận danh sách giảng viên được tạo mới và thông báo import danh sách giảng viên thành công.</p>
Ngoại lệ (nếu có)	-	

Bảng 14. Đặc tả use case Quản lý giảng viên

4.2.13 Đặc tả use case Quản lý sinh viên

Use case	Quản lý sinh viên
ID	15
Mô tả chi tiết	Cán bộ văn phòng Khoa sẽ quản lý danh sách sinh viên theo từng học kỳ đã tạo. Có quyền tạo mới, sửa, xóa sinh viên. Import danh sách sinh viên theo file excel.
Sự kiện kích hoạt	Admin nhấn vào mục Quản lý sinh viên.

Tác nhân	ADMIN	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem, tạo mới, sửa, xóa được sinh viên và import theo file danh sách sinh viên theo học kỳ được chọn.	
Luồng sự kiện	Admin	Hệ thống
	1.1 Admin truy cập vào mục Quản lý sinh viên.	1.1 Hệ thống lấy danh sách sinh viên và hiển thị theo học kỳ mặc định.
	1.2 Admin chọn học kỳ.	1.2 Hệ thống lấy danh sách sinh viên và hiển thị theo học kỳ được chọn.
	2.1 Admin nhấn nút tạo mới sinh viên.	2.1 Hệ thống hiển thị modal điền thông tin tạo mới sinh viên.
	2.2 Admin điền thông tin tạo mới sinh viên và nhấn submit.	2.2 Hệ thống ghi nhận thông tin sinh viên được tạo mới trong học kỳ được chọn và thông báo tạo mới sinh viên thành công.
	3.1 Admin nhấn nút sửa thông tin sinh viên.	3.1 Hệ thống lấy ra thông tin sinh viên được chọn và hiển thị modal sửa thông tin sinh viên chứa thông tin sẵn có.
	3.2 Admin điền thông tin cập nhật sinh viên và nhấn submit.	3.2 Hệ thống ghi nhận thông tin sinh viên được cập nhật trong học kỳ được chọn theo id và thông báo cập nhật sinh viên thành công.
	4.1 Admin nhấn nút xóa sinh viên.	4.1 Hệ thống lấy ra thông tin sinh viên được chọn và hiển

	<p>4.2 Admin xác nhận thông tin sinh viên cần xóa và nhấn submit.</p> <p>5.1 Admin nhấn nút import, chọn file cần import và nhấn submit.</p>	<p>thị modal xác nhận xóa sinh viên.</p> <p>4.2 Hệ thống xóa sinh viên trong học kỳ được chọn theo id và thông báo xóa sinh viên thành công.</p> <p>5.1 Hệ thống đọc file và ghi nhận danh sách sinh viên được tạo mới trong học kỳ được chọn và thông báo import danh sách sinh viên thành công.</p>
Ngoại lệ (nếu có)	-	

Bảng 15. Đặc tả use case Quản lý sinh viên

4.2.14 Đặc tả use case Quản lý thiết lập

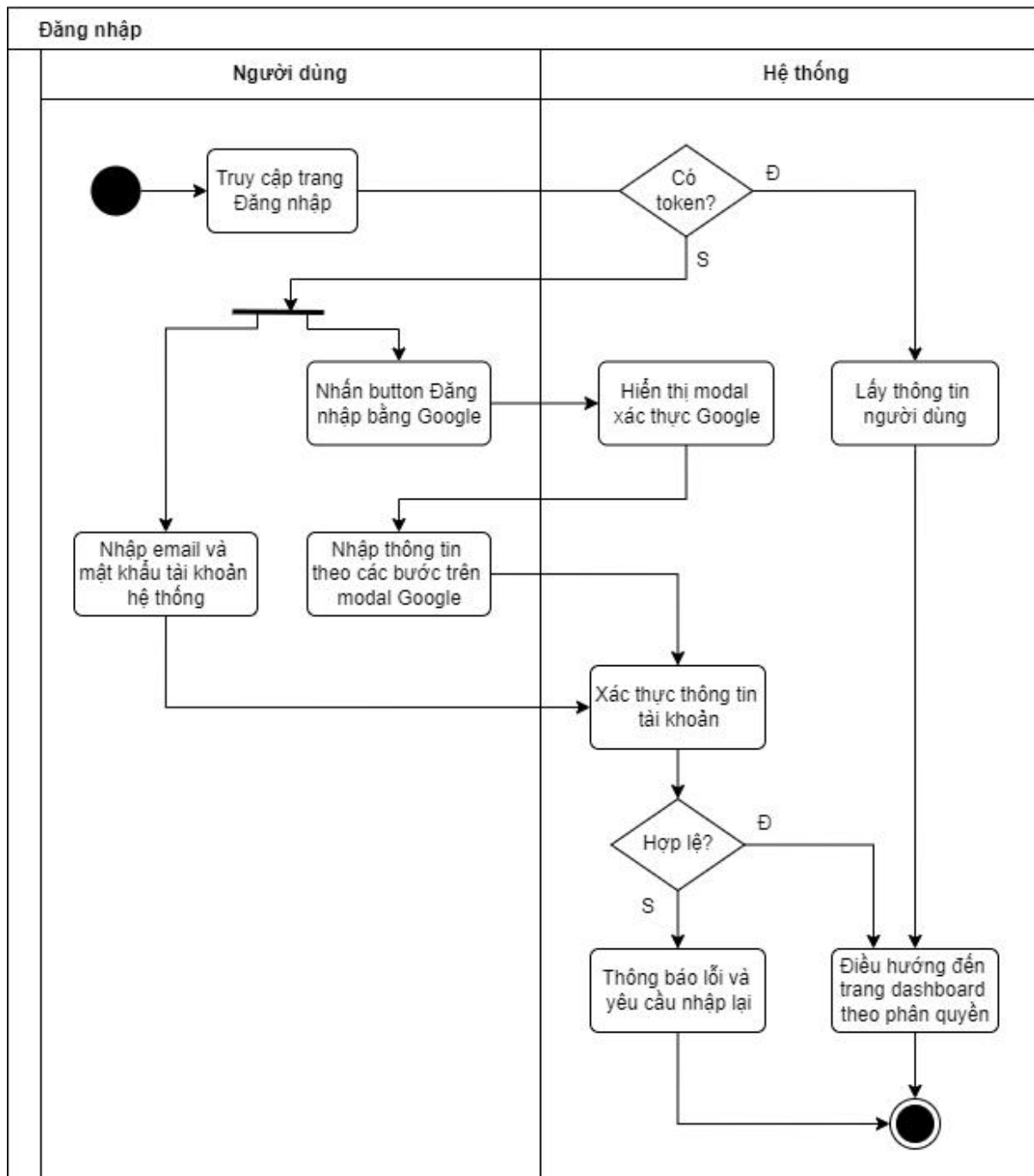
Use case	Quản lý thiết lập	
ID	16	
Mô tả chi tiết	Cán bộ văn phòng Khoa sẽ quản lý thiết lập để quản lý được số lượng tối đa mà mỗi sinh viên được phép đăng ký. Có quyền tạo mới, sửa, xóa giá trị thiết lập.	
Sự kiện kích hoạt	Admin nhấn vào mục Quản lý thiết lập.	
Tác nhân	ADMIN	
Điều kiện trước	Đăng nhập thành công.	
Điều kiện sau	Xem, tạo mới, sửa, xóa được giá trị giới hạn đăng ký của sinh viên.	
Luồng sự kiện	Admin	Hệ thống

	<p>1.1 Admin truy cập vào mục Quản lý thiết lập.</p> <p>2.1 Admin nhấn nút tạo mới giá trị giới hạn thiết lập.</p> <p>2.2 Admin điền giá trị giới hạn mới và nhấn submit.</p> <p>3.1 Admin nhấn nút sửa giá trị giới hạn.</p> <p>3.2 Admin điền giá trị giới hạn cần cập nhật và nhấn submit.</p> <p>4.1 Admin nhấn nút xóa thiết lập giá trị giới hạn.</p> <p>4.2 Admin xác nhận xóa thiết lập và nhấn submit.</p>	<p>1.1 Hệ thống lấy thông tin giá trị giới hạn đăng ký.</p> <p>2.1 Hệ thống hiển thị modal điền giá trị giới hạn tạo mới.</p> <p>2.2 Hệ thống ghi nhận giá trị giới hạn tạo mới và thông báo tạo mới sinh viên thành công.</p> <p>3.1 Hệ thống lấy ra thông tin thiết lập được chọn và hiển thị modal sửa giá trị giới hạn.</p> <p>3.2 Hệ thống ghi nhận giá trị giới hạn được cập nhật theo id và thông báo cập nhật giá trị giới hạn thành công.</p> <p>4.1 Hệ thống lấy ra thông tin thiết lập được chọn và hiển thị modal xác nhận xóa giá trị giới hạn.</p> <p>4.2 Hệ thống xóa thiết lập theo id và thông báo xóa thiết lập giá trị giới hạn thành công.</p>
Ngoại lệ (nếu có)	-	

Bảng 16. Đặc tả use case Quản lý thiết lập

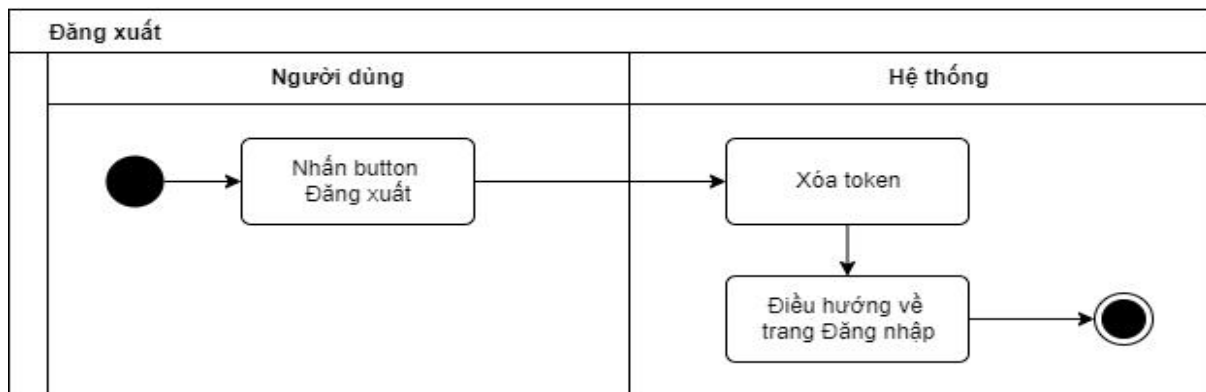
4.3 Sơ đồ hoạt động

4.3.1 Sơ đồ hoạt động Đăng nhập



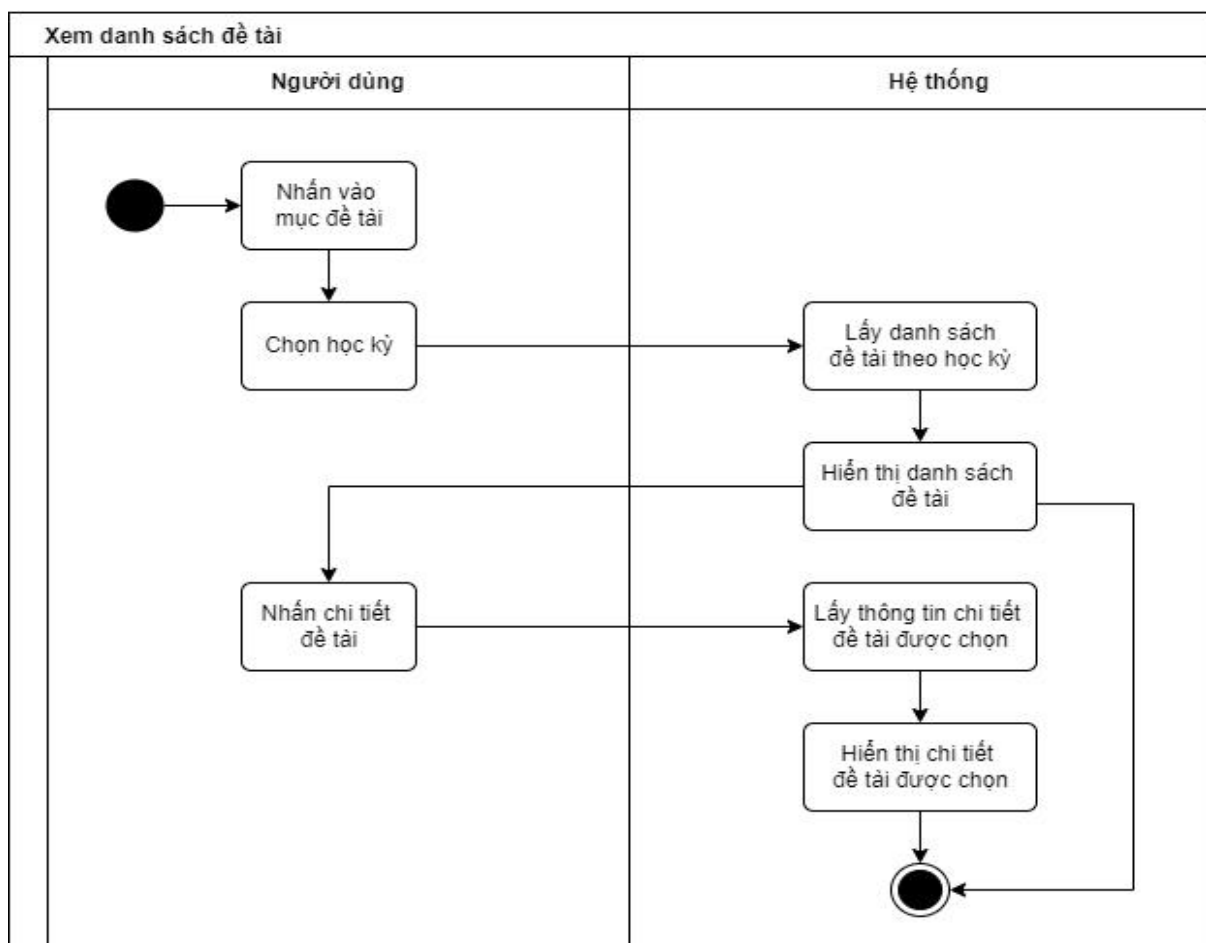
Hình 14. Sơ đồ hoạt động Đăng nhập

4.3.2 Sơ đồ hoạt động Đăng xuất



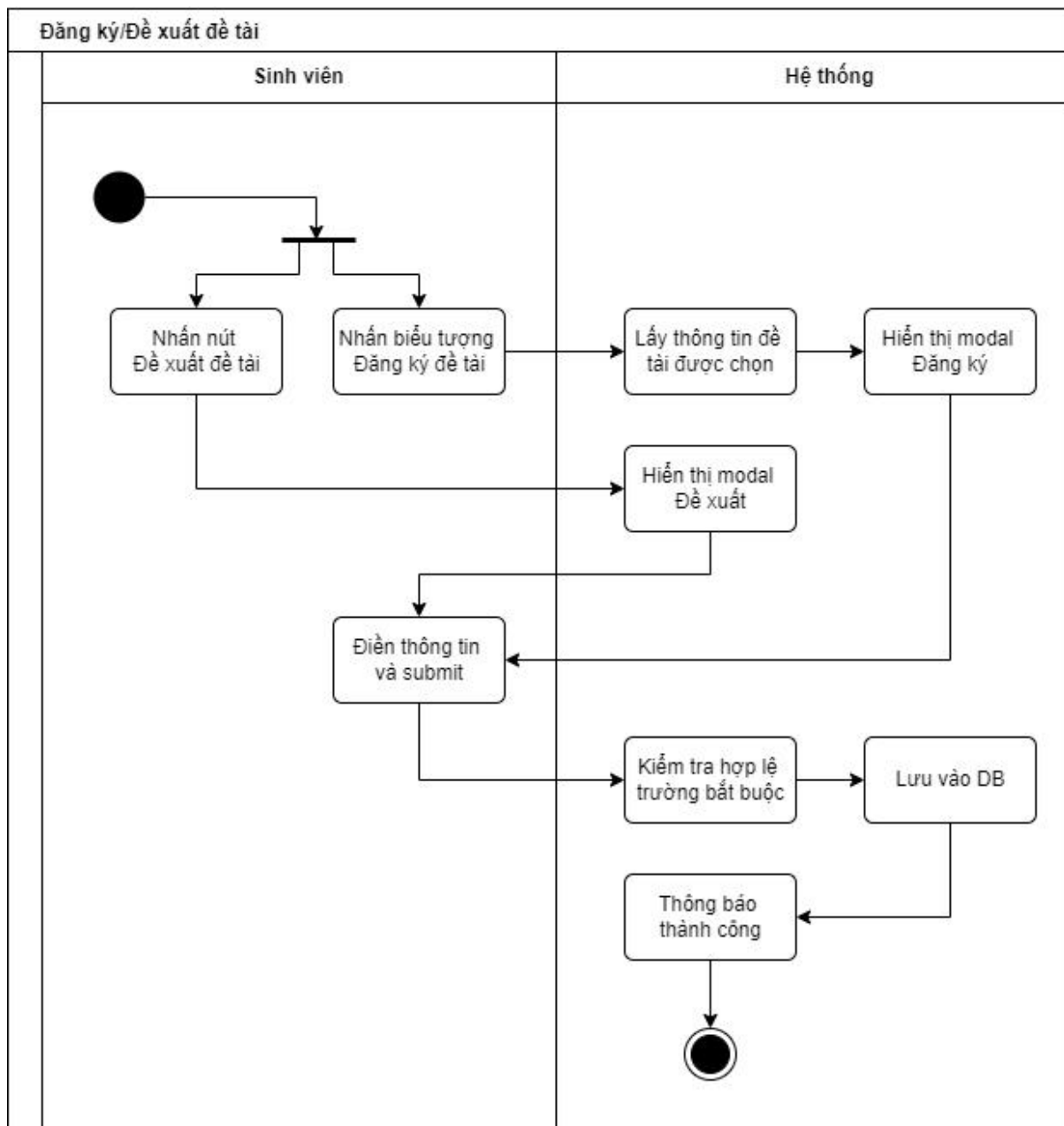
Hình 15. Sơ đồ hoạt động Đăng xuất

4.3.3 Sơ đồ hoạt động Xem danh sách đề tài



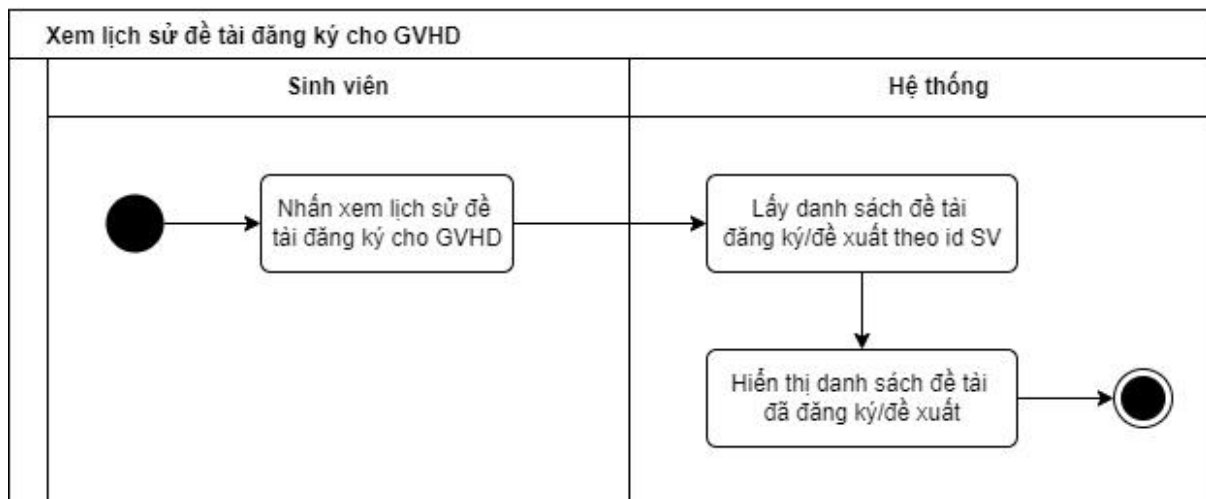
Hình 16. Sơ đồ hoạt động Xem danh sách đề tài

4.3.4 Sơ đồ hoạt động Đăng ký/Đề xuất đề tài



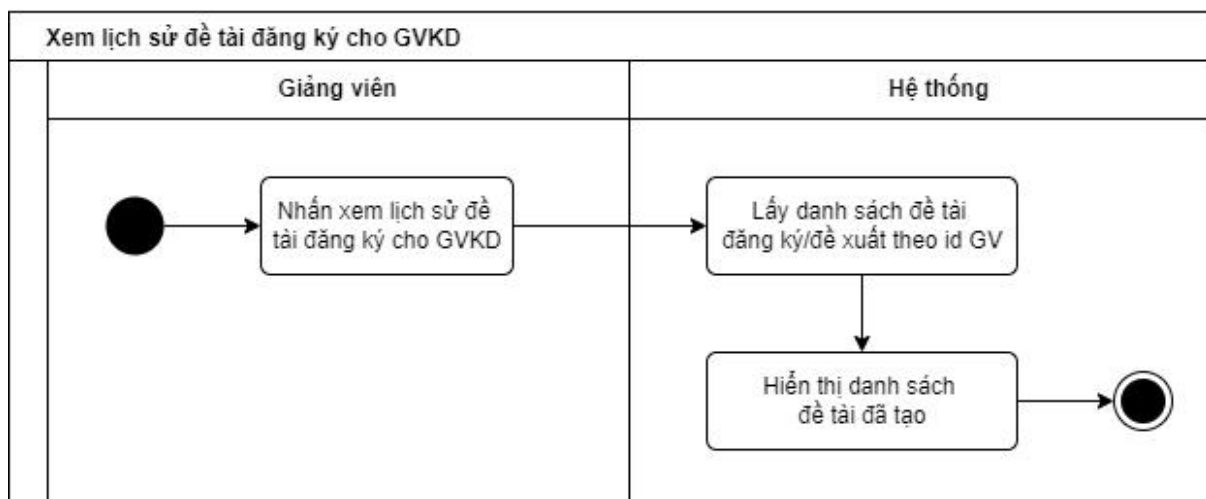
Hình 17. Sơ đồ hoạt động Đăng ký/Đề xuất đề tài

4.3.5 Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVHD



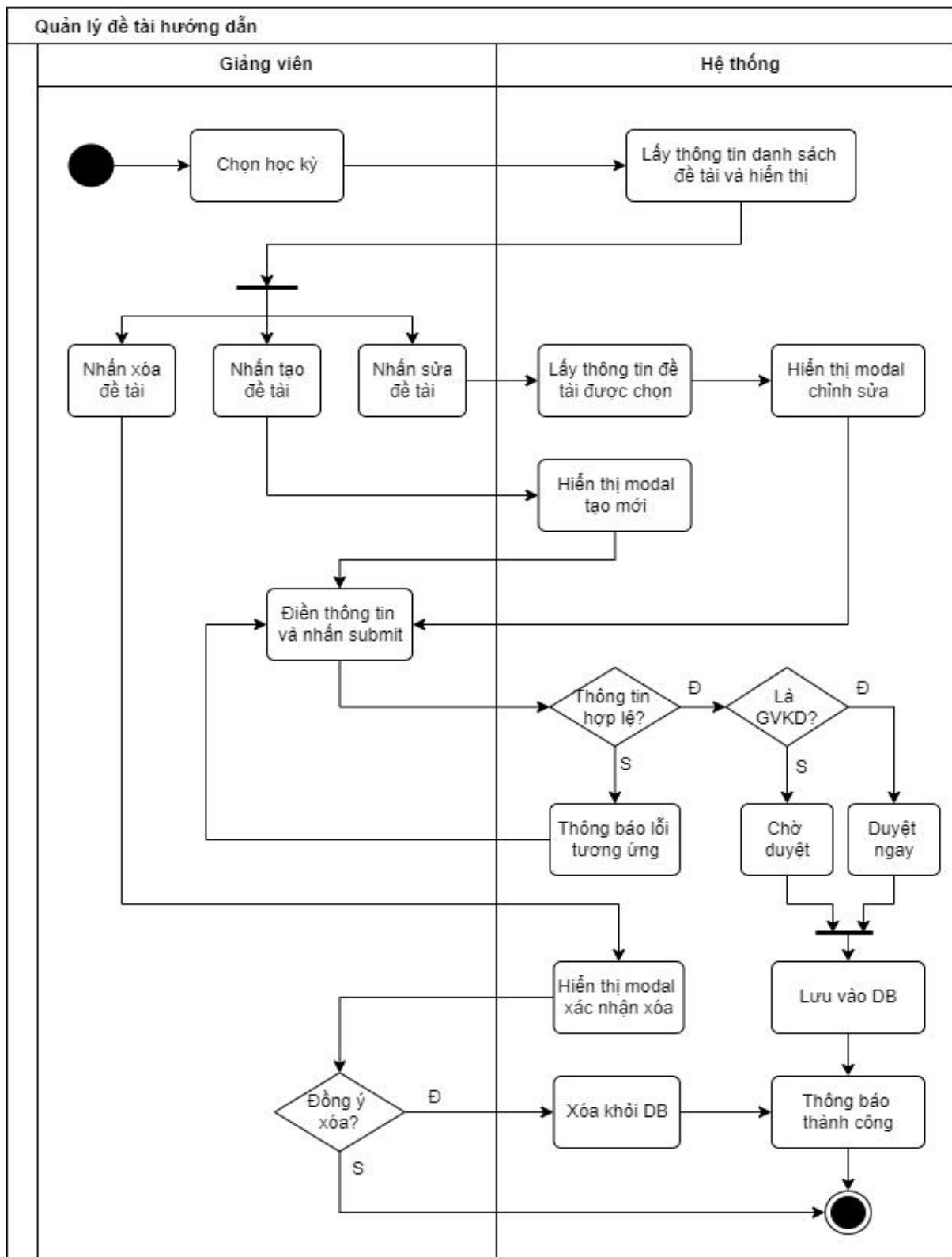
Hình 18. Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVHD

4.3.6 Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVKD



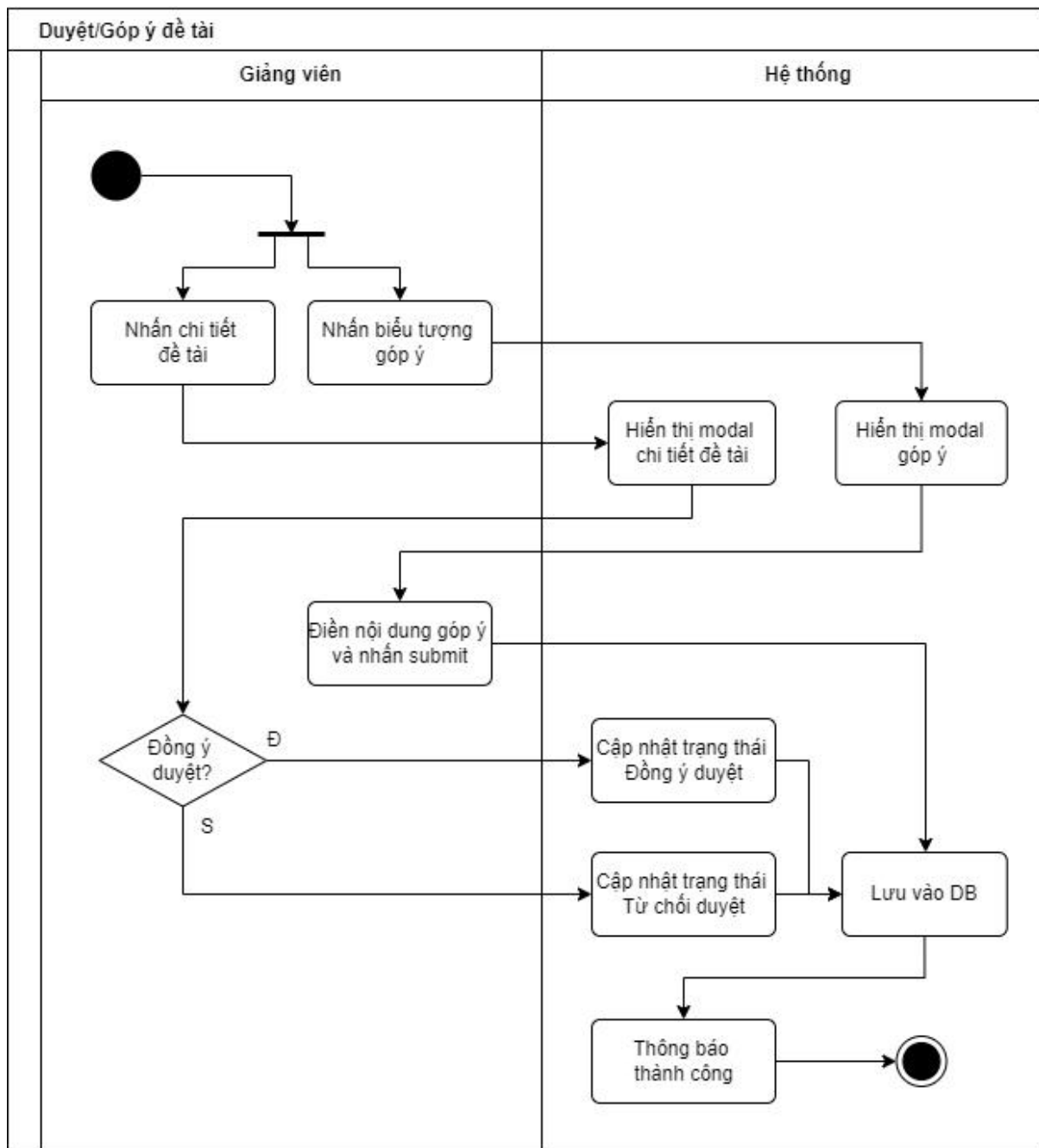
Hình 19. Sơ đồ hoạt động Xem lịch sử đề tài đăng ký cho GVKD

4.3.7 Sơ đồ hoạt động Quản lý đề tài hướng dẫn



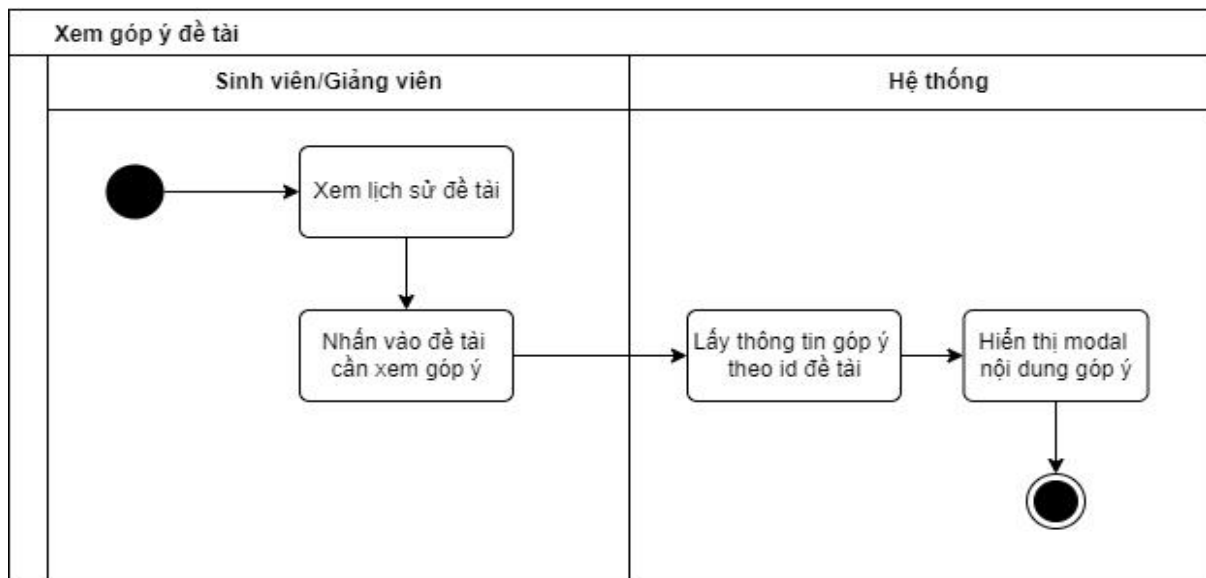
Hình 20. Sơ đồ hoạt động Quản lý đề tài hướng dẫn

4.3.8 Sơ đồ hoạt động Duyệt/Góp ý đề tài



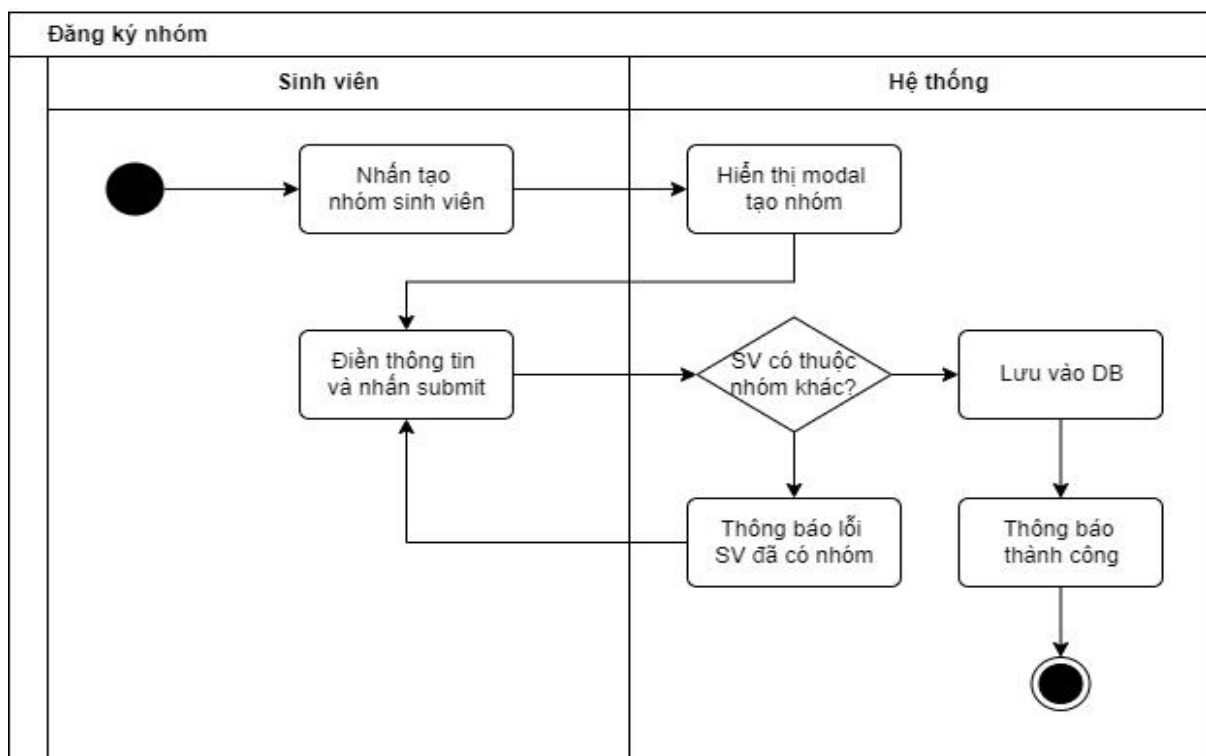
Hình 21. Sơ đồ hoạt động Duyệt/Góp ý đề tài

4.3.9 Sơ đồ hoạt động Xem góp ý đề tài



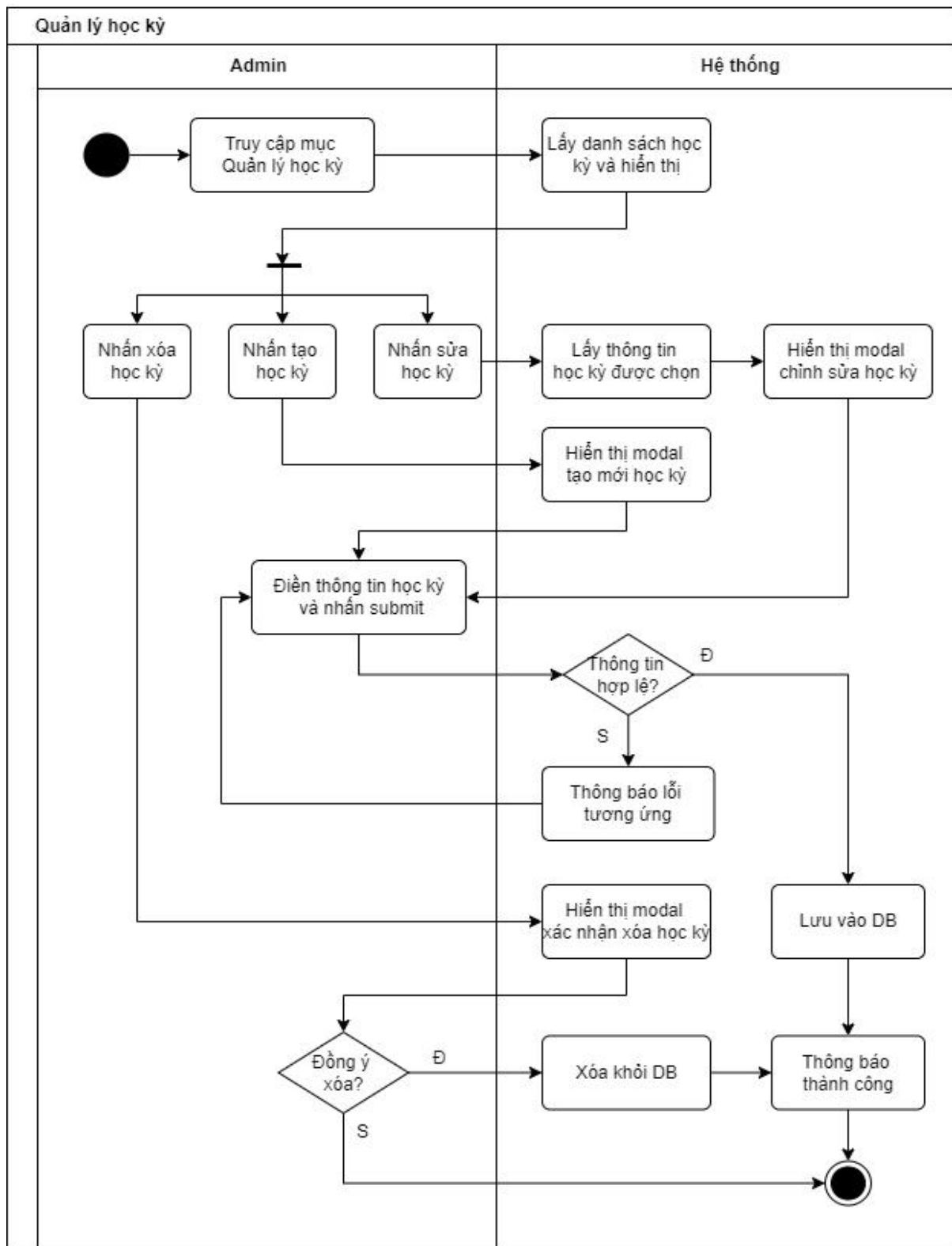
Hình 22. Sơ đồ hoạt động Xem góp ý đề tài

4.3.10 Sơ đồ hoạt động Đăng ký nhóm



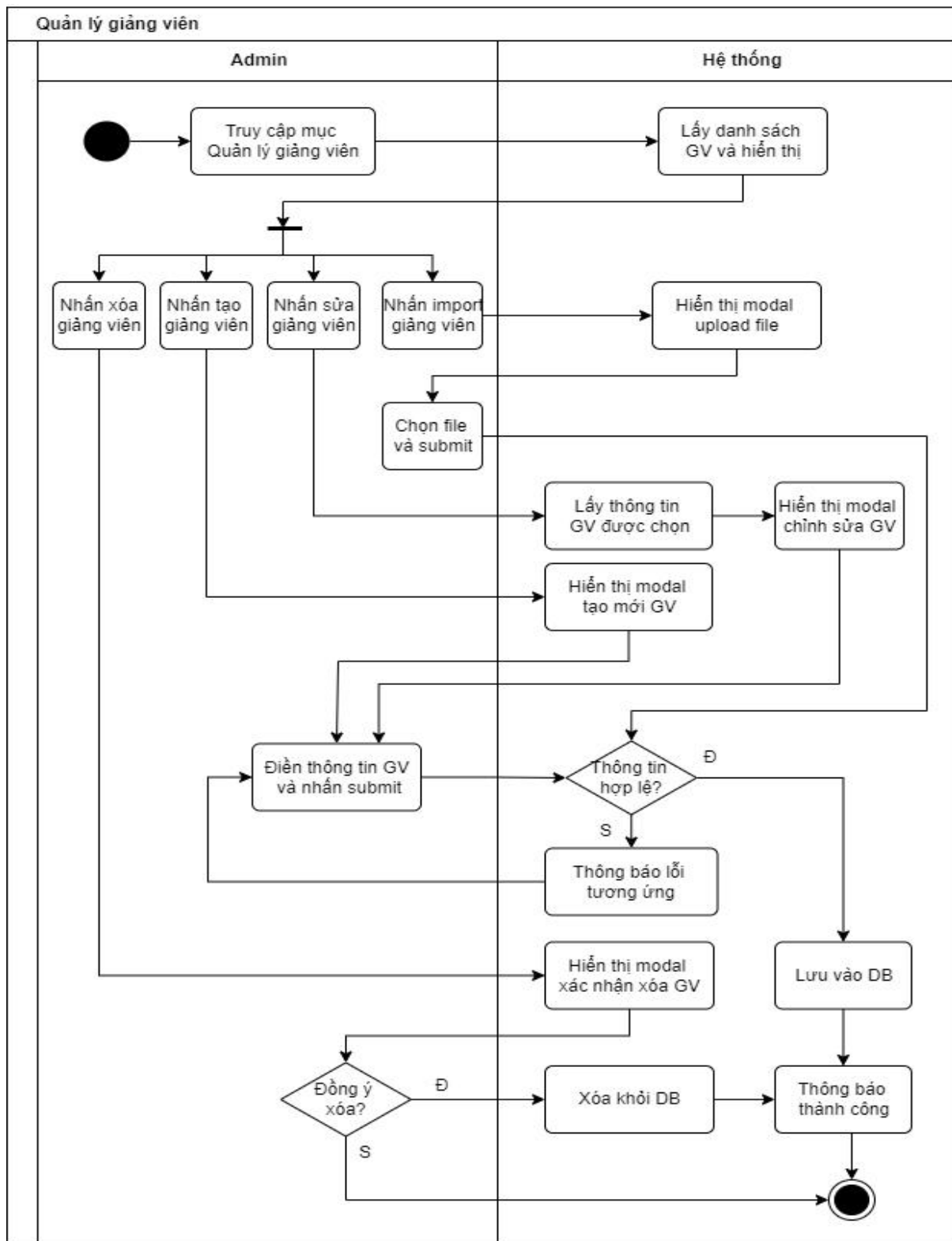
Hình 23. Sơ đồ hoạt động Đăng ký nhóm

4.3.11 Sơ đồ hoạt động Quản lý học kỳ



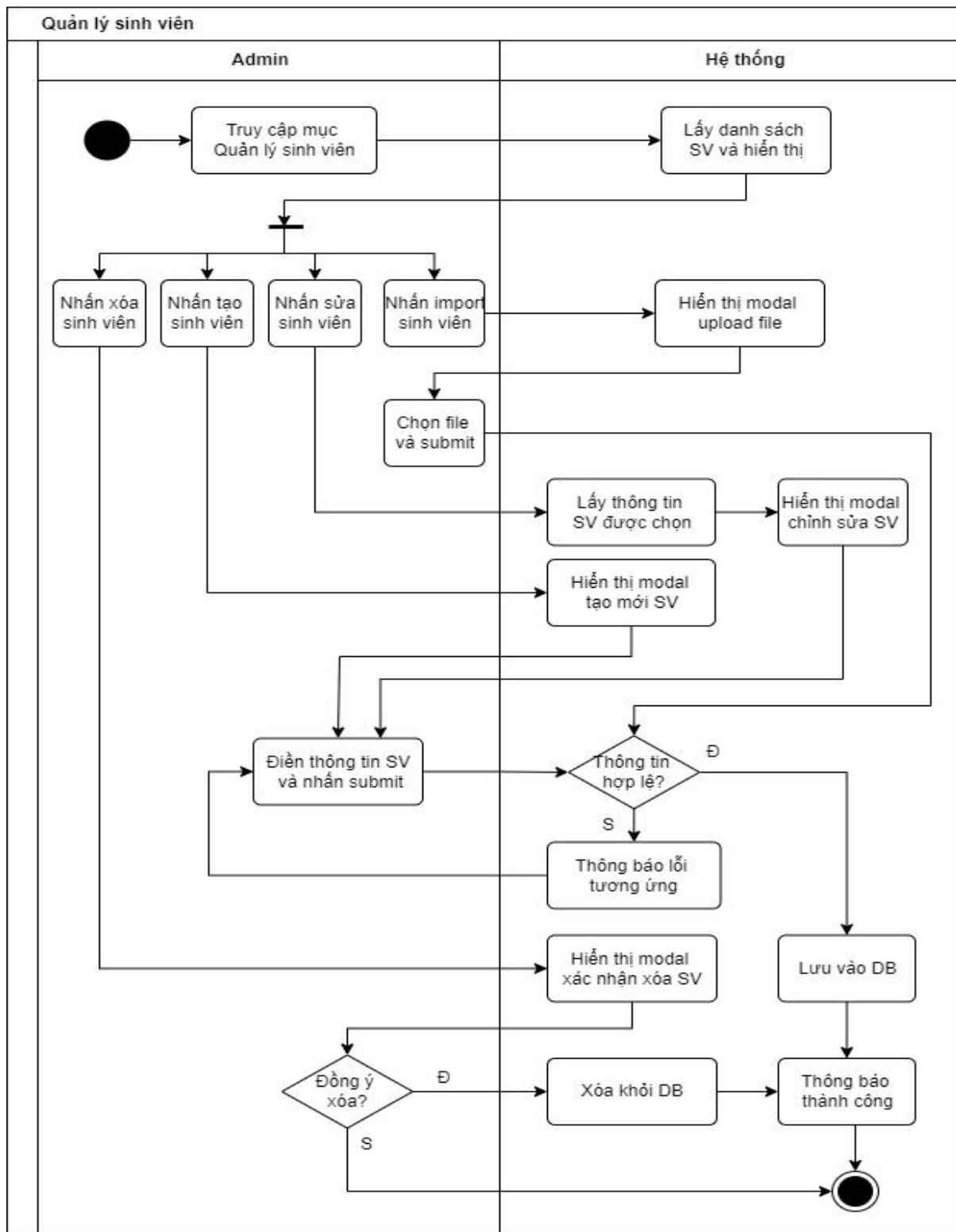
Hình 24. Sơ đồ hoạt động Quản lý học kỳ

4.3.12 Sơ đồ hoạt động Quản lý giảng viên



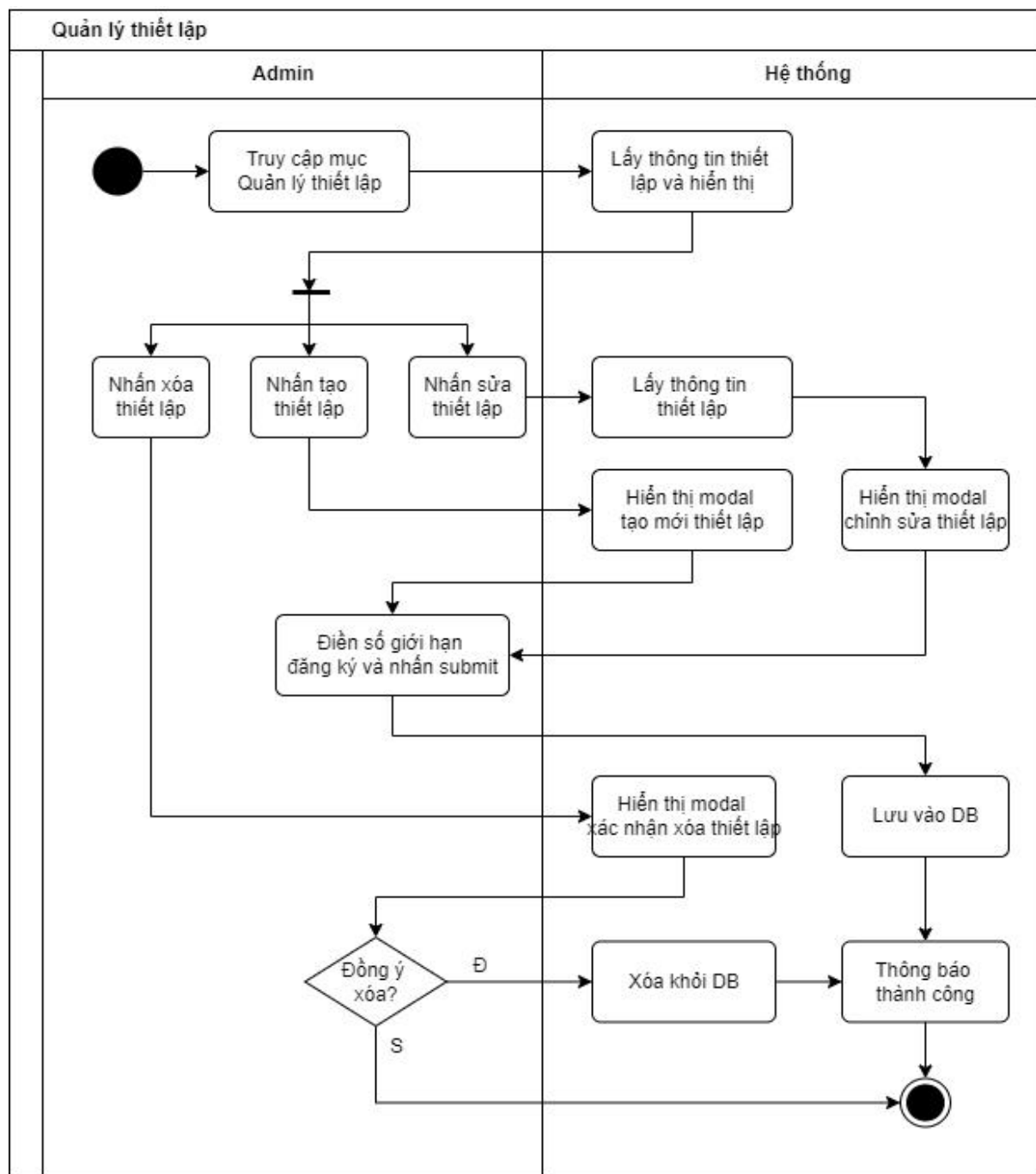
Hình 25. Sơ đồ hoạt động Quản lý giảng viên

4.3.13 Sơ đồ hoạt động Quản lý sinh viên



Hình 26. Sơ đồ hoạt động Quản lý sinh viên

4.3.14 Sơ đồ hoạt động Quản lý thiết lập

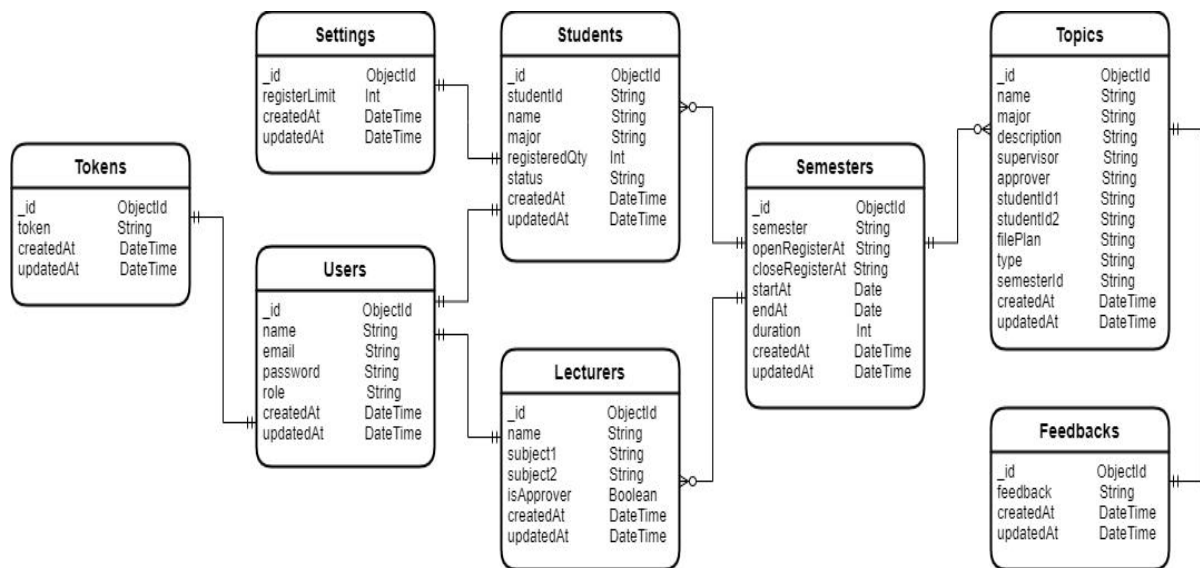


Hình 27. Sơ đồ hoạt động Quản lý thiết lập

4.4 Sơ đồ ERD

Sơ đồ ERD giúp mô tả được mối liên hệ giữa các bảng được rõ ràng hơn. Users (đại diện cho đối tượng người dùng) có thông tin lưu trữ chính là email, mật khẩu, phân quyền; Students (đại diện cho đối tượng sinh viên) có thông tin lưu trữ chính là mã

sinh viên, tên sinh viên, chuyên ngành, số lượng đăng ký, trạng thái; Lecturers (đại diện cho đối tượng giảng viên) có thông tin lưu trữ chính là tên, bộ môn chính, bộ môn phụ, đánh dấu người kiểm duyệt; Semesters (đại diện cho đối tượng học kỳ) có thông tin lưu trữ chính là tên học kỳ, ngày mở đăng ký, ngày đóng đăng ký, ngày bắt đầu thực hiện đề tài, ngày kết thúc đề tài, khoảng thời gian; Topics (đại diện cho đối tượng đề tài) có thông tin lưu trữ chính là tên đề tài, chuyên ngành, mô tả, giảng viên hướng dẫn, giảng viên kiểm duyệt, tên sinh viên, đường dẫn đến file kế hoạch, kiểu đề tài; Feedbacks (đại diện cho đối tượng góp ý) có thông tin lưu trữ là nội dung góp ý, Settings (đại diện cho đối tượng thiết lập) có thông tin lưu trữ là giá trị giới hạn Tokens (đại diện cho đối tượng mã thông báo) có thông tin lưu trữ là mã cấp quyền truy cập.



Hình 28. Sơ đồ ERD

4.5 Bản vẽ Wireframe

Sử dụng phương pháp vẽ phác thảo (Wireframe) để hình dung những gì cần làm. Nó là một bản vẽ tự do được thực hiện nhanh chóng những ý tưởng nảy ra tại thời điểm nhận yêu cầu. Trong quá trình phát triển hệ thống, thiết kế sẽ được thay đổi dần để hoàn thiện hơn. Đường dẫn đến bản vẽ:

<https://drive.google.com/drive/folders/1RZl3zz76Bi0OvwYcSZFW2d1VFIxVgwge?usp=sharing>

CHƯƠNG V: HIỆN THỰC HỆ THỐNG VÀ KẾT QUẢ

5.1 Danh sách API

STT	Tên chức năng	Đường dẫn	Phương thức
1	Đăng nhập bằng Tài khoản HT	/api/auth/sign-in	POST
2	Đăng nhập bằng Google	/api/auth/sign-in-google	POST
3	Lấy thông tin người dùng	/api/auth/profile	GET
4	Đăng xuất	/api/auth/sign-out	DELETE
5	Tạo mới người dùng	/api/auth/create	POST
6	Tạo mới token	/api/auth/token	POST
7	Lấy danh sách học kỳ	/api/semester	GET
8	Tạo mới học kỳ	/api/semester	POST
9	Cập nhật học kỳ theo id	/api/semester/{id}	PUT
10	Xóa học kỳ theo id	/api/semester/{id}	DELETE
11	Lấy danh sách giảng viên	/api/lecturer	GET
12	Tạo mới giảng viên	/api/lecturer	POST
13	Cập nhật giảng viên theo id	/api/lecturer/{id}	PUT
14	Xóa giảng viên theo id	/api/lecturer/{id}	DELETE
15	Lấy danh sách sinh viên	/api/student	GET
16	Tạo mới sinh viên	/api/student	POST
17	Cập nhật sinh viên theo id	/api/student/{id}	PUT
18	Xóa sinh viên theo id	/api/student/{id}	DELETE
19	Lấy thông tin thiết lập	/api/setting	GET

20	Tạo mới thiết lập	/api/setting	POST
21	Cập nhật thiết lập theo id	/api/setting/{id}	PUT
22	Xóa thiết lập theo id	/api/setting/{id}	DELETE
23	Lấy danh sách đề tài	/api/topic	GET
24	Tạo mới đề tài	/api/topic	POST
25	Cập nhật đề tài theo id	/api/topic/{id}	PUT
26	Xóa đề tài theo id	/api/topic/{id}	DELETE
27	Đồng ý duyệt đề tài theo id	/api/topic/approve/{id}	POST
28	Từ chối duyệt đề tài theo id	/api/topic/decline/{id}	POST
29	Góp ý đề tài theo id	/api/topic/feedback/{id}	POST
30	Xem góp ý đề tài theo id	/api/topic/feedback/{id}	GET
31	Đăng ký đề tài theo id	/api/topic/register/{id}	POST
32	Đề xuất đề tài	/api/topic/propose	POST
33	Đăng ký nhóm	/api/student/group	POST
34	Import danh sách sinh viên	/api/student/import	POST
35	Import danh sách giảng viên	/api/lecturer/import	POST

Bảng 17. Danh sách API

Đường dẫn đến tài liệu API trên postman:

<https://documenter.getpostman.com/view/12371540/Uz5GpGoe#intro>

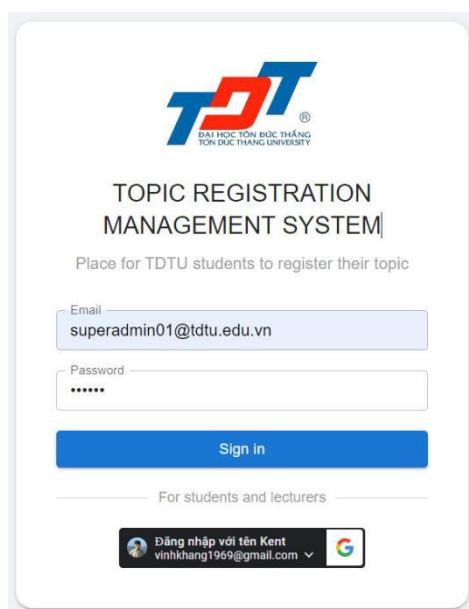
5.2 Kết quả đạt được

Đường dẫn hosting:

- Client: <https://topic-registration.netlify.app/>

- Server: <https://topic-registration-production.up.railway.app/>

5.2.1 Màn hình Đăng nhập



The screenshot shows the login interface for the TDTU Topic Registration Management System. At the top is the TDTU logo with the text "ĐẠI HỌC TÔN ĐỨC THẮNG TÔN DỤC THANG UNIVERSITY". Below the logo, the title "TOPIC REGISTRATION MANAGEMENT SYSTEM" is displayed, followed by the subtitle "Place for TDTU students to register their topic". The login form includes an "Email" field with the text "superadmin01@tdtu.edu.vn", a "Password" field with six dots, and a blue "Sign in" button. Below the button is the text "For students and lecturers". At the bottom, there is a "Đăng nhập với tên Kent" button with a profile icon and the email "vinhkhang1969@gmail.com", and a Google login button.

 Đăng nhập bằng Google

Chọn tài khoản

để tiếp tục sử dụng TDTU Topic Registration System

 Kent Vince
vinhkhang1969@gmail.com

 Tô Vinh Khang
51800408@student.tdtu.edu.vn

 Sử dụng một tài khoản khác

Hình 29. Màn hình Đăng nhập

5.2.2 Màn hình Xem danh sách đề tài

The screenshot displays the 'Registration Topic' interface. On the left, a sidebar shows the user's profile 'Tô Vinh Khang' and buttons for 'Registration Topic' and 'Submission History'. The main area features a 'Semester' dropdown set to '2A | 2021 - 2022' and a 'Propose' button. Below this is a table listing available topics for registration.

Name	Major	Description	Quantity	Supervisor	Status	Register
Build Chatbot using AI	Computer Science	AI Chatbot	2	Pham Van Huy	Open	Registered
Build a website using ASP.NET	Software Engineering	Requirements...	1	Mai Van Manh	Open	+
Design IOT environmental information tracking	Network Engineering	IOT	1	Le Anh Cuong	Open	+
Design customer care application	Computer Science	Design customer care application.	1	Tran Trung Tin	Open	+
Build a website fashion store management with MERN Stack	Software Engineering	Learn MERN Stack and develop a fashion store management website	2	Mai Van Manh	Open	+
Build a website hotel booking using Flutter	Software Engineering	Hotel Booking...	1	Doan Xuan Thanh	Open	+
Build a website manage football schedule	Software Engineering	Manage Football Field Reservation Schedule	1	Duong Huu Phuc	Open	+

At the bottom left, there is a 'Sign Out' button.

Hình 30. Màn hình Xem danh sách đề tài

5.2.3 Màn hình Đăng ký đề tài

This screenshot shows the same 'Registration Topic' interface as Figure 30, but with a 'Register Topic' modal window open in the center. The modal contains the following fields:

- Topic name: Build a website fashion store management with f
- Major: Software Engineering
- Description: Learn MERN Stack and develop a fashion store management website
- Supervisor: Mai Van Manh
- Student ID 1: Student ID 1
- Student 1: Student 1
- Student 1 Major: Student 1 Major

A 'Confirm' button is located at the bottom of the modal. The background table from Figure 30 is visible but dimmed.

Hình 31. Màn hình Đăng ký đề tài

5.2.4 Màn hình Đề xuất đề tài

Propose Topic

Topic name

Major

Description

Supervisor

Student ID 1

Student 1

Student 1 Major

Confirm

Hình 32. Màn hình Đề xuất đề tài


5.2.5 Màn hình Xem lịch sử đề tài đăng ký

Submission History

Name	Major	Description	Supervisor	Status	Semester	Type
Make an Auto Rawing Web Content with AI	Computer Science	Requirements...	Pham Van Huy	Pending	2A 2021 - 2022	Propose
Research AI & ML	Computer Science	Requirements...	Pham Van Huy	Success	2A 2021 - 2022	Register
Build an e-commerce website	Software Engineering	Build...	Mai Van Manh	Success	2A 2021 - 2022	Register
Build Master Data Management System	Software Engineering	Product...	Le Van Vang	Pending	2A 2021 - 2022	Propose
Build Master Data Management System	Software Engineering	Product Management	Le Van Vang	Pending	2A 2021 - 2022	Propose
Design logistics management in ERP	Software Engineering	ERP	Duong Huu Phuc	Pending	2A 2021 - 2022	Propose
Design CRM in life insurance	Computer Science	CRM	Pham Van Huy	Pending	2A 2021 - 2022	Propose
Build a website coffee house management	Software Engineering	NodeJS, ExpressJS, ReactJS, MongoDB	Mai Van Manh	Pending	2A 2021 - 2022	Propose

Hình 33. Màn hình Xem lịch sử đề tài đăng ký

5.2.6 Màn hình Quản lý đề tài hướng dẫn



Mai Van Manh

Registration Topic

Proposed Topic


Submission History

[Sign Out](#)

Semester
2A | 2021 - 2022

Create +

Name	Major	Description	Quantity	Supervisor	Status	Action
Build Chatbot using AI	Computer Science	AI Chatbot	2	Pham Van Huy	Open	Edit Delete
Build a website using ASP.NET	Software Engineering	Requirements...	1	Mai Van Manh	Open	Edit Delete
Build an e-commerce website	Software Engineering	Build...	3	Mai Van Manh	Open	Edit Delete
Design IOT environmental information tracking	Network Engineering	IOT	1	Le Anh Cuong	Open	Edit Delete
Design customer care application	Computer Science	Design customer care application.	1	Tran Trung Tin	Open	Edit Delete
Build a website fashion store management with MERN Stack	Software Engineering	Learn MERN Stack and develop a fashion store management website	2	Mai Van Manh	Open	Edit Delete
Build a website hotel booking using Flutter	Software Engineering	Hotel Booking...	1	Doan Xuan Thanh	Open	Edit Delete
Build a website manage football schedule	Software Engineering	Manage Football Field Reservation Schedule	1	Duong Huu Phuc	Open	Edit Delete



Mai Van Manh

Registration Topic

Proposed Topic

Submission History

[Sign Out](#)

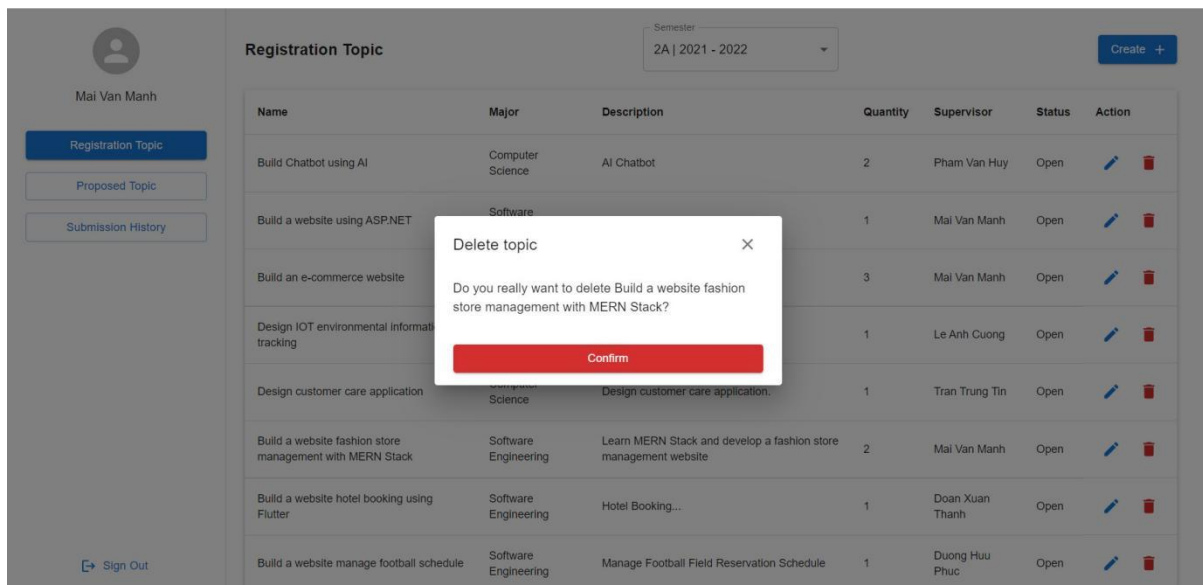
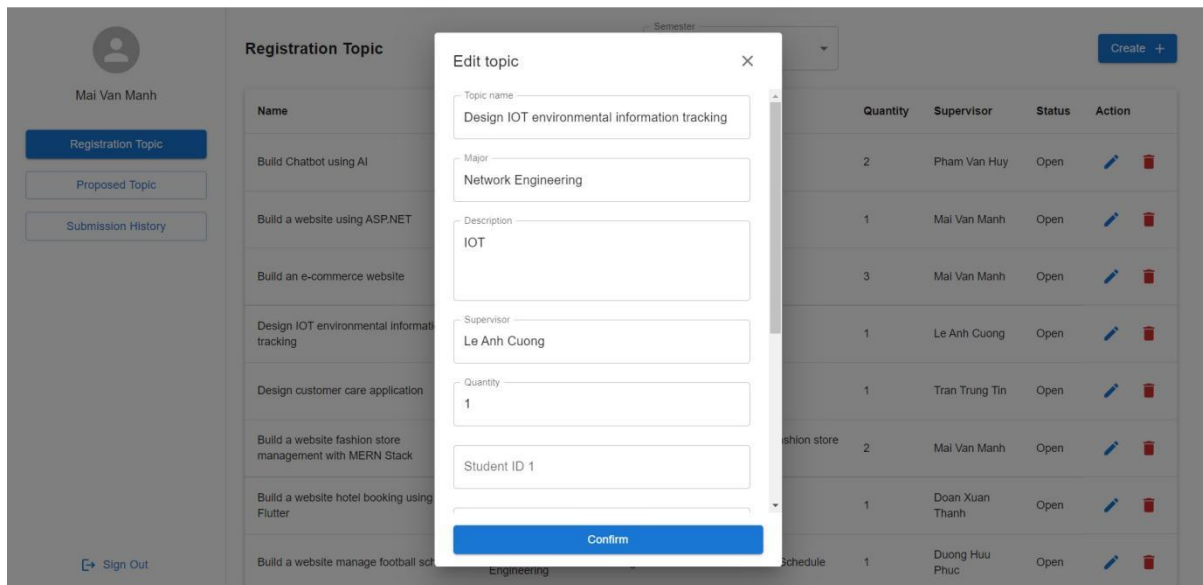
Semester
2A | 2021 - 2022

Create +

Name	Major	Description	Quantity	Supervisor	Status	Action
Build Chatbot using AI	Computer Science	AI Chatbot	2	Pham Van Huy	Open	Edit Delete
Build a website using ASP.NET	Software Engineering	Requirements...	1	Mai Van Manh	Open	Edit Delete
Build an e-commerce website	Software Engineering	Build...	3	Mai Van Manh	Open	Edit Delete
Design IOT environmental information tracking	Network Engineering	IOT	1	Le Anh Cuong	Open	Edit Delete
Design customer care application	Computer Science	Design customer care application.	1	Tran Trung Tin	Open	Edit Delete
Build a website fashion store management with MERN Stack	Software Engineering	Learn MERN Stack and develop a fashion store management website	2	Mai Van Manh	Open	Edit Delete
Build a website hotel booking using Flutter	Software Engineering	Hotel Booking...	1	Doan Xuan Thanh	Open	Edit Delete
Build a website manage football schedule	Software Engineering	Manage Football Field Reservation Schedule	1	Duong Huu Phuc	Open	Edit Delete

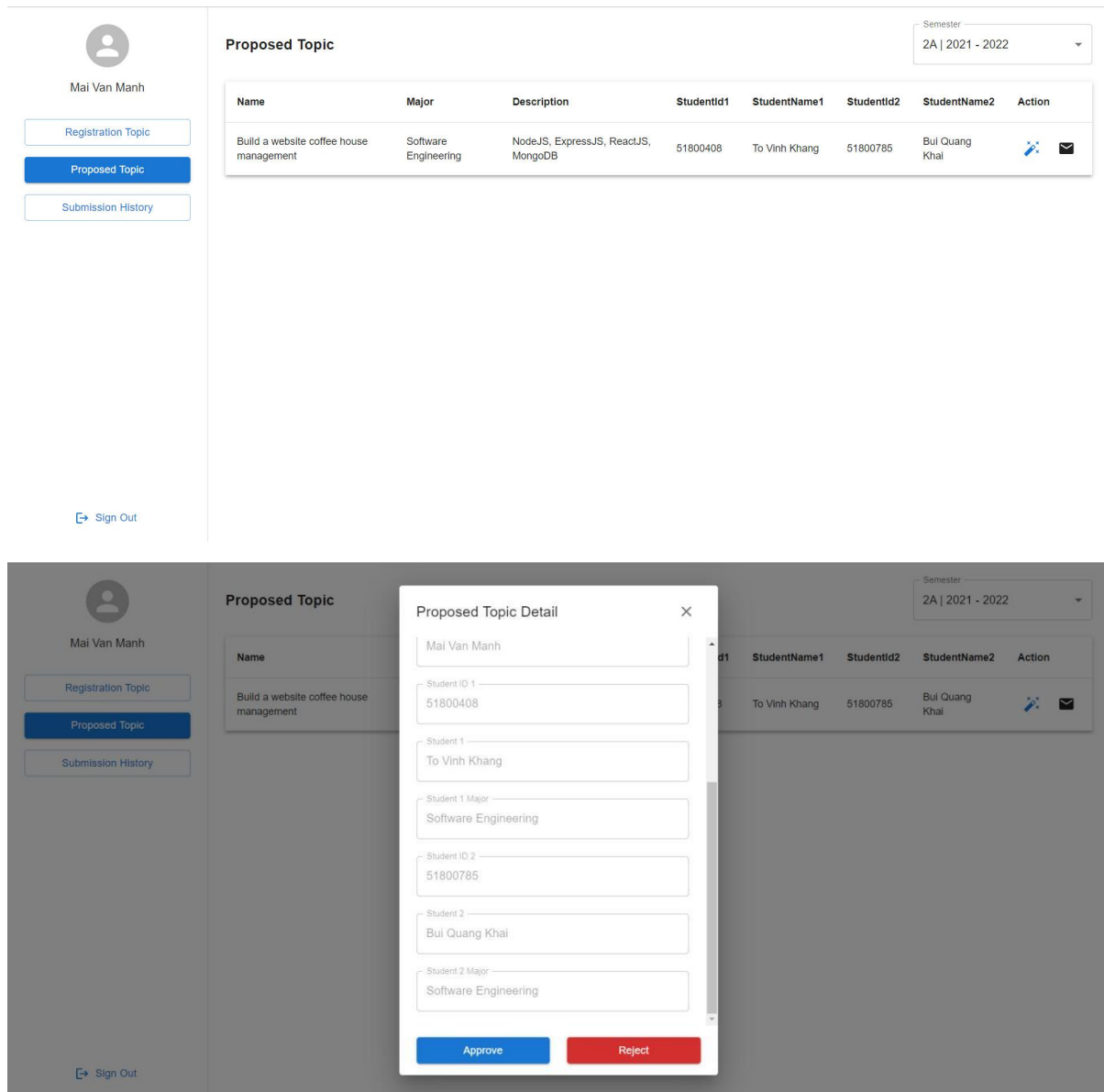
Create topic

Confirm




Hình 34. Màn hình Quản lý đề tài hướng dẫn

5.2.7 Màn hình Đề tài được đề xuất



Hình 35. Màn hình Quản lý đề tài đề xuất

5.2.8 Màn hình Xem lịch sử đề tài SV đã đăng ký



Mai Van Manh

[Registration Topic](#)

[Proposed Topic](#)

[Submission History](#)


Submission History

Name	Major	Description	Supervisor	Status	Semester	Type
Build an e-commerce website	Software Engineering	Build...	Mai Van Manh	Success	2A 2021 - 2022	Register
Research Sync Data using Hangfire	Software Engineering	Sync Data with Hangfire...	Mai Van Manh	Pending	2A 2021 - 2022	Propose

[Sign Out](#)

Hình 36. Màn hình Xem lịch sử đề tài SV đã đăng ký

5.2.9 Màn hình Quản lý học kỳ



Admin

[Semester Management](#)

[Lecturer Management](#)

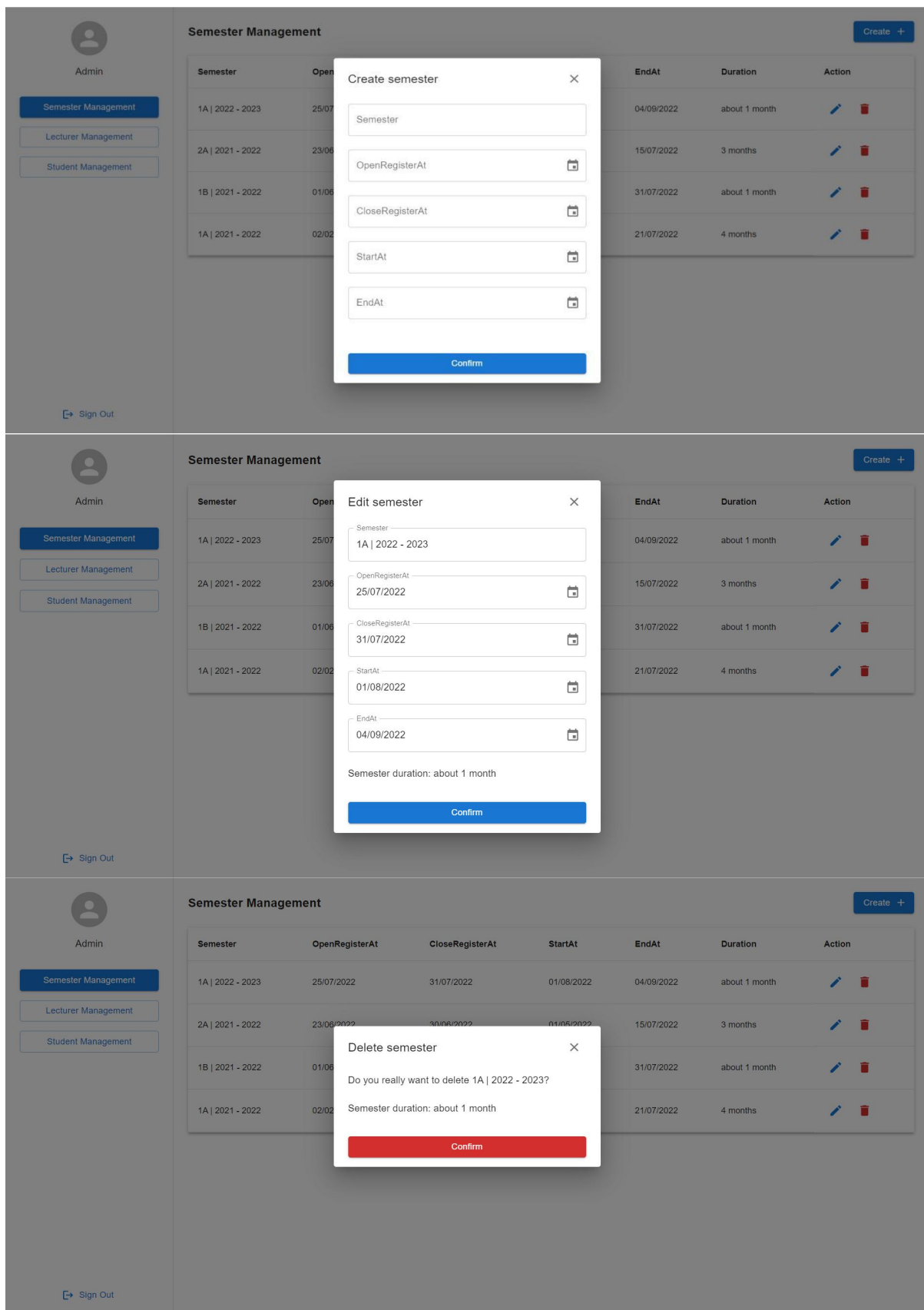
[Student Management](#)

Semester Management

Create +


Semester	OpenRegisterAt	CloseRegisterAt	StartAt	EndAt	Duration	Action
1A 2022 - 2023	25/07/2022	31/07/2022	01/08/2022	04/09/2022	about 1 month	Edit Delete
2A 2021 - 2022	23/06/2022	30/06/2022	01/05/2022	15/07/2022	3 months	Edit Delete
1B 2021 - 2022	01/06/2022	03/06/2022	22/06/2022	31/07/2022	about 1 month	Edit Delete
1A 2021 - 2022	02/02/2022	09/03/2022	10/03/2022	21/07/2022	4 months	Edit Delete

[Sign Out](#)



Hình 37. Màn hình Quản lý học kỳ

5.2.10 Màn hình Quản lý giảng viên


Admin



Semester Management

















Lecturer Management


Student Management

Sign Out

Lecturer Management

Import  Create 

Name	Subject1	Subject2	IsApprover	Action
Pham Van Huy	Computer Science	Software Engineering	<input checked="" type="checkbox"/>	 
Mai Van Manh	Software Engineering	Computer Science	<input type="checkbox"/>	 
Doan Xuan Thanh	Software Engineering	Computer Science	<input type="checkbox"/>	 
Tran Luong Quoc Dai	Computer Science		<input type="checkbox"/>	 
Dung Cam Quang	Software Engineering	Computer Science	<input type="checkbox"/>	 
Le Van Vang	Software Engineering	Computer Science	<input checked="" type="checkbox"/>	 
Bui Quy Anh	Network Engineering	Computer Science	<input checked="" type="checkbox"/>	 
Ho Thi Linh	Computer Science		<input type="checkbox"/>	 


Admin



Semester Management

















Lecturer Management

Student Management

Sign Out

Lecturer Management

Import  Create 

Name	Subject1	Subject2	IsApprover	Action
Pham Van Huy	Computer Science	Software Engineering	<input checked="" type="checkbox"/>	 
Mai Van Manh	Software Engineering	Computer Science	<input type="checkbox"/>	 
Doan Xuan Thanh	Software Engineering	Computer Science	<input type="checkbox"/>	 
Tran Luong Quoc Dai	Computer Science		<input type="checkbox"/>	 
Dung Cam Quang	Software Engineering	Computer Science	<input type="checkbox"/>	 
Le Van Vang	Software Engineering	Computer Science	<input checked="" type="checkbox"/>	 
Bui Quy Anh	Network Engineering	Computer Science	<input checked="" type="checkbox"/>	 
Ho Thi Linh	Computer Science		<input type="checkbox"/>	 

Create lecturer


Lecturer name

Subject 1

Subject 2 (Optional)

☐ Approver

Confirm


Admin



Semester Management

















Lecturer Management

Student Management

Sign Out

Lecturer Management

Import  Create 

Name	Subject1	Subject2	IsApprover	Action
Pham Van Huy	Computer Science	Software Engineering	<input checked="" type="checkbox"/>	 
Mai Van Manh	Software Engineering	Computer Science	<input type="checkbox"/>	 
Doan Xuan Thanh	Software Engineering	Computer Science	<input type="checkbox"/>	 
Tran Luong Quoc Dai	Computer Science		<input type="checkbox"/>	 
Dung Cam Quang	Software Engineering	Computer Science	<input type="checkbox"/>	 
Le Van Vang	Software Engineering	Computer Science	<input checked="" type="checkbox"/>	 
Bui Quy Anh	Network Engineering	Computer Science	<input checked="" type="checkbox"/>	 
Ho Thi Linh	Computer Science		<input type="checkbox"/>	 

Edit lecturer

Lecturer name

Mai Van Manh

Subject 1

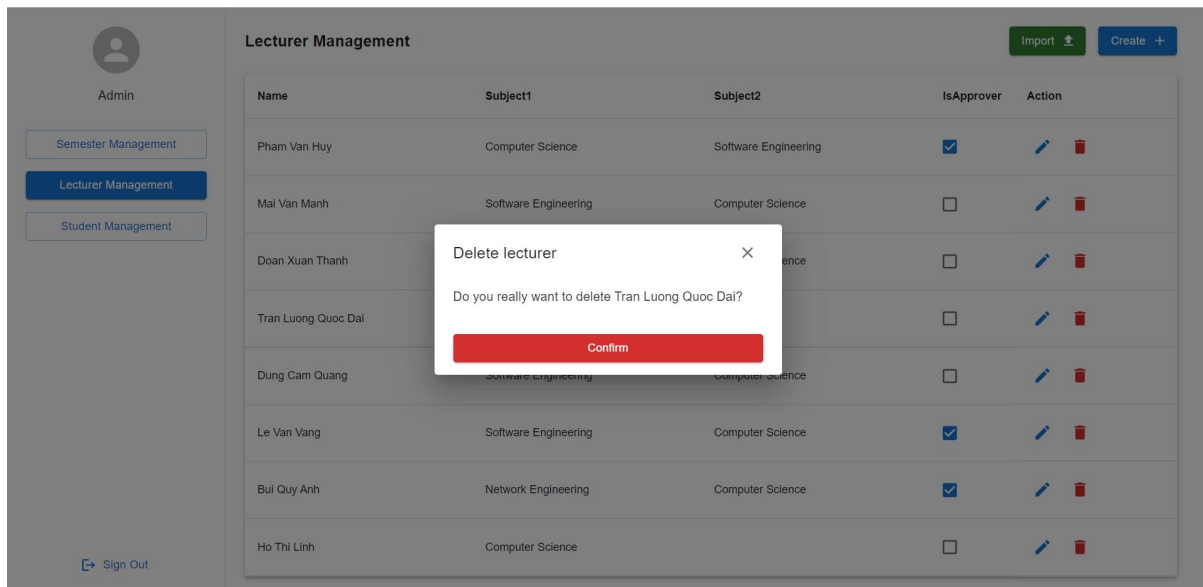
Software Engineering

Subject 2 (Optional)

Computer Science

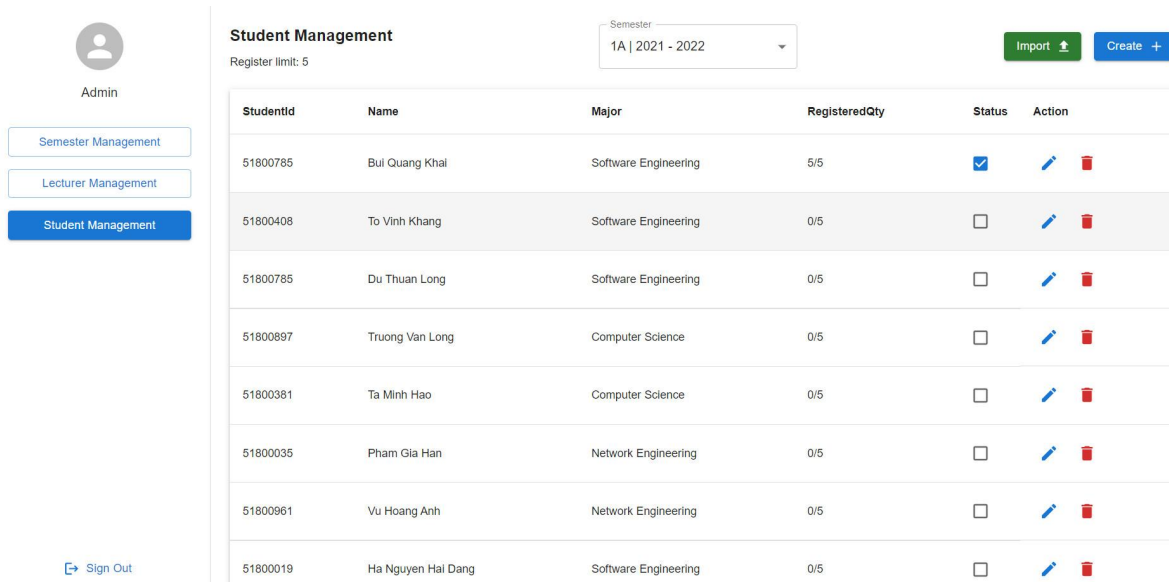
☐ Approver

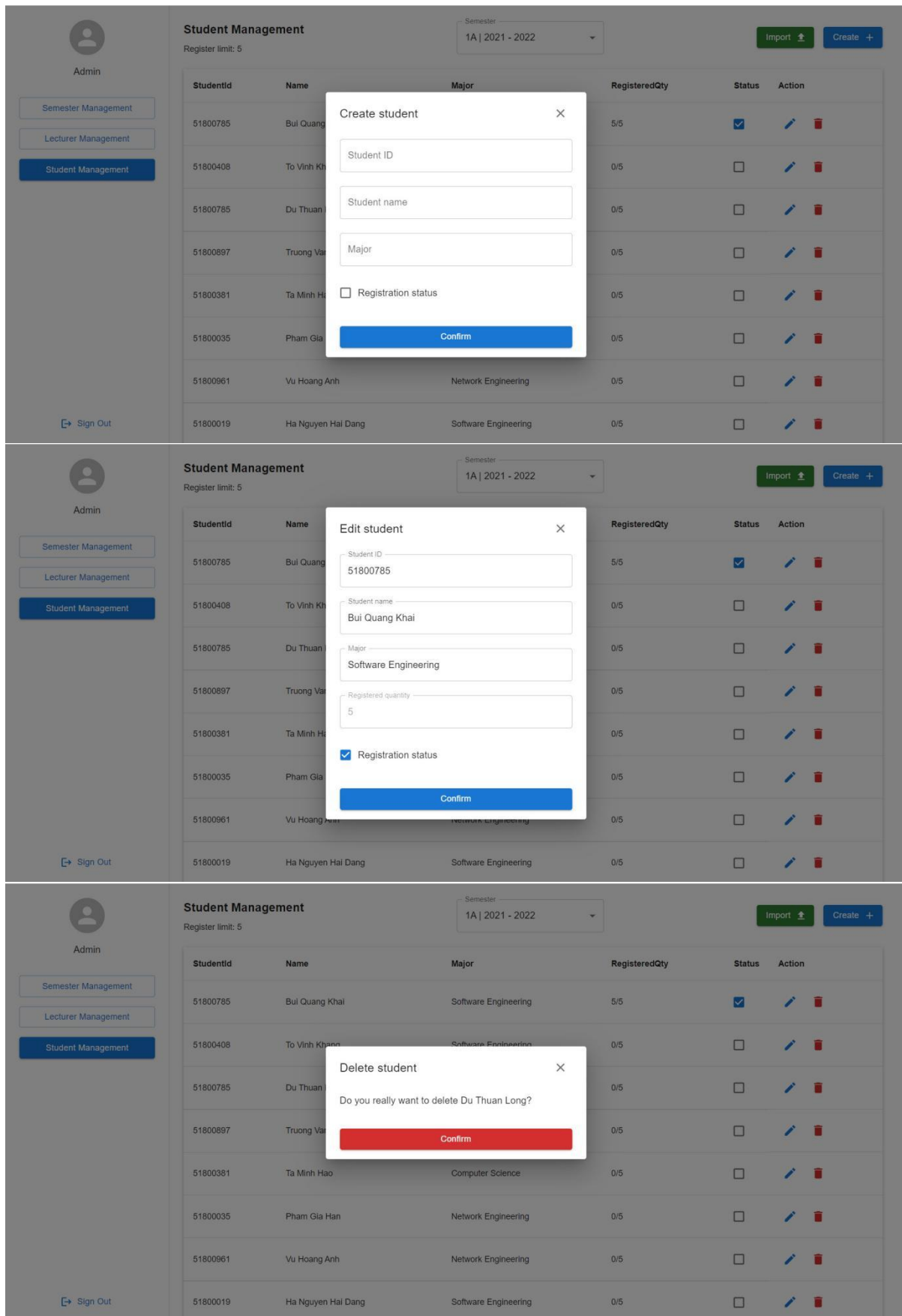
Confirm



Hình 38. Màn hình Quản lý giảng viên

5.2.11 Màn hình Quản lý sinh viên





Hình 39. Màn hình Quản lý sinh viên

CHƯƠNG VI: TỔNG KẾT

Bài báo cáo này đã phân tích yêu cầu và thiết kế một website thời gian thực, phía back-end sử dụng thư viện NodeJS-ExpressJS và lưu trữ dữ liệu thông qua MongoDB. Sau đó phía front-end sử dụng ReactJS, Redux, TypeScript, ViteJS, MaterialUI để hiển thị dữ liệu lên cho người dùng với mức độ trải nghiệm người dùng đạt được hiệu quả cao nhất. Sử dụng Netlify và Railway lần lượt để hosting phía frontend và backend. Sơ đồ use case, sơ đồ hoạt động và ERD đã phần nào miêu tả khái quát hóa các chức năng cốt lõi của hệ thống, cách thức hoạt động của từng chức năng và một số ràng buộc quan trọng.

Với đề tài “Hệ thống Quản lý Đăng ký Đề tài”, nhóm đã học hỏi thêm được rất nhiều thông qua việc tìm hiểu các tài liệu từ nhiều nguồn, các chức năng hỗ trợ tiện lợi từ nền tảng có sẵn để góp phần đưa ra ý tưởng cho việc thiết kế website này một cái nhìn tổng thể, bao quát nhất. Việc triển khai website này giúp cho việc đăng ký đề tài của các bạn sinh viên trở nên dễ dàng hơn, giảng viên cũng như văn phòng Khoa giảm tải đi được rất nhiều công đoạn thủ công hiện nay. Tuy nhiên, một vài chức năng vẫn còn chưa hoạt động được hoặc chưa hoạt động như mong muốn, bên cạnh đó nếu deploy thực tế cần có server mạnh để việc truyền tải dữ liệu được nhanh chóng. Ứng dụng này về cơ bản đã có giao diện với màu sắc và bố cục sắp xếp các thành phần mà theo cảm nhận của nhóm là phù hợp và thân thiện với người dùng.

Trong tương lai, nhóm sẽ tiếp tục phát triển cải thiện hiệu suất các tính năng đang làm và bổ sung thêm một số tính năng mở rộng khác như thông báo kết quả khi đăng ký đề xuất đề tài, nhận lời góp ý qua tin nhắn sms/mail,... Việc tạo một web thực tế để nhiều người truy cập sẽ làm cho việc phản hồi về dữ liệu đôi khi bị delay nhưng chức năng vẫn đảm bảo hoạt động tốt. Việc tìm hiểu, nghiên cứu và triển khai các ứng dụng tương tự như vậy sẽ giúp nâng cao khả năng phân tích nghiệp vụ, khả năng học hỏi công nghệ mới, góp phần cho lĩnh vực công nghệ phần mềm ngày càng phát triển cũng như những mong đợi mà nhóm đang theo đuổi.

TÀI LIỆU THAM KHẢO

- [1] J.W. Satzinger, R.B. Jackson, S.D. Burd, [2011], Systems Analysis and Design in a Changing World, Australia.
- [2] A. Banks, E. Porcello, [2017], Learning React - Functional Web Development with React and Redux, USA.
- [3] A. Boduch, [2019], React Material UI Cookbook, UK.
- [4] G. Lim, [2019], Beginning Node.js, Express & MongoDB Development, Singapore.
- [5] E. Brown, [2014], Web Development with Node and Express: Leveraging the JavaScript Stack, Portland.
- [6] E.Wilson, [2018], MERN Quick Start Guide: Build Web Applications with MongoDB, Express.js, React, and Node 1st Edition, Kindle Edition, UK.
- [7] L. Buley, The User Experience Team of One: A Research and Design Survival Guide, USA.
- [8] M.J. Hamm, [2014], Wireframing Essentials: An Introduction to User Experience Design, UK.
- [9] M. Biehl, [2016], RESTful API Design: Best Practices in API Design with REST, Kindle Edition, UK.
- [10] F. Doglio, [2018], REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development 2nd Edition, USA.
- [11] NodeJS Docs | <https://nodejs.org/en/docs/>
- [12] ExpressJS Docs | <https://devdocs.io/express/>
- [13] PNPM Docs | <https://pnpm.io/feature-comparison>

- [14] MongoDB Docs | <https://www.mongodb.com/docs/manual/core/document/>
- [15] ReactJS Docs | <https://reactjs.org/docs/introducing-jsx.html>
- [16] TypeScript Docs | <https://www.typescriptlang.org/docs/handbook/react.html>
- [17] Redux Docs | <https://redux.js.org/introduction/getting-started>
- [18] ViteJS Docs | <https://vitejs.dev/guide/>
- [19] MaterialUI Docs | <https://mui.com/material-ui/getting-started/usage/>
- [20] MongoAtlas Docs | <https://www.mongodb.com/docs/atlas/getting-started/>
- [21] FirebaseStorage Docs | <https://firebase.google.com/docs/storage>
- [21] Railway Docs | <https://docs.railway.app/>
- [22] Netlify Docs | <https://docs.netlify.com/>