

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Đồ án chuyên ngành (CO4029)

Hệ thống quản lý luận văn cho Đại học Bách Khoa (TMS)

Giảng viên hướng dẫn: Lê Đình Thuận

Lớp: L18

Thành viên nhóm:

Bùi Thế Kỷ Cương – 2210412

Lý Vĩnh Thái – 2213104

Phùng Xương Cận – 2210348

Hồ Chí Minh, ngày 29, tháng 11, năm 2025

Tóm tắt

Hệ thống Quản lý Luận văn (TMS) được phát triển nhằm số hóa quy trình quản lý đào tạo tại Khoa Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách Khoa - ĐHQG TP.HCM. Đề tài giải quyết các bất cập từ phương thức vận hành thủ công trong hai môn học Đồ án chuyên ngành và Luận văn tốt nghiệp, vốn gây khó khăn cho việc theo dõi tiến độ và đồng bộ dữ liệu của hàng trăm sinh viên mỗi kỳ.

Mục tiêu chính là xây dựng nền tảng trực tuyến giúp tự động hóa khâu gửi/duyệt đề tài, đánh giá giữa kỳ (Pass/Fail), chấm điểm cuối kỳ đa tầng và xếp lịch bảo vệ thông minh. Hệ thống áp dụng kiến trúc Microservices với 06 dịch vụ lõi (Thesis, Role, User, Council, File, Academic) phát triển bằng Golang, giao tiếp nội bộ qua gRPC hiệu năng cao. Tầng dữ liệu sử dụng chiến lược Polyglot Persistence: MySQL cho dữ liệu quan hệ, MongoDB lưu trữ logs, Redis làm hàng đợi/cache và MinIO lưu trữ file báo cáo dung lượng lớn. Server Gateway tích hợp GraphQL giúp tối ưu hóa truy vấn cho Frontend Next.js/Vite. Toàn bộ hệ thống được giám sát thời gian thực qua bộ công cụ Prometheus, Loki và Grafana.

Về mặt kỹ thuật, TMS đảm bảo yêu cầu phi chức năng với phản hồi dưới 3 giây, bảo mật RBAC/JWT và khả năng mở rộng đa khoa. Thiết kế CSDL gồm 18 thực thể, tập trung vào vòng đời của Đề tài (Topic). Kết quả kiểm thử bằng k6-tests cho thấy hệ thống chịu tải ổn định với 300 người dùng ảo, giúp giảm đến 70% thời gian thực hiện các thủ tục hành chính so với quy trình cũ.

Dù đạt mục tiêu số hóa, hệ thống vẫn tồn tại hạn chế về module AI kiểm tra đạo văn. Hướng phát triển tiếp theo tập trung vào tích hợp mô hình NLP (BERT) để phát hiện trùng lặp, triển khai kiến trúc đa khoa (Multi-tenancy) cho toàn trường và nâng cao bảo mật qua xác thực đa yếu tố.

Mục lục

Chương 1 - Giới thiệu	8
1.1. Động cơ của đề tài	8
1.2. Mục tiêu	9
1.3. Phạm vi	9
1.4. Ý nghĩa	10
1.4.1. Ý nghĩa thực tiễn:	10
1.5. Cấu trúc báo cáo.....	11
Chương 2 – Kiến thức và công nghệ nền tảng	12
2.1. Kiến trúc Microservices	12
2.1.1. Lý thuyết	12
2.1.2. Lý do lựa chọn	12
2.1.3. Ứng dụng trong hệ thống	13
2.2. Các công nghệ phát triển Back-end	13
2.2.1. Ngôn ngữ Golang (Go) & Giao thức gRPC	13
2.2.2. Node.js (Express) & BullMQ	13
2.3. Công nghệ phát triển Front-end.....	14
2.3.1. Next.js.....	14
2.3.2. Vite.....	14
2.4. Hệ quản trị Cơ sở dữ liệu (Polyglot Persistence).....	14
2.4.1. MySQL (Relational Database).....	14
2.4.2. MongoDB (NoSQL Database).....	15
2.4.3. Redis (In-memory Data Store).....	15
2.4.4. MinIO (Object Storage)	15
2.4.5. Elasticsearch (Search Engine)	15
2.5. Công cụ Giám sát và Vận hành	16
2.5.1. Docker	16
2.5.2. Bộ công cụ giám sát: Grafana, Prometheus, Loki.....	16
Chương 3 - Công trình liên quan và Hiện trạng	18

3.1. Quy trình quản lý thủ công hiện tại	18
3.1.1. Mô tả quy trình vận hành	18
3.1.2. Phân tích nhược điểm	19
3.2. Các hệ thống tương tự	19
Chương 4 – Hệ thống đề xuất.....	21
4.1. Mô tả hệ thống	21
4.1.1. Tổng quan	21
4.1.2. Các phân hệ chính	21
4.1.3. Quy trình nghiệp vụ tổng quát (Workflow)	22
4.2. Yêu cầu hệ thống	24
4.2.1. Yêu cầu chức năng	24
4.2.2. Yêu cầu phi chức năng	25
4.2.3. Yêu cầu dữ liệu	25
Chương 5 – Phân tích và Thiết kế.....	27
5.1. Phân tích	27
5.1.1. Mô hình hoá quy trình nghiệp vụ (Business Process Modeling)	27
5.1.2. Mô hình hóa tương tác:	31
5.1.2.1. Danh sách tác nhân (Actors)	31
5.1.2.2. Sơ đồ Use Case (Use Case Diagram)	32
5.1.2.3. Đặc tả Use Case (Use Case Specifications)	34
5.1.2.4. Danh sách tóm tắt Use Case	46
5.2. Giải pháp công nghệ	47
5.2.1. Kiến trúc Microservices	47
5.2.2. Chiến lược đa lưu trữ (Polyglot Persistence)	48
5.3. Thiết kế	49
5.3.1. Kiến trúc hệ thống	49
5.3.2. Thiết kế cơ sở dữ liệu	51
5.3.3. Thiết kế chi tiết bên trong	54
5.3.4. Thiết kế giao diện:	55

5.3.4. Công cụ thiết kế:	76
Chương 6 – Hiện thực và kiểm thử.....	77
6.1. Môi trường triển khai	77
6.2. Kiểm thử hiệu năng (Load Testing).....	77
6.3. Đánh giá hệ thống.....	79
6.3.1. Đánh giá hiệu suất	79
6.3.2. Đánh giá tính khả dụng và bảo mật	81
Chương 7 – Tổng kết.....	83
7.1. Kết quả đạt được	83
7.2. Hướng phát triển	84
7.2.1. Hoàn thiện và Nâng cao module AI kiểm tra đạo văn (Plagiarism Detection)....	84
7.3. Kế hoạch thực hiện Đề án tốt nghiệp (Giai đoạn 2).....	84
Tài liệu tham khảo	88
Phụ lục.....	90
Phụ lục 1: Hướng dẫn cài đặt và vận hành hệ thống	90
Phụ lục 2: Danh mục các kho lưu trữ mã nguồn và tài nguyên	90
Phụ lục 3: Sơ đồ kiến trúc và Giám sát hệ thống	91

Mục lục bảng

Bảng 1. Danh sách các tác nhân trong hệ thống.....	31
Bảng 2. Bảng đặc tả use case "Gửi đề tài"	34
Bảng 3. Bảng đặc tả use case "Duyệt đề tài".....	37
Bảng 4. Bảng đặc tả use case "Tạo Hội Đồng"	40
Bảng 5. Bảng đặc tả use case "Chấm điểm Hội Đồng"	43
Bảng 6. Danh sách các use case của hệ thống.....	46

Mục lục hình ảnh

Hình 1. Sơ đồ hoạt động của quy trình nghiệp vụ tổng quát	24
Hình 2. Biểu đồ hoạt động của quy trình đăng ký đề tài	28
Hình 3. Biểu đồ hoạt động của quy trình tổ chức hội đồng bảo vệ	29
Hình 4. Biểu đồ hoạt động của quy trình chấm điểm	30
Hình 5. Biểu đồ hoạt động của quy trình nộp bài	31
Hình 6. Use case diagram cho nhóm chức năng quản lý đề tài	33
Hình 7. Use case diagram cho nhóm chức năng quản lý Hội đồng	34
Hình 8. Use case diagram cho nhóm chức năng quản lý chấm điểm	34
Hình 9. Sơ đồ kiến trúc hệ thống TMS	50
Hình 10. Sơ đồ thiết kế ý niệm	53
Hình 11. Giao diện Landing page	56
Hình 12. Giao diện đăng nhập	57
Hình 13. Dashboard: Quản lý Giảng viên	58
Hình 14. Dashboard: Quản lý Sinh viên	59
Hình 15. Giao diện Quản lý đề tài	60
Hình 16. Dashboard: Quản lý Hội đồng	61
Hình 17. Dashboard: Lịch báo vệ	62
Hình 18. Dashboard: Giáo viên Bộ môn	63
Hình 19. Dashboard: Gửi đề tài	65
Hình 20. Dashboard: Đề tài đang hướng dẫn	65
Hình 21. Dashboard: Hội đồng chấm điểm	66
Hình 22. Dashboard: Lịch hội đồng bảo vệ	67
Hình 23. Dashboard: Quản lý học kỳ	68
Hình 24. Dashboard: Quản lý người dùng	69
Hình 25. Dashboard: Quản lý đề tài	70
Hình 26. Dashboard: Quản lý lịch Bảo vệ	71
Hình 27. Dashboard: Quản lý Hội đồng	72
Hình 28. Dashboard: Quản lý Chuyên ngành	73
Hình 29. Dashboard: Quản lý Khoa	74
Hình 30. Dashboard: Phân tích học kỳ	75
Hình 31. Kết quả Load/Stress Test (1)	78
Hình 32. Kết quả Load/Stress Test (2)	78
Hình 33. Grafana Dashboard: Services Metrics	80
Hình 34. Dashboard Grafana: Services Performance	81
Hình 35. Grafana Dashboard: Services Errors	82

Chương 1 - Giới thiệu

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại các cơ sở giáo dục đại học, việc ứng dụng công nghệ thông tin vào quản lý quy trình đào tạo là xu thế tất yếu nhằm nâng cao hiệu quả và chất lượng. Hệ thống quản lý đào tạo không chỉ đơn thuần là công cụ lưu trữ mà còn đóng vai trò cốt lõi trong việc kết nối các bên liên quan, đảm bảo tính minh bạch và chính xác của dữ liệu. Đề án này tập trung nghiên cứu và xây dựng "Hệ thống quản lý luận văn tốt nghiệp" (Thesis Management System - TMS) nhằm giải quyết các bài toán đặc thù trong quy trình tổ chức, theo dõi và đánh giá khóa luận tại Khoa Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách Khoa - ĐHQG TP.HCM.

1.1. Động cơ của đề tài

Khoa Khoa học và Kỹ thuật Máy tính là một trong những đơn vị đào tạo công nghệ thông tin hàng đầu với số lượng sinh viên tốt nghiệp hàng năm lên đến hàng trăm người. Quy mô đào tạo lớn đặt ra những thách thức không nhỏ trong công tác quản lý, đặc biệt là đối với hai học phần quan trọng: Đề án chuyên ngành (Giai đoạn 1) và Luận văn tốt nghiệp (Giai đoạn 2).

Hiện trạng quy trình quản lý tại Khoa vẫn còn tồn tại nhiều bất cập do phụ thuộc lớn vào các thao tác thủ công:

- Thủ tục rườm rà và phân tán:** Việc quản lý đề tài, phân công sinh viên, và theo dõi tiến độ chủ yếu thực hiện qua email và các bảng tính Excel rời rạc. Điều này dẫn đến sự thiếu đồng bộ dữ liệu giữa Giáo vụ, Giảng viên hướng dẫn (GVHD), Giảng viên phản biện (GVPB) và Hội đồng bảo vệ.
- Khó khăn trong quản lý theo học kỳ:** Việc theo dõi đề tài và nhân sự (sinh viên, giảng viên) theo mã học kỳ (ví dụ: HK251) thường xuyên gặp sai sót do thiếu công cụ tự động hóa. Danh sách nhân sự phải được tải lên thủ công mỗi kỳ, dễ dẫn đến nhầm lẫn trong việc gán sinh viên hoặc phân công hội đồng.
- Hạn chế trong quy trình đánh giá:** Các hoạt động đánh giá giữa kỳ (Pass/Fail) và chấm điểm cuối kỳ chưa được hỗ trợ tốt. Việc kiểm soát sai lệch điểm số giữa các

chấm thi (ví dụ: lệch > 2 điểm) hoàn toàn phụ thuộc vào sự rà soát thủ công của giáo vụ, gây chậm trễ và thiếu minh bạch trong kết quả.

Xuất phát từ những thực tế trên, nhu cầu cấp thiết là phải xây dựng một hệ thống thông tin tập trung, có khả năng số hóa toàn diện quy trình để giảm tải gánh nặng hành chính và nâng cao chất lượng quản lý đào tạo.

1.2. Mục tiêu

Đề tài hướng tới việc xây dựng một giải pháp công nghệ toàn diện với các mục tiêu cụ thể sau:

- **Mục tiêu tổng quát:** Xây dựng nền tảng trực tuyến hỗ trợ đồng bộ hóa các hoạt động quản lý luận văn, từ giai đoạn đăng ký đề tài đến khi bảo vệ thành công. Hệ thống nhằm giảm thiểu tối đa thời gian xử lý thủ tục hành chính, tăng cường tính minh bạch và hiệu quả công việc cho tất cả các bên tham gia.
- **Mục tiêu cụ thể:**
 - **Quản lý tập trung theo học kỳ:** Hệ thống phải hỗ trợ phân loại và lưu trữ dữ liệu (đề tài, người dùng, hội đồng) gắn liền với mã học kỳ cụ thể, đảm bảo tính lịch sử và dễ dàng tra cứu.
 - **Tự động hóa quy trình nghiệp vụ:** Bao gồm các chức năng: gửi và duyệt đề tài (với cơ chế kiểm tra trùng lặp), tự động gán sinh viên vào đề tài, quản lý nộp báo cáo, và quy trình đánh giá điểm đa tầng (Giữa kỳ, GVHD, GVPB, Hội đồng).
 - **Tích hợp cơ chế kiểm soát chất lượng:** Hệ thống tự động cảnh báo khi có sự chênh lệch điểm số lớn, hỗ trợ xếp lịch bảo vệ thông minh và lưu trữ hồ sơ minh chứng điện tử.

1.3. Phạm vi

Hệ thống TMS được thiết kế để phục vụ nhu cầu quản lý chuyên biệt tại Khoa Khoa học và Kỹ thuật Máy tính với các giới hạn xác định:

- **Đối tượng áp dụng:** Hệ thống tập trung hỗ trợ quy trình cho hai môn học chính là Đồ án chuyên ngành và Luận văn tốt nghiệp.
- **Giới hạn chức năng:** Hệ thống hoạt động độc lập, không tích hợp trực tiếp với hệ thống đăng ký môn học hoặc thanh toán học phí chung của Trường Đại học Bách Khoa.
- **Lưu trữ:** Hỗ trợ lưu trữ đa dạng các định dạng tài liệu (báo cáo, slide, mã nguồn) với giới hạn dung lượng tối ưu ($\leq 200\text{MB}/\text{file}$) để đảm bảo hiệu suất.

1.4. Ý nghĩa

1.4.1. Ý nghĩa thực tiễn:

Việc triển khai hệ thống TMS mang lại giá trị thiết thực cho công tác quản lý đào tạo:

- **Đối với Giáo vụ:** Giảm thiểu đáng kể khối lượng công việc nhập liệu và rà soát thủ công, hạn chế sai sót trong phân công hội đồng và tổng hợp điểm.
- **Đối với Giảng viên:** Cung cấp công cụ quản lý đê tài hiệu quả, theo dõi tiến độ sinh viên dễ dàng và thực hiện chấm điểm trực tuyến thuận tiện.
- **Đối với Sinh viên:** Đảm bảo tính minh bạch về quy trình và kết quả đánh giá, giúp sinh viên chủ động theo dõi trạng thái hồ sơ tốt nghiệp của mình.

1.4.2. Ý nghĩa khoa học:

Đề tài là sự vận dụng các kiến thức và công nghệ tiên tiến trong lĩnh vực Kỹ thuật phần mềm và Hệ thống thông tin:

- **Kiến trúc Microservices:** Ứng dụng mô hình kiến trúc vi dịch vụ để xây dựng hệ thống có khả năng mở rộng linh hoạt, độ sẵn sàng cao và dễ dàng bảo trì.
- **Xử lý quy trình nghiệp vụ phức tạp:** Mô hình hóa và giải quyết bài toán quản lý luồng công việc (Workflow) phức tạp với nhiều trạng thái và điều kiện ràng buộc trong môi trường giáo dục.

- **Quản lý dữ liệu lớn:** Ứng dụng các công nghệ lưu trữ và truy vấn hiệu năng cao (Elasticsearch, Redis) để xử lý lượng lớn dữ liệu sinh viên và tài liệu học thuật phát sinh theo thời gian.

1.5. Cấu trúc báo cáo

Báo cáo được trình bày theo trình tự logic, bao gồm 8 chương:

- **Chương 2 – Kiến thức và công nghệ nền tảng:** Giới thiệu cơ sở lý thuyết về kiến trúc Microservices và các công nghệ chủ chốt được sử dụng (Golang, Next.js, MongoDB, v.v.).
- **Chương 3 - Công trình liên quan và Hiện trạng:** Phân tích chi tiết quy trình thủ công hiện tại và so sánh với các giải pháp tương tự để làm rõ sự cần thiết của đề tài.
- **Chương 4 – Hệ thống đề xuất:** Mô tả tổng quan về hệ thống TMS, xác định các yêu cầu chức năng, phi chức năng và yêu cầu dữ liệu.
- **Chương 5 – Phân tích và Thiết kế:** Trình bày chi tiết về mô hình hóa quy trình nghiệp vụ, thiết kế kiến trúc hệ thống, thiết kế cơ sở dữ liệu và giao diện người dùng.
- **Chương 6 – Hiện thực và kiểm thử:** Mô tả quá trình triển khai hệ thống thực tế và kết quả kiểm thử chức năng, hiệu năng.
- **Chương 7 – Đánh giá hệ thống:** Phân tích hiệu quả hoạt động của hệ thống dựa trên các số liệu thực nghiệm về hiệu suất và tính khả dụng.
- **Chương 8 – Tổng kết:** Tóm tắt kết quả đạt được, nhìn nhận các hạn chế và đề xuất hướng phát triển trong tương lai.

Chương 2 – Kiến thức và công nghệ nền tảng

Chương này trình bày cơ sở lý thuyết và phân tích kỹ thuật của các công nghệ được sử dụng trong việc xây dựng hệ thống quản lý luận văn. Việc lựa chọn bộ công nghệ (Tech stack) được thực hiện dựa trên các tiêu chí về hiệu năng, khả năng mở rộng, tính ổn định và sự phù hợp với các nghiệp vụ đặc thù của môi trường giáo dục đại học.

2.1. Kiến trúc Microservices

2.1.1. Lý thuyết

Microservices (Kiến trúc vi dịch vụ) là một phong cách kiến trúc phần mềm, trong đó ứng dụng được cấu trúc thành một tập hợp các dịch vụ nhỏ, độc lập [1]. Mỗi dịch vụ chạy trong một tiến trình riêng, giao tiếp thông qua các cơ chế nhẹ (như HTTP hoặc gRPC) và thường được tổ chức xung quanh một khả năng nghiệp vụ cụ thể.

2.1.2. Lý do lựa chọn

- **Khả năng mở rộng linh hoạt (Scalability):** Hệ thống luận văn có các phân hệ chịu tải khác nhau. Ví dụ, phân hệ Nộp bài và Chấm điểm sẽ chịu tải cao vào cuối kỳ, trong khi phân hệ Quản lý người dùng ít biến động hơn. Microservices cho phép mở rộng tài nguyên cho riêng các dịch vụ chịu tải cao mà không lãng phí tài nguyên cho toàn bộ hệ thống.
- **Tính độc lập và cô lập lỗi:** Trong kiến trúc Monolithic (nguyên khối), một lỗi nhỏ có thể làm sập toàn bộ hệ thống. Với Microservices, nếu dịch vụ File gặp sự cố, các dịch vụ khác như Chấm điểm hay Tra cứu đề tài vẫn hoạt động bình thường, đảm bảo tính sẵn sàng cao.
- **Linh hoạt công nghệ (Polyglot Programming):** Cho phép sử dụng các ngôn ngữ lập trình và cơ sở dữ liệu khác nhau tối ưu cho từng nghiệp vụ (Golang cho hiệu năng cao, Node.js cho xử lý bất đồng bộ).

2.1.3. Ứng dụng trong hệ thống

Hệ thống được chia thành 6 dịch vụ backend cốt lõi độc lập: Thesis (Đề tài), Role (Phân quyền), User (Người dùng), Council (Hội đồng), File (Lưu trữ), và Academic (Học vụ).

2.2. Các công nghệ phát triển Back-end

2.2.1. Ngôn ngữ Golang (Go) & Giao thức gRPC

- Lý thuyết: Golang là ngôn ngữ lập trình biên dịch do Google phát triển [2], nổi bật với khả năng xử lý đồng thời (concurrency) mạnh mẽ thông qua Goroutines. gRPC là framework RPC hiệu năng cao sử dụng Protocol Buffers [3] để định nghĩa cấu trúc dữ liệu và tuân tự hóa nhị phân.
- Lý do lựa chọn: Golang cung cấp hiệu suất vượt trội so với các ngôn ngữ thông dịch, phù hợp để xử lý hàng ngàn yêu cầu đồng thời từ sinh viên. gRPC được chọn thay thế RESTful API cho giao tiếp nội bộ vì tốc độ nhanh hơn, dung lượng bản tin nhỏ hơn và định nghĩa kiểu dữ liệu chặt chẽ.
- Ứng dụng: Được sử dụng để phát triển toàn bộ 6 microservices chính và Server Gateway trung tâm.

2.2.2. Node.js (Express) & BullMQ

- Lý thuyết: Node.js là môi trường chạy JavaScript phía server với mô hình non-blocking I/O. BullMQ là thư viện quản lý hàng đợi (Message Queue) mạnh mẽ dựa trên Redis [4].
- Lý do lựa chọn: Node.js rất mạnh trong việc xử lý các tác vụ bất đồng bộ (I/O bound). Việc kết hợp với BullMQ giúp hệ thống xử lý các tác vụ nặng (như kiểm tra đạo văn, gửi email hàng loạt) dưới nền (background jobs) mà không làm tắc nghẽn luồng xử lý chính của người dùng.
- Ứng dụng: Xây dựng module Backend Workflow, chịu trách nhiệm xử lý các nghiệp vụ ngầm như kiểm tra trùng lặp đề tài, gửi thông báo và tự động cập nhật trạng thái đề tài.

2.3. Công nghệ phát triển Front-end

2.3.1. Next.js

- Lý thuyết: Next.js là một framework dựa trên thư viện React, hỗ trợ Server-Side Rendering (SSR) và Static Site Generation (SSG) [5].
- Lý do lựa chọn: SSR giúp cải thiện đáng kể tốc độ tải trang ban đầu (First Contentful Paint) và tối ưu hóa SEO, điều này quan trọng đối với công thông tin sinh viên nơi trải nghiệm người dùng cần sự mượt mà.
- Ứng dụng: Xây dựng giao diện người dùng chính (Main Frontend) cho tất cả các đối tượng: Sinh viên, Giảng viên và Giáo vụ.

2.3.2. Vite

- Lý thuyết: Vite là công cụ build frontend thế hệ mới, tập trung vào tốc độ khởi động server phát triển và cập nhật module nhanh (HMR).
- Lý do lựa chọn: Cung cấp môi trường phát triển cực nhanh và bản build nhẹ, phù hợp cho các trang quản trị nội bộ không yêu cầu SEO nhưng cần phản hồi tức thì.
- Ứng dụng: Xây dựng trang Admin Local, nơi quản trị viên hệ thống giám sát các thông số kỹ thuật, logs và trạng thái các job trong hàng đợi.

2.4. Hệ quản trị Cơ sở dữ liệu (Polyglot Persistence)

Hệ thống áp dụng chiến lược đa lưu trữ, sử dụng đúng loại CSDL cho đúng loại dữ liệu:

2.4.1. MySQL (Relational Database)

- **Lý thuyết:** Hệ quản trị CSDL quan hệ, tuân thủ chuẩn ACID (Atomicity, Consistency, Isolation, Durability) [6].
- **Lý do lựa chọn:** Đảm bảo tính toàn vẹn dữ liệu cao nhất cho các nghiệp vụ cốt lõi và các mối quan hệ phức tạp giữa các thực thể (Ví dụ: Một sinh viên thuộc một ngành, một đê tài thuộc một hội đồng).
- **Ứng dụng:** Lưu trữ dữ liệu nghiệp vụ chính gồm 18 bảng: Sinh viên, Giảng viên, Đề tài, Điểm số, Hội đồng, v.v..

2.4.2. MongoDB (NoSQL Database)

- **Lý thuyết:** CSDL hướng văn bản (Document-oriented), lưu trữ dữ liệu dưới dạng JSON/ BSON linh hoạt [7].
- **Lý do lựa chọn:** Khả năng ghi tốc độ cao và linh hoạt thay đổi cấu trúc dữ liệu (schema-less), phù hợp cho các dữ liệu bán cấu trúc hoặc phi cấu trúc.
- **Ứng dụng:** Lưu trữ Logs sự kiện hệ thống (Event logs), cấu hình hệ thống động (Config), và lịch sử quy trình (Workflow logs).

2.4.3. Redis (In-memory Data Store)

- **Lý thuyết:** Kho lưu trữ cấu trúc dữ liệu trong bộ nhớ (In-memory), hỗ trợ tốc độ truy xuất dưới mili-giây [8].
- **Lý do lựa chọn:** Tăng tốc độ phản hồi của hệ thống bằng cách giảm tải truy vấn vào MySQL và đóng vai trò là Broker trung gian cho cơ chế hàng đợi tin nhắn.
- **Ứng dụng:** Lưu trữ bộ nhớ đệm (Caching) cho các dữ liệu thường xuyên truy cập và làm hàng đợi (Queue) cho BullMQ để xử lý tác vụ nền.

2.4.4. MinIO (Object Storage)

- **Lý thuyết:** Hệ thống lưu trữ đối tượng hiệu năng cao, tương thích với chuẩn Amazon S3 API [9].
- **Lý do lựa chọn:** Giải quyết bài toán lưu trữ file nhị phân lớn (BLOB) mà CSDL thông thường không xử lý tốt. MinIO cho phép tự host, bảo mật cao và dễ dàng mở rộng dung lượng.
- **Ứng dụng:** Lưu trữ tập trung toàn bộ các file báo cáo luận văn, tài liệu tham khảo, và phiếu chấm điểm với giới hạn dung lượng lên tới 200MB/file.

2.4.5. Elasticsearch (Search Engine)

- **Lý thuyết:** Công cụ tìm kiếm và phân tích phân tán dựa trên Apache Lucene [10].

- **Lý do lựa chọn:** Cung cấp khả năng tìm kiếm toàn văn (Full-text search) nhanh chóng và chính xác, hỗ trợ tìm kiếm mờ (fuzzy search) mà truy vấn SQL truyền thống không đáp ứng hiệu quả.
- **Ứng dụng:** Hỗ trợ chức năng tìm kiếm để tài theo tên, mô tả, hoặc nội dung bài nộp cho sinh viên và giảng viên.

2.5. Công cụ Giám sát và Vận hành

2.5.1. Docker

- **Lý thuyết:** Nền tảng container hóa giúp đóng gói ứng dụng và toàn bộ thư viện phụ thuộc vào một container duy nhất [11].
- **Lý do lựa chọn:** Đảm bảo tính nhất quán của môi trường chạy ứng dụng từ giai đoạn phát triển (Dev) đến triển khai (Production), đơn giản hóa việc triển khai microservices.
- **Ứng dụng:** Đóng gói toàn bộ các service backend, frontend và database để triển khai hệ thống.

2.5.2. Bộ công cụ giám sát: Grafana, Prometheus, Loki

- **Lý thuyết:**
 - **Prometheus:** Thu thập và lưu trữ các chỉ số (metrics) dạng chuỗi thời gian [12].
 - **Loki:** Hệ thống tổng hợp logs tối ưu, tiết kiệm tài nguyên.
 - **Grafana:** Nền tảng phân tích và trực quan hóa dữ liệu đa nguồn.
- **Lý do lựa chọn:** Cung cấp cái nhìn toàn diện về sức khỏe hệ thống (Observability). Giúp phát hiện sớm các điểm nghẽn (bottlenecks) và khoanh vùng lỗi nhanh chóng trong môi trường phân tán.

- **Ứng dụng:** Prometheus thu thập thông số CPU/RAM; Promtail đẩy logs từ 6 microservices về Loki; Grafana hiển thị các Dashboard theo dõi thời gian thực cho quản trị viên.

Chương 3 - Công trình liên quan và Hiện trạng

Chương này tập trung phân tích sâu vào quy trình quản lý luận văn hiện hành tại Khoa Khoa học và Kỹ thuật Máy tính, đồng thời đánh giá các giải pháp phần mềm quản lý đào tạo phổ biến hiện nay. Mục tiêu là làm rõ những bất cập trong quy trình thủ công và giới hạn của các hệ thống có sẵn, từ đó xác định khoảng trống (gap) mà hệ thống TMS cần giải quyết.

3.1. Quy trình quản lý thủ công hiện tại

Hiện tại, quy trình quản lý luận văn tốt nghiệp tại Khoa vẫn đang vận hành chủ yếu dựa trên các công cụ văn phòng cơ bản và phương thức giao tiếp truyền thống. Với quy mô hàng trăm sinh viên tốt nghiệp mỗi năm, phương thức này đang bộc lộ nhiều hạn chế nghiêm trọng.

3.1.1. Mô tả quy trình vận hành

Quy trình thủ công hiện tại diễn ra qua các bước chính như sau:

- Quản lý dữ liệu và Đăng ký: Giáo vụ sử dụng các bảng tính Excel rời rạc để lưu trữ danh sách sinh viên, giảng viên và đề tài theo từng học kỳ. Việc gán sinh viên vào đề tài hoặc phân công giảng viên hướng dẫn (GVHD) được thực hiện bằng cách nhập liệu thủ công vào các file này.
- Giao tiếp và Phê duyệt: Việc gửi đề tài từ GVHD lên Bộ môn, hoặc thông báo kết quả duyệt đề tài chủ yếu thực hiện qua email. Điều này dẫn đến sự thiếu đồng bộ, giảng viên bộ môn (GVBM) có thể duyệt chậm trễ do trôi tin, và sinh viên không nắm bắt được trạng thái đề tài (Đã duyệt/Chờ duyệt/Từ chối) theo thời gian thực.
- Đánh giá và Chấm điểm:
 - Giảng viên thực hiện chấm điểm trên các biểu mẫu giấy hoặc file Excel cá nhân, sau đó gửi lại cho giáo vụ tổng hợp.
 - Quy trình kiểm tra lệch điểm (nếu điểm GVHD và GVPB lệch nhau > 2 điểm) hoàn toàn phụ thuộc vào sự rà soát thủ công của con người.

- Hồ sơ minh chứng và báo cáo được lưu trữ dưới dạng giấy tờ vật lý hoặc file phân tán, gây khó khăn cho việc tra cứu lịch sử.

3.1.2. Phân tích nhược điểm

Phương thức quản lý trên tồn tại ba nhược điểm chí mạng:

- Độ trễ và thiếu đồng bộ (Latency & Desynchronization): Thông tin không được cập nhật tức thời (Real-time). Sinh viên thường xuyên phải chờ đợi email thông báo thủ công để biết mình có được bảo vệ hay không, trong khi Hội đồng bảo vệ có thể không truy cập được file báo cáo mới nhất của sinh viên ngay tại buổi bảo vệ.
- Sai sót trong xử lý dữ liệu: Việc quản lý danh sách bằng Excel và upload thủ công mỗi học kỳ dễ dẫn đến nhầm lẫn mã số sinh viên, trùng lặp đề tài giữa các kỳ, hoặc sai sót trong việc gán Hội đồng bảo vệ. Đặc biệt, việc không có công cụ tự động kiểm tra trùng lặp đề tài ảnh hưởng đến tính liêm chính học thuật.
- Khó khăn trong tra cứu và lưu trữ: Dữ liệu bị phân mảnh qua nhiều file Excel và email khiến việc tổng hợp báo cáo thống kê hoặc tra cứu lịch sử làm việc của một giảng viên qua các kỳ trở nên rất phức tạp và tốn thời gian.

3.2. Các hệ thống tương tự

Trong môi trường giáo dục đại học, các hệ thống quản lý học tập (Learning Management System - LMS) như Moodle, Blackboard, hay Canvas được sử dụng rất phổ biến. Tại Đại học Bách Khoa, hệ thống BKEL (dựa trên Moodle) cũng đang được triển khai rộng rãi. Tuy nhiên, các hệ thống này chủ yếu được thiết kế để phục vụ việc giảng dạy các môn học lý thuyết thông thường và bộc lộ nhiều hạn chế khi áp dụng vào quy trình luận văn đặc thù:

- **Hạn chế về quy trình nghiệp vụ phức tạp:** Các LMS thường quản lý theo cấu trúc "Lớp học - Bài tập - Chấm điểm". Trong khi đó, quy trình luận văn yêu cầu một vòng đời (Life-cycle) phức tạp hơn nhiều: *Đề xuất -> Duyệt đề tài -> Phân công phản biện -> Tổ chức Hội đồng -> Bảo vệ*. Các LMS không hỗ trợ sẵn quy trình phê duyệt đa cấp (GVHD -> GVBM -> Giáo vụ).

- **Hạn chế về vai trò động (Dynamic Roles):** Trong một hệ thống LMS, một người dùng thường cố định là "Giáo viên" hoặc "Sinh viên" trong một khóa học. Ngược lại, trong quy trình bảo vệ luận văn, một giảng viên có thể đồng thời đóng nhiều vai trò khác nhau đối với các sinh viên khác nhau trong cùng một học kỳ: vừa là GVHD của sinh viên A, vừa là GVPB của sinh viên B, và là Chủ tịch hội đồng của sinh viên C. Các hệ thống LMS hiện tại không hỗ trợ phân quyền mềm dẻo và chi tiết đến mức độ này.
- **Thiếu các tính năng đặc thù:** Các chức năng như tự động kiểm tra điều kiện bảo vệ (điểm quá trình, điểm phản biện), kiểm tra độ lệch điểm giữa các giám khảo, hay xếp lịch hội đồng tự động dựa trên thời gian rảnh là những yêu cầu đặc thù của quy trình khóa luận tại Bách Khoa mà các LMS đại trà không đáp ứng được.

Chương 4 – Hệ thống đề xuất

Trên cơ sở phân tích hiện trạng và đánh giá các giải pháp công nghệ ở các chương trước, chương này sẽ trình bày tổng quan về "Hệ thống Quản lý Luận văn Tốt nghiệp" (TMS) được đề xuất. Nội dung bao gồm mô tả kiến trúc nghiệp vụ tổng thể, xác định các phân hệ người dùng và đặc tả chi tiết các yêu cầu chức năng, phi chức năng cũng như yêu cầu về dữ liệu mà hệ thống cần đáp ứng.

4.1. Mô tả hệ thống

4.1.1. Tổng quan

Hệ thống TMS là một nền tảng web tập trung, được thiết kế để số hóa toàn trình quy trình quản lý khóa luận tại Khoa Khoa học và Kỹ thuật Máy tính. Hệ thống hoạt động như một môi trường cộng tác trực tuyến, kết nối ba đối tượng chính: Giáo vụ (Quản lý), Giảng viên (Hướng dẫn/Phản biện/Hội đồng) và Sinh viên.

4.1.2. Các phân hệ chính

Hệ thống được chia thành 3 phân hệ chức năng dựa trên vai trò người dùng:

- Phân hệ Giáo vụ (Administrator/Academic Staff): Đóng vai trò quản trị vận hành. Các chức năng bao gồm thiết lập cấu hình học kỳ, quản lý danh mục (Khoa, Ngành), nhập liệu danh sách người dùng (Sinh viên, Giảng viên), phê duyệt kế hoạch bảo vệ và tổ chức Hội đồng.
- Phân hệ Giảng viên (Teacher): Hỗ trợ đa vai trò linh hoạt trong cùng một học kỳ.
 - *Vai trò Hướng dẫn (Supervisor)*: Đề xuất đề tài, theo dõi tiến độ và chấm điểm hướng dẫn.
 - *Vai trò Phản biện (Reviewer)*: Truy cập hồ sơ sinh viên được phân công để đánh giá và chấm điểm phản biện.
 - *Vai trò Hội đồng (Council Member)*: Tham gia buổi bảo vệ, thực hiện chấm điểm trực tiếp trên hệ thống.

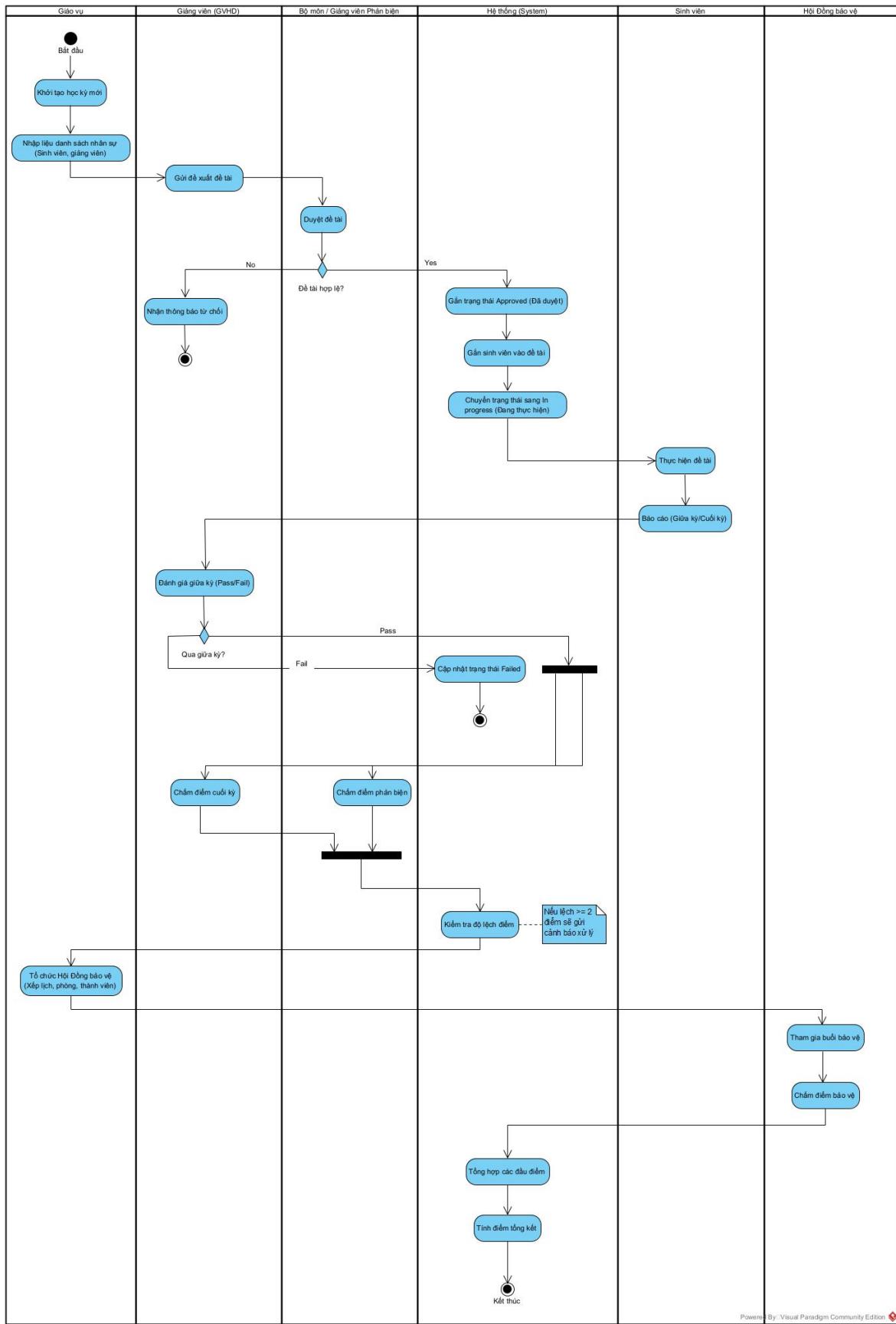
- Phân hệ Sinh viên (Student): Cho phép sinh viên theo dõi thông tin đề tài được phân công, nộp các báo cáo (giữa kỳ, cuối kỳ) và tra cứu kết quả đánh giá.

4.1.3. Quy trình nghiệp vụ tổng quát (Workflow)

Hệ thống vận hành theo một luồng quy trình khép kín, được kiểm soát chặt chẽ bằng máy trạng thái (State Machine):

1. Khởi tạo: Giáo vụ tạo Học kỳ mới và nhập liệu danh sách nhân sự.
2. Đăng ký Đề tài: Giảng viên gửi đề xuất đề tài -> Bộ môn duyệt (kiểm tra trùng lặp) -> Hệ thống gán trạng thái Approved (Đã duyệt).
3. Thực hiện: Hệ thống tự gán sinh viên vào đề tài -> Trạng thái chuyển sang In Progress (Đang thực hiện). Sinh viên thực hiện và nộp báo cáo.
4. Đánh giá:
 - Giữa kỳ: Đánh giá Pass/Fail.
 - Cuối kỳ: Giảng viên hướng dẫn và Phản biện chấm điểm độc lập. Hệ thống tự động kiểm tra độ lệch điểm.
5. Bảo vệ: Tổ chức Hội đồng, chấm điểm bảo vệ và tổng kết điểm cuối cùng để quyết định tốt nghiệp.

Quy trình tổng quát được trình bày trong Hình 1:



Hình 1. Sơ đồ hoạt động của quy trình nghiệp vụ tổng quát

4.2. Yêu cầu hệ thống

4.2.1. Yêu cầu chức năng

Hệ thống cần đáp ứng 5 nhóm chức năng cốt lõi sau:

- Quản lý Học kỳ và Danh sách:**

- Cho phép tạo mới và chuyển đổi linh hoạt giữa các học kỳ (Ví dụ: HK 241, HK 242).
- Hỗ trợ Import danh sách sinh viên, giảng viên từ file Excel/CSV theo chuẩn định dạng của nhà trường.

- Quản lý Đề tài (Topic Lifecycle):**

- Cho phép giảng viên gửi đề xuất đề tài mới hoặc tái sử dụng đề tài cũ.
- Tự động kiểm tra trùng lặp đề tài dựa trên tiêu đề và mô tả đề tài hỗ trợ ra quyết định duyệt.
- Quản lý trạng thái đề tài: *Chờ duyệt, Đã duyệt, Đang thực hiện, Hoàn thành, Hủy.*

- Quản lý Bài nộp (Submission):**

- Cho phép sinh viên tải lên (upload) các tài liệu báo cáo, slide thuyết trình.
- Ghi nhận lịch sử các phiên bản nộp bài (Version control) để giảng viên theo dõi quá trình.

- Chấm điểm và Đánh giá (Grading):**

- Hỗ trợ biểu mẫu chấm điểm số hóa cho GVHD, GVPB và Hội Đồng (gồm Chủ tịch, Thư ký, Ủy viên). GVPB cũng là thành viên trong Hội Đồng.
- Tự động tính toán điểm trung bình và phát hiện sai lệch điểm (Deviation check) lớn hơn ngưỡng quy định (ví dụ: ≥ 2.0 điểm) để cảnh báo.

- **Quản lý Hội đồng và Lịch bảo vệ:**
 - Hỗ trợ tạo lập Hội đồng bảo vệ (gồm Chủ tịch, Thư ký, Ủy viên, Phản biện).
 - Xếp lịch bảo vệ tự động hoặc bán tự động dựa trên danh sách đề tài và hội đồng.

4.2.2. Yêu cầu phi chức năng

Để đảm bảo trải nghiệm người dùng và tính ổn định, hệ thống phải tuân thủ các chỉ số kỹ thuật:

- **Hiệu suất (Performance):**
 - Thời gian phản hồi (Response time) cho các tác vụ thông thường phải dưới **3 giây** (đối với API là dưới 1 giây).
 - Hệ thống phải chịu được tải trọng đồng thời ít nhất **1000 người dùng** (concurrent users) trong các giai đoạn cao điểm như nộp bài hoặc xem điểm.
- **Bảo mật (Security):**
 - Cơ chế xác thực (Authentication) tích hợp SSO qua Google OAuth.
 - Cơ chế phân quyền (Authorization) chặt chẽ theo mô hình RBAC (Role-Based Access Control) sử dụng JWT, đảm bảo người dùng chỉ truy cập được dữ liệu trong phạm vi cho phép.
- **Khả năng mở rộng (Scalability):** Kiến trúc hệ thống phải hỗ trợ mở rộng theo chiều ngang (Horizontal Scaling) để phục vụ cho nhiều Khoa khác nhau trong tương lai.
- **Trải nghiệm người dùng (UI/UX):** Giao diện phải thân thiện, hỗ trợ Responsive trên đa thiết bị và hỗ trợ ngôn ngữ tiếng Việt.

4.2.3. Yêu cầu dữ liệu

Đặc thù của hệ thống quản lý luận văn đòi hỏi chiến lược lưu trữ dữ liệu phức tạp:

- **Lưu trữ Đa hình (Polymorphic Association):** Hệ thống cần thiết kế cơ sở dữ liệu cho phép một thực thể File (tài liệu) có thể gắn kết linh hoạt với nhiều loại đối tượng khác nhau (Đề tài, Bài nộp giữa kỳ, Bài nộp cuối kỳ, Quyết định thành lập hội đồng) mà không cần tạo nhiều bảng dữ liệu dư thừa.
- **Lưu trữ File lớn:** Hệ thống phải hỗ trợ lưu trữ các file báo cáo, bản vẽ kỹ thuật với dung lượng lớn (tối đa **200MB/file**) thông qua giải pháp Object Storage, tách biệt với cơ sở dữ liệu chính để đảm bảo hiệu năng.
- **Dữ liệu phi cấu trúc:** Hệ thống cần có khả năng lưu trữ logs hệ thống, lịch sử thao tác (Audit trail) và các cấu hình động dưới dạng phi cấu trúc (NoSQL) để phục vụ công tác giám sát và debug.

Chương 5 – Phân tích và Thiết kế

Chương này đi sâu vào chi tiết hóa các yêu cầu hệ thống thông qua các mô hình phân tích và thiết kế. Mục tiêu là chuyển đổi các yêu cầu nghiệp vụ và phi chức năng đã xác định ở Chương 4 thành các đặc tả kỹ thuật cụ thể, làm cơ sở cho việc hiện thực hóa phần mềm.

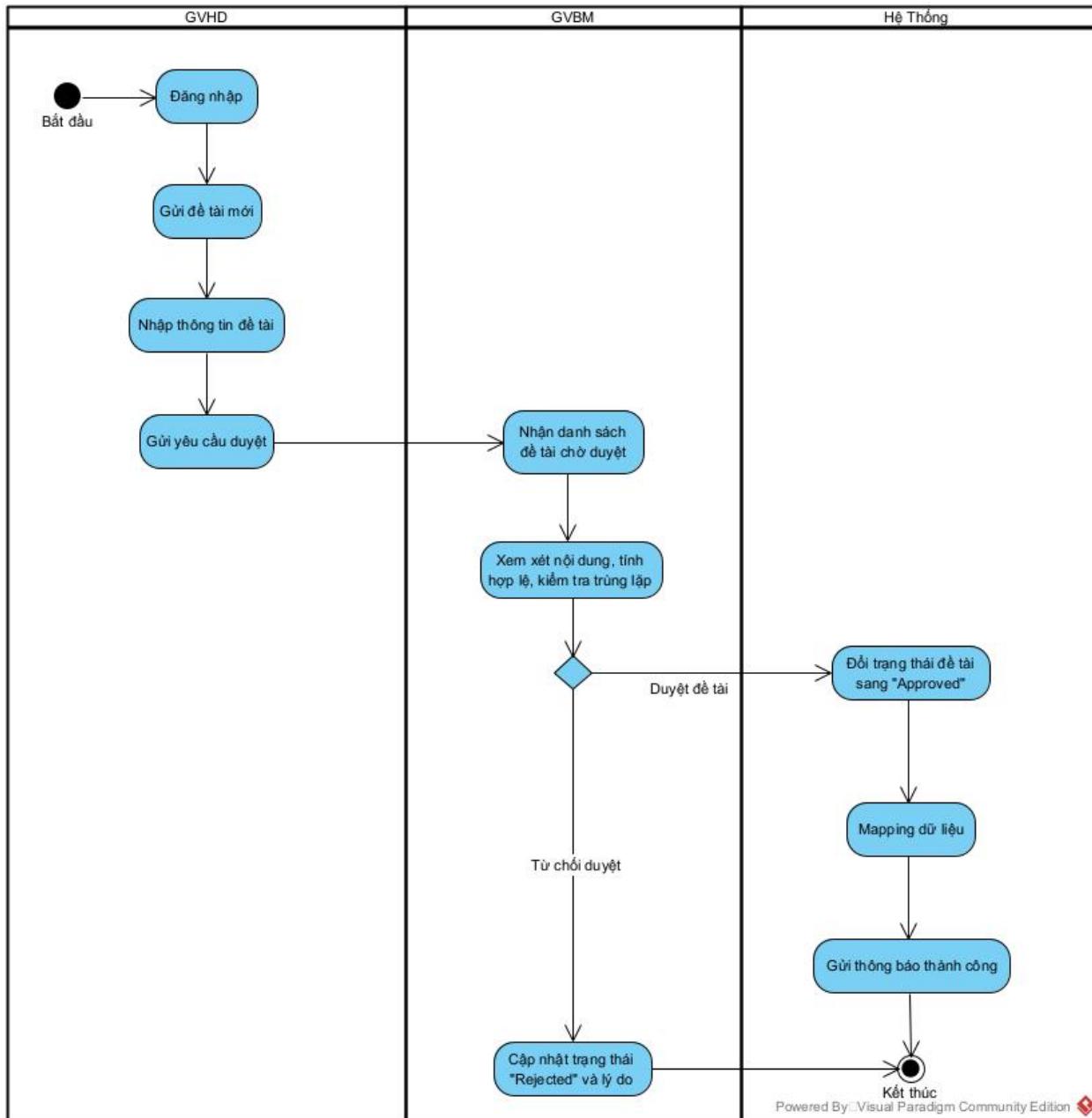
5.1. Phân tích

5.1.1. Mô hình hóa quy trình nghiệp vụ (Business Process Modeling)

Để làm rõ luồng công việc và sự tương tác giữa các tác nhân trong hệ thống, nhóm thực hiện sử dụng biểu đồ hoạt động (Activity Diagram) [13] cho 4 quy trình cốt lõi: Đăng ký đề tài, Tổ chức bảo vệ, Chấm điểm và Nộp bài.

a. Quy trình Đăng ký và Duyệt đề tài

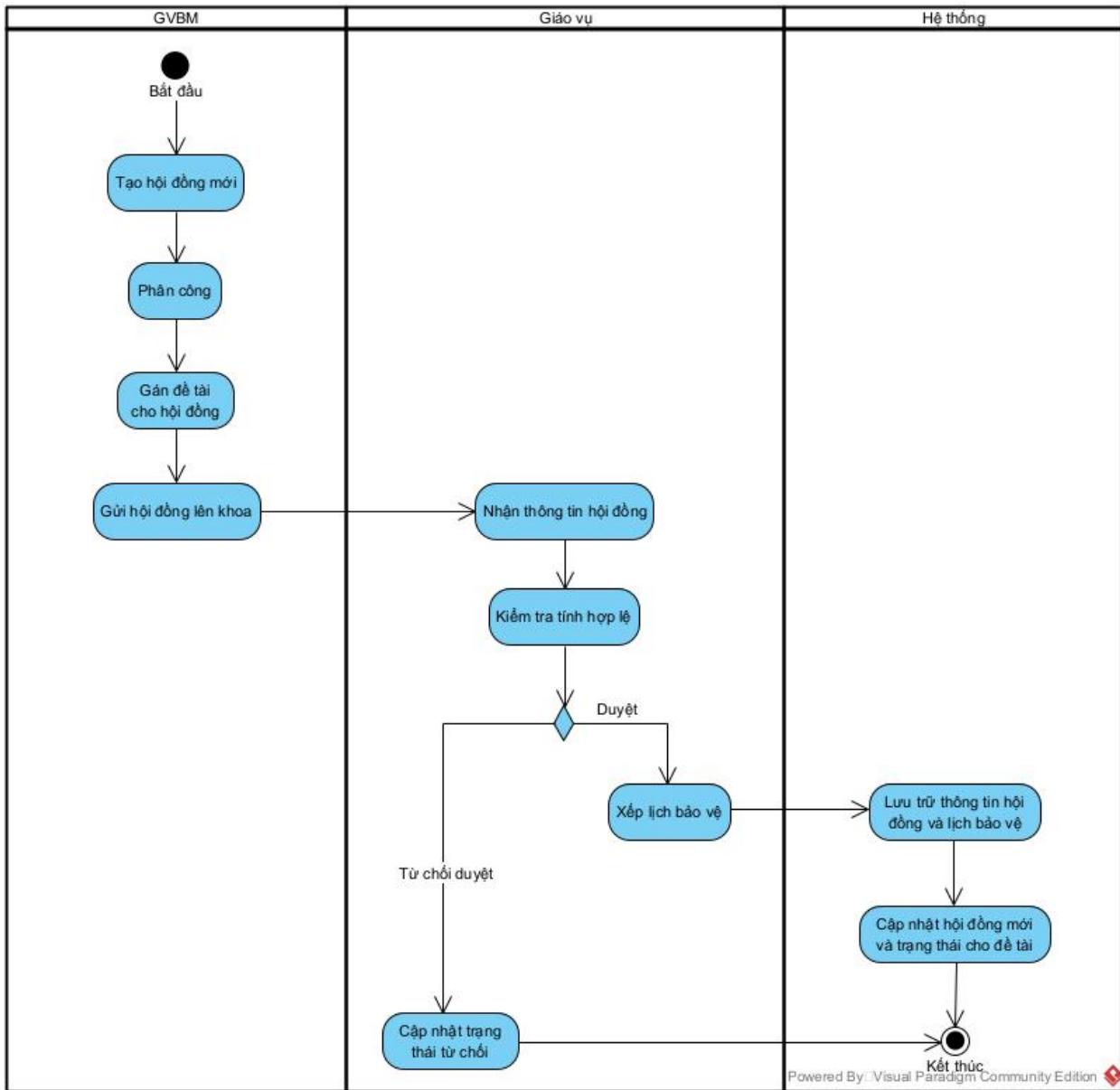
Quy trình này mô tả cách một đề tài được khởi tạo và đưa vào hệ thống. Đặc thù của hệ thống là sinh viên không tự đăng ký đề tài; thay vào đó, Giảng viên hướng dẫn (GVHD) sẽ chủ động tạo đề tài và gán sinh viên ngay từ đầu. Bộ môn sẽ duyệt và hệ thống sẽ ghi nhận một đề tài mới trên hệ thống. Quy trình tổng quát được mô tả như sơ đồ hoạt động Hình 2 dưới đây:



Hình 2. Biểu đồ hoạt động của quy trình đăng ký đề tài

b. Quy trình Tổ chức Hội đồng bảo vệ

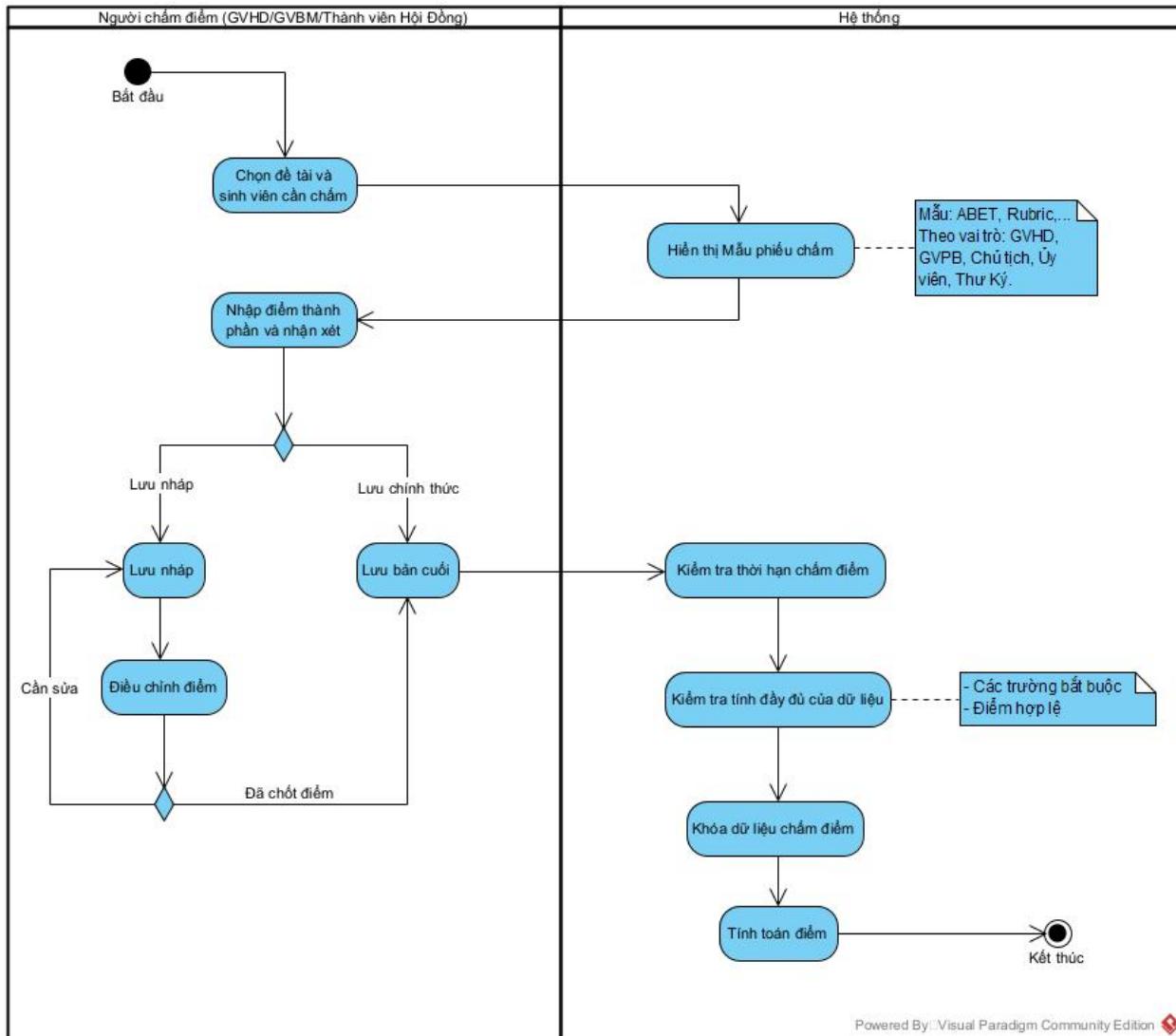
Quy trình này mô tả các bước để thiết lập một buổi bảo vệ khóa luận, bao gồm việc thành lập hội đồng chuyên môn và sắp xếp lịch trình cụ thể. Quy trình tổng quát được mô tả như sơ đồ hoạt động Hình 3 dưới đây:



Hình 3. Biểu đồ hoạt động của quy trình tổ chức hội đồng bảo vệ

c. Quy trình Chấm điểm (Grading)

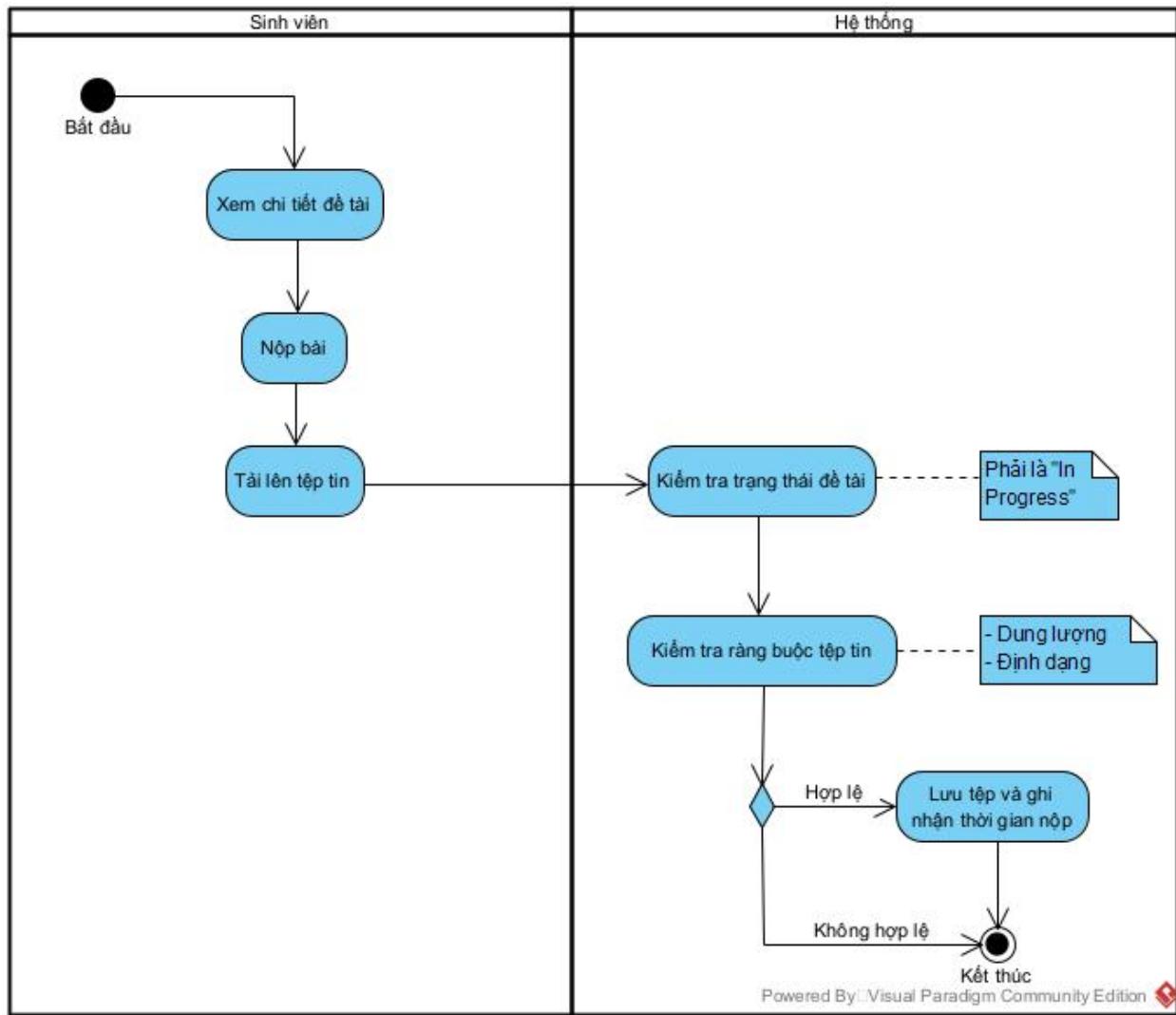
Đây là quy trình phức tạp nhất do sự đa dạng về vai trò và mẫu phiếu chấm. Hệ thống hỗ trợ cơ chế lưu nháp (Draft) và lưu chính thức (Final). Quy trình tổng quát được mô tả như sơ đồ hoạt động Hình 4 dưới đây:



Hình 4. Biểu đồ hoạt động của quy trình chấm điểm

d. Quy trình Nộp bài của Sinh viên

Quy trình này đảm bảo sinh viên nộp sản phẩm (Báo cáo, Slide, Source code) lên hệ thống để phục vụ công tác lưu trữ và chấm điểm. Quy trình tổng quát được mô tả như sơ đồ hoạt động Hình 5 dưới đây:



Hình 5. Biểu đồ hoạt động của quy trình nộp bài

5.1.2. Mô hình hóa tương tác:

5.1.2.1. Danh sách tác nhân (Actors)

Dựa trên phân tích yêu cầu người dùng , hệ thống bao gồm các tác nhân chính được trình bày trong Bảng 1:

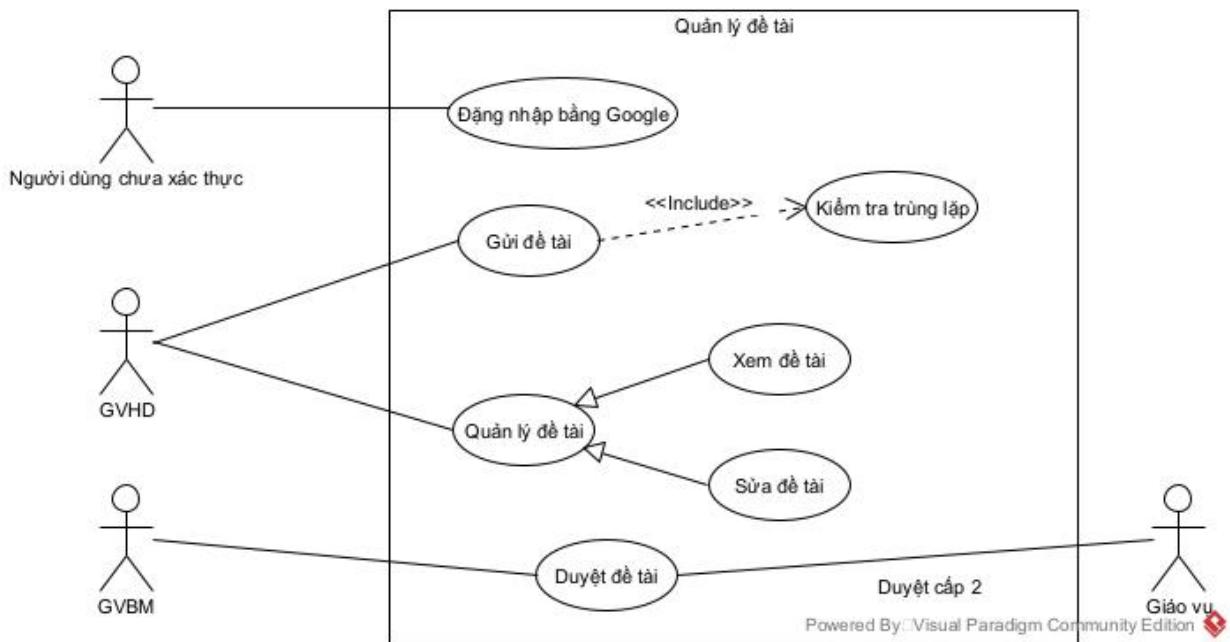
Bảng 1. Danh sách các tác nhân trong hệ thống

Tác nhân (Actor)	Mô tả vai trò

Sinh viên (Student)	Người thực hiện đề tài, nộp báo cáo và theo dõi kết quả đánh giá. Sinh viên không tự đăng ký đề tài mà được gán vào hệ thống.
Giảng viên hướng dẫn (GVHD)	Người đề xuất đề tài, hướng dẫn sinh viên thực hiện, đánh giá giữa kỳ (Pass/Fail) và chấm điểm hướng dẫn cuối kỳ.
Giảng viên bộ môn (GVBM)	Người chịu trách nhiệm phê duyệt đề tài do GVHD gửi lên và thành lập Hội đồng bảo vệ (phân công thành viên).
Giảng viên phản biện (GVPB)	Người nhận xét, đánh giá phản biện độc lập đối với các báo cáo luận văn của sinh viên.
Thành viên Hội đồng	Bao gồm Chủ tịch, Thư ký và Ủy viên. Tham gia buổi bảo vệ và chấm điểm trực tiếp trên hệ thống.
Giáo vụ (Academic Staff)	Quản trị viên nghiệp vụ: Quản lý học kỳ, nhập liệu danh sách người dùng, xếp lịch bảo vệ và quản lý điểm số tổng kết. Có thể duyệt đề tài nhưng không phải nhiệm vụ chính.
Admin (System Admin)	Quản trị viên hệ thống: Cấu hình tham số kỹ thuật, quản lý logs và giám sát vận hành (không tham gia nghiệp vụ đào tạo).

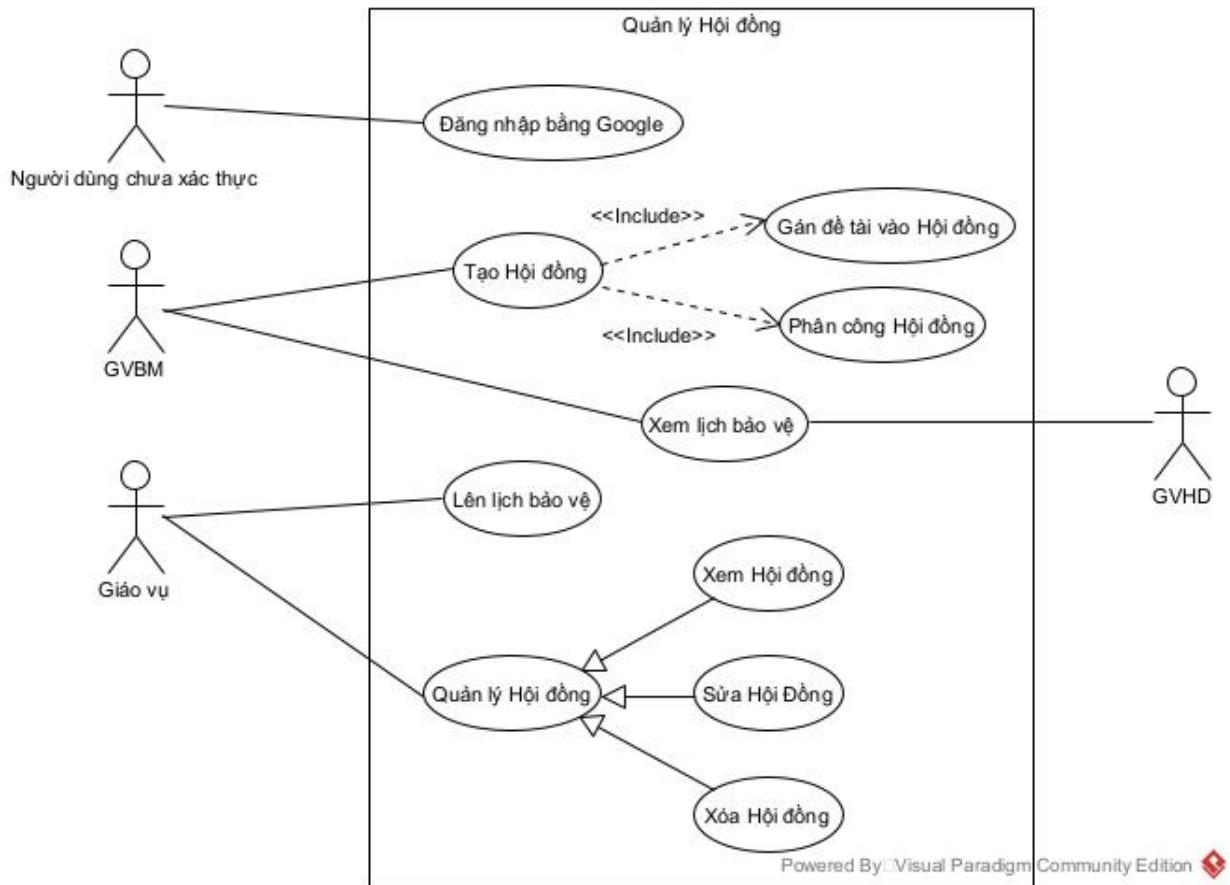
5.1.2.2. Sơ đồ Use Case (Use Case Diagram)

a. Nhóm Quản lý Đề tài: Các tính năng của nhóm này được mô tả như Hình 6:



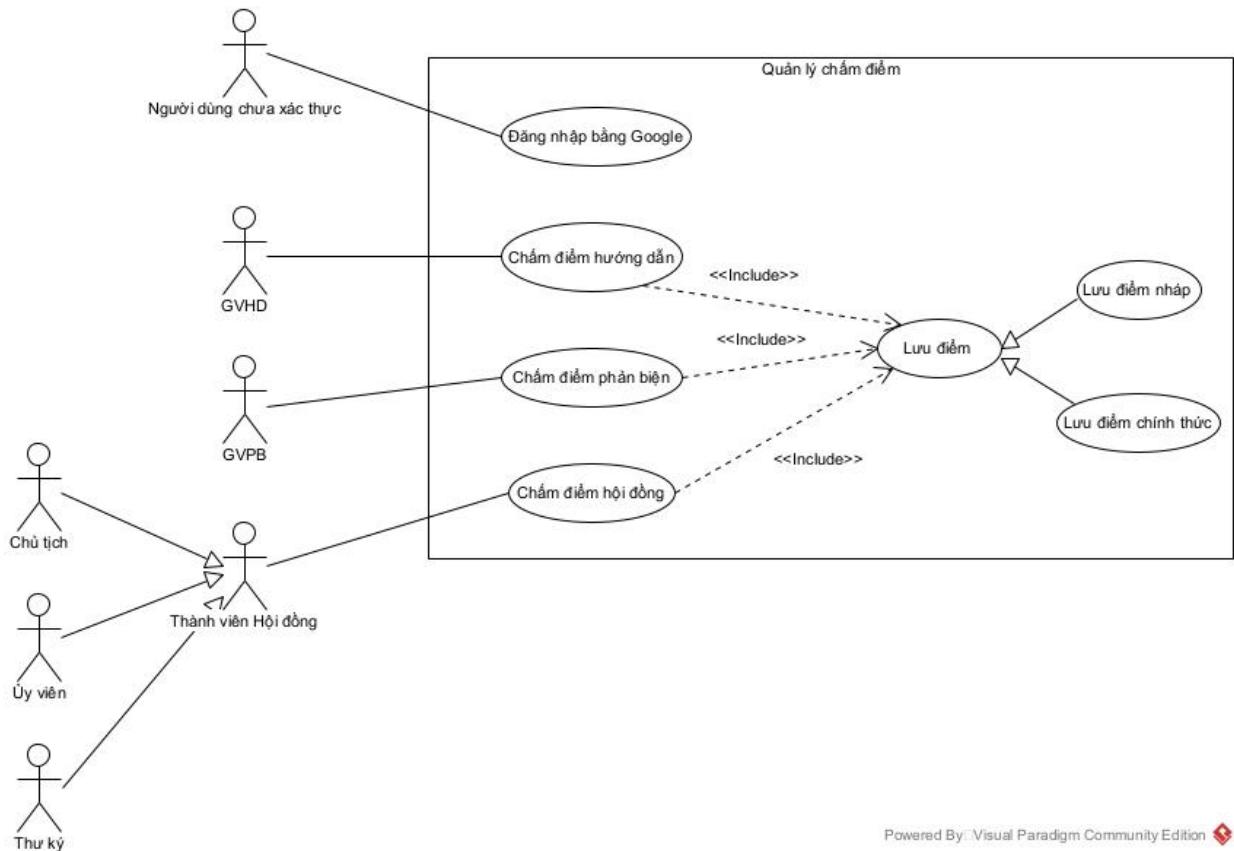
Hình 6. Use case diagram cho nhóm chức năng quản lý đề tài

b. Nhóm Hội đồng: Các tính năng của nhóm này được mô tả như Hình 7:



Hình 7. Use case diagram cho nhóm chức năng quản lý Hội đồng

c. Nhóm Chấm điểm: Các tính năng của nhóm này được mô tả như Hình 8:



Hình 8. Use case diagram cho nhóm chức năng quản lý chấm điểm

5.1.2.3. Đặc tả Use Case (Use Case Specifications)

Sau đây là các đặc tả cho các use case nghiệp vụ chính của hệ thống, bao gồm: "Gửi đề tài (GVHD)", "Duyệt đề tài" (GVBM), “Tạo hội đồng” (GVBM), "Chấm điểm hội đồng" (Thành viên Hội đồng chấm).

a) Gửi đề tài:

Đặc tả chi tiết như Bảng 2 dưới đây:

Bảng 2. Bảng đặc tả use case "Gửi đề tài"

MỤC	NỘI DUNG CHI TIẾT
1. THÔNG TIN CHUNG	

Use Case ID	UC-04
Tên Use Case	Gửi đề tài (Đăng ký đề tài mới)
Actor chính	Giảng viên hướng dẫn (GVHD)
Actor phụ	Hệ thống (System)
Mô tả ngắn	GVHD tạo mới hoặc chọn lại đề tài cũ, điền đầy đủ thông tin theo quy định và gửi lên hệ thống để chờ Bộ môn xét duyệt.
Mức độ ưu tiên	Cao
Tần suất sử dụng	Theo đợt (Đầu mỗi học kỳ)
2. ĐIỀU KIỆN	
Tiền điều kiện (Pre-condition)	1. GVHD đã đăng nhập thành công vào hệ thống. 2. Hệ thống đang trong thời gian mở đợt đăng ký đề tài của học kỳ hiện tại.
Hậu điều kiện (Post-condition)	Thành công: Thông tin đề tài được lưu vào CSDL với trạng thái " Chờ duyệt ". Sinh viên được gán tạm thời vào đề tài. Thất bại: Dữ liệu không được lưu, hệ thống hiển thị thông báo lỗi và giữ nguyên màn hình nhập liệu.
3. LUỒNG CHÍNH (Main Flow)	Bước 1: GVHD chọn chức năng " Gửi đề tài " trên side bar trong Dashboard (Trang chủ). Bước 2: Hệ thống hiển thị giao diện gửi đề tài với các trường thông tin trống. Bước 3: GVHD chọn tùy chọn " Tạo đề tài mới ". Bước 4: GVHD nhập đầy đủ các thông tin bắt buộc: - <i>Tên đề tài (Tiếng Việt & Tiếng Anh)</i> - <i>Hệ đào tạo (Chính quy/VHVL/Chất lượng cao)</i> - <i>Mã số các sinh viên thực hiện và họ tên (MSSV)</i>

	<ul style="list-style-type: none"> - Mô tả đê tài (sử dụng trình soạn thảo văn bản) - Thời gian thực hiện (Ngày bắt đầu - Ngày kết thúc) <p>Bước 5: GVHD nhấn nút "Gửi đê tài".</p> <p>Bước 6: Hệ thống kiểm tra tính hợp lệ của dữ liệu (format ngày tháng, mã số sinh viên, các trường bắt buộc).</p> <p>Bước 7: Hệ thống lưu thông tin đê tài vào CSDL với trạng thái "Chờ duyệt".</p> <p>Bước 8: Hệ thống hiển thị thông báo "Gửi đê tài thành công, vui lòng chờ duyệt".</p>
4. LUÔNG THAY THẾ / NGOẠI LỆ (Alternative / Exception Flow)	<p>A1. Sử dụng đê tài cũ (Tại bước 3):</p> <ol style="list-style-type: none"> 1. GVHD chọn tùy chọn "Chọn từ đê tài cũ". 2. Hệ thống hiển thị danh sách các đê tài GVHD đã hướng dẫn trong quá khứ. 3. GVHD chọn một đê tài từ danh sách. 4. Hệ thống tự động điền (Auto-fill) thông tin từ đê tài cũ vào form (trừ thông tin sinh viên và thời gian). 5. GVHD chỉnh sửa thông tin cần thiết (sinh viên mới, thời gian mới). 6. Tiếp tục thực hiện từ Bước 5 của Luồng chính. <p>E1. Thiếu thông tin bắt buộc (Tại bước 6):</p> <ol style="list-style-type: none"> 1. Hệ thống phát hiện thiếu Tên đê tài hoặc Hệ đào tạo. 2. Hiển thị thông báo: "<i>Vui lòng điền đầy đủ các trường bắt buộc</i>" và tô đỏ các trường thiếu. <p>E2. Sinh viên không hợp lệ (Tại bước 6):</p> <ol style="list-style-type: none"> 1. Hệ thống kiểm tra MSSV và phát hiện sinh viên không tồn tại hoặc đã có đê tài khác.

	<p>2. Hiển thị thông báo: "Mã sinh viên không hợp lệ hoặc sinh viên đã có đề tài".</p> <p>3. Yêu cầu nhập lại MSSV.</p> <p>E3. Lỗi thời gian (Tại bước 6):</p> <ol style="list-style-type: none"> 1. Hệ thống phát hiện Ngày kết thúc < Ngày bắt đầu. 2. Hiển thị thông báo lỗi và yêu cầu chọn lại ngày.
5. QUY TẮC NGHIỆP VỤ (Business Rules)	<p>1. Ngôn ngữ: Tên đề tài bắt buộc phải có cả Tiếng Anh và Tiếng Việt.</p> <p>2. Số lượng: Mỗi đề tài chỉ được gán tối đa số lượng sinh viên theo quy định (thường là 1-2 SV)</p> <p>3. Trạng thái: Tất cả đề tài khi mới gửi đều có trạng thái mặc định là "Chờ duyệt".</p> <p>4. Phạm vi: Giảng viên chỉ được xem và chọn lại các đề tài cũ do chính mình hướng dẫn.</p>
6. YÊU CẦU PHI CHỨC NĂNG (Non-functional Req)	<p>1. Hiệu năng: Thời gian tải lại thông tin đề tài cũ (Luồng A1) phải < 2 giây.</p> <p>2. Giao diện: Cung cấp bộ soạn thảo văn bản (Rich Text Editor) cho phần mô tả đề tài.</p> <p>3. Bảo mật: Dữ liệu gửi đi phải được mã hóa qua giao thức HTTPS.</p>

b) Duyệt đề tài

Đặc tả chi tiết như Bảng 3 dưới đây:

Bảng 3. Bảng đặc tả use case "Duyệt đề tài"

MỤC	NỘI DUNG CHI TIẾT
1. THÔNG TIN CHUNG	
Use Case ID	UC-05

Tên Use Case	Duyệt đề tài (Approve Topic)
Actor chính	Giảng viên bộ môn (GVBM)
Actor phụ	Hệ thống (System)
Mô tả ngắn	GVBM xem danh sách các đề tài đang chờ, xem chi tiết nội dung và đưa ra quyết định duyệt hoặc từ chối đề tài.
Mức độ ưu tiên	Cao
Tần suất sử dụng	Theo đợt (Sau đợt gửi đề tài của GVHD)
2. ĐIỀU KIỆN	
Tiền điều kiện (Pre-condition)	1. GVBM đã đăng nhập thành công vào hệ thống. 2. Có ít nhất một đề tài đang ở trạng thái " Chờ duyệt " trong hệ thống.
Hậu điều kiện (Post-condition)	Thành công (Duyệt): Trạng thái đề tài chuyển thành " Đã duyệt ". Thành công (Từ chối): Trạng thái đề tài chuyển thành " Từ chối " kèm lý do. Thất bại: Trạng thái đề tài không đổi, thông báo lỗi hiển thị.
3. LUÔNG CHÍNH (Main Flow - Duyệt chi tiết)	Bước 1: GVBM chọn chức năng " Quản lý đề tài " trên thanh menu. Bước 2: Hệ thống hiển thị danh sách các đề tài thuộc bộ môn quản lý đang ở trạng thái " Chờ duyệt " (Mặc định). Bước 3: GVBM nhấn vào tên một đề tài cụ thể để xem. Bước 4: Hệ thống hiển thị trang chi tiết đề tài (Tên, Mô tả, SV thực hiện, GVHD...). Bước 5: GVBM xem xét nội dung và nhấn nút " Duyệt ". Bước 6: Hệ thống cập nhật trạng thái đề tài sang " Đã duyệt ".

	Bước 7: Hệ thống hiển thị thông báo " <i>Duyệt để tài thành công</i> " và quay lại danh sách.
4. LUỒNG THAY THẾ / NGOẠI LỆ (Alternative / Exception Flow)	<p>A1. Duyệt nhanh (Quick Approve) - Tại bước 2:</p> <ol style="list-style-type: none"> GVBM nhấn vào biểu tượng Dấu Tick (✓) ngay trên dòng của đề tài trong danh sách. Hệ thống hiển thị hộp thoại xác nhận: "<i>Bạn có chắc muốn duyệt để tài này?</i>". GVBM chọn "Đồng ý". Hệ thống cập nhật trạng thái sang "Đã duyệt" và làm mới danh sách. <p>A2. Từ chối nhanh (Quick Reject) - Tại bước 2:</p> <ol style="list-style-type: none"> GVBM nhấn vào biểu tượng Dấu X (✗) ngay trên dòng của đề tài. Hệ thống hiển thị popup yêu cầu nhập "Lý do từ chối". GVBM nhập lý do và nhấn "Xác nhận". Hệ thống cập nhật trạng thái sang "Từ chối" và lưu lý do. <p>A3. Từ chối chi tiết (Tại bước 5 của Luồng chính):</p> <ol style="list-style-type: none"> Sau khi xem chi tiết, GVBM nhấn nút "Từ chối". Hệ thống yêu cầu nhập lý do từ chối. GVBM nhập lý do và xác nhận. Hệ thống cập nhật trạng thái và quay lại danh sách. <p>E1. Dữ liệu thay đổi (Concurrency):</p> <ol style="list-style-type: none"> Trong lúc GVBM đang xem, đề tài đã bị xóa hoặc thay đổi bởi GVHD. Khi bấm Duyệt/Từ chối, hệ thống báo lỗi: "<i>Dữ liệu không còn tồn tại hoặc đã thay đổi</i>".

	3. Tải lại danh sách mới nhất.
5. QUY TẮC NGHIỆP VỤ (Business Rules)	<p>1. Phạm vi quản lý: GVBM chỉ được nhìn thấy và duyệt các đề tài thuộc Bộ môn/Khoa mà mình quản lý.</p> <p>2. Lý do từ chối: Khi thực hiện hành động "Từ chối", hệ thống bắt buộc phải yêu cầu nhập lý do để phản hồi cho GVHD.</p> <p>3. Trùng lặp: Hệ thống nên hiển thị cảnh báo nếu phát hiện đề tài có độ trùng lặp cao (nếu tích hợp module check đạo văn) ngay tại giao diện chi tiết.</p>
6. YÊU CẦU PHI CHỨC NĂNG (Non-functional Req)	<p>1. Phản hồi: Thao tác duyệt/từ chối nhanh phải được phản hồi trên giao diện (biến mất khỏi danh sách chờ) trong vòng < 1 giây.</p> <p>2. Giao diện: Nút Duyệt (Xanh) và Từ chối (Đỏ) phải được bố trí rõ ràng, tránh nhầm lẫn.</p>

c) Tạo Hội Đồng

Đặc tả chi tiết như Bảng 4 dưới đây:

Bảng 4. Bảng đặc tả use case "Tạo Hội Đồng"

MỤC	NỘI DUNG CHI TIẾT
1. THÔNG TIN CHUNG	
Use Case ID	UC-10
Tên Use Case	Tạo hội đồng bảo vệ (Create Defense Council)
Actor chính	Giảng viên bộ môn (GVBM)
Actor phụ	Hệ thống (System)
Mô tả ngắn	GVBM tạo mới một hội đồng, thiết lập tên, phân công thành viên (Chủ tịch, Thư ký, Ủy viên và Giảng viên phản biện nếu cần) và gán các đề tài sẽ bảo vệ.
Mức độ ưu tiên	Cao

Tần suất sử dụng	Theo đợt (Trước đợt bảo vệ của mỗi học kỳ)
2. ĐIỀU KIỆN	
Tiền điều kiện (Pre-condition)	<p>1. GVBM đã đăng nhập thành công.</p> <p>2. Danh sách Giảng viên và Đề tài (trạng thái "Đang thực hiện") của học kỳ hiện tại đã sẵn sàng.</p>
Hậu điều kiện (Post-condition)	<p>Thành công: Hội đồng được lưu với trạng thái "Chờ duyệt". Danh sách thành viên (bao gồm cả phản biện nếu có) và đề tài được liên kết với hội đồng.</p> <p>Thất bại: Dữ liệu không thay đổi.</p>
3. LUỒNG CHÍNH (Main Flow)	<p>Bước 1: GVBM truy cập chức năng "Quản lý hội đồng" > nhấn nút "Tạo hội đồng".</p> <p>Bước 2: Hệ thống hiển thị form tạo mới.</p> <p>Bước 3: GVBM nhập Tên hội đồng.</p> <p>Bước 4: GVBM thực hiện Phân công thành viên:</p> <ul style="list-style-type: none"> - Tìm kiếm giảng viên. - Chọn giảng viên và gán vai trò: <i>Chủ tịch, Thư ký, Ủy viên</i>. - (Tùy chọn) Gán thêm vai trò <i>Phản biện</i> nếu hội đồng này dùng cho Giai đoạn 2 (Luận văn tốt nghiệp) hoặc theo yêu cầu chuyên môn. - Lặp lại cho đến khi đủ cơ cấu. <p>Bước 5: GVBM thực hiện Chọn đề tài:</p> <ul style="list-style-type: none"> - Hệ thống hiển thị danh sách đề tài đủ điều kiện. - GVBM tích chọn các đề tài gán cho hội đồng. <p>Bước 6: GVBM nhấn nút "Lưu và Gửi duyệt".</p> <p>Bước 7: Hệ thống kiểm tra quy tắc nghiệp vụ (đủ 3 vai trò cốt lõi: Chủ tịch, Thư ký, Ủy viên).</p>

	Bước 8: Hệ thống lưu thông tin và hiển thị thông báo "Tạo hội đồng thành công, đang chờ Giáo vụ duyệt".
4. LUỒNG THAY THẾ / NGOẠI LỆ (Alternative / Exception Flow)	<p>A1. Không phân công Phản biện (Giai đoạn 1):</p> <p>Tại bước 4, GVBM chỉ chọn Chủ tịch, Thư ký và Ủy viên, bỏ qua vai trò Phản biện. Hệ thống vẫn chấp nhận và cho phép lưu bình thường (vì Phản biện là role optional).</p> <p>E1. Vi phạm cấu trúc cứng (Tại bước 7):</p> <ol style="list-style-type: none"> 1. Hệ thống phát hiện thiếu một trong 3 vai trò bắt buộc (Chủ tịch/Thư ký/Ủy viên) hoặc tổng số thành viên < 3. 2. Báo lỗi: "<i>Hội đồng bắt buộc phải có tối thiểu Chủ tịch, Thư ký và Ủy viên</i>". 3. Yêu cầu bổ sung. <p>E2. Trùng lặp thành viên:</p> <p>GVBM gán một giảng viên giữ 2 vai trò trong cùng 1 hội đồng (Ví dụ: Vừa là Chủ tịch vừa là Phản biện). Hệ thống báo lỗi xung đột vai trò.</p>
5. QUY TẮC NGHIỆP VỤ (Business Rules)	<p>1. Cấu trúc Cốt lõi (Bắt buộc): Mỗi hội đồng phải có tối thiểu 03 người đảm nhiệm 3 vai trò: Chủ tịch, Thư ký, Ủy viên.</p> <p>2. Vai trò Phản biện (Tùy chọn):</p> <ul style="list-style-type: none"> - Có thể gán hoặc không gán tùy theo giai đoạn (GD1 thường không cần, GD2 thường cần). - Nếu được gán, Phản biện cũng được tính là một thành viên chính thức của hội đồng. <p>3. Ràng buộc Duy nhất: Mỗi hội đồng chỉ có 01 Chủ tịch. Các vai trò khác (Thư ký, Ủy viên) có thể có số lượng > 1.</p>

	4. Quyền chấm: Tất cả thành viên được gán vào hội đồng (bao gồm cả Phản biện nếu có mặt) đều có quyền chấm điểm trên hệ thống khi bảo vệ diễn ra.
6. YÊU CẦU PHI CHỨC NĂNG	1. Linh hoạt: Giao diện phân công cần cho phép thêm/bớt dòng linh hoạt để GVBM dễ dàng thêm nhiều Ủy viên hoặc nhiều Thư ký nếu cần.

d) Chấm điểm Hội Đồng

Đặc tả chi tiết như Bảng 5 dưới đây:

Bảng 5. Bảng đặc tả use case "Chấm điểm Hội Đồng"

MỤC	NỘI DUNG CHI TIẾT
1. THÔNG TIN CHUNG	
Use Case ID	UC-12
Tên Use Case	Chấm điểm hội đồng (Grade by Council)
Actor chính	Thành viên Hội đồng (Chủ tịch, Thư ký, Ủy viên, Phản biện)
Actor phụ	Hệ thống (System)
Mô tả ngắn	Tại buổi bảo vệ, từng thành viên hội đồng truy cập hệ thống để nhập điểm và nhận xét cho sinh viên dựa trên phần trình bày và trả lời câu hỏi.
Mức độ ưu tiên	Cao
Tần suất sử dụng	Theo đợt (Trong ngày bảo vệ)
2. ĐIỀU KIỆN	
Tiền điều kiện (Pre-condition)	<ol style="list-style-type: none"> Hội đồng đã được thành lập và Lịch bảo vệ đã được duyệt (UC10, UC11). Thành viên hội đồng đã đăng nhập vào hệ thống.

	3. Thời gian thực tế nằm trong khung giờ cho phép bảo vệ.
Hậu điều kiện (Post-condition)	<p>Thành công: Điểm số của từng thành viên được lưu vào CSDL. Khi tất cả thành viên đã chấm xong (Final), hệ thống tính điểm tổng kết.</p> <p>Thất bại: Điểm số không được lưu, hiển thị thông báo lỗi kết nối hoặc dữ liệu.</p>
3. LUÔNG CHÍNH (Main Flow)	<p>Bước 1: Thành viên hội đồng chọn chức năng "Hội đồng chấm điểm".</p> <p>Bước 2: Hệ thống hiển thị danh sách các sinh viên/đề tài thuộc hội đồng đang diễn ra.</p> <p>Bước 3: Thành viên chọn sinh viên đang bảo vệ và nhấn "Chấm điểm".</p> <p>Bước 4: Hệ thống hiển thị Phiếu chấm điểm điện tử (Rubric hoặc ABET) theo quy định của Khoa.</p> <p>Bước 5: Thành viên nhập điểm cho từng tiêu chí (Ví dụ: Trình bày, Nội dung, Trả lời câu hỏi) và nhập Nhận xét chung.</p> <p>Bước 6: Thành viên kiểm tra lại và nhấn nút "Lưu chính thức" (Submit Final).</p> <p>Bước 7: Hệ thống kiểm tra tính hợp lệ (điểm trong thang 0-10, không bỏ trống tiêu chí).</p> <p>Bước 8: Hệ thống hiển thị hộp thoại xác nhận: "<i>Sau khi lưu chính thức sẽ không thể chỉnh sửa. Bạn có chắc chắn?</i>".</p> <p>Bước 9: Thành viên chọn "Đồng ý".</p> <p>Bước 10: Hệ thống lưu điểm, khóa phiếu chấm của thành viên đó và hiển thị thông báo thành công.</p>
4. LUÔNG THAY THẾ / NGOẠI LỆ	A1. Lưu nháp (Save Draft) - Tại bước 6:

(Alternative / Exception Flow)	<p>1. Thành viên muốn lưu điểm tạm thời để hỏi thêm sinh viên.</p> <p>2. Nhấn nút "Lưu nháp".</p> <p>3. Hệ thống lưu dữ liệu tạm, trạng thái là "<i>Draft</i>".</p> <p>4. Thành viên có thể quay lại sửa điểm sau đó.</p> <p>E1. Vi phạm thang điểm (Tại bước 7):</p> <ol style="list-style-type: none"> 1. Hệ thống phát hiện điểm nhập vào < 0 hoặc > 10. 2. Hiển thị thông báo lỗi: "<i>Điểm số không hợp lệ</i>". 3. Yêu cầu nhập lại. <p>E2. Hết thời gian bảo vệ (Tại bước 1):</p> <ol style="list-style-type: none"> 1. Thành viên truy cập khi thời gian cho phép nhập điểm đã đóng (theo cấu hình của Giáo vụ). 2. Hệ thống hiển thị thông báo: "<i>Đã hết thời hạn chấm điểm</i>" và khóa chức năng nhập liệu (chỉ xem).
5. QUY TẮC NGHIỆP VỤ (Business Rules)	<ol style="list-style-type: none"> 1. Độc lập: Điểm số của các thành viên là độc lập, thành viên này không nhìn thấy điểm của thành viên kia khi chưa công bố. 2. Điểm liệt: Nếu điểm tổng kết < 5.0 (hoặc theo quy chế), hệ thống ghi nhận trạng thái là "Không đạt". 3. Quyền Chủ tịch/Thư ký: Thư ký có thể có thêm quyền ghi nhận biên bản hội đồng (nếu có yêu cầu) ngoài việc chấm điểm cá nhân.
6. YÊU CẦU PHI CHÚC NĂNG (Non-functional Req)	<ol style="list-style-type: none"> 1. Tính sẵn sàng: Hệ thống phải hoạt động ổn định tuyệt đối trong thời gian diễn ra bảo vệ, tránh tình trạng mất kết nối làm gián đoạn buổi lễ. 2. Giao diện: Tương thích tốt trên máy tính bảng (iPad/Tablet) để giảng viên thuận tiện thao tác tại bàn hội đồng.

5.1.2.4. Danh sách tóm tắt Use Case

Các use case nghiệp vụ trọng tâm của hệ thống được liệt kê trong bảng sau, không bao gồm toàn bộ các CRUD của một tính năng cụ thể, hay các thao tác đăng xuất, quên mật khẩu. Chi tiết như Bảng 6 dưới đây:

Bảng 6. Danh sách các use case của hệ thống

ID	Tên Use Case	Mô tả ngắn gọn
UC01	Đăng nhập	Người dùng đăng nhập vào hệ thống qua tài khoản nội bộ hoặc Google OAuth.
UC02	Tạo học kỳ	Giáo vụ khởi tạo học kỳ mới để bắt đầu quy trình quản lý.
UC03	Nhập danh sách	Giáo vụ tải lên danh sách Sinh viên/Giảng viên từ file Excel.
UC04	Gửi đề tài	GVHD tạo đề xuất đề tài mới và nhập thông tin sinh viên dự kiến thực hiện.
UC05	Duyệt đề tài	GVBM xem xét và phê duyệt hoặc từ chối các đề tài do GVHD gửi.
UC06	Nộp bài	Sinh viên tải lên các file báo cáo, slide thuyết trình vào hệ thống.
UC07	Đánh giá giữa kỳ	GVHD đánh giá kết quả làm việc của sinh viên (Pass/Fail).
UC08	Chấm điểm hướng dẫn	GVHD nhập điểm số và nhận xét cuối kỳ cho sinh viên.
UC09	Chấm điểm phản biện	GVPB nhập điểm và tải lên phiếu nhận xét phản biện.
UC10	Tạo Hội đồng	GVBM tạo danh sách hội đồng và gán các vai trò (Chủ tịch, Thư ký, Phản biện...).
UC11	Lên lịch bảo vệ	Giáo vụ duyệt và sắp xếp thời gian và phòng bảo vệ cho các Hội đồng.
UC12	Chấm điểm hội đồng	Các thành viên hội đồng nhập điểm trực tiếp tại buổi bảo vệ.

UC13	Xem kết quả	Sinh viên xem điểm số chi tiết và trạng thái đê tài.
UC14	Xuất báo cáo	Giáo vụ, Giảng viên xuất các file thông tin hoặc điểm.

5.2. Giải pháp công nghệ

Để đáp ứng các yêu cầu khắt khe về hiệu năng, khả năng mở rộng và tính toàn vẹn dữ liệu đã phân tích ở trên, nhóm thực hiện đã áp dụng các giải pháp công nghệ tiên tiến, cụ thể là kiến trúc Microservices và chiến lược lưu trữ đa hình (Polyglot Persistence).

5.2.1. Kiến trúc Microservices

Thay vì sử dụng kiến trúc nguyên khối (Monolith) truyền thống, hệ thống TMS được thiết kế và xây dựng dựa trên kiến trúc Microservices. Hệ thống được chia nhỏ thành 06 dịch vụ (services) backend chính, bao gồm: Thesis (Quản lý đê tài), Role (Quản lý vai trò), User (Quản lý người dùng), Council (Quản lý hội đồng), File (Quản lý tài liệu), và Academic (Quản lý học vụ).

Các lý do chính cho việc lựa chọn kiến trúc này bao gồm:

- Khả năng mở rộng linh hoạt (Scalability):** Hệ thống cho phép mở rộng tài nguyên cho từng dịch vụ riêng biệt dựa trên nhu cầu thực tế. Ví dụ, trong tuần lễ nộp bài, dịch vụ *File* và *Thesis* sẽ chịu tải cao và có thể được scale-up độc lập mà không cần gia tăng tài nguyên cho dịch vụ *User* hay *Role*.
- Tính độc lập và cô lập lỗi (Fault Isolation):** Mỗi dịch vụ hoạt động trên một quy trình riêng biệt và quản lý cơ sở dữ liệu riêng. Nếu một dịch vụ gặp sự cố (ví dụ: lỗi kết nối tại *File Service*), nó sẽ không gây ra hiệu ứng domino làm sập toàn bộ hệ thống, đảm bảo các chức năng khác như tra cứu thông tin vẫn hoạt động bình thường.
- Tối ưu hóa giao tiếp nội bộ:** Các dịch vụ được phát triển bằng ngôn ngữ **Golang** và giao tiếp với nhau thông qua giao thức **gRPC** thay vì RESTful API truyền

thống. Điều này giúp giảm độ trễ (latency) xuống mức thấp nhất và tăng cường hiệu suất xử lý giao dịch nội bộ giữa các phân hệ.

5.2.2. Chiến lược đa lưu trữ (Polyglot Persistence)

Hệ thống áp dụng chiến lược Polyglot Persistence, kết hợp sức mạnh của Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) và phi quan hệ (NoSQL) để tối ưu hóa việc lưu trữ cho từng loại dữ liệu đặc thù. Cụ thể, hệ thống sử dụng kết hợp MySQL và MongoDB.

a. Sử dụng MySQL (Relational Database) MySQL được chọn làm kho lưu trữ chính cho các dữ liệu nghiệp vụ cốt lõi.

- **Lý do lựa chọn:** Các dữ liệu như Sinh viên, Đè tài, Điểm số và Hội đồng có mối quan hệ ràng buộc chặt chẽ và yêu cầu tính toàn vẹn dữ liệu tuyệt đối (ACID). MySQL đảm bảo rằng các giao dịch như "Gán sinh viên vào đè tài" hoặc "Chấm điểm" được thực hiện chính xác, không xảy ra tình trạng dữ liệu rác hay mất đồng bộ.
- **Triển khai:** Hệ thống triển khai 06 instances MySQL riêng biệt cho 6 microservices, đảm bảo nguyên tắc "Database-per-service" để tránh sự phụ thuộc lẫn nhau giữa các dịch vụ.

b. Sử dụng MongoDB (NoSQL Database) MongoDB được sử dụng bổ trợ để xử lý các dữ liệu phi cấu trúc hoặc bán cấu trúc.

- **Lý do lựa chọn:**
 - **Lưu trữ Logs và Audit Trail:** Hệ thống cần ghi lại hàng triệu bản ghi logs sự kiện (Event Logs) để giám sát và truy vết. MongoDB với khả năng ghi (Write) tốc độ cao là lựa chọn tối ưu cho tác vụ này.
 - **Dữ liệu động:** Các cấu hình hệ thống (System Config) hoặc các quy trình động (Workflow) thường xuyên thay đổi cấu trúc. Dạng lưu trữ Document (JSON/BSON) của MongoDB cho phép linh hoạt thay đổi schema mà không cần downtime hay migrate dữ liệu phức tạp như SQL.

c. Các thành phần lưu trữ hỗ trợ Ngoài ra, giải pháp công nghệ còn bao gồm:

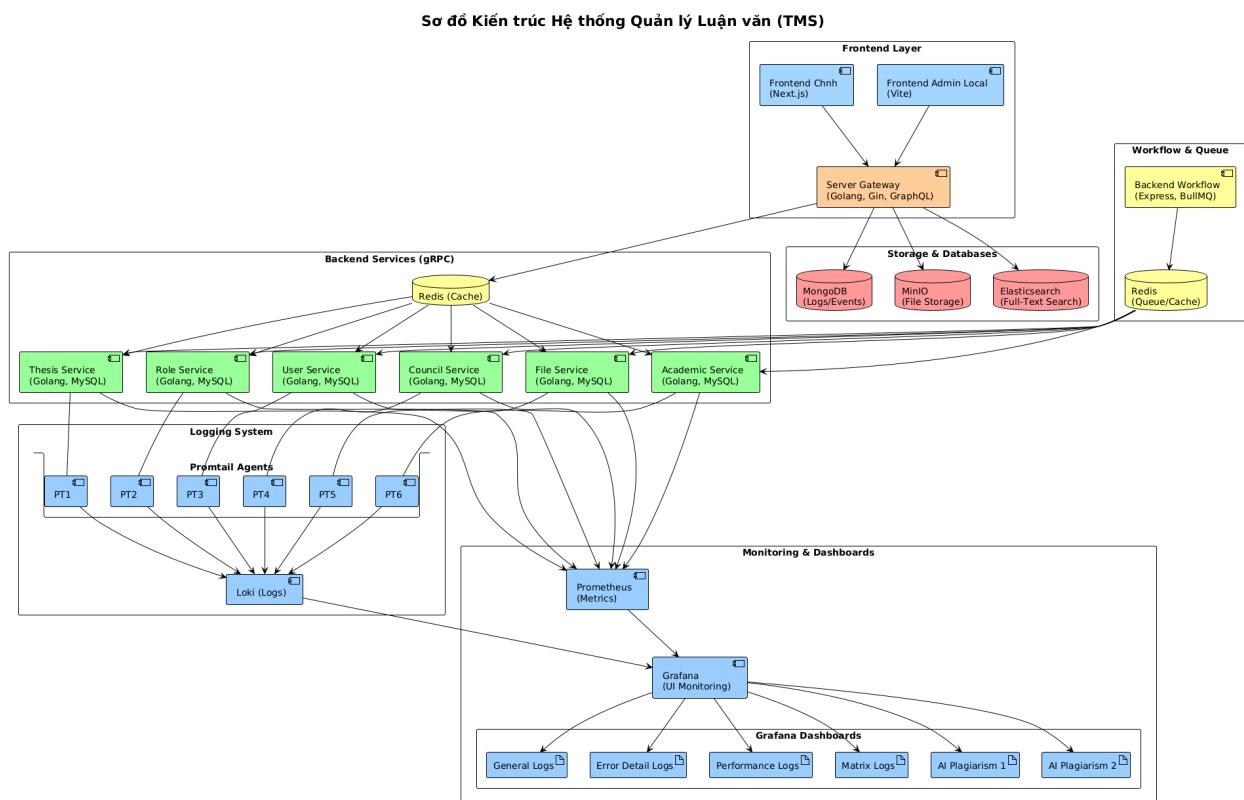
- **Redis:** Đóng vai trò là bộ nhớ đệm (Cache) và hàng đợi tin nhắn (Message Queue) để xử lý các tác vụ bất đồng bộ, giảm tải cho database chính.
- **MinIO:** Giải pháp lưu trữ đối tượng (Object Storage) dùng để chứa các file báo cáo kích thước lớn, tách biệt hoàn toàn dữ liệu nhị phân (Blob) khỏi cơ sở dữ liệu để đảm bảo hiệu năng truy vấn.

5.3. Thiết kế

5.3.1. Kiến trúc hệ thống

Hệ thống TMS được thiết kế dựa trên mô hình Microservices, trong đó các thành phần được phân tách rõ ràng để đảm bảo tính chuyên biệt và khả năng mở rộng. Luồng dữ liệu (Data Flow) trong hệ thống được tổ chức theo quy trình chặt chẽ từ Client đến lớp lưu trữ dữ liệu.

Sơ đồ kiến trúc tổng thể được trình bày như Hình 9:



Hình 9. Sơ đồ kiến trúc hệ thống TMS

Mô tả luồng xử lý dữ liệu:

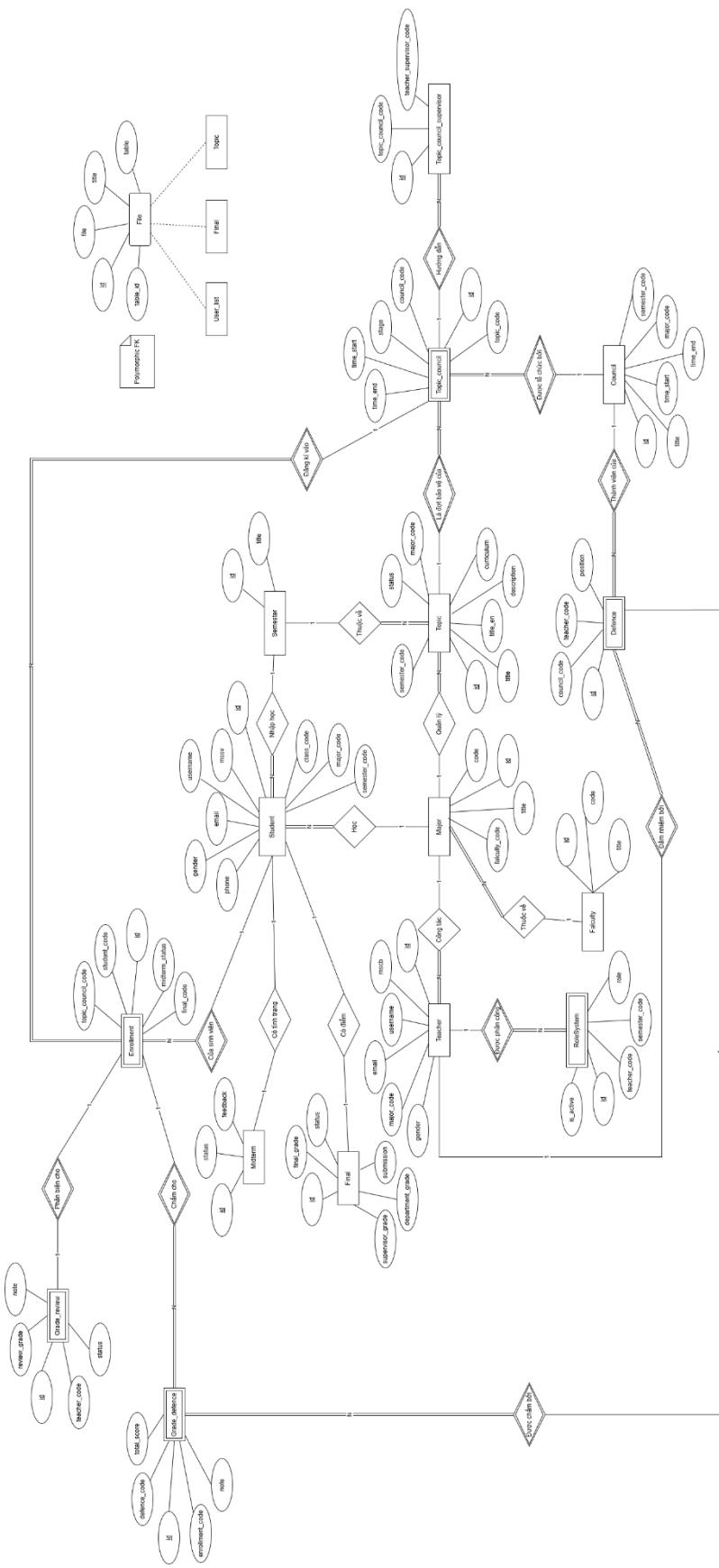
1. **Client Layer (Frontend):** Người dùng (Sinh viên, Giảng viên, Giáo vụ) tương tác với hệ thống thông qua giao diện Web được xây dựng bằng **Next.js** (cho người dùng cuối) và **Vite** (cho quản trị viên).
2. **API Gateway:** Mọi yêu cầu từ Client đều đi qua **Server Gateway** (viết bằng Golang/Gin). Tại đây, hệ thống thực hiện xác thực (Authentication) và điều hướng yêu cầu. Gateway cung cấp API dưới dạng **GraphQL** để tối ưu hóa dữ liệu trả về cho Frontend.
3. **Communication Layer (gRPC):** Gateway không xử lý nghiệp vụ trực tiếp mà gọi đến các dịch vụ Backend thông qua giao thức **gRPC**. Đây là kênh giao tiếp nội bộ hiệu năng cao, đảm bảo độ trễ thấp giữa các services.
4. **Microservices Layer:** Hệ thống bao gồm 06 dịch vụ lõi xử lý nghiệp vụ chuyên biệt:
 - Thesis Service: Xử lý logic liên quan đến đề tài.
 - Council Service: Quản lý hội đồng và chấm điểm.
 - User Service, Role Service, File Service, Academic Service.
5. **Data Persistence Layer:** Mỗi dịch vụ sẽ thao tác với cơ sở dữ liệu riêng của mình (MySQL) để đảm bảo tính độc lập. Các tác vụ ghi log hoặc cấu hình động được đẩy về MongoDB.
6. **Asynchronous Processing:** Các tác vụ nặng (như kiểm tra trùng lặp, gửi email) được Gateway đẩy vào hàng đợi **Redis Queue**, sau đó **Backend Workflow (Worker)** sẽ lấy ra xử lý ngầm để không làm chặn luồng chính.

5.3.2. Thiết kế cơ sở dữ liệu

Hệ thống sử dụng chiến lược Polyglot Persistence, kết hợp giữa CSDL quan hệ và phi quan hệ.

a. Thiết kế CSDL Quan hệ (MySQL)

Cơ sở dữ liệu MySQL được sử dụng để lưu trữ các thực thể nghiệp vụ cốt lõi có cấu trúc chặt chẽ. Mô hình thực thể - quan hệ (ERD) ý niệm được thiết kế để đảm bảo tính nhất quán dữ liệu (Data Consistency) như hình Hình 9 dưới đây:



Hình 10. Sơ đồ thiết kế ý niệm

- **Các nhóm thực thể chính:**
 - **Nhóm Tổ chức & Người dùng:** Bao gồm các bảng Student, Teacher, Faculty (Khoa), Major (Ngành) và RoleSystem để quản lý phân quyền.
 - **Nhóm Đề tài & Hội đồng:** Trung tâm là bảng Topic kết nối với Semester (Học kỳ). Bảng Council và Topic_council quản lý việc gán đề tài vào hội đồng bảo vệ.
 - **Nhóm Điểm số:** Sử dụng các bảng Midterm, Grade_review, Grade_defence và Final để lưu trữ điểm số qua từng giai đoạn đánh giá.
- **Cơ chế lưu trữ:** Mỗi Microservice sẽ sở hữu một database riêng (Database-per-service), ví dụ: Service Thesis quản lý các bảng liên quan đến Topic, Service Council quản lý bảng Council.

b. Thiết kế CSDL Phi quan hệ (MongoDB)

MongoDB được sử dụng để lưu trữ các dữ liệu có cấu trúc linh động hoặc tần suất ghi lớn. Các Collection chính bao gồm:

- **Collection workflows:** Lưu trữ định nghĩa các quy trình nghiệp vụ động, cho phép cấu hình các bước xử lý mà không cần sửa code (Orchestration logic).
- **Collection cronjob:** Quản lý lịch trình các tác vụ tự động (Scheduled Jobs) như sao lưu dữ liệu, cập nhật trạng thái đề tài.
- **Collection services:** Đóng vai trò là Service Registry, quản lý danh sách và trạng thái sức khỏe (Health status) của các microservices trong hệ thống.
- **Collection logs/event_logs:** Lưu vết toàn bộ thao tác của người dùng để phục vụ công tác giám sát và hậu kiểm.

5.3.3. Thiết kế chi tiết bên trong

Cấu trúc mã nguồn (Source Code Structure) được tổ chức theo mô hình module hóa để thuận tiện cho việc bảo trì và mở rộng.

a. Cấu trúc thư mục dự án (Project Structure) Dựa trên cây thư mục thực tế :

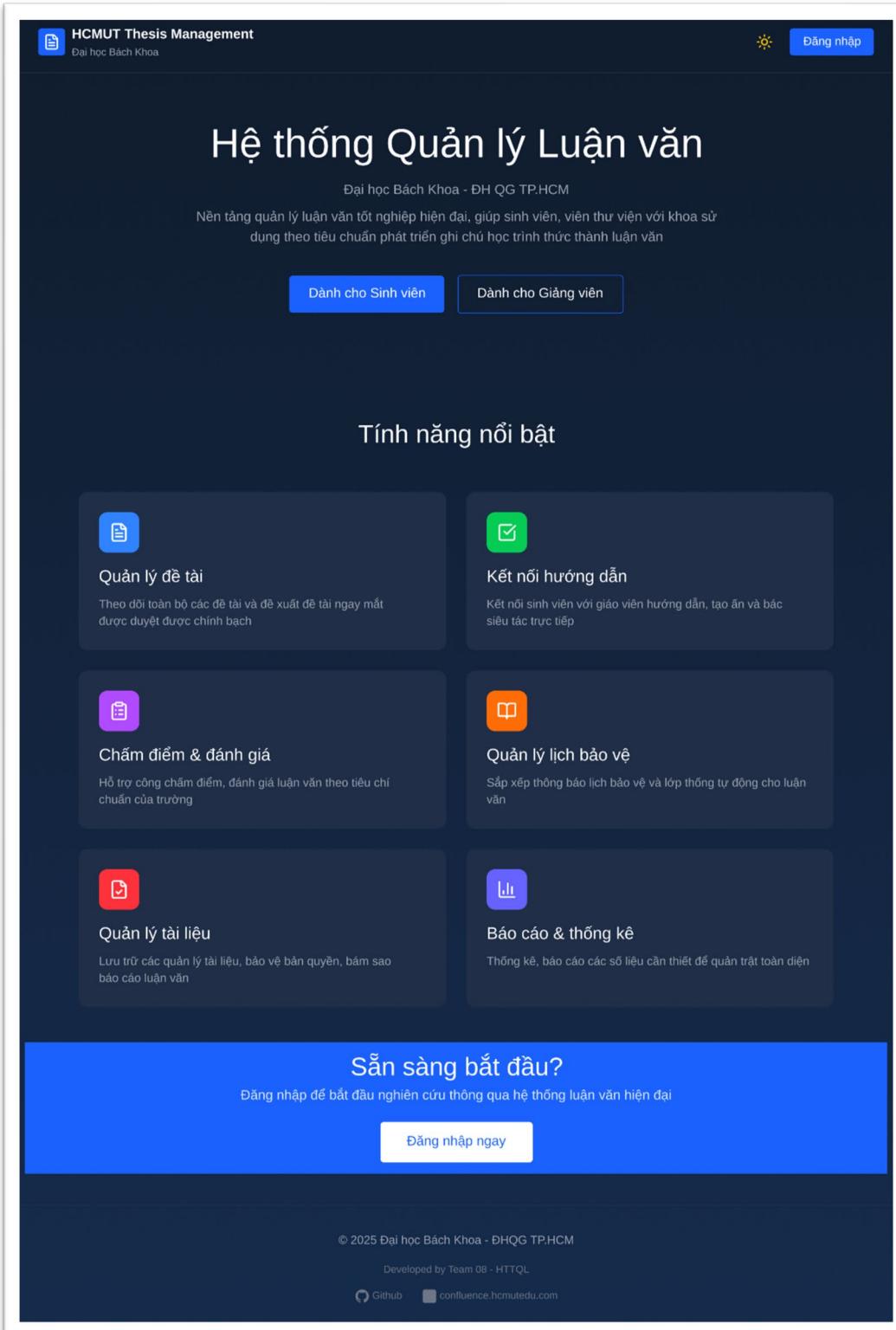
- BE_core/: Chứa mã nguồn Backend chính (Golang).
 - /api: Chứa các định nghĩa API (Proto files, GraphQL schemas).
 - /cmd: Điểm khởi chạy của các services.
 - /internal: Chứa logic nghiệp vụ cốt lõi, được chia tách theo từng domain (Thesis, User, Council...).
 - /pkg: Các thư viện dùng chung (Utility) như Logger, Database Connector.
- BE_core/queue: Chứa mã nguồn cho hệ thống xử lý hàng đợi và tác vụ ngầm.
- FE_core/: Mã nguồn Frontend (Next.js).

b. Các module chức năng chính

- **Server Gateway Module:**
 - Sử dụng framework **Gin** để routing HTTP và middleware xác thực (JWT Auth).
 - Tích hợp **GraphQL Resolver** để ánh xạ các query từ frontend thành các lệnh gọi gRPC xuống backend services.
- **Queue System (Backend Workflow):**
 - Được xây dựng trên **Node.js/Express** kết hợp với thư viện **BullMQ**.
 - Chịu trách nhiệm lắng nghe (Listen) các sự kiện từ Redis Pub/Sub và thực thi các tác vụ nền như: Tạo file PDF báo cáo, gửi email, kiểm tra đạo văn.

5.3.4. Thiết kế giao diện:

Giao diện người dùng được thiết kế theo phong cách hiện đại, tập trung vào trải nghiệm người dùng (UX) với các Dashboard trực quan. Giao diện được thiết kế bằng Figma và được hiện thực với độ chính xác giống thiết kế ban đầu gần như 100%. Toàn bộ giao diện của ứng dụng được trình bày như các hình dưới đây:



Hình 11. Giao diện Landing page

Giao diện Landing Page như Hình 11 đóng vai trò là công thông tin tập trung, giới thiệu tổng quan về hệ thống và làm nổi bật 6 nhóm tính năng cốt lõi (Quản lý đề tài, Chấm điểm, Lịch bảo vệ,...). Thiết kế phân luồng truy cập rõ ràng cho Sinh viên và Giảng viên, giúp người dùng định hướng nhanh chóng ngay khi truy cập.

← Quay lại trang chủ

Đăng nhập

Chào mừng bạn trở lại hệ thống

Đăng nhập với vai trò

Sinh viên

Giảng viên

Đăng nhập với Google

Email*

your.email@hcmut.edu.vn

Mật khẩu*

Ghi nhớ đăng nhập

Quên mật khẩu?

Đăng nhập

Chưa có tài khoản? [Đăng ký ngay](#)

Hình 12. Giao diện đăng nhập

Trang Đăng nhập **Hình 12** cho phép người dùng lựa chọn vai trò trước khi tiến hành xác thực (Sinh viên hoặc Giảng viên). Hệ thống hỗ trợ hai phương thức xác thực: Đăng nhập bằng tài khoản nội bộ (Email và Mật khẩu) và Đăng nhập bằng tài khoản Google (qua dịch vụ OAuth). Giao diện trực quan, đảm bảo tính bảo mật và cung cấp các tùy chọn phục hồi tài khoản như "Quên mật khẩu" và "Đăng ký ngay" cho người dùng mới.

The screenshot shows a dashboard titled "Quản lý Giảng viên" (Teacher Management). On the left, there is a sidebar with several sections: "Giáo viên bộ môn" (6 chức năng), "Quản lý giảng viên" (Quản lý sinh viên, Quản lý đê tài, Quản lý hội đồng, Lịch bảo vệ, Phân tích bộ môn), "Giáo viên" (4 chức năng), and "Giáo vụ" (9 chức năng). The main content area displays a table of teacher information:

MÃ GV	HỌ TÊN	EMAIL	GIỚI TÍNH	VAI TRÔ
1660116	Giang viên 7	naoizdabzr2004@gmail.com	Nam	Vai trò 5 Vai trò 6 Vai trò 7
5314866	Giang viên 170	teacher170@university.edu.vn	Nữ	Vai trò 170
2021590	Giang viên 185	teacher185@university.edu.vn	Nữ	Vai trò 185
5010295	Giang viên 192	teacher192@university.edu.vn	Nam	Vai trò 192
1681350	Giang viên 3	lyvinhthai3210@gmail.com	Nam	Vai trò 3
7057846	Giang viên 27	teacher27@university.edu.vn	Nữ	Vai trò 27
3881850	Giang viên 46	teacher46@university.edu.vn	Nữ	Vai trò 46
1584739	Giang viên 109	teacher109@university.edu.vn	Nam	Vai trò 109
9256391	Giang viên 122	teacher122@university.edu.vn	Nam	Vai trò 122

At the bottom, there is a message "Hiển thị 1 - 9 của 9" and navigation buttons: Số dòng: 10, Trước, Sau.

Hình 13. Dashboard: Quản lý Giảng viên

Hình 13 là giao diện quản lý dành cho các vai trò như Giáo vụ hoặc Giáo viên bộ môn. Chức năng chính là tra cứu và quản lý thông tin giảng viên trong khoa (Mã GV, Họ tên, Email, Giới tính, Vai trò). Giao diện tích hợp bộ lọc theo Học kỳ và khu vực Quản lý vai trò (sidebar) để truy cập nhanh các chức năng khác, đồng thời cung cấp tùy chọn tìm kiếm theo từ khóa và phân trang để xử lý tập dữ liệu lớn.

MÃ SV	HỌ TÊN	EMAIL	SỐ ĐIỆN THOẠI	LỚP	GIỚI TÍNH
3251117	Sinh viên 951	student951@university.edu.vn	0578704799	CLASS_42	Nam
2523719	Sinh viên 967	student967@university.edu.vn	0447660504	CLASS_49	Nữ
1025952	Sinh viên 964	student964@university.edu.vn	0742431686	CLASS_41	Nữ
1587561	Sinh viên 955	student955@university.edu.vn	0798196933	CLASS_14	Nam
8106636	Sinh viên 940	student940@university.edu.vn	0898184098	CLASS_05	Nam
4847400	Sinh viên 767	student767@university.edu.vn	0942389896	CLASS_06	Nữ
5430979	Sinh viên 764	student764@university.edu.vn	0828841642	CLASS_33	Nữ
2120081	Sinh viên 795	student795@university.edu.vn	0813343510	CLASS_37	Nam
2843608	Sinh viên 826	student826@university.edu.vn	0402449074	CLASS_27	Nam
5458528	Sinh viên 804	student804@university.edu.vn	0684787606	CLASS_02	Nữ

Hiển thị 1 - 10 của 52

Số dòng: 10 Trước 1 2 3 4 5 Sau

Hình 14. Dashboard: Quản lý Sinh viên

Hình 14 là giao diện quản lý dành cho các vai trò như Giáo viên bộ môn hoặc Giáo vụ. Chức năng chính là tra cứu, quản lý, và xem thông tin chi tiết của sinh viên trong khoa.

Bộ lọc và Tìm kiếm: Người dùng có thể tìm kiếm sinh viên theo các trường thông tin quan trọng như tên, email, hoặc mã số sinh viên và sử dụng các bộ lọc theo Học kỳ (hiển thị tại tiêu đề).

Hiển thị dữ liệu: Bảng dữ liệu chính hiển thị các cột thông tin cốt lõi bao gồm Mã SV, Họ tên, Email, Số điện thoại, Lớp, và Giới tính. Tên sinh viên được làm nổi bật để liên kết tới trang chi tiết của sinh viên đó.

Điều hướng nhanh: Giao diện có thanh điều hướng (sidebar) bên trái, cho phép chuyển đổi nhanh chóng giữa các chức năng quản lý khác nhau như Quản lý giảng viên, Quản lý đề tài, Lịch bảo vệ, v.v.

Xử lý dữ liệu lớn: Tích hợp tính năng phân trang và tùy chọn số lượng dòng hiển thị (Mặc định: 10) để quản lý hiệu quả tập dữ liệu sinh viên lớn (Ví dụ: Hiển thị 1 - 10 trên tổng số 52 sinh viên).

MÃ ĐỀ TÀI	TÊN ĐỀ TÀI	MÃ KHOA	TRẠNG THÁI	TIẾN ĐỘ	THAO TÁC
TOP_0009...	Đề tài 900 - MAJ_CNTT_KHMT	MAJ_CNTT_KHMT	Đã nộp	Stage 1: 0% Stage 2: 0%	
TOP_0008...	Đề tài 892 - MAJ_CNTT_AI	MAJ_CNTT_AI	Đang thực hiện	Stage 1: 0% Stage 2: 0%	
TOP_0008...	Đề tài 857 - MAJ_CNTT_ATTT	MAJ_CNTT_ATTT	Từ chối	Stage 1: 0% Stage 2: 0%	
TOP_0008...	Đề tài 851 - MAJ_CNTT_KHMT	MAJ_CNTT_KHMT	Đang thực hiện	Stage 1: 0% Stage 2: 0%	
TOP_0009...	Đề tài 977 - MAJ_CNTT_KHMT	MAJ_CNTT_KHMT	Từ chối	Stage 1: 0% Stage 2: 0%	
TOP_0009...	Đề tài 940 - MAJ_CNTT_KTPM	MAJ_CNTT_KTPM	Đang thực hiện	Stage 1: 0% Stage 2: 0%	
TOP_0007...	Đề tài 709 - MAJ_CNTT_HTTT	MAJ_CNTT_HTTT	Đang thực hiện	Stage 1: 0% Stage 2: 0%	
TOP_0007...	Đề tài 743 - MAJ_CNTT_HTTT	MAJ_CNTT_HTTT	Đang thực hiện	Stage 1: 0% Stage 2: 0%	
TOP_0006...	Đề tài 686 - MAJ_CNTT_KTPM	MAJ_CNTT_KTPM	Đang thực hiện	Stage 1: 0% Stage 2: 0%	
TOP_0005...	Đề tài 556 - MAJ_CNTT_HTTT	MAJ_CNTT_HTTT	Đã nộp	Stage 1: 0% Stage 2: 0%	

Hiển thị 1 - 10 của 37

Số dòng: 10 Trước Sau

Hình 15. Giao diện Quản lý đề tài

Giao diện **Hình 15** dành cho Giáo viên bộ môn hoặc Giáo vụ để duyệt và quản lý toàn bộ đề tài thuộc khoa trong một học kỳ cụ thể.

Chức năng chính: Hiển thị danh sách các đề tài với các thông tin cốt lõi như Mã đề tài, Tên đề tài, Khoa quản lý, Trạng thái, và Tiến độ (Stage 1%, Stage 2%).

Quản lý trạng thái: Người dùng có thể nhanh chóng tra cứu bằng công cụ Tìm kiếm theo tên đề tài hoặc lọc theo Trạng thái của đề tài (Đã nộp, Đang thực hiện, Từ chối).

Tác vụ: Cung cấp các thao tác như Import/Export dữ liệu và các Thao tác quản lý (chỉnh sửa, xóa hoặc tải file) ngay trên mỗi dòng.

Điều hướng: Tích hợp sidebar (Giáo viên bộ môn/Giáo vụ) cho phép chuyển đổi giữa các chức năng quản lý khác như Quản lý giảng viên, sinh viên, hội đồng, v.v.

MÃ HD	TÊN HỘI ĐỒNG	KHOA	THÀNH VIÊN	SỐ ĐỀ TÀI
27f2aef...	ácsid	MAJ_CNTT_HTTI	1	2
6a82792c...	ádas	MAJ_CNTT_AI	0	3
COU_0005...	Hội đồng bao ve 59	MAJ_CNTT_AI	6	5
COU_0008...	Hội đồng bao ve 82	MAJ_CNTT_KHMT	6	7

Hình 16. Dashboard: Quản lý Hội đồng

Giao diện **Hình 16** dành cho các vai trò quản lý (Giáo vụ/Giáo viên bộ môn) để tạo mới, tra cứu và quản lý các Hội đồng bảo vệ khóa luận theo từng học kỳ.

Chức năng chính: Hiển thị danh sách Hội đồng với các thông tin cốt lõi như Mã HD, Tên Hội đồng, Khoa, Số lượng Thành viên, và Số lượng Đề tài được phân vào hội đồng đó.

Tác vụ: Cung cấp nút "Tạo Hội đồng" và công cụ Tìm kiếm theo tên Hội đồng. Người dùng có thể click vào tên Hội đồng để xem hoặc chỉnh sửa thông tin chi tiết (danh sách thành viên, lịch làm việc).

Điều hướng: Tích hợp sidebar cho phép chuyển đổi giữa các chức năng quản lý khác như Quản lý giảng viên, sinh viên, đề tài, v.v., giúp quản lý quy trình bảo vệ một cách đồng bộ.

Bộ lọc: Tích hợp Bộ lọc Học kỳ ngay trên tiêu đề để xem các Hội đồng được thiết lập trong kỳ học hiện tại.

The screenshot shows the HCMUT dashboard with the following interface elements:

- Header:** BK HCMUT Giảng viên
- Top Bar:** Học kỳ 1 năm 2025-2026 Click để đổi học kỳ
- Left Sidebar:**
 - Giáo viên bộ môn** (6 chức năng): Quản lý giảng viên, Quản lý sinh viên, Quản lý đề tài, Quản lý hội đồng, Lịch bảo vệ, Phân tích bộ môn.
 - Giáo viên** (4 chức năng):
 - Giáo vụ** (9 chức năng):
- Main Content Area:**

Lịch Bảo vệ

Xem lịch bảo vệ của tất cả hội đồng

 - Tổng buổi bảo vệ: 1
 - Hội đồng: 4
 - Đề tài: 1

Search bar: 517

Date: 11/5/2025 (1 buổi)

Topic Council 517 details:
 - Hội đồng: ásdsd
 - Thời gian: 11:48 - 11:48
 - Sinh viên: 517
 - Hội đồng bảo vệ: Giảng viên 7 (PRESIDENT)

Hình 17. Dashboard: Lịch bảo vệ

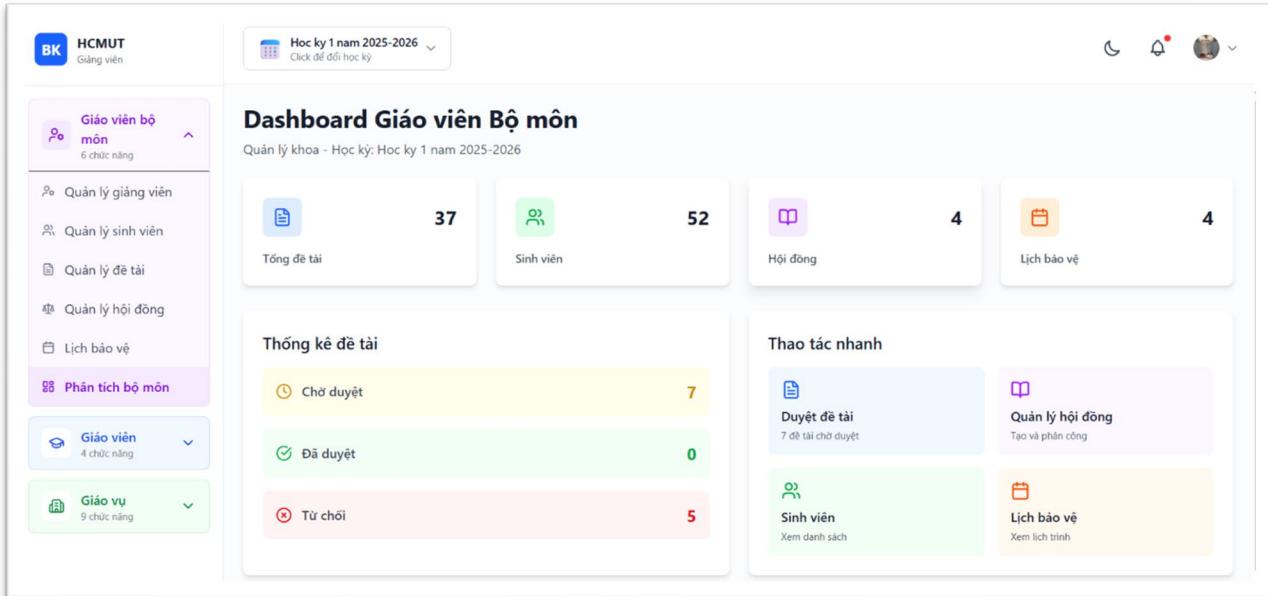
Giao diện **Hình 17** dành cho các vai trò quản lý (Giáo vụ/Giáo viên bộ môn) để theo dõi và xem chi tiết lịch bảo vệ của tất cả các hội đồng trong học kỳ.

Chức năng tổng quan: Cung cấp thông tin thống kê nhanh về Tổng buổi bảo vệ (1), Số lượng Hội đồng (4), và Số lượng Đề tài (1) được gán.

Chi tiết lịch: Hiển thị danh sách các buổi bảo vệ theo ngày (Ví dụ: 11/5/2025) và cung cấp chi tiết từng slot bảo vệ (Topic Council 517), bao gồm thời gian, số lượng sinh viên tham gia, và thông tin Giảng viên trong hội đồng (Ví dụ: Giảng viên 7 - PRESIDENT).

Tra cứu: Cho phép người dùng Tìm kiếm theo mã hội đồng hoặc sinh viên.

Điều hướng: Tích hợp bộ lọc Học kỳ và thanh sidebar để chuyển đổi giữa các chức năng quản lý khác.



Hình 18. Dashboard: Giáo viên Bộ môn

Giao diện **Hình 18** đóng vai trò là màn hình tổng quan dành cho Giáo viên Bộ môn, cung cấp các số liệu và thống kê quan trọng của học kỳ hiện tại.

Thống kê nhanh: Hiển thị các chỉ số quan trọng theo thời gian thực như Tổng đê tài (37), Tổng sinh viên (52), Tổng Hội đồng (4), và Lịch bảo vệ (4).

Phân tích: Cung cấp biểu đồ hoặc mục Thống kê đê tài chi tiết về trạng thái của các đê tài đang chờ duyệt (Chờ duyệt: 7, Đã duyệt: 0, Từ chối: 5).

Thao tác nhanh: Tích hợp các nút hành động nhanh như Duyệt đê tài, Quản lý hội đồng, và Xem danh sách sinh viên để chuyển trực tiếp đến các chức năng quản lý cốt lõi.

Điều hướng: Thanh sidebar bên trái cho phép truy cập các phân hệ quản lý chi tiết khác (Quản lý giảng viên, sinh viên, đê tài, v.v.).

BK HCMUT
Giảng viên

Học kỳ 1 năm 2025-2026

Gửi đề tài mới

Đề xuất đề tài luận văn cho sinh viên

Chọn cách tạo đề tài

Tạo đề tài mới
Tạo đề tài hoàn toàn mới cho giải đoạn 1 (DACN) hoặc giải đoạn 2 (LVTN)

Tạo từ đề tài giải đoạn 1
Chuyển đề tài đã hoàn thành giải đoạn 1 sang giải đoạn 2 (LVTN)

Tên đề tài (Tiếng Việt) *

Nhập tên đề tài bằng tiếng Việt (ít nhất 10 ký tự)

Tên đề tài (Tiếng Anh)

Nhập tên đề tài bằng tiếng Anh (hoặc click Dịch tự động)

Chương trình đào tạo

Ví dụ: Đại học chính quy, Cao học...

Loại đề tài *

Đề án chuyên ngành (DACN - Topic 1)
DACN (Topic 1) dành cho đề tài giải đoạn 1, LVTN (Topic 2) dành cho đề tài giải đoạn 2

Số lượng sinh viên tối đa

1
Số lượng sinh viên tối đa có thể đăng ký đề tài này (mặc định 1)

Sinh viên thực hiện (0/1)

Nhập MSSV sinh viên (vd: 2052001)

Tùy chọn: Thêm sinh viên vào đề tài ngay khi tạo (có thể bỏ trống và để sinh viên tự đăng ký sau)

Ngày bắt đầu *

mm/dd/yyyy

Ngày kết thúc *

mm/dd/yyyy

Mô tả chi tiết đề tài *

Bao gồm: Mô tả tổng quan, nhiệm vụ cụ thể, tài liệu tham khảo, các giải đoạn thực hiện...

Press Alt+0 for help

0 words Build with tinyMCE

Lưu ý khi gửi đề tài

- Đề tài sẽ được gửi đến bộ môn để xem xét và phê duyệt
- Tên đề tài phải rõ ràng, cụ thể và phản ánh đúng nội dung nghiên cứu
- Mô tả đề tài cần chi tiết về mục tiêu, phạm vi và phương pháp thực hiện
- Sử dụng nút "Dịch" để tự động dịch tên đề tài sang tiếng Anh
- Sau khi được phê duyệt, đề tài sẽ được hiển thị cho sinh viên đăng ký

Hình 19. Dashboard: Gửi đề tài

Giao diện Hình 19 dành cho Sinh viên hoặc Giảng viên để khởi tạo và nộp đề xuất đề tài luận văn.

Chức năng chính: Người dùng phải cung cấp đầy đủ các thông tin cốt lõi như Tên đề tài (Tiếng Việt & Tiếng Anh), Chương trình đào tạo, Loại đề tài (Đồ án chuyên ngành/Luận văn tốt nghiệp) và Số lượng sinh viên tham gia.

Thao tác: Tích hợp trình soạn thảo văn bản nâng cao để nhập Mô tả chi tiết cho đề tài.

Quy trình: Giao diện nhấn mạnh quy trình gửi đề tài: đề tài sẽ được gửi đi để chờ xem xét và phê duyệt trước khi chính thức được chuyển sang giai đoạn sinh viên đăng ký.

Điều hướng: Cung cấp tùy chọn "Tạo đề tài mới" hoặc "Tạo từ đề tài giai đoạn 1" (chuyển tiếp từ Đồ án chuyên ngành sang Luận văn tốt nghiệp).

Hình 20. Dashboard: Đề tài đang hướng dẫn

Giao diện **Hình 20** dành cho Giảng viên để quản lý và theo dõi tiến độ các đề tài mà mình đang trực tiếp hướng dẫn trong học kỳ hiện tại.

Chức năng chính: Hiển thị danh sách đề tài với các thông tin cốt lõi bao gồm Tên đề tài, Giai đoạn (ĐACN/LVTN), Sinh viên (MSSV) thực hiện, Trạng thái duyệt, Tiến độ hoàn thành (Stage 1/Stage 2), và Lịch bảo vệ dự kiến.

Tác vụ: Giảng viên có thể Tìm kiếm theo tên đề tài hoặc mã số sinh viên, và sử dụng chức năng Thao tác (như tải file) để tương tác với đề tài.

Điều hướng: Tích hợp thanh sidebar cho phép chuyển đổi giữa các chức năng giảng viên khác như Gửi đề tài mới, Hội đồng chấm điểm, và Lịch bảo vệ.

Xử lý dữ liệu: Hỗ trợ tính năng Import/Export dữ liệu và phân trang để xử lý danh sách đề tài hiệu quả.

MÃ HỘI ĐỒNG	TÊN HỘI ĐỒNG	CHỨC VỤ	SỐ SINH VIÊN ĐÃ CHẤM	THỜI GIAN BẢO VỆ
27f2aefa-fc7c-4647-b731-9a551b237da8	asdssd MAJ_CNTT_HTTT • SEM_2025_1	Chủ tịch	0 / 0	Chưa có lịch
COU_00072	Hội đồng bảo vệ 72 MAJ_DTVT_VT • SEM_2025_1	Thứ ký	1 / 1	25/11/2025 11:47

Hình 21. Dashboard: Hội đồng chấm điểm

Giao diện **Hình 21** dành riêng cho Giảng viên để quản lý và theo dõi các hội đồng mà họ được phân công tham gia chấm điểm.

Chức năng chính: Hiển thị danh sách Hội đồng mà giảng viên là thành viên, kèm theo thông tin quan trọng như Mã Hội đồng, Tên Hội đồng, Chức vụ của giảng viên (Chủ tịch, Thứ ký, v.v.), Số sinh viên đã chấm và Thời gian bảo vệ.

Tác vụ: Giảng viên có thể Tìm kiếm theo tên hội đồng và lọc theo Chức vụ của mình. Các nút Import/Export được tích hợp để xử lý dữ liệu liên quan đến chấm điểm.

Quy trình: Giảng viên sử dụng giao diện này để theo dõi tiến độ chấm điểm, đảm bảo hoàn thành vai trò của mình trong quy trình bảo vệ khóa luận.

Điều hướng: Tích hợp sidebar giúp giảng viên chuyển đổi nhanh giữa các chức năng giảng viên khác như Gửi đề tài và Đề tài hướng dẫn.

The screenshot displays the 'Lịch hội đồng bảo vệ' (Exam Committee Schedule) page. On the left, a sidebar lists three main roles: 'Giáo viên bộ môn' (selected), 'Giáo viên', and 'Giáo vụ'. Under 'Giáo viên bộ môn', there are 6 functions: 'Gửi đề tài', 'Đề tài hướng dẫn', 'Hội đồng chấm điểm', and 'Lịch bảo vệ' (selected). Under 'Giáo vụ', there are 9 functions. The main content area is titled 'Lịch hội đồng bảo vệ' and includes a sub-instruction 'Xem lịch các buổi bảo vệ mà bạn tham gia'. It shows a list of scheduled meetings for 'Thứ Ba, 25 tháng 11, 2022' (1 meeting). The first meeting is for 'Hội đồng bảo vệ 72' (COU_00072) at 11:47 on 'MAJ_DTVT_VT • SEM_2025_1'. Below this is a section for 'Chưa xác định lịch' (Not yet determined schedule) with a meeting for 'âsdasd' (ID 27f2ae7a-fc7c-4647-b731-9a551b237da8) at 'Chủ tịch'. A search bar at the top right allows users to search by committee name. At the bottom, there are buttons for 'Tim kiếm' (Search), 'Tất cả chức vụ' (All functions), and a refresh icon. The footer shows page navigation and a row of icons.

Hình 22. Dashboard: Lịch hội đồng bảo vệ

Giao diện **Hình 22** dành cho các vai trò quản lý (Giáo vụ/Giáo viên bộ môn) để tra cứu và quản lý danh sách Giảng viên trong khoa.

Chức năng chính: Hiển thị thông tin chi tiết của Giảng viên (Mã GV, Họ tên, Email, Giới tính, Vai trò) dưới dạng bảng dữ liệu.

Tìm kiếm & Lọc: Người dùng có thể nhanh chóng Tìm kiếm theo tên hoặc email của giảng viên. Tên giảng viên được làm nổi bật để liên kết tới trang chi tiết (nếu có).

Điều hướng & Phân trang: Giao diện có thanh sidebar cố định cho phép chuyển đổi giữa các chức năng quản lý khác, và tích hợp phân trang để xử lý hiệu quả danh sách giảng viên lớn.

Quản lý vai trò: Cột Vai trò cho phép người quản lý nhanh chóng xác định các chức vụ của giảng viên trong các học kỳ khác nhau.

MÃ HỌC KỲ	TÊN HỌC KỲ	NGÀY TẠO	CẬP NHẬT	THAO TÁC
hoc-ky-2-nam-2025-2026	Học kỳ 2 năm 2025-2026	21/11/2025	21/11/2025	
SEM_2023_1	Hoc ky 1 nam 2023-2024	14/10/2025	14/10/2025	
SEM_2023_2	Hoc ky 2 nam 2023-2024	14/10/2025	14/10/2025	
SEM_2024_1	Hoc ky 1 nam 2024-2025	14/10/2025	14/10/2025	
SEM_2024_2	Hoc ky 2 nam 2024-2025	14/10/2025	14/10/2025	
SEM_2025_1	Hoc ky 1 nam 2025-2026	14/10/2025	14/10/2025	

Hình 23. Dashboard: Quản lý học kỳ

Giao diện **Hình 23** dành riêng cho Giảng viên để xem lịch trình bảo vệ chi tiết của các hội đồng mà họ được phân công tham gia chấm điểm.

Chức năng chính: Hiển thị lịch bảo vệ theo dòng thời gian (Ngày, giờ) dưới hai chế độ: Sắp diễn ra và Tất cả. Mỗi mục hiển thị chi tiết về Tên Hội đồng, Mã Hội đồng, Chức vụ của giảng viên trong hội đồng (Ví dụ: Thư ký, Chủ tịch), và số lượng sinh viên tham gia/đã chấm.

Tác vụ: Giảng viên có thể Tìm kiếm theo tên hội đồng và lọc theo Chức vụ của mình để nhanh chóng xác định các buổi làm việc cần tham gia.

Quy trình: Giúp giảng viên theo dõi lịch làm việc cá nhân, đảm bảo có mặt và thực hiện nhiệm vụ chấm điểm đúng giờ.

Điều hướng: Tích hợp sidebar giúp giảng viên chuyển đổi nhanh giữa các chức năng giảng viên khác (Đề tài hướng dẫn, Hội đồng chấm điểm, Gửi đề tài).

Hình 24. Dashboard: Quản lý người dùng

Giao diện **Hình 24** dành cho vai trò Giáo vụ để tạo mới, chỉnh sửa và quản lý toàn bộ các học kỳ trong hệ thống.

Chức năng chính: Hiển thị danh sách các học kỳ hiện có dưới dạng bảng, bao gồm Mã học kỳ, Tên học kỳ, Ngày tạo, và Ngày cập nhật lần cuối.

Tác vụ quản lý: Cung cấp nút "Tạo học kỳ mới" và các thao tác chỉnh sửa (biểu tượng bút chì) hoặc xóa (biểu tượng thùng rác) cho từng bản ghi học kỳ.

Tra cứu: Hỗ trợ chức năng Tìm kiếm theo Mã hoặc Tên học kỳ để nhanh chóng tra cứu thông tin học kỳ cụ thể.

Điều hướng: Tích hợp thanh sidebar cho phép Giáo vụ chuyển đổi giữa các chức năng quản lý cấp cao khác như Quản lý người dùng, topic, hội đồng, v.v.

MÃ ĐỀ TÀI	TÊN ĐỀ TÀI	GIÁNG VIÊN HƯỚNG DẪN	TRẠNG THÁI	GIAI ĐOẠN	THAO TÁC
2e982d13...	Testing cực gắt	Lý Vĩnh Thái	Chờ duyệt	Giai đoạn 1 (DACP)	
aebfa605...	Học làm hack hệ thống của bạn Cận	Lý Vĩnh Thái	Chờ duyệt	Giai đoạn 1 (DACP)	
c6ee3e78...	Học làm hack hệ thống của Cường	Lý Vĩnh Thái	Chờ duyệt	Giai đoạn 1 (DACP)	
c344f422...	Cơ bản về Frontend nextjs	Lý Vĩnh Thái	Chờ duyệt	Giai đoạn 1 (DACP)	
396794cc...	An toàn thông tin	Lý Vĩnh Thái	Đang thực hiện	Giai đoạn 1 (DACP)	
92965d77...	xoài cát hỏa lộc tại tiền giang	Lý Vĩnh Thái	Chờ duyệt	Giai đoạn 1 (DACP)	
01f7edcf...	Kiểm tra phần mềm	Lý Vĩnh Thái	Đang thực hiện	Giai đoạn 1 (DACP)	
TOP_0008...	De tai 839 - MAJ_KTCK_CHN	Giang viên 144	Chờ duyệt	Giai đoạn 2 (LVTN)	
TOP_0008...	De tai 840 - MAJ_CNTT_KHMT	Giang viên 191	Chờ duyệt	Giai đoạn 1 (DACP)	
TOP_0008...	De tai 841 - MAJ_DTVT_KT	Giang viên 123	Đang thực hiện	Giai đoạn 1 (DACP)	

Hiển thị 1 - 10 của 1007 đề tài

Số dòng: 10 | Trước | 1 | 2 | 3 | 4 | 5 | Sau

Hình 25. Dashboard: Quản lý đề tài

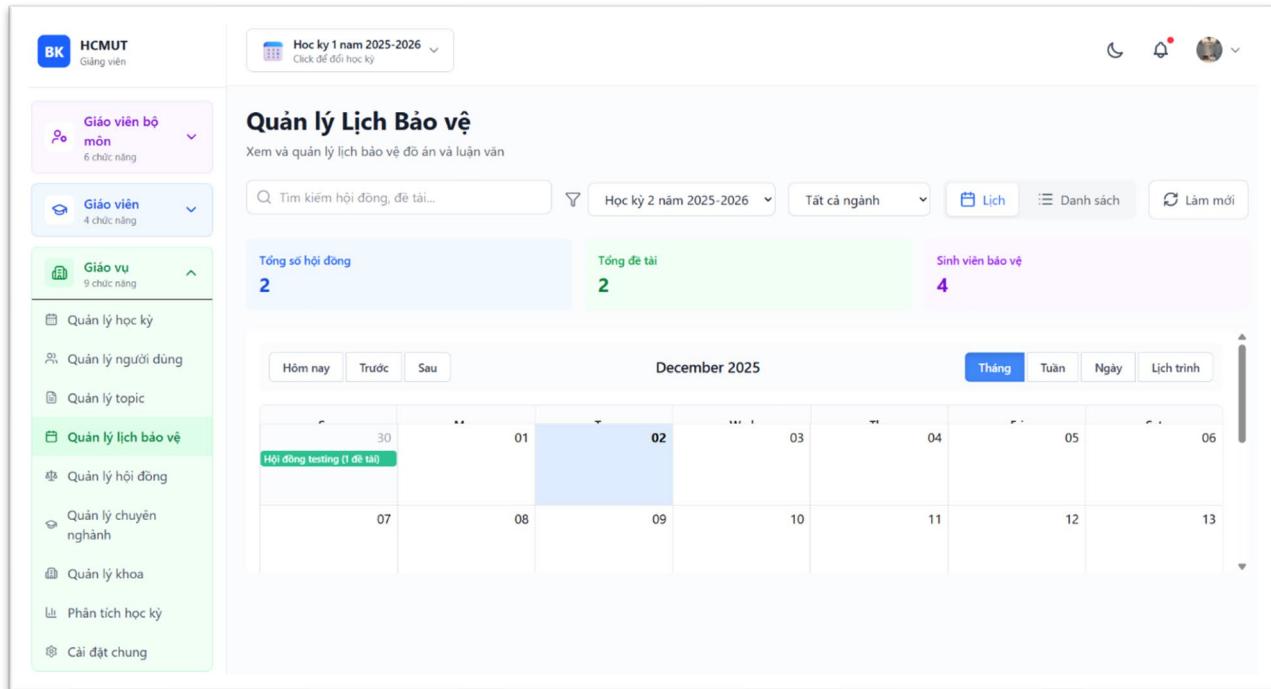
Giao diện **Hình 25** dành cho vai trò Giáo vụ để quản lý và phê duyệt toàn bộ các đề tài luận văn trong hệ thống, không giới hạn theo khoa.

Chức năng chính: Hiển thị danh sách đề tài chi tiết, bao gồm Mã đề tài, Tên đề tài, Giảng viên hướng dẫn, Trạng thái (Chờ duyệt/Đang thực hiện), và Giai đoạn (DACP/LVTN).

Quản lý trạng thái: Cung cấp công cụ Tìm kiếm theo tên đề tài và các bộ lọc theo Học kỳ và Trạng thái để quản lý quy trình phê duyệt đề tài trên toàn trường.

Tác vụ: Người dùng có thể thực hiện các Thao tác quản lý như chỉnh sửa, xóa, hoặc chuyển trạng thái đề tài (đề duyệt/từ chối) ngay trên bảng dữ liệu.

Điều hướng: Tích hợp thanh sidebar bên trái cho phép Giáo vụ chuyển đổi giữa các chức năng quản lý cấp cao khác như Quản lý người dùng, học kỳ, hội đồng, v.v.



Hình 26. Dashboard: Quản lý lịch Bảo vệ

Giao diện **Hình 26** dành cho vai trò Giáo vụ để tổng quan, xem và quản lý lịch bảo vệ của tất cả các hội đồng dưới dạng lịch (Calendar view).

Chức năng chính: Hiển thị tổng quan các số liệu (Tổng số hội đồng: 2, Tổng đề tài: 2, Sinh viên bảo vệ: 4). Lịch lớn hiển thị các buổi bảo vệ theo ngày, giúp Giáo vụ dễ dàng theo dõi và tránh trùng lịch.

Tác vụ: Hỗ trợ chuyển đổi giữa Dạng lịch và Dạng danh sách, cùng với các bộ lọc theo Học kỳ và Ngành để thu hẹp phạm vi lịch trình cần xem.

Tra cứu: Cho phép Tìm kiếm theo tên hội đồng hoặc đề tài.

Điều hướng: Tích hợp thanh sidebar cho phép Giáo vụ chuyển đổi giữa các chức năng quản lý cấp cao khác, đảm bảo tính đồng bộ trong quản lý lịch trình.

MÃ HD	TÊN HỘI ĐỒNG	NGÀNH	THỜI GIAN	SỐ ĐỀ TÀI	THAO TÁC
d0c8480f...	Hội đồng testing	MAJ_CNTT_HTTT	22:02:00 30/11/2025	1	
b21ae632...	Hội đồng hacker VN	MAJ_CNTT_HTTT	Chưa gán giờ	1	

Hình 27. Dashboard: Quản lý Hội đồng

Giao diện **Hình 27** dành cho vai trò Giáo vụ để quản lý tập trung toàn bộ tài khoản Sinh viên và Giảng viên trong hệ thống.

Chức năng chính: Hiển thị danh sách người dùng với các thông tin chi tiết như MSSV/Họ tên, Email, SĐT, Lớp, và Khoa trực thuộc, với khả năng chuyển đổi giữa danh sách Sinh viên và Giảng viên qua tab.

Tra cứu & Lọc: Cung cấp công cụ Tìm kiếm theo tên, email hoặc MSSV, kết hợp các bộ lọc theo Học kỳ, Lớp, và Khoa để tra cứu dữ liệu chính xác.

Tác vụ: Hỗ trợ các thao tác quản lý quan trọng như Thêm mới, Import/Export danh sách người dùng, và các Thao tác chỉnh sửa/xóa từng tài khoản.

Điều hướng: Tích hợp thanh sidebar cho phép Giáo vụ chuyển đổi giữa các chức năng quản lý cấp cao khác (Học kỳ, Đề tài, Hội đồng).

MÃ CHUYÊN NGÀNH	TÊN CHUYÊN NGÀNH	KHOA	THAO TÁC
MAJ_KT_QTKD	Quan tri Kinh doanh	Khoa Kinh te	
MAJ_DTVT_VT	Kỹ thuật Viễn thông	Khoa Điện tử Viễn thông	
MAJ_KHTN_HOA	Hoa học	Khoa Khoa học Tự nhiên	
MAJ_KHTN_LY	Vật lý	Khoa Khoa học Tự nhiên	
MAJ_KHTN_TOAN	Toán học	Khoa Khoa học Tự nhiên	
MAJ_KT_KT	Kinh tế	Khoa Kinh tế	
MAJ_KTXD_GT	Kỹ thuật Giao thông	Khoa Kỹ thuật Xây dựng	
MAJ_NN_ANH	Ngôn ngữ Anh	Khoa Ngoại ngữ	
MAJ_KTXD_XD	Kỹ thuật Xây dựng	Khoa Kỹ thuật Xây dựng	
MAJ_DTVT_TDH	Kỹ thuật Tự động hóa	Khoa Điện tử Viễn thông	

Hình 28. Dashboard: Quản lý Chuyên ngành

Giao diện **Hình 28** dành cho vai trò Giáo vụ để tạo mới, chỉnh sửa và quản lý danh sách các Chuyên ngành (Bộ môn) trực thuộc các Khoa trong hệ thống.

Chức năng chính: Hiển thị danh sách Chuyên ngành dưới dạng bảng, bao gồm Mã chuyên ngành, Tên chuyên ngành, và Khoa quản lý trực tiếp.

Tác vụ quản lý: Cung cấp nút "Tạo chuyên ngành mới" và các thao tác chỉnh sửa/xóa cho từng bản ghi. Chức năng Tìm kiếm theo tên chuyên ngành và Lọc theo Khoa giúp tra cứu dữ liệu hiệu quả.

Điều hướng: Tích hợp thanh sidebar bên trái cho phép Giáo vụ chuyển đổi giữa các chức năng quản lý cấp cao khác như Quản lý người dùng, học kỳ, đẻ tài, và hội đồng.

Mục đích: Đảm bảo dữ liệu nền tảng về tổ chức được cập nhật, phục vụ cho việc gán sinh viên/giảng viên vào đúng bộ môn công tác.

Khoa	Số chuyên ngành
KHMTH	3
FAC_CNSH	2
FAC_CNTT	5
FAC_DTVT	3
FAC_KHTN	3
FAC_KT	2
FAC_KTKC	2
FAC_KTXD	2
FAC_NN	1

Hình 29. Dashboard: Quản lý Khoa

Giao diện **Hình 29** dành cho vai trò Giáo vụ để tạo mới, chỉnh sửa và quản lý danh sách các Khoa trong hệ thống. Đây là giao diện trong quá trình hiện thực, hiện chưa hoạt động chính thức trong hệ thống. Trong tương lai khi hệ thống mở rộng ra các khoa khác trong

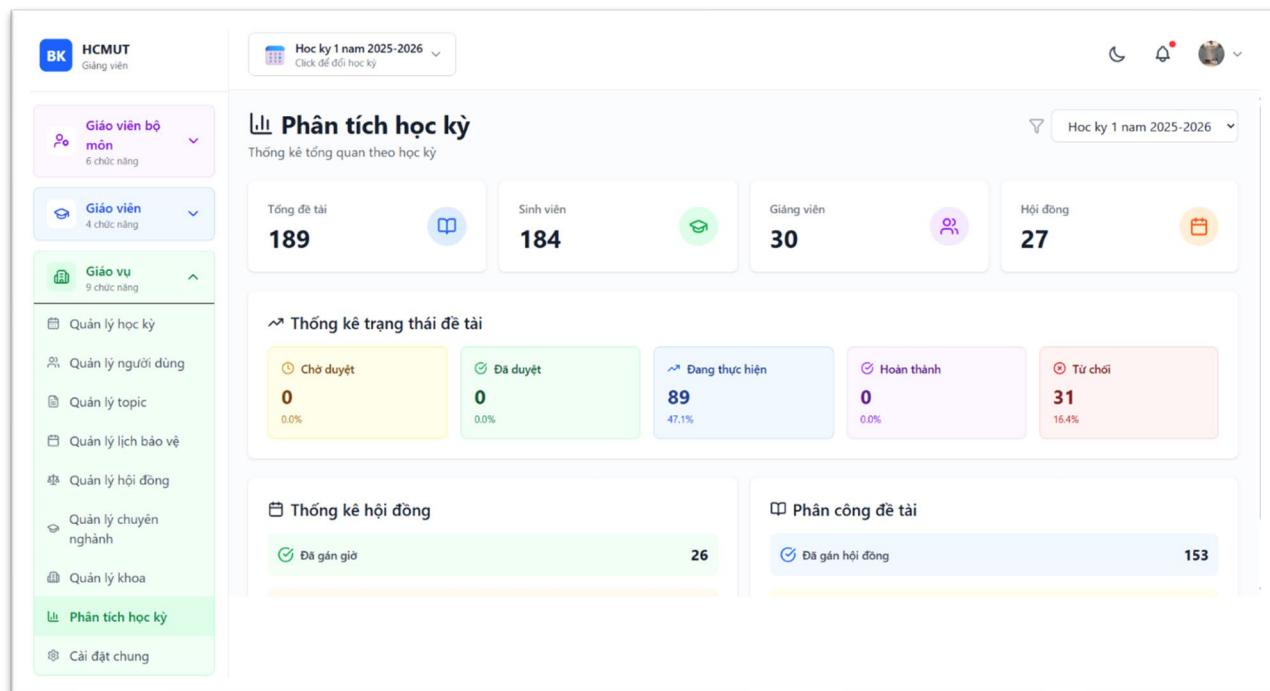
trường Đại học Bách Khoa thì đây sẽ là giao diện quản lý các khoa. Mô tả chức năng dự tính như sau:

Chức năng chính: Hiển thị danh sách các Khoa hiện có dưới dạng thẻ, bao gồm Mã Khoa, Tên Khoa, và số lượng chuyên ngành trực thuộc mỗi khoa.

Tác vụ quản lý: Cung cấp nút "Tạo Khoa mới" và các thao tác chỉnh sửa/xóa cho từng Khoa. Chức năng Tìm kiếm theo tên khoa giúp tra cứu dữ liệu hiệu quả.

Mục đích: Đảm bảo tính toàn vẹn của dữ liệu tổ chức, phục vụ cho việc phân quyền và gán các chuyên ngành.

Điều hướng: Tích hợp thanh sidebar bên trái cho phép Giáo vụ chuyển đổi giữa các chức năng quản lý cấp cao khác (Người dùng, Học kỳ, Chuyên ngành, v.v.).



Hình 30. Dashboard: Phân tích học kỳ

Giao diện **Hình 30** dành cho vai trò Giáo vụ để cung cấp thống kê tổng quan và phân tích chi tiết các hoạt động chính trong một học kỳ cụ thể.

Chức năng chính: Hiển thị các chỉ số tổng hợp quan trọng như Tổng đè tài (189), Tổng sinh viên (184), Tổng giảng viên (30), và Tổng Hội đồng (27).

Phân tích: Cung cấp biểu đồ hoặc mục thống kê chi tiết về trạng thái đè tài (Chờ duyệt, Đang thực hiện, Đã duyệt, Từ chối) và tỷ lệ gán Hội đồng/Lịch cho các đè tài, giúp người quản lý nắm bắt tình hình tức thì.

Điều hướng: Cho phép Giáo vụ chọn lọc theo Học kỳ để xem số liệu phân tích của các kỳ học khác nhau.

Mục đích: Hỗ trợ ra quyết định và đánh giá hiệu suất quản lý quy trình luận văn trong từng học kỳ.

5.3.4. Công cụ thiết kế:

Nhóm dùng công cụ Figma để hoàn thiện từng bước thiết kế: Từ sketch, đến wireframe, sau đó mock-ups và sau đó là prototype.

Link prototype trên Figma: <https://bit.ly/4qjnPCr>

Chương 6 – Hiện thực và kiểm thử

Chương này trình bày quá trình chuyển đổi từ thiết kế sang hệ thống thực tế, bao gồm môi trường triển khai và đánh giá hiệu năng hệ thống thông qua các kịch bản kiểm thử tải (Load Testing).

6.1. Môi trường triển khai

Hệ thống được đóng gói và triển khai dựa trên nền tảng Containerization để đảm bảo tính nhất quán giữa môi trường phát triển (Development) và vận hành (Production).

- **Hạ tầng Container (Docker):** Toàn bộ các thành phần của hệ thống bao gồm 6 microservices, Gateway, và các cơ sở dữ liệu đều được đóng gói thành các Docker Containers. Việc điều phối các container này được thực hiện thông qua Docker Compose (đối với môi trường Dev) hoặc Kubernetes (đối với môi trường Prod).
- **Cấu hình Server:**
 - **Ngôn ngữ & Framework:** Backend sử dụng **Golang 1.20** kết hợp với **Gin 1.8** cho các dịch vụ lõi. Frontend chạy trên **Node.js 13.0**.
 - **Cơ sở dữ liệu:** Hệ thống vận hành trên cụm cơ sở dữ liệu bao gồm **MySQL 8.0** (cho dữ liệu quan hệ), **MongoDB 7.0** (cho logs), **Redis 7.0** (cho cache/queue) và **MinIO** (lưu trữ object).
 - Các cấu hình chi tiết có trong mã nguồn của hệ thống.

6.2. Kiểm thử hiệu năng (Load Testing)

Để đảm bảo hệ thống có khả năng chịu tải cao trong các giai đoạn cao điểm (như tuần lễ nộp bài hoặc đăng ký đề tài), nhóm thực hiện đã sử dụng công cụ k6 để thực hiện kiểm thử tải (Stress Test).

Kết quả đo đạc thực tế được trình bày như **Hình 31** và **Hình 32** bên dưới đây:

```

[ THRESHOLDS
  http_req_duration
    ✓ 'p(95)<5000' p(95)=250.01ms

  http_req_failed
    ✓ 'rate<0.2' rate=0.00%]

[ TOTAL RESULTS
  checks_total.....: 250056 751.866104/s
  checks_succeeded.: 100.00% 250056 out of 250056
  checks_failed....: 0.00% 0 out of 250056

  ✓ status is 200
  ✓ no errors in response

  CUSTOM
  admin_councils_duration.....: avg=34.195004 min=0 med=3 max=1042 p(90)=92 p(95)=214
  admin_enrollments_duration...: avg=40.704866 min=0 med=4 max=6931 p(90)=113 p(95)=242.45
  admin_errors.....: 0.00% 0 out of 125028
  admin_faculties_duration....: avg=36.381371 min=0 med=4 max=1252 p(90)=89 p(95)=223.45
  admin_grade_defences_duration.: avg=34.417075 min=0 med=3 max=958 p(90)=90 p(95)=229
  admin_majors_duration.....: avg=39.99057 min=0 med=4 max=3689 p(90)=95 p(95)=236
  admin_semesters_duration....: avg=37.13425 min=0 med=2 max=1115 p(90)=108 p(95)=263
  admin_students_duration.....: avg=43.815361 min=0 med=3 max=1032 p(90)=133.9 p(95)=300.45
  admin_teachers_duration.....: avg=44.757342 min=0 med=3 max=2029 p(90)=137.9 p(95)=297
  admin_topics_duration.....: avg=39.458465 min=0 med=5 max=1101 p(90)=108.9 p(95)=233

  HTTP
  http_req_duration.....: avg=38.8ms min=360.57µs med=3.14ms max=6.93s p(90)=105.04ms p(95)=250.01ms
    { expected_response:true }....: avg=38.8ms min=360.57µs med=3.14ms max=6.93s p(90)=105.04ms p(95)=250.01ms
  http_req_failed.....: 0.00% 0 out of 125028
  http_reqs.....: 125028 375.933052/s
]

```

Hình 31. Kết quả Load/Stress Test (1)

```

HTTP
http_req_duration.....: avg=38.8ms min=360.57µs med=3.14ms max=6.93s p(90)=105.04ms p(95)=250.01ms
  { expected_response:true }....: avg=38.8ms min=360.57µs med=3.14ms max=6.93s p(90)=105.04ms p(95)=250.01ms
http_req_failed.....: 0.00% 0 out of 125028
http_reqs.....: 125028 375.933052/s

EXECUTION
iteration_duration.....: avg=4.08s min=3.71s med=3.8s max=10.67s p(90)=5.12s p(95)=5.68s
iterations.....: 13892 41.770339/s
vus.....: 6 min=2 max=300
vus_max.....: 300 min=300 max=300

NETWORK
data_received.....: 1.2 GB 3.6 MB/s
data_sent.....: 129 MB 387 kB/s

running (5m32.6s), 000/300 VUs, 13892 complete and 0 interrupted iterations
default ✓ [=====] 000/300 VUs 5m30s

✓ Stress Test completed

```

Hình 32. Kết quả Load/Stress Test (2)

Dựa trên kết quả từ k6 Benchmark, hệ thống thể hiện khả năng xử lý ánh tượng:

- Khả năng chịu tải (Concurrency): Hệ thống duy trì hoạt động ổn định với 300 người dùng ảo (VUs) truy cập và thao tác liên tục trong khoảng thời gian 5 phút 30 giây.

- **Thông lượng (Throughput):** Tổng số yêu cầu xử lý thành công là 125,028 requests, đạt tốc độ trung bình khoảng 375 requests/giây. Tổng lượng dữ liệu đã xử lý lên tới 1.2 GB.
- **Độ trễ (Latency):**
 - Thời gian phản hồi trung bình (Avg Response Time) rất thấp, chỉ 38.8ms.
 - Độ trễ tại phân vị thứ 95 (P95) là 250.01ms, thấp hơn rất nhiều so với ngưỡng yêu cầu là 5000ms. Điều này đảm bảo 95% người dùng sẽ có trải nghiệm phản hồi gần như tức thì.
- **Độ tin cậy:** Tỷ lệ lỗi (Error Rate) là 0.00%, toàn bộ 125,028 yêu cầu đều trả về HTTP Status 200, chứng tỏ hệ thống không bị quá tải hay mất kết nối dưới áp lực cao.

6.3. Đánh giá hệ thống

Sau quá trình triển khai và kiểm thử tải, hệ thống đã được đưa vào vận hành thử nghiệm và giám sát chặt chẽ thông qua bộ công cụ Observability (Grafana, Prometheus, Loki). Các số liệu thu thập được dưới đây là cơ sở khách quan để đánh giá hiệu quả của giải pháp công nghệ đã đề xuất.

6.3.1. Đánh giá hiệu suất

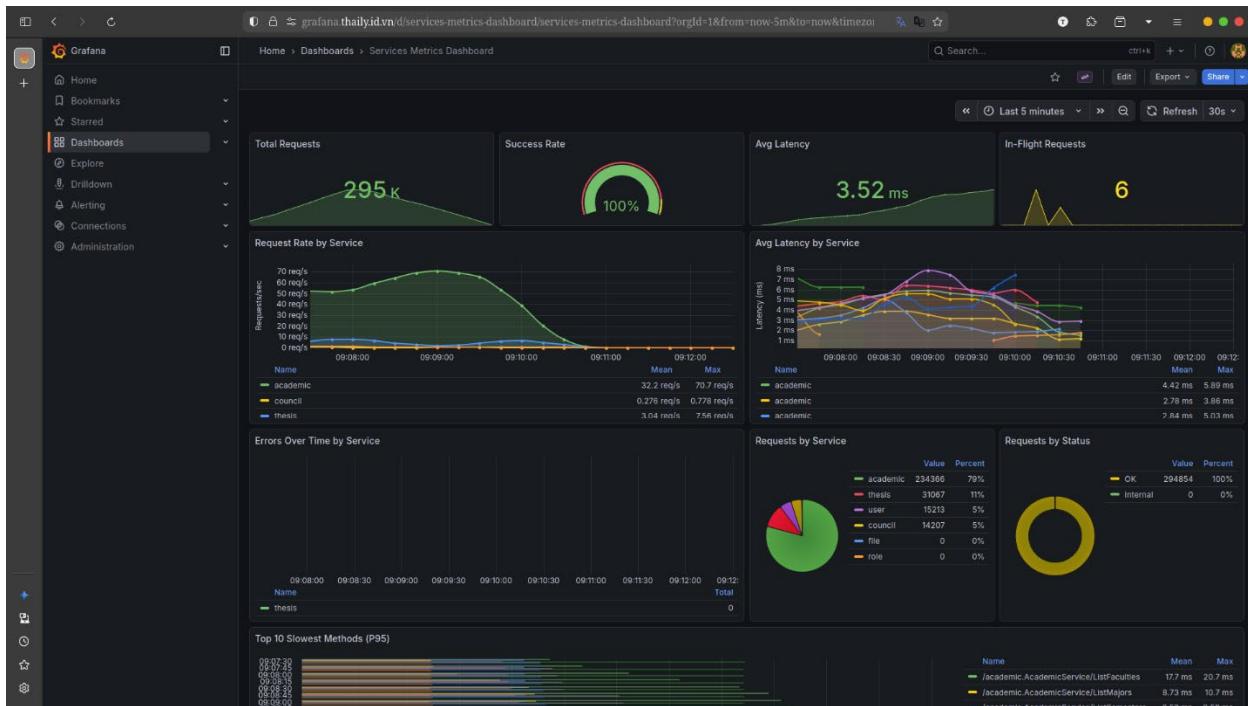
Hiệu suất của hệ thống được đánh giá dựa trên các chỉ số KPI quan trọng: Độ trễ (Latency), Lưu lượng xử lý (Request Rate) và Tỷ lệ thành công (Success Rate).

a. Lưu lượng và Tỷ lệ thành công Dữ liệu giám sát thời gian thực cho thấy hệ thống có khả năng xử lý ổn định với lưu lượng truy cập lớn.

- **Tổng quan:** Trong khung thời gian giám sát cao điểm 5 phút, hệ thống đã tiếp nhận và xử lý 295.000 yêu cầu (Total Requests).
- **Tỷ lệ thành công:** Hệ thống đạt tỷ lệ phản hồi thành công tuyệt đối là 100% (Success Rate), không ghi nhận bất kỳ yêu cầu nào bị rớt hoặc trả về lỗi 5xx trong giai đoạn này.

- Phân bổ tải: Dịch vụ academic chịu tải lớn nhất, chiếm 79% tổng lưu lượng, cho thấy nhu cầu tra cứu thông tin học vụ là tác vụ phổ biến nhất.

Minh họa cho các phân tích trên được hiển thị trong **Hình 33**:

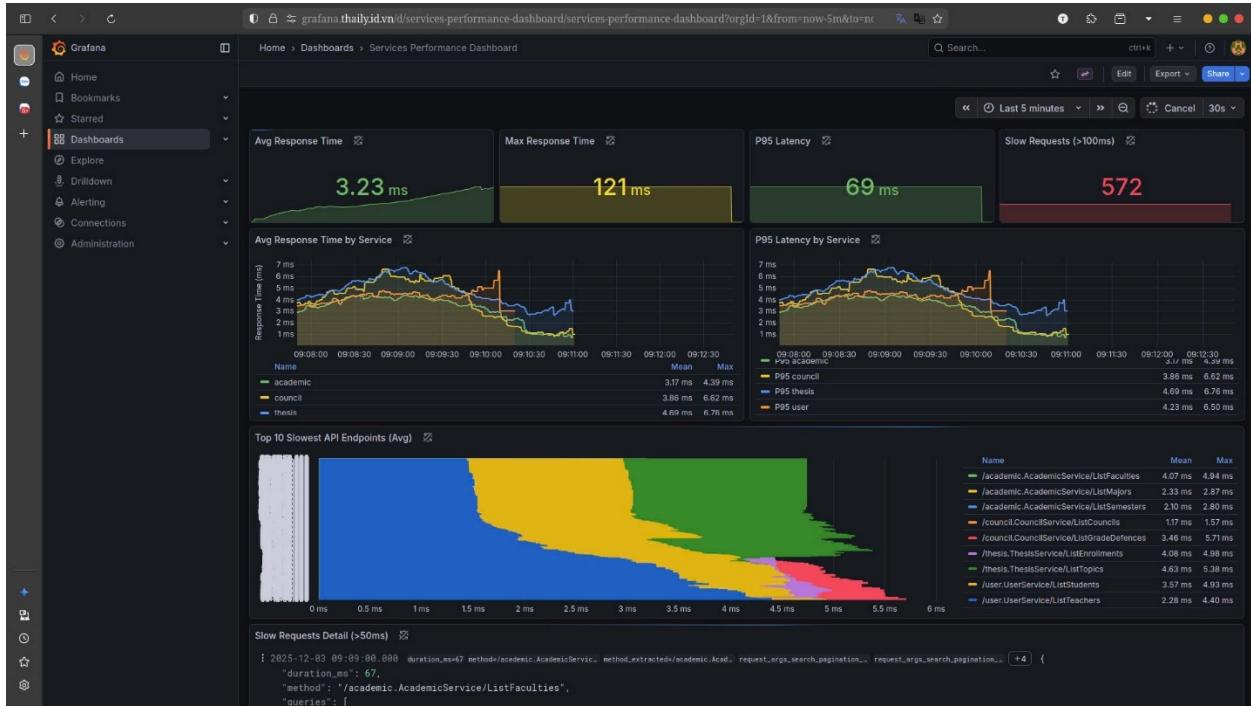


Hình 33. Grafana Dashboard: Services Metrics

b. Thời gian phản hồi (Latency) Độ trễ của hệ thống được kiểm soát ở mức rất thấp, đảm bảo trải nghiệm người dùng mượt mà ngay cả khi chịu tải.

- Thời gian phản hồi trung bình (Avg Response Time): Dao động ở mức cực thấp, từ 2.83ms đến 3.52ms cho các giao dịch nội bộ giữa các microservices.
- Độ trễ phân vị thứ 95 (P95 Latency): 95% số lượng yêu cầu được xử lý xong trong thời gian dưới 121ms. Đây là chỉ số ánh tượng đối với một hệ thống phân tán, thấp hơn nhiều so với ngưỡng chấp nhận thông thường (500ms - 1s).
- Điểm nghẽn: Hệ thống giám sát cũng tự động phát hiện các API có độ trễ cao ("Slow Requests > 100ms"), cụ thể là 488 yêu cầu chậm, giúp đội ngũ kỹ thuật khoanh vùng để tối ưu hóa.

Minh họa cho các phân tích trên được hiển thị trong **Hình 34**:



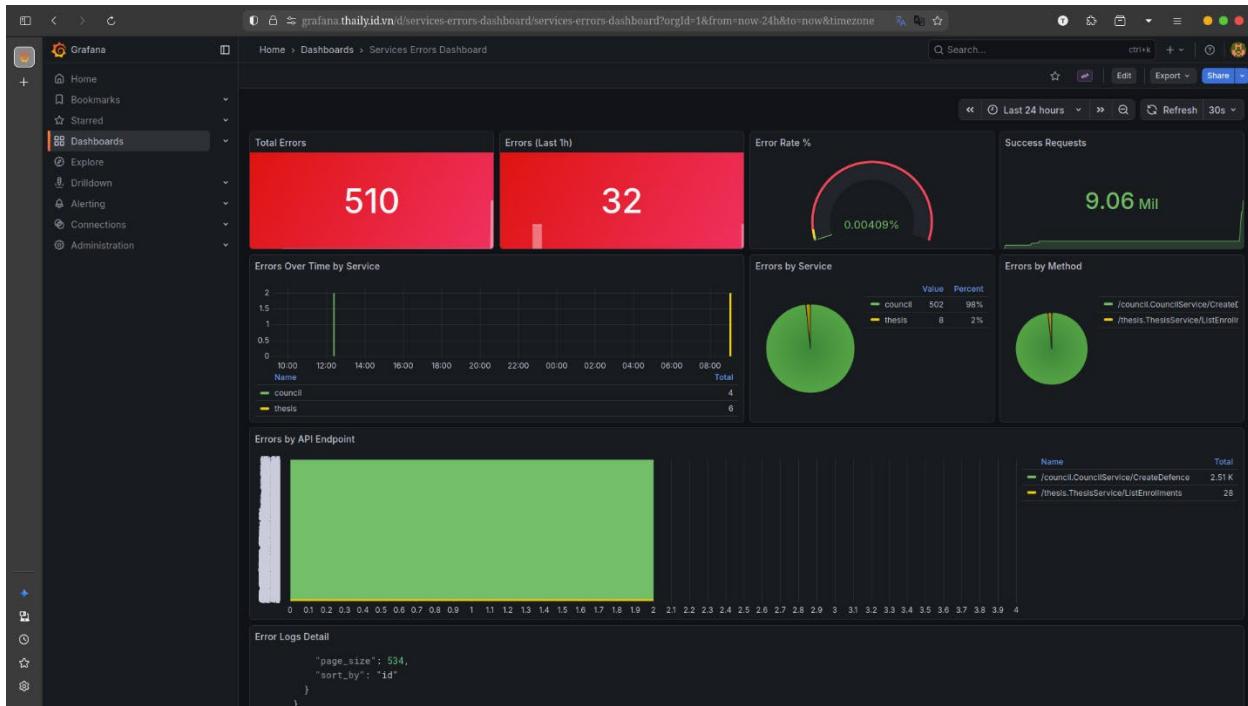
Hình 34. Dashboard Grafana: Services Performance

6.3.2. Đánh giá tính khả dụng và bảo mật

a. Tính khả dụng và Độ ổn định (Reliability) Hệ thống thể hiện độ tin cậy cao thông qua việc duy trì hoạt động liên tục với tỷ lệ lỗi không đáng kể.

- Tỷ lệ lỗi thực tế: Khi vận hành với lưu lượng tổng cộng hơn 9.06 triệu yêu cầu, hệ thống chỉ ghi nhận 510 lỗi ngoại lệ, tương đương tỷ lệ lỗi chỉ 0.00409%.
- Khoanh vùng lỗi: Công cụ giám sát logs tập trung (Loki) giúp xác định ngay nguồn gốc lỗi chủ yếu phát sinh từ dịch vụ council (chiếm 98%), cụ thể là tại các endpoint tạo hội đồng, giúp việc khắc phục sự cố diễn ra nhanh chóng.

Minh họa cho các phân tích trên được hiển thị trong **Hình 35**:



Hình 35. Grafana Dashboard: Services Errors

b. Đánh giá cơ chế bảo mật (Security)

- Cơ chế xác thực (Authentication): Việc tích hợp Google OAuth đã hoạt động hiệu quả, giúp đơn giản hóa quy trình đăng nhập cho sinh viên/giảng viên sử dụng email nhà trường, đồng thời loại bỏ rủi ro lộ lọt mật khẩu do không phải lưu trữ mật khẩu người dùng trên database hệ thống.
- Kiểm soát truy cập (Authorization): Cơ chế phân quyền RBAC (Role-Based Access Control) sử dụng JWT (JSON Web Token) được áp dụng chặt chẽ tại Gateway. Kết quả kiểm thử cho thấy người dùng không thể truy cập trái phép vào các API ngoài phạm vi vai trò của mình (Ví dụ: Sinh viên không thể gọi API chấm điểm của Giảng viên). Dữ liệu nhạy cảm như điểm số được bảo vệ toàn vẹn.

Chương 7 – Tổng kết

Sau quá trình nghiên cứu, phân tích và hiện thực hóa, đồ án "Xây dựng Hệ thống Quản lý Luận văn Tốt nghiệp" đã hoàn thành giai đoạn phát triển nền tảng (Đồ án chuyên ngành). Chương này sẽ tổng hợp các kết quả đã đạt được so với mục tiêu ban đầu, xác định các hướng phát triển công nghệ chuyên sâu và thiết lập kế hoạch làm việc chi tiết cho học kỳ tiếp theo (Luận văn tốt nghiệp).

7.1. Kết quả đạt được

Dựa trên các mục tiêu đã đề ra ở Chương 1, nhóm thực hiện đã hoàn thành các hạng mục công việc sau, đưa hệ thống vào trạng thái sẵn sàng vận hành thử nghiệm (MVP - Minimum Viable Product):

- Về mặt Kiến trúc & Công nghệ:
 - Đã xây dựng thành công kiến trúc Microservices hoàn chỉnh với 6 dịch vụ backend (Thesis, Council, User, Role, File, Academic) sử dụng ngôn ngữ Golang và giao tiếp qua gRPC.
 - Triển khai thành công chiến lược Polyglot Persistence, kết hợp MySQL cho dữ liệu quan hệ, MongoDB cho logs/workflow, và MinIO cho lưu trữ file lớn.
 - Tích hợp hệ thống giám sát thời gian thực với Grafana, Prometheus, Loki, đảm bảo khả năng quan sát (observability) toàn diện.
- Về mặt Chức năng Nghiệp vụ:
 - Quản lý quy trình khép kín: Đã số hóa toàn trình từ khâu Giáo vụ tạo học kỳ -> GVHD gửi đề tài -> GVBM duyệt -> Gán sinh viên -> Nộp bài -> Chấm điểm.
 - Cơ chế Hội đồng: Hoàn thiện chức năng tạo hội đồng phức tạp với các ràng buộc về vai trò (Chủ tịch, Thư ký, Ủy viên) và xếp lịch bảo vệ.
 - Cơ chế Chấm điểm: Hỗ trợ quy trình chấm điểm đa tầng (Hướng dẫn, Phản biện, Hội đồng) với khả năng lưu nhập (Draft) và chốt điểm (Final).

- Về mặt Hiệu năng:
 - Hệ thống đã được kiểm thử tải (Stress Test) và chứng minh khả năng chịu tải lên tới 300 VUs (người dùng ảo) hoạt động liên tục với thông lượng 375 requests/s mà vẫn duy trì độ trễ trung bình thấp (~38.8ms).

7.2. Hướng phát triển

Để chuyển đổi từ một sản phẩm đồ án thành một hệ thống thực tế có khả năng ứng dụng rộng rãi và thông minh hơn, nhóm xác định hai hướng phát triển trọng tâm cho giai đoạn 2:

7.2.1. Hoàn thiện và Nâng cao module AI kiểm tra đạo văn (Plagiarism Detection)

- Hiện trạng: Trong giai đoạn 1, tính năng kiểm tra trùng lặp mới chỉ dừng lại ở mức cơ bản, dựa trên cơ chế tìm kiếm văn bản (Full-text search) của Elasticsearch để so sánh tiêu đề và mô tả đề tài.
- Hướng phát triển: Tích hợp các mô hình Học sâu (Deep Learning) và Xử lý ngôn ngữ tự nhiên (NLP) như BERT hoặc các thư viện Hugging Face Transformers [14]. Mục tiêu là không chỉ so sánh từ khóa (keyword matching) mà còn phân tích ngữ nghĩa (semantic analysis) để phát hiện các trường hợp đạo văn tinh vi như viết lại câu (paraphrasing) hoặc dịch thuật từ tài liệu nước ngoài.

7.2.2. Mở rộng quy mô hệ thống (Multi-tenancy)

- Hiện trạng: Hệ thống hiện tại đang được cấu hình cứng (hard-coded) các quy định nghiệp vụ đặc thù cho Khoa Khoa học và Kỹ thuật Máy tính.
- Hướng phát triển: Tái cấu trúc service Academic để hỗ trợ cấu hình động cho nhiều khoa khác nhau. Cho phép mỗi khoa tự định nghĩa quy chế chấm điểm, số lượng thành viên hội đồng và quy trình duyệt riêng biệt, hướng tới mục tiêu triển khai hệ thống cho toàn trường Đại học Bách Khoa.

7.3. Kế hoạch thực hiện Đồ án tốt nghiệp (Giai đoạn 2)

7.3.1. Định nghĩa bài toán Đạo văn (Plagiarism Scope)

Trước khi xây dựng module AI, nhóm cần thống nhất định nghĩa về "Đạo văn" trong phạm vi hệ thống TMS. Số bộ có thể đưa ra các quan điểm như sau:

1. **Sao chép nguyên văn (Direct Copying):** Sao chép các đoạn văn bản từ các khóa luận cũ trong kho lưu trữ của trường mà không trích dẫn.
2. **Viết lại câu (Paraphrasing):** Thay đổi từ ngữ, cấu trúc câu nhưng giữ nguyên ý tưởng và luồng nội dung của bài gốc.

7.3.2. Cấu trúc phân rã công việc (WBS - Work Breakdown Structure)

- **WP1: Nghiên cứu & Chuẩn bị dữ liệu**
 - Thu thập dữ liệu mẫu (các báo cáo khóa luận cũ).
 - Nghiên cứu các thuật toán NLP (TF-IDF, Cosine Similarity, BERT).
- **WP2: Phát triển Module AI (Python Service hoặc tương tự)**
 - Xây dựng Service xử lý văn bản (Preprocessing).
 - Huấn luyện/Fine-tune model kiểm tra độ tương đồng.
 - Xây dựng API gRPC để tích hợp với Backend chính.
- **WP3: Tích hợp & Tối ưu hóa**
 - Tích hợp quy trình check đạo văn vào Workflow nộp bài.
 - Tối ưu hóa hiệu năng xử lý (sử dụng Queue để xử lý bất đồng bộ).
- **WP4: Mở rộng & Hoàn thiện UI/UX**
 - Nâng cấp giao diện, tăng cường trải nghiệm người dùng.
 - Cấu hình đa khoa (Multi-faculty support).
- **WP5: Kiểm thử & Viết báo cáo**
 - Kiểm thử chính xác (Accuracy Test) cho module AI.

- Viết báo cáo luận văn tốt nghiệp.

7.3.3. Bảng kế hoạch thực hiện (Gantt Chart)

Dưới đây là kế hoạch dự kiến cho 15 tuần thực hiện Đồ án tốt nghiệp (Giai đoạn 2):

Tuần (Week)	Hạng mục công việc (Task)	Chi tiết hoạt động	Kết quả bàn giao (Deliverables)
1 - 2	Khởi động & Nghiên cứu	<ul style="list-style-type: none"> - Thu thập dữ liệu báo cáo cũ làm tập huấn luyện. - Nghiên cứu, lựa chọn Model NLP phù hợp (BERT/RoBERTa). 	<ul style="list-style-type: none"> - Bộ dữ liệu thô. - Báo cáo đề xuất giải pháp AI. - Định nghĩa thế nào là “đạo văn” trên hệ thống.
3 - 5	Phát triển Core AI	<ul style="list-style-type: none"> - Xây dựng module Plagiarism Service bằng Python. - Hiện thực thuật toán trích xuất đặc trưng (Feature Extraction). 	<ul style="list-style-type: none"> - Source code module Python. - API kiểm tra văn bản đơn giản.
6 - 8	Tích hợp Backend	<ul style="list-style-type: none"> - Kết nối Golang (Thesis Service) với Python Service qua gRPC. - Cấu hình BullMQ/Redis để xử lý hàng đợi file lớn. 	<ul style="list-style-type: none"> - Luồng tích hợp hoàn chỉnh: Nộp bài -> Check AI -> Trả kết quả.
9 - 10	Phát triển Giao diện & Báo cáo	<ul style="list-style-type: none"> - Thiết kế UI hiển thị phần trăm trùng lặp và tông màu đoạn văn trùng. - Xây dựng chức năng xuất báo cáo đạo văn PDF. 	<ul style="list-style-type: none"> - Màn hình "Kết quả kiểm tra đạo văn".
11 - 12	Tối ưu hóa & Mở rộng	<ul style="list-style-type: none"> - Tối ưu tốc độ xử lý (Caching kết quả). 	<ul style="list-style-type: none"> - Hệ thống chạy ổn định với file lớn.

		<ul style="list-style-type: none"> - Cấu hình tham số động cho các Khoa khác (Multi-tenancy). 	<ul style="list-style-type: none"> - Demo cấu hình đa khoa.
13	Kiểm thử toàn diện	<ul style="list-style-type: none"> - User Acceptance Testing (UAT) với dữ liệu thực tế. - Fix bugs tồn đọng. 	<ul style="list-style-type: none"> - Bảng kết quả kiểm thử. - Hệ thống hoàn chỉnh (Release Candidate).
14 - 15	Tổng kết & Bảo vệ	<ul style="list-style-type: none"> - Viết báo cáo khóa luận tốt nghiệp. - Chuẩn bị slide và demo bảo vệ. 	<ul style="list-style-type: none"> - Báo cáo LVTN (Hard copy/Soft copy). - Slide thuyết trình.

Tài liệu tham khảo

- [1] S. Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, 2021.
- [2] A. A. A. Donovan and B. W. Kernighan, The Go Programming Language, Addison-Wesley, 2015.
- [3] gRPC Authors, "gRPC Introduction," 12 November 2024. [Online]. Available: <https://grpc.io/docs/what-is-grpc/introduction/>. [Accessed 21 12 2025].
- [4] Taskforce.sh, "BullMQ Documentation and Guide," 2024. [Online]. Available: <https://docs.bullmq.io/>. [Accessed 21 12 2025].
- [5] Vercel, "Next.js Documentation," 2024. [Online]. Available: <https://nextjs.org/docs>. [Accessed 21 12 2025].
- [6] A. Silberschatz, H. F. Korth and S. Sudarshan, Database System Concepts, McGraw-Hill, 2019.
- [7] MongoDB Inc., "MongoDB Manual," 2024. [Online]. Available: <https://www.mongodb.com/docs/manual/>. [Accessed 21 12 2025].
- [8] Redis Ltd., "Introduction to Redis," 2025. [Online]. Available: <https://redis.io/docs/latest/develop/get-started/introduction/>. [Accessed 21 12 2025].
- [9] MinIO Inc, "MinIO Object Storage Documentation for Linux," 2025. [Online]. Available: <https://min.io/docs/minio/linux/index.html>. [Accessed 21 12 2025].
- [10] Elastic, "Elasticsearch Guide," 2024. [Online]. Available: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>. [Accessed 12 21 2025].
- [11] Docker Inc., "What is a Container?," 2025. [Online]. Available: <https://www.docker.com/resources/what-container/>. [Accessed 21 12 2025].
- [12] Prometheus Authors, "Prometheus Overview," 2024. [Online]. Available: <https://prometheus.io/docs/introduction/overview/>. [Accessed 21 12 2025].
- [13] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 2003.

- [14] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

Phụ lục

Phụ lục 1: Hướng dẫn cài đặt và vận hành hệ thống

Phần này hướng dẫn các bước cần thiết để triển khai hệ thống TMS dựa trên cấu trúc các dịch vụ đã được phân tách trong hệ sinh thái GitHub của dự án.

1. Yêu cầu chuẩn bị

- Cơ sở hạ tầng:** Cài đặt Docker và Docker Desktop (hỗ trợ Docker Compose).
- Mã nguồn:** Hệ thống bao gồm nhiều repository độc lập, cần được tải về cùng một thư mục gốc để thuận tiện cho việc điều phối:
 - Backend Core** (be-lvtn): Chứa 06 microservices chính.
 - Workflow System** (be_queue): Xử lý hàng đợi BullMQ.
 - AI Module** (plagiarism): Dịch vụ kiểm tra đạo văn (Python) đang hiện thực.

2. Quy trình triển khai nhanh (Docker Compose)

1. Clone các dịch vụ cần thiết:

```
git clone https://github.com/cms-lvtn-2025/be-lvtn  
git clone https://github.com/cms-lvtn-2025/fe_new  
git clone https://github.com/cms-lvtn-2025/be_queue
```

2. **Cấu hình mạng nội bộ:** Đảm bảo các container có thể thông qua Gateway để giao tiếp gRPC và truy cập vào các tầng lưu trữ (MySQL, MongoDB, MinIO, Elasticsearch).

3. **Khởi động:** Tại thư mục chứa file docker-compose.yml tổng quát, chạy:

```
docker-compose up -d
```

Chi tiết nằm trong file hướng dẫn cài đặt (user manual) trong mã nguồn github, tại repository documents/README.md (<https://github.com/cms-lvtn-2025/documents>).

Phụ lục 2: Danh mục các kho lưu trữ mã nguồn và tài nguyên

Dưới đây là chi tiết các thành phần trong hệ sinh thái dự án TMS trên GitHub:

1. Các thành phần Backend

- **be-lvtn (Golang)**: Kho lưu trữ lõi chứa Server Gateway (GraphQL) và 06 Microservices nghiệp vụ (Thesis, Role, User, Council, File, Academic). Đây là thành phần chính xử lý logic và kết nối CSDL MySQL.
- **be_queue (TypeScript)**: Chứa dịch vụ Backend Workflow, sử dụng Express và BullMQ để quản lý các tác vụ bất đồng bộ thông qua Redis Queue.
- **plagiarism (Python)**: Module nghiên cứu và phát triển tính năng kiểm tra đạo văn sử dụng AI, phục vụ cho lộ trình phát triển giai đoạn 2.

2. Các thành phần Frontend

- **fe_new (TypeScript/Next.js)**: Giao diện chính dành cho Sinh viên và Giảng viên, tích hợp Apollo Client để truy vấn dữ liệu qua GraphQL từ Gateway.
- **fe_admin (TypeScript/Vite)**: Giao diện quản trị nội bộ (Dashboard) dành cho Giáo vụ và Quản trị viên hệ thống.

3. Tài liệu và Thiết kế

- **documents**: Lưu trữ các tài liệu đặc tả, sơ đồ thiết kế và báo cáo liên quan đến dự án.
- **Figma Prototype**: Bản mẫu tương tác chi tiết của toàn bộ màn hình hệ thống.
 - **Đường dẫn:** <https://bit.ly/4qjnPCr>

Phụ lục 3: Sơ đồ kiến trúc và Giám sát hệ thống

Hệ thống vận hành dựa trên kiến trúc Microservices phức tạp với tầng giám sát (Monitoring) và ghi log (Logging) chuyên sâu được cấu hình sẵn trong mã nguồn:

- **Monitoring**: Sử dụng **Prometheus** để thu thập metrics từ các dịch vụ Go và hiển thị trực quan trên **Grafana** với các Dashboard chuyên biệt như sau: General, Error, Performance, Matrix.
- **Logging**: Toàn bộ log từ các service được **Promtail** đẩy về **Loki** để quản lý tập trung, giúp hỗ trợ quá trình hậu kiểm và xử lý sự cố.

- **Storage:** Kết hợp đồng thời MySQL (Nghiệp vụ), MongoDB (Logs/Events), MinIO (File Storage) và Elasticsearch (Search Engine).