

Tài liệu SRS CollabSphere

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI THÀNH PHỐ HỒ CHÍ MINH VIỆN CÔNG NGHỆ THÔNG TIN VÀ ĐIỆN, ĐIỆN TỬ



ĐỀ TÀI: HỆ THỐNG HỖ TRỢ VIỆC HỌC THEO PHƯƠNG PHÁP PROJECT-BALEARNING THEO KIẾN TRÚC MICROSERVICE

GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN VĂN CHIẾN

TÊN HỌC PHẦN: XÂY DỰNG PHẦN MỀM HƯỚNG ĐỐI TƯƠNG

MÃ HỌC PHẦN: 010112201008

Tên sinh viên thực hiện : Lê Xuân Trung MSSV: 075204015217

Lê Đại Nghĩa MSSV: 075205009074

Trịnh Nguyễn Phi Hào MSSV: 075205003612

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Văn Chiến, người đã tận tình giảng dạy và trang bị cho em những kiến thức chuyên môn quý báu trong suốt học kì này, tạo điều kiện thuận lợi nhất để em hoàn thành đồ án này. Em xin gửi lời cảm ơn đặc biệt đến thầy Nguyễn Văn Chiến, người đã trực tiếp hướng dẫn, tận tình chỉ bảo và hỗ trợ em giải quyết những vướng mắc trong suốt quá trình thực hiện đề tài.

Mặc dù em đã cố gắng hết sức, nhưng do kiến thức và kinh nghiệm thực tế còn hạn chế, đồ án không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm cùng những ý kiến đóng góp quý báu từ thầy cô để bài làm của em có thể hoàn thiện hơn.

Một lần nữa, em xin chân thành cảm ơn!

GIỚI THIÊU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc ứng dụng công nghệ thông tin vào lĩnh vực giáo dục đã trở thành xu hướng tất yếu. Đặc biệt, các phương pháp giảng dạy hiện đại như Project-Based Learning (PBL – học tập dựa trên dự án) đang ngày càng được quan tâm và áp dụng rộng rãi nhằm phát huy tính chủ động, sáng tạo và khả năng làm việc nhóm của người học.

Tuy nhiên, trong quá trình triển khai phương pháp PBL, việc quản lý các thông tin liên quan như: phân công nhiệm vụ trong nhóm, theo dõi tiến độ dự án, lưu trữ tài liệu, đánh giá đóng góp cá nhân, giao tiếp giữa giảng viên và sinh viên, cũng như tổng hợp và báo cáo kết quả học tập còn gặp nhiều khó khăn. Nếu thực hiện thủ công hoặc qua nhiều công cụ rời rạc, quá trình này dễ gây thiếu đồng bộ, khó theo dõi tiến độ và chưa đảm bảo tính minh bạch, công bằng trong đánh giá.

Xuất phát từ thực tế đó, nhóm chúng em thực hiện đề tài “Project-Based Learning Support System – Hệ thống hỗ trợ việc học theo phương pháp Project-Based Learning”. Hệ thống được xây dựng với mục tiêu hỗ trợ giảng viên và sinh viên trong việc tổ chức, quản lý và thực hiện các dự án học tập một cách hiệu quả, thông qua các chức năng như: quản lý tài khoản người dùng, tạo và quản lý dự án học tập, phân công nhiệm vụ, theo dõi tiến độ, trao đổi trực tuyến, đánh giá kết quả và ghi nhận mức độ đóng góp của từng thành viên trong nhóm.

Bên cạnh đó, hệ thống còn cung cấp các công cụ hỗ trợ giảng viên trong việc giám sát quá trình học tập, đánh giá công bằng, tổng hợp kết quả và lập báo cáo học tập một cách trực quan, chính xác.

Đề tài không chỉ mang ý nghĩa về mặt kỹ thuật mà còn có giá trị thực tiễn cao, góp phần nâng cao chất lượng giảng dạy và học tập, thúc đẩy phương pháp học tập chủ động, sáng tạo và gắn liền với thực tiễn. Đây là một giải pháp thiết thực trong việc ứng dụng công nghệ thông tin vào đổi mới giáo dục và đào tạo tại Việt Nam.

Mục đích

Xây dựng hệ thống phần mềm hỗ trợ việc học theo phương pháp project-based learning, giúp nhà trường và sinh viên tối ưu hóa quy trình thực hiện đồ án. Hệ thống cho phép người dùng dễ dàng quản lý tài khoản, phân quyền dựa trên vai trò (Admin, Nhân viên, Giảng viên, Sinh viên) và tự động hóa việc khởi tạo dữ liệu học thuật từ các tập tin.

Hệ thống cho phép Giảng viên thiết lập các môn học, giáo trình, xây dựng danh sách dự án và các mốc thời gian (Milestones) một cách khoa học. Thông qua quy trình phê duyệt minh bạch từ Trưởng bộ môn, các dự án được phân bổ công bằng đến các lớp học và nhóm sinh viên thực tế.

Phần mềm tích hợp các công cụ cộng tác hiện đại bao gồm:

- Quản lý Task: Giúp sinh viên tự chủ trong việc phân công công việc và theo dõi tiến độ thời gian thực.
- Tương tác nhóm: Tích hợp kênh chat và bảng trắng kỹ thuật số (Whiteboard) để tối ưu hóa việc thảo luận và sáng tạo ý tưởng.
- Hệ thống đánh giá đa chiều: Tự động hóa quy trình chấm điểm Checkpoint của giảng viên, kết hợp với cơ chế đánh giá chéo (Peer Review) giữa các thành viên nhằm đảm bảo tính công bằng và khách quan trong kết quả học tập.

Mục tiêu cốt lõi của đề tài là xây dựng một giải pháp công nghệ toàn diện, có tính mở rộng cao (Scalability) nhờ kiến trúc Microservices và Docker. Hệ thống không chỉ hỗ trợ nhà trường giám sát tiến độ dự án, quản lý chất lượng đào tạo mà còn tạo ra môi trường làm việc chuyên nghiệp, thúc đẩy tinh thần làm việc nhóm và nâng cao hiệu quả thực hiện dự án của sinh viên trong môi trường giáo dục.

CHƯƠNG 1: KHẢO SÁT THỰC TRẠNG VÀ XÁC ĐỊNH YÊU CẦU HỆ THỐNG

1.1. Khảo sát thực trạng

Trong bối cảnh giáo dục hiện đại, phương pháp Học qua Dự án (Project-Based Learning - PBL) đã trở thành cách tiếp cận chủ đạo giúp sinh viên phát triển tư duy phản biện và kỹ năng làm việc nhóm thông qua các bài toán thực tế.

Tuy nhiên, việc tổ chức và quản lý các dự án nhóm trong môi trường PBL hiện tại đang đối mặt với những rào cản lớn:

- 1. Sự phân mảnh của công cụ làm việc:** Giảng viên và sinh viên phải dựa vào quá nhiều nền tảng độc lập để thực hiện các tác vụ khác nhau như giao tiếp, họp trực tuyến, chia sẻ tài liệu và lên ý tưởng. Việc liên tục chuyển đổi giữa các ứng dụng rời rạc này làm gián đoạn quy trình làm việc (workflow), giảm sự tập trung và ảnh hưởng tiêu cực đến hiệu suất chung của nhóm.
- 2. Hạn chế trong khả năng cộng tác thời gian thực:** Các nền tảng hiện có thường chưa được tối ưu hóa để tích hợp đa phân hệ trong cùng một không gian. Thiếu các công cụ đồng bộ mạnh mẽ như bảng trắng (whiteboard) hay vẽ biểu đồ để các thành viên có thể cùng nhau thao tác, ghi chú và lên ý tưởng tức thì.
- 3. Khó khăn trong quản lý và đánh giá minh bạch:**
 - **Thiếu cái nhìn toàn diện:** Không có một hệ thống thống nhất để theo dõi sao tiến độ dự án, lên lịch họp và tổ chức công việc.
 - **Đánh giá thiếu chính xác:** Thiếu các công cụ tích hợp để giảng viên bình luận, đánh giá cá nhân, cũng như thiếu cơ chế đánh giá ngang hàng (peer review) hiệu quả giữa các thành viên. Điều này dẫn đến việc khó ghi nhận chính xác và công bằng sự đóng góp của từng người vào kết quả chung.

Giải pháp đề xuất: Xây dựng hệ thống "**CollabSphere (COSRE)**" – nền tảng Web-app hợp nhất hỗ trợ quản lý dự án và cộng tác thời gian thực. Hệ thống tích hợp các công cụ hội nghị truyền hình, bảng trắng, chat nhóm và AI vào một không gian làm việc liền mạch, giúp hợp lý hóa quy trình giao tiếp, minh bạch hóa tiến độ và nâng cao hiệu quả đánh giá trong môi trường Học qua Dự án.

1.2. Yêu cầu hệ thống

Dựa trên hiện trạng và giải pháp đề xuất, hệ thống CollabSphere cần đáp ứng các yêu cầu chức năng và phi chức năng sau:

1.2.1. Yêu cầu chức năng (Functional Requirements)

• Quản trị Hệ thống & Đào tạo (System & Academic Administration):

- Cho phép Giáo vụ nhập (import) tập tin để tự động tạo và quản lý dữ liệu môn học, đề cương, lớp học cũng như tài khoản người dùng (Giảng viên, Sinh viên).
- Cung cấp quyền cho Admin vô hiệu hóa tài khoản và xem báo cáo hệ thống; Trưởng bộ môn phê duyệt hoặc từ chối các dự án do giảng viên đề xuất.

• Quản lý Dự án & Tiến độ (Project & Progress Management):

- Giảng viên thiết lập mục tiêu, tạo các mốc quan trọng (Milestones) và phân công dự án cho các lớp/nhóm cụ thể dựa trên đề cương môn học.
- Cung cấp công cụ để Nhóm trưởng tạo và quản lý các điểm kiểm tra (Checkpoints), phân công nhiệm vụ (Tasks) trong các chu kỳ (Sprints) và giám sát bài nộp của thành viên.

• Không gian làm việc & Cộng tác (Workspace & Collaboration):

- Tích hợp bộ công cụ cộng tác thời gian thực bao gồm: Bảng trắng (Whiteboard) để lên ý tưởng, Trình soạn thảo văn bản để viết báo cáo và vẽ biểu đồ đồng bộ.
- Hỗ trợ hệ thống giao tiếp đa phương tiện: Họp video/âm thanh chất lượng cao, chia sẻ màn hình và trò chuyện nhóm (Chat) liên tục ngay trong không gian làm việc.

• Đánh giá & Phản hồi (Assessment & Feedback):

- Hỗ trợ cơ chế đánh giá đa chiều: Giảng viên đánh giá nhóm/cá nhân qua các milestones; Sinh viên thực hiện đánh giá ngang hàng (Peer Review) để ghi nhận mức độ đóng góp thực tế.
- Hệ thống tự động theo dõi tỷ lệ đóng góp của từng thành viên trong quá trình thực hiện dự án.

• Trí tuệ nhân tạo hỗ trợ (AI Integration):

- Tích hợp AI Chatbot để hỗ trợ sinh viên lên ý tưởng, gợi ý giải pháp và hướng dẫn thực hiện dự án.

- Sử dụng AI để tự động sinh thông tin dự án (mục tiêu, thời gian biểu) và hỗ trợ giảng viên/sinh viên phân tích tiến độ để đưa ra lời khuyên phù hợp.

- **Quản lý Tài nguyên & Thông báo (Resource & Notification):**

- Cho phép lưu trữ, quản lý và chia sẻ tài nguyên (tài liệu, slides) tập trung theo từng lớp và nhóm.
- Hệ thống thông báo thời gian thực (Real-time) và qua Email cho các hoạt động quan trọng như: Nộp bài, hoàn thành milestone, lịch họp sắp tới hoặc nhận phản hồi mới.

1.3. Giới thiệu về công nghệ sử dụng

Hệ thống được xây dựng dựa trên kiến trúc Client-Server hiện đại, tận dụng sức mạnh của điện toán đám mây (Cloud Computing) và các công nghệ giao tiếp thời gian thực (Real-time Communication) tiên tiến:

1.3.1. Frontend (Giao diện người dùng - Client-side)

- **Framework: Reactjs.** Đây là thư viện JavaScript chủ đạo được sử dụng để xây dựng giao diện người dùng dưới dạng Single Page Application (SPA), mang lại trải nghiệm mượt mà và phản hồi nhanh chóng cho sinh viên và giảng viên.
- **Kiến trúc:** Component-based. Giao diện được chia nhỏ thành các thành phần tái sử dụng, giúp dễ dàng quản lý và nâng cấp các phân hệ phức tạp như Bảng trắng hay Trình soạn thảo.
- **Tương tác thời gian thực:** Tích hợp trực tiếp các client library của [Socket.IO](#) và [WebRTC](#) để xử lý các luồng dữ liệu vẽ bảng trắng và hội nghị truyền hình ngay trên trình duyệt mà không cần cài đặt thêm phần mềm.

1.3.2. Backend (Server-side & Core Services)

- **Framework chính:** [ASP.NET Core, an open-source web development framework | .NET](#)

Core Web API. Đóng vai trò là lõi xử lý logic nghiệp vụ, cung cấp các RESTful API hiệu năng cao và bảo mật để phục vụ cho Web Client.

- **Cơ sở dữ liệu (RDBMS): PostgreSQL.** Hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, dùng để lưu trữ dữ liệu có cấu trúc như thông tin người dùng, dự án, lớp học và các bài nộp.
- **Caching & In-memory Data: Upstash (Redis).** Sử dụng để lưu trữ session, caching dữ liệu truy cập thường xuyên nhằm tăng tốc độ phản hồi của hệ thống.
- **Công nghệ Giao tiếp & Thời gian thực:**
 - **WebRTC:** Xử lý lỗi cho tính năng gọi video, họp âm thanh và chia sẻ màn hình với độ trễ thấp.
 - **Signal:** Hệ thống chuyên biệt xử lý tin nhắn chat nhóm bảo mật.

- ⚡ [Socket.IO](#) : Đồng bộ hóa dữ liệu thời gian thực cho tính năng Bảng trắng (Whiteboard), đảm bảo mọi nét vẽ của thành viên đều hiển thị tức thì cho cả nhóm.
- **Trí tuệ nhân tạo (AI): AWS Bedrock.** Nền tảng AI được tích hợp để cung cấp dịch vụ Chatbot, hỗ trợ lên ý tưởng và biên dịch mã thông minh.

1.3.3. Hạ tầng & Dịch vụ đám mây (Infrastructure & Cloud Services)

Hệ thống không phụ thuộc vào hạ tầng vật lý cụ bộ mà được triển khai hoàn toàn trên nền tảng đám mây để đảm bảo tính sẵn sàng và khả năng mở rộng:

- **Hosting Server (Backend): Microsoft Azure.** Môi trường đám mây lưu trữ và vận hành các dịch vụ API của [ASP.NET Core, an open-source web development framework | .NET](#) Core.
- **Hosting Client (Frontend): AWS (Amazon Web Services).** Sử dụng để lưu trữ và phân phối mã nguồn frontend (Reactjs) đến người dùng cuối với tốc độ cao.
- **Lưu trữ Media: Cloudinary.** Dịch vụ lưu trữ đám mây chuyên dụng để quản lý tài nguyên hình ảnh, video, tài liệu và các tệp tin đính kèm trong dự án, giúp giảm tải cho server chính.

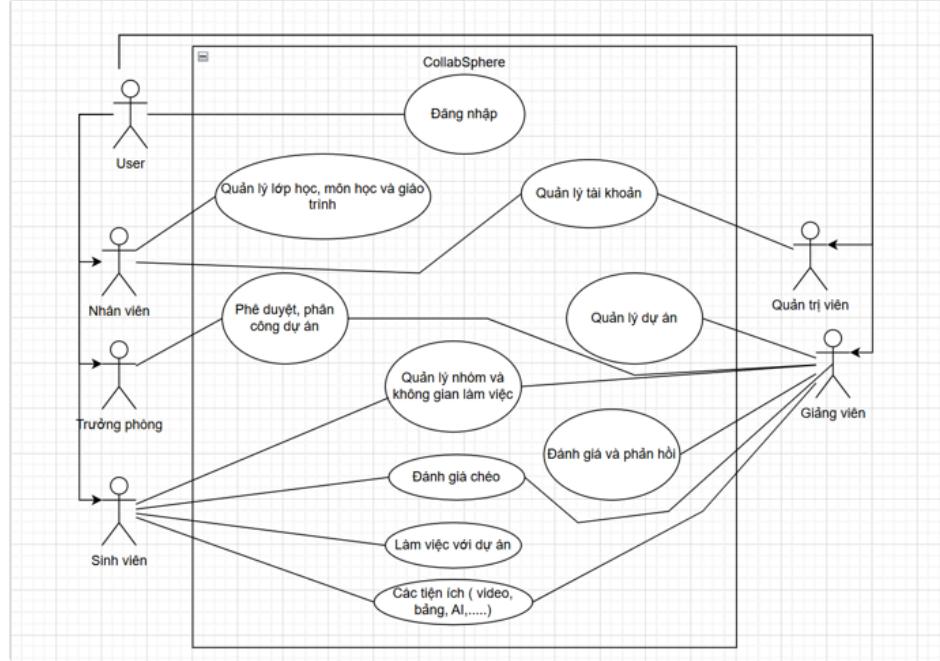
1.4. Sơ kết chương

Chương 1 đã phân tích rõ thực trạng phân mảnh và thiếu tính liên kết của các công cụ hỗ trợ phương pháp Học qua Dự án (PBL) hiện nay, từ đó đề xuất giải pháp hệ thống CollabSphere tích hợp toàn diện. Hệ thống được thiết kế dựa trên các yêu cầu nghiệp vụ chuyên sâu (từ quản lý tiến độ, cộng tác thời gian thực đến đánh giá năng lực) và vận dụng nền tảng công nghệ hiện đại ([ASP.NET Core, an open-source web development framework | .NET](#) Core Web API, Reactjs, PostgreSQL, Cloud Services) để đảm bảo tính tương tác tức thì, hiệu năng cao và trải nghiệm người dùng mượt mà. Đây là tiền đề vững chắc để đi vào thiết kế chi tiết kiến trúc và cơ sở dữ liệu ở các chương sau.

CHƯƠNG 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Biểu đồ usecase

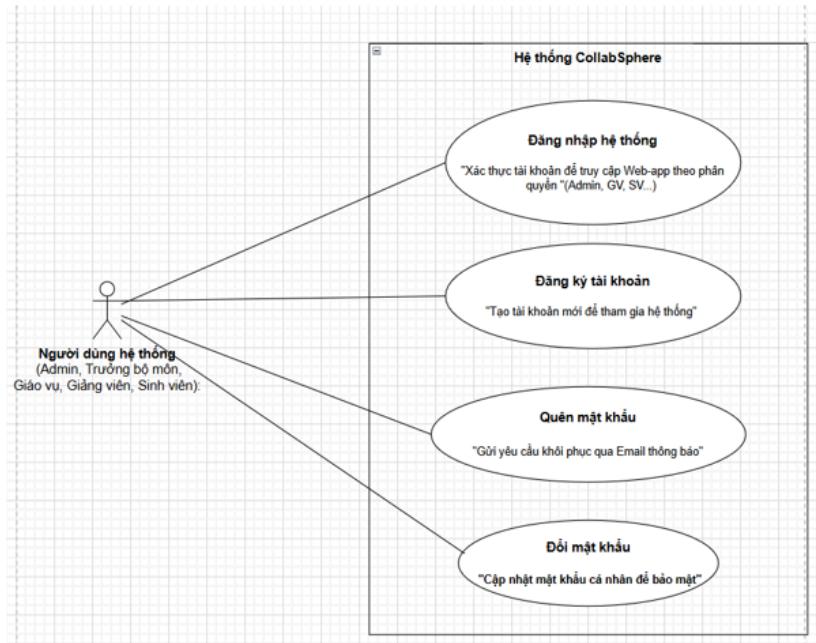
2.1.1. Biểu đồ usecase tổng quát



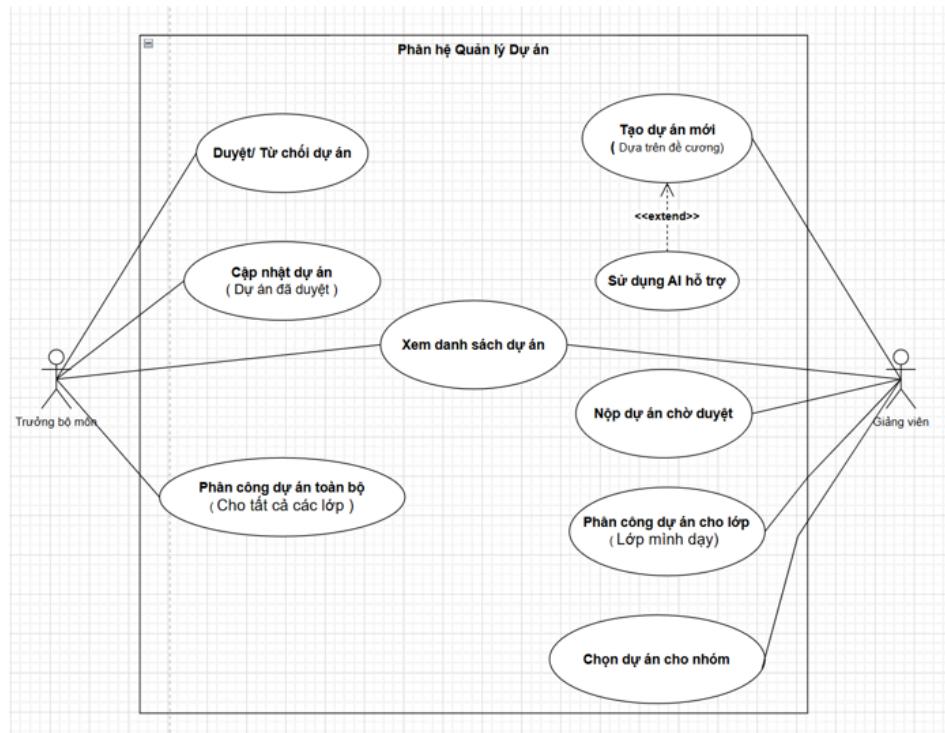
Sơ đồ tổng quát

2.1.2 Biểu đồ usecase chi tiết

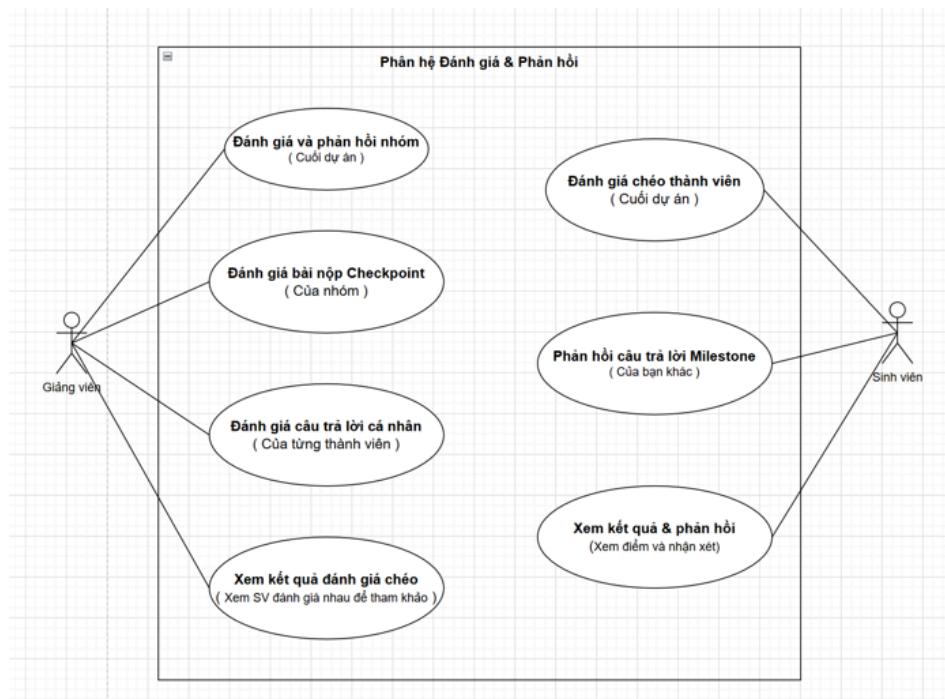
Biểu đồ đăng nhập đăng ký tài khoản



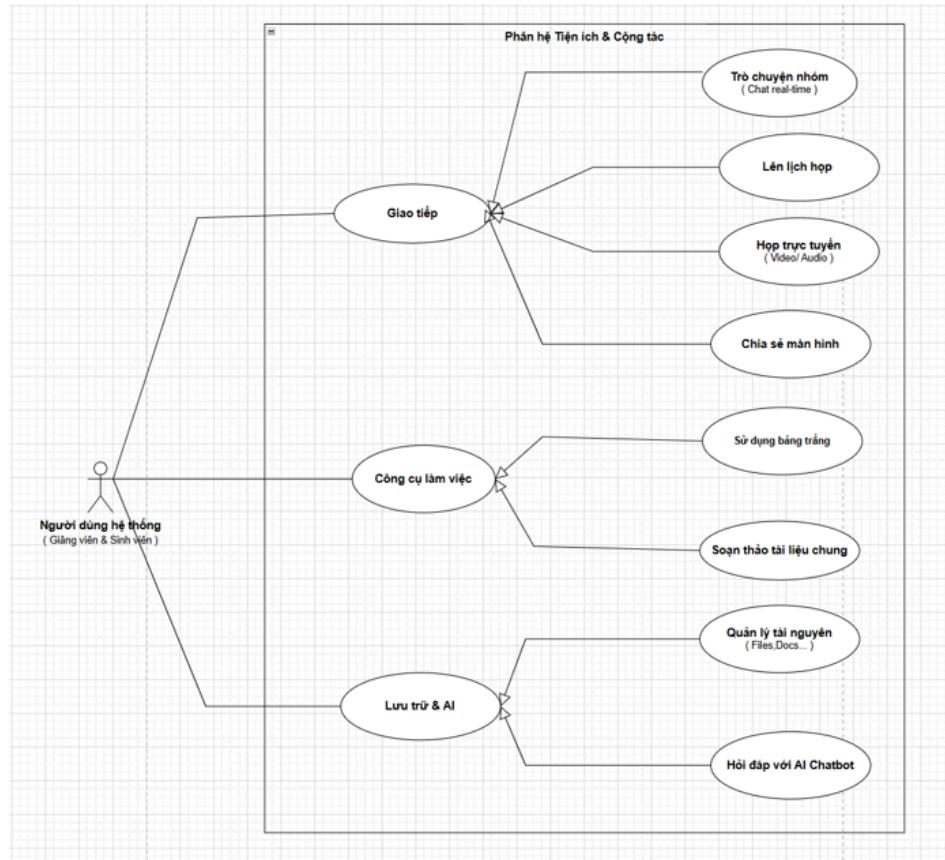
Biểu đồ quản lý dự án



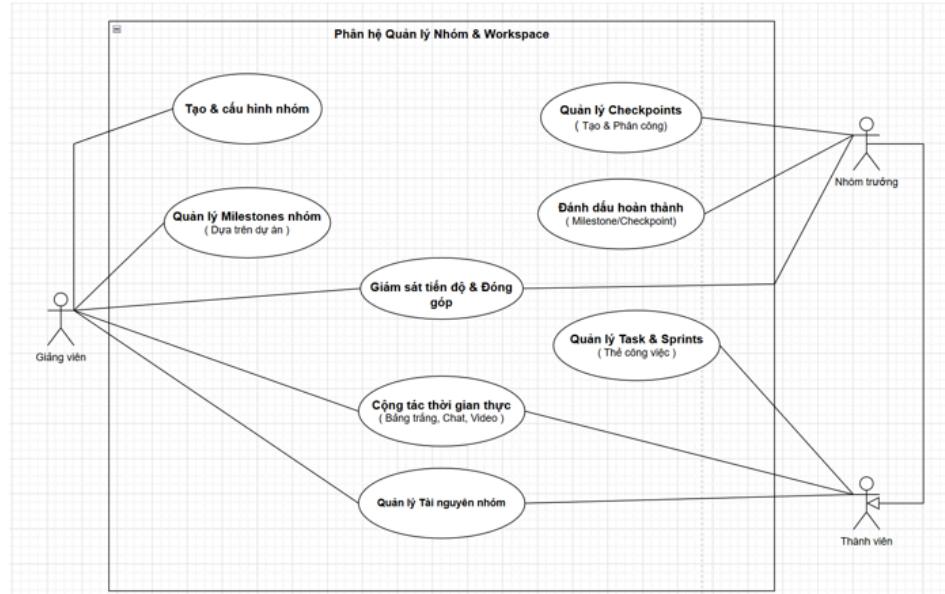
Biểu đồ đánh giá và phản hồi



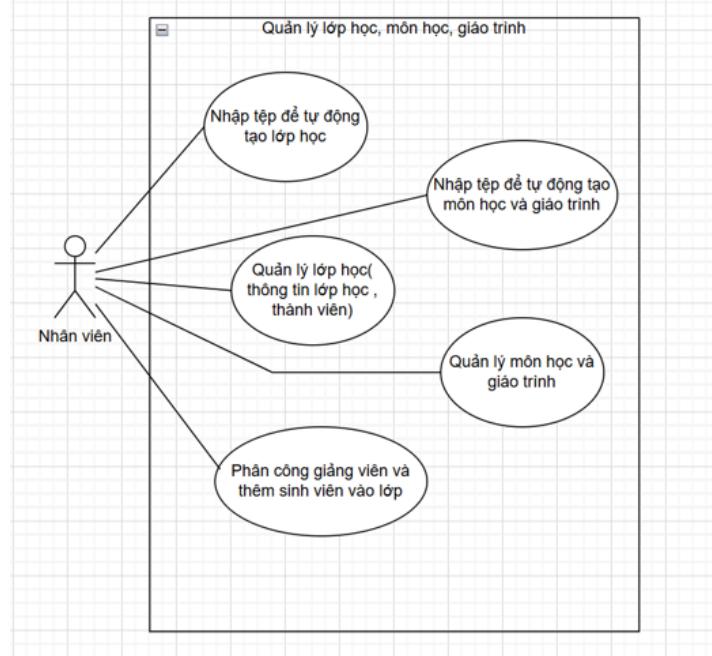
Biểu đồ tiện ích



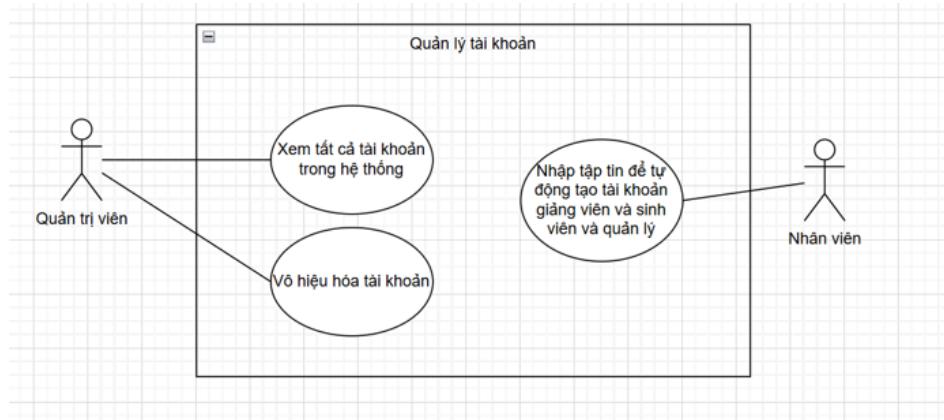
Biểu đồ Quản lý nhóm



Biểu đồ Quản lý lớp học



Biểu đồ Quản lý tài khoản



2.2 . Xây dựng kịch bản cho usecase

Bảng 2.1: Kịch bản Usecase Đăng ký tài khoản

Tên Usecase	Đăng ký tài khoản
Tác nhân	Người dùng khách (Chưa có tài khoản).
Tiền điều kiện	Người dùng chưa đăng nhập.
Hậu điều kiện	Tài khoản mới được tạo trong cơ sở dữ liệu với trạng thái chờ duyệt hoặc kích hoạt.

Kịch bản chính	<ol style="list-style-type: none"> 1. Trên màn hình Đăng nhập, người dùng chọn liên kết "Đăng ký ngay". 2. Hệ thống hiển thị biểu mẫu Đăng ký. 3. Người dùng nhập các thông tin bắt buộc: Họ tên, Email, Mật khẩu, Xác nhận mật khẩu, Mã số sinh viên (nếu là SV). 4. Người dùng nhấn nút "Đăng ký". 5. Hệ thống kiểm tra định dạng dữ liệu và kiểm tra xem Email đã tồn tại trong hệ thống chưa. 6. Hệ thống tạo tài khoản mới và lưu vào cơ sở dữ liệu. 7. Hệ thống hiển thị thông báo "Đăng ký thành công, vui lòng đăng nhập". 8. Hệ thống chuyển hướng về trang Đăng nhập.
Kịch bản ngoại lệ	<p>5a. Email đã tồn tại:</p> <ul style="list-style-type: none"> · Hệ thống báo lỗi: "Email này đã được sử dụng". <p>5b. Mật khẩu và Xác nhận mật khẩu không khớp:</p> <ul style="list-style-type: none"> · Hệ thống báo lỗi: "Mật khẩu xác nhận không trùng khớp". <p>5c. Mật khẩu không đủ mạnh (nếu có yêu cầu):</p> <ul style="list-style-type: none"> · Hệ thống báo lỗi: "Mật khẩu phải có ít nhất 8 ký tự, bao gồm chữ hoa và số".

Bảng 2.1: Kịch bản Usecase Đăng nhập

Tên Usecase	Đăng nhập
Tác nhân	Người dùng hệ thống (Admin, Trưởng bộ môn, Giáo vụ, Giảng viên, Sinh viên).
Tiền điều kiện	<ul style="list-style-type: none"> Người dùng đã có tài khoản được cấp hoặc đã đăng ký thành công trên hệ thống. Hệ thống đang hoạt động bình thường và kết nối cơ sở dữ liệu ổn định.
Hậu điều kiện	<ol style="list-style-type: none"> Nếu đăng nhập thành công: Hệ thống chuyển hướng người dùng đến trang chủ (Dashboard) tương ứng với vai trò (Role) của họ. Hệ thống lưu phiên làm việc (Session/Token).
Kịch bản chính	<ol style="list-style-type: none"> Người dùng truy cập vào trang chủ của ứng dụng web CollabSphere. Hệ thống hiển thị màn hình Đăng nhập. Người dùng nhập Email và Mật khẩu. Người dùng nhấn nút "Đăng nhập". Hệ thống kiểm tra tính hợp lệ của định dạng dữ liệu (Validation). Hệ thống xác thực thông tin tài khoản trong cơ sở dữ liệu. Hệ thống xác định vai trò (Role) của người dùng (Ví dụ: Sinh viên, Giảng viên...). Hệ thống hiển thị thông báo "Đăng nhập thành công" và

	chuyển hướng vào trang quản lý chính.
Kịch bản ngoại lệ	<p>3a. Người dùng bỏ trống Email hoặc Mật khẩu:</p> <ul style="list-style-type: none"> Hệ thống hiển thị thông báo lỗi ngay tại trường bỏ trống: "Vui lòng không để trống thông tin". Kết thúc luồng, quay lại bước 3. <p>6a. Sai Email hoặc Mật khẩu:</p> <ul style="list-style-type: none"> Hệ thống hiển thị thông báo: "Tài khoản hoặc mật khẩu không chính xác". Hệ thống xóa trường mật khẩu để người dùng nhập lại. <p>6b. Tài khoản bị khóa (Vô hiệu hóa):</p> <ul style="list-style-type: none"> Hệ thống hiển thị thông báo: "Tài khoản của bạn đã bị vô hiệu hóa. Vui lòng liên hệ Admin". Kết thúc luồng.

Bảng 2.3 Kịch bản Usecase Tạo dự án mới (Có mở rộng sử dụng AI)

Hạng mục	Nội dung
Tên Usecase	Tạo dự án mới.
Tác nhân	Giảng viên.
Tiền điều kiện	<ol style="list-style-type: none"> Giảng viên đã đăng nhập vào hệ thống thành công. Hệ thống có sẵn dữ liệu về đề cương môn học.
Hậu điều kiện	Một bản ghi dự án mới được tạo ở trạng thái "Bản nháp" (Draft) hoặc sẵn sàng để nộp.

Kịch bản chính	<ol style="list-style-type: none"> Giảng viên chọn chức năng "Quản lý Dự án" từ menu, sau đó chọn "Tạo dự án mới". Hệ thống hiển thị biểu mẫu tạo dự án (gồm các trường: Tên dự án, Mô tả, Mục tiêu, Công nghệ yêu cầu, Đề cương liên quan...). Giảng viên nhập các thông tin cần thiết vào biểu mẫu. Giảng viên nhấn nút "Lưu". Hệ thống kiểm tra tính hợp lệ của dữ liệu (không bỏ trống các trường bắt buộc). Hệ thống lưu dự án và thông báo "Tạo dự án thành công".
Kịch bản ngoại lệ	<ol style="list-style-type: none"> Tại bước 5: Nếu dữ liệu nhập thiếu hoặc sai định dạng, Hệ thống báo lỗi và yêu cầu nhập lại. Lỗi hệ thống: Nếu mất kết nối khi lưu, hệ thống báo "Lỗi kết nối, vui lòng thử lại sau".

Bảng 2.4: Kịch bản Usecase Duyệt/Từ chối dự án

Tên Usecase	Duyệt/Từ chối dự án
Tác nhân	Trưởng bộ môn
Tiền điều kiện	<ol style="list-style-type: none"> Trưởng bộ môn đã đăng nhập. Có danh sách các dự án ở trạng thái "Chờ duyệt" (Pending) do Giảng viên nộp lên.
Hậu điều kiện	Trạng thái dự án chuyển sang "Đã duyệt" (Approved) hoặc "Bị từ chối" (Rejected).

Kịch bản chính (Luồng Duyệt)	<ol style="list-style-type: none"> 1. Trưởng bộ môn chọn chức năng "Xem danh sách dự án". 2. Hệ thống hiển thị danh sách các dự án, bao gồm các dự án có trạng thái "Chờ duyệt". 3. Trưởng bộ môn chọn một dự án cụ thể để xem chi tiết. 4. Hệ thống hiển thị đầy đủ thông tin dự án. 5. Trưởng bộ môn xem xét nội dung và nhấn nút "Duyệt". 6. Hệ thống cập nhật trạng thái dự án thành "Đã duyệt" và gửi thông báo cho Giảng viên sở hữu dự án.
Kịch bản ngoại lệ	<p>Tại bước 5: Trưởng bộ môn thấy nội dung chưa đạt, nhấn nút "Từ chối".</p> <p>5a. Hệ thống hiển thị hộp thoại yêu cầu nhập "Lý do từ chối".</p> <p>5b. Trưởng bộ môn nhập lý do (ví dụ: "Mục tiêu chưa rõ ràng") và nhấn "Gửi".</p> <p>5c. Hệ thống cập nhật trạng thái dự án thành "Bị từ chối" kèm theo lý do, và gửi thông báo cho Giảng viên để chỉnh sửa</p>

Bảng 2.5: Kịch bản Phân công dự án (Dành cho Giảng viên)

Tên Usecase	Phân công dự án cho lớp (Lớp mình dạy)
Tác nhân	Giảng viên
Tiền điều kiện	<ol style="list-style-type: none"> 1. Dự án đã ở trạng thái "Đã duyệt". 2. Giảng viên đang phụ trách ít nhất một lớp học phần.

Hậu điều kiện	Dự án được gán cho một lớp cụ thể, sinh viên trong lớp có thể nhìn thấy và đăng ký.
Kịch bản chính (Thêm tỷ lệ)	<ol style="list-style-type: none"> Giảng viên tìm kiếm và chọn một dự án "Đã duyệt" trong kho dự án. Giảng viên nhấn nút "Phân công". Hệ thống hiển thị danh sách các lớp học phần mà Giảng viên này đang dạy. Giảng viên tích chọn một hoặc nhiều lớp muốn áp dụng dự án này. Giảng viên nhấn "Xác nhận". Hệ thống liên kết dự án với mã lớp học phần và thông báo thành công
Kịch bản ngoại lệ	Dự án đã phân công: Nếu dự án đã được phân công cho lớp đó rồi, hệ thống cảnh báo "Dự án này đã tồn tại trong lớp được chọn" để tránh trùng lặp.

Bảng 2.6: Kịch bản Nộp dự án chờ duyệt

Tên Usecase	Nộp dự án chờ duyệt
Tác nhân	Giảng viên
Tiền điều kiện	<ol style="list-style-type: none"> Dự án đang ở trạng thái "Bản nháp" (mới tạo) hoặc "Bị từ chối" (đã sửa xong). Dự án đã có đầy đủ thông tin bắt buộc.
Hậu điều kiện	Trạng thái dự án chuyển thành "Chờ duyệt".

Kịch bản chính (Luồng Duyệt)	<ol style="list-style-type: none"> 1. Giảng viên mở xem chi tiết dự án đang soạn thảo. 2. Giảng viên rà soát lại thông tin và nhấn nút "Nộp duyệt". 3. Hệ thống hiển thị thông báo xác nhận "Bạn có chắc chắn muốn nộp dự án này cho Trưởng bộ môn?". 4. Giảng viên chọn "Đồng ý". 5. Hệ thống cập nhật trạng thái và thông báo "Nộp thành công".
Kịch bản ngoại lệ	Dự án thiếu thông tin: Hệ thống yêu cầu bổ sung trước khi nộp.

Bảng 2.7: Kịch bản Usecase Đánh giá và phản hồi nhóm (Cuối dự án)

Tên Usecase	Đánh giá và phản hồi nhóm.
Tác nhân	Giảng viên.
Tiền điều kiện	Dự án đã kết thúc hoặc đến hạn nộp cuối cùng; Nhóm đã nộp sản phẩm.
Hậu điều kiện	Điểm số và nhận xét của nhóm được lưu vào hệ thống; Trạng thái dự án chuyển thành "Đã chấm".
Kịch bản chính	<ol style="list-style-type: none"> 1. Giảng viên chọn lớp học và chọn nhóm cần đánh giá. 2. Hệ thống hiển thị toàn bộ tài liệu/sản phẩm cuối cùng của nhóm. 3. Giảng viên nhập Điểm số tổng và Nhận xét chung cho dự án. 4. Giảng viên nhấn nút "Công bố điểm". 5. Hệ thống lưu kết quả và gửi thông báo đến toàn bộ thành viên

	nhóm.
Kịch bản ngoại lệ	<p>Bước 3 (Điểm không hợp lệ): Giảng viên nhập điểm < 0 hoặc > 10. Hệ thống báo lỗi "Điểm không hợp lệ".</p> <p>Bước 4 (Chưa nhập nhận xét): Hệ thống cảnh báo "Vui lòng nhập nhận xét trước khi công bố".</p>

Bảng 2.8: Kịch bản Usecase Đánh giá bài nộp Checkpoint

Tên Usecase	Đánh giá bài nộp Checkpoint.
Tác nhân	Giảng viên.
Tiền điều kiện	Nhóm đã nộp bài cho Checkpoint (hoặc đã quá hạn nộp).
Hậu điều kiện	Checkpoint được cập nhật trạng thái "Đã đánh giá".
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên vào giao diện quản lý tiến độ, chọn Checkpoint cần chấm. Hệ thống hiển thị file bài nộp hoặc link sản phẩm của nhóm. Giảng viên xem nội dung và nhập Đánh giá/Góp ý. Giảng viên nhấn "Gửi phản hồi". Hệ thống lưu phản hồi và thông báo cho nhóm trưởng.
Kịch bản ngoại lệ	Bước 2 (Chưa nộp bài): Hệ thống báo "Nhóm chưa nộp bài cho Checkpoint này". Giảng viên có thể chọn chấm 0 điểm hoặc nhắc nhở.

Bảng 2.9: Kịch bản Usecase Đánh giá chéo thành viên (Peer Review)

Tên Usecase	Đánh giá chéo thành viên.
Tác nhân	Sinh viên (Thành viên nhóm).
Tiền điều kiện	Dự án đang ở giai đoạn "Đánh giá cuối kỳ"; Sinh viên là thành viên hợp lệ của nhóm.
Hậu điều kiện	Dữ liệu đánh giá chéo được ghi nhận để Giảng viên tham khảo.
Kịch bản chính	<ol style="list-style-type: none"> 1. Sinh viên truy cập mục "Đánh giá thành viên". 2. Hệ thống hiển thị danh sách các thành viên còn lại trong nhóm. 3. Sinh viên chọn một thành viên, nhập Điểm đóng góp (%) và Nhận xét. 4. Lặp lại bước 3 cho tất cả thành viên khác. 5. Sinh viên nhấn "Hoàn tất đánh giá". 6. Hệ thống lưu kết quả (chế độ ẩn danh với các bạn khác).
Kịch bản ngoại lệ	Bước 5 (Tổng tỷ lệ không hợp lý): Nếu hệ thống yêu cầu tổng % đóng góp cả nhóm phải là 100% mà sinh viên nhập lệch, hệ thống báo lỗi yêu cầu điều chỉnh.

Bảng 2.10: Kịch bản Usecase Xem kết quả đánh giá chéo

Tên Usecase	Xem kết quả đánh giá chéo.
Tác nhân	Giảng viên.
Tiền điều kiện	Các thành viên trong nhóm đã hoàn thành việc đánh giá chéo.

Hậu điều kiện	Giảng viên có cơ sở để điều chỉnh điểm cá nhân.
Kịch bản chính	<ol style="list-style-type: none"> 1. Giảng viên chọn nhóm cần xem xét. 2. Giảng viên chọn tab "Đánh giá ngang hàng". 3. Hệ thống hiển thị bảng ma trận đánh giá (Ai đánh giá ai, điểm số, nhận xét). 4. Hệ thống gợi ý Điểm cá nhân dựa trên điểm nhóm và tỷ lệ đóng góp trung bình.
Kịch bản ngoại lệ	Bước 3 (Chưa đủ đánh giá): Hệ thống cảnh báo "Thành viên Nguyễn Văn A chưa thực hiện đánh giá chéo".

Bảng 2.11: Kịch bản Usecase Xem kết quả & phản hồi

Tên Usecase	Xem kết quả & phản hồi.
Tác nhân	Sinh viên.
Tiền điều kiện	Giảng viên đã công bố điểm/nhận xét.
Hậu điều kiện	Sinh viên nắm được kết quả học tập.
Kịch bản chính	<ol style="list-style-type: none"> 1. Sinh viên nhận được thông báo "Đã có kết quả dự án". 2. Sinh viên click vào thông báo hoặc vào mục "Kết quả học tập". 3. Hệ thống hiển thị chi tiết: Điểm nhóm, Điểm cá nhân, Nhận xét của giảng viên.

	4. Sinh viên xem chi tiết các lỗi sai (nếu có).
Kịch bản ngoại lệ	<i>Bước 2 (Chưa công bố):</i> Hệ thống hiển thị "Dự án đang trong quá trình chấm điểm".

Bảng 2.12: Kịch bản Usecase Họp trực tuyến (Video/Audio)

Tên Usecase	Họp trực tuyến.
Tác nhân	Người dùng hệ thống (Giảng viên, Sinh viên).
Tiền điều kiện	Người dùng đã đăng nhập và đang ở trong không gian làm việc của nhóm.
Hậu điều kiện	Kết nối thành công tới phòng họp ảo; hình ảnh/âm thanh được truyền đi.
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng nhấn nút "Tham gia cuộc họp" hoặc biểu tượng Camera. 2. Hệ thống hiển thị popup yêu cầu cấp quyền truy cập Camera/Microphone. 3. Người dùng chọn "Cho phép" (Allow). 4. Hệ thống kiểm tra thiết bị và kết nối tới máy chủ WebRTC. 5. Hệ thống hiển thị giao diện phòng họp (lưới video các thành viên). 6. Người dùng giao tiếp và thấy video của các thành viên khác.
Kịch bản ngoại lệ	<ol style="list-style-type: none"> 1. <i>Bước 3 (Từ chối quyền):</i> Người dùng chặn quyền truy cập thiết bị. Hệ thống thông báo "Không

	<p>thể truy cập Camera/Mic" và không cho vào phòng.</p> <p>2. <i>Bước 4 (Mất kết nối):</i> Đường truyền mạng không ổn định. Hệ thống báo "Đang kết nối lại..." (Reconnecting).</p>
--	--

Bảng 2.13: Kịch bản Usecase Sử dụng bảng trắng (Whiteboard)

Tên Usecase	Sử dụng bảng trắng tương tác.
Tác nhân	Giảng viên, Sinh viên.
Tiền điều kiện	Đang trong phiên làm việc nhóm.
Hậu điều kiện	Các nét vẽ/đối tượng được đồng bộ hiển thị cho tất cả thành viên.
Kịch bản chính	<ol style="list-style-type: none"> Người dùng chọn tab "Bảng trắng". Hệ thống tải trạng thái hiện tại của bảng (các hình đã vẽ trước đó). Người dùng chọn công cụ (Bút vẽ, Hình khối, Note) và thao tác trên bảng. Hệ thống ghi nhận tọa độ và gửi dữ liệu đồng bộ (socket) tức thì đến các thành viên khác. Màn hình của các thành viên khác hiển thị nét vẽ mới ngay lập tức.
Kịch bản ngoại lệ	<p><i>Bước 4 (Lỗi đồng bộ):</i> Máy chủ socket bị ngắt. Hệ thống hiển thị cảnh báo "Mất kết nối máy chủ, đang thử lại" và tạm khóa chức năng vẽ để tránh mất dữ liệu.</p>

Bảng 2.14: Kịch bản Usecase Soạn thảo tài liệu chung

Tên Usecase	Soạn thảo tài liệu chung (Real-time Doc).
Tác nhân	Giảng viên, Sinh viên.
Tiền điều kiện	Tài liệu đã được tạo trong nhóm.
Hậu điều kiện	Nội dung văn bản được lưu và cập nhật cho cả nhóm.
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng mở một file tài liệu (doc). 2. Hệ thống hiển thị trình soạn thảo và vị trí con trỏ của các thành viên đang online khác. 3. Người dùng gõ nội dung hoặc chỉnh sửa văn bản. 4. Hệ thống tự động lưu (Auto-save) và cập nhật thay đổi lên màn hình người khác.
Kịch bản ngoại lệ	<i>Xung đột phiên bản:</i> Hai người sửa cùng một dòng cùng lúc. Hệ thống ưu tiên người có timestamp đến sau hoặc hiển thị cả hai để người dùng tự gộp (merge).

Bảng 2.15: Kịch bản Usecase Hỏi đáp với AI Chatbot

Tên Usecase	Hỏi đáp với AI Chatbot.
Tác nhân	Giảng viên, Sinh viên.
Tiền điều kiện	Hệ thống AI (AWS Bedrock) đang hoạt động.
Hậu điều kiện	Người dùng nhận được câu trả lời từ AI.

Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng mở cửa sổ Chatbot AI ở góc màn hình. 2. Người dùng nhập câu hỏi (ví dụ: "Gợi ý ý tưởng cho dự án Web bán hàng"). 3. Hệ thống gửi truy vấn (Prompt) đến dịch vụ AI. 4. Hệ thống nhận phản hồi và hiển thị câu trả lời lên màn hình chat.
Kịch bản ngoại lệ	<p><i>Lỗi dịch vụ AI:</i> Hệ thống AI bị quá tải hoặc lỗi API. Hệ thống phản hồi: "Hiện tại AI đang bận, vui lòng thử lại sau".</p>

Bảng 2.16: Kịch bản Usecase Lên lịch họp

Tên Usecase	Lên lịch họp.
Tác nhân	Giảng viên, Sinh viên (Thường là Nhóm trưởng).
Tiền điều kiện	Đã đăng nhập.
Hậu điều kiện	Cuộc họp được lưu vào hệ thống và thông báo được gửi đi.
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng "Tạo lịch hẹn". 2. Hệ thống hiển thị form nhập: Tiêu đề, Thời gian bắt đầu, Thời gian kết thúc. 3. Người dùng nhập thông tin và nhấn "Tạo". 4. Hệ thống kiểm tra xem thời gian có bị trùng với lịch họp khác của nhóm không. 5. Hệ thống lưu lịch họp và gửi email/thông báo cho các thành

	viên tham gia.
Kịch bản ngoại lệ	<i>Bước 4 (Trùng lịch):</i> Hệ thống phát hiện trùng lịch. Hệ thống cảnh báo "Thời gian này đã có lịch họp khác" và yêu cầu chọn giờ khác.

Bảng 2.17: Kịch bản Usecase Tạo và cấu hình nhóm

Tên Usecase	Tạo và cấu hình nhóm.
Tác nhân	Giảng viên.
Tiền điều kiện	Giảng viên đã đăng nhập và lớp học đã có danh sách sinh viên.
Hậu điều kiện	Nhóm mới được tạo, có không gian làm việc (workspace) riêng và đã được gán thành viên.
Kịch bản chính	<ol style="list-style-type: none"> Giảng viên truy cập vào chi tiết lớp học và chọn tab "Danh sách nhóm". Giảng viên nhấn nút "Tạo nhóm mới". Hệ thống hiển thị danh sách sinh viên chưa có nhóm. Giảng viên nhập Tên nhóm, chọn Nhóm trưởng và tích chọn các thành viên khác. Giảng viên nhấn "Lưu & Tạo không gian làm việc". Hệ thống tạo nhóm, cấp quyền truy cập workspace cho các thành viên được chọn.
Kịch bản ngoại lệ	<ol style="list-style-type: none"> <i>Bước 4 (Tên trùng):</i> Tên nhóm đã tồn tại trong lớp. Hệ thống báo lỗi và yêu cầu đặt tên khác.

	2. <i>Bước 5 (Thiếu thành viên):</i> Số lượng thành viên ít hơn quy định tối thiểu (ví dụ < 3). Hệ thống cảnh báo.
--	--

Bảng 2.18: Kịch bản Usecase Quản lý Task & Sprints (Thẻ công việc)

Tên Usecase	Quản lý Task & Sprints.
Tác nhân	Nhóm trưởng, Thành viên.
Tiền điều kiện	Nhóm đã có không gian làm việc; Sprint (đợt làm việc) đã được tạo.
Hậu điều kiện	Công việc (Task) được tạo mới, cập nhật trạng thái hoặc gán người làm.
Kịch bản chính	<ol style="list-style-type: none"> 1. Nhóm trưởng/Thành viên vào giao diện "Bảng công việc" (Kanban/Scrum board). 2. Tại cột "To Do", người dùng nhấn "Thêm thẻ mới". 3. Người dùng nhập: Tiêu đề task, Mô tả, Hạn chót và Gán cho thành viên (Assignee). 4. Người dùng nhấn "Thêm". 5. Thẻ công việc xuất hiện trên bảng và hệ thống gửi thông báo cho người được gán.
Kịch bản ngoại lệ	<i>Bước 3 (Deadline sai):</i> Hạn chót của Task nằm ngoài thời gian của Sprint. Hệ thống cảnh báo "Deadline phải nằm trong thời gian Sprint".

Bảng 2.19: Kịch bản Usecase Quản lý Checkpoints (Điểm kiểm tra)

Tên Usecase	Quản lý Checkpoints.
--------------------	----------------------

Tác nhân	Nhóm trưởng.
Tiền điều kiện	Dự án đã có các Milestone (Cột mốc) do giảng viên thiết lập.
Hậu điều kiện	Checkpoint được tạo ra để các thành viên nộp bài cụ thể.
Kịch bản chính	<ol style="list-style-type: none"> 1. Nhóm trưởng chọn một Milestone đang thực hiện (ví dụ: "Thiết kế giao diện"). 2. Nhóm trưởng nhấn "Thêm Checkpoint". 3. Nhóm trưởng nhập yêu cầu: "Nộp file thiết kế Figma màn hình Login", đặt thời hạn. 4. Nhóm trưởng phân công thành viên chịu trách nhiệm chính. 5. Hệ thống lưu Checkpoint và hiển thị trên dòng thời gian của nhóm.
Kịch bản ngoại lệ	<i>Quá hạn Milestone:</i> Thời hạn của Checkpoint trễ hơn thời hạn của Milestone cha. Hệ thống yêu cầu điều chỉnh lại ngày.

Bảng 2.20: Kịch bản Usecase Giám sát tiến độ & Đóng góp

Tên Usecase	Giám sát tiến độ & Đóng góp.
Tác nhân	Giảng viên, Nhóm trưởng.
Tiền điều kiện	Các thành viên đã thực hiện các tác vụ trên hệ thống (hoàn thành task, commit code, nộp bài).
Hậu điều kiện	Người dùng nắm được tình hình thực tế của dự án qua các chỉ số.
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng truy cập vào Dashboard của nhóm.

	<p>2. Hệ thống hiển thị biểu đồ Burndown Chart (tiến độ hoàn thành công việc) và biểu đồ tròn Tỷ lệ đóng góp.</p> <p>3. Người dùng rê chuột vào tên từng thành viên để xem chi tiết số task đã làm, số giờ online.</p> <p>4. Hệ thống đưa ra cảnh báo nếu tiến độ đang chậm hơn kế hoạch (dựa trên Milestone).</p>
Kịch bản ngoại lệ	<i>Chưa có dữ liệu:</i> Dự án mới bắt đầu, chưa có hoạt động nào. Hệ thống hiển thị thông báo "Chưa đủ dữ liệu để phân tích".

Bảng 2.21: Kịch bản Đánh dấu hoàn thành (Milestone/Checkpoint)

Tên Usecase	Đánh dấu hoàn thành.
Tác nhân	Nhóm trưởng.
Tiền điều kiện	Các công việc trong Milestone/Checkpoint đã được thực hiện xong.
Hậu điều kiện	Trạng thái chuyển sang "Hoàn thành" (Completed) hoặc "Chờ duyệt" (Pending Review).
Kịch bản chính	<ol style="list-style-type: none"> Nhóm trưởng kiểm tra lại các bài nộp/task trong Milestone. Nhóm trưởng tích vào ô "Mark as Done" (Đánh dấu hoàn thành). Hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn đóng Milestone này không?". Nhóm trưởng xác nhận "Đồng ý".

	5. Hệ thống cập nhật trạng thái, khóa chỉnh sửa (tùy cấu hình) và gửi thông báo cho Giảng viên vào chấm điểm.
Kịch bản ngoại lệ	<i>Còn Task chưa xong:</i> Hệ thống phát hiện vẫn còn Task ở trạng thái "Doing". Hệ thống cảnh báo "Vẫn còn công việc chưa hoàn thành, bạn có muốn tiếp tục không?".

Bảng 2.22 Kịch bản Xem tất cả tài khoản

Tên Usecase	Xem tất cả tài khoản trong hệ thống.
Tác nhân	Quản trị viên (Admin).
Tiền điều kiện	Quản trị viên đã đăng nhập thành công vào hệ thống.
Hậu điều kiện	Danh sách tài khoản được hiển thị đầy đủ với các thông tin cơ bản.
Kịch bản chính	<ol style="list-style-type: none"> Quản trị viên chọn menu "Quản lý người dùng" trên thanh điều hướng. Hệ thống truy xuất dữ liệu và hiển thị danh sách tất cả các tài khoản hiện có (Giảng viên, Sinh viên, Giáo vụ...). Quản trị viên sử dụng bộ lọc (theo vai trò, trạng thái) hoặc thanh tìm kiếm để định vị tài khoản cụ thể. Hệ thống cập nhật danh sách hiển thị theo tiêu chí tìm kiếm.
Kịch bản ngoại lệ	<i>Danh sách rỗng:</i> Hệ thống chưa có dữ liệu người dùng nào. Hệ thống hiển thị thông báo "Không tìm thấy tài khoản nào".

Bảng 2.23 Kịch Bản Vô hiệu hóa tài khoản

Tên Usecase	Vô hiệu hóa tài khoản.
Tác nhân	Quản trị viên (Admin).
Tiền điều kiện	Quản trị viên đã xác định được tài khoản cần khóa trong danh sách.
Hậu điều kiện	Trạng thái tài khoản chuyển sang "Vô hiệu hóa" (Inactive/Locked); Người dùng sở hữu tài khoản đó không thể đăng nhập.
Kịch bản chính	<ol style="list-style-type: none"> Trên danh sách tài khoản, Quản trị viên nhấn nút "Khóa" hoặc biểu tượng Vô hiệu hóa tương ứng với tài khoản cần xử lý. Hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn vô hiệu hóa tài khoản [Email] không?". Quản trị viên nhấn "Đồng ý". Hệ thống cập nhật trạng thái tài khoản trong cơ sở dữ liệu. Hệ thống hiển thị thông báo "Đã vô hiệu hóa tài khoản thành công".
Kịch bản ngoại lệ	<i>Tài khoản Admin:</i> Quản trị viên cố gắng vô hiệu hóa chính mình hoặc một Admin khác. Hệ thống báo lỗi "Không thể vô hiệu hóa tài khoản Quản trị viên".

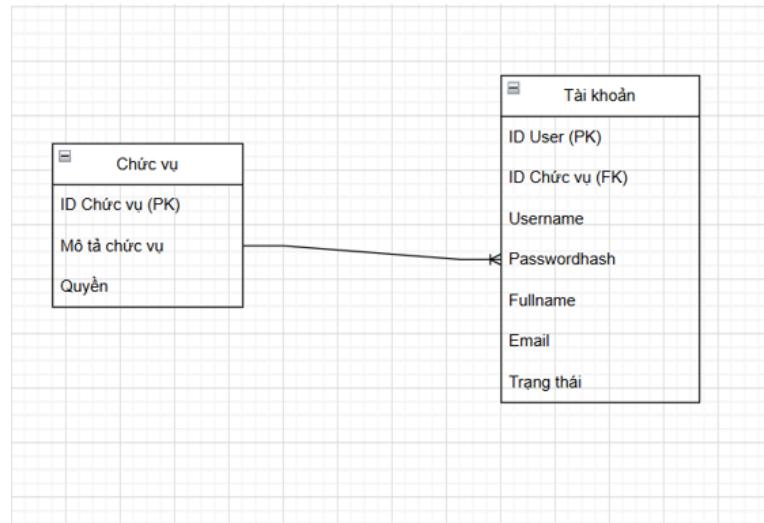
Bảng 2.24 Kịch Bản Nhập tập tin tạo tài khoản tự động

Tên Usecase	Nhập tập tin để tự động tạo tài khoản (Import).
Tác nhân	Nhân viên (Giáo vụ).
Tiền điều kiện	Nhân viên đã có file danh sách (Excel/CSV) đúng mẫu quy định; Đã đăng nhập hệ thống.
Hậu điều kiện	Các tài khoản mới được tạo hàng loạt trong hệ thống và thông tin đăng nhập được gửi đi (nếu có cấu hình).
Kịch bản chính	<ol style="list-style-type: none"> 1. Nhân viên truy cập chức năng "Import tài khoản". 2. Hệ thống hiển thị giao diện tải lên tập tin. 3. Nhân viên nhấn "Chọn tệp" và tải lên file danh sách Giảng viên/Sinh viên từ máy tính. 4. Hệ thống đọc file và hiển thị bản xem trước (Preview) dữ liệu hợp lệ. 5. Nhân viên kiểm tra sơ bộ và nhấn nút "Tiến hành nhập". 6. Hệ thống xử lý tạo tài khoản hàng loạt và hiển thị báo cáo kết quả (Số dòng thành công, số dòng lỗi).
Kịch bản ngoại lệ	<ol style="list-style-type: none"> 1. <i>Sai định dạng file:</i> Nhân viên tải lên file ảnh hoặc file Word thay vì Excel. Hệ thống báo lỗi "Định dạng không hỗ trợ". 2. <i>Dữ liệu trùng lặp:</i> File chứa Email đã tồn tại trong hệ thống. Hệ thống sẽ bỏ qua các dòng đó và

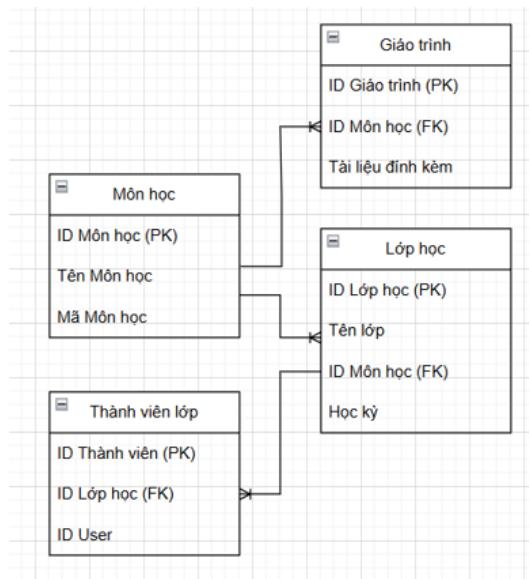
báo lỗi cụ thể cho từng dòng
trong báo cáo kết quả.

CHƯƠNG 3 : THIẾT KẾ HỆ THỐNG

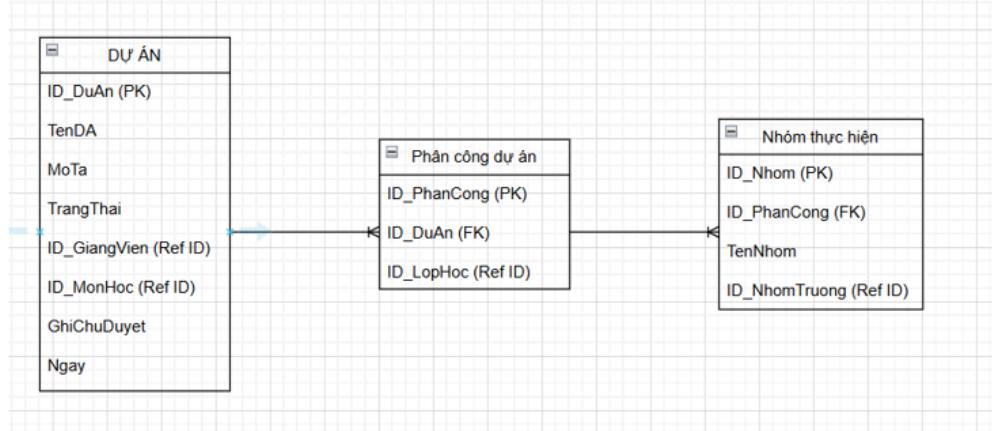
3.1. Xây dựng cơ sở dữ liệu



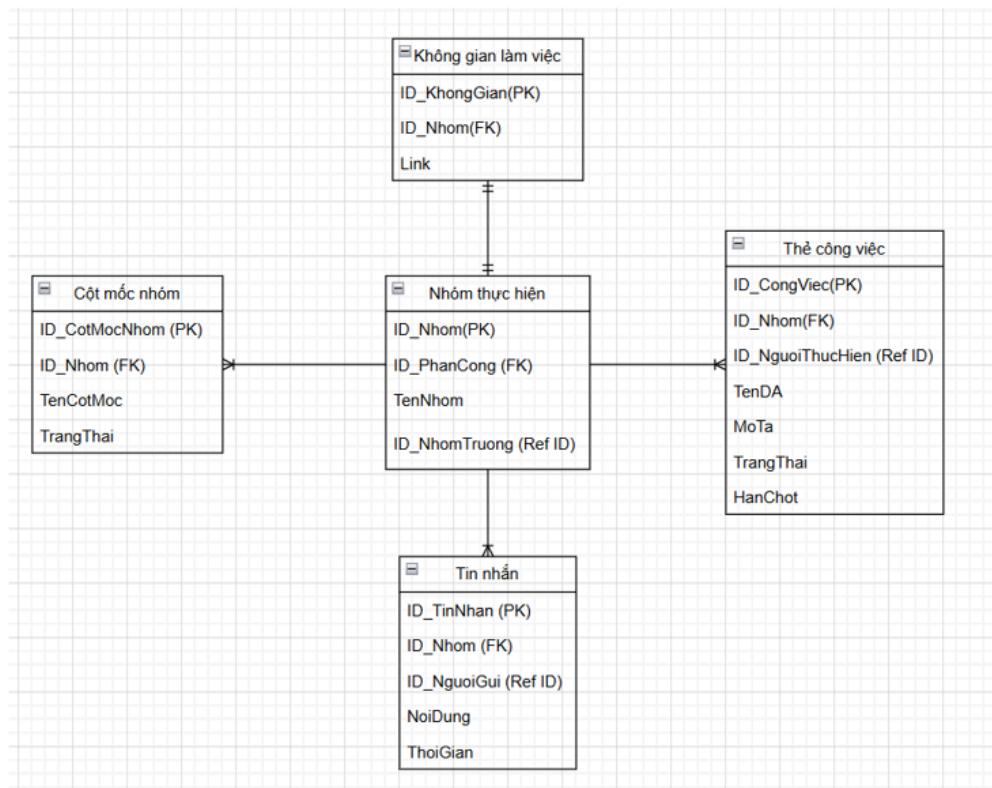
Sơ đồ ERD quản lý tài khoản



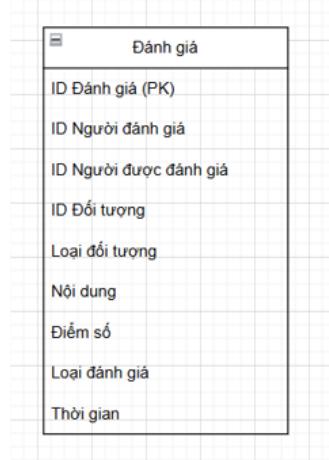
Sơ đồ ERD quản lý lớp, môn học, giáo trình



Sơ đồ ERD quản lý dự án

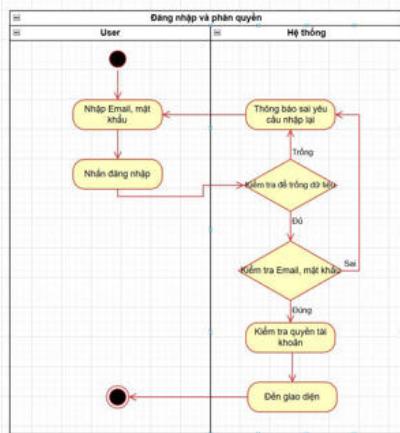


Sơ đồ ERD quản lý nhóm và không gian làm việc

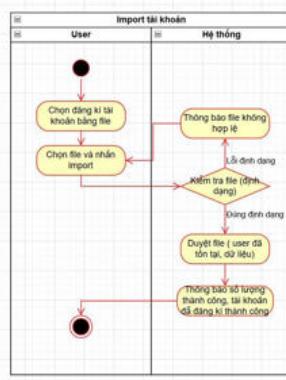


Sơ đồ ERD đánh giá

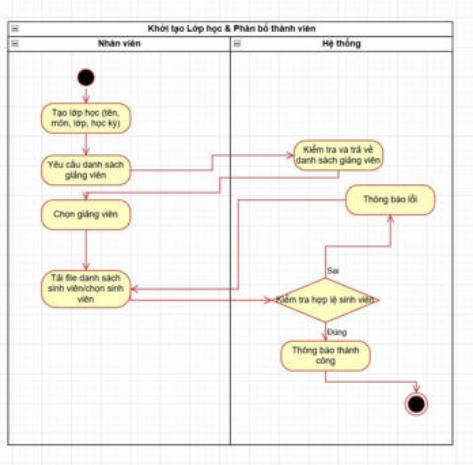
3.2 Xây dựng sơ đồ hoạt động



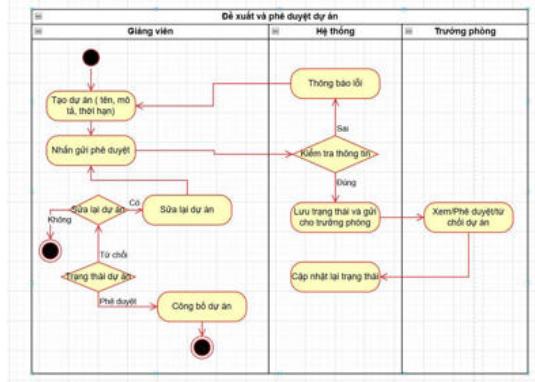
Sơ đồ hoạt động Đăng nhập & Phân quyền



Sơ đồ hoạt động Import tài khoản hàng loạt



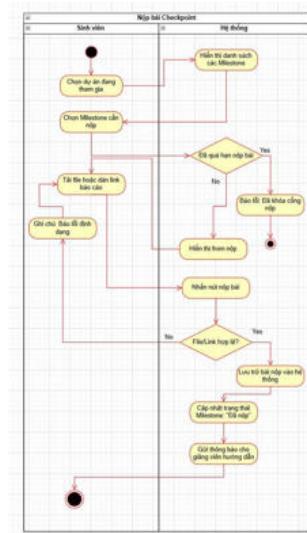
Sơ đồ hoạt động Khởi tạo Lớp học & Phân bổ thành viên



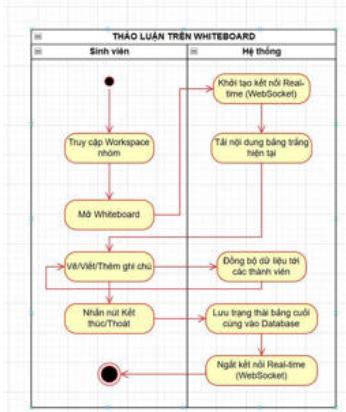
Sơ đồ hoạt động Đề xuất & Phê duyệt Dự án



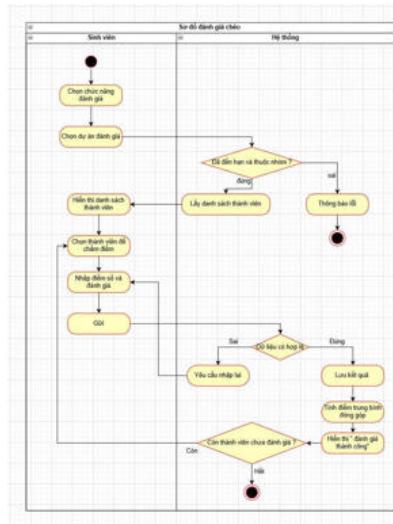
Sơ đồ hoạt động Quản lý Task
(Công việc)



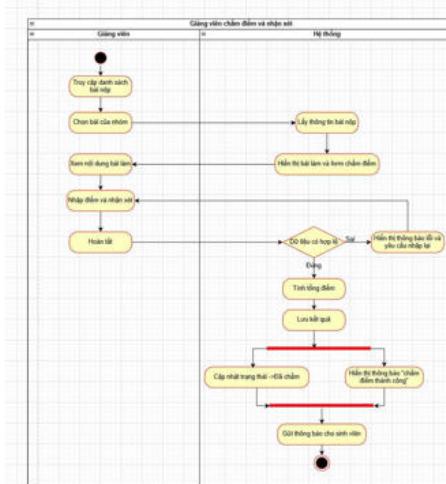
Sơ đồ hoạt động Nộp bài
Checkpoint



Sơ đồ hoạt động Thảo luận trên Whiteboard

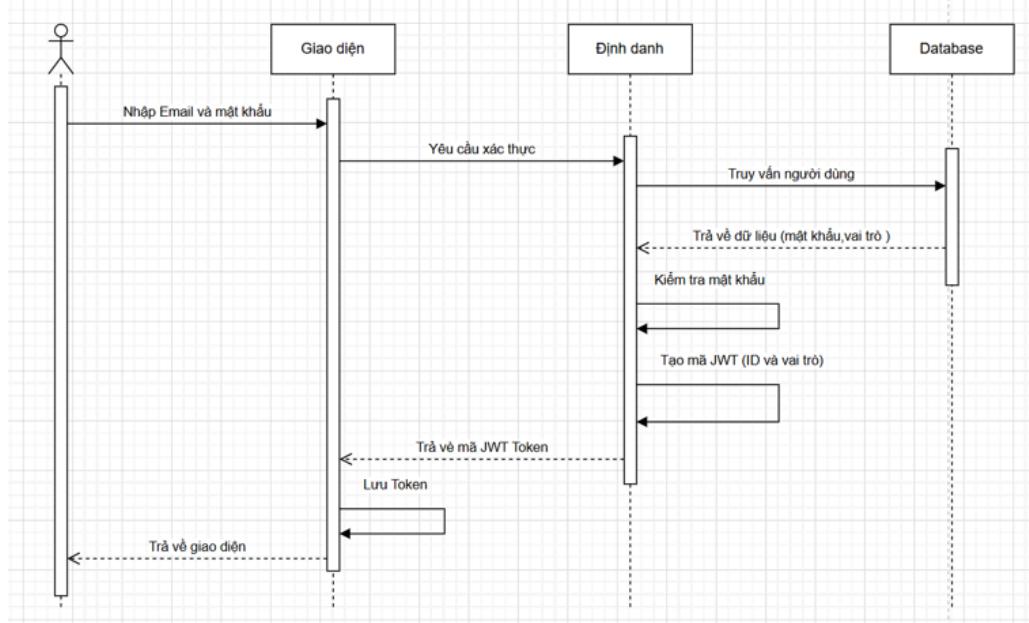


Sơ đồ hoạt động Đánh giá chéo

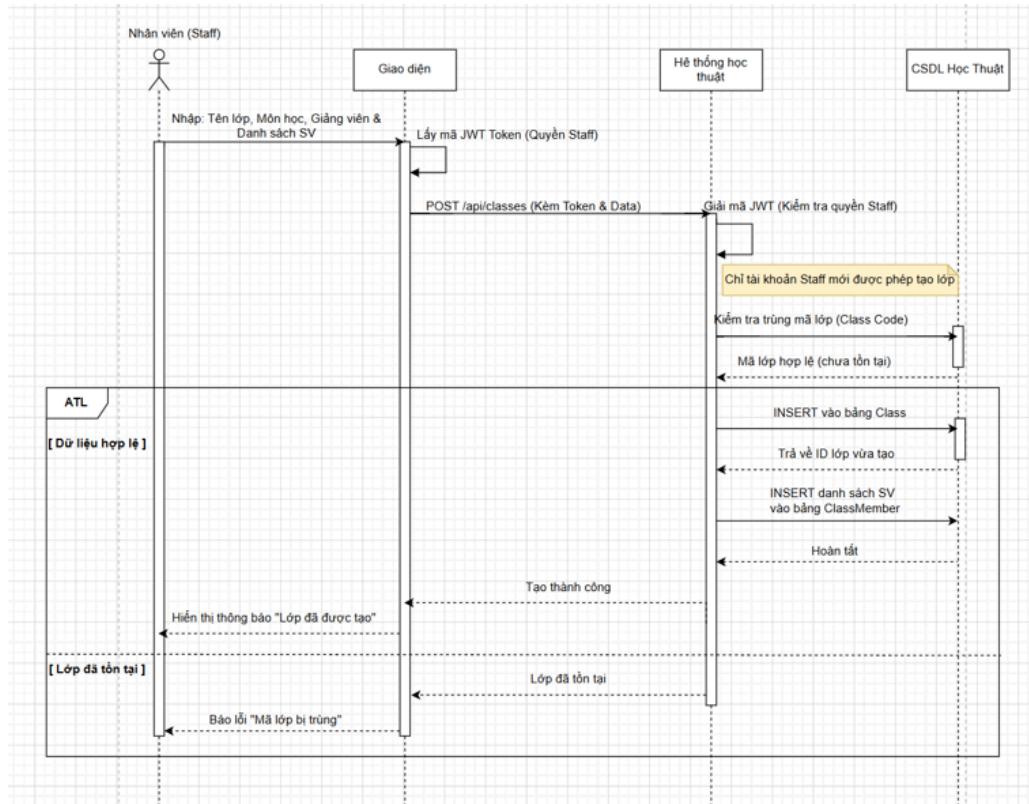


Sơ đồ hoạt động giảng viên chấm điểm và nhận xét

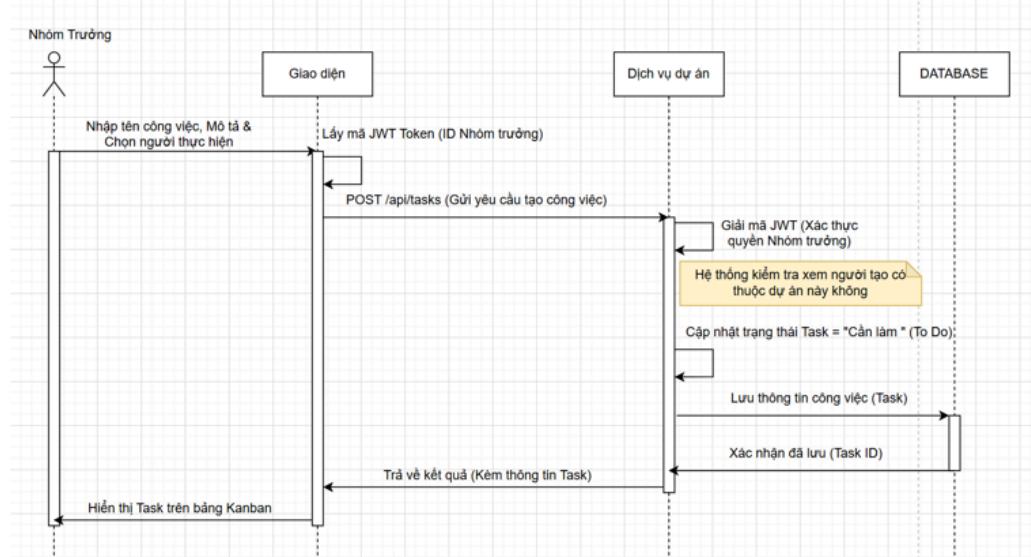
3.3. Xây dựng biểu đồ tuần tự



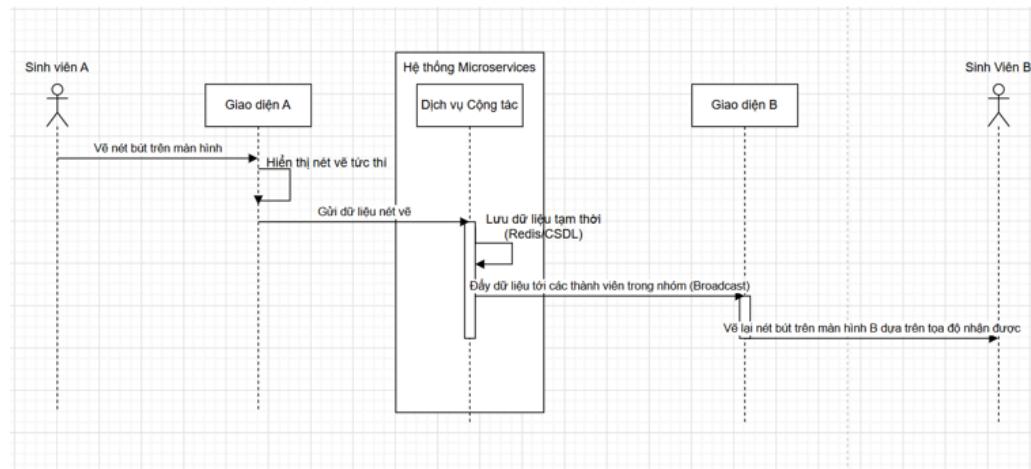
Sơ đồ tuần tự Đăng nhập & Cấp phát JWT



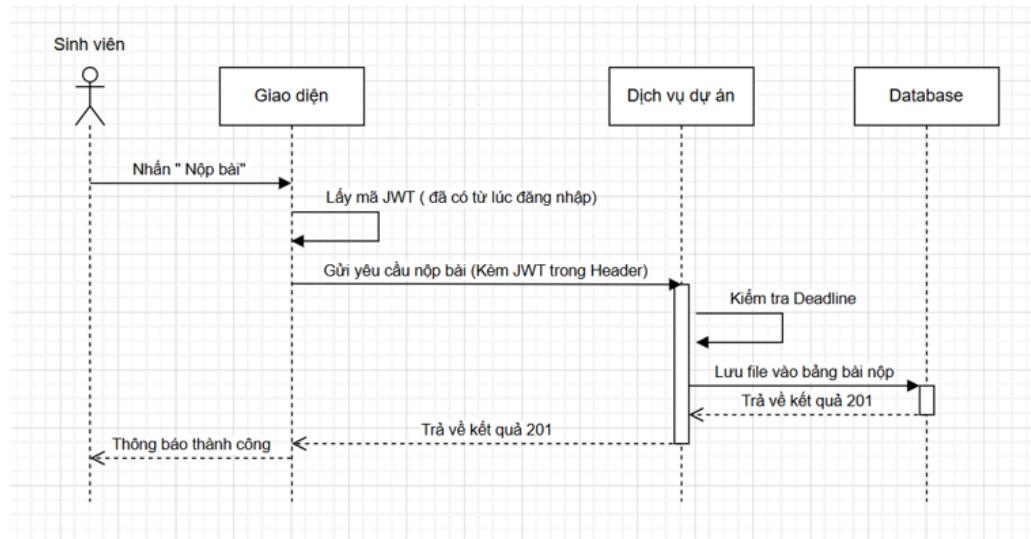
Sơ đồ tuần tự Tạo Lớp học



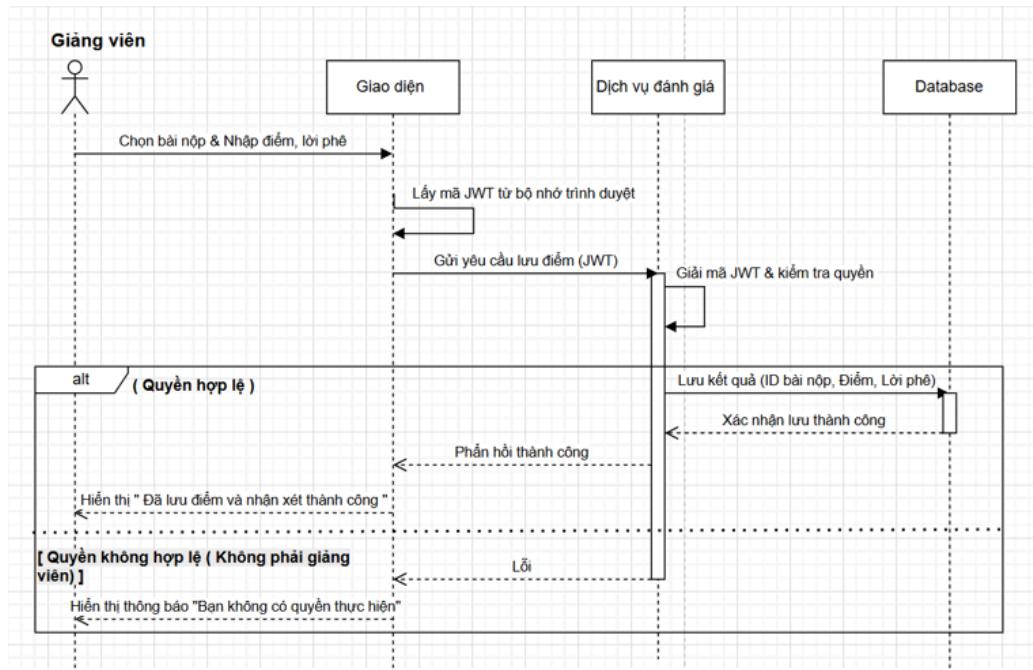
Sơ đồ tuần tự Đề xuất dự án & Giao việc



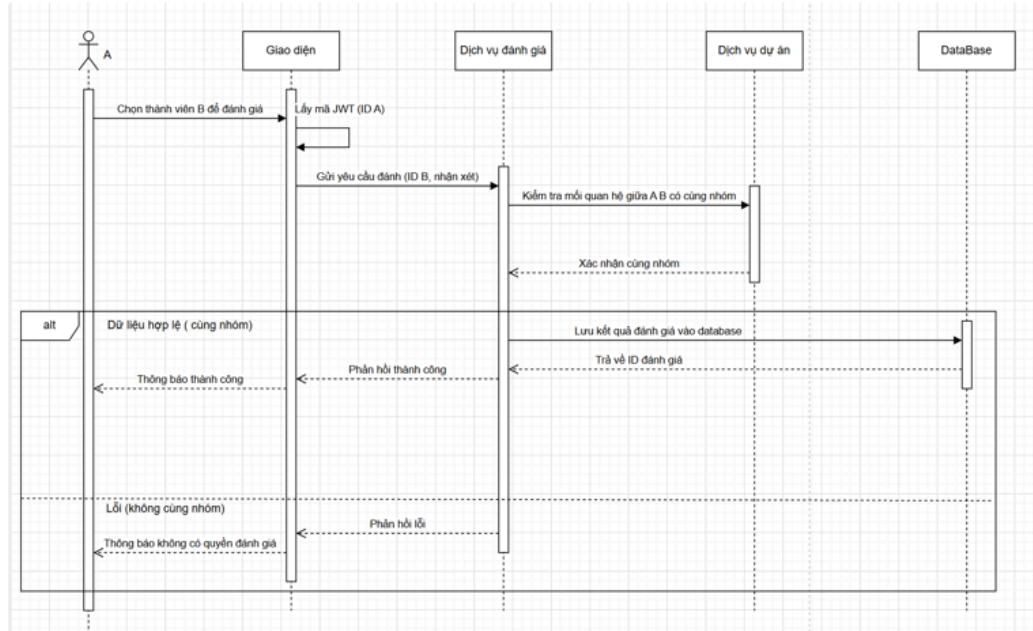
Sơ đồ tuần tự Đồng bộ dữ liệu Whiteboard



Sơ đồ tuần tự Nộp bài Checkpoint



Sơ đồ tuần tự Giảng viên chấm điểm & Nhận xét



Sơ đồ tuần tự Đánh giá chéo thành viên

CHƯƠNG IV. NHIỆM VỤ CỦA CÁC DỊCH VỤ MICROSERVICES

Hệ thống Quản lý Dự án Sinh viên (CollabSphere) được kiến trúc theo mô hình Microservices, trong đó các nghiệp vụ phức tạp được phân rã thành các dịch vụ độc lập. Kiến trúc này đảm bảo tính lỏng lẻo (loose coupling), khả năng mở rộng linh hoạt cho số lượng lớn sinh viên và tính sẵn sàng cao trong các đợt nộp bài cao điểm.

4.1 Xác định và mô tả các dịch vụ cụ thể

STT	Tên Dịch vụ (Service Name)	Cổng (Port)	Cơ sở dữ liệu (Database)	Nhiệm vụ chính
1	API Gateway	8084	(Không lưu trữ)	<ul style="list-style-type: none"> Sử dụng Ocelot/YA RP làm cổng vào duy nhất cho ReactJS Client. Điều phối request và thực hiện xác thực JWT sơ cấp.
2	Identity Service	8083	Postgres QL	<ul style="list-style-type: none"> Quản lý đăng ký, đăng nhập, cấp phát mã định danh JWT. Quản lý hồ sơ (Profile) người dùng và phân quyền hệ thống.
3	Project Service	8086	PostgreSQL + Cloudinary	<ul style="list-style-type: none"> Quản lý nhóm, bảng Kanban và

				<p>tiến độ dự án.</p> <ul style="list-style-type: none"> • Xử lý nộp bài Checkpoint: File được đẩy lên Cloudinary, Database chỉ lưu URL.
4	Evaluation Service	8081	PostgreSQL	<ul style="list-style-type: none"> • Quản lý tiêu chí chấm điểm và bảng điểm. • Xử lý quy trình Đánh giá chéo (Peer Review) giữa các sinh viên.
5	Collab Service	8085	Upstash (Redis)	<ul style="list-style-type: none"> • Quản lý Bảng trắng (Whiteboard) thời gian thực sử dụng  . • Tích hợp WebRTC để gọi video/audio và chia

				sẽ màn hình.
6	Chat Service	8088	PostgreS QL	<ul style="list-style-type: none"> Hệ thống chat nhóm bảo mật sử dụng giao thức Signal. Lưu trữ lịch sử tin nhắn nhóm.
7	AI Assistant	8090	(Không lưu trữ)	<ul style="list-style-type: none"> Tích hợp AWS Bedrock để cung cấp tính năng gợi ý code và tóm tắt nội dung dự án.

4.2 Mô tả cách các dịch vụ hoạt động độc lập và tương tác với nhau

Hệ thống tuân thủ nguyên tắc "Shared Nothing" để đảm bảo khả năng mở rộng và bảo trì dễ dàng.

4.2.1 Hoạt động độc lập

- Độc lập về Dữ liệu (Database per Service):** Mỗi Microservice sở hữu một Database PostgreSQL riêng biệt (Private Schema). Dịch vụ *Project Service* không thể truy cập trực tiếp vào bảng User của *Identity Service*, đảm bảo tính toàn vẹn dữ liệu.
- Độc lập về Triển khai:** Các dịch vụ được đóng gói độc lập. Việc cập nhật tính năng cho *Chat Service* không gây gián đoạn cho *Evaluation Service* đang chạy.

4.2.2 Cơ chế tương tác (Inter-service Communication)

Do dữ liệu bị phân tán, các dịch vụ cần giao tiếp chặt chẽ thông qua các cơ chế sau:

- **Giao tiếp đồng bộ (Synchronous HTTP):** Sử dụng `HttpClient` hoặc thư viện `Refit` trong .NET để gọi API nội bộ.

- Ví dụ: Khi sinh viên thực hiện đánh giá chéo, **Evaluation Service** gọi API sang **Project Service** để xác thực quan hệ thành viên nhóm trước khi lưu điểm.

- **Giao tiếp thời gian thực (Real-time Communication):**

- **Socket.IO & Upstash (Redis):** Đồng bộ nét vẽ trên Bảng trắng ngay lập tức tới tất cả thành viên đang online với độ trễ thấp.
- **WebRTC:** Thiết lập kết nối ngang hàng (Peer-to-Peer) cho luồng Audio/Video, giảm tải băng thông cho Server.

- **Quản lý tài nguyên Media:**

- Thay vì lưu file nặng vào PostgreSQL, hệ thống tải file trực tiếp lên **Cloudinary**. Các dịch vụ chỉ trao đổi với nhau đường dẫn (URL) của file, giúp tối ưu hóa hiệu suất hệ thống.

4.3 Phân tích cách mỗi dịch vụ đóng góp vào toàn bộ hệ thống

4.3.1. Nhóm Dịch Vụ Cốt Lõi (Core Domain Services)

Đây là các dịch vụ chứa logic nghiệp vụ chính, sử dụng PostgreSQL để đảm bảo tính toàn vẹn dữ liệu.

Tên Dịch Vụ	Port	Đóng góp & Vai trò kỹ thuật	Mối quan hệ chính
Identity Service	8083	(Security Foundation) Chịu trách nhiệm về định danh. Quản lý xác thực người dùng và mã hóa mật khẩu. Đảm bảo mọi request trong hệ thống đều gắn với một User ID hợp lệ.	Cung cấp thông tin User (qua JWT) cho Project và Chat Service để định danh người gửi.
Project Service	8086	(Business Logic & Storage) Quản lý vòng đời dự	Cung cấp dữ liệu "Thành viên nhóm" cho

		án. Điểm đặc biệt là khả năng tích hợp Cloudinary để xử lý file nộp bài, giúp giảm tải cho Database chính.	Evaluation Service và Collab Service.
Evaluation Servicee	8081	(Assessment Engine) Xử lý logic chấm điểm phức tạp. Đảm bảo tính công bằng trong quy trình Đánh giá chéo bằng cách xác minh chéo dữ liệu nhóm.	Gọi API sang Project Service để kiểm tra quyền hạn trước khi cho phép sinh viên chấm điểm lẫn nhau.

4.3.2 Nhóm Dịch Vụ Hỗ Trợ & Quản Trị (Support & Admin Services)

Các dịch vụ này sử dụng giao thức đặc biệt (**Socket/Signal**) để tạo trải nghiệm cộng tác tức thì.

Tên Dịch Vụ	Port	Đóng góp & Vai trò kỹ thuật	Mối quan hệ chính
Collab Service	8085	(Sync Engine) Quản lý Bảng trắng (Whiteboard). Sử dụng  Socket.I kết hợp với Upstash (Redis) để đồng bộ nét vẽ tới tất cả thành viên với độ trễ thấp nhất. có điểm số mới.	Sử dụng thông tin Nhóm từ Project Service để tạo các "phòng ảo" (Room) riêng biệt cho từng nhóm.

Chat Service	8088	(Secure Communication) Hệ thống tin nhắn nhóm. Sử dụng giao thức Signal (tương thích tốt với môi trường .NET) để đảm bảo tin nhắn được chuyển đi an toàn và tức thì.	Gắn kết chặt chẽ với Identity Service để hiển thị đúng Avatar và Tên người chat.
AI Assistant	8090	(Intelligence) Tích hợp AWS Bedrock . Đóng vai trò trợ lý ảo, giúp sinh viên giải quyết vấn đề kỹ thuật hoặc tóm tắt tài liệu nhanh chóng.	Nhận ngữ cảnh từ Project Service để đưa ra gợi ý phù hợp.

4.3.3 Nhóm Dịch Vụ Hạ Tầng (Infrastructure Services)

Đóng vai trò nền tảng kết nối Frontend ReactJS với hệ thống Backend.

Tên Dịch Vụ	Port	Đóng góp & Vai trò trong hệ thống
API Gateway	8084	(Traffic Controller) Sử dụng thư viện Ocelot hoặc YARP . Đây là điểm duy nhất mà ReactJS Client nhìn thấy. Nó che giấu sự phức tạp của hệ thống Microservices bên dưới.

