

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



Công nghệ phần mềm - CO3001

Báo cáo bài tập lớn

Hệ thống hỗ trợ Tutor tại Trường Đại học Bách Khoa – ĐHQG TP.HCM

Advisor(s): Nguyễn Thành Công

Lê Dinh Thuận

Mai Đức Trung

Bùi Công Tuấn

Student(s): Trương Thiên Ân 2310190

Nguyễn Huỳnh Thiên Hậu 2310930

Lại Nguyễn Hoàng Hưng 2311327

Bùi Ngọc Phúc 2312665

Nguyễn Thái Sơn 2312968

HO CHI MINH CITY, NOVEMBER 2025

Danh sách thành viên & Khối lượng công việc

STT	Họ và tên	MSSV	Phần trăm công việc
1	Trương Thiên Ân	2310190	100%
2	Nguyễn Huỳnh Thiên Hậu	2310930	100%
3	Lại Nguyễn Hoàng Hưng	2311327	100%
4	Bùi Ngọc Phúc	2312665	100%
5	Nguyễn Thái Sơn	2312968	100%



Contents

1 Project details specification	7
1.1 Project context	7
1.2 Project stakeholders	7
1.3 Objectives and Scopes	8
1.3.1 Objectives	8
1.3.2 Scopes	9
2 Functional requirements	12
2.1 Use-case Diagram cho toàn hệ thống:	12
2.2 Use-case Details/scenario:	17
2.2.1 Use-case Đăng nhập:	17
2.2.2 Use-case Đăng ký nhóm hướng dẫn:	17
2.2.3 Use-case Tổ chức buổi học:	23
2.2.4 Use-case Đăng ký buổi học:	25
2.3 Non-interactive Functional Requirement	28
3 Non-Fucntional Requirements	28
3.1 Product Requirements (Yêu cầu Sản phẩm)	28
3.2 Organisational Requirements (Yêu cầu Tổ chức)	29
4 System Modelling	30
4.1 UI design Mockup	30
4.1.1 Trang chủ và đăng nhập	30
4.1.2 Sinh viên đăng ký nhóm và tutor nhận/từ chối yêu cầu	31
4.1.3 Thông tin nhóm và thao tác đăng ký nhóm mới	31
4.1.4 Thông tin giảng viên và môn học	32
4.1.5 Tutor quản lý buổi học	33
4.1.6 Sinh viên đăng ký buổi học	34
4.1.7 Sinh viên truy cập buổi học	34
4.1.8 Turtor truy cập buổi học	35
4.1.9 Tutor tổ chức và đánh giá buổi học	35
4.1.10 Phòng Công tác Sinh viên	36
4.1.11 Khoa/Bộ môn	37
4.1.12 Phòng Đào tạo & Điều phối viên	38

4.2	Sequence diagrams	39
4.2.1	Sinh viên đăng ký nhóm:	39
4.2.2	Tutor chấp nhận/từ chối nhóm:	41
4.2.3	Tutor mở buổi học:	43
4.2.4	Sinh viên tham gia buổi học:	45
4.2.5	Sinh viên truy cập tài liệu:	47
4.3	Activity diagrams	49
4.3.1	Sinh viên đăng ký nhóm:	49
4.3.2	Tutor chấp nhận/từ chối nhóm yêu cầu:	50
4.3.3	Tutor mở buổi học:	52
4.3.4	Sinh viên tham gia buổi học:	53
4.3.5	Sinh viên truy cập tài liệu:	55
4.4	State-chart diagrams	57

Danh sách hình vẽ

2.1	Use-case Diagram cho toàn hệ thống	12
2.2	Use-case Đăng ký nhóm	13
2.3	Use-case Đăng nhập	13
2.4	Use-case Tổ chức buổi học	14
2.5	Use-case Đăng ký buổi học	14
2.6	Use-case Đăng nhập	17
2.7	Use-case Đăng ký nhóm hướng dẫn	17
2.8	Use-case Tổ chức buổi học	23
2.9	Use-case Đăng ký buổi học	25
4.1	Giao diện trang chủ và đăng nhập: một ảnh ở trên, hai ảnh ở dưới.	30
4.2	Các giao diện của chức năng đăng ký nhóm học.	31
4.3	Các bước thao tác trong quá trình đăng ký nhóm học.	31
4.4	Các giao diện thông tin của giảng viên.	32
4.5	Đăng ký mở buổi học	33
4.6	Chỉnh sửa thông tin buổi học	33
4.7	Giao diện chính quản lý buổi học	33
4.8	Giao diện quản lý buổi học của giảng viên.	33
4.9	Giao diện chức năng đăng ký buổi học của sinh viên.	34
4.10	Các giao diện chức năng của sinh viên.	34



4.11	Các giao diện quản lý buổi học của Tutor.	35
4.12	Các giao diện tổ chức và đánh giá buổi học.	35
4.13	Phòng CTSV xem báo cáo.	36
4.14	Khoa/Bộ môn xem báo cáo.	37
4.15	Chức năng xem báo cáo và phản hồi của PDT & Điều phối viên.	38
4.16	Sequence diagram: Sinh viên đăng ký nhóm	39
4.17	Sequence diagram: Tutor chấp nhận/từ chối nhóm	41
4.18	Sequence diagram: Tutor mở buổi học	43
4.19	Sequence diagram: Sinh viên tham gia buổi học	45
4.20	Sequence diagram: Truy cập tài liệu	47
4.21	Activity diagram: Sinh viên đăng ký nhóm	49
4.22	Activity diagram: Tutor chấp nhận/từ chối nhóm yêu cầu	50
4.23	Activity diagram: Tutor mở buổi học	52
4.24	Activity diagram: Sinh viên tham gia buổi học	53
4.25	Activity diagram: Truy cập tài liệu	55
4.26	State diagram của đối tượng buổi học	57

Danh sách bảng

1.1	Stakeholder trong hệ thống	8
2.1	Mô tả sơ bộ các Use-case	15
2.2	Use-case scenario: Đăng ký nhóm	18
2.3	Use-case scenario: Xem tutor phù hợp	19
2.4	Use-case scenario của use-case Nhận/Từ chối nhóm được yêu cầu	21
2.5	Use-case scenario: Truy cập tài liệu	23
2.6	Use-case scenario: Nhận thông báo	26

Listings

1 Project details specification

1.1 Project context

Trường Đại học Bách Khoa – ĐHQG TP.Hồ Chí Minh(HCMUT) có triển khai chương trình Tutor/Mentor hỗ trợ sinh viên trong quá trình học tập và rèn luyện nhằm giúp sinh viên nâng cao năng lực và phát triển kỹ năng. Chương trình với sự tham gia của rất nhiều bên liên quan, bao gồm tutor (giảng viên, nghiên cứu sinh, hoặc sinh viên năm trên có thành tích học tập tốt) và các sinh viên có nhu cầu được hỗ trợ, bên cạnh đó còn có các phòng ban quản lý (CTSV, Đào tạo) và cả bộ phận điều phối của chương trình.

Do quy mô gồm nhiều bên tham gia, việc quản lý và vận hành là rất khó khăn, trong đó phải kể đến các khó khăn trong việc quản lý thông tin, đăng ký, sắp xếp lịch hẹn, theo dõi, đánh giá và phản hồi...

Chính vì vậy, nhà trường muốn xây dựng một hệ thống phần mềm giúp quản lý và điều phối toàn bộ hoạt động của chương trình, bên cạnh đó còn cần liên kết và đồng bộ với các dịch vụ có sẵn tại trường (dịch vụ xác thực tập trung HCMU_SSO, hệ thống thư viện, dữ liệu của từng tutor/sinh viên sẽ được đồng bộ với hệ thống HCMUT_DATACORE). Bên cạnh đó chương trình còn cần có thể được mở rộng thêm các tính năng thông minh khác nhằm hỗ trợ tối đa việc học tập cho sinh viên.

1.2 Project stakeholders

Stakeholder	Role	Expectation
Sinh viên	Người nhận hỗ trợ từ tutor	Đăng ký, hủy, theo dõi lịch học dễ dàng; chọn tutor phù hợp; phản hồi sau buổi học; nhận được thông báo khi có thay đổi.
Tutor	Người cung cấp hỗ trợ học tập	Thiết lập lịch rảnh; mở/hủy các buổi tư vấn; xem danh sách người học; quản lý các buổi họp; đánh giá sinh viên trực tuyến; nhận phản hồi; theo dõi tiến độ người học; nhận được thông báo khi có thay đổi.



Khoa/Bộ môn	Quản lý đào tạo chuyên môn trong ngành/môn học	Khai thác dữ liệu đánh giá từ hệ thống để nắm tình hình học tập của sinh viên theo từng môn cụ thể, từ đó có cơ sở điều chỉnh hoạt động giảng dạy hoặc tăng cường hỗ trợ; khoa chỉ quản lý cho các môn học/lĩnh vực của khoa.
Phòng Đào tạo	Bộ phận giám sát chương trình, đánh giá hiệu quả để phân bổ nguồn lực	Nhận báo cáo tổng quan của chương trình; đọc được phản hồi của người học; sử dụng dữ liệu báo cáo, phản hồi để tối ưu phân bổ nguồn lực cho toàn hệ thống (tăng/giảm số lượng tutor, ưu tiên môn học).
Phòng CTSV	Bộ phận theo dõi và đánh giá hoạt động sinh viên	Nhận được tiến độ và kết quả tham gia của người học để cộng điểm rèn luyện hoặc xét học bổng cho sinh viên.
Điều phối viên	Người quản lý hệ thống	Duy trì hệ thống ổn định, hỗ trợ khi hệ thống gặp vấn đề.

Bảng 1.1: Stakeholder trong hệ thống

1.3 Objectives and Scopes

1.3.1 Objectives

Hệ thống **Tutor Support** được phát triển nhằm đạt được các mục tiêu sau:

Mục tiêu tổng quát: Xây dựng một nền tảng phần mềm toàn diện, hiện đại và thân thiện với người dùng để số hóa và tối ưu hóa toàn bộ quy trình vận hành Chương trình tutor/mentor tại HCMUT, từ khâu đăng ký, ghép cặp đến theo dõi, đánh giá và báo cáo.

Mục tiêu cụ thể:

- **Đối với quy trình hoạt động:**

- **Tự động hóa và tối ưu hóa:** Giảm thiểu đáng kể các thao tác thủ công, giảm thời gian chờ đợi và sai sót trong các quy trình như đăng ký, sắp xếp lịch hẹn, ghép cặp Tutor-Sinh viên.
- **Thiết lập kênh trung gian:** Tạo một kênh giao tiếp giữa sinh viên, tutor và các phòng ban (Phòng Đào tạo, Phòng CTSV, khoa/bộ môn), cho phép các bên trao

đổi, báo cáo và theo dõi thông tin một cách an toàn, được bảo mật và lưu trữ trên hệ thống.

- **Nâng cao chất lượng dịch vụ:** Cung cấp công cụ để thu thập phản hồi, đánh giá liên tục, từ đó giúp tất cả các bên tham gia cải thiện chất lượng hoạt động và hiệu quả công việc dựa trên dữ liệu một cách tổng thể.

- **Đối với người dùng:**

- **Sinh viên:** Được cung cấp trải nghiệm đăng ký dễ dàng, chủ động lựa chọn hoặc được đề xuất tutor phù hợp, quản lý lịch học linh hoạt và dễ dàng đưa ra phản hồi.
- **Tutor:** Được cung cấp công cụ quản lý lịch trình, danh sách học viên, theo dõi tiến độ và ghi chú buổi học một cách hiệu quả, chuyên nghiệp.
- **Các phòng ban:** (Phòng Đào tạo, CTSV, khoa/bộ môn) được cung cấp góc nhìn tổng quan (dashboard) và các báo cáo chi tiết, dữ liệu phân tích để ra quyết định dựa trên dữ liệu về hiệu quả chương trình và phân bổ nguồn lực.

- **Đối với hệ thống công nghệ:**

- **Tích hợp và đồng bộ:** Đảm bảo tính hợp liền mạch và an toàn với hạ tầng công nghệ hiện có của HCMUT (HCMUT_SSO, HCMUT_DATACORE, HCMUT_LIBRARY), đảm bảo tính nhất quán dữ liệu và bảo mật.
- **Khả năng mở rộng và bảo trì:** Thiết kế kiến trúc hệ thống module hóa, linh hoạt, dễ dàng mở rộng để triển khai các tính năng nâng cao (như AI Matching, cộng đồng trực tuyến) trong tương lai.

1.3.2 Scopes

In-Scope Để đảm bảo dự án khả thi và tập trung, phạm vi được xác định rõ ràng cho giai đoạn đầu tiên (MVP – Minimum Viable Product) như sau:

- **Quản lý người dùng và phân quyền:**

- Hỗ trợ 4 vai trò chính: Sinh viên, tutor, điều phối viên, các phòng ban (Phòng Đào tạo, CTSV, khoa/bộ môn).
- Tích hợp xác thực tập trung HCMUT_SSO.
- Đồng bộ hóa dữ liệu cá nhân từ HCMUT_DATACORE.
- Phân quyền tự động dựa trên vai trò từ hệ thống trung tâm.



- **Cập nhật nhu cầu:** Sinh viên có thể cập nhật các môn học cần được hỗ trợ.
- **Đăng ký và ghép cặp:**
 - Sinh viên đăng ký tham gia chương trình theo nhóm hoặc cá nhân.
 - Sinh viên có thể tự do tìm kiếm và chọn tutor liên quan đến môn học cần được hỗ trợ.
 - Tutor có thể đăng ký làm người hỗ trợ, đồng ý hoặc từ chối nhóm được yêu cầu.
- **Đăng ký buổi học:**
 - Sinh viên có thể tham gia, hủy tham gia buổi học.
 - Tutor có thể mở, huỷ buổi học, điều chỉnh các thông tin của buổi học.
 - Hệ thống tự động gửi thông báo và nhắc lịch qua email.
- **Theo dõi và đánh giá:**
 - Sinh viên gửi phản hồi, đánh giá sau mỗi buổi học.
 - Tutor theo dõi tiến độ, ghi chú cho từng sinh viên.
- **Báo cáo và thống kê:**
 - Cung cấp báo cáo tổng quan về số lượng buổi học, mức độ hài lòng cho Phòng Đào tạo.
 - Cung cấp báo cáo kết quả tham gia cho Phòng CTSV.
- **Tích hợp thư viện số:** Cho phép tutor và sinh viên truy cập, liên kết đến tài liệu học tập từ HCMUT_LIBRARY trong khuôn khổ buổi học.

Out-of-Scope

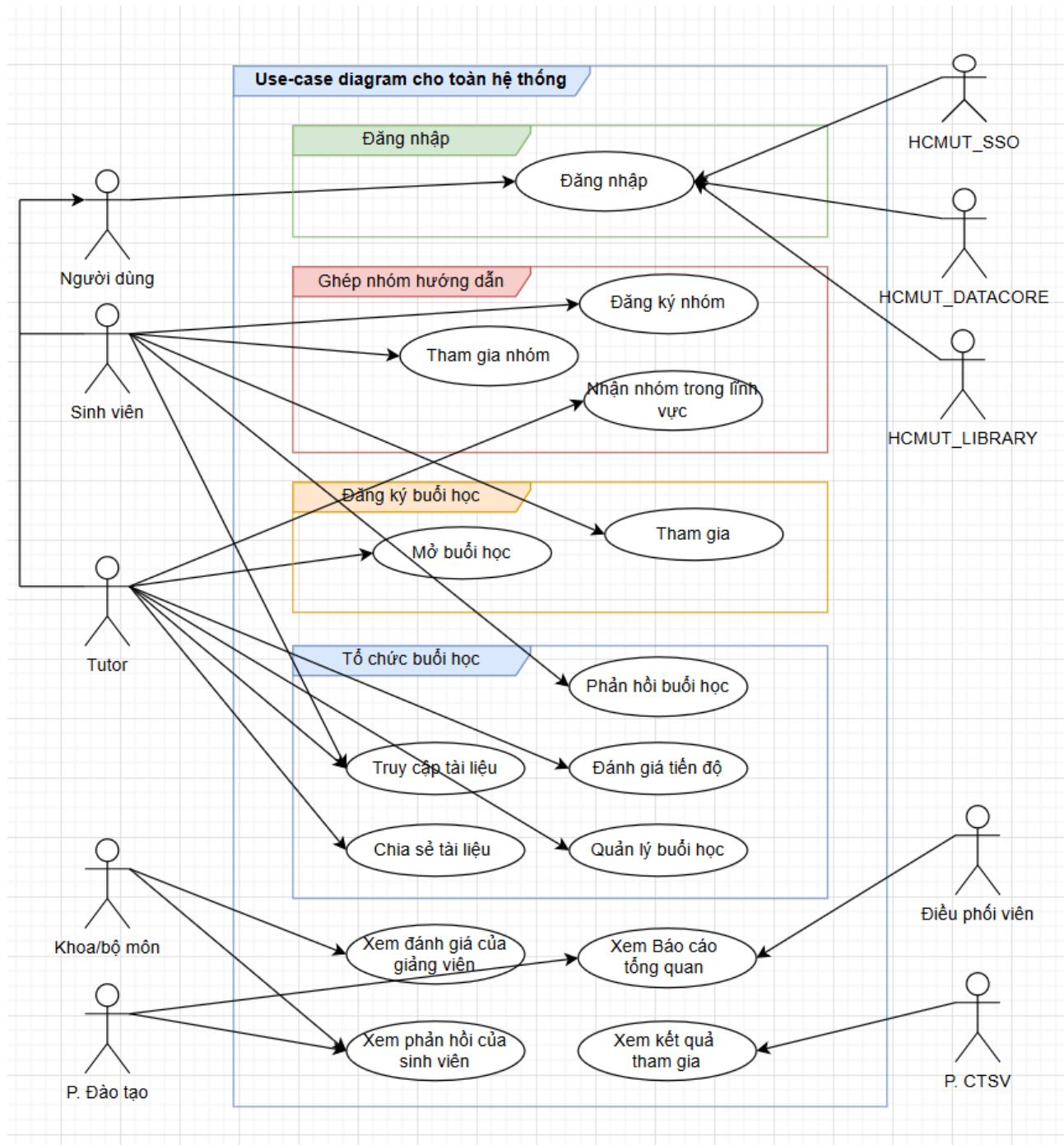
- **Tính năng nâng cao và AI:** Các chức năng như AI Matching tự động, gợi ý thông minh dựa trên học máy, phân tích nâng cao hành vi học tập của sinh viên sẽ chưa được triển khai.
- **Tích hợp ngoài HCMUT:** Các hệ thống bên ngoài HCMUT (như LMS quốc tế, nền tảng MOOC, Zoom/MS Teams nâng cao ngoài hạ tầng trường) không nằm trong phạm vi tích hợp.

- **Thanh toán và quản lý tài chính:** Hệ thống không xử lý các chức năng liên quan đến phí dịch vụ, quản lý tài chính hay thanh toán trực tuyến.
- **Quản lý cộng đồng trực tuyến:** Các tính năng diễn đàn, mạng xã hội học tập, chat nhóm mở rộng, chia sẻ tài nguyên ngang hàng chưa được đưa vào.
- **Chức năng Mobile App:** Chỉ hỗ trợ phiên bản web.

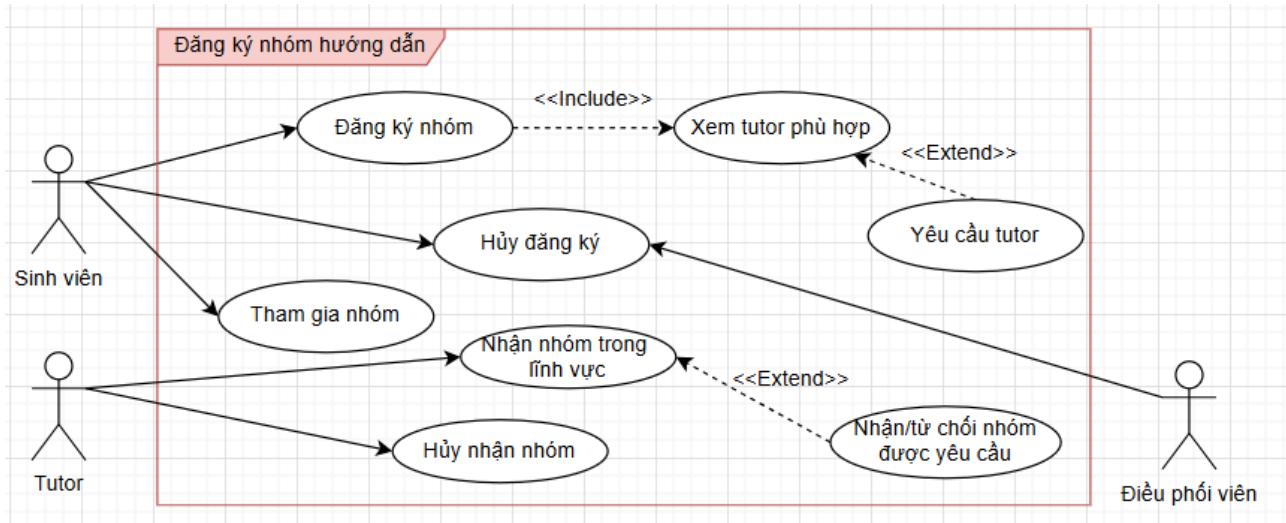


2 Functional requirements

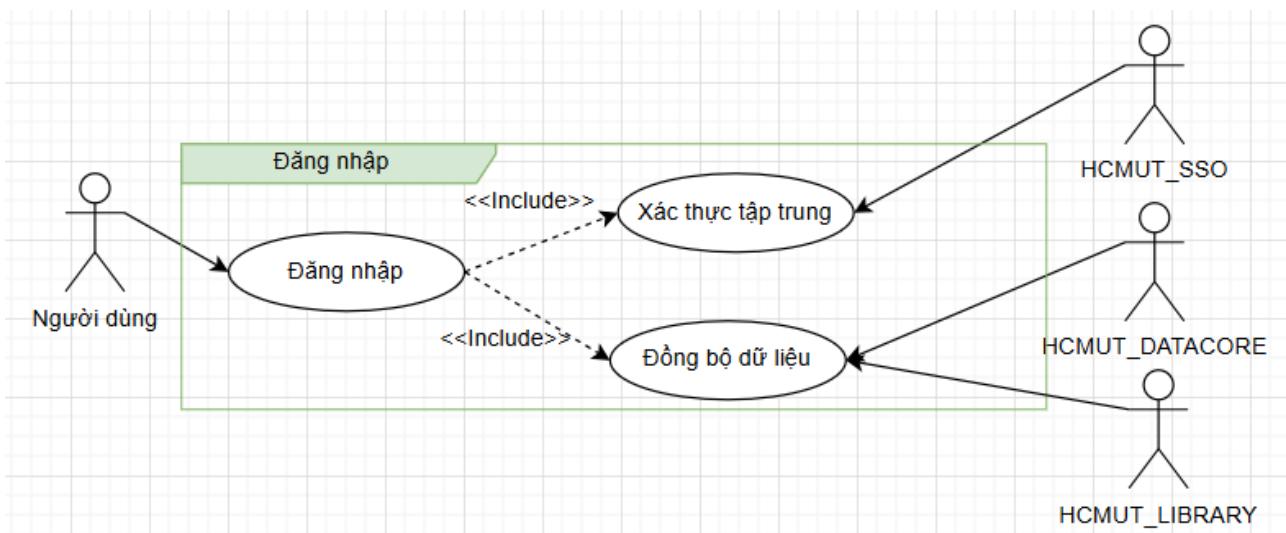
2.1 Use-case Diagram cho toàn hệ thống:



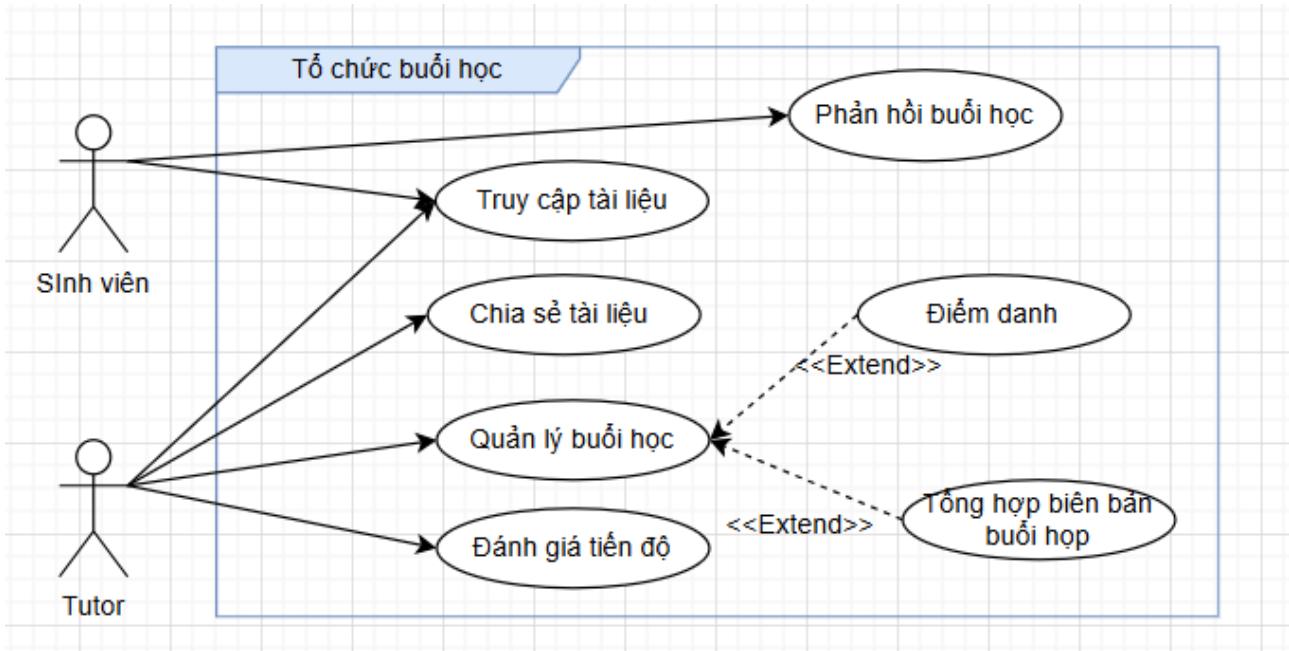
Hình 2.1: Use-case Diagram cho toàn hệ thống



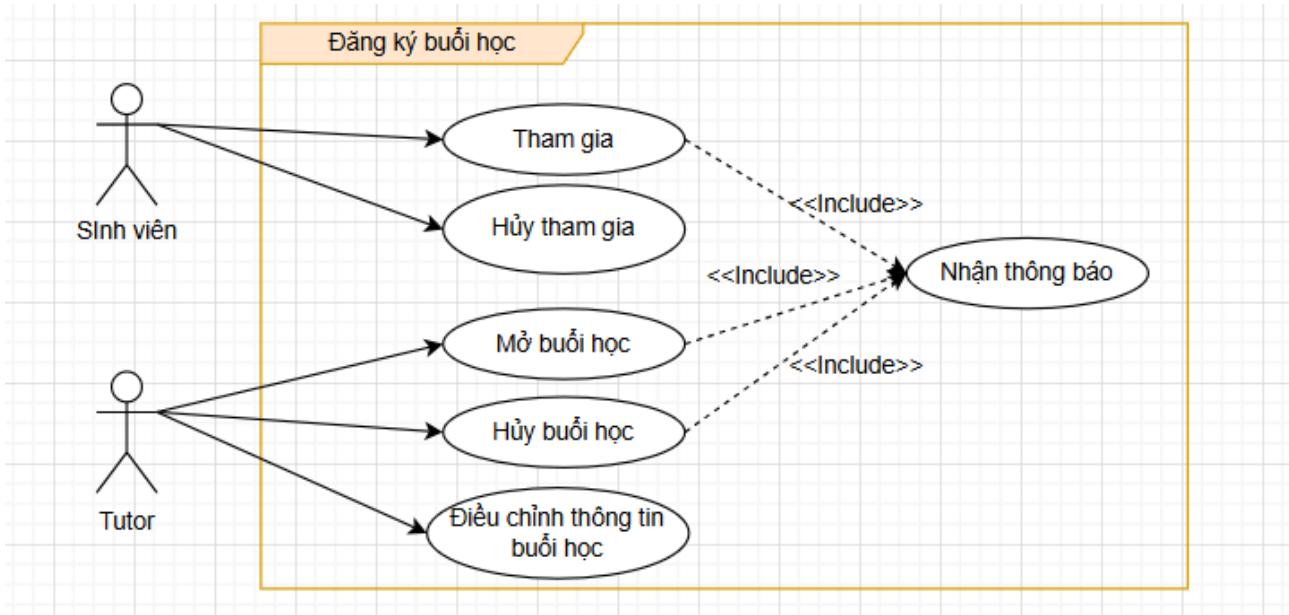
Hình 2.2: Use-case Đăng ký nhóm



Hình 2.3: Use-case Đăng nhập



Hình 2.4: Use-case Tổ chức buổi học



Hình 2.5: Use-case Đăng ký buổi học

Bảng 2.1: Mô tả sơ bộ các Use-case

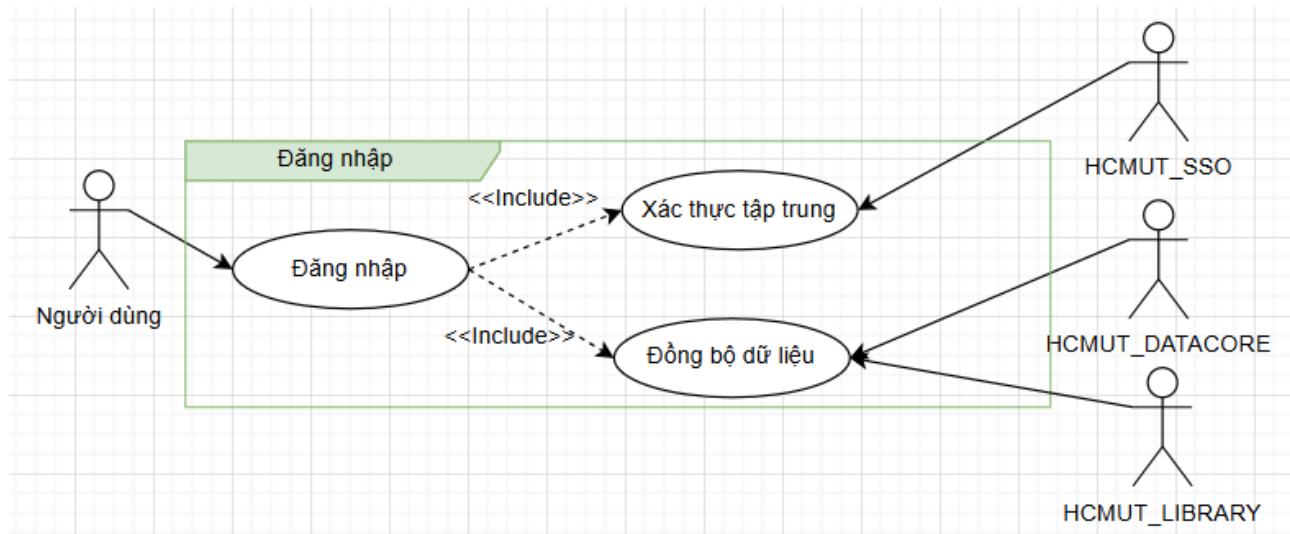
Tên Use-case	Mô tả sơ bộ
Dăng nhập	Người dùng (sinh viên, tutor, cán bộ) đăng nhập vào hệ thống bằng tài khoản BKNetID thông qua hệ thống SSO của trường (HCMUT_SSO).
Xác thực tập trung	Hệ thống kết nối với HCMUT_SSO để thực hiện phân quyền tự động cho người dùng.
Đồng bộ dữ liệu	Hệ thống đồng bộ dữ liệu từ HCMUT_DATACORE để đảm bảo dữ liệu luôn được cập nhật mới nhất và chính xác.
Xem đánh giá của giảng viên	Khoa/bộ môn xem lại các nhận xét, kết quả và đánh giá của giảng viên dành cho sinh viên.
Xem phản hồi của sinh viên	Khoa/bộ môn và phòng Đào tạo xem lại những phản hồi của sinh viên về công tác giảng dạy, phương pháp giảng dạy, đánh giá giảng viên...
Xem báo cáo tổng quan	Phòng Đào tạo, Điều phối viên quản lý mở trang báo cáo để xem số buổi học, mức độ tham gia, tỉ lệ huỷ/đổi lịch, điểm/feedback, và xu hướng theo thời gian. Hệ thống hiển thị biểu đồ/bảng và cho phép lọc theo học kỳ, khoa, môn, tutor, nhóm sinh viên để phục vụ theo dõi và ra quyết định.
Xem kết quả tham gia	Phòng Công tác Sinh viên xem tổng hợp quá trình tham gia chương trình tutor: số buổi học, trạng thái hoàn thành, điểm rèn luyện hoặc tiêu chí xét học bỗng. Hệ thống hiển thị kết quả theo từng học kỳ và cho phép tải hoặc in báo cáo khi cần.
Dăng ký nhóm	Sinh viên đăng ký nhu cầu hỗ trợ theo cơ chế nhóm, hệ thống tạo và thêm yêu cầu vào các danh sách yêu cầu.
Xem tutor phù hợp	Hiển thị danh sách các tutor phù hợp trong lĩnh vực.
Yêu cầu tutor	Sinh viên gửi yêu cầu xin được hướng dẫn nhu cầu hỗ trợ đã đăng ký.
Huỷ đăng ký	Sinh viên huỷ tham gia nhóm đã đăng ký.
Tham gia nhóm	Khi sinh viên có nhu cầu hướng dẫn theo nhóm, sinh viên đăng ký tham gia nhóm đã được đăng ký thành công.
Huỷ nhận nhóm	Người hướng dẫn huỷ hướng dẫn nhóm đã nhận hướng dẫn.



Tên Use-case	Mô tả sơ bộ
Chia sẻ tài liệu	Tutor có thể tải lên và chia sẻ tài liệu học tập hoặc bài tập cho các buổi học.
Truy cập tài liệu	Sinh viên và tutor có thể truy cập vào kho tài liệu đã được chia sẻ để phục vụ cho buổi học.
Quản lý buổi học	Tutor quản lý nội dung, thời gian, và các hoạt động trong buổi học. Ngoài ra có thể điểm danh và tổng hợp biên bản buổi học.
Phản hồi buổi học	Sinh viên có thể gửi phản hồi sau mỗi buổi học về nội dung, cách giảng dạy và mức độ hiệu quả.
Dánh giá tiến độ	Tutor có thể đánh giá tiến độ học tập, sự tiến bộ của các sinh viên trong quá trình học tập.
Tham gia buổi học	Sinh viên có thể chọn tham gia vào buổi học đã được mở sẵn.
Huỷ tham gia buổi học	Sinh viên có thể chọn huỷ tham gia khỏi buổi học mà bản thân đã đăng ký tham gia trước.
Mở buổi học	Tutor có thể mở những buổi học phù hợp với lịch trình cá nhân để sinh viên tham gia.
Huỷ buổi học	Tutor có thể huỷ các buổi học bản thân đã mở.
Điều chỉnh thông tin buổi học	Tutor có thể linh hoạt chỉnh sửa thông tin của buổi học: thời gian, phòng học, các thông báo đến sinh viên.
Nhận thông báo	Hệ thống gửi thông báo cho Tutor và Sinh viên khi có sự kiện liên quan đến buổi học (nhắc lịch, thông báo huỷ buổi học).

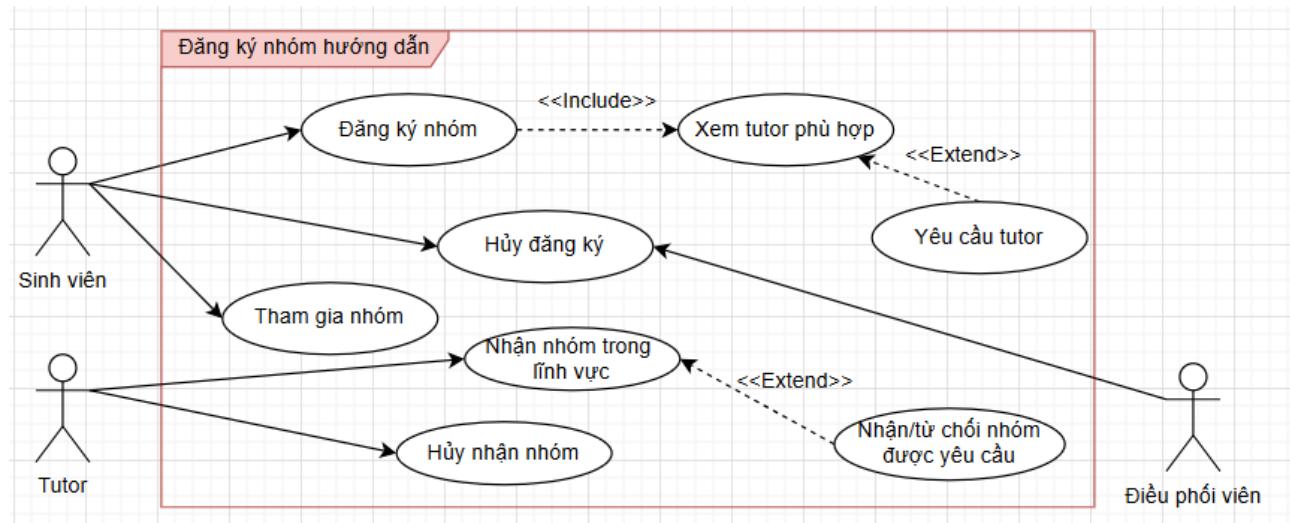
2.2 Use-case Details/scenario:

2.2.1 Use-case Đăng nhập:



Hình 2.6: Use-case Đăng nhập

2.2.2 Use-case Đăng ký nhóm hướng dẫn:



Hình 2.7: Use-case Đăng ký nhóm hướng dẫn



Bảng 2.2: Use-case scenario: Đăng ký nhóm

Use-case name	Đăng ký nhóm
Use-case ID	DK-01
Use-case overview	Sinh viên đăng ký nhu cầu hỗ trợ theo cơ chế tạo nhóm, hệ thống tạo và thêm yêu cầu vào kho danh sách các yêu cầu.
Actors	Sinh viên
Preconditions	Người dùng đã đăng nhập thành công vào hệ thống.
Trigger	Sinh viên đăng ký một nhóm mới.
Normal flow	<ol style="list-style-type: none">1. Sinh viên chọn đăng ký nhóm.2. Sinh viên chọn lĩnh vực/môn học cần hỗ trợ, đặt tiêu đề, điền mô tả nội dung cần hỗ trợ, chọn số lượng thành viên và gửi yêu cầu.3. Hệ thống tạo yêu cầu thành công và đặt trạng thái đăng ký là "đang chờ hướng dẫn".4. Hệ thống thêm yêu cầu vào kho danh sách yêu cầu của lĩnh vực.5. Hệ thống thêm yêu cầu vào kho danh sách yêu cầu của sinh viên đăng ký.6. Hệ thống xác nhận tạo yêu cầu thành công và gửi alert.
Alternative flow	Không có.
Exception flow	<ul style="list-style-type: none">• 2a. Sinh viên huỷ yêu cầu → Use-case dừng lại.• 4a. Hệ thống thêm yêu cầu vào kho danh sách không thành công → hiển thị lỗi và Use-case dừng lại.



Post-conditions	<ul style="list-style-type: none"> Sinh viên đăng ký nhóm thành công, và yêu cầu được thêm vào danh sách các yêu cầu của sinh viên. Hệ thống tạo yêu cầu thành công và yêu cầu được thêm vào danh sách các yêu cầu của lĩnh vực.
------------------------	--

Bảng 2.3: Use-case scenario: Xem tutor phù hợp

Use-case name	Xem tutor phù hợp
Use-case ID	GN02
Use-case overview	Hệ thống hiển thị danh sách các Tutor để Sinh viên lựa chọn đăng ký.
Actors	Sinh viên
Trigger	Sau khi Sinh viên hoàn thành “Chọn lĩnh vực” (trong luồng Đăng ký nhóm), hệ thống tự động chuyển sang bước hiển thị danh sách Tutor.
Preconditions	<ul style="list-style-type: none"> Sinh viên đã đăng nhập thành công vào hệ thống. Sinh viên đã hoàn thành use-case “Chọn lĩnh vực”. Hệ thống có sẵn thông tin về các Tutor đang hoạt động và lĩnh vực hướng dẫn của họ.



Main flow	<ol style="list-style-type: none">1. Hệ thống tiếp nhận tiêu chí mà Sinh viên đã chọn.2. Truy vấn CSDL để tìm tất cả các Tutor có chuyên môn phù hợp.3. Sắp xếp danh sách theo tên (thứ tự alphabet – mặc định).4. Hiển thị danh sách Tutor phù hợp.5. Sinh viên có thể xem thông tin chi tiết của Tutor.6. Sinh viên chọn một Tutor để đăng ký hoặc yêu cầu Tutor phù hợp.
Alternative flows	<p>Không tìm thấy Tutor phù hợp:</p> <ul style="list-style-type: none">• Hệ thống hiển thị thông báo "Xin lỗi, hiện không có Tutor nào trong lĩnh vực đã chọn".• Gợi ý các Tutor ở lĩnh vực tương tự.• Dưa ra tùy chọn: quay lại chọn lĩnh vực khác, hoặc gửi yêu cầu Tutor phù hợp.
Exception flows	<p>Lỗi kết nối CSDL:</p> <ul style="list-style-type: none">• Hệ thống không thể truy cập CSDL Tutor tại bước 2.• Hiển thị thông báo lỗi kỹ thuật, yêu cầu Sinh viên thử lại sau.
Post-conditions	Sinh viên đã xem được danh sách Tutor phù hợp, có thể xem hồ sơ và lựa chọn hành động tiếp theo (chọn Tutor hoặc gửi yêu cầu).

Bảng 2.4: Use-case scenario của use-case Nhận/Từ chối nhóm được yêu cầu

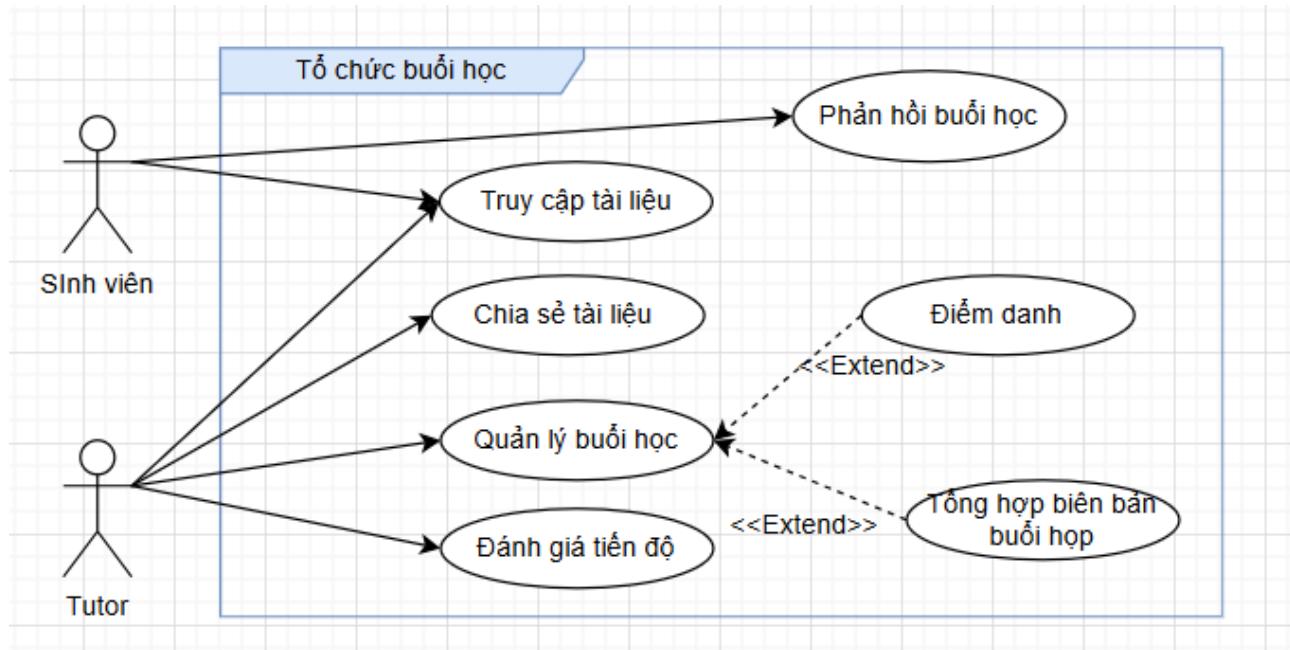
Use-case name	Nhận / Từ chối nhóm được yêu cầu
Use-case ID	GN01
Use-case overview	Cho phép Tutor chấp nhận hoặc từ chối yêu cầu hướng dẫn nhóm sinh viên trong lĩnh vực phụ trách.
Actors	Tutor
Preconditions	<ul style="list-style-type: none"> • Tutor có lĩnh vực hướng dẫn. • Có ít nhất một nhóm sinh viên đã đăng ký trong lĩnh vực đó.
Trigger	Hệ thống phát sinh yêu cầu mới cho Tutor khi có sinh viên đăng ký nhóm trong lĩnh vực mà Tutor hướng dẫn, đồng thời gửi thông báo đến Tutor.
Main flow	<ol style="list-style-type: none"> 1. Tutor nhận thông báo (email/Dashboard). 2. Xem thông tin nhóm (thành viên, mô tả, lịch mong muốn). 3. Chọn Chấp nhận hoặc Từ chối và (tuỳ chọn) nhập lý do. 4. Hệ thống ghi nhận quyết định. 5. Cập nhật trạng thái nhóm: <ul style="list-style-type: none"> • Nếu chấp nhận: trạng thái Đã nhận và gắn nhóm với Tutor. • Nếu từ chối: trạng thái Bị từ chối, giải phóng nhóm và (nếu bật) gợi ý Tutor khác. 6. Hệ thống xếp hàng gửi thông báo cho sinh viên.



Alternative flows	Tutor đã đủ số lượng nhóm (vượt quota) → Hệ thống chặn lưu, hiển thị cảnh báo và gợi ý chuyển cho Điều phối viên.
Exception flows	Nếu mất kết nối hoặc timeout: <ul style="list-style-type: none">Hiển thị lỗi “Không thể lưu quyết định, hãy thử lại trong vài phút”.Trạng thái nhóm không đổi (Chờ xử lý).
Post-conditions	<ul style="list-style-type: none">Nếu chấp nhận: trạng thái Đã nhận, gắn Tutor; thông báo đã được xếp hàng gửi; ghi audit log.Nếu từ chối: trạng thái Bị từ chối; thông báo đã được xếp hàng gửi; ghi audit log.



2.2.3 Use-case Tổ chức buổi học:



Hình 2.8: Use-case Tổ chức buổi học

Bảng 2.5: Use-case scenario: Truy cập tài liệu

Use-case name	Truy cập tài liệu
Use-case ID	TC02
Use-case overview	Sinh viên hoặc Tutor muốn truy cập kho tài liệu học tập để xem hoặc tải về nhằm phục vụ cho buổi học.
Actors	Sinh viên, Tutor
Trigger	Người dùng (Sinh viên, Tutor) muốn mở và sử dụng tài liệu liên quan đến buổi học.

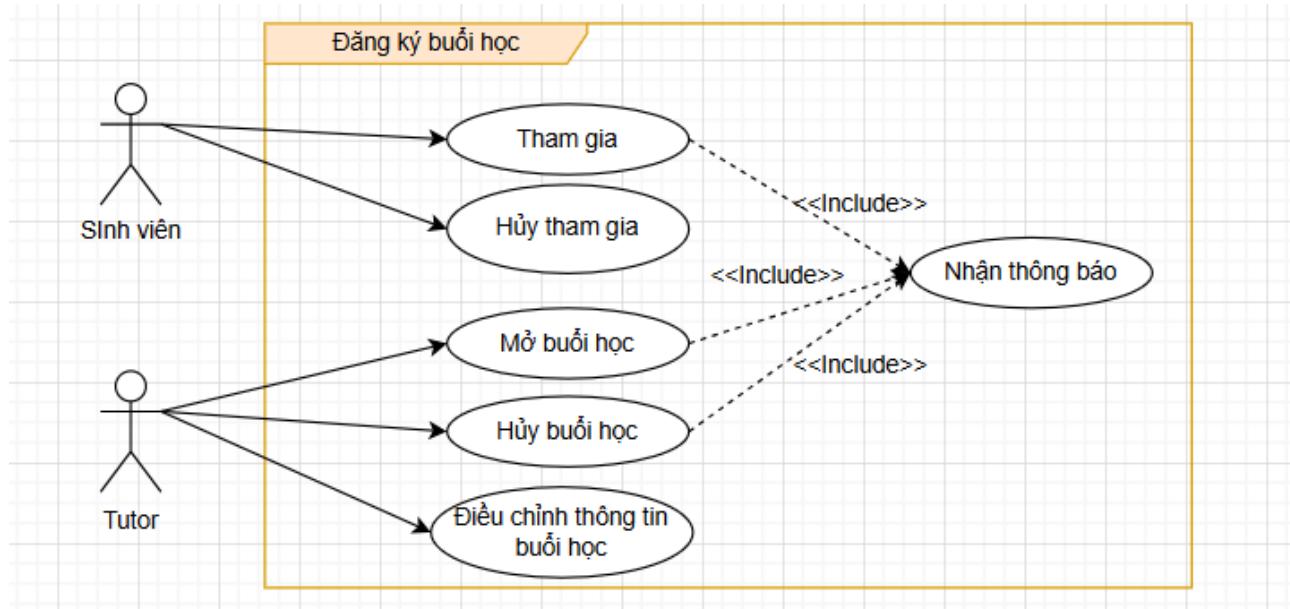


Preconditions	<ul style="list-style-type: none">• Người dùng đã đăng nhập hệ thống thành công.• Người dùng có quyền truy cập vào tài liệu của nhóm/buổi học.• Tài liệu đã được Tutor hoặc thành viên chia sẻ và lưu trữ trong hệ thống.
Main flow	<ol style="list-style-type: none">1. Người dùng đăng nhập vào hệ thống.2. Chọn mục “Truy cập tài liệu” trong menu buổi học.3. Hệ thống hiển thị danh sách tài liệu đã được chia sẻ.4. Người dùng chọn tài liệu cần xem hoặc tải về.5. Hệ thống mở tài liệu (nếu xem trực tuyến) hoặc tải file xuống thiết bị.6. Hệ thống ghi nhận người dùng đã truy cập tài liệu.
Alternative flows	<ul style="list-style-type: none">• 3a. Danh sách trống:<ul style="list-style-type: none">– Hệ thống hiển thị thông báo “Chưa có tài liệu nào được chia sẻ”.– Use-case kết thúc.• 4a. Tìm kiếm tài liệu theo từ khóa:<ul style="list-style-type: none">– Người dùng nhập từ khóa tìm kiếm.– Hệ thống lọc và hiển thị kết quả phù hợp.– Use-case tiếp tục từ bước 5.



Exception flows	<ul style="list-style-type: none"> • 5a. Lỗi kết nối hoặc file không tồn tại: <ul style="list-style-type: none"> – Hệ thống hiển thị “Không thể truy cập tài liệu, vui lòng thử lại sau”. – Use-case kết thúc. • Không có quyền truy cập: <ul style="list-style-type: none"> – Hệ thống hiển thị “Bạn không có quyền xem tài liệu này”. – Use-case kết thúc.
Post-conditions	<ul style="list-style-type: none"> • Người dùng xem hoặc tải về tài liệu thành công. • Hệ thống ghi nhận hoạt động truy cập.

2.2.4 Use-case Đăng ký buổi học:



Hình 2.9: Use-case Đăng ký buổi học



Bảng 2.6: Use-case scenario: Nhận thông báo

Use-case name	Nhận thông báo
Use-case ID	DK06
Use-case overview	Hệ thống gửi thông báo cho Tutor và Sinh viên khi có sự kiện liên quan đến buổi học (nhắc lịch, thông báo hủy buổi học).
Actors	Tutor, Sinh viên
Preconditions	<ul style="list-style-type: none">• Hệ thống đã có thông tin về buổi học.• Người dùng (Tutor/Sinh viên) đã đăng ký tham gia hoặc tạo buổi học.• Kênh thông báo của hệ thống đang hoạt động.
Trigger	<ul style="list-style-type: none">• Tutor mở buổi học mới.• Sinh viên đăng ký tham gia buổi học.• Tutor hủy buổi học.



Main flow	<ol style="list-style-type: none"> 1. Hệ thống kiểm tra sự kiện kích hoạt (mở buổi học, tham gia, hủy buổi học). 2. Hệ thống xác định danh sách người nhận thông báo: <ul style="list-style-type: none"> • Khi Tutor mở buổi học → Tutor nhận thông báo nhắc lịch dạy. • Khi Sinh viên tham gia buổi học → Sinh viên nhận thông báo nhắc lịch học. • Khi Tutor hủy buổi học → Tất cả sinh viên trong buổi học nhận thông báo hủy. 3. Hệ thống tạo nội dung thông báo (thời gian, địa điểm, trạng thái). 4. Hệ thống gửi thông báo qua kênh thông báo của hệ thống. 5. Người dùng nhận thông báo.
Alternative flows	<ul style="list-style-type: none"> • Tutor thay đổi thông tin buổi học → Hệ thống gửi thông báo cập nhật thông tin cho Sinh viên. • Sinh viên rút khỏi buổi học trước khi Tutor hủy → Hệ thống chỉ gửi thông báo đến những Sinh viên còn lại.
Exception flows	<ul style="list-style-type: none"> • Kết nối Internet của Tutor/Sinh viên bị gián đoạn → Thông báo không được nhận. • Lỗi đồng bộ lịch từ server → Thông báo bị trễ hoặc không gửi được.
Post-conditions	Người dùng (Tutor/Sinh viên) nhận được thông báo phù hợp với sự kiện, đúng đối tượng, đúng thời điểm.



2.3 Non-interactive Functional Requirement

Non-interactive Functional Requirement là các chức năng hệ thống tự động thực hiện, không cần người dùng thao tác trực tiếp tại thời điểm đó. Thường là các quy trình ngầm, kích hoạt bởi sự kiện hoặc thời gian. Theo đặc tả của bài tập lớn lần này, nhóm em liệt kê các **Non-interactive Functional Requirement** sau:

- Tự động gửi thông báo nhắc lịch đến cho sinh viên và giảng viên.
- Đồng bộ dữ liệu từ HCMUT_SSO, HCMUT_DATACORE.
- Phân quyền tự động dựa vào hệ thống của trường.
- Kết nối với HCMUT_LIBRARY để truy cập và chia sẻ tài liệu.
- Cung cấp công cụ ghi biên bản cuộc họp.
- Cung cấp dụng cụ phản hồi và đánh giá (cho sinh viên và giảng viên).
- Phân công và gợi ý Tutor (cho sinh viên).
- Cung cấp tiến độ của sinh viên (% bài làm, điểm, điểm danh — để tutor theo dõi và ghi nhận).
- Cung cấp công cụ phân tích và thống kê dựa trên kết quả người học (hỗ trợ PDT xét học bổng/điểm rèn luyện).
- Hỗ trợ các định dạng tệp phổ biến: .doc, .docx, .pdf.

3 Non-Fucntional Requirements

3.1 Product Requirements (Yêu cầu Sản phẩm)

- Thông báo thay đổi/hủy lịch (Tutor): Sau khi ghi nhận tutor đổi/hủy lịch buổi học, hệ thống phải đảm bảo gửi thông báo đến email sinh viên trong vòng 5 phút.
- Thông báo đặt/đổi/hủy tham gia (Sinh viên): Sau khi ghi nhận sinh viên đặt/đổi/hủy tham gia buổi học, hệ thống gửi thông báo về email của sinh viên trong vòng 5 phút.
- Email nhắc nhở sự kiện: Email nhắc nhở sự kiện phải được gửi chính xác trong khoảng thời gian \pm 5 phút so với mốc 60 phút trước giờ sự kiện bắt đầu.



- Thời gian tải trang (Load Time): Thời gian tải (load time) của các trang chính (Dashboard, Đặt lịch, Hồ sơ) không được vượt quá 3 giây khi có đồng thời 500 người dùng đang hoạt động.
- Đồng bộ dữ liệu HCMUT _ DATACORE: Mỗi lần ghi nhận dữ liệu thay đổi trên HCMUT _ DATACORE hệ thống phải tự động cập nhật thay đổi trong vòng 2 giờ; các thay đổi được hiển thị cho người dùng ở mỗi phiên đăng nhập mới của người dùng.
- Thời gian hoạt động (Uptime): Hệ thống phải hoạt động tất cả các ngày trong tháng, uptime tối thiểu 99.5% mỗi tháng (thời gian không hoạt động tương đương 3.5 giờ cho 1 tháng 30 ngày); thời gian bảo trì không được xếp vào khoảng thời gian cao điểm 7h00-19h00.
- Tự động gửi lại Email/Thông báo lỗi: Các email và thông báo nếu gửi lỗi sẽ được hệ thống tự động gửi lại tối đa 5 lần trong 5 phút.
- Chức năng trên Mobile: Hệ thống trên mobile đảm bảo thực hiện 90% chức năng so với hệ thống dùng trên máy tính; các chức năng quan trọng: gửi yêu cầu hỗ trợ, chọn tutor, quản lý lịch, nộp/ghi báo cáo phải thực hiện được trên mobile.

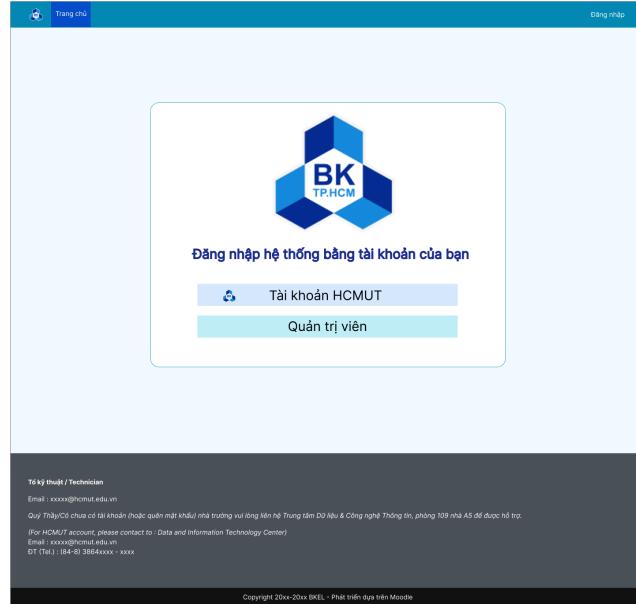
3.2 Organisational Requirements (Yêu cầu Tổ chức)

- Cơ chế đăng nhập: Tất cả người dùng hệ thống (sinh viên, giảng viên, cán bộ) sử dụng BKNetID và mật khẩu để đăng nhập vào hệ thống.
- Chính sách khóa tài khoản: Hệ thống phải tự động khóa tạm thời tài khoản trong 10 phút nếu có 5 lần đăng nhập thất bại liên tiếp trong vòng 5 phút.
- Đào tạo và sử dụng: Tất cả người dùng được kỳ vọng thông thạo cách sử dụng hệ thống sau tối đa 2 giờ training.

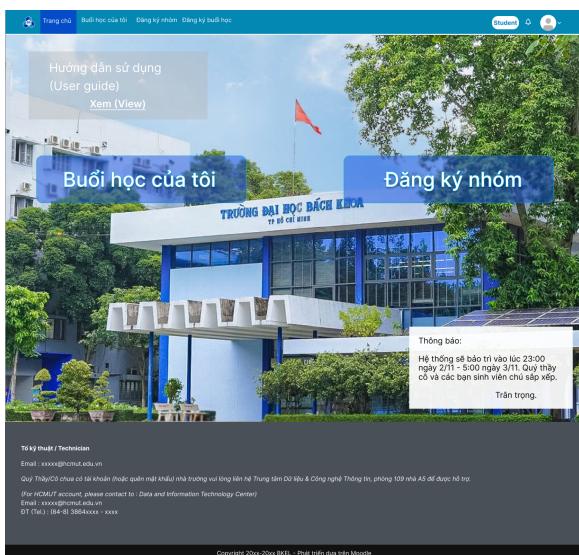
4 System Modelling

4.1 UI design Mockup

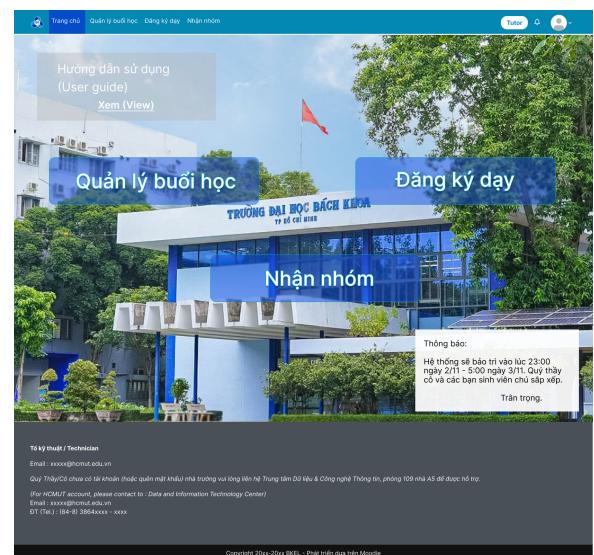
4.1.1 Trang chủ và đăng nhập



(a) Trang đăng nhập



(b) Trang chủ sinh viên



(c) Trang chủ giảng viên

Hình 4.1: Giao diện trang chủ và đăng nhập: một ảnh ở trên, hai ảnh ở dưới.

4.1.2 Sinh viên đăng ký nhóm và tutor nhận/từ chối yêu cầu

(a) Tutor tiếp nhận nhóm hướng dẫn

(b) Sinh viên đăng ký nhóm học

Hình 4.2: Các giao diện của chức năng đăng ký nhóm học.

4.1.3 Thông tin nhóm và thao tác đăng ký nhóm mới

(a) Thông tin chi tiết nhóm

(b) Form đăng ký mới

(c) Danh sách tutor phù hợp

Hình 4.3: Các bước thao tác trong quá trình đăng ký nhóm học.



4.1.4 Thông tin giảng viên và môn học

Thông tin tutor ×

Họ và tên: Nguyễn Văn H Tuổi: 32

Học vị: Tiến sĩ

Email: h.nguyenvan@hcmut.edu.vn

Khoa: Khoa học và kỹ thuật máy tính

Chuyên ngành: Khoa học máy tính

Vai trò: Trưởng khoa

[Danh sách môn đăng ký dạy](#)

Danh sách môn đăng ký dạy ×

Cấu trúc dữ liệu và giải thuật (CO2015)	04-08-2025 19:05
Kỹ thuật lập trình (CO1007)	04-08-2025 16:25
Hệ điều hành (CO3027)	03-08-2025 14:25
Học máy (CO3045)	02-08-2025 20:30

Tim Mới nhất ▾

(a) Thông tin chi tiết giảng viên

(b) Danh sách môn tutor đăng ký dạy

Hình 4.4: Các giao diện thông tin của giảng viên.

4.1.5 Tutor quản lý buổi học

Đăng ký buổi dạy mới

Các chủ đề giảng dạy

Kiến trúc Máy Tính

Kiến trúc bộ xử lý song song

Kỹ thuật lập trình

OOP nâng cao

Cấu trúc dữ liệu và giải thuật

Cấu trúc cây nâng cao

Cấu trúc rời rạc cho khoa học máy tính

Ứng dụng toán rời rạc

Hệ điều hành

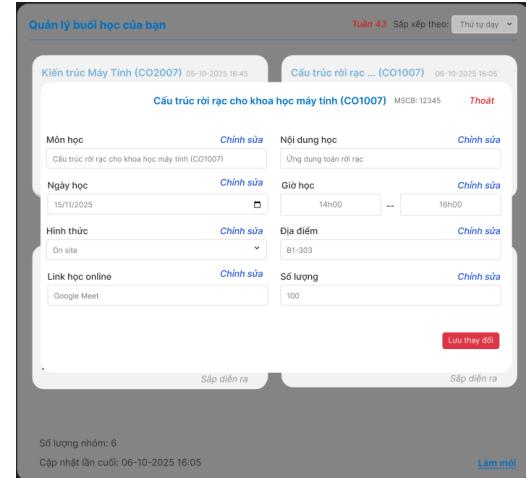
CO2007

Công nghệ phần mềm

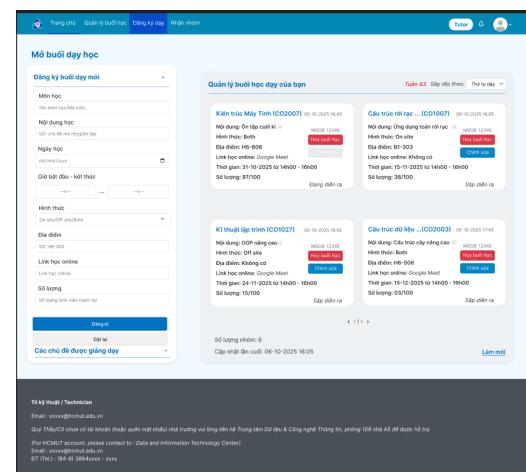
CO1027

Hình 4.5: Đăng ký mở buổi học

Hình 4.6: Giao diện quản lý buổi học của giảng viên.



Hình 4.6: Chính sửa thông tin buổi học



Hình 4.7: Giao diện chính quản lý buổi học



**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**

4.1.6 Sinh viên đăng ký buổi học

The screenshot shows a list of courses under the heading "Tham dự lớp học". It includes course names like "Kiến trúc Máy Tính (CO2007)", "Ki thuật lập trình(CO1027)", and "Cấu trúc dữ liệu và giải thuật (CO2003)". Each course entry provides details such as tutor, meeting type, location, time, and registration status.

(a) Trang danh sách lớp học

This screenshot displays the detailed information for the "Kiến trúc Máy Tính (CO2007)" course. It shows the course name, code, tutor, meeting type, location, time, and registration status. Below this, there are sections for "Đánh giá" (Reviews), "Lịch trình" (Schedule), and "Tài liệu" (Materials).

(b) Chi tiết thông tin lớp học

Hình 4.9: Giao diện chức năng đăng ký buổi học của sinh viên.

4.1.7 Sinh viên truy cập buổi học

The screenshot shows a list of courses under the heading "Danh sách buổi học". It includes course names like "Kiến trúc Máy Tính (CO2007)", "Cấu trúc dữ liệu và giải thuật (CO2003)", and "Ki thuật lập trình(CO1027)". Each course entry provides details such as tutor, meeting type, location, time, and registration status.

(a) Danh sách buổi học

This screenshot displays the detailed information for the "Kiến trúc Máy Tính (CO2007)" course. It shows the course name, code, tutor, meeting type, location, time, and registration status. Below this, there are sections for "Đánh giá" (Reviews), "Lịch trình" (Schedule), and "Tài liệu" (Materials).

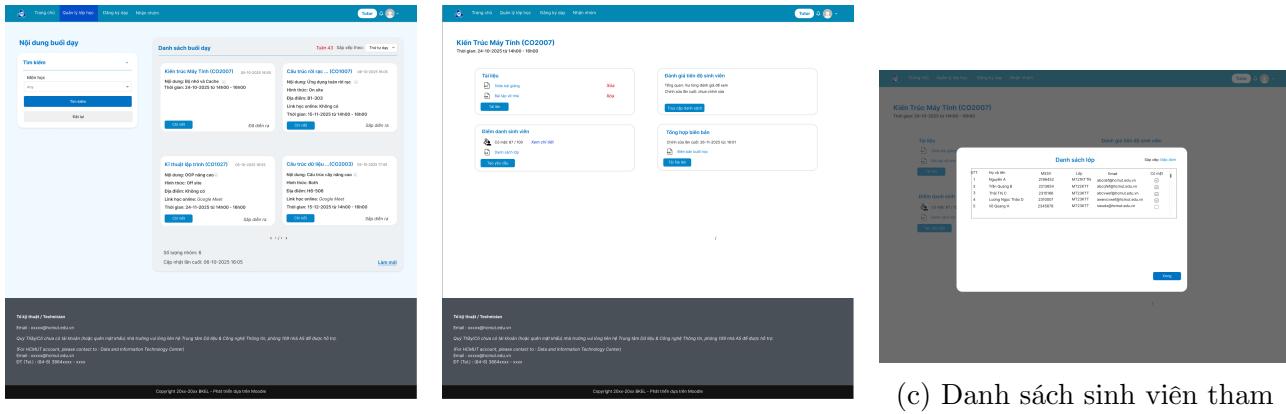
(b) Chi tiết buổi học

This screenshot shows the feedback section for the "Kiến trúc Máy Tính (CO2007)" course. It includes fields for "Đánh giá buổi học" (Evaluate class), "Đánh giá bài giảng" (Evaluate lecture), and "Phản hồi chất lượng buổi học" (Feedback on class quality). A large text area is provided for general comments.

(c) Phản hồi của sinh viên

Hình 4.10: Các giao diện chức năng của sinh viên.

4.1.8 Tutor truy cập buổi học



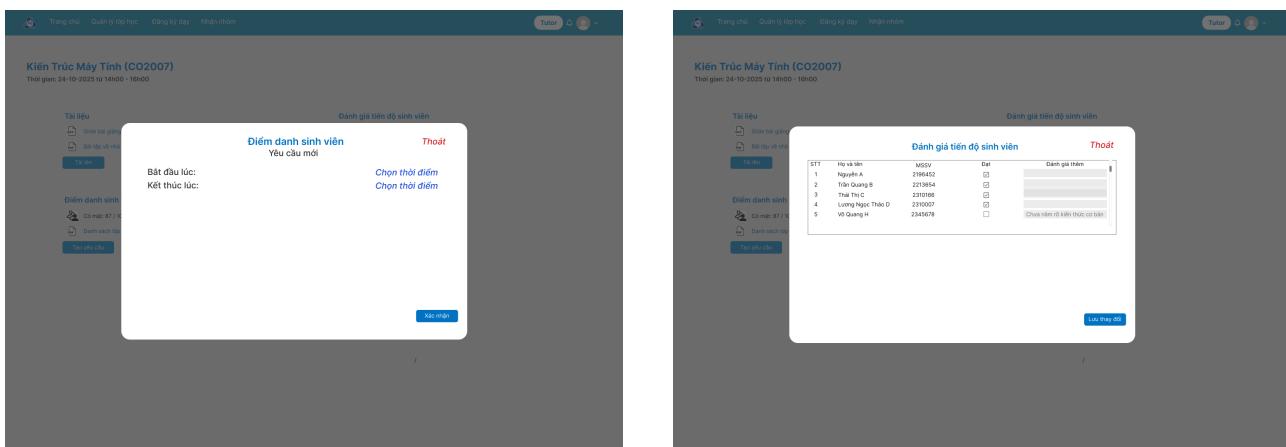
(a) Danh sách buổi học

(b) Chi tiết buổi học

(c) Danh sách sinh viên tham gia

Hình 4.11: Các giao diện quản lý buổi học của Tutor.

4.1.9 Tutor tổ chức và đánh giá buổi học

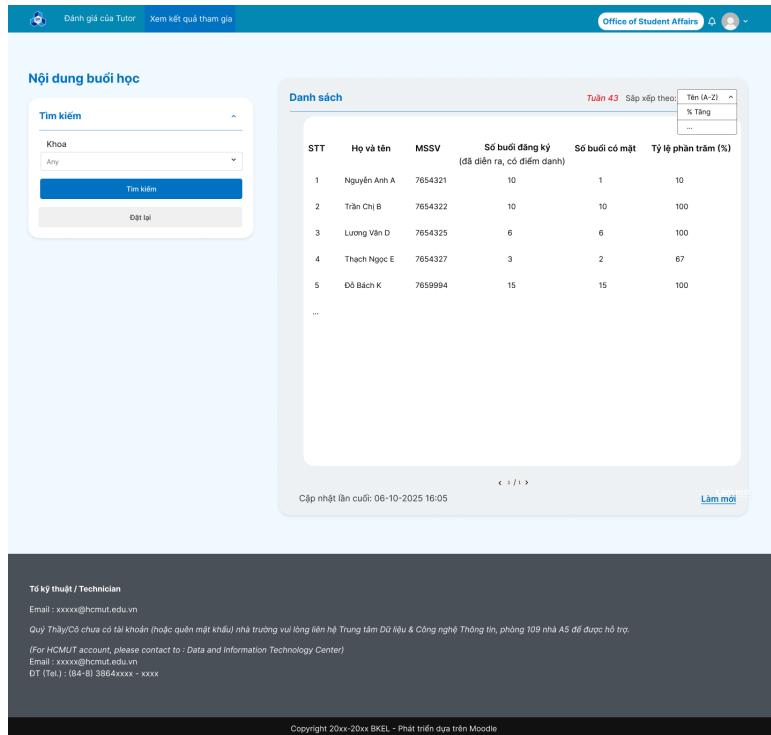


(a) Chi tiết điểm danh buổi học

(b) Chi tiết đánh giá buổi học

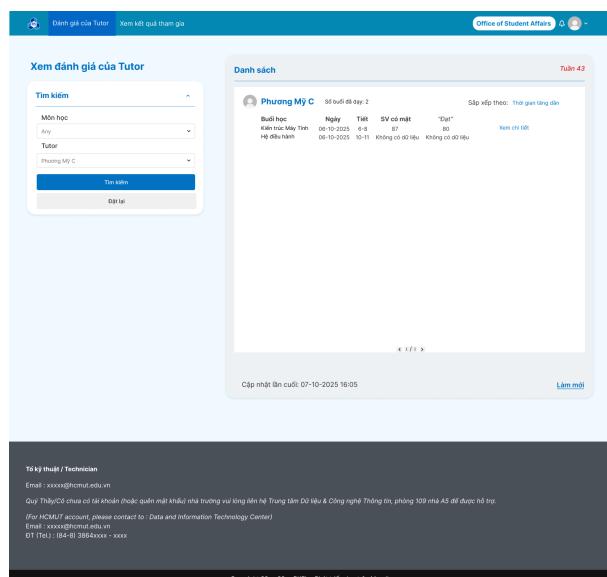
Hình 4.12: Các giao diện tổ chức và đánh giá buổi học.

4.1.10 Phòng Công tác Sinh viên



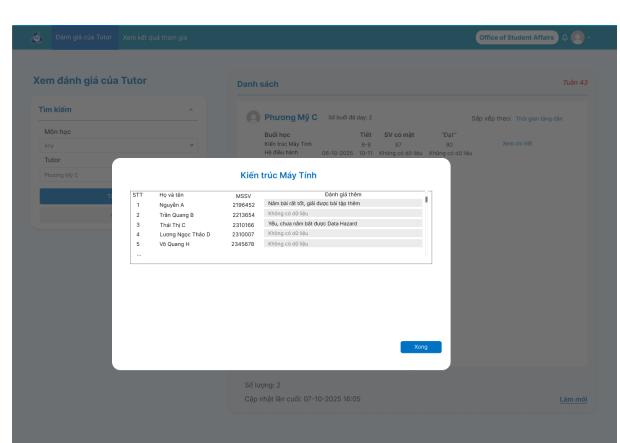
STT	Họ và tên	MSSV	Số buổi đăng ký (đã diễn ra, có điểm danh)	Số buổi có mặt	Tỷ lệ phản trả (%)
1	Nguyễn Anh A	7654321	10	1	10
2	Trần Chí B	7654322	10	10	100
3	Lương Văn D	7654325	6	6	100
4	Thạch Ngọc E	7654327	3	2	67
5	Đỗ Bá Cảnh K	7659994	15	15	100
...					

(a) Kết quả tham gia của sinh viên



Môn học	Ngày	Tiết	SV có mặt	Đđt	Năm chí tiết
Phương Mỹ C	06-10-2023	8-9	87	80	
	06-10-2023	10-11	Không có số liệu	Không có số liệu	

(b) Các nhóm của tutor

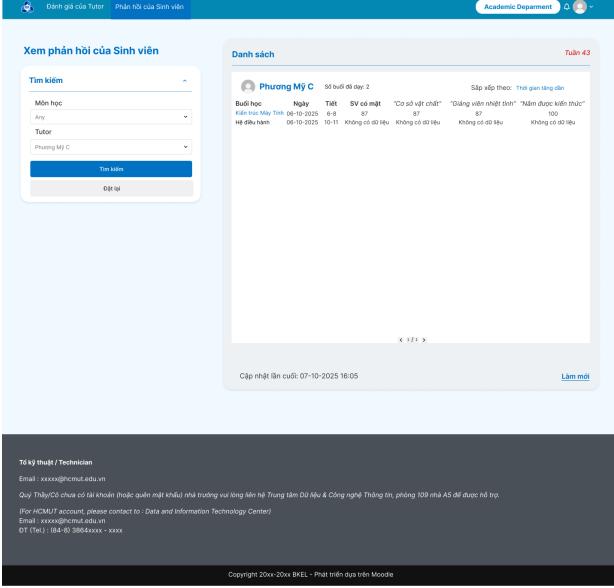


Số	Họ và tên	MSSV	Đđt	Năm chí tiết	
1	Nguyễn A	7654321	100	Năm chí tiết	
2	Trần Chí B	7654322	100	Năm chí tiết	
3	Thái Thị C	2310166	98	Năm chí tiết	
4	Lương Ngọc Thảo D	2300007	98	Năm chí tiết	
5	Võ Duy H	2300070	98	Năm chí tiết	
...					

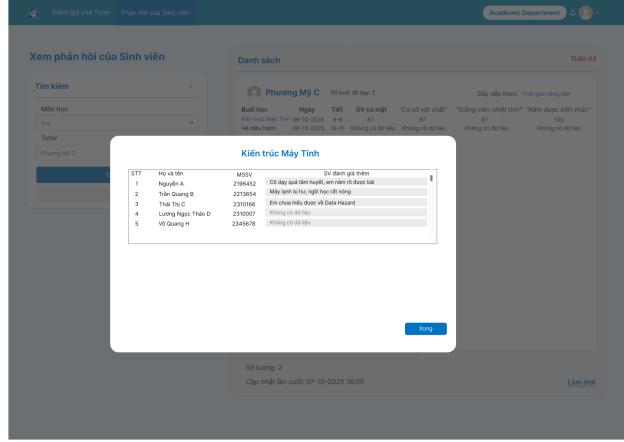
(c) Tutor đánh giá sinh viên

Hình 4.13: Phòng CTSV xem báo cáo.

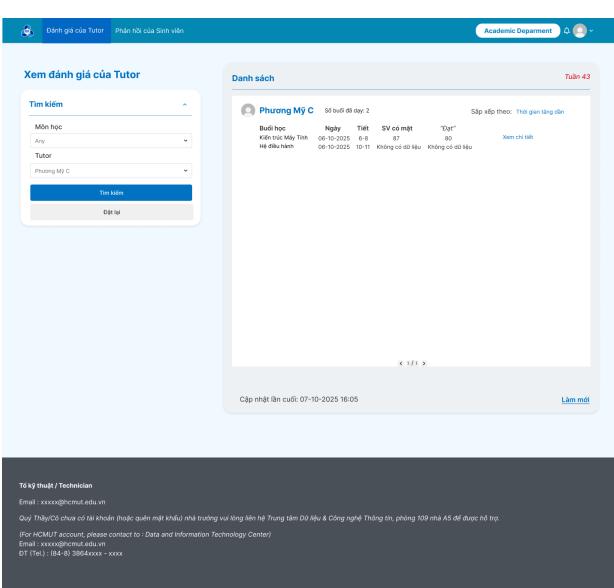
4.1.11 Khoa/Bộ môn



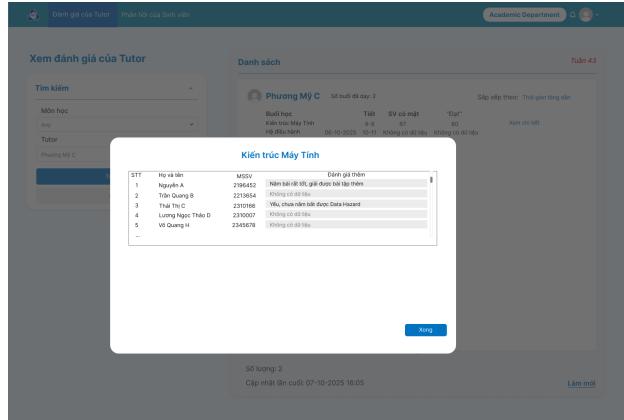
(a) Phản hồi của sinh viên (1)



(b) Phản hồi của sinh viên (2)



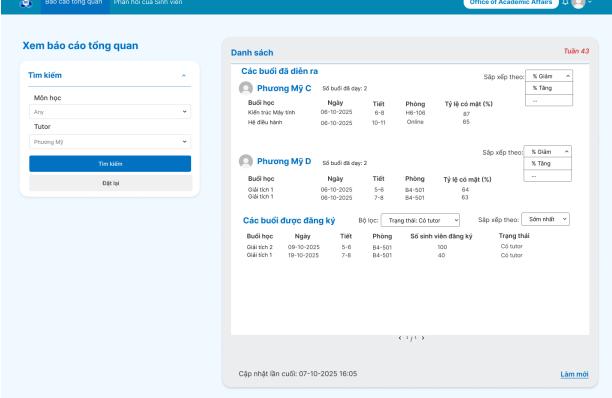
(c) Các nhóm của tutor



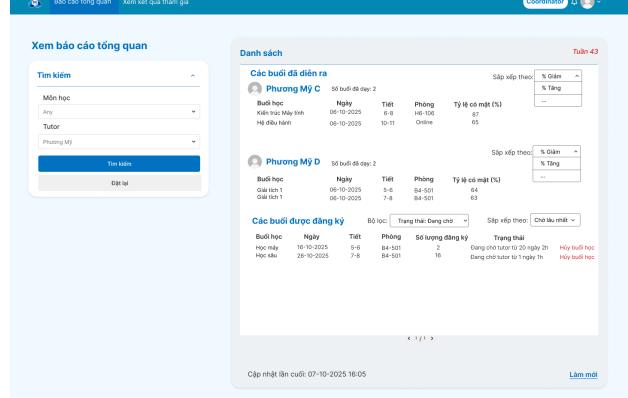
(d) Tutor đánh giá sinh viên

Hình 4.14: Khoa/Bộ môn xem báo cáo.

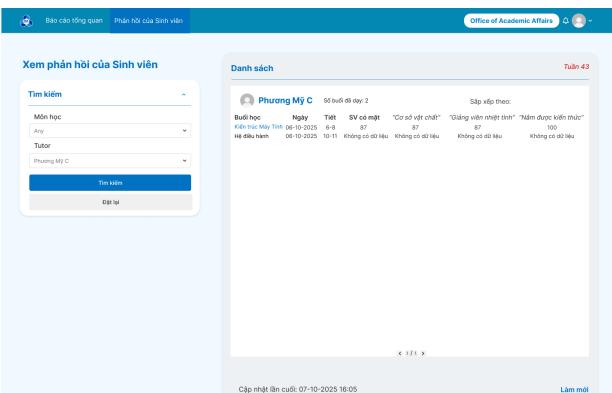
4.1.12 Phòng Đào tạo & Điều phối viên



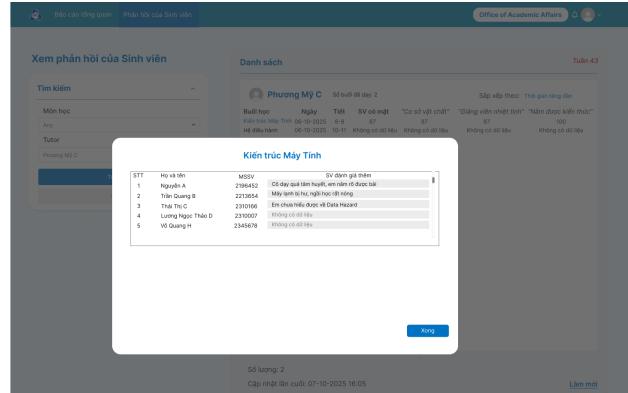
(a) Báo cáo tổng quan (PDT)



(b) Báo cáo tổng quan (Diều phối viên)



(c) Phản hồi sinh viên (1)



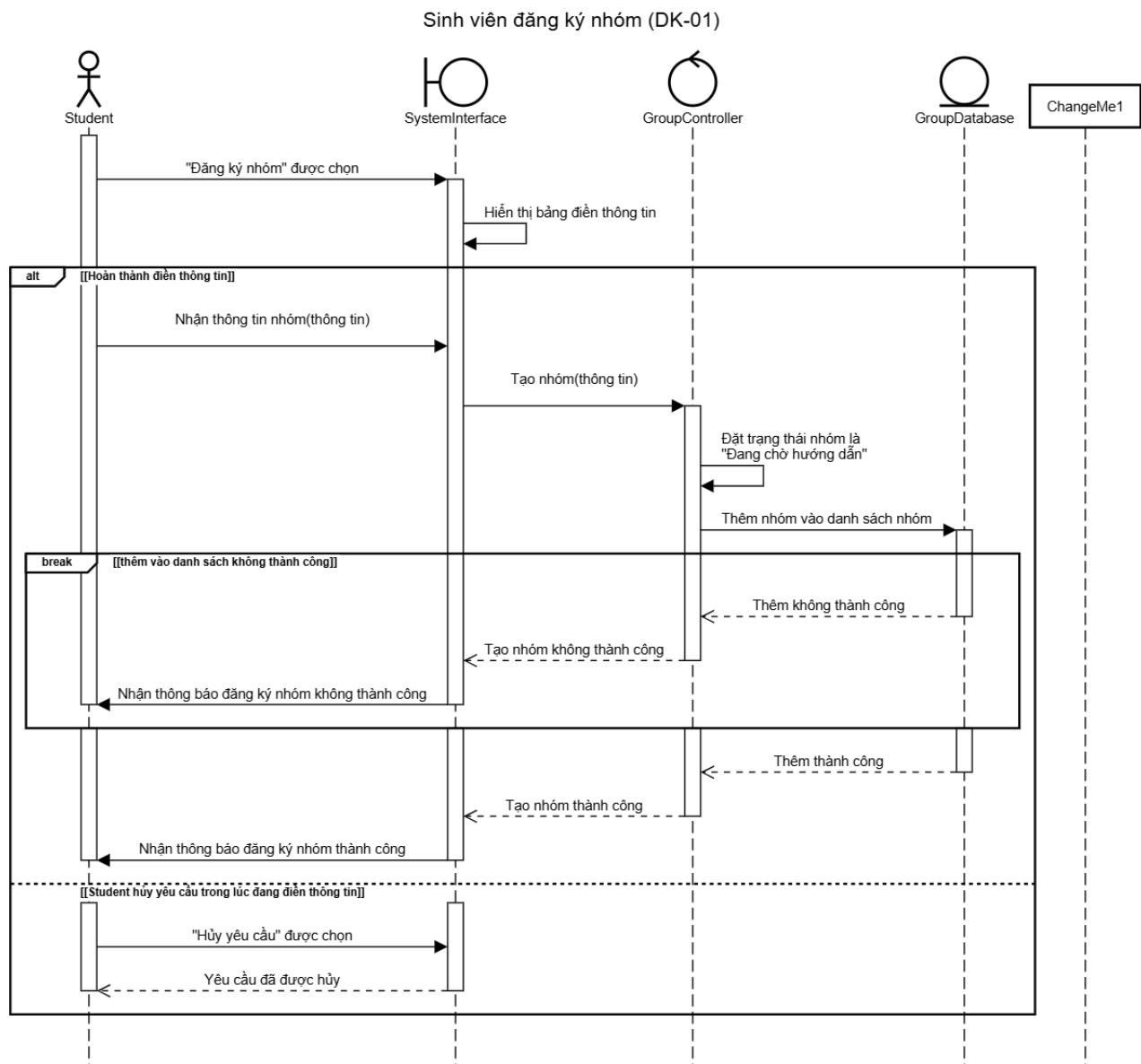
(d) Phản hồi sinh viên (2)

Hình 4.15: Chức năng xem báo cáo và phản hồi của PDT & Diều phối viên.



4.2 Sequence diagrams

4.2.1 Sinh viên đăng ký nhóm:



Hình 4.16: Sequence diagram: Sinh viên đăng ký nhóm

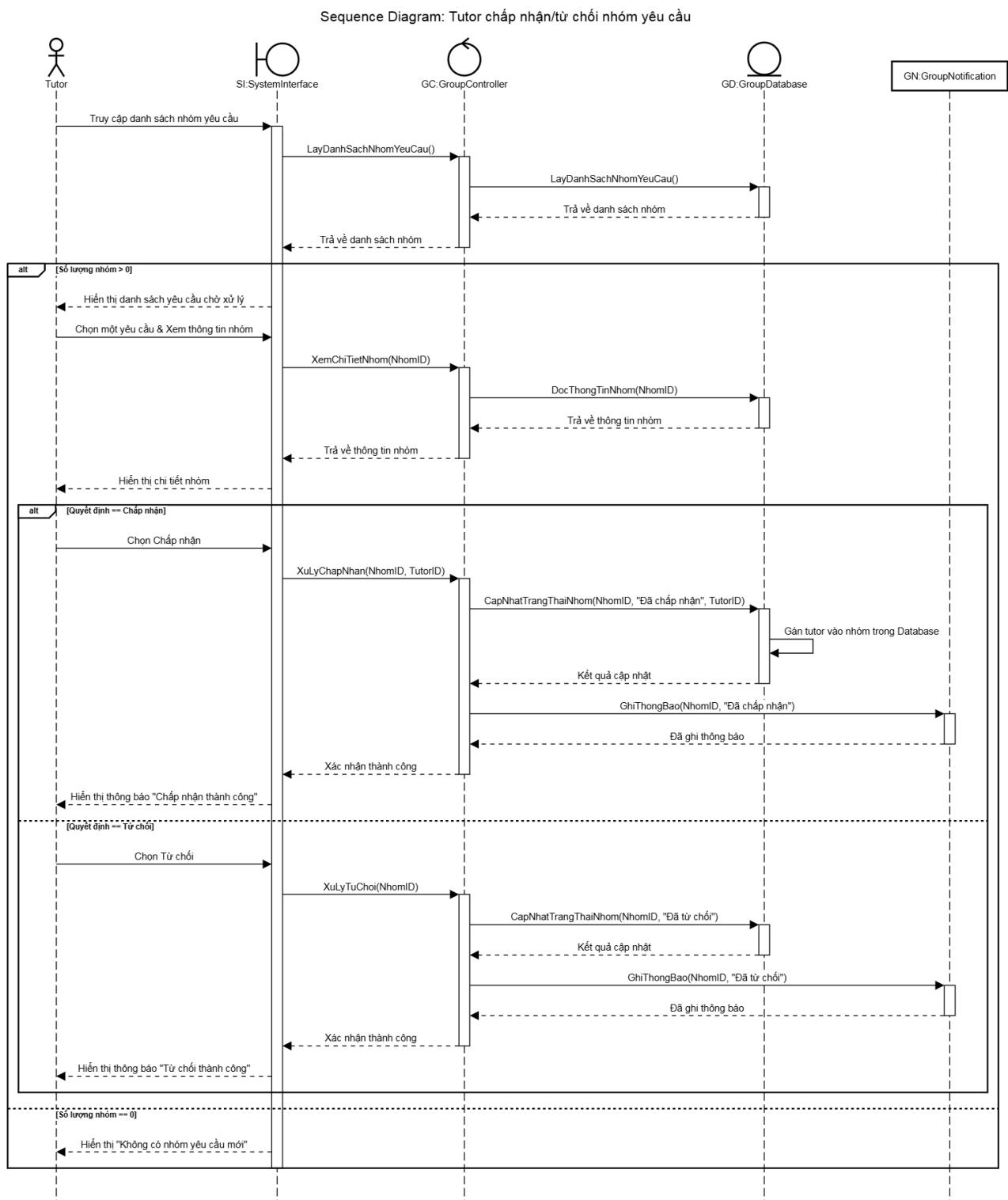


Mô tả sequence diagram: Sinh viên đăng ký nhóm mới

Diagram gồm 4 lifelines bao gồm Actor là Sinh viên, Boundary là SystemInterface (SI) đóng gói lại hệ thống và cung cấp phương thức cho người dùng tương tác, với bộ phận Control điều khiển logic cho quá trình đăng ký nhóm là GroupController (GC), cuối cùng là hệ thống dữ liệu Entity là GroupDatabase (GD) chịu trách nhiệm lưu trữ dữ liệu của các nhóm được tạo. Khi Sinh viên nhấn nút đăng ký nhóm trên giao diện, SI sẽ nhận tín hiệu và hiển thị bảng để sinh viên điền thông tin. Trường hợp đầu tiên, sinh viên nhấn gửi thông tin, SI sẽ nhận thông tin đó và yêu cầu GC tạo một đối tượng là nhóm với các thuộc tính được người dùng điền vào và đặt trạng thái cho nhóm vừa tạo là “đang chờ hướng dẫn” sau đó yêu cầu lưu nhóm vừa tạo vào GB. GB nhận yêu cầu và thực hiện lưu nhóm vừa tạo vào cơ sở dữ liệu, nếu xảy ra vấn đề dẫn đến không lưu được, GB sẽ phản hồi lại thông tin cho GC, GC báo cho SI, và SI hiển thị cho người dùng biết là tạo nhóm không thành công. Trường hợp thứ hai là khi Sinh viên đang điền thông tin mà bấm hủy thì SI sẽ nhận yêu cầu hủy và chấm dứt hành động.



4.2.2 Tutor chấp nhận/từ chối nhóm:



Hình 4.17: Sequence diagram: Tutor chấp nhận/từ chối nhóm



Mô tả sequence diagram: Tutor chấp nhận/từ chối nhóm

Diagram gồm 5 lifelines bao gồm *Actor* là Tutor, *Boundary* là *SystemInterface* (SI) đóng gói lại hệ thống và cung cấp phương thức cho người dùng tương tác, với bộ phận *Control* điều khiển logic cho quá trình là *GroupController* (GC), và hệ thống dữ liệu *Entity* là *GroupDatabase* (GD) chịu trách nhiệm lưu trữ dữ liệu nhóm.

Khi Tutor truy cập vào danh sách nhóm yêu cầu, SI sẽ gửi yêu cầu đến GC để lấy danh sách. GC sẽ yêu cầu GD trả về danh sách các nhóm có trạng thái “chờ xử lý”. Kết quả trả về sẽ có hai trường hợp:

- **Trường hợp 1:** Số lượng nhóm = 0, SI sẽ hiển thị thông báo “Không có nhóm yêu cầu mới”.
- **Trường hợp 2:** Số lượng nhóm > 0, SI sẽ hiển thị danh sách các yêu cầu chờ xử lý. Tutor có thể chọn một yêu cầu để xem chi tiết thông tin nhóm. Khi đó, SI sẽ yêu cầu GC lấy thông tin chi tiết của nhóm đó từ GD và hiển thị lên cho Tutor.

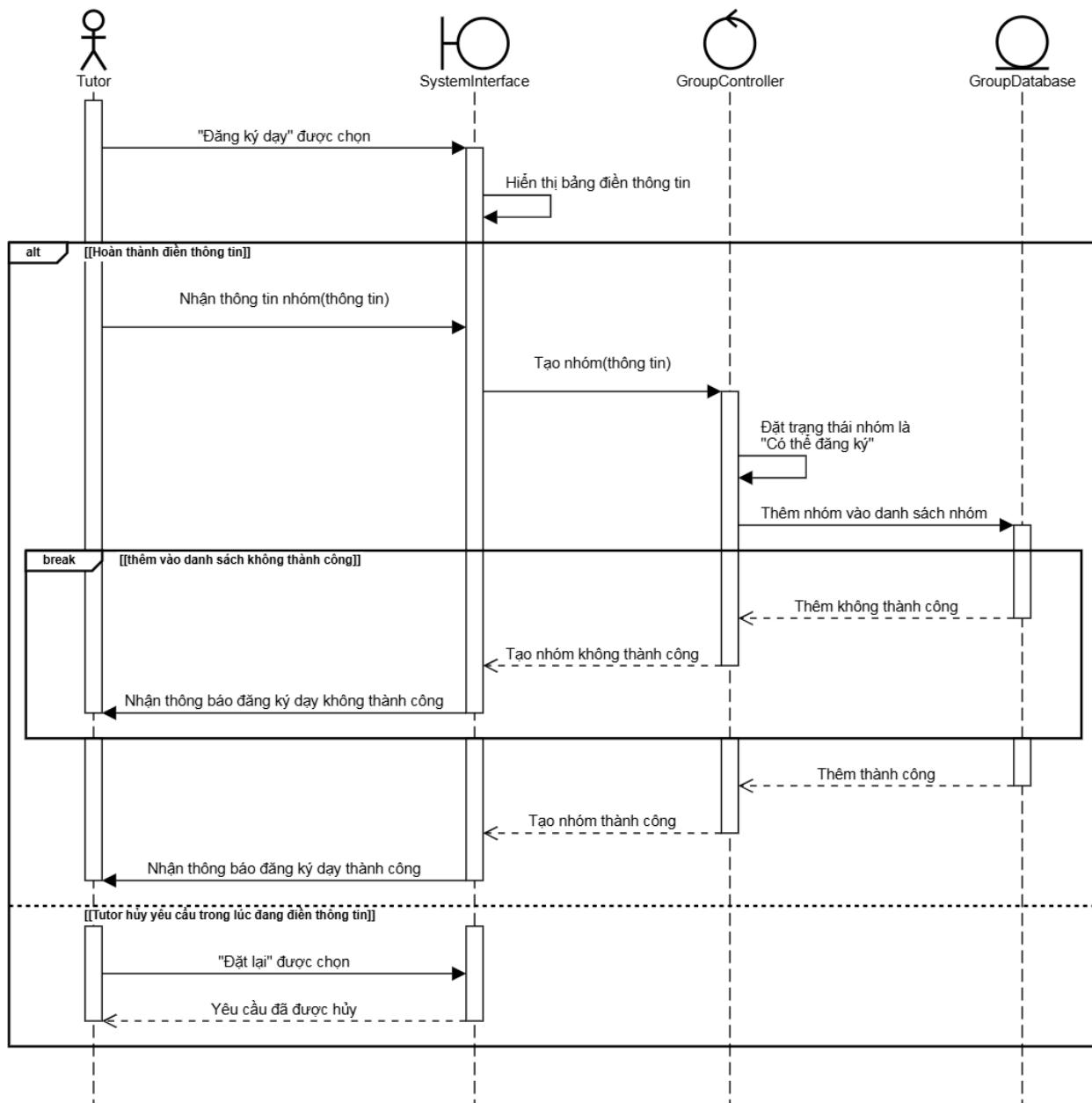
Sau khi xem xét, Tutor có hai lựa chọn:

- **Chấp nhận:** Tutor chọn “Chấp nhận”. SI sẽ gửi yêu cầu xử lý chấp nhận đến GC. GC sẽ cập nhật trạng thái của nhóm trong GD (thành “Đã chấp nhận”). Khi các thao tác này hoàn tất, SI sẽ hiển thị thông báo “Chấp nhận thành công” cho Tutor.
- **Từ chối:** Tutor chọn “Từ chối”. SI sẽ gửi yêu cầu xử lý từ chối đến GC. GC sẽ cập nhật trạng thái nhóm trong GD (thành “Đã từ chối”). Khi hoàn tất, SI sẽ hiển thị thông báo “Từ chối thành công” cho Tutor.



4.2.3 Tutor mở buổi học:

Sequence Diagram: Tutor đăng ký dạy (DK-02)



Hình 4.18: Sequence diagram: Tutor mở buổi học



Mô tả sequence diagram: Tutor mở buổi học

Diagram gồm bốn lifelines: Actor là Tutor; Boundary là SystemInterface (SI) chịu trách nhiệm tiếp nhận thao tác từ tutor và hiển thị giao diện; Control là GroupController (GC) điều khiển toàn bộ logic khi tạo nhóm dạy; cuối cùng là Entity-GroupDatabase (GD) đảm nhiệm việc lưu trữ dữ liệu nhóm vào cơ sở dữ liệu.

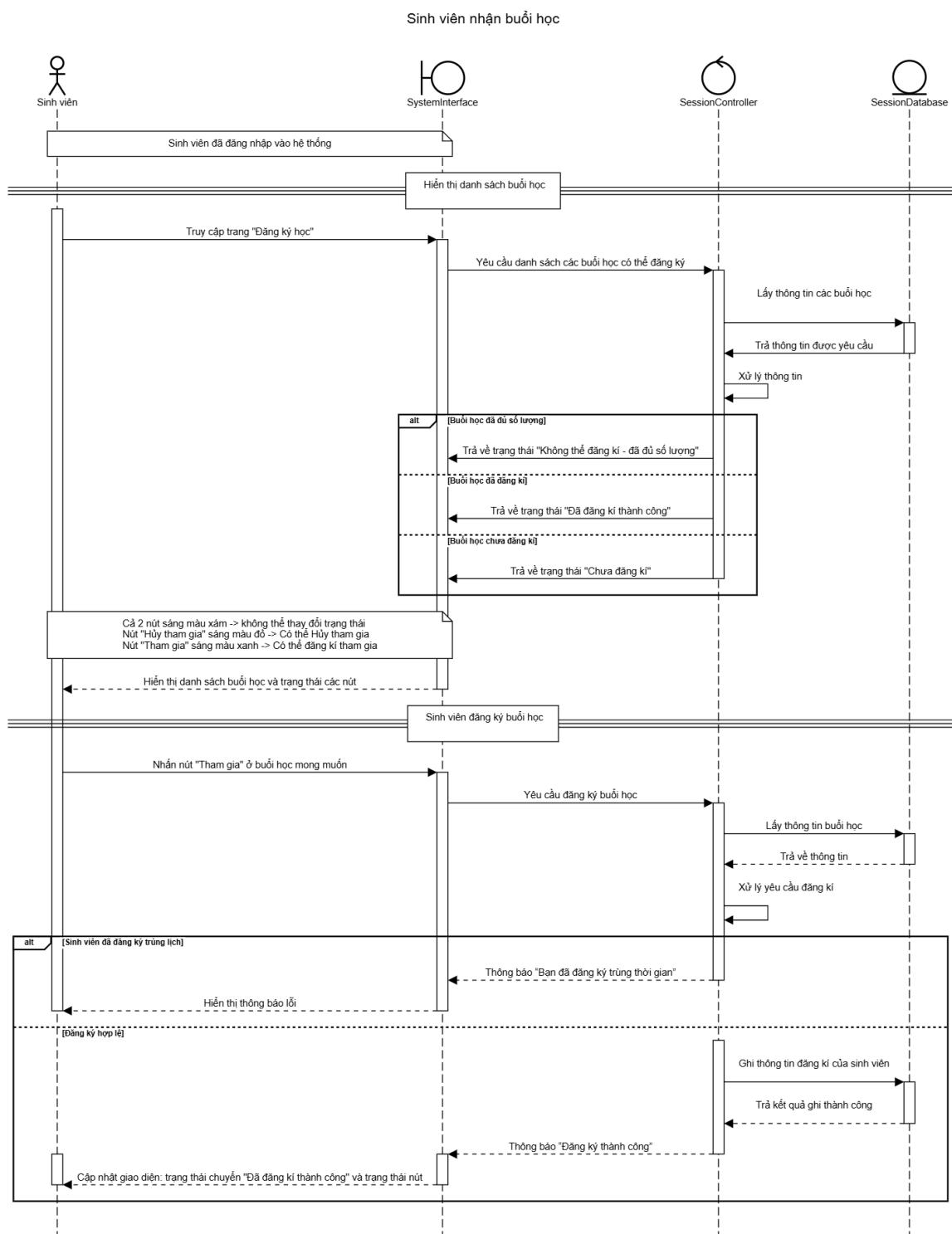
Khi Tutor chọn chức năng “Đăng ký dạy”, SI nhận tín hiệu và hiển thị bảng cho tutor nhập thông tin. Nếu tutor nhấn “Đăng ký”, SI tiếp nhận dữ liệu và gửi thông tin sang GC, yêu cầu tạo một nhóm mới. GC tạo đối tượng nhóm dựa trên dữ liệu do tutor cung cấp, đặt trạng thái nhóm là “Có thể đăng ký”, sau đó yêu cầu GD thêm nhóm này vào danh sách nhóm trong cơ sở dữ liệu.

GD tiếp nhận yêu cầu lưu. Nếu xảy ra lỗi làm nhóm không được thêm vào danh sách, GD phản hồi về GC, GC gửi tín hiệu cho SI, và SI thông báo cho tutor rằng đăng ký dạy không thành công. Ngược lại, nếu việc lưu thành công, GC phản hồi ngược về SI và SI hiển thị thông báo đăng ký dạy thành công cho tutor.

Trong trường hợp tutor đang nhập thông tin nhưng chọn “Đặt lại”, SI sẽ nhận yêu cầu hủy thao tác và kết thúc quy trình mà không tạo nhóm.



4.2.4 Sinh viên tham gia buổi học:



Hình 4.19: Sequence diagram: Sinh viên tham gia buổi học



Mô tả sequence diagram: Sinh viên đăng ký tham gia buổi học

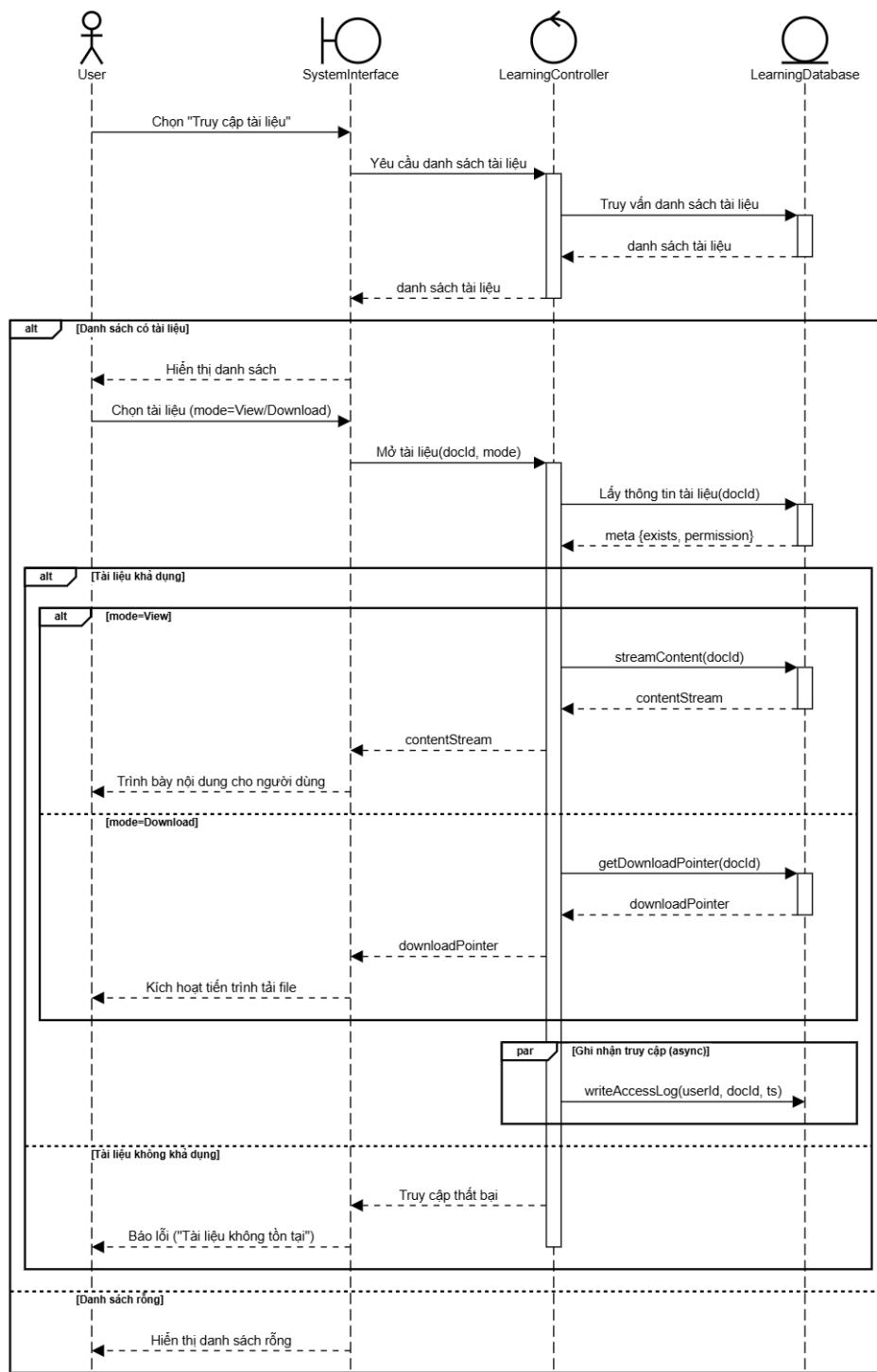
Sequence diagram gồm 4 lifelines: Actor Sinh viên, người trực tiếp thao tác nhận buổi học trên giao diện. Boundary SystemInterface (SI) chịu trách nhiệm hiển thị giao diện và cung cấp các chức năng cho sinh viên tương tác. Control SessionController (SC) đảm nhiệm xử lý logic nghiệp vụ liên quan đến việc lấy danh sách buổi học và đăng ký tham gia. Entity SessionDatabase (SD) thực hiện truy xuất và lưu trữ dữ liệu thông tin buổi học và thông tin đăng ký của sinh viên.

Sau khi sinh viên đăng nhập vào hệ thống, SI thực hiện hiển thị danh sách các buổi học. Khi sinh viên truy cập trang “Đăng ký học”, SI gửi yêu cầu đến SC để lấy danh sách các buổi học mà sinh viên có thể tham gia. SC tiếp tục yêu cầu SD truy xuất dữ liệu tất cả các buổi học hiện có. Sau khi nhận được dữ liệu phản hồi từ SD, SC xử lý thông tin và trả kết quả về cho SI. Tùy vào trạng thái buổi học thì SC sẽ gửi cho SI các thông báo tương ứng đồng thời SI cũng hiển thị danh sách buổi học và các nút tương ứng.

Khi sinh viên nhấn nút “tham gia” thì SI gửi yêu cầu đăng ký đến SC để SC tiến hành kiểm tra thông tin các buổi học từ SD. Tùy theo tình trạng mà SC xử lý (trùng lịch, đăng ký thành công) thì sẽ hiển thị thông báo tương ứng ở SI và nếu đăng ký thành công thì SD sẽ lưu thông tin vào hệ thống.



4.2.5 Sinh viên truy cập tài liệu:



Hình 4.20: Sequence diagram: Truy cập tài liệu



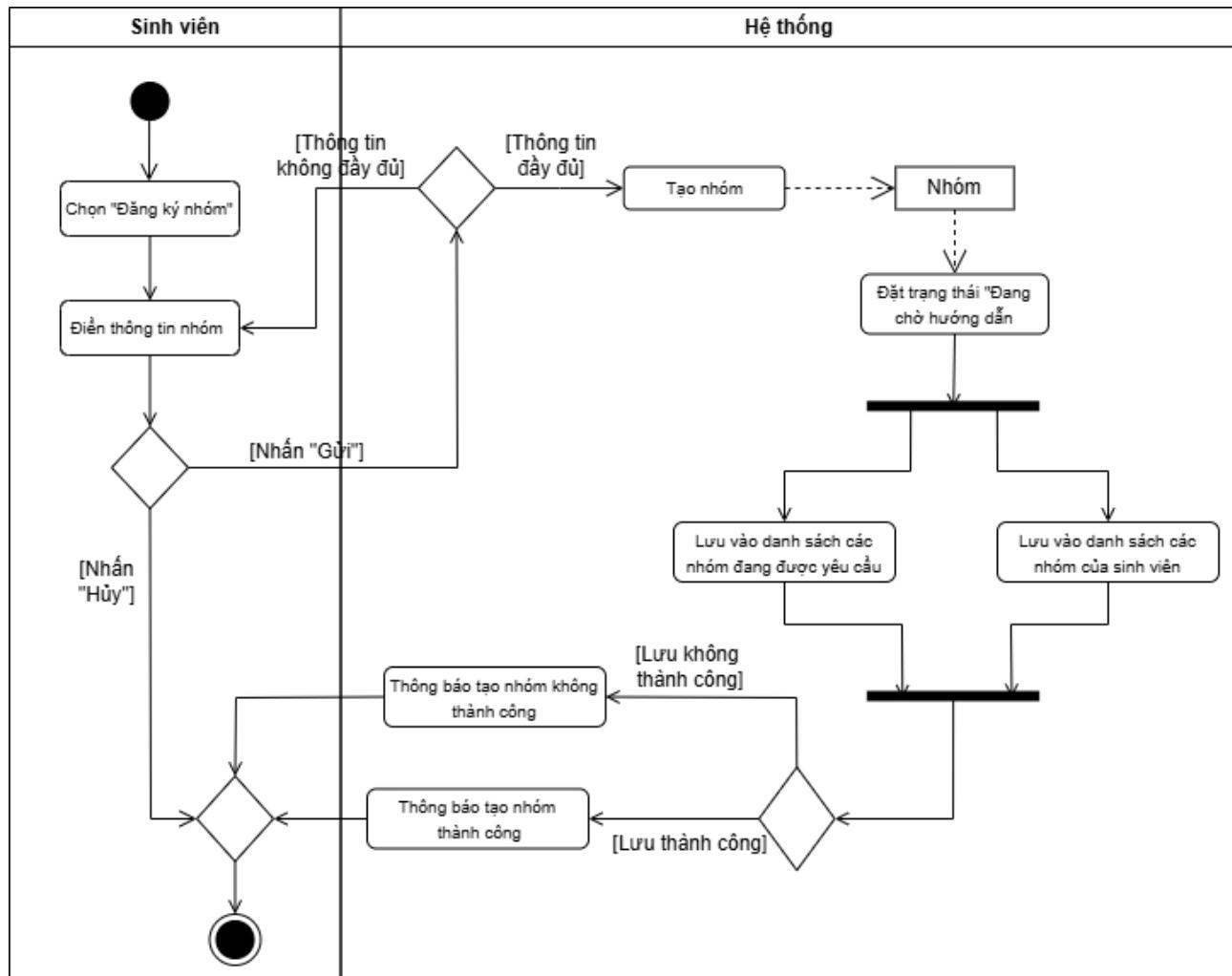
Mô tả sequence diagram: Sinh viên truy cập tài liệu

Sequence diagram “Truy cập tài liệu” mô tả chi tiết cách các thành phần trong hệ thống khi người dùng thực hiện truy cập tài liệu. Đầu tiên, User gửi yêu cầu truy cập bằng cách chọn chức năng “Truy cập tài liệu” trên giao diện. SystemInterface (SI) tiếp nhận và chuyển tiếp yêu cầu lấy danh sách tài liệu cho LearningController (LC). LC truy vấn LearningDatabase (DB), nhận về danh sách tài liệu rồi trả lại cho SI. Nếu danh sách không rỗng, SI hiển thị cho người dùng; nếu rỗng, hệ thống chỉ đơn giản thông báo danh sách trống.

Khi người dùng chọn một tài liệu cụ thể cùng chế độ thao tác (xem hoặc tải xuống), SI gửi yêu cầu mở tài liệu đến LC. LC tiếp tục truy vấn DB để lấy thông tin meta của tài liệu, bao gồm tồn tại và quyền truy cập. Nếu tài liệu khả dụng, với chế độ View, LC yêu cầu DB stream nội dung và chuyển luồng dữ liệu ngược lại cho người dùng xem trực tiếp. Với chế độ Download, LC lấy thông tin cho phép tải file và kích hoạt tiến trình tải xuống trên SI. Song song đó, LC ghi nhận log truy cập (user, tài liệu, thời điểm) xuống DB ở dạng bất đồng bộ. Ngược lại, nếu tài liệu không khả dụng, LC báo truy cập thất bại để SI hiển thị thông báo lỗi “Tài liệu không tồn tại” cho người dùng.

4.3 Activity diagrams

4.3.1 Sinh viên đăng ký nhóm:



Hình 4.21: Activity diagram: Sinh viên đăng ký nhóm

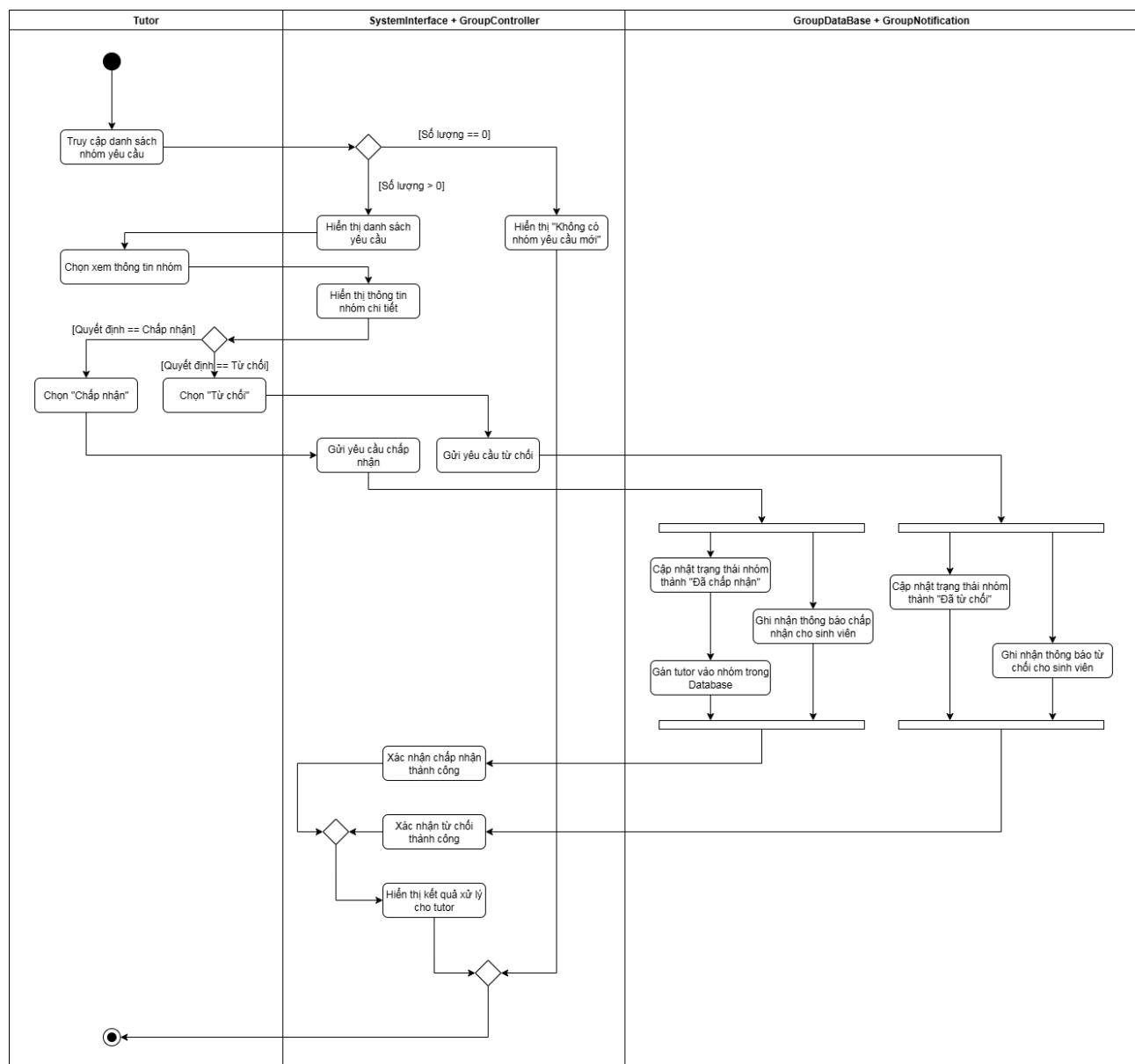
Mô tả Activity diagram: Sinh viên đăng ký nhóm

Diagram mô tả các hành động mà sinh viên và hệ thống thực hiện trong quá trình Sinh viên đăng ký nhóm mới. Để đăng ký nhóm, sinh viên sẽ chọn nút “Đăng ký nhóm” sau đó sinh viên điền thông tin, nếu đổi ý thông tin có thể nhấn “Hủy” và kết thúc hành động, hoặc nếu sinh viên quyết định sẽ tạo nhóm thì sẽ nhấn “Gửi”. Sau khi hệ thống nhận được thông tin, hệ thống sẽ kiểm tra nếu thông tin, nếu thông tin không đầy đủ, hệ thống sẽ yêu cầu sinh viên điền lại. Nếu thông tin đã đầy đủ, hệ thống thực hiện hành động và tạo ra đối tượng là “Nhóm” rồi đặt trạng thái cho nhóm vừa tạo là “Đang chờ hướng dẫn”. Sau khi tạo đối tượng “Nhóm”, hệ thống



tiếp tục lưu vào danh sách nhóm trong cơ sở dữ liệu. Lưu dữ liệu xong nếu thành công hoặc không thành công hệ thống đề thông báo cho người dùng tùy trường hợp rồi kết thúc hành động.

4.3.2 Tutor chấp nhận/từ chối nhóm yêu cầu:



Hình 4.22: Activity diagram: Tutor chấp nhận/từ chối nhóm yêu cầu

Mô tả Activity diagram: Tutor chấp nhận/từ chối nhóm yêu cầu:

Diagram mô tả các hành động mà Tutor và hệ thống thực hiện trong quá trình xử lý yêu cầu



đăng ký nhóm với 2 swimlane: Tutor và hệ thống. Để bắt đầu, Tutor sẽ chọn chức năng “Xem danh sách nhóm yêu cầu”. Hệ thống sau đó sẽ kiểm tra và lấy danh sách các nhóm đang chờ phê duyệt.

- Nếu không có nhóm nào (Số lượng = 0), hệ thống sẽ hiển thị thông báo “Không có nhóm yêu cầu mới” và kết thúc luồng xử lý.
- Nếu có ít nhất một nhóm (Số lượng > 0), hệ thống sẽ hiển thị danh sách các yêu cầu này. Tutor có thể chọn một yêu cầu để “Xem thông tin nhóm chi tiết”. Sau khi xem xét, Tutor sẽ đưa ra quyết định.

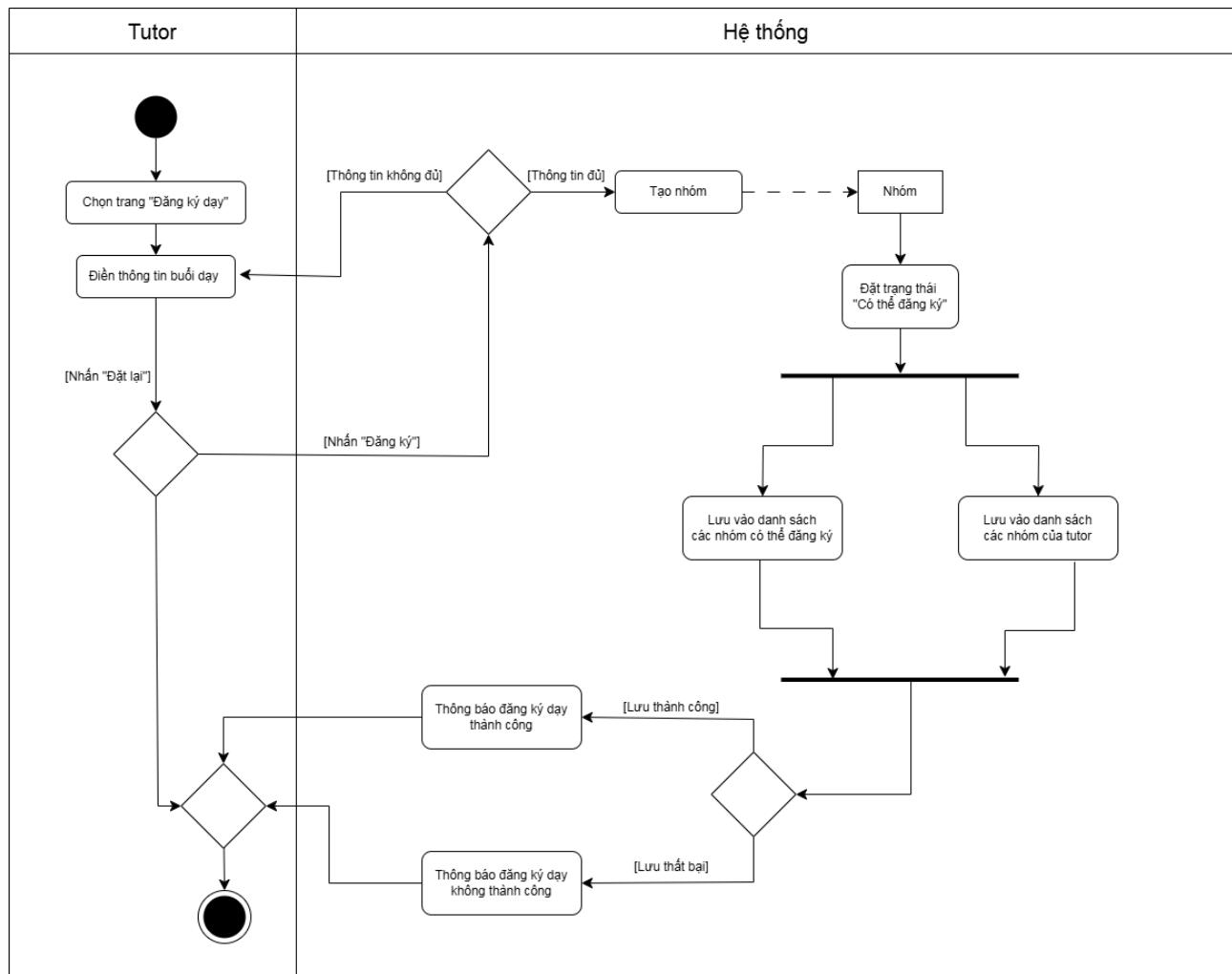
Tutor có hai lựa chọn:

- **Chấp nhận:** Tutor chọn “Chấp nhận”. Hệ thống sẽ thực hiện chuỗi hành động: cập nhật trạng thái nhóm thành “Đã chấp nhận”. Cuối cùng, hệ thống hiển thị kết quả “Xác nhận chấp nhận thành công” cho Tutor.
- **Từ chối:** Tutor chọn “Từ chối”. Hệ thống sẽ thực hiện chuỗi hành động: cập nhật trạng thái nhóm thành “Đã từ chối”. Cuối cùng, hệ thống hiển thị thông báo “Từ chối thành công” cho Tutor.

Sau khi mọi hành động được hoàn tất, quy trình sẽ kết thúc.



4.3.3 Tutor mở buổi học:



Hình 4.23: Activity diagram: Tutor mở buổi học

Mô tả Activity diagram: Tutor đăng ký buổi dạy mới

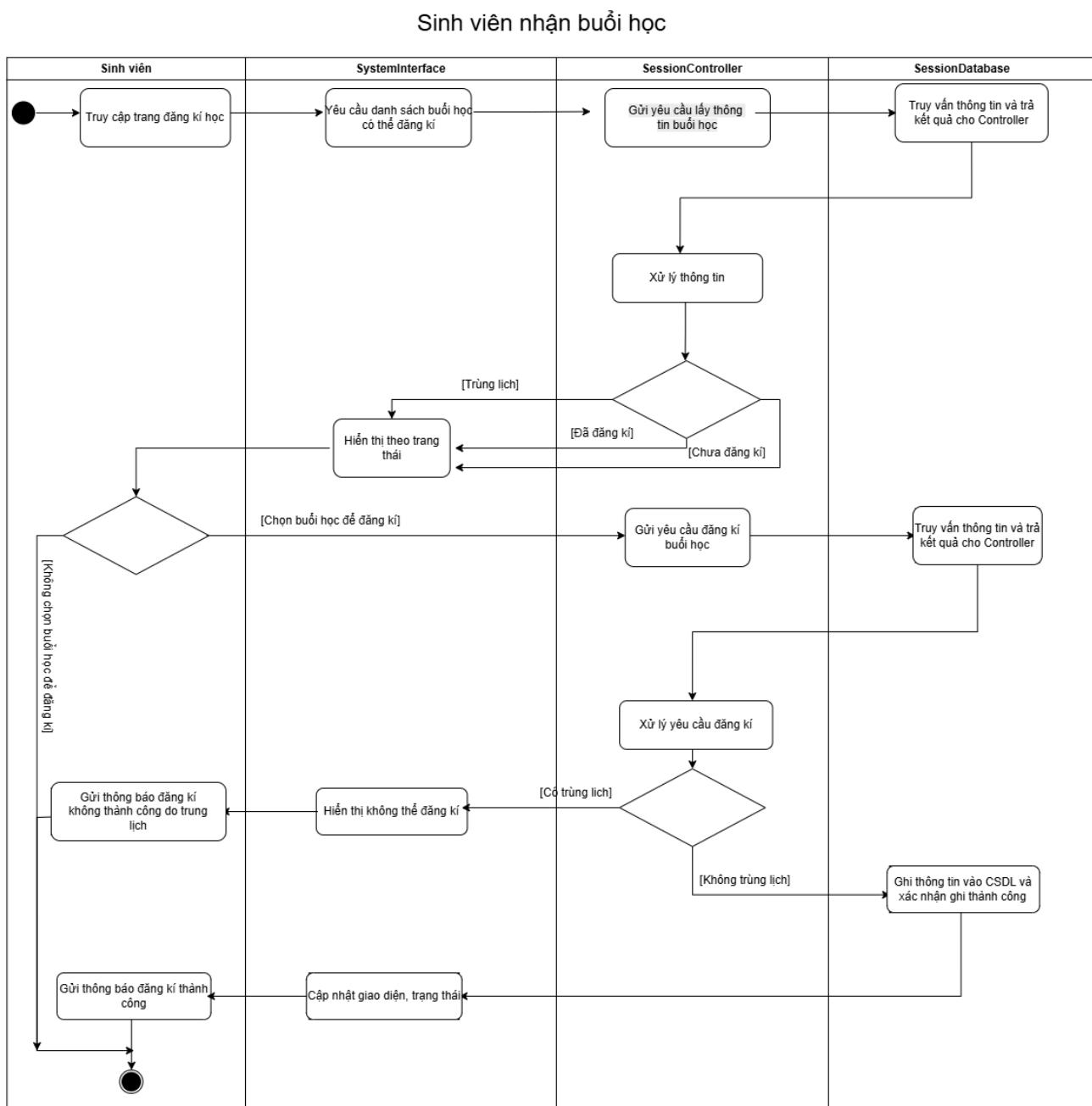
Tutor chọn chức năng “Đăng ký dạy” và nhập thông tin buổi dạy. Nếu muốn chỉnh lại dữ liệu, tutor có thể nhấn “Đặt lại” để quay về bước nhập thông tin. Nếu muốn gửi yêu cầu, tutor nhấn “Đăng ký”.

Hệ thống nhận thông tin và kiểm tra. Nếu thông tin chưa đầy đủ, hệ thống yêu cầu tutor điền lại. Nếu hợp lệ, hệ thống tạo đối tượng “Nhóm”, đặt trạng thái “Có thể đăng ký”, rồi lưu nhóm vào danh sách nhóm có thể đăng ký và danh sách nhóm của tutor.

Cuối cùng, hệ thống thông báo kết quả tùy theo việc lưu dữ liệu thành công hoặc thất bại, rồi kết thúc quy trình.



4.3.4 Sinh viên tham gia buổi học:



Hình 4.24: Activity diagram: Sinh viên tham gia buổi học

Mô tả Activity diagram: Sinh viên tham gia buổi học

Activity diagram mô tả chuỗi hoạt động giữa sinh viên và hệ thống trong quá trình sinh

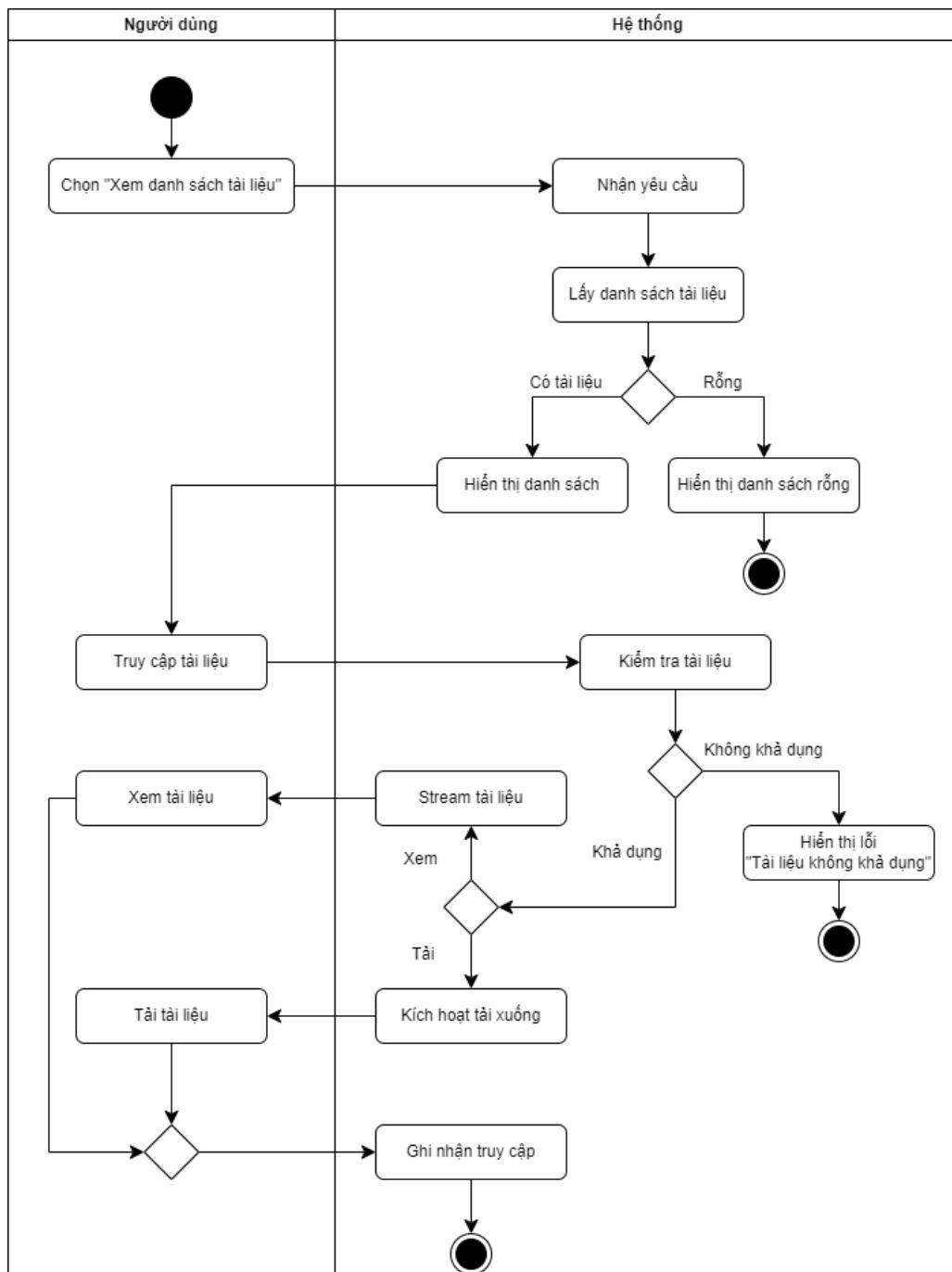


viên xem danh sách và đăng ký một buổi học. Ban đầu, sinh viên truy cập vào trang đăng ký học. Hệ thống tiếp nhận yêu cầu và tiến hành truy vấn cơ sở dữ liệu để lấy thông tin các buổi học hiện có, sau đó trả kết quả về cho giao diện để hiển thị theo từng trạng thái như: buổi học đã đăng ký, buổi học chưa đăng ký hoặc buổi học bị trùng lịch.

Khi đã xem danh sách, sinh viên có thể lựa chọn một buổi học để đăng ký. Nếu không chọn buổi học nào, tiến trình kết thúc ngay tại giao diện. Trường hợp sinh viên chọn buổi học để đăng ký, hệ thống sẽ gửi yêu cầu kiểm tra thông tin đăng ký. Nếu bị trùng lịch với buổi đăng kí trước đó thì hệ thống sẽ xử lý và trả về “không thể đăng kí do trùng lịch”, sau đó giao diện hiển thị lỗi và kết thúc hành động. Ngược lại nếu không trùng lịch thì hệ thống gửi yêu cầu đăng ký vào hệ cơ sở dữ liệu, sau khi lưu thành công, hệ thống phải hồi lại và giao diện cập nhập trạng thái buổi học và gửi thông báo “Đăng ký thành công” cho sinh viên.



4.3.5 Sinh viên truy cập tài liệu:



Hình 4.25: Activity diagram: Truy cập tài liệu



Mô tả Activity diagram: Truy cập tài liệu

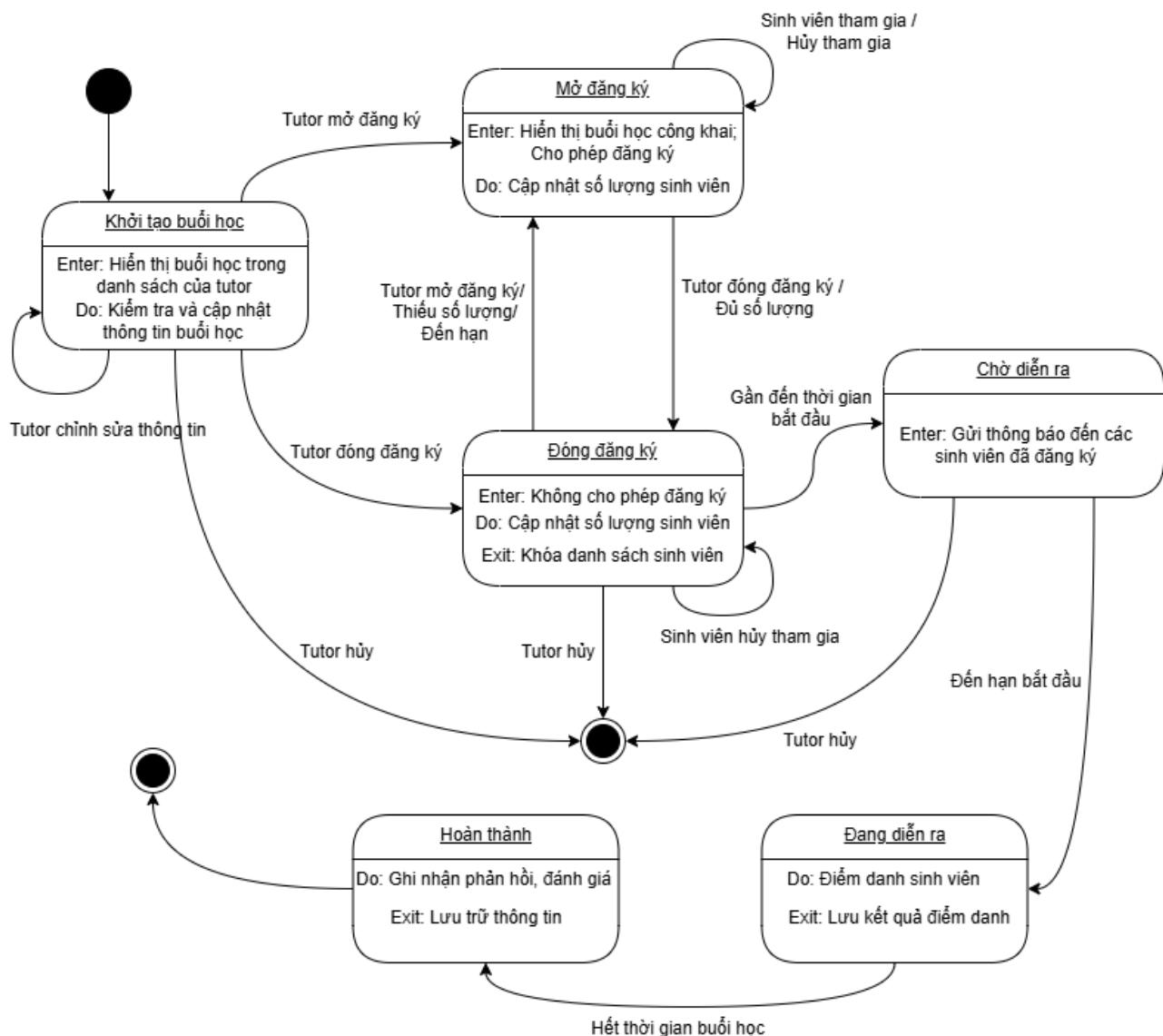
Activity diagram mô tả quy trình người dùng truy cập và xem tài liệu trong hệ thống. Quy trình bắt đầu khi người dùng chọn chức năng “Xem danh sách tài liệu”. Hệ thống tiếp nhận yêu cầu và thực hiện bước lấy danh sách tài liệu từ kho dữ liệu. Tại đây, hệ thống kiểm tra xem danh sách có trống hay không. Nếu không có tài liệu, hệ thống hiển thị thông báo danh sách rỗng và kết thúc quy trình. Nếu có dữ liệu, hệ thống hiển thị danh sách tài liệu cho người dùng.

Người dùng sau đó chọn một tài liệu cụ thể để truy cập, người dùng chọn xem hoặc tải tài liệu. Yêu cầu này được gửi đến hệ thống, hệ thống tiến hành kiểm tra tính khả dụng của tài liệu. Trong trường hợp tài liệu không khả dụng (ví dụ: bị xóa, hết quyền truy cập, lỗi lưu trữ), hệ thống hiển thị thông báo lỗi “Tài liệu không khả dụng” và kết thúc tiến trình. Nếu tài liệu khả dụng, hệ thống cho phép người dùng tiếp tục. Nếu người dùng chọn xem, hệ thống thực hiện stream tài liệu và hiển thị nội dung. Nếu người dùng chọn tải xuống, hệ thống kích hoạt chức năng tải file về thiết bị. Sau khi hoàn tất việc xem hoặc tải, hệ thống tiến hành bước cuối cùng là ghi nhận lại lịch sử truy cập, bao gồm thông tin về loại thao tác và thời điểm truy cập.



4.4 State-chart diagrams

State-chart diagram của đối tượng buổi học:



Hình 4.26: State diagram của đối tượng buổi học



Mô tả state diagram của đối tượng buổi học

State chart mô tả toàn bộ vòng đời của một buổi học, từ khi được tạo ra cho đến khi hoàn thành. Buổi học bắt đầu ở trạng thái Khởi tạo, nơi tutor tạo buổi học, thiết lập các thông tin cơ bản như tiêu đề, thời gian, mô tả, số lượng sinh viên tối đa và có thể cập nhật lại các thông tin sau đó.

Khi tutor mở đăng ký, buổi học chuyển sang trạng thái Mở đăng ký. Ở trạng thái này, buổi học được hiển thị công khai, cho phép sinh viên xem buổi học và đăng ký tham gia. Hệ thống theo dõi số lượng sinh viên tham gia và phản ánh lên giao diện. Sinh viên có thể đăng ký hoặc hủy đăng ký trong thời gian này.

Khi đủ số lượng, hết hạn đăng ký hoặc tutor chủ động đóng, buổi học chuyển sang Đóng đăng ký. Lúc này, hệ thống khóa form đăng ký và cập nhật danh sách sinh viên chính thức.

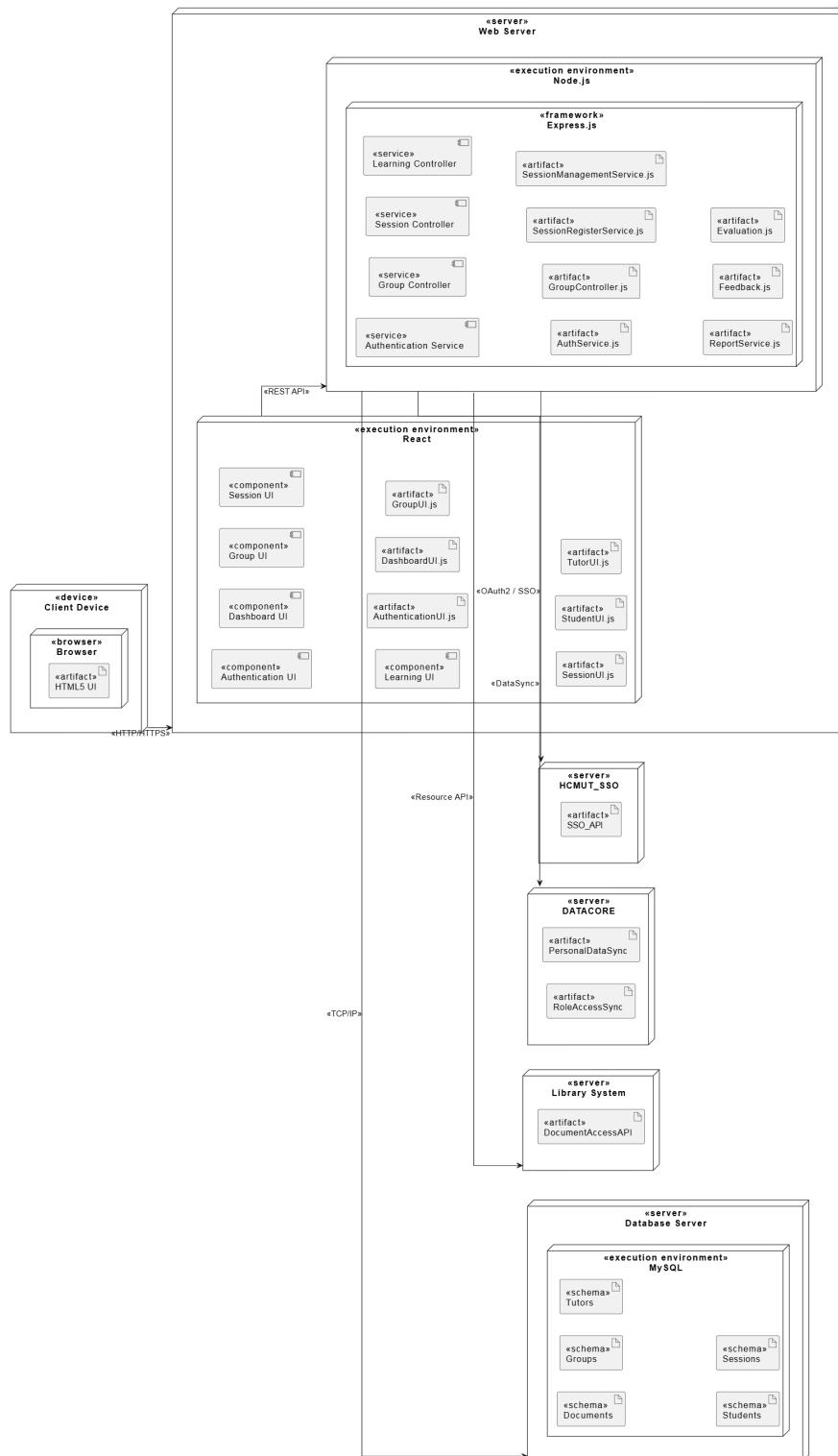
Khi tutor đóng đăng ký, hết thời hạn đăng ký hoặc đủ số lượng sinh viên, buổi học chuyển sang Đóng đăng ký. Tại đây hệ thống khóa đăng ký tham gia nhưng vẫn cho phép sinh viên hủy đăng ký. Gần thời gian bắt đầu, hệ thống chốt danh sách và chuyển buổi học sang trạng thái Chờ diễn ra, nơi hệ thống gửi thông báo cho các sinh viên đã đăng ký.

Đến thời điểm bắt đầu, tutor khởi động buổi học và trạng thái chuyển sang Đang diễn ra, bao gồm điểm danh và theo dõi quá trình học. Khi buổi học kết thúc, hệ thống chuyển sang Hoàn thành, nơi ghi nhận phản hồi, đánh giá và lưu trữ kết quả. Trong toàn bộ quá trình, tutor có thể hủy buổi học ở nhiều thời điểm, dẫn đến trạng thái kết thúc sớm.





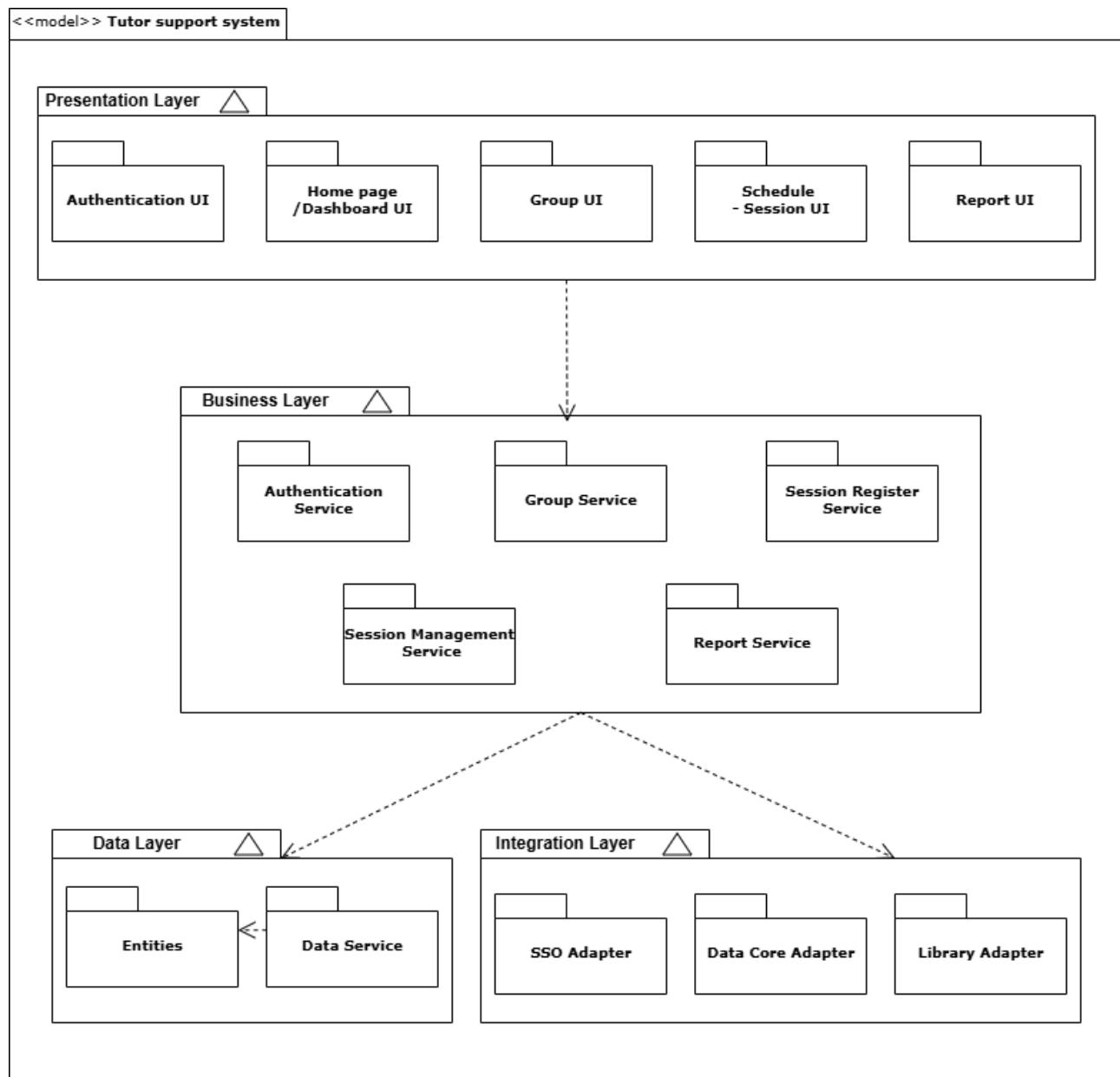
5 Deployment view



Hình 5.1: Deployment view



6 Development/Implementation view

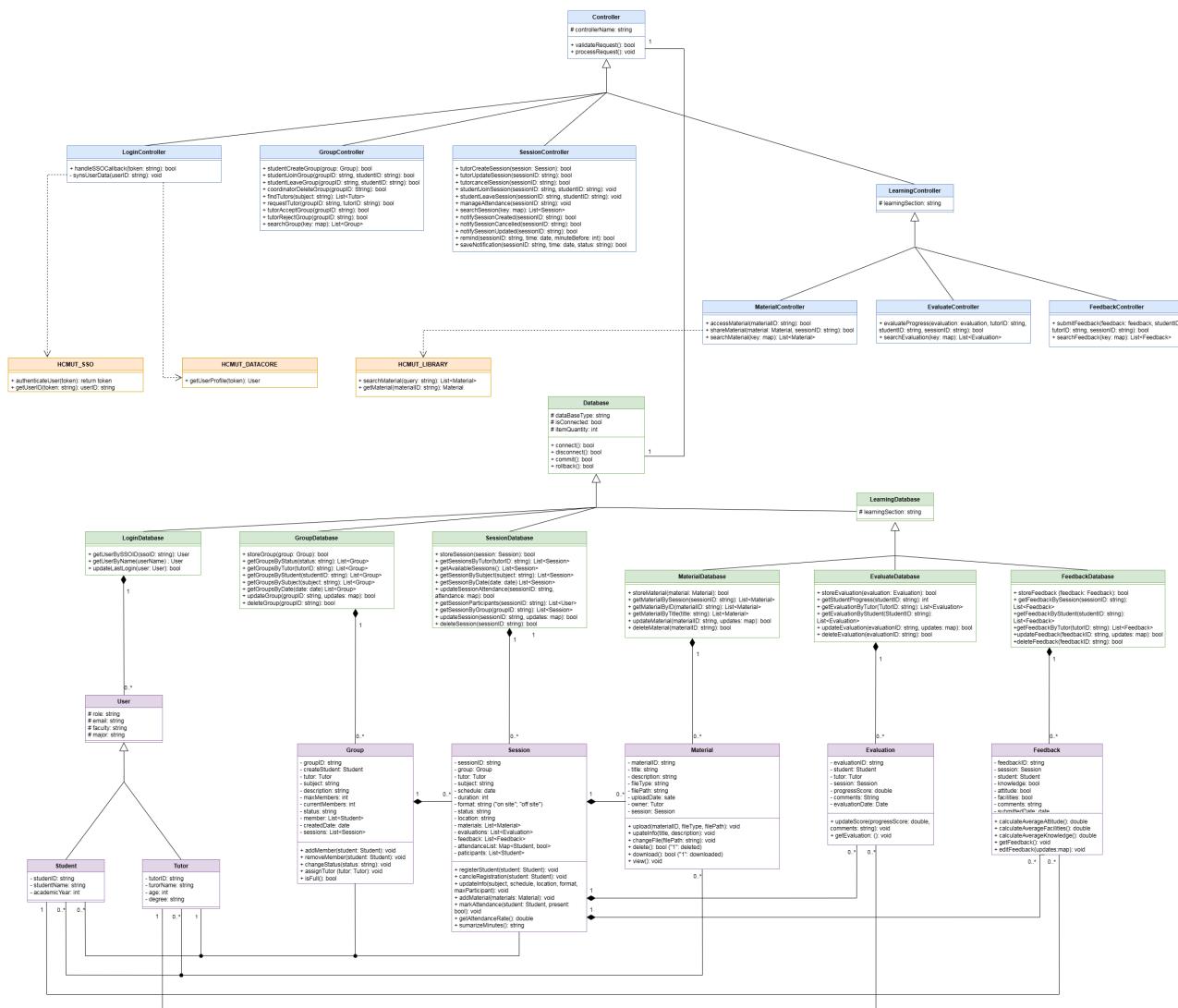


Hình 6.1: Development/Implementation view



7 Class diagram and Method descriptions

7.1 Class diagram



Hình 7.1: Class diagram

7.2 Method descriptions

Class	Attribute/Method	Mô tả
Session		
Session	- sessionID: string	Mã định danh buổi học.
	- group: Group	Nhóm học mà buổi học thuộc về.
	- tutor: Tutor	Tutor phụ trách buổi học.
	- subject: string	Môn học hoặc chủ đề của buổi học.
	- schedule: date	Ngày và giờ diễn ra buổi học.
	- duration: int	Thời lượng buổi học.
	- format: string	Hình thức học: " <i>on site</i> " hoặc " <i>off site</i> ".
	- status: string	Trạng thái buổi học (đang mở đăng ký, đã đóng đăng ký, đã kết thúc...).
	- location: string	Địa điểm diễn ra buổi học.
	- materials: List<Material>	Danh sách tài liệu chia sẻ trong buổi học.
	- evaluations: List<Evaluation>	Danh sách các đánh giá của tutor cho sinh viên.
	- feedback: List<Feedback>	Danh sách phản hồi của sinh viên.
	- attendanceList: Map<Student, bool>	Danh sách điểm danh: mỗi sinh viên kèm theo trạng thái có mặt.



Class	Attribute/Method	Mô tả
AttendanceManager	- participants: List<Student>	Danh sách sinh viên tham gia buổi học.
	+ registerStudent(student: Student): void	Đăng ký sinh viên vào buổi học.
	+ cancelRegistration(student: Student): void	Hủy đăng ký của một sinh viên.
	+ updateInfo(subject, schedule, location, format, maxParticipant): void	Cập nhật thông tin buổi học.
	+ addMaterial(material: Material): void	Thêm tài liệu mới cho buổi học.
	+ markAttendance(student: Student, present: bool): void	Đánh dấu điểm danh cho sinh viên.
	+ getAttendanceRate(): double	Tính và trả về tỉ lệ tham dự buổi học.
	+ summarizeMinutes(): string	Tổng hợp nội dung buổi học dưới dạng văn bản.
User		
User	+ userID: string	Mã định danh của mỗi người dùng.
	+ userName: string	Tên của người dùng.
	+ email: string	Địa chỉ email của người dùng.
	+ role: string	Vai trò của người dùng (student, tutor).
	+ faculty: string	Khoa, bộ môn của người dùng.

Class	Attribute/Method	Mô tả
	+ major: string	Chuyên ngành của người dùng.
Student		
Student	- studentID: string	Mã số sinh viên.
	- studentName: string	Họ tên của sinh viên.
	- academicYear: int	Năm học của sinh viên (ví dụ: 1, 2, 3, 4).
	- email: string	Địa chỉ email của sinh viên (có thể trùng với email trong User).
	- faculty: string	Khoa mà sinh viên theo học.
	- major: string	Chuyên ngành cụ thể của sinh viên.
Tutor		
Tutor	- tutorID: string	Mã định danh của tutor.
	- tutorName: string	Họ tên của tutor.
	- age: int	Tuổi của tutor.
	- email: string	Địa chỉ email của tutor.
	- faculty: string	Khoa mà tutor trực thuộc.
	- major: string	Chuyên ngành giảng dạy chính của tutor.
	- degree: string	Bằng cấp cao nhất của tutor (Cử nhân, Thạc sĩ, Tiến sĩ, ...).



Class	Attribute/Method	Mô tả
Group		
Group	- groupID: string	Mã định danh của nhóm
	- createStudent: Student	Sinh viên tạo nhóm
	- tutor: Tutor	Tutor của nhóm
	- subject: string	Môn học của nhóm
	- description: string	Mô tả về nhóm
	- maxMembers: int	Số lượng thành viên tối đa
	- currentMembers: int	Số lượng thành viên hiện tại
	- status: string	Trạng thái của nhóm
	- member: List<Student>	Danh sách các sinh viên trong nhóm
	- createdDate: date	Ngày tạo nhóm
	- sessions: List<Session>	Danh sách các buổi học của nhóm
	+ addMember(student: Student): void	Nhận đối tượng Student, thêm sinh viên vào nhóm hiện tại.
	+ removeMember(student: Student): void	Nhận đối tượng Student, xóa sinh viên khỏi nhóm hiện tại.
	+ changeStatus(status: string): void	Nhận string thể hiện trạng thái mới, thay đổi trạng thái của nhóm hiện tại.
	+ assignTutor(tutor: Tutor): void	Nhận đối tượng Tutor, giao nhóm hiện tại cho Tutor.

Class	Attribute/Method	Mô tả
	+ isFull(): bool	Kiểm tra nhóm đã đạt số tối đa hay chưa, trả về True nếu nhóm đầy, False nếu nhóm còn chỗ.
Session		
Session	- sessionID: string	Mã định danh của buổi học
	- group: Group	Nhóm của buổi học
	- tutor: Tutor	Tutor của buổi học
	- subject: string	Môn học
	- schedule: date	Lịch học
	- duration: int	Thời lượng buổi học
	- format: string	Hình thức ("on site"; "off site")
	- status: string	Trạng thái của buổi học
	- location: string	Địa điểm học
	- materials: List<Material>	Danh sách tài liệu
	- evaluations: List<Evaluation>	Danh sách đánh giá
	- feedback: List<Feedback>	Danh sách phản hồi
	- attendanceList: Map<Student, bool>	Danh sách điểm danh
	- participants: List<Student>	Danh sách người tham gia
	+ registerStudent(student: Student): void	Nhận đối tượng Student, giúp sinh viên đăng ký buổi học mới.



Class	Attribute/Method	Mô tả
	+ cancelRegistration(student: Student): void	Nhận đối tượng Student, hủy đăng ký của sinh viên.
	+ updateInfo(subject, schedule, location, format, maxParticipant): void	Nhận môn học subject, thời gian schedule, địa điểm location, hình thức học format (trực tiếp, trực tuyến, kết hợp), số lượng tối đa maxParticipant, cập nhật các thông tin cho buổi học.
	+ addMaterial(materials: Material): void	Nhận tài liệu cần thêm Material, thêm tài liệu cho buổi học.
	+ markAttendance(student: Student, present: bool): void	Nhận đối tượng Student, trạng thái có mặt present, giúp sinh viên điểm danh (present == True tức là có mặt, ngược lại tức là vắng).
	+ getAttendanceRate(): double	Trả về số thực biểu thị tỷ lệ phần trăm sinh viên có mặt.
	+ summarizeMinutes(): string	Tạo và trả về chuỗi tóm tắt biên bản buổi học.
Material		
Material	- materialID: string	Mã định danh của tài liệu
	- title: string	Tiêu đề tài liệu
	- description: string	Mô tả tài liệu
	- fileType: string	Loại file
	- filePath: string	Đường dẫn file

Class	Attribute/Method	Mô tả
	- uploadDate: date	Ngày tải lên
	- owner: Tutor	Tutor sở hữu tài liệu
	- session: Session	Buổi học của tài liệu
	+ upload(materialID, fileType, filePath): void	Nhận materialID, fileType, filePath, giúp tải tài liệu lên hệ thống (biên bản buổi học, tài liệu học tập).
	+ delete(): bool	Xóa tập tin hiện tại khỏi hệ thống, trả về True nếu tập tin được xóa thành công, ngược lại trả về False.
	+ download(): bool	Tải về tập tin hiện tại, trả về True nếu tập tin được tải về thành công, ngược lại trả về False.
	+ view(): void	Hiển thị nội dung tập tin hiện tại.
Evaluation		
Evaluation	- evaluationID: string	Mã định danh của đánh giá
	- student: Student	Sinh viên được đánh giá
	- tutor: Tutor	Tutor thực hiện đánh giá
	- session: Session	Buổi học được đánh giá
	- progressScore: double	Điểm tiến độ
	- comments: String	Nhận xét
	- evaluationDate: Date	Ngày đánh giá



Class	Attribute/Method	Mô tả
	+ updateScore(progressScore: double, comments: string): void	Nhận progressScore (double) và comments (string), cập nhật điểm và nhận xét.
	+ getEvaluation(): void	Hiển thị đánh giá của tutor cho sinh viên.
Feedback		
Feedback	- feedbackID: string	Mã định danh của phản hồi
	- session: Session	Buổi học được phản hồi
	- student: Student	Sinh viên gửi phản hồi
	- knowledge: bool	Đánh giá về kiến thức
	- attitude: bool	Đánh giá về thái độ
	- facilities: bool	Đánh giá về cơ sở vật chất
	- comments: string	Nhận xét
	- submittedDate: date	Ngày gửi phản hồi
	+ calculateAverageAttitude(): double	Tính toán và trả về số thực biểu thị tỷ lệ phần trăm phản hồi có mục "Giảng viên nhiệt tình" được đánh dấu "Đạt".
	+ calculateAverageFacilities(): double	Tính toán và trả về số thực biểu thị tỷ lệ phần trăm phản hồi có mục "Cơ sở vật chất tốt" được đánh dấu "Đạt".

Class	Attribute/Method	Mô tả
	+ calculateAverageKnowledge(): double	Tính toán và trả về số thực biểu thị tỷ lệ phần trăm phản hồi có mục "Năm bắt được kiến thức" được đánh dấu "Đạt".
	+ getFeedback(): void	Trả về phản hồi của sinh viên.
	+ editFeedback(knowledge: bool, attitude: bool, facilities: bool, comments: string): void	Nhận vào các tham số knowledge (True == Năm bắt được kiến thức), attitude (True == Giảng viên nhiệt tình), facilities (True == Cơ sở vật chất tốt), string comments là phản hồi thêm nếu có. Chức năng là điều chỉnh phản hồi của sinh viên

Class	Attribute/Method	Mô tả
Controller		
Controller	# controllerName: string	Tên của bộ điều khiển (ví dụ: 'SessionController')
	+ validateRequest(): bool	Kiểm tra tính hợp lệ của yêu cầu từ người dùng và trả về bool (True/False)
	+ processRequest(): void	Xử lý yêu cầu đã được xác thực
SessionController		



Class	Attribute/Method	Mô tả
Session Controller	+ tutorCreateSession(session: Session): bool	Nhận vào đối tượng session, tạo một buổi học mới (yêu cầu từ Tutor) và trả về bool (True/False).
	+ tutorUpdateSession(sessionID: string, updates: map): bool	Nhận vào sessionID và các thông tin cập nhật, cập nhật thông tin buổi học (yêu cầu từ Tutor) và trả về bool (True/False).
	+ tutorCancelSession(sessionID: string): bool	Nhận vào sessionID, hủy một buổi học (yêu cầu từ Tutor) và trả về bool (True/False).
	+ studentJoinSession(sessionID: string, studentID: string): void	Nhận vào sessionID và studentID. Thêm học viên vào buổi học (yêu cầu từ sinh viên) và trả về bool (True/False).
	+ studentLeaveSession(sessionID: string, studentID: string): void	Nhận vào sessionID và studentID. Xoá học viên khỏi buổi học (yêu cầu từ sinh viên) và trả về bool (True/False).
	+ manageAttendance(sessionID): void	Nhận vào sessionID, quản lý việc điểm danh cho buổi học.
	+ searchSession(key: map): List<Session>	Nhận vào các khoá, tìm kiếm buổi học theo tiêu chí (ví dụ: môn học, ngày) và trả về danh sách session.



Class	Attribute/Method	Mô tả
	+ notifySessionCreated(sessionID: string): bool	Nhận vào sessionID, gửi thông báo khi một buổi học được tạo thành công và trả về bool (True/False).
	+ notifySessionCancelled(sessionID: string): bool	Nhận vào sessionID, gửi thông báo khi một buổi học bị hủy và trả về bool (True/False).
	+ notifySessionUpdated(sessionID: string): bool	Nhận vào sessionID, gửi thông báo khi thông tin buổi học được cập nhật và trả về bool (True/False).
	+ remindSession(sessionID: string, time: date, minuteBefore: int): bool	Nhận vào sessionID, thời gian buổi học, thời gian trước buổi học. Đặt lịch nhắc nhở buổi học trước một khoảng thời gian nhất định và trả về bool (True/False).
	+ saveNotification(sessionID: string, time: date, status: string): bool	Nhận vào sessionID, thời gian buổi học, trạng thái buổi học. Lưu thông báo về buổi học .
LearningController		
Learning Controller	# learningSection: string	Loại phần học tập (ví dụ: 'Material', 'Evaluate', 'Feedback')
MaterialController		
Material Controller	+ accessLearningMaterial(materialID: string): bool	Nhận vào materialID, truy cập tài liệu học tập của buổi học và trả về bool (True/False).



Class	Attribute/Method	Mô tả
	+ shareLearningMaterial(material: Material, sessionID: string): bool	Nhận vào đối tượng material, sessionID. Chia sẻ tài liệu học tập vào một buổi học, và trả về bool (True/False)
	+ searchMaterial(key: map): List<Material>	Nhận vào các khoá, tìm kiếm tài liệu học tập theo tiêu chí và trả về danh sách phản hồi.
EvaluateController		
Evaluate Controller	+ evaluateProgress(evaluation: evaluation, tutorID: string, studentID: string, sessionID: string): bool	Nhận vào đối tượng evalution, tutorID, studentID, sessionID, tutor đánh giá tiến độ của học viên và trả về bool (True/False).
	+ searchEvaluation(key: map): List<Evaluation>	Nhận vào các khoá, tìm kiếm đánh giá theo tiêu chí và trả về danh sách đánh giá.
FeedbackController		
Feedback Controller	+ submitFeedback(feedback: Feedback, studentID: tutorID: string, sessionID: string): bool	Nhận vào đối tượng feedback, studentID, tutorID, sessionID, gửi phản hồi về buổi học và trả về bool (True/False).
	+ searchFeedback(key: map): List<Feedback>	Nhận vào các khoá, tìm kiếm phản hồi theo tiêu chí và trả danh sách phản hồi.
LoginController		
LoginController	# currUser: User	Đối tượng đại diện cho user hiện tại đang dùng hệ thống

Class	Attribute/Method	Mô tả
	+ handleSSOCallback(token: string): bool	Nhận vào token là tín hiệu trả về khi hệ thống SSO xác thực thành công và gọi synsUserData(userID: string) để thêm User vào DB (nếu chưa có) và cập nhật currUser và trả về kết quả bool.
	- synsUserData(userID: string): void	Nhận vào UserID và kiểm tra người dùng đã có trong DB chưa, nếu chưa có thì tạo đối tượng mới lưu vào DB dựa trên dữ liệu nhận được từ DATACORE, nếu đã có thì cập nhật lại dữ liệu của đối tượng.
GroupController		
GroupController	+ studentCreateGroup(group: Group): bool	Nhận thông tin cơ bản về group từ UI, tạo đối tượng Group và gọi storeGroup để lưu vào DB, trả về True nếu thành công
	+ studentJoinGroup(groupID: string, studentID: string): bool	Nhận vào groupID và studentID, thêm student vào group có ID tương ứng, True nếu thành công.
	+ studentLeaveGroup(groupID: string, studentID: string): bool	Nhận vào groupID và studentID, xóa student đó khỏi group có ID tương ứng, True nếu thành công.



Class	Attribute/Method	Mô tả
	+ coordinatorDeleteGroup(groupID: String): bool	Nhận vào groupID, xóa group có ID tương ứng, True nếu thành công.
	+ findTutors(subject: string): List<Tutor>	Nhận vào tên môn học, tìm và trả về danh sách các tutor của môn học đó.
	+ requestTutor(groupID: string, tutorID: string): bool	Nhận vào groupID và tutorID, tạo yêu cầu tutor nhận nhóm có ID tương ứng và True nếu thành công.
	+ tutorAcceptGroup(groupID: string): bool	Nhận vào groupID, tutor thực hiện sẽ được thêm vào group tương ứng, và đặt lại trạng thái của group là "đã được hướng dẫn" True nếu thành công.
	+ tutorRejectGroup(groupID: string): bool	Nhận vào groupID, xóa group có ID tương ứng khỏi danh sách nhóm được yêu cầu của tutor, và trả về True nếu thành công.
	+ searchGroup(key: map): List<Group>	Nhận vào giá trị khóa key và trả về danh sách các group tương ứng với khóa đó.
HCMUT_SSO		
HCMUT_SSO	+ getUserId(token: string): userID: string	Nhận vào giá trị token và trả về userID tương ứng của token đó

Class	Attribute/Method	Mô tả
	+ authenticateUser(userID: string, password: string): token: string	Nhận vào ID và password lấy từ UI và xác thực người dùng, nếu xác thực đúng thì trả về token.
HCMUT _ DATACORE		
HCMUT _ DATACORE	+ getUserProfile(token): User	Nhận vào token, một user được khởi tạo bên trong hàm với các thông tin lấy từ datacore và trả về đối tượng user đó.
HCMUT _ LIBRARY		
HCMUT _ LIBRARY	+ searchMaterial(query: string): List<Material>	Nhận vào giá trị query là từ khóa để truy vấn tài liệu, tìm và trả về danh sách các tài liệu tương ứng với từ khóa.
	+ getMaterial(materialID: string): Material	Nhận vào materialID, đối tượng material được tạo trong hàm với các thông tin lấy từ LIBRARY và trả về đối tượng vừa tạo.
Database		
Database	# DataBaseType: string	Loại database (Login, Group, Session, Learning)
	# isConnected: bool	Một biến boolean để kiểm tra xem đã kết nối với database chưa
	# itemQuantity: int	Số lượng bản ghi lưu trong database



Class	Attribute/Method	Mô tả
LoginDatabase	+ connect(): void	Thiết lập kết nối đến database.
	+ disconnect(): void	Ngắt kết nối với database.
	+ commit(): void	Ghi lại những thay đổi.
	+ rollback(): void	Hoàn tác thay đổi.
LoginDatabase		
LoginDatabase	+ getUserByUserID(userID: string): User	Nhận giá trị userID, trả về một object User.
	+ getUserByName(username: string): User	Nhận giá trị username, trả về một object User.
	+ updateLastLogin(user: User): bool	Nhận giá trị là một đối tượng User, cập nhật thời điểm đăng nhập gần nhất và trả về bool (True/False).
GroupDatabase		
GroupDatabase	+ storeGroup(group: Group): bool	Nhận giá trị là một đối tượng Group, lưu nhóm mới và trả về bool (True/False).
	+ getGroupByStatus(status: string): List<Group>	Nhận giá trị status, trả về danh sách các nhóm (List<Group>) theo trạng thái.
	+ getGroupByTutor(tutorID: string): List<Group>	Nhận giá trị tutorID, trả về danh sách các nhóm (List<Group>) theo ID của tutor.

Class	Attribute/Method	Mô tả
	+ getGroupByStudent(studentID: string): List<Group>	Nhận giá trị studentID, trả về danh sách các nhóm (List<Group>) của Student.
	+ getGroupBySubject(subject: string): List<Group>	Nhận giá trị subject, trả về danh sách các nhóm (List<Group>) theo môn học.
	+ getGroupByDate(date: Date): List<Group>	Nhận giá trị date, trả về danh sách nhóm (List<Group>) theo ngày.
	+ updateGroup(groupID: string, update: map): bool	Nhận giá trị groupID và map update, cập nhật nhóm và trả về bool (True/False).
	+ deleteGroup(groupID: string): bool	Nhận giá trị groupID, xóa nhóm và trả về bool (True/False).

SessionDatabase

SessionDatabase	+ storeSession(session: Session): bool	Nhận giá trị là một đối tượng Session, lưu buổi học và trả về bool (True/False).
	+ getSessionsByTutor(tutorID: string): List<Session>	Nhận giá trị tutorID, trả về danh sách buổi học (List<Session>) theo tutorID.
	+ getAvailableSessions(): List<Session>	Trả về danh sách các buổi học (List<Session>) có thể tham gia.
	+ getSessionBySubject(subject: string): List<Session>	Nhận giá trị subject, trả về danh sách các buổi học (List<Session>) theo môn.



Class	Attribute/Method	Mô tả
	+ getSessionByDate(date: date): List<Session>	Nhận giá trị date, trả về danh sách các buổi học (List<Session>) theo ngày.
	+ updateSessionAttendance(sessionID: string, attendance: map): bool	Nhận giá trị sessionID và map attendance, cập nhật điểm danh và trả về bool (True/False).
	+ getSessionParticipants(sessionID: string): List<User>	Nhận giá trị sessionID, trả về danh sách người tham dự (List<User>).
	+ getSessionByGroup(groupID: string): List<Session>	Nhận giá trị groupID, trả về danh sách các lớp học (List<Session>) theo groupID.
	+ updateSession(sessionID: string, updates: map): bool	Nhận giá trị sessionID và map updates, cập nhật thông tin buổi học và trả về bool (True/False).
	+ deleteSession(sessionID: string): bool	Nhận giá trị sessionID, xóa buổi học và trả về bool (True/False).
LearningDatabase		
LearningDatabase	# learningSection: string	Loại learning database
MaterialDatabase		
	+ storeMaterial(material: Material): bool	Nhận giá trị là một đối tượng Material, lưu tài liệu học tập và trả về bool (True/False).

MaterialDatabase

Class	Attribute/Method	Mô tả
	+ getMaterialBySession(sessionID: string): List<Material>	Nhận giá trị sessionID, trả về danh sách tài liệu (List<Material>) theo ID buổi học.
	+ getMaterialByID(materialID: string): List<Material>	Nhận giá trị materialID, trả về danh sách tài liệu (List<Material>) theo ID tài liệu.
	+ getMaterialByTitle(title: string): List<Material>	Nhận giá trị title, trả về danh sách tài liệu (List<Material>) theo tên tài liệu.
	+ updateMaterial(materialID: string, updates: map): bool	Nhận giá trị materialID và map updates, cập nhật tài liệu và trả về bool (True/False).
	+ deleteMaterial(materialID: string): bool	Nhận giá trị materialID, xóa tài liệu và trả về bool (True/False).
EvaluateDatabase		
EvaluateDatabase	+ storeEvaluation(evaluation: Evaluation): bool	Nhận giá trị là một đối tượng Evaluation, lưu đánh giá và trả về bool (True/False).
	+ getStudentProgress(studentID: string): int	Nhận giá trị studentID, trả về tiến độ của sinh viên (int).
	+ getEvaluationByTutor(TutorID: string): List<Evaluation>	Nhận giá trị TutorID, trả về danh sách đánh giá (List<Evaluation>) của Tutor.



Class	Attribute/Method	Mô tả
	+ getEvaluationByStudent(StudentID: string): List<Evaluation>	Nhận giá trị StudentID, trả về danh sách đánh giá (List<Evaluation>) của học sinh.
	+ updateEvaluation(evaluationID: string, updates: map): bool	Nhận giá trị evaluationID và map updates, cập nhật đánh giá và trả về bool (True/False).
	+ deleteEvaluation(evaluationID: string): bool	Nhận giá trị evaluationID, xóa đánh giá và trả về bool (True/False).
FeedbackDatabase		
FeedbackDatabase	+ storeFeedback (feedback: Feedback): bool	Nhận giá trị là một đối tượng Feedback, lưu phản hồi và trả về bool (True/False).
	+ getFeedbackBySession(sessionID: string): List<Feedback>	Nhận giá trị sessionID, trả về danh sách phản hồi (List<Feedback>) theo buổi học.
	+ getFeedbackByStudent(studentID: string): List<Feedback>	Nhận giá trị studentID, trả về danh sách phản hồi (List<Feedback>) theo sinh viên.
	+ getFeedbackByTutor(tutorID: string): List<Feedback>	Nhận giá trị tutorID, trả về danh sách phản hồi (List<Feedback>) của Tutor.

Class	Attribute/Method	Mô tả
	+ updateFeedback(feedbackID: string, updates: map): bool	Nhận giá trị feedbackID và map updates, cập nhật phản hồi và trả về bool (True/False).
	+ deleteFeedback(feedbackID: string): bool	Nhận giá trị feedbackID, xóa phản hồi và trả về bool (True/False).

8 Testcase

Test case ID	DK-01
Description	Kiểm tra khi sinh viên điền thông tin hợp lệ, hệ thống thêm nhóm thành công
Pre-conditions	Người dùng đã đăng nhập thành công vào hệ thống
Steps	<ol style="list-style-type: none"> Sinh viên chọn đăng ký nhóm. Hệ thống hiện bảng điền thông tin nhóm. Sinh viên chọn lĩnh vực/môn học cần hỗ trợ, đặt tiêu đề, điền mô tả nội dung cần hỗ trợ, chọn số lượng thành viên và chọn “Đăng ký”.



Expected Result	<ol style="list-style-type: none">Hệ thống tạo một đối tượng là nhóm với các thông tin người dùng đã điền và đặt trạng thái đăng ký là đang chờ hướng dẫn.Hệ thống thêm nhóm vào kho danh sách các nhóm đang yêu cầu của lĩnh vực.Hệ thống thêm nhóm vào kho danh sách nhóm của sinh viên đăng ký.Hệ thống gửi thông báo xác nhận đăng ký thành công.
Actual Result	
Status	

Test case ID	DK-02
Description	Kiểm tra khi sinh viên huỷ yêu cầu trong lúc điền thông tin
Pre-conditions	<ol style="list-style-type: none">Hệ thống đang hiển thị bảng điền thông tin nhóm.
Steps	<ol style="list-style-type: none">Sinh viên chọn “Huỷ yêu cầu”.
Expected Result	<ol style="list-style-type: none">Hệ thống huỷ tác vụ và quay lại trang đăng ký nhóm.Hệ thống thông báo đã huỷ yêu cầu.
Actual Result	
Status	

Test case ID	TG-01
Description	Kiểm tra khi Tutor nhận một yêu cầu hướng dẫn của nhóm sinh viên



Pre-conditions	1. Có ít nhất một nhóm gửi yêu cầu hướng dẫn.
Steps	<ol style="list-style-type: none"> 1. Tutor chọn xem danh sách nhóm yêu cầu. 2. Hệ thống hiển thị danh sách các nhóm đang yêu cầu. 3. Tutor chọn xem thông tin chi tiết của một nhóm. 4. Tutor chọn “Chấp nhận”.
Expected Result	<ol style="list-style-type: none"> 1. Hệ thống cập nhật trạng thái nhóm thành “Đã chấp nhận”. 2. Hệ thống tạo thông báo chấp nhận gửi cho sinh viên. 3. Hệ thống hiển thị thông báo: “Xác nhận chấp nhận thành công”.
Actual Result	
Status	

Test case ID	TG-02
Description	Kiểm tra khi Tutor từ chối một yêu cầu hướng dẫn của nhóm sinh viên
Pre-conditions	<ol style="list-style-type: none"> 1. Có ít nhất một nhóm gửi yêu cầu hướng dẫn.



Steps	<ol style="list-style-type: none">1. Tutor chọn xem danh sách nhóm yêu cầu.2. Hệ thống hiển thị danh sách các nhóm đang yêu cầu.3. Tutor chọn xem thông tin chi tiết của một nhóm.4. Tutor chọn “Từ chối”.
Expected Result	<ol style="list-style-type: none">1. Hệ thống cập nhật trạng thái nhóm thành “Đã từ chối”.2. Hệ thống tạo thông báo từ chối gửi cho sinh viên.3. Hệ thống hiển thị thông báo: “Xác nhận từ chối thành công”.
Actual Result	
Status	

Test case ID	BH-01
Description	Kiểm tra khi Tutor đăng ký buổi dạy mới
Pre-conditions	<ol style="list-style-type: none">1. Tutor đã đăng nhập vào hệ thống.
Steps	<ol style="list-style-type: none">1. Tutor chọn chức năng “Đăng ký buổi dạy”.2. Hệ thống hiển thị bảng điều kiện thông tin buổi dạy.3. Tutor nhập đầy đủ tất cả thông tin.4. Tutor chọn “Đăng ký”.



Expected Result	<ul style="list-style-type: none"> 1. Hệ thống kiểm tra dữ liệu hợp lệ. 2. Hệ thống tạo một đối tượng buổi học và đặt trạng thái “có thể đăng ký”. 3. Hệ thống thêm buổi học vào danh sách buổi học của tutor. 4. Hệ thống thêm buổi học vào danh sách buổi học mở đăng ký. 5. Hệ thống gửi thông báo xác nhận đăng ký thành công.
Actual Result	
Status	