

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



Bài tập lớn Công nghệ Phần mềm (CO3001) - HK251

---

*TUTOR SUPPORT SYSTEM AT  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY*

---

GVHD: Mai Đức Trung

Tên nhóm: Nhóm 6

Thành viên:

Nguyễn Đăng Khoa	2311614
Hồ Minh Nhiên	2312517
Vũ Huy Gia Khang	2311486
Nguyễn Văn Hiếu	2310967
Lê Thúy Hiền	2310990
Hoàng Thị Hằng	2310901
Đỗ Quang Long	2311896

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025



## Danh sách thành viên

STT	Họ và tên	MSSV	Tỉ lệ hoàn thành
1	Nguyễn Đăng Khoa	2311614	100%
2	Hồ Minh Nhiên	2312517	100%
3	Vũ Huy Gia Khang	2311486	100%
4	Nguyễn Văn Hiếu	2310967	100%
5	Lê Thúy Hiền	2310990	100%
6	Hoàng Thị Hằng	2310901	100%
7	Dỗ Quang Long	2311896	100%



## Mục lục

<b>1 Task 1</b>	<b>5</b>
1.1 Phân tích ngữ cảnh . . . . .	5
1.1.1 Bối cảnh dự án . . . . .	5
1.1.2 Relevant stakeholders . . . . .	6
1.1.3 Mục tiêu . . . . .	8
1.1.4 Phạm vi . . . . .	9
1.2 Yêu cầu chức năng . . . . .	10
1.2.1 Đối với sinh viên . . . . .	10
1.2.2 Đối với Tutor . . . . .	11
1.2.3 Đối với Điều phối viên – Coordinator . . . . .	13
1.2.4 Đối với Ban quản lý (Phòng Đào tạo, Phòng Công tác Sinh viên, Khoa/Bộ môn) . . . . .	13
1.2.5 Đối với HCMUT_DATACORE . . . . .	14
1.2.6 Đối với HCMUT_SSO . . . . .	14
1.2.7 Đối với HCMUT_Library . . . . .	14
1.3 Yêu cầu phi chức năng (Non-functional Requirements) . . . . .	15
1.3.1 Hiệu năng và khả năng mở rộng (Performance & Scalability) . . . . .	15
1.3.2 Bảo mật và tuân thủ pháp lý (Security & Compliance) . . . . .	15
1.3.3 Tính khả dụng và khôi phục (Availability & Disaster Recovery) . . . . .	15
1.3.4 Khả năng tương thích (Compatibility) . . . . .	16
1.3.5 Khả năng sử dụng (Usability) . . . . .	16
1.3.6 Khả năng bảo trì (Maintainability) . . . . .	16
1.3.7 Hỗ trợ kỹ thuật (Technical Support) . . . . .	16
1.3.8 Độ tin cậy (Reliability) . . . . .	17
1.4 Use-case diagram . . . . .	18
1.4.1 Các actor của hệ thống . . . . .	18
1.4.2 Các nhánh use case chính . . . . .	19
1.4.3 Use-case ch0 toàn bộ hệ thống . . . . .	20
1.5 Use-case detail/scenario . . . . .	21
1.5.1 Đăng nhập . . . . .	21
1.5.2 Quên mật khẩu . . . . .	24
1.5.3 Đăng xuất . . . . .	25
1.5.4 Xem tài liệu . . . . .	27
1.5.5 Xem hồ sơ cá nhân . . . . .	28
1.5.6 Điều phối khung chương trình chung cho các tutor . . . . .	29
1.5.7 Gửi báo cáo đến các phòng ban . . . . .	30
1.5.8 Đăng ký Tutor . . . . .	32
1.5.9 Đặt lịch hẹn . . . . .	34
1.5.10 Phản hồi chất lượng . . . . .	35
1.5.11 Đăng ký buổi tư vấn . . . . .	36
1.5.12 Hủy đăng ký buổi gặp mặt . . . . .	37
1.5.13 Xem danh sách buổi gặp mặt của sinh viên . . . . .	38
1.5.14 Tạo buổi tư vấn . . . . .	40
1.5.15 Hủy buổi gặp . . . . .	42
1.5.16 Ghi nhận tiến độ học tập của sinh viên . . . . .	43
1.5.17 Thiết lập lịch rảnh . . . . .	45
1.5.18 Đăng tải tài liệu . . . . .	47



1.5.19 Xóa tài liệu . . . . .	48
1.5.20 Xử lý yêu cầu đặt lịch hẹn . . . . .	50
1.5.21 Xem danh sách buổi gặp mặt của Tutor . . . . .	53
<b>2 Task 2</b>	<b>55</b>
2.1 Sequence Diagram và Activity Diagram . . . . .	55
2.1.1 Đăng nhập . . . . .	55
2.1.2 Quên mật khẩu . . . . .	59
2.1.3 Thiết lập lịch rảnh . . . . .	62
2.1.4 Đặt lịch hẹn . . . . .	66
2.1.5 Ghi nhận tiến độ . . . . .	69
2.1.6 Đăng tải tài liệu . . . . .	73
2.1.7 Phản hồi chất lượng buổi học . . . . .	76
2.1.8 Hủy buổi gặp mặt . . . . .	78
2.1.9 Xử lý lịch hẹn . . . . .	82
2.1.10 Tạo buổi tư vấn . . . . .	86
2.1.11 Điều phối khung chương trình chung . . . . .	90
2.2 Mockup . . . . .	94
2.2.1 Trang đăng nhập . . . . .	94
2.2.2 Quên mật khẩu . . . . .	95
2.2.3 Đăng ký tutor . . . . .	95
2.2.4 Sinh viên . . . . .	98
2.2.5 Tutor . . . . .	102
2.2.6 Điều phối viên . . . . .	108
2.3 State-chart diagram . . . . .	109
2.3.1 Yêu cầu đặt lịch hẹn . . . . .	109
2.3.2 Thiết lập lịch rảnh . . . . .	110
2.3.3 Đăng ký Tutor . . . . .	112
2.3.4 Đăng tải và xóa tài liệu . . . . .	113
<b>3 Task 3</b>	<b>115</b>
3.1 Deployment view . . . . .	115
3.2 Development/Implementation view . . . . .	117
3.2.1 Subsystem 1: Authentication . . . . .	117
3.2.2 Subsystem 2: StudentManager . . . . .	118
3.2.3 Subsystem 3: TutorManager . . . . .	118
3.2.4 Subsystem 4: Appointment . . . . .	118
3.2.5 Subsystem 5: Document . . . . .	119
3.2.6 Subsystem 6: FeedbackManager . . . . .	119
3.2.7 Subsystem 7: Coordinator . . . . .	119
3.3 Class diagram . . . . .	120
3.4 Design Pattern áp dụng . . . . .	120
3.4.1 MVC mở rộng . . . . .	120
3.4.2 Facade . . . . .	121
3.4.3 Strategy Pattern . . . . .	123
3.5 Mô tả phương thức . . . . .	124
3.5.1 View Method . . . . .	124
3.5.2 Controller Method . . . . .	133
3.5.3 Service Method . . . . .	139



---

3.5.4	Repository Method (Implementation) . . . . .	146
3.5.5	Repository Interface Method . . . . .	152
3.5.6	Service Interface Method . . . . .	155
3.5.7	Entity Method . . . . .	160
3.6	Test case . . . . .	165
3.6.1	Dăng nhập . . . . .	166
3.6.2	Dăng xuất . . . . .	173
3.6.3	Dăng ký Tutor . . . . .	177
3.6.4	Đặt lịch hẹn . . . . .	180
3.6.5	Phản hồi chất lượng buổi học . . . . .	182
3.6.6	Xem danh sách buổi gặp mặt . . . . .	187
3.6.7	Hủy buổi gặp mặt . . . . .	188
3.6.8	Xử lý yêu cầu đặt lịch hẹn . . . . .	189
<b>4</b>	<b>Task 4</b>	<b>190</b>
4.1	Hiện thực . . . . .	190
4.1.1	Spring Boot (Backend) . . . . .	191
4.1.2	ReactJS (Frontend) . . . . .	191
4.2	Luồng hoạt động của hệ thống . . . . .	192
4.2.1	Student . . . . .	193
4.2.2	Tutor . . . . .	199
4.3	Về việc sử dụng AI . . . . .	203
<b>5</b>	<b>Nhật ký thay đổi</b>	<b>204</b>



## 1 Task 1

### 1.1 Phân tích ngữ cảnh

#### 1.1.1 Bối cảnh dự án

Trong môi trường giáo dục đại học hiện nay, việc hỗ trợ sinh viên không chỉ dừng lại ở giảng dạy kiến thức trên lớp mà còn bao gồm tư vấn, kèm cặp và phát triển kỹ năng. Nhằm đáp ứng nhu cầu đó, Trường Đại học Bách Khoa – ĐHQG TP.HCM (HCMUT) đã triển khai chương trình Tutor/Mentor, trong đó các tutor là giảng viên, nghiên cứu sinh hoặc sinh viên năm trên có thành tích học tập tốt, được phân công để đồng hành và hỗ trợ các nhóm sinh viên trong quá trình học tập.

Để quản lý chương trình hiệu quả hơn, mở rộng quy mô và tránh quản lý thủ công, rườm rà và dễ sai sót, nhà trường mong muốn xây dựng một hệ thống phần mềm để quản lý và vận hành chương trình Tutor một cách hiệu quả. Hệ thống sẽ hỗ trợ quản lý hồ sơ sinh viên và tutor, đăng ký và ghép cặp, tổ chức và quản lý lịch tư vấn, đồng thời cung cấp công cụ thông báo, nhắc nhở và đánh giá. Bên cạnh đó, các khoa, phòng ban có thể khai thác dữ liệu tổng hợp để giám sát chất lượng đào tạo, phân bổ nguồn lực, cộng điểm rèn luyện hoặc xét học bổng cho sinh viên.

Hệ thống sẽ được tích hợp với các hạ tầng công nghệ của HCMUT như **HCMUT\_SSO** (đăng nhập tập trung), **HCMUT\_DATACORE** (đồng bộ dữ liệu cá nhân và phân quyền tự động) và **HCMUT\_LIBRARY** (truy cập, chia sẻ học liệu chính thống). Việc tích hợp này giúp đảm bảo an toàn, đồng bộ dữ liệu và tạo điều kiện thuận lợi cho sinh viên, tutor cũng như các phòng ban trong quá trình sử dụng. Ngoài các chức năng cơ bản, hệ thống cũng có thể mở rộng với các tính năng nâng cao như: Ghép cặp tutor – sinh viên thông minh dựa trên AI, Xây dựng cộng đồng trực tuyến cho tutor và mentee, Tổ chức các chương trình hỗ trợ học thuật và phi học thuật, Cung cấp dịch vụ hỗ trợ học tập cá nhân hóa.



### 1.1.2 Relevant stakeholders

Stakeholders	Roles	Expectations
Tutor	<ul style="list-style-type: none"><li>- Đăng nhập qua HCMUT_SSO</li><li>- Xem hồ sơ cá nhân</li><li>- Thiết lập lịch rảnh</li><li>- Mở/Tạo/Hủy lịch hẹn với sinh viên</li><li>- Quản lý buổi học trực tuyến/trực tiếp</li><li>- Theo dõi và ghi nhận tiến độ học tập sinh viên</li><li>- Truy cập, chia sẻ tài liệu qua HCMUT_LIBRARY</li><li>- Nhận phản hồi từ sinh viên</li></ul>	<ul style="list-style-type: none"><li>- Quản lý hồ sơ cá nhân và chuyên môn dễ dàng</li><li>- Được bảo mật thông tin</li><li>- Lịch rảnh dễ quản lý, linh hoạt</li><li>- Hệ thống nhắc lịch tự động</li><li>- Công cụ theo dõi tiến độ, xuất báo cáo thuận tiện</li><li>- Nhận phản hồi minh bạch sau buổi học</li></ul>
Student	<ul style="list-style-type: none"><li>- Đăng nhập bằng HCMUT_SSO</li><li>- Đăng ký/Hủy đăng ký tham gia chương trình Tutor</li><li>- Xem/cập nhật hồ sơ, nhu cầu học tập</li><li>- Tìm/chọn tutor hoặc nhận gợi ý</li><li>- Đăng ký/Hủy đăng ký lịch hẹn với tutor</li><li>- Tham gia buổi học trực tiếp/trực tuyến</li><li>- Truy cập tài liệu qua HCMUT_LIBRARY</li><li>- Gửi phản hồi, đánh giá chất lượng buổi học</li><li>- Theo dõi tiến độ học tập</li></ul>	<ul style="list-style-type: none"><li>- Đăng nhập an toàn, đồng bộ</li><li>- Giao diện dễ dùng, thao tác đơn giản</li><li>- Công cụ tìm/gợi ý tutor phù hợp</li><li>- Quản lý lịch hẹn dễ dàng, nhắc lịch tự động</li><li>- Học online/offline linh hoạt</li><li>- Truy cập học liệu chính thống</li><li>- Được phản hồi/đánh giá tutor minh bạch</li><li>- Công cụ theo dõi tiến độ, kết quả cuối khóa minh bạch</li></ul>



Phòng Đào tạo	<ul style="list-style-type: none"><li>- Quản lý học vụ toàn trường</li><li>- Phân bổ nguồn lực</li><li>- Nhận báo cáo tổng quan chương trình</li></ul>	<ul style="list-style-type: none"><li>- Báo cáo tổng hợp nhiều định dạng</li><li>- Thống kê phân bổ tutor, phòng học</li><li>- Phân tích dữ liệu nhiều học kỳ để nhận diện xu hướng môn cần hỗ trợ</li></ul>
Phòng Công tác Sinh viên	<ul style="list-style-type: none"><li>- Theo dõi kết quả tham gia của sinh viên</li><li>- Dùng dữ liệu cộng điểm rèn luyện, xét học bỗng</li></ul>	<ul style="list-style-type: none"><li>- Báo cáo chi tiết về kết quả tham gia</li><li>- Dữ liệu minh bạch, đáng tin cậy</li><li>- Tích hợp sẵn, giảm thao tác thủ công</li></ul>
Khoa / Bộ môn	<ul style="list-style-type: none"><li>- Giám sát chất lượng học tập qua dữ liệu đánh giá</li><li>- Kiểm tra hiệu quả chương trình theo môn/chuyên ngành</li><li>- Đề xuất cải tiến chương trình</li></ul>	<ul style="list-style-type: none"><li>- Dữ liệu thống kê rõ ràng theo từng môn học</li><li>- Phân tích tiến bộ sinh viên</li><li>- Căn cứ khách quan để ra quyết định</li></ul>
HCMUT _ DATACORE	<ul style="list-style-type: none"><li>- Cung cấp, đồng bộ dữ liệu cá nhân (họ tên, MSSV, khoa, email, tình trạng học tập/giảng dạy)</li></ul>	<ul style="list-style-type: none"><li>- Dữ liệu chính xác, nhất quán</li><li>- Tích hợp, tránh nhập liệu thủ công</li><li>- Bảo mật dữ liệu theo quy định</li></ul>
HCMUT _ SSO	<ul style="list-style-type: none"><li>- Xác thực tập trung</li><li>- Cấp quyền truy cập theo vai trò</li></ul>	<ul style="list-style-type: none"><li>- Đăng nhập một lần (SSO)</li><li>- Bảo mật thông tin cá nhân</li><li>- Phân quyền chính xác, hạn chế truy cập sai vai trò</li></ul>
HCMUT _ Library	<ul style="list-style-type: none"><li>- Cung cấp học liệu, tài nguyên tích hợp hệ thống</li></ul>	<ul style="list-style-type: none"><li>- Đảm bảo tính chính thống và bản quyền</li><li>- Đồng bộ học liệu với môn học/tutor</li><li>- Cho phép truy cập/chia sẻ học liệu</li><li>- Có thống kê truy cập để cải thiện dịch vụ</li></ul>



Coordinator	<ul style="list-style-type: none"><li>- Quản lý danh sách tutor/sinh viên</li><li>- Điều phối khung chương trình chung</li><li>- Gửi báo cáo đến các phòng ban</li><li>- Thu thập phản hồi, đề xuất cải tiến</li></ul>	<ul style="list-style-type: none"><li>- Quản lý tập trung giáo trình, slide, đề thi</li><li>- Công cụ báo cáo, thống kê giảng dạy</li><li>- Đảm bảo tính nhất quán giữa các tutor</li><li>- Tiết kiệm thời gian, nâng cao chất lượng môn học</li></ul>
-------------	--	--

### 1.1.3 Mục tiêu

Dự án Tutor/Mentor tại HCMUT được khởi xướng với mục tiêu xây dựng một hệ thống phần mềm hiện đại, đồng bộ và thân thiện, đóng vai trò là nền tảng trung tâm để quản lý, điều phối và nâng cao chất lượng chương trình. Thay vì vận hành rắc rối và thủ công, hệ thống này hướng đến số hóa toàn diện, từ khâu quản lý hồ sơ, đăng ký tham gia cho đến quản lý lịch hẹn, phản hồi và đánh giá. Qua đó, không chỉ giảm tải đáng kể khối lượng công việc hành chính cho cán bộ mà còn mang lại trải nghiệm thuận tiện, minh bạch và chủ động hơn cho sinh viên cũng như đội ngũ Tutor.

Bên cạnh chức năng quản lý cốt lõi, dự án còn nhấn mạnh việc khai thác dữ liệu thông minh để hỗ trợ các đơn vị chức năng như Phòng Đào tạo, Phòng Công tác Sinh viên theo dõi tiến độ, đánh giá hiệu quả và tối ưu hóa nguồn lực. Các dữ liệu này cũng trở thành căn cứ quan trọng để cộng điểm rèn luyện, xét học bổng hay triển khai chính sách hỗ trợ sinh viên phù hợp. Điều này không chỉ giúp cải thiện chất lượng hoạt động Tutor/Mentor mà còn nâng cao toàn diện trải nghiệm học tập và phúc lợi cho người học.

Trên nền tảng dữ liệu thống nhất và công nghệ hiện đại, dự án còn tăng cường việc phát triển một môi trường học tập thông minh và năng động hơn thông qua các tính năng mở rộng như AI gợi ý ghép cặp, cộng đồng học tập trực tuyến, tích hợp các chương trình học thuật và phi học thuật cùng khả năng hỗ trợ cá nhân hóa học tập. Những tính năng này sẽ giúp sinh viên chủ động hơn trong việc học và kết nối. Đây cũng là nền tảng để nhà trường tối ưu và phát triển thêm các dịch vụ học tập số, hướng tới một hệ sinh thái giáo dục số bền vững và linh hoạt, đáp ứng nhu cầu ngày càng đa dạng của người học.

Như vậy, mục tiêu của dự án không chỉ dừng lại ở việc tạo ra một công cụ quản lý đơn thuần



mà còn là một bước tiến quan trọng trong chiến lược chuyển đổi số của HCMUT. Hệ thống sẽ trở thành cầu nối giữa sinh viên, tutor và nhà trường, thúc đẩy tương tác, lan tỏa tri thức và định hình một môi trường học tập hiện đại, sáng tạo và lấy người học làm trung tâm.

#### 1.1.4 Phạm vi

Trong phạm vi (*scope*) của dự án, hệ thống Tutor Support sẽ tập trung vào các chức năng cốt lõi để hỗ trợ hiệu quả cho cả sinh viên và tutor. Cụ thể, hệ thống cho phép quản lý hồ sơ tutor và sinh viên thông qua việc đồng bộ dữ liệu cá nhân cơ bản (họ tên, MSSV/Mã cán bộ, khoa/chuyên ngành, email học vụ...) từ HCMUT \_ DATACORE, giúp giảm thiểu nhập liệu thủ công và đảm bảo tính chính xác. Sinh viên có thể đăng nhập bằng tài khoản SSO để đăng ký tham gia chương trình. Ngoài ra, hệ thống hỗ trợ việc ghép cặp tutor – sinh viên theo hai cách: sinh viên tự chọn hoặc được gợi ý dựa trên nhu cầu chuyên môn cần hỗ trợ và lịch rảnh. Sau khi ghép cặp, sinh viên và tutor có thể chủ động đặt lịch, hủy hoặc đổi lịch, đồng thời hệ thống sẽ tự động gửi thông báo và nhắc nhở để cả hai bên có thể nhận được những thay đổi mới nhất. Các buổi gặp có thể diễn ra trực tiếp hoặc trực tuyến. Bên cạnh đó, hệ thống cho phép sinh viên gửi phản hồi, đánh giá chất lượng buổi học, còn tutor có thể ghi nhận tiến bộ của sinh viên; các dữ liệu này sẽ được tổng hợp thành báo cáo cho khoa/bộ môn và các phòng ban liên quan nhằm theo dõi hiệu quả của chương trình.

Về ranh giới (*boundary*) với các hệ thống khác, Tutor Support sẽ giao tiếp trực tiếp với HCMUT \_ SSO để thực hiện xác thực đăng nhập, đồng bộ dữ liệu cá nhân từ HCMUT \_ DATACORE, và kết nối với HCMUT \_ LIBRARY để sinh viên và tutor có thể truy cập vào nguồn tài liệu học thuật. Đây là các tích hợp nhằm đảm bảo tính đồng bộ, chính thống và an toàn dữ liệu. Dự án cũng xác định rõ những chức năng nằm ngoài phạm vi: không xử lý các vấn đề liên quan đến thanh toán học phí, không thay thế hoàn toàn hệ thống LMS của trường, không can thiệp vào việc quản lý chương trình giảng dạy, lịch học hay điểm số. Tutor Support chỉ giữ vai trò hỗ trợ bổ sung, đồng thời tạo giá trị riêng trong việc kết nối sinh viên và tutor một cách thuận tiện hơn.



## 1.2 Yêu cầu chức năng

### 1.2.1 Đối với sinh viên

- Đăng nhập:
  - Hệ thống phải cho phép sinh viên đăng nhập bằng tài khoản đã được cấp thông qua HCMUT\_SSO.
  - Hệ thống phải kiểm tra tính hợp lệ của tên đăng nhập và mật khẩu.
  - Sau khi xác thực thành công, hệ thống phải gán quyền “sinh viên” để truy cập các chức năng phù hợp.
  - Nếu xác thực thất bại, hệ thống phải hiển thị thông báo lỗi và yêu cầu sinh viên nhập lại thông tin.
- Đăng xuất:

Hệ thống phải cho phép sinh viên đăng xuất để kết thúc phiên làm việc hiện tại. Sau khi đăng xuất, hệ thống phải đưa sinh viên về trang đăng nhập và vô hiệu hóa quyền truy cập còn lại.
- Xem hồ sơ cá nhân:

Hệ thống phải cho phép sinh viên xem thông tin cá nhân của mình được đồng bộ từ HCMUT\_DATACORE, bao gồm họ tên, mã số sinh viên (MSSV), khoa, chuyên ngành, email học vụ và các thông tin liên quan đến học tập.
- Xem tài liệu:
  - Hệ thống phải cho phép sinh viên truy cập và xem các tài liệu học tập được cung cấp từ HCMUT\_LIBRARY.
  - Hệ thống phải cho phép sinh viên xem trực tuyến hoặc tải xuống tài liệu để sử dụng.
- Đăng ký Tutor:

Hệ thống phải cho phép sinh viên chọn tutor phù hợp dựa trên chuyên môn, lĩnh vực hỗ trợ hoặc gợi ý thông minh. Sau khi sinh viên gửi yêu cầu đăng ký, hệ thống phải lưu yêu cầu ở trạng thái “Chờ duyệt” trong 12 giờ và cho phép sinh viên hủy yêu cầu trong thời gian này.



- Đăng ký buổi tư vấn:

Hệ thống phải cho phép sinh viên chọn và đăng ký buổi tư vấn còn chỗ trống do tutor tổ chức.

- Hủy đăng ký buổi gặp mặt:

Hệ thống phải cho phép sinh viên huỷ các buổi tư vấn và buổi hẹn đã đăng ký trong thời gian cho phép (trước khi buổi gặp mặt bắt đầu).

- Đặt lịch hẹn:

Hệ thống phải cho phép sinh viên đặt lịch hẹn riêng với tutor dựa trên lịch rảnh do tutor cung cấp và nhập nội dung cần được hỗ trợ cho buổi hẹn.

- Phản hồi chất lượng buổi học:

- Hệ thống phải cung cấp form phản hồi để sinh viên đánh giá chất lượng buổi học sau mỗi buổi tư vấn.

- Hệ thống phải cho phép sinh viên xem lại lịch sử phản hồi của mình.

- Form phản hồi phải bao gồm:

- \* Điểm số đánh giá (1 đến 5).

- \* Nhận xét ngắn gọn về nội dung buổi học, thái độ và khả năng truyền đạt của tutor.

### 1.2.2 Đối với Tutor

- Đăng nhập:

- Hệ thống phải cho phép tutor đăng nhập bằng tài khoản HCMUT\_SSO.

- Hệ thống phải kiểm tra tính hợp lệ của tên đăng nhập và mật khẩu.

- Sau khi xác thực thành công, hệ thống phải gán quyền “tutor” để truy cập các chức năng tương ứng.

- Nếu xác thực thất bại, hệ thống phải hiển thị thông báo lỗi và yêu cầu tutor nhập lại thông tin.

- Đăng xuất: Hệ thống phải cho phép tutor đăng xuất để kết thúc phiên làm việc hiện tại.

Sau khi đăng xuất, hệ thống phải đưa tutor về trang đăng nhập và vô hiệu hóa quyền truy cập còn lại.



- Xem hồ sơ cá nhân: Hệ thống phải cho phép tutor xem thông tin cá nhân của mình được đồng bộ từ HCMUT\_DATACORE, bao gồm họ tên, mã định danh, email, số điện thoại và thông tin liên hệ khác.
- Thiết lập lịch rảnh:  
Hệ thống phải cho phép tutor thiết lập lịch rảnh để sinh viên tham khảo khi đặt lịch hẹn. Lịch rảnh bao gồm thông tin về ngày, giờ. Khi tutor thay đổi hoặc hủy lịch rảnh, hệ thống phải tự động cập nhật lại.
- Tạo buổi tư vấn :  
Hệ thống cho phép tutor tạo các buổi tư vấn với thông tin về ngày, giờ, chủ đề và hình thức (online hoặc offline). Sau khi tạo, hệ thống phải gửi thông báo buổi tư vấn đến các sinh viên đã chọn tutor.
- Xử lý yêu cầu đặt lịch hẹn: Hệ thống phải cho phép tutor xem các yêu cầu đặt buổi gặp từ sinh viên và duyệt hoặc từ chối từng yêu cầu.
- Hủy buổi gặp mặt:  
Hệ thống cho phép tutor hủy các buổi tư vấn đã tạo, buổi hẹn đã duyệt và nhập lý do .
- Xem tài liệu:
  - Hệ thống phải cho phép tutor truy cập và xem các tài liệu học tập được cung cấp từ HCMUT\_LIBRARY.
  - Hệ thống phải cho phép tutor xem trực tuyến hoặc tải xuống tài liệu để sử dụng.
- Đăng tải tài liệu:  
Hệ thống phải cho phép tutor đăng tải tài liệu học tập với thông tin cơ bản (file, tiêu đề, mô tả, loại tài liệu). Tài liệu được lưu ở trạng thái “Chờ duyệt” và chỉ hiển thị cho sinh viên sau khi được quản trị viên phê duyệt.
- Xóa tài liệu:  
Hệ thống phải cho phép tutor xóa các tài liệu đã đăng tải nếu không còn sử dụng hoặc muốn thay thế.
- Ghi nhận tiến độ học tập của sinh viên:
  - Hệ thống phải cho phép tutor ghi nhận tiến độ và phản hồi về quá trình học tập của sinh viên.



- Hệ thống phải cho phép lưu trữ dữ liệu này để sử dụng trong các báo cáo thống kê.

### 1.2.3 Đối với Điều phối viên – Coordinator

- Quản lý thông tin tutor và sinh viên:

Hệ thống phải cho phép điều phối viên xem và quản lý thông tin của các tutor và sinh viên trong chương trình. Điều này bao gồm việc theo dõi các hồ sơ cá nhân, chuyên môn, và nhu cầu hỗ trợ của sinh viên.

- Hệ thống phải cho phép điều phối viên đăng nhập/đăng xuất: Gọi API SSO, gán quyền “Coordinator”

- Gửi báo cáo cho các phòng ban:

Hệ thống phải cho phép điều phối viên gửi các báo cáo đã được hệ thống tạo sẵn tới các phòng ban và khoa/bộ môn. Điều phối viên có thể chọn báo cáo, định dạng xuất báo cáo, và phương thức gửi.

- Điều phối khung chương trình chung cho các tutor:

Hệ thống phải cho phép điều phối viên kiểm soát khung chương trình chung cho các tutor, từ đó các tutor thực hiện và đảm bảo tính đồng bộ trong các buổi hẹn. Các buổi tư vấn diễn ra theo một chương trình đã được chuẩn bị sẵn, với các chủ đề, thời gian và tài liệu học tập được thống nhất.

### 1.2.4 Đối với Ban quản lý (Phòng Đào tạo, Phòng Công tác Sinh viên, Khoa/Bộ môn)

- Hệ thống phải cung cấp báo cáo tổng quan về chương trình: số lượng Tutor/sinh viên đăng ký tham gia, phòng học sử dụng, số buổi đã tổ chức, tỷ lệ tham gia/hoàn thành,...
- Hệ thống phải cho phép xuất báo cáo (PDF/Excel/CSV) để lưu trữ hoặc tích hợp hệ thống khác
- Hệ thống cho phép gửi báo cáo định kỳ tự động theo tuần/tháng hoặc theo chu kỳ do quản trị viên cấu hình.
- Hệ thống cho phép gửi báo cáo thủ công khi cần: điều phối viên là người chính, quản trị viên có thể can thiệp khi cần.



- Hệ thống cho phép các phòng ban nhận báo cáo qua email mà không cần tài khoản đăng nhập hệ thống Tutor.

#### 1.2.5 Đối với HCMUT \_ DATACORE

- Hệ thống phải cho phép cung cấp dữ liệu cá nhân của sinh viên và tutor (họ tên, MSSV, mã tutor, khoa/chuyên ngành, email học vụ, trạng thái học tập/giảng dạy, ...) cho các hệ thống khác để phục vụ quá trình quản lý và học tập.
- Hệ thống phải cho phép đồng bộ dữ liệu cá nhân với các hệ thống liên quan để đảm bảo tính chính xác và nhất quán.
- Hệ thống phải cho phép chia sẻ dữ liệu người dùng, tình trạng học tập và điểm số giữa các hệ thống HCMUT (DATACORE, SSO và Library) nhằm hỗ trợ tra cứu và quản lý.

#### 1.2.6 Đối với HCMUT \_ SSO

- Hệ thống phải cho phép người dùng đăng nhập tập trung một lần để truy cập các ứng dụng và dịch vụ khác của HCMUT mà không cần đăng nhập lại.
- Hệ thống phải cho phép người dùng đổi mật khẩu và khôi phục tài khoản để đảm bảo an toàn và bảo mật thông tin cá nhân.
- Hệ thống phải cho phép cấp quyền truy cập theo vai trò (sinh viên, tutor, điều phối viên, ...) để giới hạn phạm vi chức năng phù hợp với từng loại người dùng.

#### 1.2.7 Đối với HCMUT \_ Library

- Hệ thống phải cho phép liên kết các khóa học với tài liệu số (giáo trình, bài giảng, tài liệu tham khảo, ...) để sinh viên và tutor có thể truy cập hoặc tải về phục vụ học tập.
- Hệ thống phải cho phép tutor đăng tải và xóa tài liệu học tập cho sinh viên thông qua hệ thống thư viện số.
- Hệ thống phải cho phép ghi nhận và thống kê lượt truy cập hoặc tải tài liệu để hỗ trợ công tác quản lý và cải thiện chất lượng học liệu.



### 1.3 Yêu cầu phi chức năng (Non-functional Requirements)

#### 1.3.1 Hiệu năng và khả năng mở rộng (Performance & Scalability)

- Phản hồi hệ thống:** Hệ thống phải cung cấp phản hồi trong vòng 3 giây đối với các thao tác tiêu chuẩn (ví dụ: đăng nhập, xem tài liệu, xem hồ sơ cá nhân,...).
- Xử lý đồng thời:** Hệ thống phải hỗ trợ ít nhất 5,000 sinh viên sử dụng đồng thời mà không gián đoạn dịch vụ.
- Thời gian phản hồi các thao tác nặng:** Các thao tác như ghép cặp AI, tạo báo cáo, tải tài liệu phải hoàn thành trong  $\leq 2$  giây cho 95% các yêu cầu.
- Khả năng mở rộng:** Hệ thống có thể mở rộng để phục vụ tối đa 8,000 sinh viên mà không gián đoạn dịch vụ.

#### 1.3.2 Bảo mật và tuân thủ pháp lý (Security & Compliance)

- Mã hóa dữ liệu:** 100% dữ liệu nhạy cảm (thông tin cá nhân, điểm số, nhật ký học tập) phải được mã hóa theo TLS 1.3 khi truyền tải và AES-256 khi lưu trữ.
- Xác thực người dùng:** Tích hợp HCMUT\_SSO, chỉ cho phép đăng nhập thông qua Single Sign-On (SSO). Vai trò người dùng được đồng bộ tự động từ hệ thống tập trung.
- Phân quyền truy cập:** Kiểm soát truy cập theo vai trò (RBAC), quyền truy cập được kiểm tra định kỳ 6 tháng. Chỉ cho phép người dùng truy cập dữ liệu và tài liệu mà họ được phép.
- Bảo vệ chống tấn công:** Chống DDoS, SQL injection, XSS; kiểm tra bảo mật định kỳ hàng quý.
- Tuân thủ pháp lý:** Hệ thống phải tuân thủ Luật An ninh Mạng 2018, GDPR và các quy định địa phương liên quan đến dữ liệu cá nhân và giao dịch học thuật.

#### 1.3.3 Tính khả dụng và khôi phục (Availability & Disaster Recovery)

- Tính khả dụng:** Hệ thống phải đảm bảo uptime  $\geq 99\%$ , tương đương không quá 88 giờ downtime/năm, đặc biệt trong các giai đoạn cao điểm như kỳ thi hoặc thời hạn nộp bài.



- **Khôi phục sau sự cố:** Thời gian khôi phục sau sự cố (RTO)  $\leq$  3 giờ, phục hồi dữ liệu trong vòng 24 giờ (RPO  $\leq$  24 giờ).
- **Bảo trì:** Thời gian bảo trì  $\leq$  5 giờ/tháng, thực hiện trong giờ thấp điểm (0h00–1h00), thông báo trước ít nhất 24 giờ.

#### 1.3.4 Khả năng tương thích (Compatibility)

- Hệ thống tương thích với các trình duyệt phổ biến: Chrome, Firefox, Edge (phiên bản mới nhất).
- Hỗ trợ cả thiết bị máy tính và thiết bị di động.
- Tích hợp với HCMUT\_DATACORE và HCMUT\_LIBRARY để đảm bảo dữ liệu đồng bộ và chính xác.

#### 1.3.5 Khả năng sử dụng (Usability)

- Giao diện trực quan, song ngữ (Tiếng Việt/Anh), người dùng mới có thể nắm bắt cách sử dụng cơ bản trong khoảng thời gian 10 phút.
- Thông báo lỗi rõ ràng, hiển thị song ngữ và kèm hướng dẫn khắc phục.

#### 1.3.6 Khả năng bảo trì (Maintainability)

- Hệ thống module hóa (ví dụ: module quản lý lịch, module tải tài liệu, module ghép cắp AI) để dễ dàng cập nhật mà không ảnh hưởng toàn hệ thống.
- Tài liệu kỹ thuật đầy đủ, đảm bảo lập trình viên tương lai có thể hiểu và chỉnh sửa hệ thống dễ dàng.

#### 1.3.7 Hỗ trợ kỹ thuật (Technical Support)

- 100% yêu cầu hỗ trợ kỹ thuật phải được phản hồi trong vòng 2 giờ làm việc.
- Vấn đề phải được giải quyết trong vòng 24 giờ từ khi tiếp nhận.



### 1.3.8 Độ tin cậy (Reliability)

- **Tỷ lệ lỗi của người dùng:** Khi thực hiện các thao tác trong hệ thống (ví dụ: đăng nhập, ghép cặp Tutor-Sinh viên, tải tài liệu), tỷ lệ lỗi do người dùng không được vượt quá 5%.
- **Chống lỗi tạm thời (Fault Tolerance):** Hệ thống phải có cơ chế tự động phục hồi khi gặp các lỗi tạm thời, ví dụ: mất kết nối với HCMUT\_DATACORE hoặc lỗi tải tài liệu.
- **Thông báo sự cố:** Khi sự cố kéo dài, hệ thống phải thông báo rõ ràng cho người dùng và hướng dẫn các bước khắc phục (ví dụ: thử lại thao tác hoặc liên hệ bộ phận hỗ trợ).



## 1.4 Use-case diagram

### 1.4.1 Các actor của hệ thống

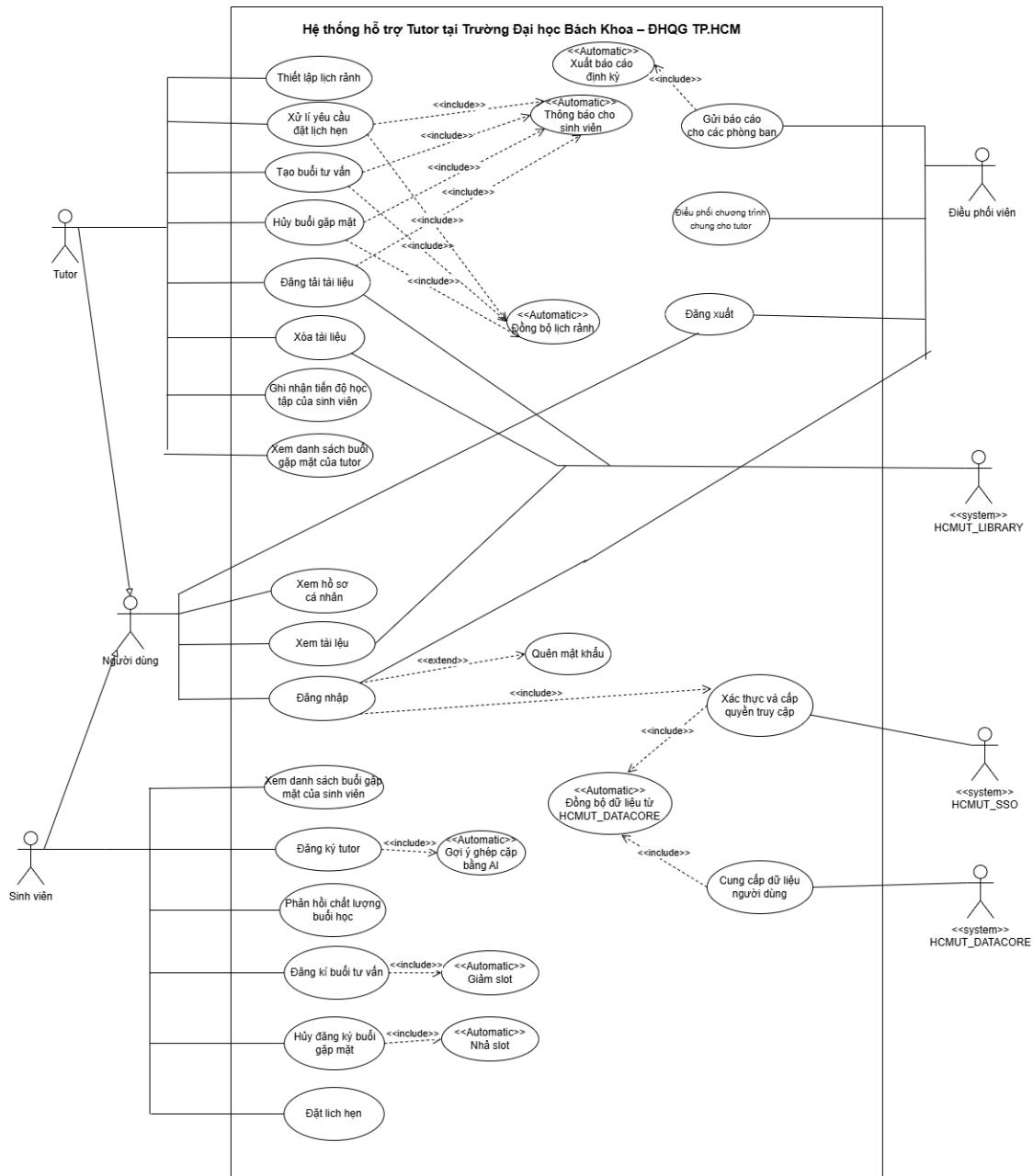
STT	Actor	Mô tả
1	Tutor	Người hỗ trợ, tạo buổi tư vấn, xử lý lịch hẹn, đăng/xóa tài liệu và ghi nhận tiến độ học tập
2	Người dùng	Người sử dụng hệ thống (có thể xem tài liệu, hồ sơ cá nhân, đăng nhập/đăng xuất)
3	Sinh viên	Đăng ký tutor, đăng ký/hủy buổi tư vấn, phản hồi chất lượng buổi học
4	Điều phối viên	Quản lý, gửi báo cáo, điều phối chương trình tutor chung
5	HCMUT_LIBRARY	Hệ thống thư viện trực tuyến
6	HCMUT_SSO	Hệ thống xác thực và cấp quyền truy cập
7	HCMUT_DATACORE	Hệ thống cung cấp và đồng bộ dữ liệu người dùng



#### 1.4.2 Các nhánh use case chính

STT	Use Case Name	Mô tả
1	Dăng nhập	Người dùng đăng nhập vào hệ thống
2	Dăng xuất	Người dùng thoát khỏi hệ thống
3	Xem tài liệu	Người dùng/tutor xem các tài liệu đã được tải lên
4	Xem hồ sơ cá nhân	Người dùng xem thông tin cá nhân
5	Điều phối chương trình chung cho Tutor	Điều phối viên điều phối khung chương trình chung cho Tutor
6	Gửi báo cáo cho các phòng ban	Điều phối viên gửi báo cáo đến các phòng ban
7	Dăng ký tutor	Sinh viên đăng ký tutor
8	Đặt lịch hẹn	Sinh viên gửi yêu cầu đặt lịch với tutor
9	Phản hồi chất lượng buổi học	Sinh viên phản hồi sau buổi học
10	Dăng ký buổi tư vấn	Sinh viên đặt lịch buổi tư vấn
11	Hủy đăng ký buổi gặp mặt	Tutor hủy lịch tư vấn đã tạo
12	Tạo buổi tư vấn	Tutor tạo lịch tư vấn cho sinh viên
13	Hủy buổi gặp	Tutor hủy lịch buổi tư vấn đã lên lịch
14	Ghi nhận tiến độ học tập của sinh viên	Tutor ghi nhận tình hình học tập của sinh viên
15	Thiết lập lịch rảnh	Tutor thiết lập khung giờ rảnh để sinh viên có thể đặt lịch
16	Dăng tải tài liệu	Tutor đăng tài liệu hỗ trợ sinh viên
17	Xóa tài liệu	Tutor xóa tài liệu đã tải lên
18	Xử lý yêu cầu đặt lịch hẹn	Tutor xử lý và phản hồi các yêu cầu hẹn từ sinh viên

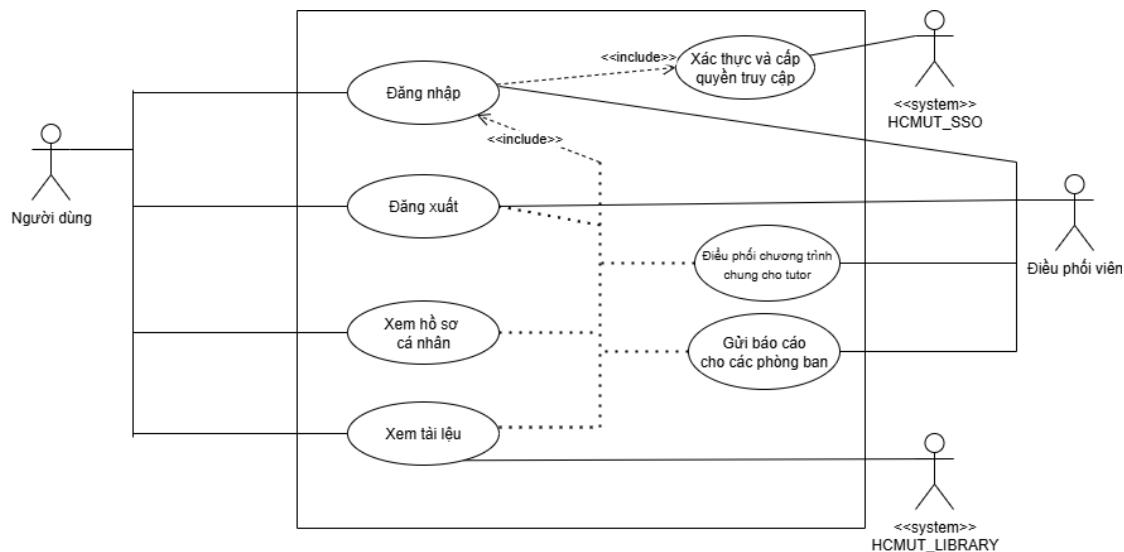
### 1.4.3 Use-case ch0 toàn bộ hệ thống



Hình 1: Use-case cho toàn bộ hệ thống

Xem rõ hơn [tại đây](#).

## 1.5 Use-case detail/scenario



Hình 2: Nhóm chức năng của người dùng

### 1.5.1 Đăng nhập

<b>Use Case ID</b>	UC-01
<b>Use Case Name</b>	Đăng nhập
<b>Actor(s)</b>	Người dùng (Sinh viên / Tutor) hoặc Điều phối viên
<b>Description</b>	User đăng nhập vào hệ thống thông qua dịch vụ xác thực tập trung HCMUT_SSO để truy cập các chức năng của hệ thống.
<b>Trigger</b>	User truy cập vào hệ thống Tutor Support System và chưa đăng nhập.
<b>Pre-Condition(s)</b>	User có tài khoản hợp lệ trong hệ thống HCMUT_SSO. Hệ thống HCMUT_SSO đang hoạt động bình thường.
<b>Post-Condition(s)</b>	User được xác thực thành công và truy cập vào hệ thống với vai trò tương ứng. Phiên làm việc của user được khởi tạo. Thông tin cá nhân cơ bản được đồng bộ từ HCMUT_DATACORE.



<b>Normal Flow</b>	<ol style="list-style-type: none"><li>User truy cập vào trang chủ của hệ thống Tutor Support System.</li><li>Hệ thống hiển thị trang đăng nhập với nút "Đăng nhập qua HC-MUT_SSO".</li><li>User nhấp vào nút "Đăng nhập qua HCMUT_SSO".</li><li>Hệ thống chuyển hướng user đến trang đăng nhập của HCMUT_SSO.</li><li>User nhập thông tin đăng nhập (Tài khoản BKNetID và mật khẩu).</li><li>HCMUT_SSO xác thực thông tin đăng nhập.</li><li>Nếu hợp lệ, HCMUT_SSO trả về token xác thực cho hệ thống.</li><li>Hệ thống sử dụng token để truy vấn thông tin cơ bản từ HC-MUT_DATACORE (họ tên, mã định danh, khoa/chuyên ngành, email, vai trò, trạng thái tài khoản).</li><li>Hệ thống kiểm tra vai trò và trạng thái tài khoản để xác định quyền truy cập.</li><li>Hệ thống phân quyền và chuyển hướng user đến trang chủ phù hợp với vai trò (Sinh viên / Tutor / Điều phối viên).</li></ol>
<b>Alternative Flow</b>	<i>(Không có — chức năng quên mật khẩu đã được chuyển thành Use Case riêng.)</i>



<b>Exception Flow</b>	<p>E1. Thông tin đăng nhập không chính xác.</p> <p>E1.1 HCMUT_SSO hiển thị thông báo lỗi trực tiếp trên form đăng nhập.</p> <p>E1.2 Use case quay lại bước 5.</p> <p>E2. Tài khoản bị khóa / hết hạn / yêu cầu đổi mật khẩu.</p> <p>E2.1 HCMUT_SSO hiển thị thông báo lỗi tương ứng trên form đăng nhập.</p> <p>E2.2 Use case kết thúc không thành công tại SSO.</p> <p>E3. Không có quyền truy cập hệ thống.</p> <p>E3.1 Sau khi đăng nhập thành công, hệ thống kiểm tra vai trò và phát hiện user không thuộc nhóm được phép sử dụng hệ thống.</p> <p>E3.2 Tutor Support System hiển thị thông báo "Bạn không có quyền truy cập hệ thống."</p> <p>E3.3 Phiên đăng nhập không được tạo; use case kết thúc.</p> <p>E4. Lỗi kết nối đến HCMUT_SSO.</p> <p>E4.1 Hệ thống hiển thị thông báo "Không thể kết nối đến dịch vụ xác thực, vui lòng thử lại sau."</p> <p>E4.2 Use case kết thúc không thành công.</p> <p>E5. Lỗi đồng bộ dữ liệu từ HCMUT_DATACORE.</p> <p>E5.1 Hệ thống hiển thị thông báo "Không thể tải thông tin người dùng, vui lòng thử lại sau."</p> <p>E5.2 Hệ thống hủy phiên đăng nhập.</p> <p>E5.3 Use case kết thúc không thành công.</p>
-----------------------	---



### 1.5.2 Quên mật khẩu

Use Case ID	UC-01a
Use Case Name	Quên mật khẩu
Actor(s)	Người dùng (Sinh viên / Tutor / Điều phối viên)
Description	Cho phép người dùng khôi phục mật khẩu khi quên thông qua hệ thống HCMUT_SSO. Hệ thống kiểm tra tính hợp lệ của email, gửi liên kết đặt lại mật khẩu, và cho phép người dùng đặt lại mật khẩu mới an toàn.
Trigger	Người dùng chọn liên kết “Quên mật khẩu” tại trang đăng nhập.
Pre-Condition(s)	Người dùng đang ở giao diện đăng nhập của hệ thống. Hệ thống HCMUT_SSO và dịch vụ Email hoạt động bình thường.
Post-Condition(s)	Người dùng đã đặt lại mật khẩu thành công. Hệ thống ghi nhận nhật ký thay đổi mật khẩu.
Normal Flow	<ol style="list-style-type: none"><li>Người dùng chọn liên kết “Quên mật khẩu” trên form đăng nhập.</li><li>Hệ thống chuyển hướng đến <b>Form quên mật khẩu</b>.</li><li>Người dùng nhập email học vụ hoặc mã số sinh viên.</li><li>Hệ thống gửi yêu cầu kiểm tra email đến HCMUT_SSO.</li><li>HCMUT_SSO kiểm tra định dạng và sự tồn tại của email.</li><li>Nếu hợp lệ, HCMUT_SSO gửi yêu cầu đến <b>Email Service</b> để gửi email khôi phục mật khẩu.</li><li><b>Email Service</b> gửi email chứa liên kết đặt lại mật khẩu cho người dùng và phản hồi thành công.</li><li>Hệ thống hiển thị thông báo “Email đã được gửi đến địa chỉ của bạn”.</li><li>Người dùng mở email và nhấn vào liên kết đặt lại mật khẩu.</li><li>Trình duyệt chuyển đến <b>Form đổi mật khẩu</b>.</li><li>Người dùng nhập mật khẩu mới và xác nhận lại.</li><li><b>Form đổi mật khẩu</b> gửi yêu cầu đặt lại mật khẩu đến HC-MUT_SSO.</li><li>HCMUT_SSO kiểm tra chính sách bảo mật, cập nhật mật khẩu và phản hồi “Đặt lại mật khẩu thành công”.</li><li>Hệ thống hiển thị thông báo thành công và chuyển hướng người dùng về form đăng nhập.</li></ol>



<b>Alternative Flow</b>	A1. Email không hợp lệ (sai định dạng hoặc không tồn tại). A1.1 HCMUT_SSO phản hồi lỗi. A1.2 Form quên mật khẩu hiển thị thông báo “Email không hợp lệ”.  A2. Người dùng không nhận được email khôi phục. A2.1 Người dùng chọn “Gửi lại Email”. A2.2 Hệ thống gửi lại yêu cầu đến Email Service. A2.3 Hệ thống hiển thị thông báo “Đã gửi lại Email”.
<b>Exception Flow</b>	E1. Lỗi kết nối đến HCMUT_SSO hoặc Email Service. E1.1 Hệ thống hiển thị thông báo “Không thể kết nối đến máy chủ, vui lòng thử lại sau.” E1.2 Use Case kết thúc không thành công.  E2. Liên kết đặt lại mật khẩu đã hết hạn. E2.1 Hệ thống hiển thị thông báo “Liên kết không còn hiệu lực, vui lòng gửi lại yêu cầu mới.” E2.2 Người dùng được chuyển về Form quên mật khẩu.  E3. Mật khẩu mới không tuân thủ chính sách bảo mật (ví dụ: độ dài, ký tự đặc biệt). E3.1 HCMUT_SSO trả về thông báo lỗi “Mật khẩu không hợp lệ.” E3.2 Hệ thống yêu cầu người dùng nhập lại theo quy định.
<b>Include / Extend</b>	Extend: UC-01 Đăng nhập

### 1.5.3 Đăng xuất

<b>Use Case ID</b>	UC-02
<b>Use Case Name</b>	Dăng xuất
<b>Actor(s)</b>	Người dùng (Sinh viên / Tutor) và Điều phối viên
<b>Description</b>	User kết thúc phiên làm việc và đăng xuất khỏi hệ thống.
<b>Trigger</b>	User chọn chức năng "Đăng xuất" trong hệ thống.



<b>Pre-Condition(s)</b>	User đã đăng nhập thành công vào hệ thống. Phiên làm việc của user đang hoạt động.
<b>Post-Condition(s)</b>	Phiên làm việc của user được kết thúc. User được chuyển về trang đăng nhập. Token xác thực được vô hiệu hóa.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. User nhấp vào biểu tượng tài khoản hoặc menu người dùng.</li><li>2. Hệ thống hiển thị menu với tùy chọn "Đăng xuất".</li><li>3. User chọn "Đăng xuất".</li><li>4. Hệ thống hiển thị hộp thoại xác nhận "Bạn có chắc chắn muốn đăng xuất?".</li><li>5. User xác nhận đăng xuất.</li><li>6. Hệ thống vô hiệu hóa token xác thực hiện tại.</li><li>7. Hệ thống xóa thông tin phiên làm việc.</li><li>8. Hệ thống chuyển hướng user về trang đăng nhập.</li><li>9. Hệ thống hiển thị thông báo "Đăng xuất thành công".</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>5a. User chọn "Hủy" trong hộp thoại xác nhận.<ol style="list-style-type: none"><li>5a1. Hệ thống đóng hộp thoại xác nhận.</li><li>5a2. User tiếp tục sử dụng hệ thống.</li><li>5a3. Use case kết thúc.</li></ol></li></ol>
<b>Exception Flow</b>	<p>E1. Lỗi khi vô hiệu hóa token.</p> <p>E1.1 Hệ thống ghi log lỗi.</p> <p>E1.2 Hệ thống vẫn xóa thông tin phiên làm việc tại client.</p> <p>E1.3 Hệ thống chuyển hướng user về trang đăng nhập.</p> <p>E2. Phiên làm việc đã hết hạn trước khi đăng xuất.</p> <p>E2.1 Hệ thống phát hiện phiên đã hết hạn.</p> <p>E2.2 Hệ thống chuyển hướng user về trang đăng nhập.</p> <p>E2.3 Hệ thống hiển thị thông báo "Phiên làm việc đã hết hạn".</p>



#### 1.5.4 Xem tài liệu

<b>Use Case ID</b>	UC-03
<b>Use Case Name</b>	Xem tài liệu
<b>Actor(s)</b>	User, gồm Sinh viên và Tutor
<b>Description</b>	User truy cập và xem tài liệu học tập được đồng bộ từ hệ thống HC-MUT_LIBRARY.
<b>Trigger</b>	User truy cập vào trang tài liệu của môn học trong hệ thống.
<b>Pre-Condition(s)</b>	User đã đăng nhập thành công bằng HCMUT_SSO.
<b>Post-Condition(s)</b>	Tài liệu được hiển thị thành công cho user.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>User chọn chức năng “Danh sách tài liệu”.</li><li>Hệ thống gửi yêu cầu truy vấn tài liệu đến HCMUT_LIBRARY.</li><li>HCMUT_LIBRARY trả về danh sách các tài liệu khả dụng.</li><li>Hệ thống hiển thị danh sách tài liệu (tên, mô tả, loại file, ngày cập nhật).</li><li>User chọn một tài liệu cụ thể.</li><li>Hệ thống dẫn đến đường dẫn của tài liệu trên HCMUT_LIBRARY.</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>a. User chọn “Tìm kiếm”.<ol style="list-style-type: none"><li>User nhập các thông tin vào bộ lọc: tên tài liệu, từ khóa, tác giả, ngày cập nhật,...</li><li>User bấm “Tìm kiếm” hoặc phím Enter.</li><li>Hệ thống hiển thị kết quả phù hợp từ HCMUT_LIBRARY.</li></ol></li></ol>
<b>Exception Flow</b>	<ol style="list-style-type: none"><li>Lỗi kết nối tới HCMUT_LIBRARY.<ol style="list-style-type: none"><li>Hệ thống hiển thị thông báo “Không thể kết nối tới thư viện, vui lòng thử lại sau”.</li><li>Use case kết thúc không thành công.</li></ol></li><li>Không tìm thấy tài liệu khả dụng theo tiêu chí tìm kiếm.<ol style="list-style-type: none"><li>Hệ thống hiển thị thông báo “Không tìm thấy tài liệu phù hợp”.</li><li>Quay lại bước 5a2 để user nhập lại tiêu chí tìm kiếm.</li></ol></li></ol>



### 1.5.5 Xem hồ sơ cá nhân

<b>Use Case ID</b>	UC-04
<b>Use Case Name</b>	Xem hồ sơ cá nhân
<b>Actor(s)</b>	Sinh viên / Tutor / Điều phối viên
<b>Description</b>	User đăng nhập hệ thống và truy cập mục “Hồ sơ cá nhân” để xem thông tin đã lưu: thông tin cá nhân, liên hệ, tài khoản, lịch sử hoạt động liên quan.
<b>Trigger</b>	Người dùng muốn kiểm tra thông tin của chính mình (nhấn vào biểu tượng “Hồ sơ cá nhân”).
<b>Pre-Condition(s)</b>	User đã đăng nhập hợp lệ vào hệ thống.
<b>Post-Condition(s)</b>	Thông tin hồ sơ được hiển thị thành công.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>User truy cập menu “Hồ sơ cá nhân”.</li><li>Hệ thống truy vấn dữ liệu hồ sơ từ cơ sở dữ liệu từ HC-MUT_DATACORE.</li><li>Hệ thống hiển thị các thông tin: họ tên, mã định danh, email, số điện thoại, vai trò (SV/Tutor/Điều phối viên), và lịch sử hoạt động gần nhất.</li><li>Actor có thể chọn xem chi tiết từng phần (ví dụ: thông tin liên hệ, lịch sử đăng nhập, lịch hẹn).</li></ol>
<b>Exception Flow</b>	<p>E1. Phiên đăng nhập hết hạn. E1.1 Hệ thống từ chối hiển thị, chuyển đến màn hình đăng nhập.</p> <p>E2. Lỗi truy vấn cơ sở dữ liệu. E2.1 Hệ thống hiển thị thông báo “Không thể tải hồ sơ, vui lòng thử lại sau”.</p>



### 1.5.6 Điều phối khung chương trình chung cho các tutor

<b>Use Case ID</b>	UC-05
<b>Use Case Name</b>	Điều phối khung chương trình chung cho các tutor
<b>Actor(s)</b>	Điều phối viên (Coordinator)
<b>Description</b>	Điều phối viên soạn thảo và kiểm soát khung chương trình chung cho các tutor, từ đó hướng dẫn các tutor thực hiện và đảm bảo tính đồng bộ trong các buổi tư vấn.
<b>Trigger</b>	Điều phối viên truy cập vào mục “Quản lý khung chương trình”.
<b>Pre-Condition(s)</b>	Điều phối viên đã đăng nhập vào hệ thống. Điều phối viên có quyền quản trị và chỉnh sửa khung chương trình.
<b>Post-Condition(s)</b>	Khung chương trình được tạo và lưu thành công; hệ thống gửi thông báo đến các tutor.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>Điều phối viên mở “Quản lý khung chương trình”.</li><li>Hệ thống hiển thị danh sách khung chương trình và các tùy chọn.</li><li>Điều phối viên chọn “Tạo khung chương trình mới”.</li><li>Hệ thống hiển thị form với các trường (chủ đề, thời gian, tài liệu, yêu cầu học tập).</li><li>Điều phối viên điền thông tin và nhấn “Lưu”.</li><li>Hệ thống kiểm tra ràng buộc (các trường thông tin bắt buộc, trùng chủ đề) và lưu khung chương trình.</li><li>Hệ thống gửi thông báo cho các tutor về khung chương trình mới.</li></ol>



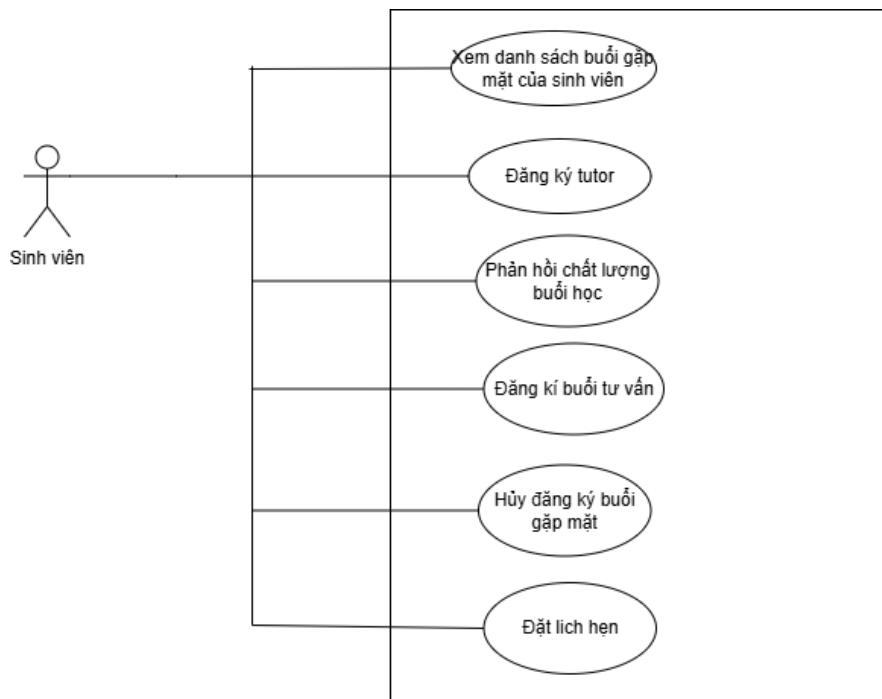
<b>Alternative Flow(s)</b>	<p>3a1. Điều phối viên chọn “Cập nhật khung chương trình cũ” và chọn một khung từ danh sách.</p> <p>3a2. Hệ thống hiển thị form đã điền sẵn.</p> <p>3a3. Điều phối viên chỉnh sửa thông tin và nhấn “Lưu”.</p> <p>3a4. Hệ thống kiểm tra ràng buộc và cập nhật khung chương trình.</p> <p>3a5. Hệ thống gửi thông báo cho các tutor về khung chương trình đã cập nhật.</p> <p>3b. Điều phối viên chọn “Hủy”; hệ thống quay về danh sách khung chương trình. Use case kết thúc không thay đổi dữ liệu.</p>
<b>Exception Flow</b>	<p>6a1. Hệ thống hiển thị lỗi (thiếu dữ liệu / chủ đề đã tồn tại).</p> <p>6a2. Hệ thống hiển thị lại form thông tin; Điều phối viên chỉnh sửa và quay lại Bước 5.</p> <p>6e1. Hệ thống thông báo: “Lưu chương trình thất bại, vui lòng thử lại sau”.</p> <p>6e2. Hệ thống ghi log lỗi, trạng thái dữ liệu chưa được lưu.</p> <p>6e3. Use case kết thúc không thành công.</p>

#### 1.5.7 Gửi báo cáo đến các phòng ban

<b>Use Case ID</b>	UC-06
<b>Use Case Name</b>	Gửi báo cáo cho các phòng ban
<b>Actor(s)</b>	Điều phối viên (Coordinator)
<b>Description</b>	Điều phối viên chọn báo cáo được hệ thống tạo và gửi đến các đơn vị liên quan (Phòng Đào tạo, Phòng CTSV, Khoa/Bộ môn) dưới định dạng phù hợp.
<b>Trigger</b>	Điều phối viên mở mục “Báo cáo” và chọn gửi báo cáo đã có cho các phòng ban.



<b>Pre-Condition(s)</b>	Điều phối viên đã đăng nhập và có quyền gửi báo cáo. Báo cáo đã được tạo/sinh sẵn (theo kỳ, theo môn, theo chương trình).
<b>Post-Condition(s)</b>	Báo cáo được gửi thành công tới phòng ban đã chọn. Hệ thống ghi nhận nhật ký gửi (thời gian, người gửi, nơi nhận, định dạng).
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>Điều phối viên vào mục “Báo cáo” và chọn một báo cáo có sẵn.</li><li>Hệ thống hiển thị thông tin báo cáo (tên, kỳ, phạm vi, bản xem trước).</li><li>Điều phối viên chọn nơi nhận: Phòng Đào tạo, Phòng CTSV, Khoa/Bộ môn (có thể chọn nhiều).</li><li>Điều phối viên chọn định dạng xuất (PDF/Excel) và kênh gửi (email/link nội bộ).</li><li>Điều phối viên nhấn “Gửi”.</li><li>Hệ thống tạo file (nếu cần), gửi tới nơi nhận đã chọn, ghi audit log.</li><li>Hệ thống hiển thị “Gửi báo cáo thành công” và cung cấp mã tra cứu.</li></ol>
<b>Alternative Flow</b>	<p>4a. Điều phối viên thêm ghi chú kèm báo cáo.</p> <p>4a1. Hệ thống đính kèm ghi chú vào email/thông điệp gửi.</p> <p>4a2. Quay lại bước 5.</p> <p>3b. Điều phối viên chọn lịch gửi định kỳ (hàng tháng/quý).</p> <p>3b1. Hệ thống lưu lịch và sẽ tự động gửi theo chu kỳ.</p>
<b>Exception Flow</b>	<p>2a. Không tồn tại báo cáo cho phạm vi đã chọn.</p> <p>2a1. Hệ thống thông báo “Chưa có báo cáo phù hợp”, gợi ý tạo báo cáo trước.</p> <p>2a2. Use case kết thúc.</p> <p>5a. Lỗi dịch vụ gửi (email/notification).</p> <p>5a1. Hệ thống hiển thị “Gửi thất bại”, ghi log lỗi và giữ báo cáo ở trạng thái chưa gửi.</p> <p>5a2. Điều phối viên có thể thử gửi lại hoặc tải file về để gửi thủ công.</p>



Hình 3: Nhóm chức năng của sinh viên

#### 1.5.8 Đăng ký Tutor

<b>Use Case ID</b>	UC-07
<b>Use Case Name</b>	Đăng ký Tutor
<b>Actor(s)</b>	Sinh viên
<b>Description</b>	Cho phép sinh viên đăng ký tham gia chương trình Tutor/Mentor và chọn Tutor phù hợp.
<b>Trigger</b>	Sinh viên truy cập hệ thống và chọn chức năng “Đăng ký Tutor”.
<b>Pre-Condition(s)</b>	<ul style="list-style-type: none"><li>- Sinh viên đăng nhập thành công.</li><li>- Chương trình Tutor đang mở đăng ký.</li></ul>
<b>Post-Condition(s)</b>	<ul style="list-style-type: none"><li>- Yêu cầu đăng ký tutor được lưu và chờ duyệt trong 12 giờ.</li><li>- Sau khi duyệt thành công, sinh viên sẽ được gắn cặp với tutor đã đăng ký. Nếu như hủy sẽ bắt đầu lại.</li></ul>



<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Sinh viên vào chức năng Đăng ký Tutor.</li><li>2. Hệ thống hiển thị giao diện cho phép sinh viên chọn lĩnh vực/môn học cần hỗ trợ.</li><li>3. Sinh viên điền thông tin và bấm tiếp theo.</li><li>4. Hệ thống hiển thị danh sách tutor phù hợp.</li><li>5. Sinh viên chọn Tutor hoặc chọn gợi ý Tutor thông minh.</li><li>6. Sinh viên bấm xác nhận đăng ký.</li><li>7. Hệ thống lưu yêu cầu đăng ký với trạng thái Chờ duyệt (12h).</li><li>8. Sau 12h, hệ thống xác nhận đăng ký thành công.</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>3a. Sinh viên không điền thông tin nhưng bấm tiếp theo.<ol style="list-style-type: none"><li>3a.1. Hệ thống thông báo lỗi yêu cầu nhập lại.</li><li>3a.2. Quay lại bước 2.</li></ol></li><li>4a. Không có tutor phù hợp.<ol style="list-style-type: none"><li>4a.1. Hệ thống thông báo và đưa lựa chọn “Đăng ký tutor khác”.</li><li>4a.2. Nếu sinh viên đồng ý, hệ thống hiển thị toàn bộ tutor còn slot đăng ký, quay về bước 5.</li><li>4a.3. Nếu sinh viên không đồng ý, hệ thống quay lại màn hình chính. Kết thúc usecase.</li></ol></li><li>6a. Sinh viên không xác nhận đăng ký, chọn kết thúc.<ol style="list-style-type: none"><li>6a.1. Hệ thống quay về màn hình chính. Kết thúc usecase.</li></ol></li><li>6b. Sinh viên không xác nhận đăng ký, chọn sửa đổi.<ol style="list-style-type: none"><li>6b.1. Quay về bước 2.</li></ol></li><li>8a. Sinh viên nhấn hủy đăng ký trong khi chờ duyệt.<ol style="list-style-type: none"><li>8a.1. Hệ thống hiển thị hủy thành công, đưa lựa chọn sinh viên có muốn đăng ký mới.</li><li>8a.2. Nếu sinh viên đồng ý quay về bước 2.</li><li>8a.3. Nếu sinh viên không đồng ý quay về màn hình chính, kết thúc usecase.</li></ol></li></ol>



<b>Exception Flow</b>	<ul style="list-style-type: none"><li>- Lỗi mạng trong quá trình đăng ký → Hệ thống hiển thị thông báo lỗi mạng, yêu cầu sinh viên thực hiện lại sau.</li><li>- Dợt đăng ký Tutor đã kết thúc → Hệ thống hiển thị thông báo “Dợt đăng ký Tutor đã kết thúc” → Kết thúc Use Case.</li><li>- Lỗi lưu dữ liệu đăng ký → Hệ thống hiển thị thông báo “Không lưu được dữ liệu, vui lòng thử lại” → Quay về bước 6 Normal Flow.</li></ul>
-----------------------	---

### 1.5.9 Đặt lịch hẹn

<b>Use Case ID</b>	UC-08
<b>Use Case Name</b>	Đặt lịch hẹn
<b>Actor(s)</b>	Sinh viên, Tutor
<b>Description</b>	Cho phép sinh viên đặt lịch hẹn riêng với tutor dựa trên lịch rảnh do tutor cung cấp và nhập nội dung cần được hỗ trợ cho buổi hẹn.
<b>Trigger</b>	Sinh viên chọn chức năng “Đặt lịch hẹn” trong hệ thống.
<b>Pre-Condition(s)</b>	<ul style="list-style-type: none"><li>- Sinh viên đã đăng nhập thành công.</li><li>- Sinh viên đã chọn tutor.</li><li>- Tutor đã cung cấp lịch rảnh.</li></ul>
<b>Post-Condition(s)</b>	<ul style="list-style-type: none"><li>- Yêu cầu đặt lịch hẹn được lưu ở trạng thái chờ duyệt.</li><li>- Tutor nhận thông báo về yêu cầu.</li></ul>
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Sinh viên truy cập chức năng “Đặt lịch hẹn”.</li><li>2. Hệ thống hiển thị lịch rảnh của tutor.</li><li>3. Sinh viên chọn khoảng thời gian phù hợp và nhập nội dung cần hỗ trợ.</li><li>4. Sinh viên xác nhận đặt lịch.</li><li>5. Hệ thống lưu yêu cầu và gửi thông báo cho tutor.</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>3a. Nếu sinh viên không chọn thời gian hoặc không nhập nội dung:<ol style="list-style-type: none"><li>3a1. Hệ thống thông báo lỗi và yêu cầu nhập lại.</li><li>3a2. Quay lại bước 3.</li></ol></li><li>3b. Nếu tutor không còn lịch rảnh:<ol style="list-style-type: none"><li>3b1. Hệ thống thông báo Tutor bận.</li><li>3b2. Hệ thống quay về màn hình chính. Kết thúc usecase.</li></ol></li></ol>



<b>Exception Flow</b>	- Lỗi hệ thống hoặc mất kết nối. Hệ thống thông báo lỗi và yêu cầu sinh viên thử lại.
-----------------------	---

### 1.5.10 Phản hồi chất lượng

<b>Use Case Name</b>	Phản hồi chất lượng
<b>Actor(s)</b>	Sinh viên (Student)
<b>Description</b>	Sinh viên muốn gửi phản hồi về chất lượng buổi học hoặc chất lượng hỗ trợ của Tutor.
<b>Trigger</b>	Sau khi kết thúc buổi học, sinh viên chọn mục “Phản hồi chất lượng”.
<b>Pre-Condition(s)</b>	Sinh viên đã đăng nhập vào hệ thống. Sinh viên đã tham gia ít nhất một buổi học với Tutor.
<b>Post-Condition(s)</b>	Phản hồi được lưu trữ thành công trong hệ thống. Hệ thống gửi thông báo đến Tutor và ghi nhận phản hồi cho báo cáo tổng hợp.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>Sinh viên truy cập mục “Phản hồi chất lượng”.</li><li>Hệ thống hiển thị danh sách các buổi học đã tham gia.</li><li>Sinh viên chọn buổi học cần phản hồi.</li><li>Hệ thống hiển thị form phản hồi (mức độ hài lòng, nội dung chi tiết).</li><li>Sinh viên nhập nội dung phản hồi và chọn mức độ đánh giá.</li><li>Sinh viên nhấn “Gửi”.</li><li>Hệ thống lưu phản hồi vào cơ sở dữ liệu.</li><li>Hệ thống gửi thông báo xác nhận cho sinh viên.</li></ol>



<b>Alternative Flow</b>	6a. Sinh viên không nhập nội dung phản hồi. 6a1. Hệ thống hiển thị thông báo: “Vui lòng nhập nội dung phản hồi trước khi gửi”. 6a2. Giao diện quay lại bước 4.  6b. Sinh viên chọn “Quay lại” thay vì gửi phản hồi. 6b1. Hệ thống hủy thao tác và không lưu thông tin. 6b2. Use case kết thúc sớm.
<b>Exception Flow</b>	7a. Sau khi sinh viên nhấn “Gửi” (bước 6), hệ thống cố gắng lưu dữ liệu nhưng gặp lỗi cơ sở dữ liệu (DB error). 7a1. Hệ thống hiển thị thông báo: “Gửi phản hồi thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật”. 7a2. Hệ thống ghi log lỗi và giữ trạng thái chưa gửi phản hồi. 7a3. Use case kết thúc không thành công.

#### 1.5.11 Đăng ký buổi tư vấn

<b>Use Case ID</b>	UC-10
<b>Use Case Name</b>	Đăng ký buổi tư vấn
<b>Description</b>	Cho phép sinh viên đăng ký buổi tư vấn với tutor đã chọn trước đó.
<b>Trigger</b>	Sinh viên chọn chức năng “Đăng ký buổi tư vấn” trong hệ thống.
<b>Primary Actor</b>	Sinh viên
<b>Secondary Actor</b>	Tutor
<b>Pre-condition</b>	- Sinh viên đã đăng nhập thành công. - Sinh viên đã chọn tutor. - Tutor đã mở lịch hẹn.
<b>Post-condition</b>	- Sinh viên đã đăng ký thành công buổi tư vấn với tutor. - Hệ thống giảm slot còn lại. - Tutor và sinh viên đều nhận thông báo xác nhận.



<b>Normal flow</b>	<ol style="list-style-type: none"><li>Sinh viên chọn chức năng “Đăng ký buổi tư vấn”.</li><li>Hệ thống hiển thị danh sách buổi tư vấn còn slot.</li><li>Sinh viên chọn buổi tư vấn phù hợp.</li><li>Hệ thống ghi nhận đăng ký, hiển thị thông báo "Đăng ký thành công" và gửi xác nhận.</li><li>Hệ thống tự động cập nhật số lượng sinh viên tham gia.</li></ol>
<b>Alternative flow</b>	<p>2a. Không có buổi tư vấn còn slot.</p> <ul style="list-style-type: none"><li>- Hệ thống hiển thị “Không có sẵn buổi tư vấn” và quay về màn hình chính.</li></ul>
<b>Exception flow</b>	<ul style="list-style-type: none"><li>- Lỗi hệ thống hoặc mất kết nối → Hệ thống hiển thị “Không thể đăng ký, vui lòng thử lại sau”.</li></ul>

#### 1.5.12 Hủy đăng ký buổi gặp mặt

<b>Use Case ID</b>	UC-11
<b>Use Case Name</b>	Hủy đăng ký buổi gặp mặt
<b>Description</b>	Cho phép sinh viên hủy buổi tư vấn hoặc lịch hẹn đã đăng ký với tutor khi có thay đổi trong lịch trình hoặc không thể tham gia.
<b>Trigger</b>	Sinh viên chọn chức năng “Hủy đăng ký buổi gặp mặt”.
<b>Primary Actor</b>	Sinh viên
<b>Secondary Actor</b>	Tutor
<b>Pre-condition</b>	<ul style="list-style-type: none"><li>- Sinh viên đã đăng nhập.</li><li>- Sinh viên đã đăng ký buổi gặp mặt.</li><li>- Buổi gặp mặt chưa bắt đầu.</li></ul>
<b>Post-condition</b>	<ul style="list-style-type: none"><li>- Hủy thành công, hệ thống cập nhật slot và thêm lại khung giờ vào lịch rảnh của tutor.</li><li>- Sinh viên nhận thông báo hủy.</li></ul>



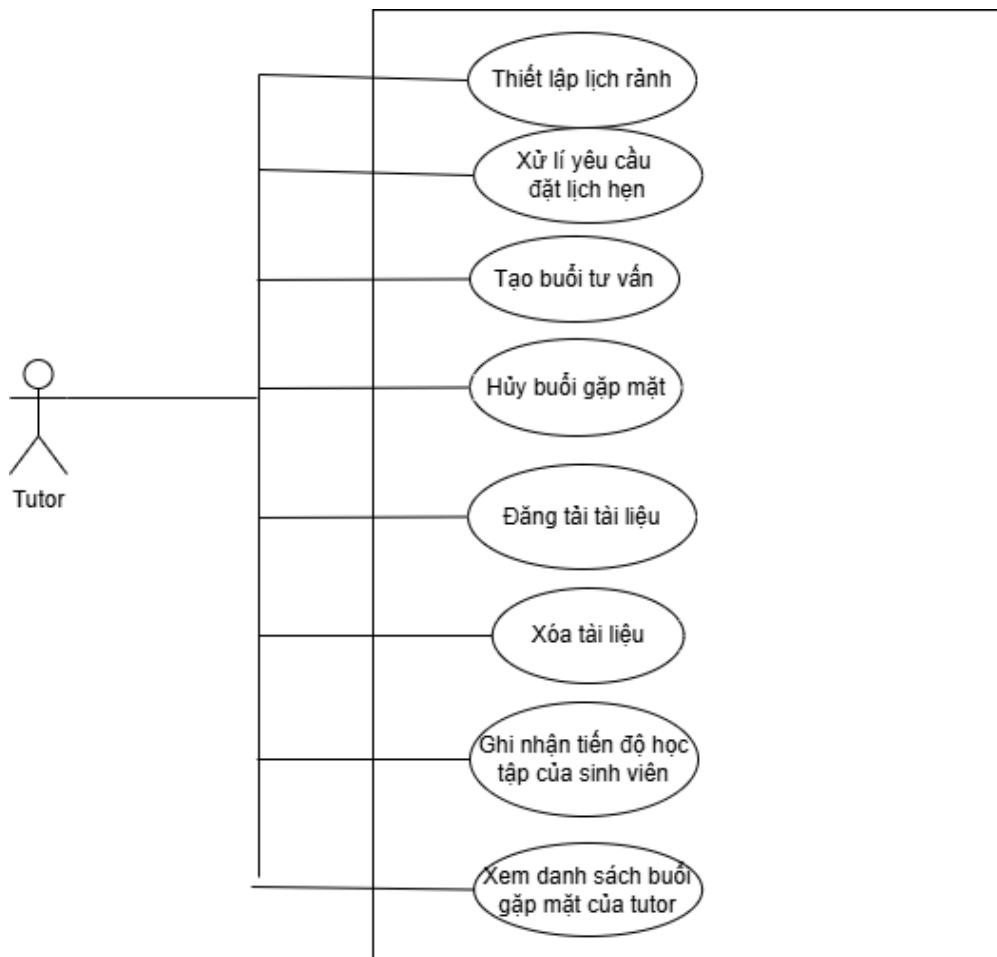
<b>Normal flow</b>	<ol style="list-style-type: none"><li>Sinh viên chọn “Buổi gặp mặt của tôi”.</li><li>Hệ thống hiển thị danh sách buổi gặp mặt đã đăng ký.</li><li>Sinh viên chọn buổi muốn hủy.</li><li>Hệ thống yêu cầu sinh viên nhập lý do hủy.</li><li>Sinh viên nhập lý do.</li><li>Hệ thống yêu cầu xác nhận hủy.</li><li>Sinh viên xác nhận.</li><li>Hệ thống kiểm tra thời gian và hủy buổi gặp mặt.</li><li>Hệ thống gửi thông báo “Hủy thành công”.</li></ol>
<b>Alternative flow</b>	<ol style="list-style-type: none"><li>Không có buổi gặp mặt đã đăng ký. Hệ thống hiển thị “Không có buổi gặp mặt nào”.</li><li>Sinh viên không nhập lý do.<ol style="list-style-type: none"><li>Hệ thống hiển thị thông báo “Vui lòng nhập lý do hủy”.</li><li>Giao diện quay lại bước 4.</li></ol></li><li>Sinh viên hủy thao tác hủy. Hệ thống quay lại danh sách buổi gặp mặt.</li></ol>
<b>Exception flow</b>	<ul style="list-style-type: none"><li>Lỗi hệ thống hoặc mất kết nối. Hệ thống hiển thị “Không thể hủy, vui lòng thử lại sau”.</li></ul>

#### 1.5.13 Xem danh sách buổi gặp mặt của sinh viên

<b>Use Case ID</b>	UC-12
<b>Use Case Name</b>	Xem danh sách buổi gặp mặt của sinh viên
<b>Description</b>	Cho phép sinh viên xem danh sách các buổi gặp mặt (sessions) mà mình đã đặt lịch hẹn, bao gồm thông tin giảng viên, thời gian, trạng thái buổi và chủ đề tư vấn.
<b>Trigger</b>	Sinh viên chọn chức năng “Xem danh sách buổi gặp” trong hệ thống.
<b>Primary Actor</b>	Sinh viên
<b>Secondary Actor</b>	Tutor
<b>Pre-condition</b>	<ul style="list-style-type: none"><li>Sinh viên đã đăng nhập thành công.</li><li>Hệ thống có ít nhất một buổi hẹn đã đăng ký thành công.</li></ul>



<b>Post-condition</b>	- Danh sách buổi gấp được hiển thị đầy đủ. - Sinh viên có thể thực hiện hành động liên quan (xem chi tiết, hủy nếu cần).
<b>Normal flow</b>	1. Sinh viên truy cập chức năng “Xem danh sách buổi gấp”. 2. Hệ thống hiển thị danh sách các buổi gấp (thời gian, giảng viên, trạng thái, chủ đề). 3. Sinh viên chọn một buổi để xem chi tiết.
<b>Alternative flow</b>	2a. Không có buổi gấp nào. Hệ thống thông báo “Danh sách trống” và quay về màn hình chính. 3a. Sinh viên hủy thao tác xem chi tiết. Hệ thống quay lại danh sách buổi gấp.
<b>Exception flow</b>	- Lỗi hệ thống hoặc mất kết nối. Hệ thống thông báo lỗi và yêu cầu sinh viên thử lại sau.



Hình 4: Nhóm chức năng của Tutor

#### 1.5.14 Tạo buổi tư vấn

Use Case ID	UC-13
Use Case Name	Tạo buổi tư vấn
Actor(s)	Tutor
Description	Tutor muốn tạo một buổi tư vấn để sinh viên có thể đăng ký tham gia sau đó. Buổi tư vấn có thể nhằm hỗ trợ học tập, hướng nghiệp hoặc phát triển kỹ năng mềm.
Trigger	Tutor muốn tạo mới một buổi tư vấn và nhấp vào nút “Tạo buổi tư vấn”.



<b>Pre-Condition(s)</b>	- Tutor đã đăng nhập thành công vào hệ thống. - Kết nối mạng ổn định và hệ thống đang hoạt động bình thường.
<b>Post-Condition(s)</b>	Buổi tư vấn được tạo thành công và hiển thị trong danh sách buổi tư vấn mở cho sinh viên đăng ký. Hệ thống gửi thông báo xác nhận cho Tutor. Lịch rảnh của Tutor được cập nhật tương ứng.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Tutor bấm vào nút “Tạo buổi tư vấn”.</li><li>2. Hệ thống hiển thị form nhập thông tin buổi tư vấn (ngày/giờ, tiêu đề, mô tả nội dung, hình thức tổ chức — online hoặc trực tiếp, và số lượng sinh viên tối đa nếu có).</li><li>3. Tutor nhập đầy đủ thông tin và bấm “Xác nhận”.</li><li>4. Hệ thống kiểm tra xung đột thời gian với lịch rảnh và các buổi tư vấn/lịch hẹn khác.</li><li>5. Hệ thống tạo buổi tư vấn mới, lưu vào cơ sở dữ liệu và cập nhật lịch rảnh của Tutor.</li><li>6. Hệ thống gửi thông báo xác nhận cho Tutor và hiển thị buổi tư vấn trong danh sách để sinh viên có thể đăng ký.</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>4a. Hệ thống phát hiện xung đột thời gian (trùng với buổi khác hoặc thời gian không hợp lệ như đã qua).<ol style="list-style-type: none"><li>4a1. Hệ thống hiển thị thông báo lỗi: “Thời gian không hợp lệ hoặc bị trùng với lịch khác”.</li><li>4a2. Giao diện quay lại bước 3 để Tutor chỉnh sửa.</li></ol></li><li>3a. Tutor hủy thao tác.<ol style="list-style-type: none"><li>3a1. Trong quá trình nhập thông tin, Tutor chọn “Hủy thao tác”.</li><li>3a2. Hệ thống không tạo buổi tư vấn mới.</li><li>3a3. Use case kết thúc sớm.</li></ol></li></ol>



<b>Exception Flow</b>	<ul style="list-style-type: none"><li>- Mất kết nối mạng trong quá trình thao tác → Hệ thống hiển thị thông báo lỗi kết nối.</li><li>- Lỗi hệ thống (server không phản hồi) → Hệ thống thông báo “Có lỗi xảy ra, vui lòng thử lại sau”.</li><li>5a. Sau khi kiểm tra thời gian và lưu dữ liệu (bước 5), hệ thống gặp lỗi cơ sở dữ liệu (DB error).<ul style="list-style-type: none"><li>5a1. Hệ thống hiển thị thông báo: “Tạo buổi tư vấn thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật”.</li><li>5a2. Hệ thống ghi log lỗi và không tạo buổi tư vấn.</li><li>5a3. Use case kết thúc không thành công.</li></ul></li></ul>
-----------------------	--

### 1.5.15 Hủy buổi gặp

<b>Use Case ID</b>	UC-14
<b>Use Case Name</b>	Hủy buổi hẹn cho Tutor
<b>Description</b>	Cho phép Tutor hủy buổi tư vấn hoặc lịch hẹn đã tạo khi có thay đổi trong lịch trình hoặc không thể tham gia.
<b>Trigger</b>	Tutor chọn chức năng “Hủy buổi gặp mặt” trong mục "Buổi gặp mặt của tôi".
<b>Primary Actor</b>	Tutor
<b>Secondary Actor</b>	Sinh viên
<b>Pre-condition</b>	<ul style="list-style-type: none"><li>- Tutor đã đăng nhập.</li><li>- Tutor đã tạo hoặc đăng ký buổi gặp mặt.</li><li>- Buổi gặp mặt chưa bắt đầu.</li></ul>
<b>Post-condition</b>	<ul style="list-style-type: none"><li>- Hủy thành công, hệ thống cập nhật slot.</li><li>- Sinh viên nhận thông báo hủy.</li></ul>



<b>Normal flow</b>	<ol style="list-style-type: none"><li>Tutor chọn “Buổi gặp mặt của tôi”.</li><li>Hệ thống hiển thị danh sách buổi gặp mặt đã tạo.</li><li>Tutor chọn buổi muốn hủy.</li><li>Hệ thống yêu cầu nhập lý do hủy.</li><li>Tutor nhập lý do.</li><li>Hệ thống yêu cầu xác nhận hủy.</li><li>Tutor xác nhận.</li><li>Hệ thống hỏi có muốn thêm lại khung thời gian đã hủy vào lịch rảnh không. Tutor chọn có hoặc không.</li><li>Tutor chọn có hoặc không.</li><li>Hệ thống gửi thông báo “Hủy thành công”.</li></ol>
<b>Alternative flow</b>	<ol style="list-style-type: none"><li>Không có buổi hẹn. Hệ thống hiển thị “Không có buổi gặp mặt nào” và quay về màn hình chính.</li><li>Tutor không nhập lý do.</li><li>5a1. Hệ thống hiển thị thông báo “Vui lòng nhập lý do hủy”.</li><li>5a2. Giao diện quay lại bước 4.</li><li>6a. Tutor hủy thao tác hủy. Hệ thống quay lại danh sách buổi gặp mặt.</li><li>9a. Tutor chọn “có”. Hệ thống thêm lại slot vào lịch rảnh và gửi thông báo hủy.</li><li>9b. Tutor chọn “không”. Hệ thống chỉ hủy buổi gặp mặt và gửi thông báo hủy.</li></ol>
<b>Exception flow</b>	<ul style="list-style-type: none"><li>Lỗi hệ thống hoặc mất kết nối. Hệ thống hiển thị “Không thể hủy, vui lòng thử lại sau”.</li></ul>

#### 1.5.16 Ghi nhận tiến độ học tập của sinh viên

<b>Use Case ID</b>	UC-15
<b>Use Case Name</b>	Ghi nhận tiến độ học tập của sinh viên
<b>Actor(s)</b>	Tutor
<b>Description</b>	Tutor muốn ghi nhận và cập nhật tiến độ học tập của sinh viên dựa trên các buổi tư vấn hoặc hỗ trợ, để theo dõi hiệu quả học tập và tổng hợp báo cáo gửi cho phòng/ban.



<b>Trigger</b>	Tutor muốn ghi nhận tiến độ học tập của sinh viên sau một buổi tư vấn. Tutor nhấp vào nút “Ghi Nhận Tiến Độ”.
<b>Pre-Condition(s)</b>	Tutor đã đăng nhập vào hệ thống. Tutor đã thực hiện ít nhất một buổi tư vấn/hỗ trợ với sinh viên. Dữ liệu sinh viên (họ tên, MSSV, trạng thái học tập) đã được đồng bộ từ HCMUT_DATACORE.
<b>Post-Condition(s)</b>	Tiến độ học tập của sinh viên được cập nhật thành công trong hệ thống. Hệ thống gửi thông báo cho tutor và sinh viên .
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Tutor truy cập mục “Ghi Nhận Tiến Độ” ở trang chủ.</li><li>2. Hệ thống hiển thị danh sách sinh viên đã được ghép đôi với Tutor và các buổi tư vấn liên quan.</li><li>3. Tutor chọn một sinh viên và buổi tư vấn cụ thể.</li><li>4. Hệ thống hiển thị form nhập tiến độ (nội dung học, đánh giá, ghi chú).</li><li>5. Tutor nhập thông tin tiến độ (bắt buộc: nội dung học, đánh giá; tùy chọn: ghi chú) và bấm “Xác nhận”.</li><li>6. Hệ thống kiểm tra tính hợp lệ của dữ liệu.</li><li>7. Hệ thống lưu thông tin tiến độ.</li><li>8. Hệ thống gửi thông báo xác nhận cho Tutor và sinh viên.</li></ol>



<b>Alternative Flow</b>	<p>6a. Hệ thống kiểm tra tính hợp lệ của dữ liệu và phát hiện trường bắt buộc bị bỏ trống.</p> <p>6a1. Hệ thống hiển thị thông báo: “Vui lòng điền đầy đủ thông tin bắt buộc”.</p> <p>6a2. Giao diện hệ thống trở lại bước 5.</p> <p>5a. Trong quá trình nhập thông tin, Tutor chọn “Hủy ghi nhận”.</p> <p>5a1. Hệ thống quay lại màn hình trang chủ.</p> <p>5a2. Use case kết thúc sớm.</p>
<b>Exception Flow</b>	<p>7a. Sau khi kiểm tra dữ liệu (bước 6) và cố gắng lưu/đồng bộ (bước 7), hệ thống mất kết nối mạng.</p> <p>7a1. Hệ thống hiển thị thông báo: “Không thể kết nối. Vui lòng thử lại sau”.</p> <p>7a2. Hệ thống ghi log lỗi và không cập nhật dữ liệu.</p> <p>7a3. Use case kết thúc không thành công.</p> <p>7b. Sau khi kiểm tra dữ liệu (bước 6) và cố gắng lưu/đồng bộ (bước 7), hệ thống gặp lỗi server không phản hồi.</p> <p>7b1. Hệ thống hiển thị thông báo: “Có lỗi xảy ra, vui lòng thử lại sau”.</p> <p>7b2. Hệ thống ghi log lỗi và không cập nhật dữ liệu.</p> <p>7b3. Use case kết thúc không thành công.</p>

#### 1.5.17 Thiết lập lịch rảnh

<b>Use Case ID</b>	UC-16
<b>Use Case Name</b>	Thiết lập lịch rảnh



<b>Actor(s)</b>	Tutor
<b>Description</b>	Tutor xem và cập nhật danh sách các khung giờ rảnh trong một ngày cụ thể để học viên có thể đăng ký.
<b>Trigger</b>	Tutor bấm vào nút lịch rảnh trên giao diện hồ sơ cá nhân.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"><li>1. Tutor đã đăng nhập vào hệ thống thành công.</li><li>2. Tài khoản có quyền hạn của Tutor.</li><li>3. Hệ thống đang hoạt động bình thường.</li></ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"><li>1. Các khung giờ rảnh cũ của ngày đó bị xóa bỏ.</li><li>2. Các khung giờ rảnh mới được lưu vào hệ thống.</li><li>3. Giao diện lịch (Calendar) hiển thị cập nhật cho ngày vừa lưu.</li></ol>
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Tutor chọn mục “Lịch rảnh” (hoặc Hồ sơ cá nhân).</li><li>2. Hệ thống hiển thị lịch theo tháng.</li><li>3. Tutor chọn một ngày cụ thể.</li><li>4. Hệ thống hiển thị danh sách các khung giờ rảnh hiện tại.</li><li>5. Tutor nhấn “Thiết lập lịch rảnh”.</li><li>6. Hệ thống chuyển sang giao diện thiết lập lịch rảnh.</li><li>7. Tutor thêm/xóa/sửa các khung giờ.</li><li>8. Tutor nhấn “Lưu”.</li><li>9. Hệ thống hiển thị thông báo “Lưu thành công”.</li></ol>



<b>Alternative Flow</b>	3a. Chọn ngày quá khứ → hệ thống chặn chỉnh sửa.  7a. Nhấn “Hủy” → quay lại không lưu.  8a. Nhập sai (start $\geq$ end, để trống) → báo lỗi.
<b>Exception Flow</b>	E1. Mất mạng lúc Lưu → báo lỗi kết nối.  E2. Lỗi hệ thống → báo lỗi và ghi log.

#### 1.5.18 Đăng tải tài liệu

<b>Use Case ID</b>	UC-17
<b>Use Case Name</b>	Đăng tải tài liệu
<b>Actor(s)</b>	Tutor, HCMUT_LIBRARY
<b>Description</b>	Tutor đăng tải tài liệu học tập (giáo trình, slide, ...) lên hệ thống thư viện để chia sẻ cho sinh viên. Tài liệu phải chờ quản trị viên (HCMUT_LIBRARY) duyệt trước khi hiển thị chính thức.
<b>Trigger</b>	Tutor bấm vào nút “Đăng tải tài liệu”.
<b>Pre-Condition(s)</b>	- Tutor đã đăng nhập thành công vào hệ thống. - Tài khoản Tutor có quyền thêm tài liệu. - Kết nối mạng ổn định và hệ thống đang hoạt động bình thường.
<b>Post-Condition(s)</b>	- Tài liệu được lưu vào hệ thống ở trạng thái “Chờ duyệt”. - Sau khi HCMUT_LIBRARY duyệt thành công, tài liệu được hiển thị cho sinh viên.



<b>Normal Flow</b>	<ol style="list-style-type: none"><li>Tutor chọn chức năng “Đăng tải tài liệu” từ giao diện hệ thống.</li><li>Hệ thống hiển thị giao diện đăng tải tài liệu.</li><li>Tutor chọn file.</li><li>Hệ thống tải tài liệu lên.</li><li>Hệ thống kiểm tra định dạng, dung lượng file.</li><li>Hệ thống lưu tài liệu ở trạng thái “Chờ duyệt”.</li><li>Hệ thống hiển thị thông báo “Yêu cầu đăng tải đã được gửi. Tài liệu sẽ được hiển thị sau khi quản trị viên phê duyệt”.</li><li>HCMUT_LIBRARY kiểm duyệt tài liệu.</li><li>Nếu được duyệt, tài liệu được cập nhật trạng thái “Đã duyệt” và hiển thị cho sinh viên.</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>Tutor chọn “Hủy”.<ol style="list-style-type: none"><li>Hệ thống hủy thao tác đăng tải.</li><li>Quay lại màn hình trước đó, không lưu bất kỳ thông tin nào.</li></ol></li><li>File sai định dạng.<ol style="list-style-type: none"><li>Hệ thống hiển thị thông báo “Định dạng không được hỗ trợ”.</li><li>Quay lại bước 3.</li></ol></li><li>File vượt dung lượng (&gt;2GB).<ol style="list-style-type: none"><li>Hệ thống hiển thị thông báo “File của bạn lớn hơn 2GB”.</li><li>Quay lại bước 3.</li></ol></li></ol>
<b>Exception Flow</b>	<ul style="list-style-type: none"><li>Lỗi lưu trữ hoặc mất kết nối → Hệ thống hiển thị thông báo “Không thể kết nối. Vui lòng thử lại sau”.</li><li>HCMUT_LIBRARY từ chối duyệt → Hệ thống gửi thông báo cho Tutor “Tài liệu bị từ chối. Vui lòng kiểm tra lại nội dung hoặc liên hệ quản trị viên.”</li></ul>

#### 1.5.19 Xóa tài liệu



<b>Use Case ID</b>	UC-18
<b>Use Case Name</b>	Xóa tài liệu
<b>Actor(s)</b>	Tutor, HCMUT_LIBRARY
<b>Description</b>	Tutor xóa tài liệu đã đăng tải (trước hoặc sau khi duyệt) nhằm quản lý lại nội dung trong thư viện số. Yêu cầu xóa cần được quản trị viên (HCMUT_LIBRARY) duyệt trước khi tài liệu bị gỡ bỏ khỏi hệ thống.
<b>Trigger</b>	Tutor chọn chức năng “Xóa tài liệu”.
<b>Pre-Condition(s)</b>	<ul style="list-style-type: none"><li>- Tutor đã đăng nhập thành công vào hệ thống.</li><li>- Tài liệu cần xóa thuộc quyền sở hữu của Tutor.</li><li>- Hệ thống hoạt động ổn định và có kết nối mạng.</li></ul>
<b>Post-Condition(s)</b>	<ul style="list-style-type: none"><li>- Tài liệu được chuyển sang trạng thái “Chờ duyệt xóa”.</li><li>- Sau khi HCMUT_LIBRARY duyệt, tài liệu bị gỡ bỏ khỏi hệ thống.</li></ul>
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Tutor chọn chức năng “Quản lý tài liệu”.</li><li>2. Hệ thống hiển thị danh sách tài liệu mà Tutor đã đăng tải.</li><li>3. Tutor chọn một hoặc nhiều tài liệu cần xóa.</li><li>4. Tutor chọn nút “Xóa tài liệu”.</li><li>5. Hệ thống hiển thị hộp thoại xác nhận.</li><li>6. Tutor chọn “Đồng ý”.</li><li>7. Hệ thống ghi nhận yêu cầu xóa và chuyển tài liệu sang trạng thái “Chờ duyệt xóa”.</li><li>8. Hệ thống hiển thị thông báo: “Yêu cầu xóa đã được gửi. Tài liệu sẽ bị gỡ bỏ sau khi quản trị viên phê duyệt”.</li><li>9. HCMUT_LIBRARY kiểm duyệt yêu cầu xóa.</li><li>10. Nếu được duyệt, hệ thống gỡ bỏ tài liệu khỏi thư viện số.</li></ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"><li>6a. Tutor chọn “Hủy” tại hộp thoại xác nhận.<ol style="list-style-type: none"><li>6a1. Hệ thống hủy thao tác xóa.</li><li>6a2. Hệ thống quay lại màn hình danh sách tài liệu, không có thay đổi nào được thực hiện.</li></ol></li></ol>



<b>Exception Flow</b>	<ul style="list-style-type: none"><li>- Lỗi khi ghi nhận yêu cầu xóa → Hệ thống hiển thị thông báo “Không thể gửi yêu cầu xóa. Vui lòng thử lại sau”.</li><li>- Mất kết nối mạng trong quá trình thao tác → Hệ thống hiển thị thông báo lỗi kết nối.</li><li>- Lỗi hệ thống (server không phản hồi) → Hệ thống thông báo “Có lỗi xảy ra, vui lòng thử lại sau”.</li><li>- HCMUT_LIBRARY từ chối yêu cầu xóa → Hệ thống gửi thông báo cho Tutor “Yêu cầu xóa bị từ chối. Vui lòng kiểm tra lại hoặc liên hệ quản trị viên.”</li></ul>
-----------------------	--

#### 1.5.20 Xử lý yêu cầu đặt lịch hẹn

<b>Use Case ID</b>	UC-19
<b>Use Case Name</b>	Xử lý yêu cầu đặt lịch hẹn
<b>Actor(s)</b>	Tutor
<b>Description</b>	Tutor xem danh sách các yêu cầu đặt lịch từ sinh viên, sau đó phê duyệt hoặc từ chối từng yêu cầu.
<b>Trigger</b>	Tutor bấm vào nút “Xử lý yêu cầu đặt lịch hẹn”.
<b>Pre-Condition(s)</b>	<ul style="list-style-type: none"><li>- Tutor đã đăng nhập vào hệ thống.</li><li>- Sinh viên đã gửi yêu cầu đặt lịch hẹn cho Tutor.</li><li>- Đối với mỗi yêu cầu đang chờ xử lý, slot rảnh tương ứng trong lịch của Tutor được hệ thống tạm giữ.</li><li>- Hệ thống hoạt động bình thường và có kết nối mạng.</li></ul>
<b>Post-Condition(s)</b>	<ul style="list-style-type: none"><li>- Yêu cầu lịch hẹn được Tutor xử lý (phê duyệt hoặc từ chối).</li><li>- Hệ thống gửi thông báo kết quả cho sinh viên.</li><li>- Nếu duyệt: lịch hẹn được ghi nhận chính thức vào hệ thống, slot tương ứng được đánh dấu là bận trong lịch của Tutor.</li><li>- Nếu từ chối: yêu cầu bị hủy và không ghi vào hệ thống, slot tương ứng có thể được mở lại hoặc bị khóa tùy vào thiết lập của Tutor.</li></ul>



<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Tutor truy cập mục “Xử lý yêu cầu lịch hẹn”.</li><li>2. Hệ thống hiển thị danh sách các yêu cầu lịch hẹn đang chờ xử lý.</li><li>3. Tutor chọn một yêu cầu cụ thể để xem chi tiết (thông tin sinh viên, thời gian, nội dung).</li><li>4. Hệ thống hiển thị chi tiết thông tin yêu cầu.</li><li>5. Tutor chọn “Phê duyệt” yêu cầu.</li><li>6. Hệ thống ghi nhận quyết định của Tutor.</li><li>7. Hệ thống cập nhật trạng thái slot của Tutor thành bận, không cho sinh viên khác đăng ký và gửi thông báo kết quả xử lý cho sinh viên.</li></ol>
--------------------	--



<b>Alternative Flow</b>	<p>2a. Không có yêu cầu nào đang chờ xử lý.</p> <p>2a1. Hệ thống hiển thị thông báo: “Không có yêu cầu lịch hẹn nào”.</p> <p>2a2. Tutor chọn “Quay lại” để trở về trang quản lý.</p> <p>2a3. Use case kết thúc sớm.</p> <p>4a. Tutor chọn “Từ chối” yêu cầu.</p> <p>4a1. Hệ thống hiển thị hộp thoại yêu cầu Tutor nhập lý do từ chối (tùy chọn) và chọn cách xử lý slot.</p> <p>4a2. Tutor nhập lý do (hoặc bỏ trống) và chọn một trong hai tùy chọn xử lý slot:</p> <p>4a2.1. “Mở lại slot”: Slot được đánh dấu lại là rảnh trong lịch của Tutor, sinh viên khác có thể đặt.</p> <p>4a2.2. “Không mở lại slot”: Slot vẫn được giữ ở trạng thái bận, không hiển thị như một slot rảnh cho sinh viên khác.</p> <p>4a3. Tutor xác nhận thao tác từ chối.</p> <p>4a4. Hệ thống ghi nhận quyết định từ chối, xử lý trạng thái slot theo lựa chọn ở bước 4a2 và gửi thông báo cho sinh viên.</p> <p>4b. Tutor đổi ý và chọn “Hủy thao tác” trước khi xác nhận.</p> <p>4b1. Hệ thống không lưu bất kỳ thay đổi nào.</p> <p>4b2. Giao diện quay lại danh sách yêu cầu (hoặc màn hình chi tiết yêu cầu tùy thiết kế).</p> <p>4b3. Use case kết thúc sớm.</p>
-------------------------	---



<b>Exception Flow</b>	<p>5a. Khi ghi nhận quyết định, hệ thống gặp lỗi cơ sở dữ liệu hoặc mất kết nối.</p> <p>5a1. Hệ thống hiển thị thông báo: “Xử lý thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật”.</p> <p>5a2. Hệ thống ghi log lỗi, không thay đổi trạng thái yêu cầu và trạng thái slot.</p> <p>5a3. Use case kết thúc không thành công.</p>
-----------------------	---

#### 1.5.21 Xem danh sách buổi gặp mặt của Tutor

<b>Use Case ID</b>	UC-20
<b>Use Case Name</b>	Xem danh sách buổi gặp mặt của Tutor
<b>Description</b>	Cho phép Tutor xem danh sách các buổi gặp mặt (sessions) mà mình đã nhận từ yêu cầu đặt lịch hẹn, bao gồm thông tin sinh viên, thời gian, trạng thái buổi và chủ đề tư vấn.
<b>Trigger</b>	Tutor chọn chức năng “Xem danh sách buổi gặp” trong hệ thống.
<b>Primary Actor</b>	Tutor
<b>Secondary Actor</b>	Sinh viên
<b>Pre-condition</b>	- Tutor đã đăng nhập thành công. - Hệ thống có ít nhất một buổi gặp liên quan đến Tutor.
<b>Post-condition</b>	- Danh sách buổi gặp được hiển thị đầy đủ. - Tutor có thể thực hiện hành động liên quan (xem chi tiết, hủy nếu cần).
<b>Normal flow</b>	1. Tutor truy cập chức năng “Xem danh sách buổi gặp”. 2. Hệ thống hiển thị danh sách các buổi gặp (thời gian, sinh viên, trạng thái, chủ đề). 3. Tutor chọn một buổi để xem chi tiết.

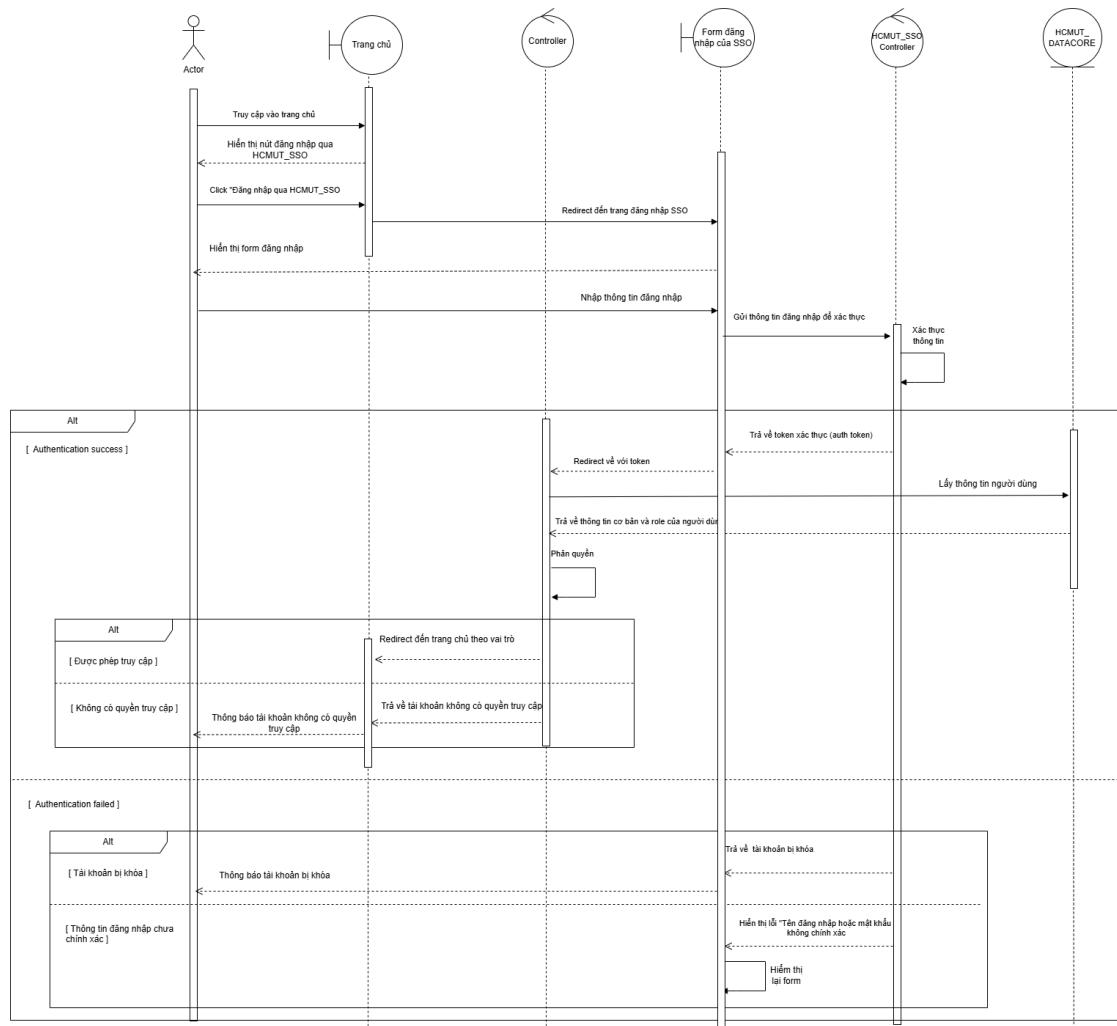


<b>Alternative flow</b>	2a. Không có buổi gặp nào. Hệ thống thông báo “Danh sách trống” và quay về màn hình chính. 3a. Tutor hủy thao tác xem chi tiết. Hệ thống quay lại danh sách buổi gặp.
<b>Exception flow</b>	- Lỗi hệ thống hoặc mất kết nối. Hệ thống thông báo lỗi và yêu cầu Tutor thử lại sau.

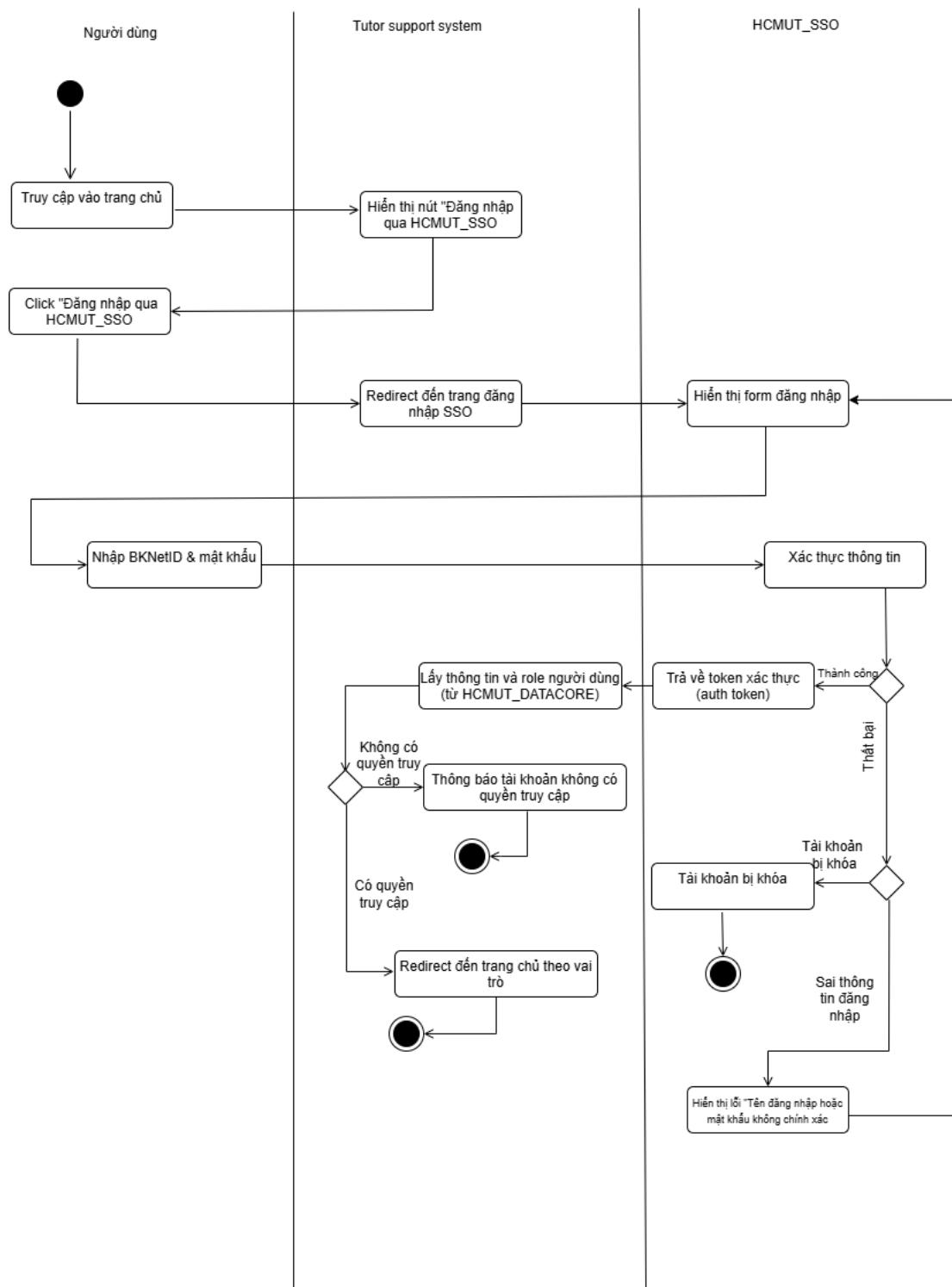
## 2 Task 2

### 2.1 Sequence Diagram và Activity Diagram

#### 2.1.1 Đăng nhập



Hình 5: Sequence Diagram cho UC Đăng nhập



Hình 6: Activity Diagram cho UC Đăng nhập



**Mô tả:**

**Luồng sự kiện chính: Đăng nhập thành công**

1. **Người dùng** truy cập vào trang chủ của hệ thống **Tutor Support System**.
2. **System** hiển thị giao diện đăng nhập với nút “**Đăng nhập qua HCMUT\_SSO**”.
3. **Người dùng** nhấp vào nút “Đăng nhập qua HCMUT\_SSO”.
4. **System** chuyển hướng người dùng đến trang đăng nhập của **HCMUT\_SSO**.
5. **Người dùng** nhập thông tin đăng nhập (BKNetID và mật khẩu).
6. **HCMUT\_SSO** xác thực thông tin đăng nhập.
7. Nếu thông tin hợp lệ, **HCMUT\_SSO** trả về **auth token** cho **Tutor Support System**.
8. **Tutor Support System** sử dụng token để truy vấn **HCMUT\_DATACORE** và lấy thông tin người dùng (họ tên, mã số, email, vai trò, trạng thái).
9. **Tutor Support System** kiểm tra quyền truy cập:
  - Nếu người dùng có vai trò hợp lệ, hệ thống phân quyền và khởi tạo phiên đăng nhập.
10. **System** chuyển hướng người dùng đến trang chủ tương ứng với vai trò (Sinh viên / Tutor / Điều phối viên).

**Các luồng phụ và luồng lỗi (Alternative & Error Flows)**

1. **Thông tin đăng nhập không chính xác:**

Nếu người dùng nhập sai BKNetID hoặc mật khẩu:

- **HCMUT\_SSO** hiển thị thông báo lỗi trực tiếp trên form đăng nhập.

Use Case quay lại bước nhập thông tin đăng nhập (bước 5).

2. **Tài khoản bị khóa :**

Nếu tài khoản không thể xác thực do trạng thái tài khoản không hợp lệ:

- **HCMUT\_SSO** hiển thị thông báo lỗi tương ứng trên form đăng nhập.

Use Case kết thúc không thành công tại SSO.



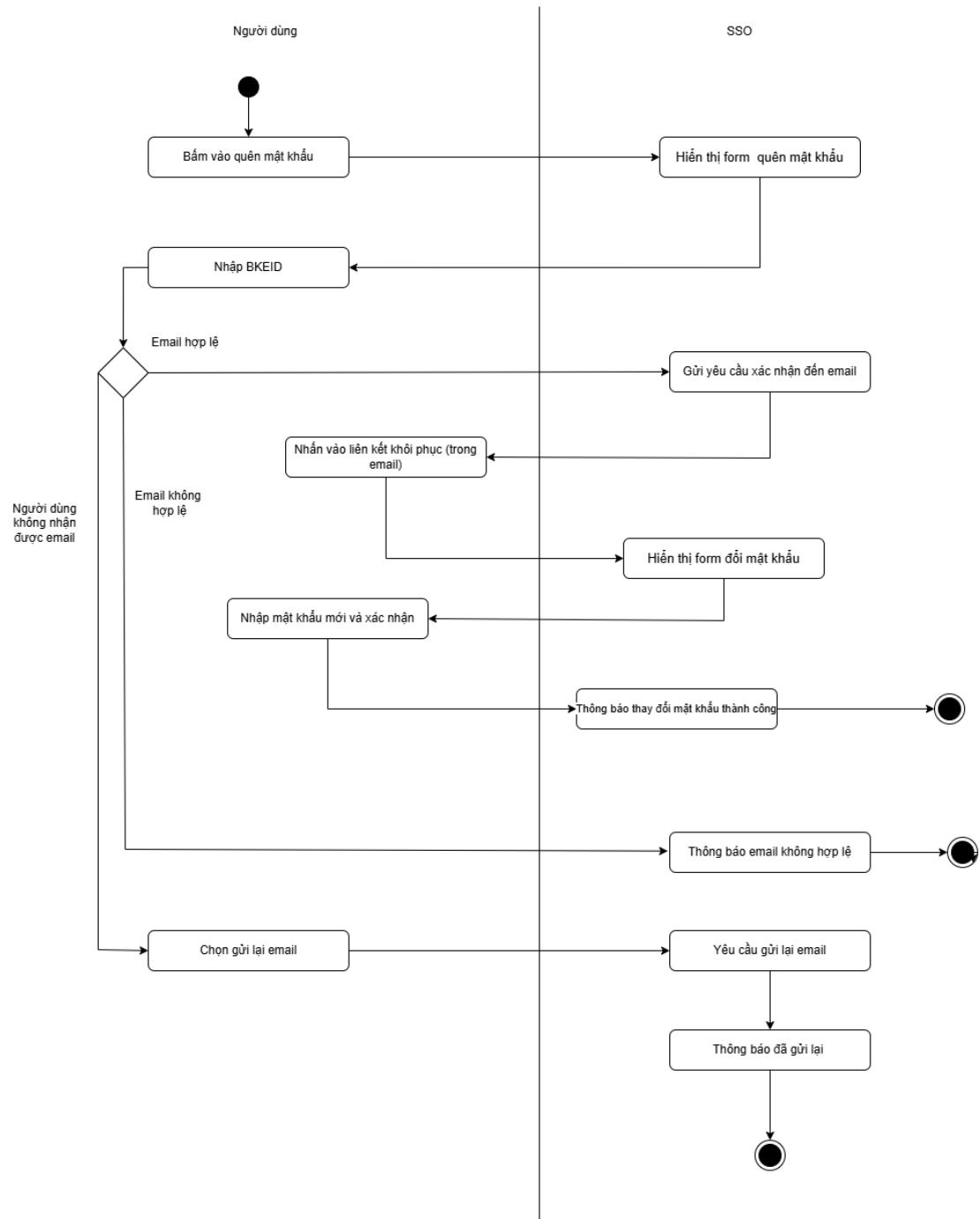
### 3. Không có quyền truy cập hệ thống:

Nếu người dùng đăng nhập thành công nhưng vai trò không thuộc nhóm được phép sử dụng hệ thống:

- **Tutor Support System** hiển thị thông báo: “*Bạn không có quyền truy cập hệ thống.*”

Phiên đăng nhập không được tạo; Use Case kết thúc.

### 2.1.2 Quên mật khẩu



Hình 7: Activity Diagram cho UC Quên mật khẩu



**Mô tả:** Use Case mô tả quy trình người dùng yêu cầu đặt lại mật khẩu tài khoản thông qua hệ thống HCMUT\_SSO. Người dùng nhập BKID, nhận email khôi phục, truy cập liên kết đặt lại mật khẩu và tạo mật khẩu mới.

**Luồng sự kiện chính: Khôi phục mật khẩu thành công**

1. **Người dùng** truy cập trang đăng nhập và chọn nút “**Quên mật khẩu**”.
2. **SSO** hiển thị form yêu cầu nhập BKID.
3. **Người dùng** nhập BKID.
4. **SSO** kiểm tra email tương ứng với BKID.
5. Nếu email hợp lệ, **SSO** gửi email chứa liên kết khôi phục mật khẩu.
6. **Người dùng** mở email và nhấn vào liên kết đặt lại mật khẩu.
7. **SSO** hiển thị form đặt lại mật khẩu.
8. **Người dùng** nhập mật khẩu mới và xác nhận mật khẩu.
9. **SSO** kiểm tra tính hợp lệ của mật khẩu mới.
10. **SSO** cập nhật mật khẩu vào hệ thống.
11. **SSO** hiển thị thông báo “**Thay đổi mật khẩu thành công**”.

**Các luồng phụ và luồng lỗi (Alternative & Error Flows)**

**1. Email không hợp lệ hoặc không tồn tại:**

Nếu BKID không có email hợp lệ:

- **SSO** hiển thị thông báo “Email không hợp lệ” trên form.

Use Case kết thúc không thành công.

**2. Người dùng không nhận được email:**

Nếu người dùng chờ nhưng không nhận email:

- Người dùng chọn “**Gửi lại email**”.
- **SSO** gửi lại email khôi phục.
- **SSO** hiển thị thông báo “Email đã được gửi lại”.



**3. Liên kết khôi phục hết hạn hoặc không hợp lệ:**

- **SSO** hiển thị thông báo “Liên kết không hợp lệ hoặc đã hết hạn”.

Use Case kết thúc; người dùng phải thực hiện lại từ đầu.

**4. Mật khẩu mới không hợp lệ:**

Khi người dùng nhập mật khẩu mới không đạt yêu cầu:

- **SSO** hiển thị lỗi tương ứng (quá ngắn, không trùng khớp...).

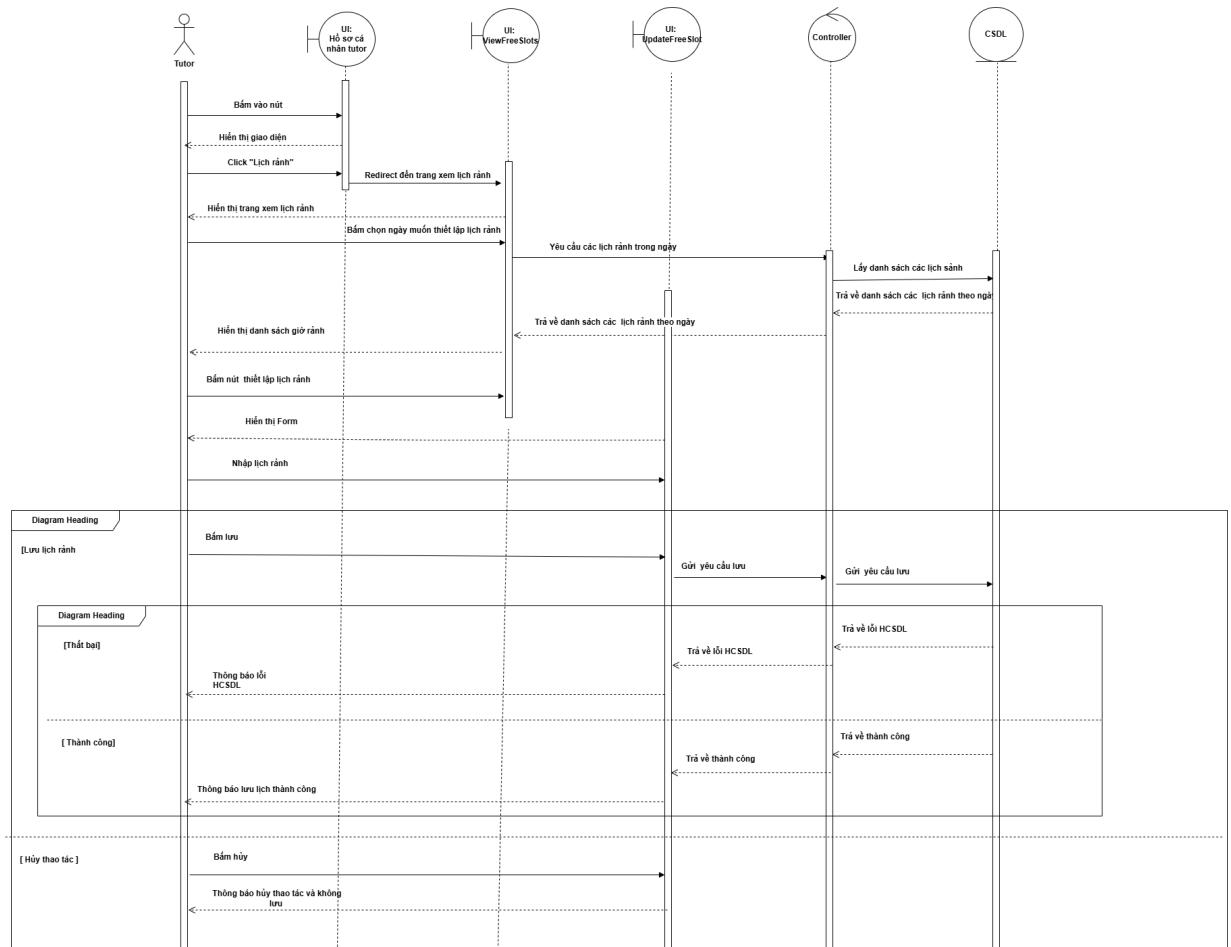
Use Case quay lại bước nhập mật khẩu mới.

**5. Lỗi hệ thống khi cập nhật mật khẩu:**

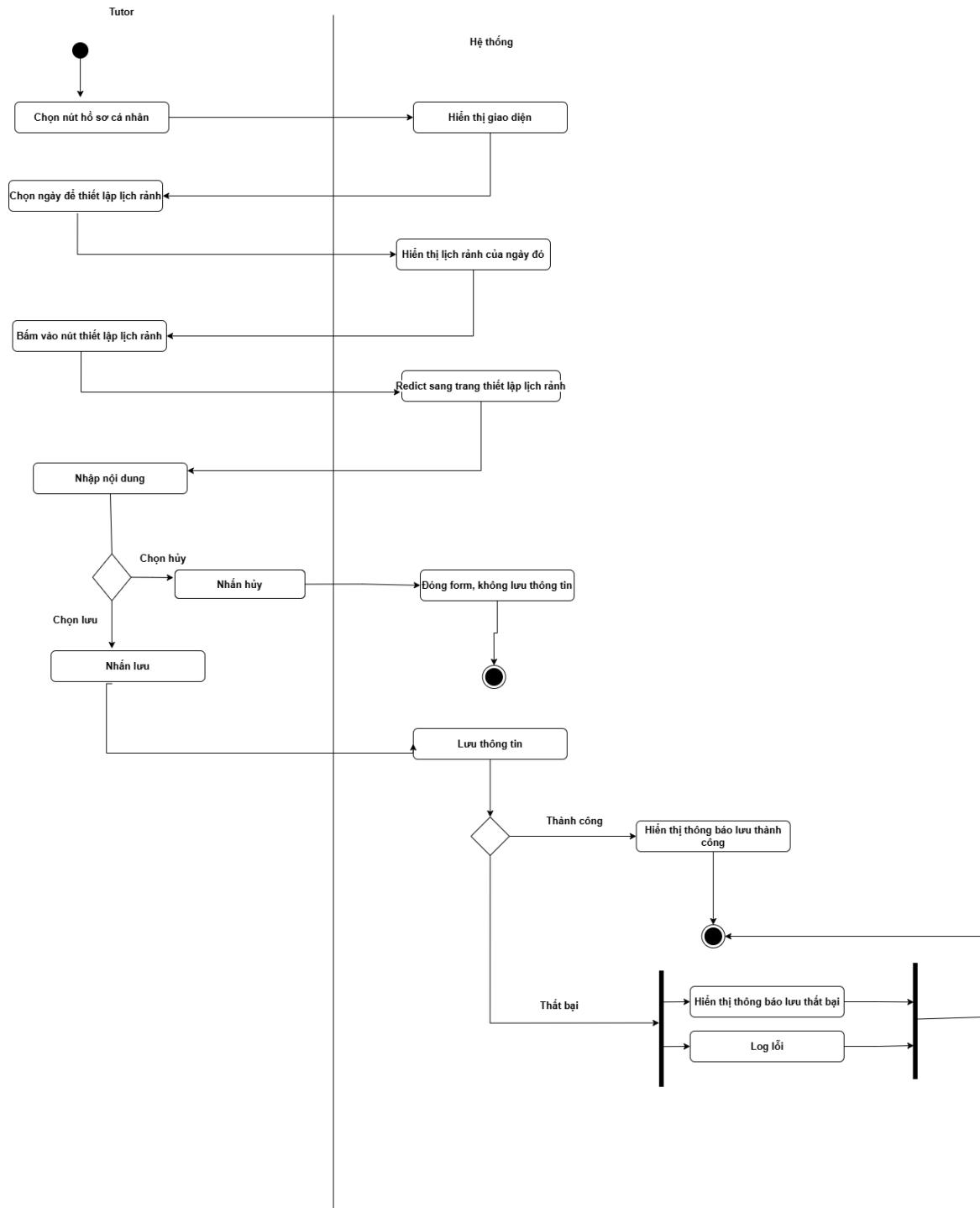
- **SSO** hiển thị thông báo lỗi: “Không thể thay đổi mật khẩu. Vui lòng thử lại sau.”

Use Case kết thúc không thành công.

### 2.1.3 Thiết lập lịch rảnh



Hình 8: Sequence Diagram cho Use Case “Thiết lập lịch rảnh”



Hình 9: Activity Diagram cho Use Case “Thiết lập lịch rảnh”



## Mô tả

### Luồng sự kiện chính: Thêm lịch rảnh thành công

1. **Tutor** truy cập vào chức năng “Thêm lịch rảnh” trên giao diện hệ thống.
2. **UI** hiển thị Form **Thêm lịch rảnh** cho **Tutor** nhập thông tin chi tiết.
3. **Tutor** nhập các thông tin cần thiết: ngày, giờ bắt đầu và kết thúc, khoảng thời gian rảnh.
4. Khi hoàn tất, **Tutor** nhấn nút **Xác nhận**. **UI** gửi dữ liệu tới **System**.
5. **System** thực hiện kiểm tra hợp lệ: các trường bắt buộc đã nhập, định dạng thời gian đúng.
6. Nếu hợp lệ, **System** truy vấn **Database** để kiểm tra: lịch rảnh mới có trùng với lịch cũ hay không.
7. Nếu không có xung đột, **System** lưu thông tin lịch rảnh mới vào **Database**.
8. **System** trả về thông báo thành công cho **UI**.
9. **UI** hiển thị thông báo: “*Thêm lịch rảnh thành công*” và cập nhật danh sách lịch rảnh trên giao diện.

### Các luồng phụ và luồng lỗi (Alternative & Error Flows)

#### 1. Thông tin không hợp lệ:

Nếu **Tutor** không điền đủ thông tin hoặc nhập sai định dạng, **System** từ chối xử lý và trả về lỗi cho **UI**.

**UI** hiển thị: “*Không điền đủ thông tin. Vui lòng kiểm tra lại.*”

Form giữ nguyên để **Tutor** chỉnh sửa và gửi lại.

#### 2. Lịch rảnh trùng với lịch cũ:

Khi kiểm tra **Database**, nếu lịch rảnh mới trùng với lịch đã tồn tại, **System** trả về cảnh báo.

**UI** hiển thị: “*Lịch rảnh này đã tồn tại.*”

**Tutor** có thể chỉnh sửa lại thời gian và gửi lại.

#### 3. Tutor nhấn Hủy thao tác:

Nếu **Tutor** nhấn nút **Hủy** trong khi nhập form, **UI** đóng form và không lưu bất kỳ thông tin nào.

Use Case kết thúc tại đây.



#### 4. Thiếu thông tin bắt buộc:

Nếu Tutor bỏ trống các trường bắt buộc, **System** thông báo lỗi.

**UI** hiển thị: “*Không điền đủ thông tin.*”

Tutor phải nhập đầy đủ trước khi tiếp tục.

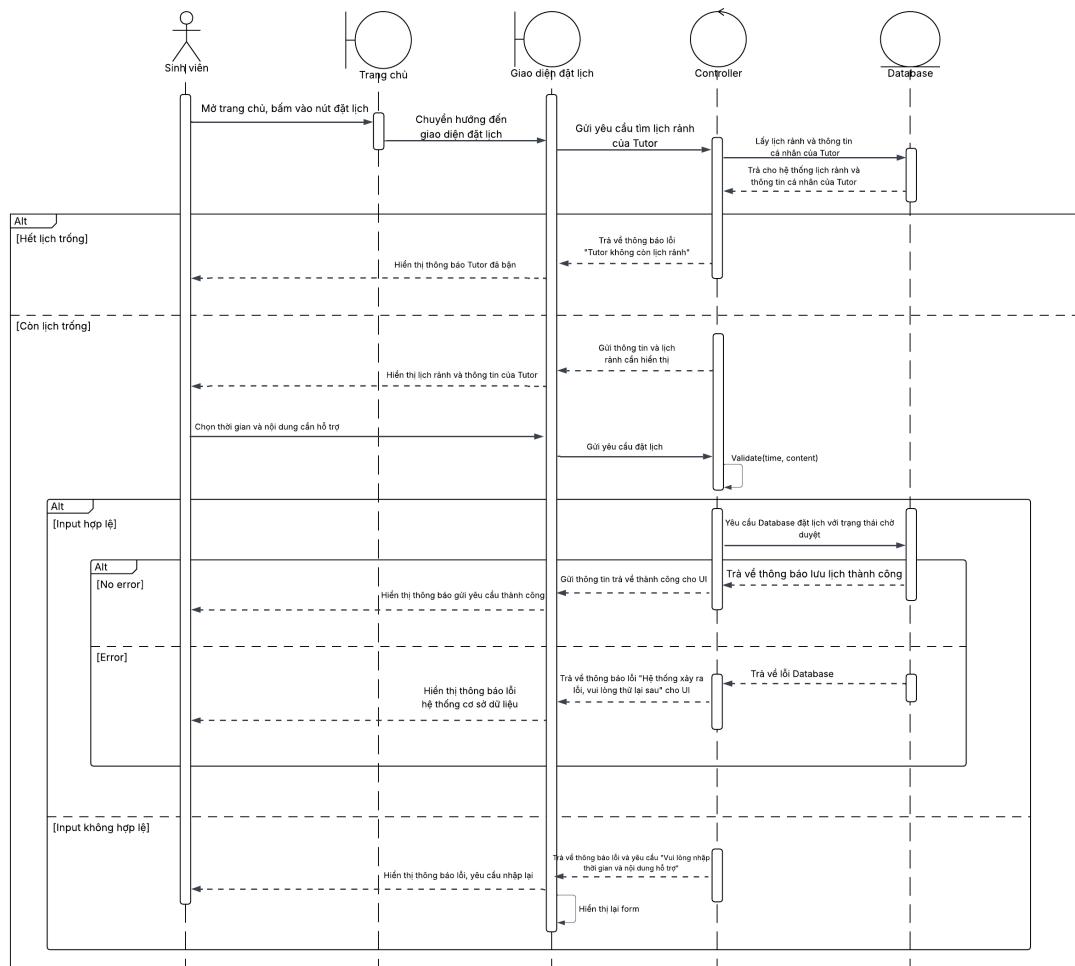
#### 5. Lỗi hệ thống / mất kết nối:

Nếu xảy ra mất kết nối mạng hoặc lỗi server khi lưu dữ liệu, **System** trả về thông báo lỗi.

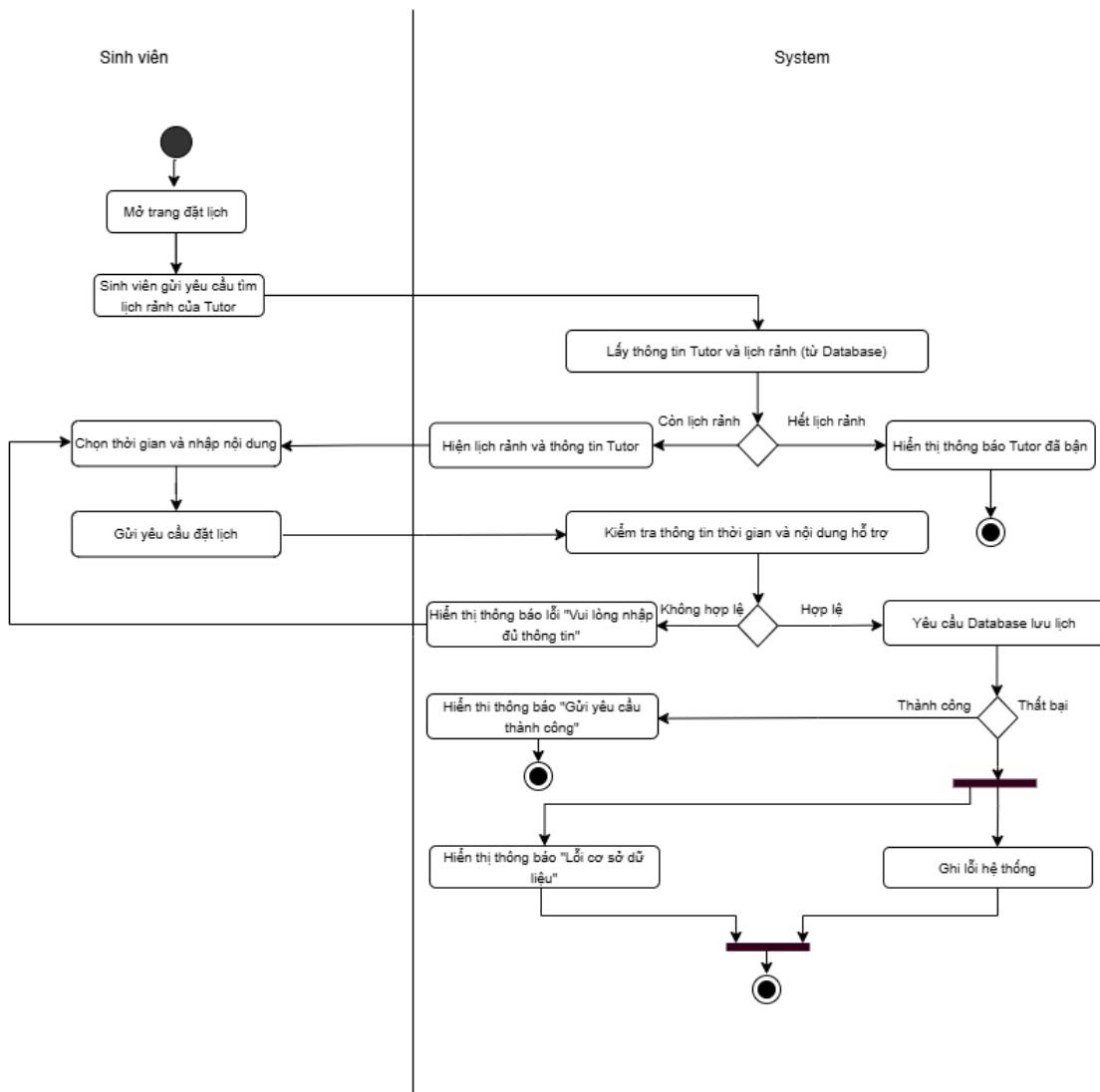
**UI** hiển thị: “*Không thể kết nối. Vui lòng thử lại sau.*”

Use Case kết thúc không thành công.

### 2.1.4 Đặt lịch hẹn



Hình 10: Sequence Diagram cho UC Đặt lịch hẹn



Hình 11: Activity Diagram cho UC Đặt lịch hẹn

#### Mô tả: Luồng sự kiện chính: Đặt lịch thành công

1. Sinh viên truy cập vào trang **đặt lịch** của một giảng viên cụ thể.
2. UI gửi yêu cầu tìm lịch rảnh của Tutor tới **System**.
3. **System** thực hiện truy vấn **Database** để lấy đồng thời:
  - Thông tin chi tiết của giảng viên (tên, email);
  - Danh sách các khung giờ rảnh (`time_available`).



4. **System** tổng hợp dữ liệu và trả về cho **UI** để hiển thị đầy đủ cho Sinh viên.
5. Sinh viên xem lịch và chọn một khung giờ phù hợp, sau đó nhấn nút **Xác nhận**.
6. **UI** gửi yêu cầu đặt lịch kèm theo thông tin cần thiết đến **System**.
7. **System** tạo một bản ghi yêu cầu mới và lưu vào **Database** với trạng thái ban đầu là **pending** (đang chờ duyệt).
8. **System** trả về thông báo thành công cho **UI**.
9. **UI** hiển thị thông báo: “*Yêu cầu đã được gửi, vui lòng chờ Tutor duyệt.*”

#### Các luồng phụ và luồng lỗi (Alternative & Error Flows)

##### 1. Thông tin không hợp lệ:

Nếu thông tin mà Sinh viên gửi lên không hợp lệ (ví dụ: không chọn khung giờ hoặc nội dung yêu cầu trống), **System** sẽ từ chối xử lý và trả về thông báo lỗi cho **UI**. **UI** hiển thị: “*Thời gian bạn chọn không hợp lệ, vui lòng chọn lại*”. Lúc này, giao diện màn hình quay về trang chọn thời gian và nhập nội dung hỗ trợ.

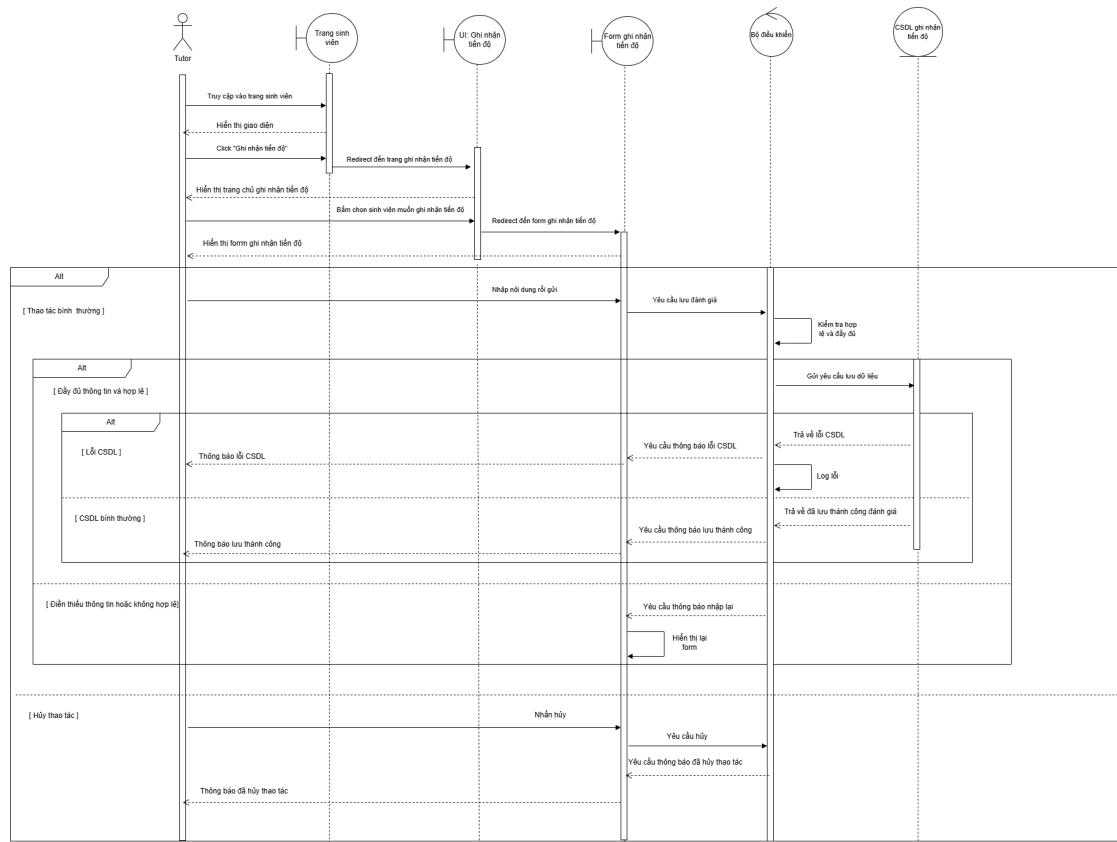
##### 2. Không có lịch trống:

Khi **System** truy vấn **Database** ở bước xem lịch, nếu kết quả trả về là danh sách rỗng, **System** sẽ thông báo cho **UI**. **UI** hiển thị: “*Tutor đã hết lịch trống.*”

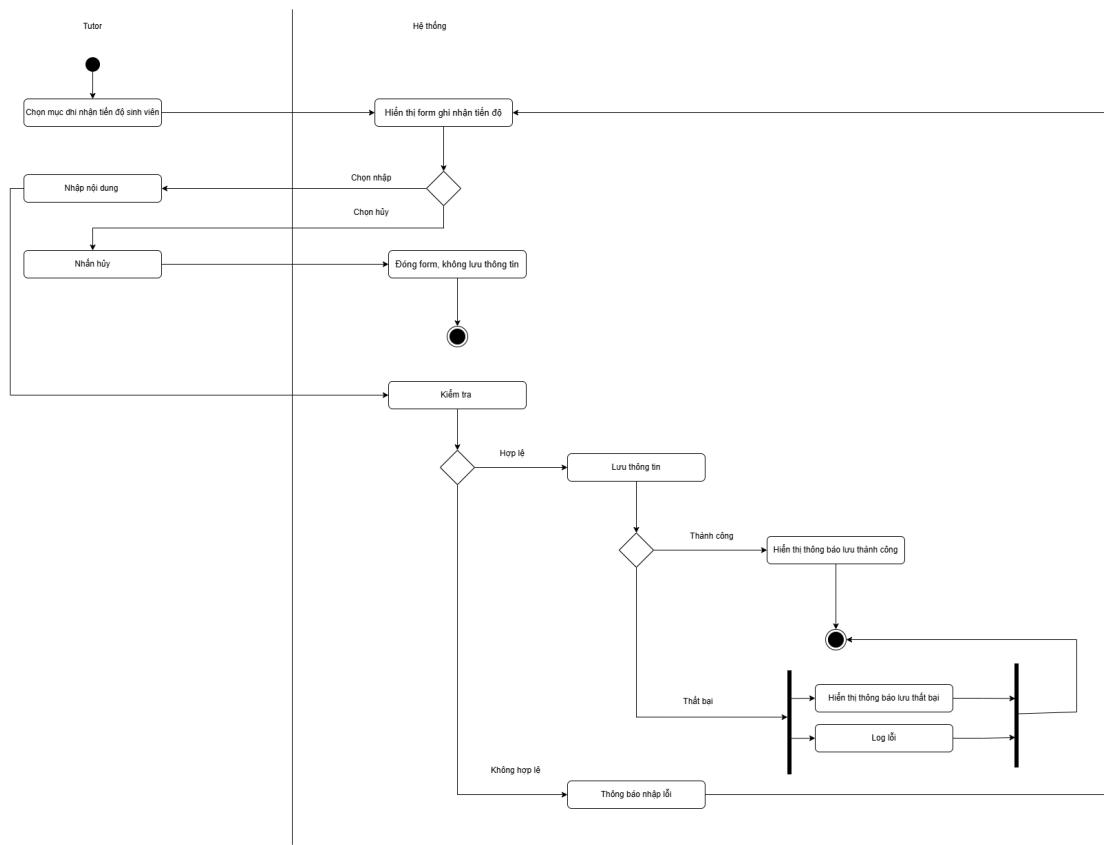
##### 3. Lỗi khi lưu dữ liệu:

Trong quá trình lưu thông tin đặt lịch, nếu xảy ra lỗi (ví dụ: mất kết nối, lỗi truy vấn), **Database** ném ngoại lệ. **System** trả về thông báo lỗi chung cho người dùng. “*Hệ thống đang gặp sự cố, vui lòng thử lại sau.*”

### 2.1.5 Ghi nhận tiền độ



Hình 12: Sequence Diagram cho UC Ghi nhận tiền độ



Hình 13: Activity Diagram cho UC Ghi nhận tiến độ

#### Mô tả:

Use case mô tả quy trình Tutor ghi nhận tiến độ học tập hoặc kết quả buổi học của sinh viên trong hệ thống Tutor Support System. Tiến độ bao gồm nội dung mô tả, đánh giá, mức độ hoàn thành và các ghi chú liên quan.

#### Luồng sự kiện chính: Ghi nhận tiến độ thành công

1. **Tutor** truy cập vào hệ thống **Tutor Support System**.
2. **System** hiển thị giao diện chính với mục chức năng “**Ghi nhận tiến độ**”.
3. **Tutor** chọn chức năng “**Ghi nhận tiến độ**”.
4. **System** hiển thị danh sách sinh viên hoặc buổi học mà Tutor đang phụ trách.
5. **Tutor** chọn một sinh viên hoặc buổi học cần ghi nhận tiến độ.



6. **System** hiển thị form ghi nhận tiền độ, bao gồm: mô tả, đánh giá, mức độ hoàn thành, kết quả học tập.
7. **Tutor** nhập đầy đủ các thông tin tiền độ.
8. **Tutor** nhấn nút “**Gửi báo cáo**”.
9. **System** kiểm tra dữ liệu nhập (các trường bắt buộc, định dạng).
10. Nếu dữ liệu hợp lệ, **System** lưu tiền độ vào cơ sở dữ liệu.
11. **System** gửi thông báo xác nhận ghi nhận thành công.
12. **System** hiển thị thông báo “**Ghi nhận tiền độ thành công**”.

#### Các luồng phụ và luồng lỗi (Alternative & Error Flows)

##### 1. Chọn nhầm sinh viên hoặc buổi học:

Nếu Tutor chọn sai đối tượng:

- **System** hiển thị thông báo yêu cầu chọn lại.

Use Case quay lại bước chọn sinh viên/buổi học (bước 5).

##### 2. Tutor hủy thao tác trước khi gửi báo cáo:

Nếu Tutor quay lại hoặc thoát form ghi nhận:

- **System** hủy các dữ liệu đang nhập.

Use Case quay về danh sách sinh viên/buổi học.

##### 3. Dữ liệu nhập không hợp lệ:

Khi Tutor nhập thiếu thông tin hoặc sai định dạng:

- **System** hiển thị lỗi tương ứng trực tiếp trên form.

Use Case quay lại bước 7.

##### 4. Lỗi kết nối cơ sở dữ liệu:

Nếu hệ thống không thể lưu tiền độ:

- **System** hiển thị thông báo: “Không thể lưu tiền độ. Vui lòng thử lại sau.”



---

Use Case kết thúc không thành công.

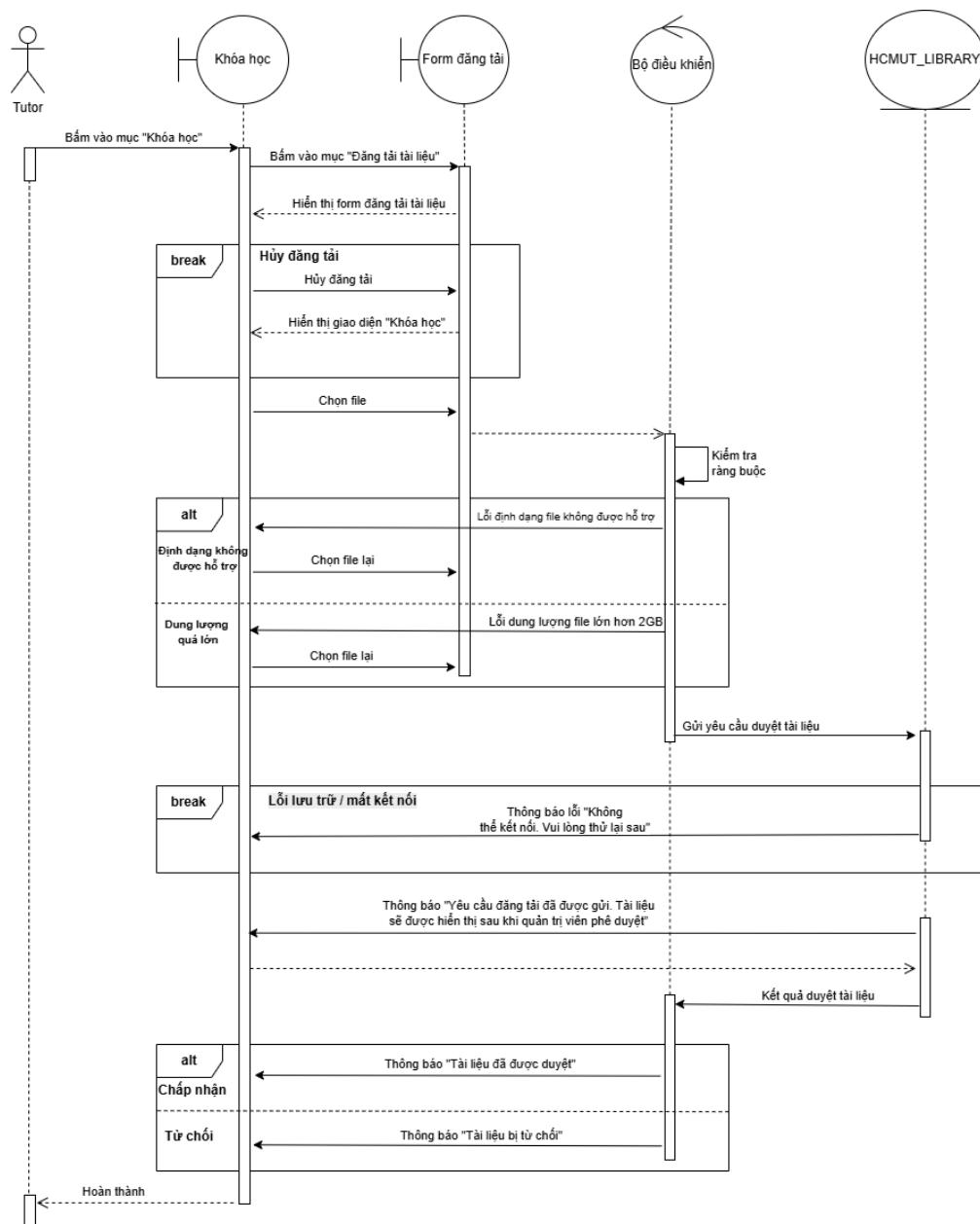
**5. Lỗi khi gửi thông báo xác nhận:**

Nếu tiến độ đã được lưu nhưng hệ thống không gửi được thông báo:

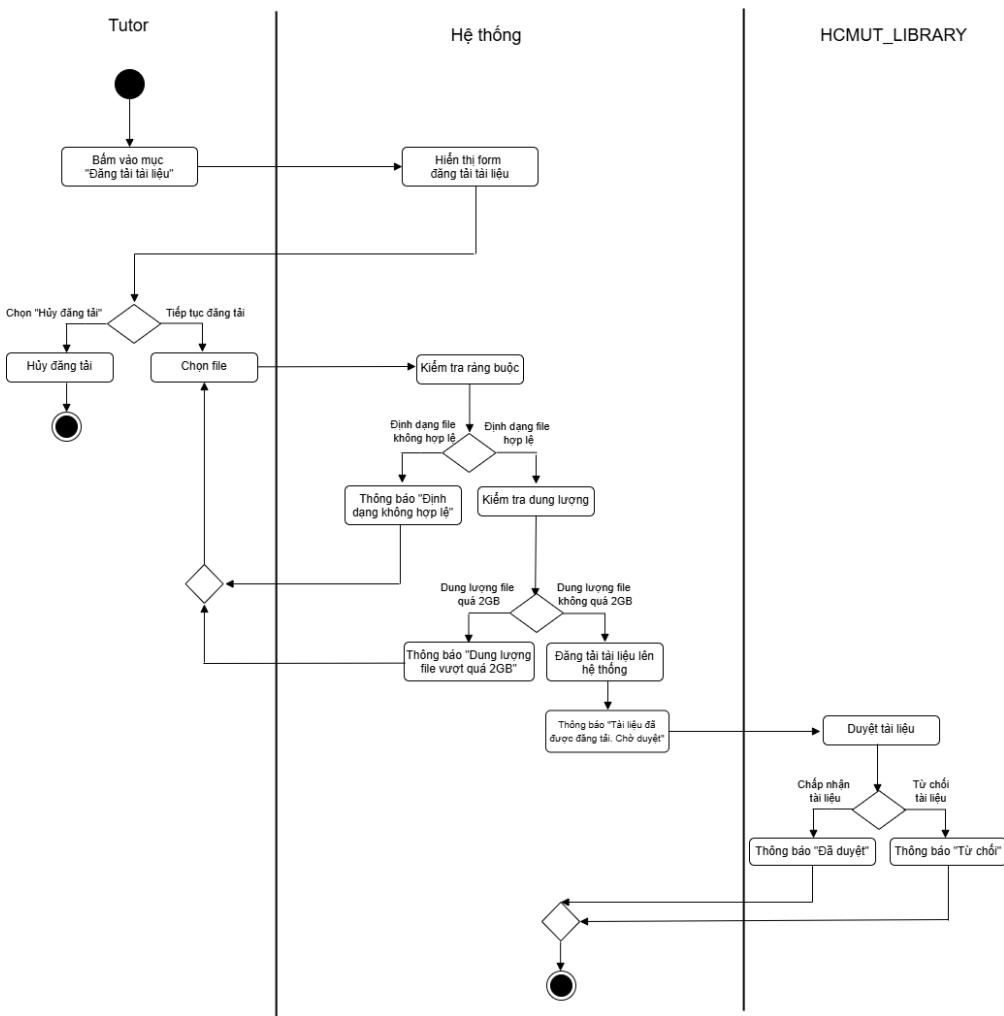
- **System** hiển thị cảnh báo: “Ghi nhận thành công nhưng không gửi được thông báo.”

Use Case kết thúc.

### 2.1.6 Đăng tải tài liệu



Hình 14: Sequence diagram: Đăng tải tài liệu



Hình 15: Activity diagram: Đăng tải tài liệu

## Mô tả

### Luồng sự kiện chính: Đăng tải tài liệu thành công

1. Tutor bấm vào mục “Đăng tải tài liệu” trong trang “Khóa học”.
2. Hệ thống hiển thị form đăng tải tài liệu.
3. Tutors nhấn “Chọn file” và chọn tệp cần đăng hoặc kéo / thả file cần đăng.
4. Hệ thống kiểm tra ràng buộc:
  - (a) Kiểm tra định dạng tệp có được hỗ trợ.



- (b) Kiểm tra dung lượng tệp không vượt quá 2GB.
5. Nếu hợp lệ, Hệ thống tải tệp lên kho lưu trữ.
6. Hệ thống thông báo “Tài liệu đã được đăng tải. Chờ duyệt.”
7. Hệ thống gửi yêu cầu duyệt sang *HCMUT\_LIBRARY*.
8. *HCMUT\_LIBRARY* duyệt tài liệu và chấp nhận.
9. Hệ thống hiển thị thông báo “Đã duyệt” và tài liệu xuất hiện trong danh sách của khóa học.
10. Use Case kết thúc thành công.

#### Các luồng phụ và luồng lỗi (Alternative & Error Flows)

##### Hủy đăng tải

1. Ở bước 2, Tutor chọn “Hủy”.
2. Hệ thống quay về giao diện “Khóa học”. Use Case kết thúc.

##### Định dạng không được hỗ trợ

1. Ở bước 4(a), nếu tệp không đúng định dạng cho phép, hệ thống thông báo “Định dạng không hợp lệ”.
2. Quay về bước 3 để tutor chọn file lại.

##### Dung lượng vượt quá giới hạn

1. Ở bước 4(b), nếu dung lượng > 2GB, Hệ thống thông báo “Dung lượng file vượt quá 2GB”.
2. Quay về bước 3 để tutor chọn file lại.

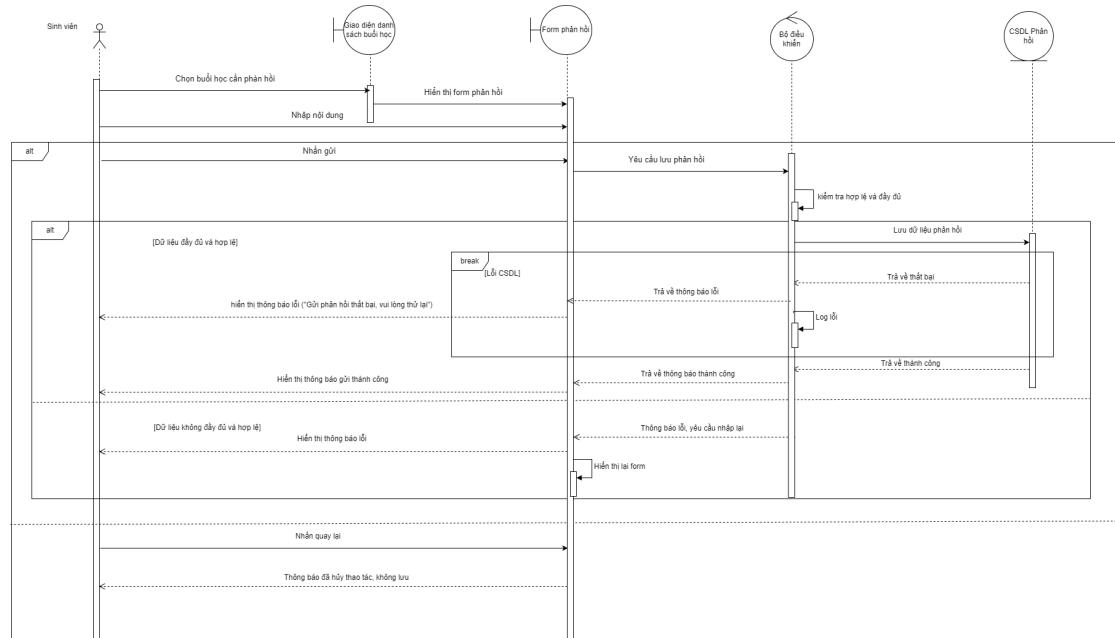
##### Lỗi lưu trữ / mất kết nối

1. Trong bước 5, nếu lỗi phát sinh, Hệ thống thông báo “Không thể kết nối. Vui lòng thử lại sau.”
2. Use Case tạm dừng; Tutor có thể thực hiện lại từ bước 1 khi kết nối ổn định.

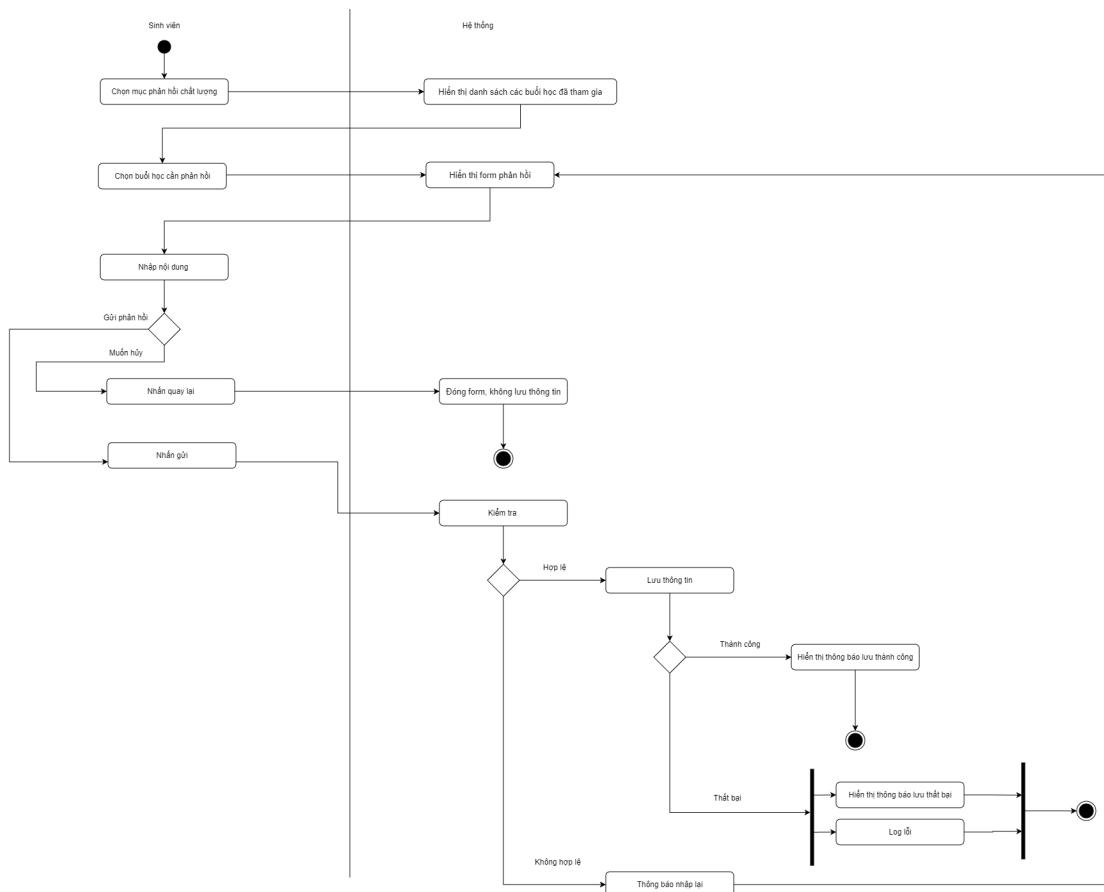
##### Từ chối phê duyệt

1. Ở bước 8, nếu bị từ chối, Hệ thống hiển thị “Tài liệu bị từ chối” (kèm lý do nếu có).
2. Tutor có thể chỉnh sửa/dổi tệp và đăng lại từ bước 1. Use Case kết thúc không thành công.

### 2.1.7 Phản hồi chất lượng buổi học



Hình 16: Sequence Diagram cho Phản hồi chất lượng buổi học



Hình 17: Activity Diagram cho Phản hồi chất lượng buổi học

#### Mô tả:

##### Luồng sự kiện chính: Gửi phản hồi thành công

- Sinh viên truy cập vào hệ thống và chọn mục “Phản hồi chất lượng”
- Hệ thống hiển thị danh sách các buổi học đã tham gia.
- Sinh viên chọn buổi học cần phản hồi.
- Hệ thống (UI) hiển thị form phản hồi bao gồm: mức độ hài lòng và ô nhập nội dung.
- Sinh viên nhập nội dung phản hồi, chọn mức độ đánh giá, sau đó nhấn “Gửi”.
- Hệ thống nhận dữ liệu, thực hiện kiểm tra tính hợp lệ (đủ nội dung, không chứa ký tự lõi, định dạng hợp lệ).



7. Nếu dữ liệu hợp lệ, Hệ thống lưu phản hồi vào **CSDL Phản hồi**.
8. Khi lưu thành công, Hệ thống gửi phản hồi xác nhận về cho UI.
9. UI hiển thị thông báo: “Gửi phản hồi thành công.”
10. Use case kết thúc.

### **Luồng phụ và luồng lỗi**

#### **1. Sinh viên nhấn “Quay lại”**

- Khi form phản hồi đang hiển thị, nếu sinh viên chọn “Quay lại”, Hệ thống đóng form và không lưu bất kỳ thông tin nào.
- Use case kết thúc sớm.

#### **2. Dữ liệu không hợp lệ**

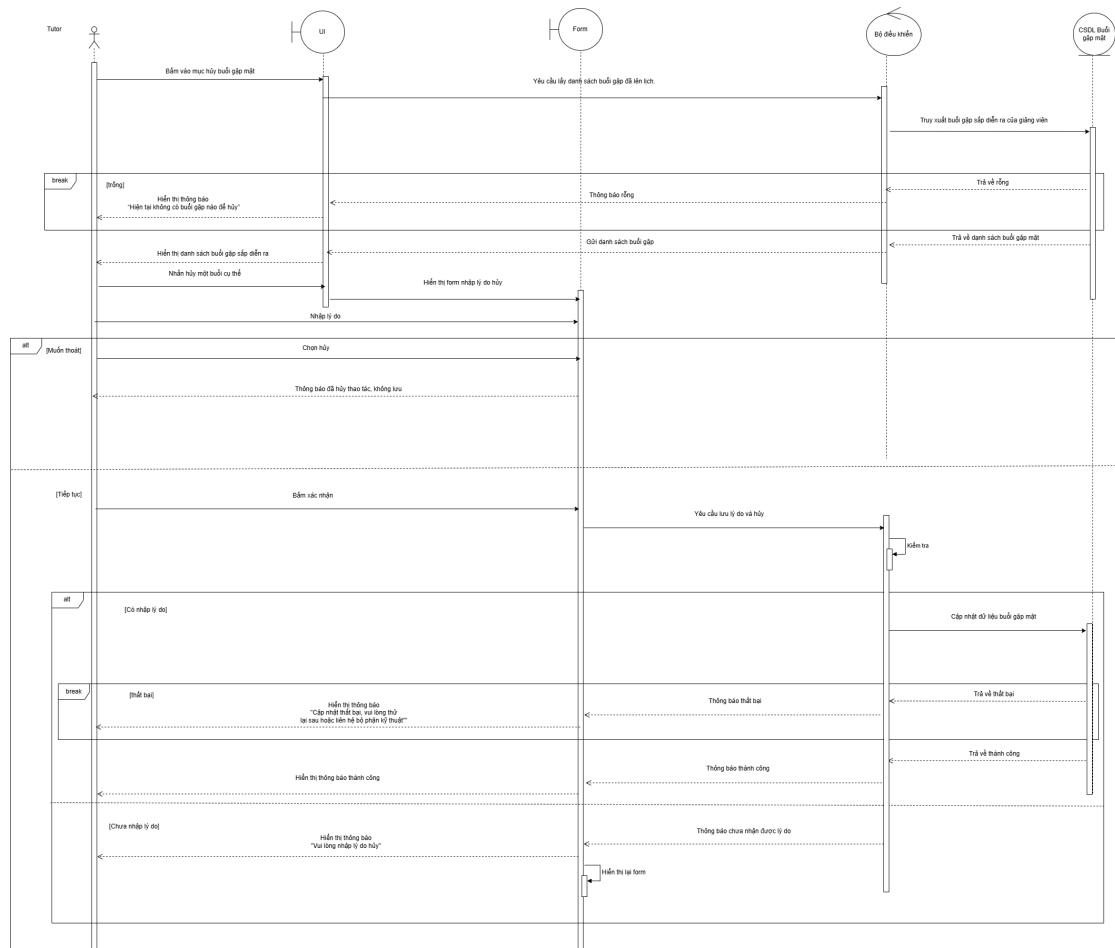
- Nếu sinh viên không nhập đủ nội dung phản hồi, Hệ thống hiển thị thông báo lỗi: “Nội dung phản hồi không hợp lệ, vui lòng nhập lại.”
- Sau đó, giao diện quay lại form phản hồi ban đầu (reset form rỗng).
- Sinh viên có thể chọn nhập lại nội dung và gửi lại hoặc nhấn hủy.

#### **3. Lỗi khi lưu dữ liệu**

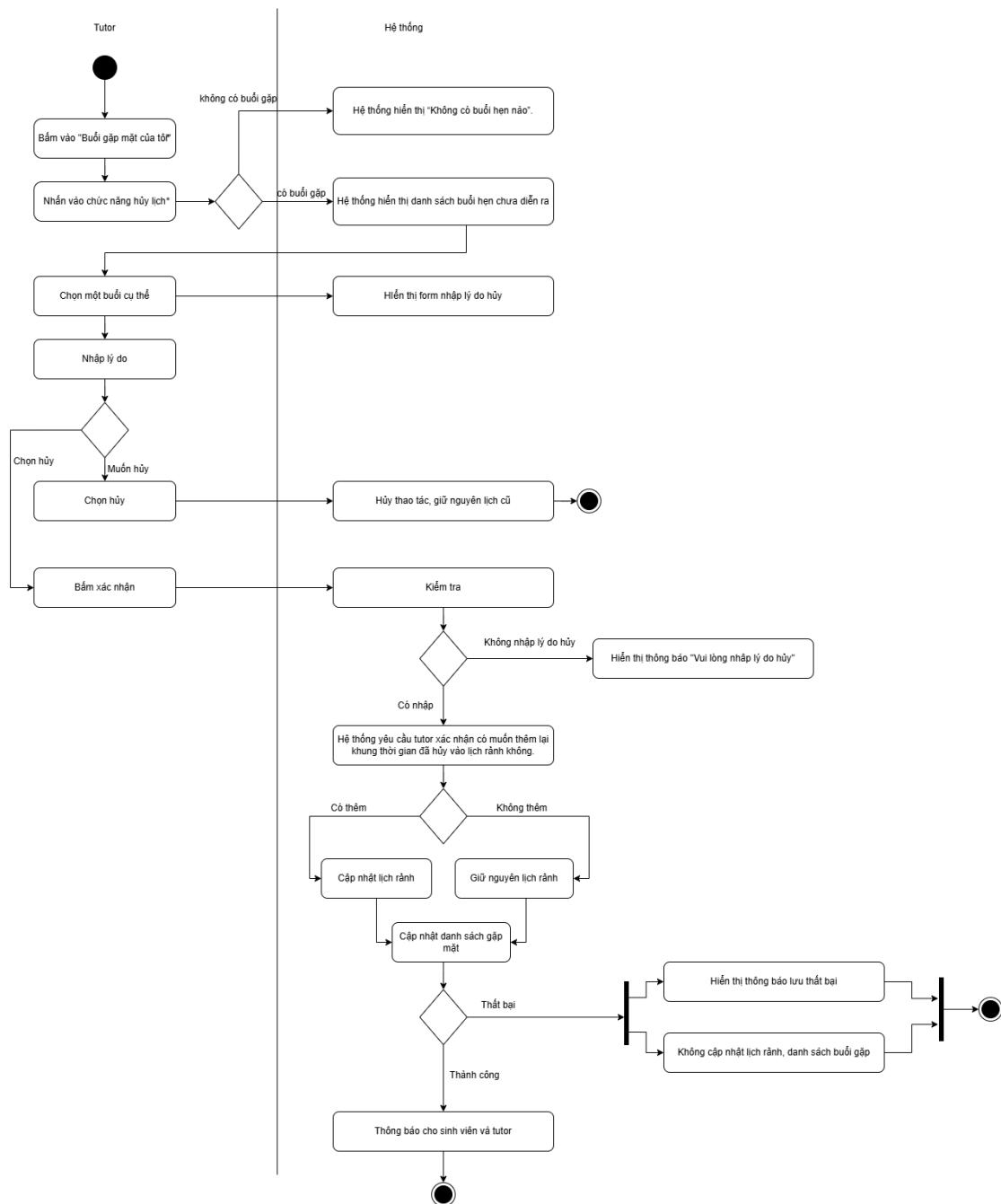
- Trong quá trình lưu phản hồi, nếu CSDL xảy ra lỗi (ví dụ: mất kết nối, lỗi truy vấn), Hệ thống bắt ngoại lệ và gọi bộ ghi log (Log lỗi) để ghi chi tiết sự cố.
- Sau đó, Hệ thống gửi phản hồi lỗi về cho UI.
- UI hiển thị thông báo: “Gửi phản hồi thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật.”
- Use case kết thúc không thành công.

#### **2.1.8 Hủy buổi gặp mặt**

**Đối với Tutor:**



Hình 18: Sequence Diagram cho Hủy buổi gặp mặt



Hình 19: Activity Diagram cho Hủy buổi gặp mặt của tutor

**Mô tả:**

**Luồng sự kiện chính: Hủy buổi gặp thành công**



1. Tutor truy cập vào buổi hẹn của tôi và chọn mục “Hủy lịch”.
2. Hệ thống (UI) hiển thị danh sách buổi hẹn có thẻ hủy.
3. Tutor chọn một buổi gặp cụ thể cần hủy.
4. Hệ thống hiển thị form yêu cầu nhập lý do hủy.
5. Tutor nhập lý do và nhấn “Xác nhận”.
6. Hệ thống kiểm tra tính hợp lệ của dữ liệu (có nhập lý do, không chứa ký tự lỗi).
7. Hệ thống yêu cầu tutor xác nhận có muốn khôi phục lại lịch rảnh đã hủy không.
8. Hệ thống cập nhật trạng thái buổi gặp trong cơ sở dữ liệu (hủy buổi gặp) và cập nhật lại lịch rảnh của Tutor.
9. Khi lưu thay đổi thành công, hệ thống gửi phản hồi xác nhận cho UI.
10. UI hiển thị thông báo: “Hủy buổi gặp thành công.”
11. Use case kết thúc.

### **Luồng phụ và luồng lỗi**

#### **1. Sinh viên nhấn “Hủy”**

- Khi form phản hồi đang hiển thị, nếu sinh viên chọn “Hủy”, Hệ thống đóng form và không lưu bất kỳ thông tin nào.
- Use case kết thúc sớm.

#### **2. Dữ liệu không hợp lệ**

- Nếu Tutor không nhập nội dung phản hồi hoặc nội dung không đạt yêu cầu, Hệ thống hiển thị thông báo lỗi:

“Nội dung phản hồi không hợp lệ, vui lòng nhập lại.”

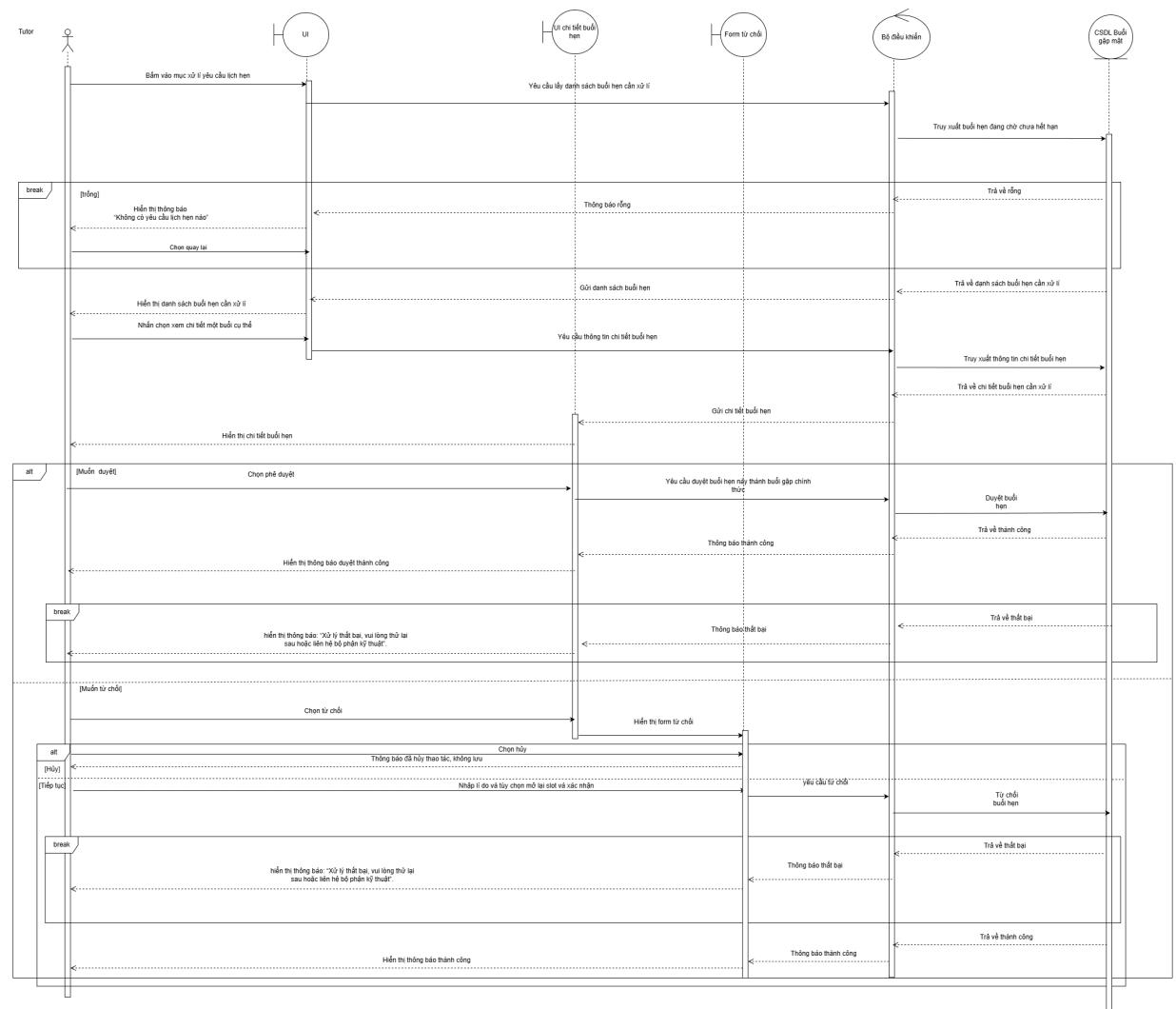
- Sau đó, giao diện quay lại form phản hồi ban đầu (reset form rỗng).
- Tutor có thể chọn nhập lại nội dung và gửi lại hoặc nhấn hủy.

#### **3. Lỗi khi lưu dữ liệu**

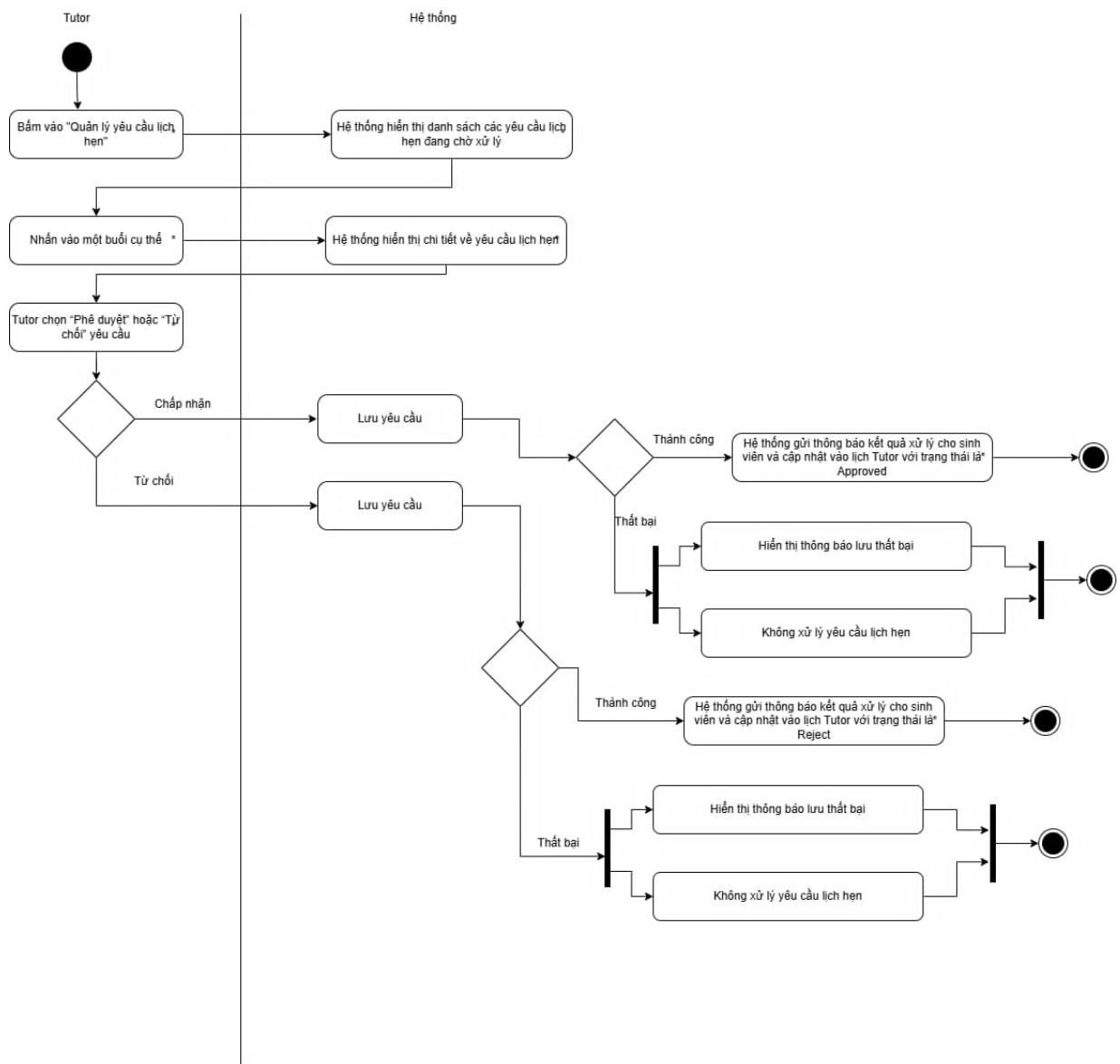
- Trong quá trình lưu phản hồi, nếu CSDL xảy ra lỗi (ví dụ: mất kết nối, lỗi truy vấn), Hệ thống bắt ngoại lệ và gọi bộ ghi log (Log lỗi) để ghi chi tiết sự cố.
  - Sau đó, Hệ thống gửi phản hồi lỗi về cho UI.
  - UI hiển thị thông báo:

“Gửi phản hồi thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật.”
  - Use case kết thúc không thành công.

### 2.1.9 Xử lý lịch hen



Hình 20: Sequence Diagram cho Use Case “Xử lý lịch hẹn”



Hình 21: Activity Diagram cho Use Case “Xử lý lịch hẹn”

**Mô tả:** Tutor xem và xử lý các yêu cầu đặt lịch hẹn từ sinh viên, bao gồm phê duyệt hoặc từ chối yêu cầu, đồng thời hệ thống cập nhật trạng thái slot tương ứng và gửi thông báo cho sinh viên.

#### Luồng sự kiện chính: Xử lý yêu cầu đặt lịch hẹn thành công

- Tutor truy cập vào mục Quản lý yêu cầu lịch hẹn”.



2. Hệ thống hiển thị danh sách các yêu cầu lịch hẹn đang chờ xử lý.
3. Tutor chọn một yêu cầu cụ thể để xem chi tiết thông tin (thông tin sinh viên, thời gian, nội dung).
4. Tutor chọn Phê duyệt” yêu cầu.
5. Hệ thống ghi nhận quyết định phê duyệt của Tutor.
6. Hệ thống cập nhật trạng thái slot tương ứng của Tutor thành bận, không cho sinh viên khác đăng ký.
7. Hệ thống gửi thông báo kết quả xử lý đến sinh viên.
8. UI hiển thị thông báo xác nhận: “Yêu cầu đã được phê duyệt”.
9. Use case kết thúc.

### **Luồng phụ và luồng lỗi**

#### **1. Không có yêu cầu nào đang chờ xử lý**

- Nếu danh sách yêu cầu trống, Hệ thống hiển thị thông báo: Không có yêu cầu lịch hẹn nào”.
- Tutor chọn Quay lại” để trở về trang quản lý.
- Use case kết thúc sớm.

#### **2. Tutor từ chối yêu cầu**

- Tutor chọn “Từ chối” yêu cầu. Hệ thống hiển thị hộp thoại để nhập lý do từ chối (tùy chọn) và chọn cách xử lý slot:
  - “Mở lại slot”: Slot trở về trạng thái rảnh, sinh viên khác có thể đặt.
  - “Không mở lại slot”: Slot vẫn bận, không hiển thị cho sinh viên khác.
- Tutor xác nhận thao tác từ chối.
- Hệ thống ghi nhận quyết định từ chối, cập nhật trạng thái slot theo lựa chọn và gửi thông báo cho sinh viên.
- Use case kết thúc.



### 3. Tutor hủy thao tác từ chối

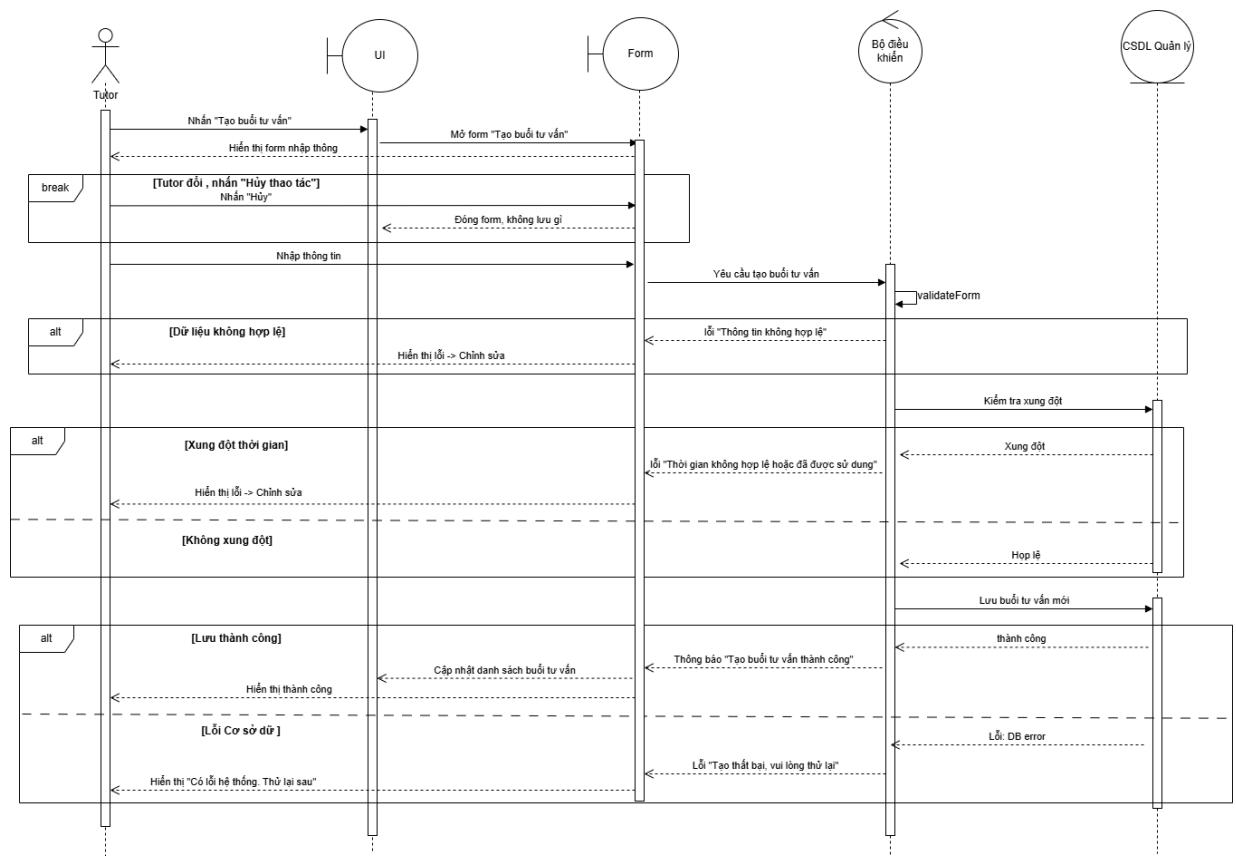
- Nếu Tutor chọn “Hủy thao tác” trước khi xác nhận, hệ thống không lưu bất kỳ thay đổi nào.
- Giao diện quay lại danh sách yêu cầu hoặc màn hình chi tiết yêu cầu.
- Use case kết thúc sớm.

### 4. Lỗi khi ghi nhận quyết định

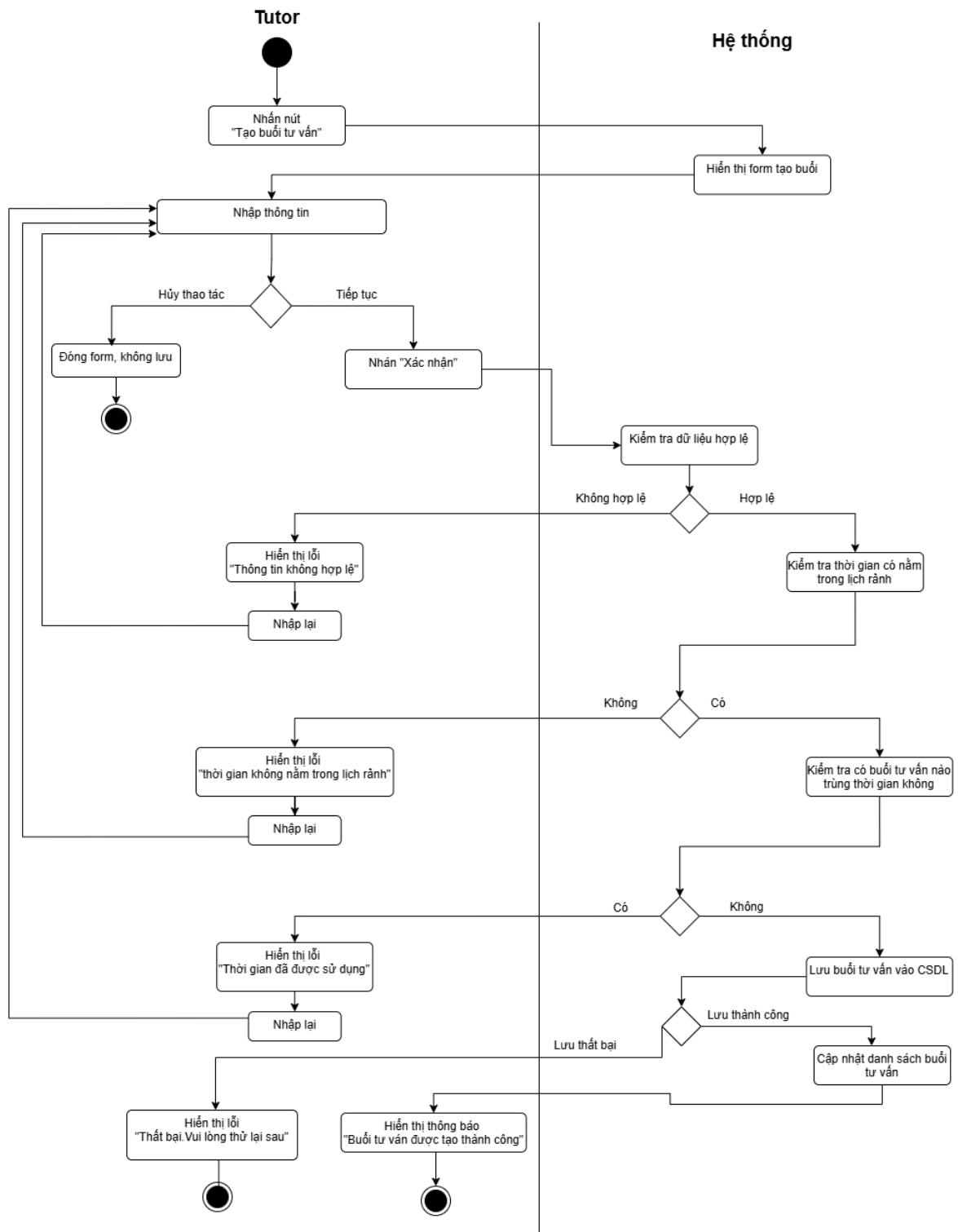
- Nếu hệ thống gặp lỗi cơ sở dữ liệu hoặc mất kết nối khi ghi nhận quyết định, Hệ thống hiển thị thông báo lỗi:

“Xử lý thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật.”
- Hệ thống ghi log lỗi, không thay đổi trạng thái yêu cầu và trạng thái slot.
- Use case kết thúc không thành công.

### 2.1.10 Tạo buổi tư vấn



Hình 22: Sequence Diagram cho Use Case “Tạo buổi tư vấn”



Hình 23: Activity Diagram cho Use Case “Tạo buổi tư vấn”



## Mô tả

### Luồng sự kiện chính: Tạo buổi tư vấn thành công

1. **Tutor** truy cập vào chức năng “Tạo buổi tư vấn” trên giao diện hệ thống.
2. **UI** hiển thị Form **Tạo buổi tư vấn** cho **Tutor** nhập thông tin chi tiết.
3. **Tutor** nhập các thông tin cần thiết, bao gồm: chủ đề, thời gian bắt đầu và kết thúc, hình thức (trực tiếp / trực tuyến), mô tả nội dung và địa điểm (nếu có).
4. Khi hoàn tất, **Tutor** nhấn nút **Xác nhận**. **UI** gửi yêu cầu tạo buổi kèm dữ liệu tới **System**.
5. **System** thực hiện các kiểm tra hợp lệ trên dữ liệu nhận được: kiểm tra trường bắt buộc, định dạng thời gian, và độ dài nội dung.
6. Nếu hợp lệ, **System** truy vấn **Database** để kiểm tra: thời gian có nằm trong lịch rảnh của **Tutor** và có trùng với buổi tư vấn nào khác hay không.
7. Nếu không có xung đột, **System** lưu thông tin buổi tư vấn mới vào **Database**.
8. **System** trả về thông báo thành công cho **UI**.
9. **UI** hiển thị thông báo: “*Tạo buổi tư vấn thành công*” và cập nhật danh sách buổi tư vấn trên giao diện.

### Các luồng phụ và luồng lỗi (Alternative & Error Flows)

#### 1. Thông tin không hợp lệ:

Nếu dữ liệu **Tutor** nhập bị thiếu hoặc sai định dạng (ví dụ: để trống thời gian, nhập sai định dạng giờ), **System** sẽ từ chối xử lý và trả về thông báo lỗi cho **UI**.

**UI** hiển thị: “*Thông tin không hợp lệ, vui lòng kiểm tra lại.*”

Form giữ nguyên để **Tutor** chỉnh sửa và gửi lại.

#### 2. Thời gian không nằm trong lịch rảnh:

Khi kiểm tra **Database**, nếu thời gian nhập không thuộc các khung giờ rảnh đã đăng ký, **System** trả về lỗi cho **UI**.

**UI** hiển thị: “*Thời gian bạn chọn không nằm trong lịch rảnh.*”

**Tutor** chọn lại thời gian hợp lệ.



### 3. Xung đột thời gian (đã có buổi khác):

Nếu **System** phát hiện có buổi tư vấn khác trùng hoặc chồng lấn khung giờ, hệ thống phản hồi lỗi tới **UI**.

**UI** hiển thị: “*Thời gian đã được sử dụng. Vui lòng chọn khung giờ khác.*”

Tutor có thể chỉnh sửa và gửi lại.

### 4. Hủy thao tác:

Nếu **Tutor** nhấn nút **Hủy** trong khi nhập form, **UI** đóng form và không lưu bất kỳ thông tin nào.

Use Case kết thúc tại đây.

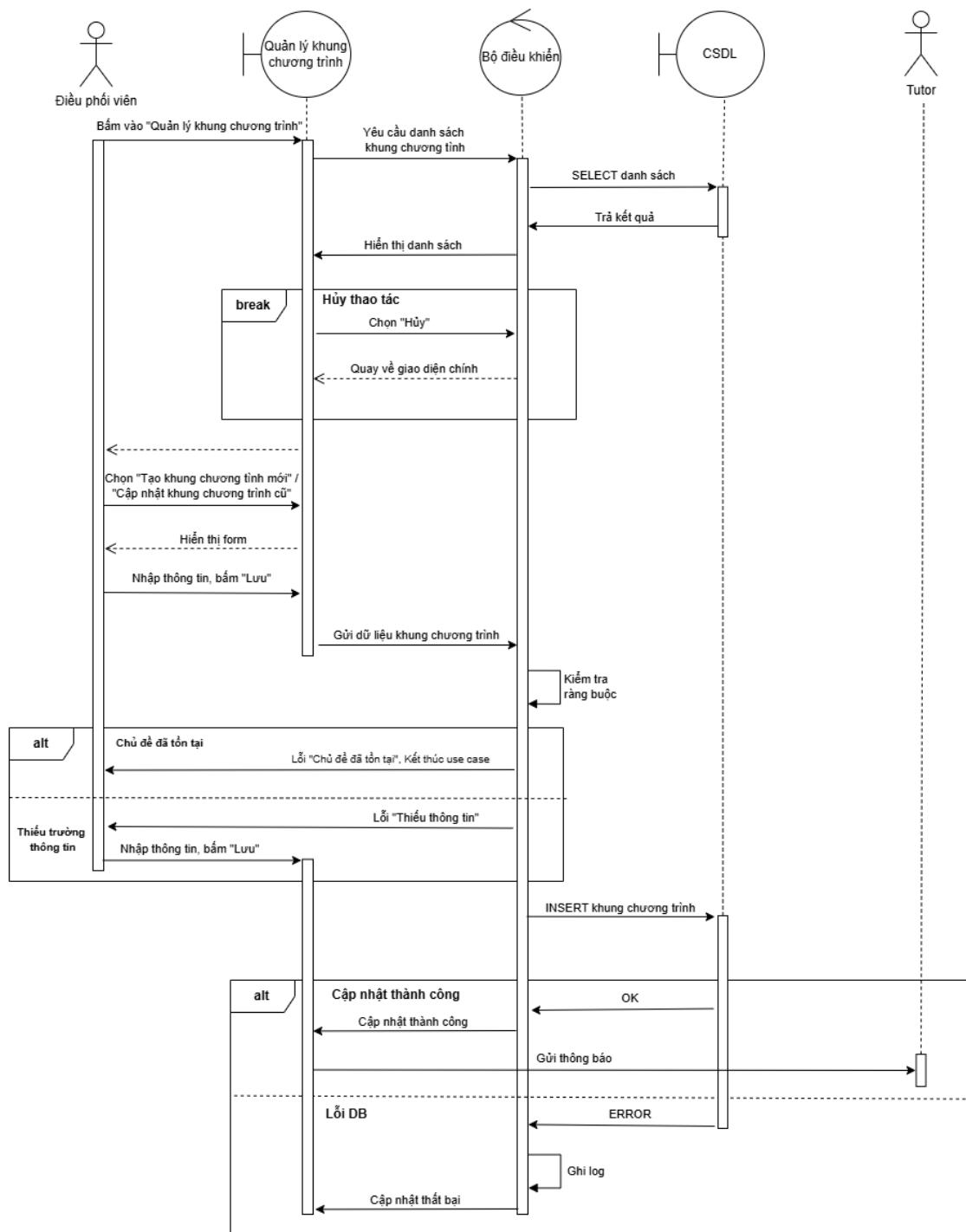
### 5. Lỗi hệ thống / lỗi khi lưu dữ liệu:

Trong quá trình lưu dữ liệu, nếu xảy ra lỗi kết nối hoặc lỗi truy vấn, **System** trả về thông báo lỗi chung.

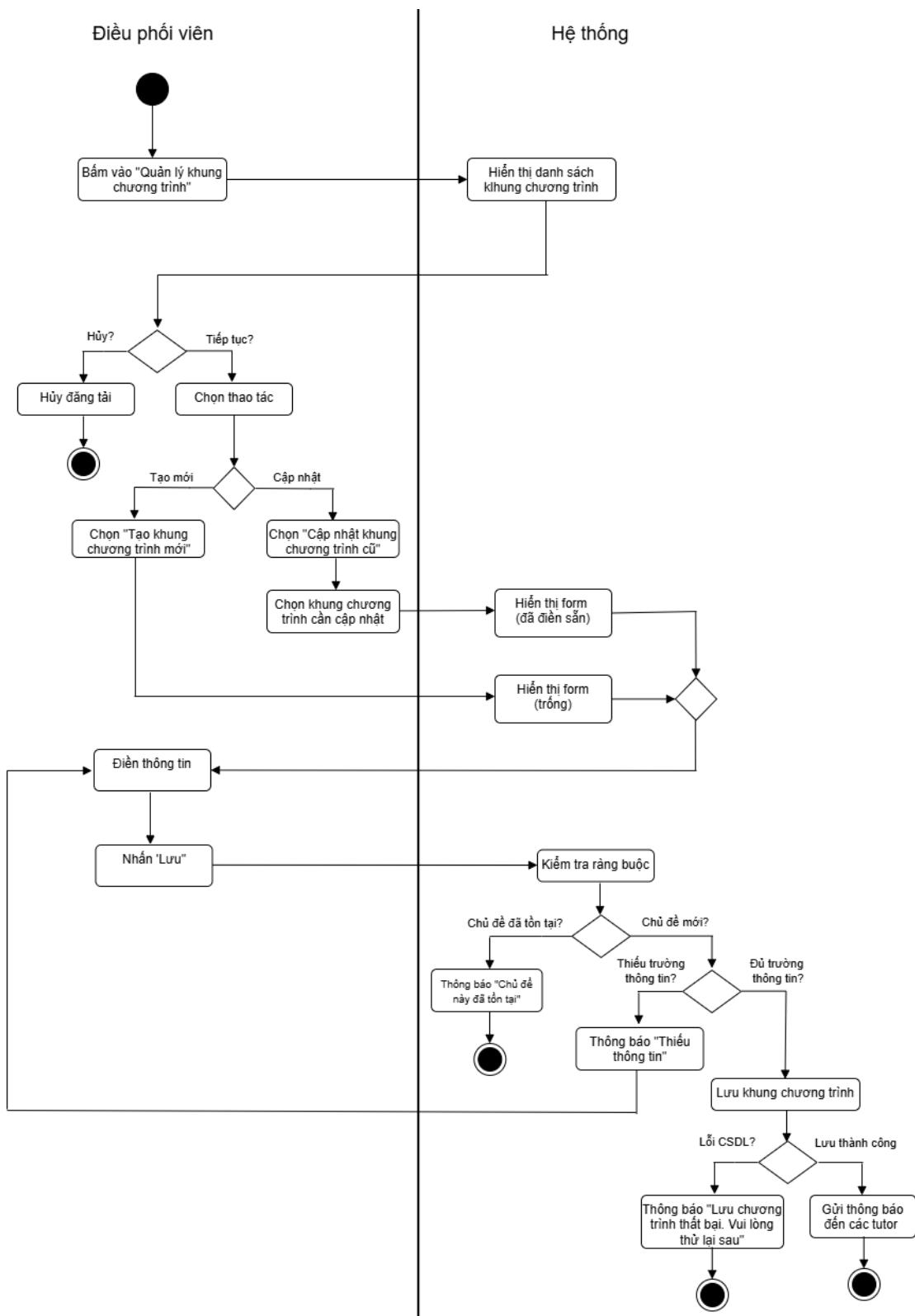
**UI** hiển thị: “*Hệ thống đang gặp sự cố, vui lòng thử lại sau.*”

Use Case kết thúc không thành công.

### 2.1.11 Điều phối khung chương trình chung



Hình 24: Sequence Diagram cho UC Điều phối khung chương trình





**Mô tả: Luồng chính: Cập nhật hoặc tạo khung chương trình thành công**

1. **Điều phối viên** truy cập vào hệ thống và chọn chức năng “Quản lý khung chương trình”.
2. **Hệ thống (UI)** gửi yêu cầu truy vấn danh sách khung chương trình hiện có đến **Bộ điều khiển**.
3. **Bộ điều khiển** thực hiện lệnh SELECT tới **CSDL** và nhận kết quả trả về.
4. **Hệ thống** hiển thị danh sách các khung chương trình hiện có trên giao diện cho điều phối viên.
5. **Điều phối viên** chọn thao tác:
  - Tạo khung chương trình mới, hoặc
  - Cập nhật khung chương trình cũ.
6. Hệ thống hiển thị form nhập thông tin khung chương trình tương ứng (mới hoặc đã điền sẵn).
7. **Điều phối viên** nhập đầy đủ thông tin cần thiết (chủ đề, nội dung, thời lượng, mô tả...) và nhấn nút Lưu.
8. **Hệ thống** nhận dữ liệu, gửi đến **Bộ điều khiển** để kiểm tra ràng buộc:
  - Đủ các trường thông tin bắt buộc.
  - Chủ đề chưa bị trùng.
9. Nếu hợp lệ, **Bộ điều khiển** thực hiện INSERT hoặc UPDATE vào **CSDL**.
10. **CSDL** phản hồi thành công (OK).
11. **Hệ thống** hiển thị thông báo “Cập nhật khung chương trình thành công” và gửi thông báo đến các **Tutor**.
12. Use case kết thúc thành công.

**Luồng phụ và luồng lỗi:**

1. **Điều phối viên** chọn “Hủy” thao tác



- Khi đang ở form tạo/cập nhật, nếu điều phối viên chọn “Hủy”, hệ thống quay lại giao diện chính và không lưu bất kỳ thay đổi nào.
- Use case kết thúc sớm.

## 2. Thiếu thông tin

- Khi điều phối viên nhấn “Lưu” nhưng chưa nhập đủ các trường bắt buộc (chủ đề, nội dung...), **Bộ điều khiển** phát hiện lỗi ràng buộc.
- Hệ thống hiển thị thông báo lỗi:  
“Thiếu thông tin, vui lòng nhập đầy đủ các trường bắt buộc.”
- Giao diện quay lại form nhập dữ liệu để người dùng bổ sung.

## 3. Chủ đề đã tồn tại

- Khi thực hiện kiểm tra ràng buộc, nếu phát hiện chủ đề nhập vào đã tồn tại trong CSDL, hệ thống hiển thị thông báo:  
“Chủ đề này đã tồn tại, vui lòng chọn tên khác.”
- Use case kết thúc mà không lưu dữ liệu.

## 4. Lỗi CSDL (Database Error)

- Trong quá trình lưu dữ liệu, nếu xảy ra lỗi như mất kết nối hoặc lỗi truy vấn, **Bộ điều khiển** nhận phản hồi ERROR.
- Hệ thống ghi log lỗi vào bộ ghi log nội bộ.
- UI hiển thị thông báo:  
“Lưu khung chương trình thất bại. Vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật.”
- Use case kết thúc không thành công.



## 2.2 Mockup

### 2.2.1 Trang đăng nhập

Hình 26: Trang login của hệ thống

Hình 27: Sau khi ấn đăng nhập hệ thống chuyển sang HCMUT\_SSO



### 2.2.2 Quên mật khẩu

Hình 28: Trang quên mật khẩu

### 2.2.3 Đăng ký tutor

Sau khi đăng nhập sinh viên phải đăng ký Tutor

Hình 29: Bước 1: Nhập thông tin



The screenshot shows a user interface for selecting a tutor. At the top, there is a navigation bar with icons for a graduation cap, a bell, and a user profile labeled 'Nguyễn Văn A'. Below the navigation bar, a progress bar indicates three steps: 'Nhập thông tin' (Step 1), 'Chọn tutor' (Step 2, currently selected), and 'Xác nhận' (Step 3). On the left, a sidebar menu includes 'Trang chủ', 'Buổi gặp mặt', 'Khóa học', and 'Hồ sơ cá nhân'. The main content area displays three tutor profiles in cards:

- Nguyễn Lý Ngôn Ngữ Lập Trình**: Rating 4.8k, HK251, Students 13/15. Enroll button.
- Nguyễn Lý Ngôn Ngữ Lập Trình**: Rating 3.8k, HK251, Students 10/15. Enroll button.
- Nguyễn Lý Ngôn Ngữ Lập Trình**: Rating 3.2k, HK251, Students 11/15. Enroll button.

At the bottom of the page, there is a footer section with social media icons for Facebook, Twitter, and Instagram, followed by '#TUTOR SUPPORT SYSTEM'. It also includes links for Home, About, Service, Contact us, Policy, Account, and a 'GET IN TOUCH' section with address, email, and phone number. The footer also contains a 'Terms of Use - Privacy Policy' link.

Hình 30: Bước 2: Chọn Tutor



The screenshot shows a user interface for booking a meeting. On the left, there is a sidebar with icons for 'Trang chủ' (Home), 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), 'Tutor', and 'Hồ sơ cá nhân' (Personal Profile). The main area has a blue header bar with a back arrow, the text 'Quay lại', and a user profile for 'Nguyễn Văn A'. Below this is a horizontal progress bar with three steps: 'Nhập thông tin' (Step 1), 'Chọn tutor' (Step 2), and 'Xác nhận' (Step 3). Step 2 is highlighted with a green circle. A dashed box encloses the central content area. Inside this box, there is a circular icon with a person's face and the text 'HỆ THỐNG XÁC NHẬN ĐĂNG KÝ THÀNH CÔNG' (Booking Confirmation System) followed by 'NGUYỄN LÝ NGÔN NGỮ LẬP TRÌNH (CO3005)'. Below this, it lists: 'Sinh viên: Nguyễn Văn A', 'MSSV: 23xxxx', and 'Tutor: Nguyễn Thị B'. Underneath, there is a section titled 'LƯU Ý' (Attention) with two bullet points: '• Yêu cầu sẽ ở trạng thái "Chờ duyệt" nếu Tutor cần xác nhận.' and '• Hủy trước 12 giờ để không mất slot.'. At the bottom right of this box is a red 'Quay lại' button.

Hình 31: Bước 3: Chờ xác nhận



#### 2.2.4 Sinh viên

The screenshot shows the student's dashboard. At the top right, there is a notification bell icon and the name 'Nguyễn Văn A'. On the left, a sidebar menu includes 'Trang chủ' (selected), 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), and 'Hồ sơ cá nhân' (Personal Profile). The main content area features a welcome message 'Welcome back, Nguyễn Văn A!' with a background illustration of books and a smartphone. Below this is a blue button labeled 'Đăng kí tutor'. At the bottom, there is a footer section with social media icons (#TUTOR SUPPORT SYSTEM), navigation links (Home, About, Service, Contact us, Policy, Account), address information (268 Lý Thường Kiệt, Phường 14, Quận 10, TP. Hồ Chí Minh), email (tutorsupport@hcmut.edu.vn), phone number (+84363699696), and links to 'Terms of Use' and 'Privacy Policy'.

Hình 32: Trang chủ hệ thống của sinh viên



The screenshot shows a student profile for 'Nguyễn Văn A'. Personal details include: Họ và tên (Name): Nguyễn Văn A; Mã sinh viên (Student ID): 1234567; Khoa/Bộ môn (Faculty/Department): Khoa học và Kỹ thuật máy tính; Chuyên ngành (Major): Khoa học máy tính; Email: abc.zzzz@hcmut.edu.vn; Số điện thoại (Phone Number): 09955776613. Below this is a table titled 'Lịch sử đăng ký' (Registration History) showing two entries:

Ngày	Môn	Tutor	Trạng thái
20-09-2025	Toán rời rạc	Nguyễn Thị B	Hoàn tất
19-09-2025	Giải tích	Trần Văn C	Hủy

At the bottom of the page, there is a footer for '#TUTOR SUPPORT SYSTEM' with links to Home, About, Service, Contact us, Policy, Account, and social media icons. It also includes a 'GET IN TOUCH' section with address, email, and phone number.

Hình 33: Xem thông tin cá nhân

The screenshot shows a booking interface. On the left, there is a sidebar with navigation links: Trang chủ, Buổi gặp mặt (selected), Khóa học, and Hồ sơ cá nhân. In the center, there is a 'Danh sách buổi gặp mặt' (List of meetings) button, a 'Đặt lịch hẹn' (Schedule) button, and a 'Đăng kí buổi tư vấn' (Register consultation) button. Below these are sections for 'Tutor của bạn' (Your tutor) showing 'Trần Văn B' (Khoa: Khoa học và Kỹ thuật máy tính) and a booking calendar for October 17, 2025. The calendar shows the week from Sunday to Saturday with specific times highlighted. To the right of the calendar is a booking form with fields for Ngày (Date), Giờ (Time), and Nội dung (Content). At the bottom is a 'Gửi yêu cầu' (Send request) button. The footer is identical to the one in Figure 33.

Hình 34: Đặt lịch hẹn với Tutor



The screenshot shows the 'Tutor Support System' interface. At the top right, there is a user profile for 'Nguyễn Văn A'. The main content area displays a list of scheduled sessions under the heading 'Các buổi tư vấn sắp tới'. Each session entry includes the tutor's name (Trần Văn B), subject, date, time, and a 'Đăng kí' (Sign up) button. The footer contains social media icons, a navigation menu with 'Home', 'About', 'Service', 'Contact us', 'Policy', and 'Account', and links for 'GET IN TOUCH'.

Hình 35: Đăng ký buổi tư vấn

The screenshot shows the 'Tutor Support System' interface. At the top right, there is a user profile for 'Trần Văn B'. The main content area displays a list of booked sessions under the heading 'Buổi gặp mặt của tôi'. Each session entry includes the tutor's name (Trần Văn B), subject, date, time, and a status indicator (e.g., 'BUỔI HỘI THẢO', 'BUỔI HẸN') followed by a 'Bỏ điện thoại' (End call) or 'Hủy' (Cancel) button. The footer contains social media icons, a navigation menu with 'Home', 'About', 'Service', 'Contact us', 'Policy', and 'Account', and links for 'GET IN TOUCH'.

Hình 36: Danh sách buổi tư vấn



The screenshot shows the 'Meeting' section of the system. A specific meeting entry for 'Hội thảo chuyên đề Tri tuệ nhân tạo – Viettel AI Race 2025' on October 27, 2025, is selected. A confirmation dialog box asks if the user wants to cancel the meeting. The user has chosen to cancel it. Below this, another meeting entry for 'Báo cáo tiền đồ đồ án tổng hợp' on October 30, 2025, is shown with similar cancellation options.

Hình 37: Hủy buổi gặp mặt

The screenshot shows the 'List of documents' section. It displays four identical PDF files named 'slide-01.pdf' (94 KB) with 'Xem' (View) and 'Tải về' (Download) buttons. The footer of the page includes social media links and contact information for the Tutor Support System.

Hình 38: Danh sách tài liệu



Nội dung: Hội thảo chuyên đề Trí tuệ nhân tạo – Viettel AI Race 2025

Thời gian: 14:00 - 17:00, ngày 27/10/2025

Chất lượng buổi học: ★★★★☆

Nhận xét:

Gửi

Lịch sử phản hồi:

2025-10-26  
★ 5/5 — Giải bài mẫu chi tiết

Hình 39: Phản hồi chất lượng

### 2.2.5 Tutor

Họ và tên: Trần Văn B

Mã giảng viên: 1234567

Khoa/Bộ môn: Khoa học và kỹ thuật máy tính

Chuyên ngành: Khoa học máy tính

Email: abc.zzzz@hcmut.edu.vn

Số điện thoại: 0995577613

Hình 40: Hồ sơ cá nhân của tutor



#TUTOR SUPPORT SYSTEM

GET IN TOUCH

Home Contact us

About Policy

Service Account

© 2025 Tutor Support System - Trường Đại Học Bách Khoa TP.HCM.

Terms of Use - Private Policy

Hình 41: Tạo buổi tư vấn

#TUTOR SUPPORT SYSTEM

GET IN TOUCH

Home Contact us

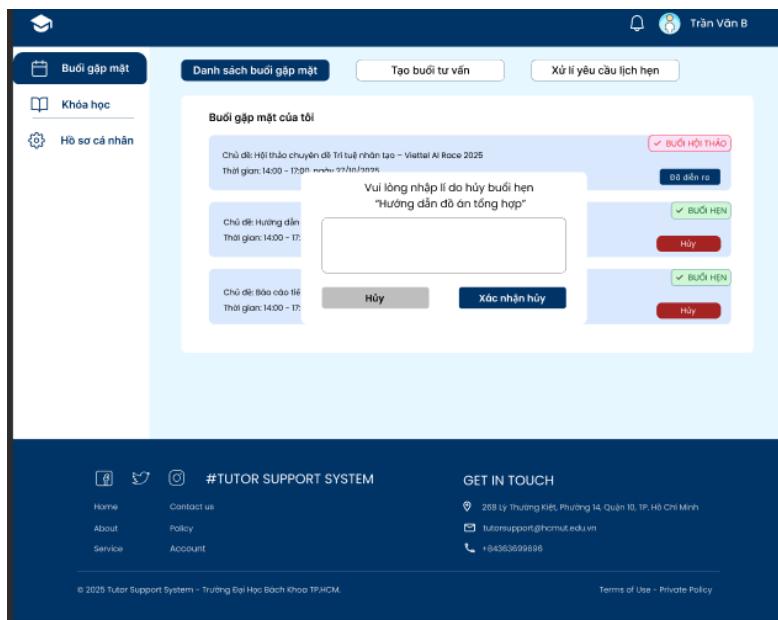
About Policy

Service Account

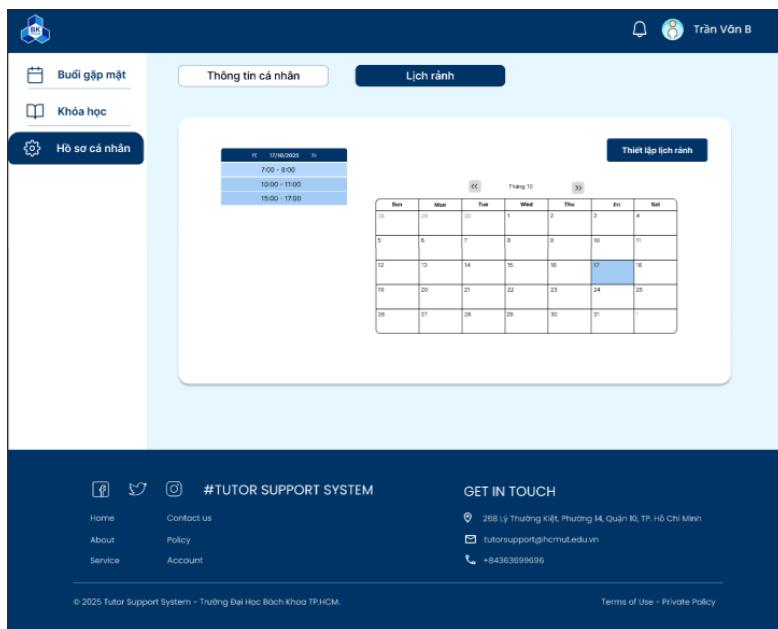
© 2025 Tutor Support System - Trường Đại Học Bách Khoa TP.HCM.

Terms of Use - Private Policy

Hình 42: Danh sách buổi gặp mặt



Hình 43: Hủy buổi gặp mặt



Hình 44: Thiết lập lịch rảnh



The screenshot shows a user interface for managing student records. At the top right, there is a profile icon for 'Trần Văn B' and a notification bell. The main content area has three tabs: 'Buổi gặp mặt' (Meeting), 'Danh sách tài liệu' (List of documents), and 'Danh sách sinh viên' (List of students). The 'Danh sách sinh viên' tab is active, displaying a list of five students: Nguyễn Văn A, Nguyễn Văn C, Nguyễn Văn C, Nguyễn Văn C, and Nguyễn Văn C. Each student entry has a 'Chọn' (Select) button to its right. Below this list, there is a footer section with social media icons, the system name '#TUTOR SUPPORT SYSTEM', and links to 'Home', 'About', 'Service', 'Contact us', 'Policy', and 'Account'. On the right side of the footer, there is a 'GET IN TOUCH' section with address, email, and phone number information.

Hình 45: Danh sách sinh viên

The screenshot shows a form for recording student progress. The left sidebar has the same navigation as in Figure 45. The main form contains fields for 'Họ và tên sinh viên' (Student name) with 'Nguyễn Văn A' entered, 'Nội dung' (Content) with 'Báo cáo tiến độ dự án ABCD' entered, 'Thời gian' (Time) with a date input field, 'Dánh giá' (Grade) with an empty input field, and 'Ghi chú' (Remarks) with an empty input field. At the bottom right of the form are two buttons: 'Xác nhận' (Confirm) and 'Hủy' (Cancel). The footer is identical to Figure 45.

Hình 46: Ghi nhận tiến độ sinh viên



The screenshot shows a user interface for managing scheduled meetings. At the top, there are three buttons: 'Danh sách buổi gặp mặt' (List of scheduled meetings), 'Tạo buổi tư vấn' (Create a consultation session), and 'Xử lý yêu cầu lịch hẹn' (Handle scheduling requests). On the left, a sidebar has icons for 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), and 'Hồ sơ cá nhân' (Personal profile). The main area displays a list of scheduling requests:

- Nội dung: Báo cáo tiến độ dự án ABCD**  
Thứ bảy, 25/10/2025 11:00 PM từ Nguyễn Văn A Xem chi tiết
- Nội dung: Hướng dẫn đồ án**  
Thứ bảy, 25/10/2025 11:00 PM từ Nguyễn Văn B Xem chi tiết
- Nội dung: Giải đáp thắc mắc về mô hình ABCDF**  
Chủ nhật, 26/10/2025 11:00 PM từ Nguyễn Văn C Xem chi tiết

At the bottom of the page, there is a footer with social media links (#TUTOR SUPPORT SYSTEM) and contact information.

Hình 47: Danh sách buổi hẹn do sinh viên yêu cầu

The screenshot shows a form for creating a new scheduling request. The top bar includes buttons for 'Danh sách buổi gặp mặt', 'Tạo buổi tư vấn', and 'Xử lý yêu cầu lịch hẹn'. The sidebar on the left is identical to the previous screenshot. The main form fields are:

- Họ tên sinh viên:** Nguyễn Văn A
- Nội dung:** Báo cáo đồ án
- Ngày:** dd/mm/yyyy  **Khung giờ:** h:mm
- Từ chối yêu cầu:** Nhập lý do từ chối...
- Thiết lập lại slot rảnh?**  Có  Không

At the bottom, there are two buttons: 'Từ chối' (Reject) and 'Phê duyệt' (Approve).

The footer contains social media links and contact information.

Hình 48: Xử lý yêu cầu đặt lịch hẹn



Hình 49: Đăng tải tài liệu

Hình 50: Danh sách tài liệu



## 2.2.6 Điều phối viên

The screenshot shows a web application interface for managing shared programs. At the top right, there is a user profile for 'Trần Văn B'. On the left, a sidebar has a 'Khung chương trình' (Program Frame) section with a 'Gửi báo cáo' (Report Submission) button. The main content area is titled 'Danh sách khung chương trình' (List of Program Frames) and displays five identical items, each with the subject 'Hội thảo chuyên đề Trí tuệ nhân tạo – Viettel AI Race 2025'. At the bottom of this list are two buttons: 'Tạo chương trình mới' (Create new program) and 'Cập nhật chương trình cũ' (Update old program). The footer contains social media icons, navigation links (Home, About, Service), contact information (Address: 268 Lý Thường Kiệt, Phường 14, Quận 10, TP. Hồ Chí Minh; Email: tutorsupport@hcmut.edu.vn; Phone: +84363699696), and legal links (Terms of Use - Private Policy).

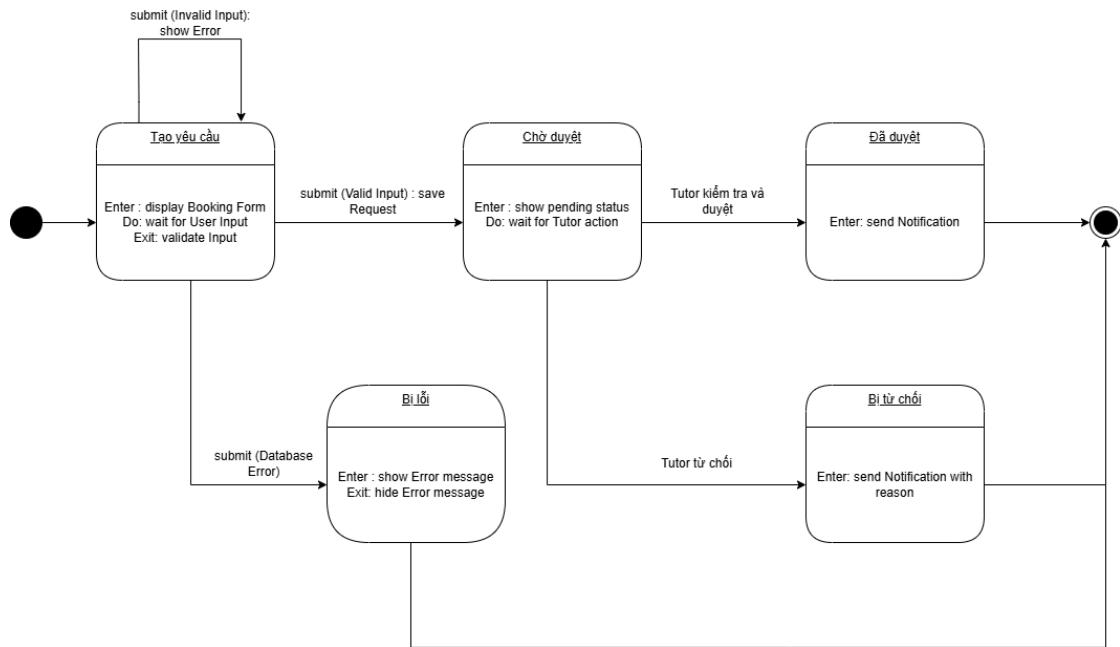
Hình 51: Quản lý khung chương trình chung cho các Tutor

This screenshot shows the 'Tutor Support System' interface for creating a new program frame. The left sidebar has a 'Khung chương trình' (Program Frame) section with a 'Gửi báo cáo' (Report Submission) button. The main area is titled 'Danh sách khung chương trình' (List of Program Frames) and contains a form with fields for 'Chủ đề' (Subject), 'Mô tả' (Description), 'Thời gian' (Time), 'Chuẩn đầu ra' (Output Standard), 'Tài liệu' (Materials), and 'Ghi chú' (Notes). Below the form are 'Hủy' (Cancel) and 'Lưu' (Save) buttons. The footer is identical to the previous screenshot, featuring social media icons, navigation links, contact information, and legal links.

Hình 52: Quản lý khung chương trình chung cho các Tutor

## 2.3 State-chart diagram

### 2.3.1 Yêu cầu đặt lịch hẹn



Hình 53: State Diagram cho yêu cầu đặt lịch hẹn

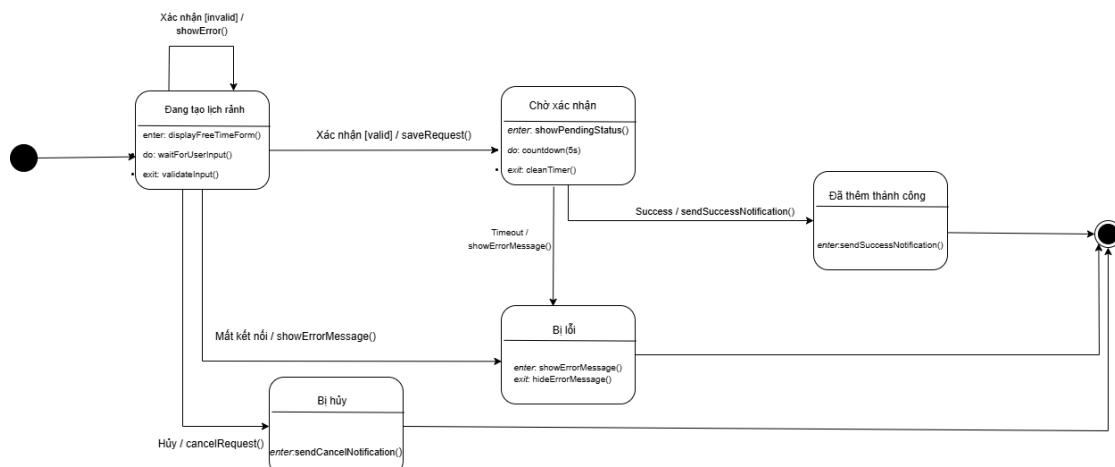
Bảng 24: Bảng mô tả trạng thái

Trạng thái	Mô tả
Dang tạo yêu cầu	Sinh viên đang nhập thông tin để tạo yêu cầu.
Chờ duyệt	Yêu cầu đã được gửi thành công và đang chờ Tutor duyệt.
Đã duyệt	Yêu cầu được Tutor duyệt.
Bi từ chối	Yêu cầu đặt lịch bị từ chối vì lý do nào đó.
Bi lỗi	Xảy ra lỗi hệ thống.

Bảng 25: Bảng Stimulus (Event/Transition)

Stimulus	Mô tả
submit (valid Input): save Request	Sinh viên gửi yêu cầu với dữ liệu hợp lệ, hệ thống lưu yêu cầu vào cơ sở dữ liệu và chuyển sang trạng thái chờ duyệt.
submit (invalid Input): show Error	Sinh viên gửi yêu cầu nhưng dữ liệu không hợp lệ. Hệ thống hiển thị lỗi và giữ nguyên trạng thái tạo yêu cầu.
Tutor từ chối	Tutor từ chối yêu cầu đặt lịch của sinh viên và cung cấp lý do chính đáng. Hệ thống cập nhật trạng thái thành "Đã hủy".
Tutor kiểm tra và duyệt	Tutor truy cập vào hệ thống và duyệt yêu cầu đặt lịch của sinh viên. Hệ thống cập nhật trạng thái thành "Đã duyệt"
submit (Database Error)	Lỗi hệ thống khi lưu dữ liệu vào cơ sở dữ liệu.

### 2.3.2 Thiết lập lịch rảnh



Hình 54: State Diagram cho thiết lập lịch rảnh



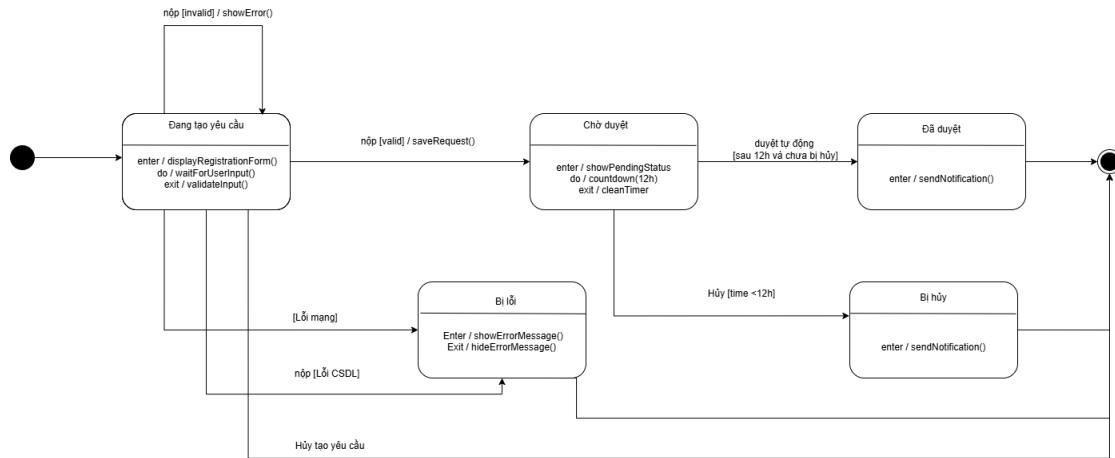
Bảng 26: Bảng mô tả trạng thái trong UC thiết lập lịch rảnh

Trạng thái	Mô tả
Dang tạo lịch rảnh	Sinh viên đang nhập thông tin khung giờ rảnh và hệ thống hiển thị form để người dùng điền. Hệ thống chờ người dùng xác nhận.
Chờ xác nhận	Hệ thống đã nhận yêu cầu hợp lệ và đang xử lý. Hệ thống hiển thị trạng thái chờ trong thời gian ngắn (đêm ngược).
Đã thêm thành công	Yêu cầu tạo lịch rảnh đã được lưu thành công. Hệ thống gửi thông báo thành công đến người dùng.
Bị lỗi	Quá trình gửi yêu cầu hoặc xử lý gặp lỗi (timeout, mất kết nối, lỗi hệ thống). Hệ thống hiển thị thông báo lỗi và yêu cầu thực hiện lại.
Bị huỷ	Người dùng tự hủy thao tác hoặc hệ thống hủy yêu cầu do lỗi nghiêm trọng. Thông báo hủy được gửi đến người dùng.

Bảng 27: Bảng Stimulus (Event/Transition) trong UC thiết lập lịch rảnh

Stimulus / Event	Mô tả
Xác nhận (invalid) : showError()	Người dùng nhấn nút xác nhận nhưng dữ liệu nhập không hợp lệ. Hệ thống hiển thị lỗi và giữ nguyên trạng thái Dang tạo lịch rảnh.
Xác nhận (valid) : saveRequest()	Người dùng gửi yêu cầu hợp lệ. Hệ thống lưu yêu cầu và chuyển sang trạng thái Chờ xác nhận.
Timeout : showErrorMessage()	Trong trạng thái Chờ xác nhận, hệ thống bị timeout xử lý. Hệ thống chuyển sang trạng thái Bị lỗi.
Success : sendSuccessNotification()	Yêu cầu đã được xử lý thành công trong trạng thái Chờ xác nhận. Hệ thống chuyển sang trạng thái Đã thêm thành công.
Mất kết nối : showErrorMessage()	Khi đang xử lý, hệ thống mất kết nối. Hệ thống chuyển sang trạng thái Bị lỗi.
Hủy : cancelRequest()	Người dùng chủ động hủy thao tác. Hệ thống chuyển sang trạng thái Bị hủy.

### 2.3.3 Đăng ký Tutor



Hình 55: State Diagram cho yêu cầu đăng ký Tutor

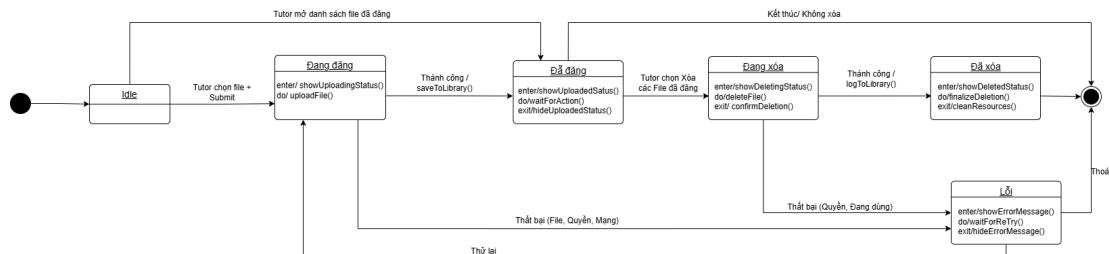
Bảng 28: Bảng mô tả trạng thái

Trạng thái	Mô tả
Đang tạo yêu cầu	Sinh viên đang nhập thông tin để tạo yêu cầu.
Chờ duyệt	Yêu cầu đã được gửi thành công và đang chờ hệ thống tự động duyệt trong vòng 12 giờ.
Đã duyệt	Yêu cầu được hệ thống tự động duyệt sau 12 giờ và không bị sinh viên bấm hủy.
Bị hủy	Sinh viên hủy yêu cầu (trong lúc tạo hoặc trong 12 giờ sau khi gửi).
Bị lỗi	Xảy ra lỗi hệ thống hoặc mất kết nối mạng.

Bảng 29: Bảng Stimulus (Event/Transition)

Stimulus	Mô tả
nộp [valid] / saveRequest()	Sinh viên gửi yêu cầu với dữ liệu hợp lệ, hệ thống lưu yêu cầu vào cơ sở dữ liệu và chuyển sang trạng thái chờ duyệt.
nộp [invalid] / showError()	Sinh viên gửi yêu cầu nhưng dữ liệu không hợp lệ. Hệ thống hiển thị lỗi và giữ nguyên trạng thái tạo yêu cầu.
Hủy tạo yêu cầu	Sinh viên hủy khi đang tạo yêu cầu (yêu cầu chưa được lưu).
Hủy [time < 12h]	Sinh viên hủy yêu cầu sau khi đã gửi (trong 12 giờ). Hệ thống cập nhật trạng thái thành “Đã hủy”.
đuyệt tự động [sau 12h và chưa bị hủy]	Hệ thống tự động duyệt yêu cầu sau 12 giờ nếu sinh viên không hủy.
Lỗi mạng	Lỗi kết nối trong quá trình gửi yêu cầu, hệ thống thông báo lỗi.
nộp [lỗi CSDL]	Lỗi hệ thống khi lưu dữ liệu vào cơ sở dữ liệu.

### 2.3.4 Đăng tải và xóa tài liệu



Hình 56: State Diagram cho quản lý tài liệu (Đăng và Xóa)



Bảng 30: Bảng mô tả trạng thái trong UC Quản lý tài liệu (Đăng và Xóa)

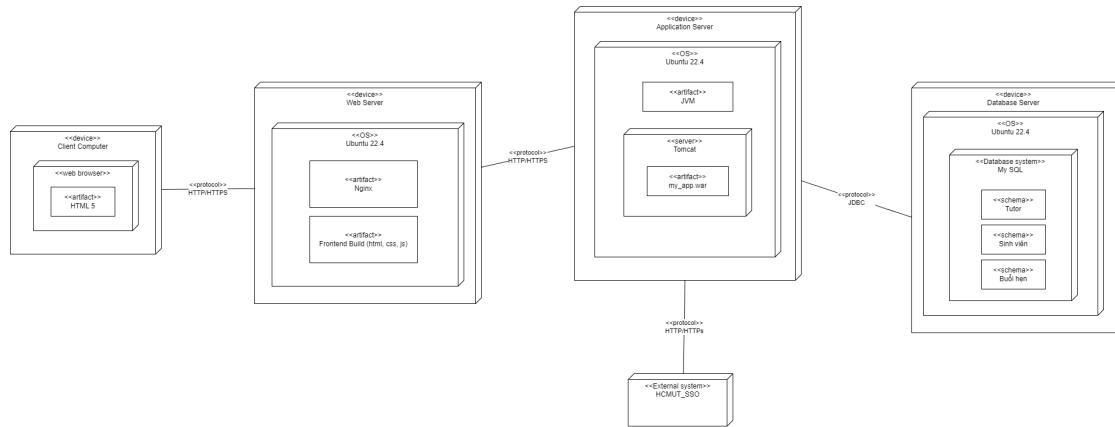
Trạng thái	Mô tả
Idle	Trạng thái ban đầu hoặc sau khi giảng viên mở danh sách tài liệu đã đăng. Hệ thống chờ người dùng chọn tệp để đăng hoặc thao tác tiếp theo.
Đang đăng	Hệ thống đang tải lên tệp mà giảng viên chọn. Trong trạng thái này, hệ thống hiển thị tiến trình tải và thực hiện hành động tải tệp thực tế.
Đã đăng	Tệp đã được tải lên thành công và lưu vào thư viện tài liệu. Hệ thống hiển thị thông báo “Tải lên hoàn tất” và chờ hành động tiếp theo từ giảng viên (như xóa hoặc kết thúc).
Đang xóa	Giảng viên chọn hành động xóa tài liệu đã đăng. Hệ thống hiển thị tiến trình xóa, thực hiện thao tác xóa và xác nhận kết quả trước khi thoát khỏi trạng thái này.
Đã xóa	Tài liệu đã được xóa thành công. Hệ thống dọn dẹp tài nguyên, ghi nhận log hành động và kết thúc use case.
Lỗi	Có lỗi xảy ra trong quá trình tải lên hoặc xóa (ví dụ: lỗi tệp, lỗi quyền truy cập, hoặc sự cố mạng). Hệ thống hiển thị thông báo lỗi, cho phép người dùng thử lại hoặc thoát.

Bảng 31: Bảng mô tả Sự kiện/Chuyển trạng thái trong UC Quản lý tài liệu (Đăng và Xóa)

Sự kiện / Kích thích	Mô tả
Tutor chọn file + Submit	Giảng viên chọn tệp cần đăng và gửi yêu cầu. Hệ thống chuyển từ trạng thái Idle sang Dang đăng.
Thành công / saveToLibrary()	Khi quá trình tải lên hoàn tất thành công, hệ thống lưu tài liệu vào thư viện và chuyển sang trạng thái Đã đăng.
Thất bại (File, Quyền, Mạng)	Nếu có lỗi trong quá trình tải lên, hệ thống chuyển sang trạng thái Lỗi. Người dùng có thể thử lại thao tác.
Tutor mở danh sách file đã đăng	Từ trạng thái Idle, giảng viên có thể mở danh sách tài liệu đã đăng để xem hoặc xóa.
Tutor chọn Xóa các file đã đăng	Giảng viên chủ động chọn hành động “Xóa tài liệu”. Hệ thống chuyển sang trạng thái Đang xóa và thực hiện thao tác xóa.
Thành công / logToLibrary()	Sau khi xóa thành công, hệ thống ghi log và chuyển sang trạng thái Đã xóa.
Thất bại (Quyền, Đang dùng)	Nếu không thể xóa do lỗi quyền hoặc tài liệu đang được sử dụng, hệ thống chuyển sang trạng thái Lỗi.
Thử lại	Từ trạng thái Lỗi, giảng viên chọn “Thử lại” → hệ thống quay lại Dang đăng để thực hiện lại quy trình tải lên.
Thoát	Từ trạng thái Lỗi hoặc sau khi hoàn thành thao tác, giảng viên chọn “Thoát” → hệ thống kết thúc.
Kết thúc / Không xóa	Giảng viên không chọn xóa hoặc kết thúc sau khi tài liệu đã được tải lên → hệ thống kết thúc.

### 3 Task 3

#### 3.1 Deployment view



Hình 57: Deployment Diagram cho hệ thống

##### 1. Client Computer (Máy tính máy khách)

- Thiết bị này đại diện cho phía client, nơi người dùng cuối truy cập vào hệ thống thông qua web browser (trình duyệt web).
- Giao diện phía client được hiện thực bằng HTML 5 (cùng với CSS, JavaScript), là artifact được tải về từ Web Server và render bởi trình duyệt.
- Component trình duyệt đảm nhiệm việc gửi yêu cầu tới máy chủ và hiển thị kết quả cho người dùng.
- Thiết bị này kết nối đến Web Server thông qua giao thức HTTPS (trên nền TCP/IP).

##### 2. Web Server (Máy chủ Web)

- Máy chủ này chạy trên hệ điều hành Ubuntu 22.4 và sử dụng Nginx (một artifact phần mềm) làm máy chủ web.
- Phục vụ nội dung tĩnh: Lưu trữ và cung cấp artifact là Frontend Build (html, css, js) cho Client.
- Reverse Proxy: Nhận yêu cầu từ Client và chuyển tiếp (forward) các yêu cầu nghiệp vụ đến Application Server.



- Máy chủ Web giao tiếp với Client và Application Server qua giao thức HTTP/HTTPS (trên nền TCP/IP).

### 3. Application Server (Máy chủ Ứng dụng)

- Máy chủ này chạy trên hệ điều hành Ubuntu 22.4, phục vụ cho việc xử lý logic nghiệp vụ (backend) của hệ thống.
- Nó sử dụng JVM (Java Virtual Machine) làm môi trường thực thi để chạy server (máy chủ ứng dụng) Tomcat.
- Thành phần chính được triển khai là artifact my\_app.war (Web Application Archive), chứa toàn bộ logic nghiệp vụ (ví dụ: các lớp Service, Controller, và logic truy cập dữ liệu).
- Máy chủ Ứng dụng giao tiếp với Web Server, Database Server và hệ thống HCMUT\_SSO qua các giao thức trên nền TCP/IP.

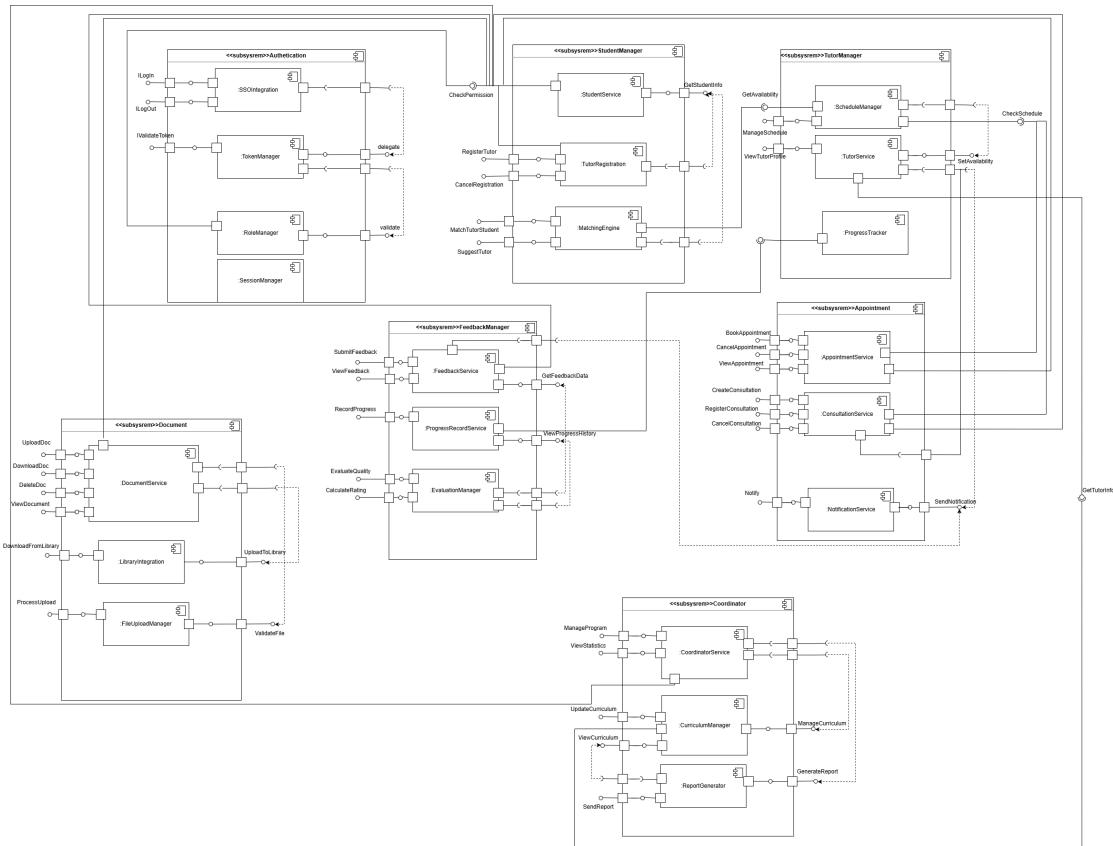
### 4. Database Server (Máy chủ Cơ sở dữ liệu)

- Máy chủ này chạy trên hệ điều hành Ubuntu 22.4 và sử dụng MySQL làm hệ quản trị cơ sở dữ liệu (database system).
- Máy chủ Cơ sở dữ liệu kết nối với Máy chủ Ứng dụng qua giao thức JDBC (trên nền TCP/IP) để trao đổi dữ liệu.

### 5. HCMUT\_SSO (Hệ thống Xác thực)

- Đây là một External system (hệ thống bên ngoài) cung cấp dịch vụ xác thực tập trung (Single Sign-On).
- Nó cung cấp một hệ thống xác thực người dùng, phục vụ cho việc đăng nhập và bảo mật truy cập vào hệ thống, giúp ứng dụng không cần tự quản lý mật khẩu.
- Hệ thống xác thực kết nối với Máy chủ Ứng dụng qua giao thức HTTP/HTTPS (trên nền TCP/IP).

### 3.2 Development/Implementation view



Hình 58: Component Diagram cho hệ thống

#### 3.2.1 Subsystem 1: Authentication

**Mục tiêu.** Subsystem *Authentication* chịu trách nhiệm xác thực người dùng và kiểm soát truy cập dựa trên vai trò (role). Mọi yêu cầu cần đăng nhập, kiểm tra token hoặc kiểm tra quyền đều đi qua subsystem này.

- Các component nội bộ:

- **SSOIntegration:** kết nối tới hệ thống HCMUT\_SSO và HCMUT\_DATACORE để xác thực và đồng bộ thông tin người dùng.
- **TokenManager:** tạo và kiểm tra tính hợp lệ của token đăng nhập.
- **RoleManager:** quản lý role/permission và cung cấp chức năng kiểm tra quyền cho các action nghiệp vụ.



- **SessionManager:** quản lý phiên làm việc (session) của người dùng, ví dụ gia hạn hoặc huỷ session khi logout.

### 3.2.2 Subsystem 2: StudentManager

**Mục tiêu.** Subsystem *StudentManager* quản lý các chức năng phía sinh viên như: đăng ký tutor, tra cứu tutor phù hợp, xem thông tin cá nhân,...

- **Các component nội bộ:**

- **StudentService:** xử lý các thao tác liên quan đến hồ sơ sinh viên và giao diện sinh viên (xem thông tin bản thân, trạng thái tham gia).
- **TutorRegistration:** tiếp nhận và xử lý yêu cầu đăng ký tutor từ sinh viên.
- **MatchingEngine:** thực hiện ghép cặp sinh viên–tutor, đề xuất Tutor

### 3.2.3 Subsystem 3: TutorManager

**Mục tiêu.** Subsystem *TutorManager* quản lý thông tin tutor, lịch rảnh và tiến độ học tập của sinh viên,... do tutor phụ trách.

- **Các component nội bộ:**

- **ScheduleManager:** quản lý lịch rảnh và lịch làm việc của tutor.
- **TutorService:** xử lý các thao tác cập nhật thông tin, lịch rảnh của tutor.
- **ProgressTracker:** lưu trữ và truy vấn tiến độ học tập của sinh viên qua các buổi tư vấn.

### 3.2.4 Subsystem 4: Appointment

**Mục tiêu.** Subsystem *Appointment* quản lý toàn bộ vòng đời của cuộc hẹn và buổi tư vấn (scheduling & consultation), đồng thời cung cấp dịch vụ gửi thông báo chung cho toàn hệ thống.

- **Các component nội bộ:**

- **AppointmentService:** xử lý đặt lịch, xem lịch, huỷ lịch hẹn giữa sinh viên và tutor.
- **ConsultationService:** quản lý các buổi tư vấn theo nhóm hoặc lớp, cho phép tạo, đăng ký và huỷ tham gia.



- **NotificationService:** gửi email hoặc thông báo trong hệ thống tới sinh viên, tutor và coordinator.

### 3.2.5 Subsystem 5: Document

**Mục tiêu.** Subsystem *Document* quản lý tài liệu học tập được sử dụng trong quá trình tư vấn, bao gồm lưu trữ nội bộ và tích hợp với hệ thống thư viện.

- **Các component nội bộ:**

- **DocumentService:** cung cấp CRUD tài liệu trong hệ thống nội bộ.
- **LibraryIntegration:** đóng vai trò adapter kết nối với external system HCMUT\_LIBRARY.
- **FileUploadManager:** xử lý và kiểm tra file upload (định dạng, kích thước, ...).

### 3.2.6 Subsystem 6: FeedbackManager

**Mục tiêu.** Subsystem *FeedbackManager* thu thập feedback từ sinh viên, quản lý dữ liệu tiến độ phục vụ đánh giá và tính toán các chỉ số chất lượng hoạt động tutor.

- **Các component nội bộ.**

- **FeedbackService:** tiếp nhận, lưu trữ và cung cấp dữ liệu feedback.
- **ProgressRecordService:** ghi nhận và truy xuất lịch sử tiến độ (view progress history) phục vụ đánh giá.
- **EvaluationManager:** kết hợp feedback và tiến độ để tính điểm chất lượng và các thống kê đánh giá.

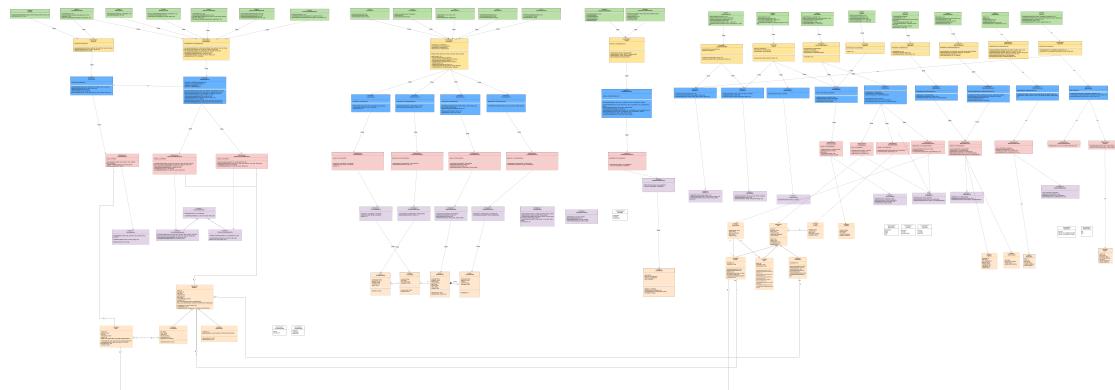
### 3.2.7 Subsystem 7: Coordinator

**Mục tiêu.** Subsystem *Coordinator* chịu trách nhiệm điều phối khung chương trình chung (common program framework) cho hoạt động tutor và tổng hợp, gửi các báo cáo thống kê.

- **Các component nội bộ:**

- **CoordinatorService:** cung cấp API chính dành cho vai trò coordinator.
- **CurriculumManager:** quản lý và cập nhật khung chương trình chung, bao gồm cấu trúc buổi tư vấn, các chủ đề và quy tắc áp dụng.
- **ReportGenerator:** thu thập số liệu từ các subsystem khác và tạo báo cáo tổng hợp.

### 3.3 Class diagram



Hình 59: Class Diagram cho hệ thống

Xem rõ hơn [tại đây](#).

### 3.4 Design Pattern áp dụng

#### 3.4.1 MVC mở rộng

Hệ thống được xây dựng theo mô hình MVC mở rộng 5 tầng (Multi-layer MVC):

View → Controller (API) → Service ((Interface + Implementation)) → Repository (Interface +  
Implementation)  
→ Entity (Model)

So với MVC truyền thống (Model – View – Controller), phiên bản mở rộng bổ sung thêm hai tầng abstraction:

- **Service Layer** (Interface + Implementation)
- **Repository Layer** (Interface + Implementation)

Hệ thống được xây dựng theo mẫu thiết kế MVC mở rộng (Multi-layer MVC), trong đó mỗi nhóm lớp được tổ chức theo đúng vai trò của từng tầng trong kiến trúc. So với MVC truyền thống, hệ thống bổ sung thêm tầng Service và Repository để tách biệt rõ UI, điều phối, nghiệp vụ và truy xuất dữ liệu.



- Ở tầng View, các lớp như AppointmentManagementUI, AppointmentBookingUI, StudentMeetingListUI, ... chỉ chịu trách nhiệm hiển thị thông tin và nhận thao tác từ người dùng (chọn buổi hẹn, gửi yêu cầu hủy, xem chi tiết...). Tầng này không chứa bất kỳ logic nghiệp vụ nào, đảm bảo nguyên tắc Separation of Concerns.
- Tầng Controller như TutorSchedulingAPI, ... tiếp nhận request từ View/FE, thực hiện kiểm tra đầu vào cơ bản và chuyển tiếp yêu cầu xuống Service tương ứng. Controller hoàn toàn không xử lý nghiệp vụ mà chỉ đóng vai trò điều phối luồng dữ liệu giữa View và Service.
- Tầng Service là nơi chứa toàn bộ nghiệp vụ cốt lõi. Gồm interface như ITutorSchedulingService,... và lớp triển khai như TutorSchedulingService,... Tại đây hệ thống xử lý các hành động chặng hạn duyệt lịch hẹn (approveAppointment), từ chối lịch hẹn (rejectAppointment), hủy buổi gặp (cancelMeeting), xem chi tiết (viewMeetingDetails, viewAppointmentDetails).... Việc Service tách thành interface + implementation giúp controller không phụ thuộc vào cách nghiệp vụ được cài đặt, dễ mở rộng và dễ thay thế.
- Tầng Repository gồm interface như IMeetingRepository,... và lớp triển khai như MeetingRepository,... chịu trách nhiệm cung cấp dữ liệu cho Service. Repository lưu trữ danh sách Meeting/Appointment/Consultation dưới dạng bộ nhớ tạm (vì phiên bản này không dùng DB). Tầng này mô phỏng việc truy xuất dữ liệu như một kho dữ liệu thực sự, giúp Service không cần biết dữ liệu được lấy từ đâu.
- Cuối cùng, tầng Entity (Model) gồm các lớp dữ liệu như Meeting, Appointment, Consultation,... Đây là các cấu trúc đối tượng thuần, dùng để biểu diễn thông tin buổi hẹn và được luân chuyển giữa các tầng.

Việc áp dụng MVC mở rộng trong dự án giúp hệ thống phân tách rõ ràng vai trò từng lớp, giảm sự phụ thuộc giữa các tầng, thuận lợi cho bảo trì, mở rộng chức năng và kiểm thử. Kiến trúc này đồng thời phản ánh đúng cách phát triển ứng dụng hiện đại theo chuẩn Spring Boot.

### 3.4.2 Facade

#### Nhận diện Facade

Trong hệ thống, interface LearningFacade được sử dụng như một **Facade Pattern**. Facade đóng vai trò là điểm truy cập duy nhất cho client như sinh viên, tutor hoặc nhóm hệ thống khác.



Qua Facade, chi tiết phức tạp của các service được ẩn đi và việc tương tác trở nên đơn giản hơn.

Client chỉ làm việc với các phương thức:

- `addCourse(dto)`
- `enroll(dto)`
- `submitRequest(dto)`
- `submitFeedback(dto)`
- `getCourses()`

#### Các class bị giấu

Facade che giấu các lớp nghiệp vụ sau:

- `CourseService`
- `EnrollmentService`
- `RequestService`
- `FeedbackService`

Ngoài ra, mẫu thiết kế Facade được thể hiện thông qua lớp `TutorRegistrationFacade`. Lớp này đóng vai trò như một điểm truy cập duy nhất, cung cấp một giao diện đơn giản cho phía client (controller) khi làm việc với nhiều service phức tạp bên dưới. Thay vì để controller trực tiếp tương tác và phối hợp với nhiều lớp xử lý khác nhau, `TutorRegistrationFacade` chịu trách nhiệm tập trung các thao tác liên quan đến đăng ký, hủy đăng ký và chấp nhận yêu cầu tư vấn.

#### Lý do sử dụng Facade

1. Ẩn bớt sự phức tạp của nhiều tầng logic.
2. Đơn giản hóa giao tiếp giữa client và hệ thống.
3. Dễ bảo trì do chỉ cần thay đổi trong Facade khi logic thay đổi.
4. Tăng tính bảo mật vì client không truy cập trực tiếp vào tầng nghiệp vụ.
5. Hỗ trợ tích hợp giữa các nhóm phát triển dễ dàng hơn.



### 3.4.3 Strategy Pattern

#### Nhận diện Strategy Pattern

Trong hệ thống, `SlotOperationStrategy` được sử dụng như một **Strategy Pattern**. Mỗi loại nghiệp vụ xử lý slot (ghi đè – đặt – huỷ) được đóng gói thành một chiến lược riêng, giúp `FreeSlotService` tách biệt logic và dễ mở rộng.

#### Cấu trúc áp dụng Strategy

- **Interface Strategy:** `SlotOperationStrategy`
- **Concrete Strategies:**
  - `OverwriteStrategy`
  - `ReserveStrategy`
  - `ReleaseStrategy`
- **Context:** `FreeSlotService`

**Vai trò Client / Context** `FreeSlotService` đóng vai trò là Context, lựa chọn strategy phù hợp và gọi `execute()` mà không cần biết chi tiết xử lý.

#### Interface: SlotOperationStrategy

- `execute(tutorId: Long, request: FreeSlotRequest): void` – Thực hiện nghiệp vụ xử lý slot tùy theo strategy được chọn.

#### Chi tiết các Concrete Strategy

- **OverwriteStrategy**
  - `execute(...)` – Ghi đè slot, xử lý va chạm và cập nhật dữ liệu.
  - `mergeInputRanges(...)` – Gộp các khoảng thời gian liền kề hoặc chồng lấn.
  - `splitRangeByAppointments(...)` – Cắt slot rảnh lớn thành các slot nhỏ dựa trên lịch hẹn.
- **ReleaseStrategy**
  - `execute(...)` – Chuyển slot đã đặt thành rảnh.



- `mergeTutorSlots(...)` – Hợp các slot rảnh liền nhau.

- **ReserveStrategy**

- `execute(...)` – Tìm slot rảnh phù hợp và cắt phần thời gian bận ra khỏi slot gốc.

### Lý do sử dụng Strategy Pattern

1. **Tách biệt logic phức tạp:** Mỗi nghiệp vụ xử lý slot được viết trong class riêng, giúp code rõ ràng và dễ theo dõi.
2. **Tuân thủ Open/Closed Principle:** Khi thêm nghiệp vụ mới (ví dụ “Khoá lịch tạm thời”), chỉ cần tạo strategy mới mà không chỉnh sửa lớp `FreeSlotService`.
3. **Dễ bảo trì và mở rộng:** Client chỉ gọi `execute()` từ interface, không quan tâm chi tiết xử lý.

## 3.5 Mô tả phương thức

### 3.5.1 View Method

#### MeetingListUI

- `displayMeetingList(meetings: List<Meeting>): void`  
Hiển thị danh sách các buổi gặp mặt của người dùng.
- `selectMeeting(meetingId: int): void`  
Chọn một buổi gặp mặt để xem chi tiết hoặc thao tác (hủy).

#### MeetingCancelUI

- `displayCancellableMeetings(meetings: List<Meeting>): void`  
Hiển thị danh sách các buổi gặp mặt có thể hủy (còn chưa diễn ra).
- `showCancelForm(meetingId: int): void`  
Hiển thị form nhập lý do hủy khi người dùng chọn nút hủy.
- `submitCancelForm(meetingId: int, reason: String): void`  
Gửi lý do hủy và thực hiện cập nhật trạng thái CANCELLED.

#### AppointmentBookingUI



- showBookingForm(): void  
Hiển thị form đặt lịch hẹn.
- submitBookingForm(studentId: int, tutorId: int, date: String, startTime: String, topic: String): bool  
Gửi yêu cầu đặt lịch hẹn đến SchedulingAPI. Trả về true nếu thành công.
- displayAppointmentDetails(appointment: Appointment): void  
Hiển thị thông tin chi tiết của một lịch hẹn.

### **AppointmentManagementUI**

- displayPendingAppointments(tutorId: int): void  
Hiển thị danh sách các lịch hẹn đang chờ duyệt.
- approveAppointment(appointmentId: int): bool  
Duyệt một lịch hẹn, cập nhật trạng thái APPROVED và thêm vào Meeting list.
- rejectAppointment(appointmentId: int): bool  
Từ chối một lịch hẹn, cập nhật trạng thái REJECTED.
- showAppointmentDetails(appointmentId: int): void  
Hiển thị chi tiết thông tin lịch hẹn.

### **ConsultationCreateUI**

- showcreateForm(): void  
Hiển thị form tạo buổi tư vấn.
- submitConsultationForm(data: Map<String, Any>): bool  
Gửi dữ liệu tạo buổi tư vấn đến SchedulingAPI. Trả về true nếu thành công.
- selectRoom(): void  
Lấy danh sách phòng trống từ RoomAPI và chọn phòng (nếu offline).
- displayConsultationDetails(consultation: Consultation): void  
Hiển thị thông tin chi tiết của buổi tư vấn.

### **ConsultationRegistrationUI**



- **showRegistrationForm(): void**  
Hiển thị form đăng ký tham gia buổi tư vấn.
- **submitRegistrationForm(consultationId: int, studentId: int): bool**  
Gửi yêu cầu đăng ký buổi tư vấn. Trả về true nếu thành công.
- **displayRegisteredConsultationDetails(consultation: Consultation): void**  
Hiển thị thông tin chi tiết buổi tư vấn đã đăng ký.

### RoomUI

- **displayAvailableRooms(date: String, startTime: String, endTime: String): void**  
Hiển thị danh sách phòng trống theo thời gian.
- **showRoomDetails(roomId): void**  
Hiển thị chi tiết thông tin phòng cho người dùng.

### LoginUI

- **displayLoginForm(): void**  
Hiển thị form đăng nhập gồm username và password.
- **getUsernameInput(): String**  
Lấy giá trị username người dùng nhập.
- **getPasswordInput(): String**  
Lấy giá trị password người dùng nhập.
- **showLoginSuccessMessage(): void**  
Hiển thị thông báo đăng nhập thành công.
- **showLoginErrorMessage(msg: String): void**  
Hiển thị thông báo lỗi với nội dung msg.
- **showAccessDeniedMessage(): void**  
Hiển thị thông báo quyền truy cập bị từ chối.
- **onLoginButtonClick(callback: (username, password) → void): void**  
Gán callback khi người dùng nhấn nút đăng nhập.

### ForgotPasswordUI



- `displayForgotPasswordForm(): void` – Form nhập ID/email để reset.
- `displayResetPasswordForm(token: String): void` – Form nhập mật khẩu mới.
- `showPasswordResetSuccessMessage(): void` – Thông báo reset thành công.
- `showErrorMessage(msg: String): void` – Thông báo lỗi.
- `onRequestResetClick(callback): void` – Xử lý sự kiện gửi yêu cầu reset.
- `onResetPasswordClick(callback): void` – Xử lý sự kiện lưu mật khẩu mới.

### UserProfileUI

- `onOpenProfile(): void` – Mở trang cá nhân, gọi API lấy dữ liệu.
- `displayProfile(user: UserEntity): void` – Hiển thị thông tin chi tiết người dùng.
- `showErrorMessage(msg: String): void` – Hiển thị thông báo lỗi.

### LogoutUI

- **`displayLogoutConfirmation(): void`**  
Hiển thị hộp thoại yêu cầu xác nhận đăng xuất.
- **`showLogoutSuccessMessage(): void`**  
Hiển thị thông báo đăng xuất thành công.
- **`showErrorMessage(msg: String): void`**  
Hiển thị thông báo lỗi với nội dung `msg`.
- **`onConfirmLogout(callback: () → void): void`**  
Gán callback khi người dùng xác nhận logout.
- **`onCancelLogout(callback: () → void): void`**  
Gán callback khi người dùng hủy thao tác logout.

### MaterialUI

- **`displayMaterialList(materials: List<MaterialEntity>): void`**  
Hiển thị danh sách các tài liệu.



- **displayMaterialContent(materialId: String, content: String): void**  
Hiển thị nội dung chi tiết của tài liệu.
- **showErrorMessage(msg: String): void**  
Hiển thị thông báo lỗi.
- **onSelectMaterial(callback: (materialId: String) → void): void**  
Gán callback khi người dùng chọn tài liệu.
- **onAdvancedSearch(callback: (criteria: SearchCriteria) → void): void**  
Gán callback khi người dùng thực hiện tìm kiếm nâng cao.

#### TutorSlotCalendarView

- **renderCalendar(tutorID: String, slots: List): void**  
Hiển thị lịch của gia sư (tutorID) với các slot (slots) đã cung cấp.
- **showAddSlotModal(): void**  
Hiển thị cửa sổ (modal) để cho phép gia sư thêm một slot thời gian mới.
- **showEditSlotModal(slot: AvailableSlot): void**  
Hiển thị cửa sổ (modal) để chỉnh sửa thông tin một slot đã có.
- **enableDragAndDrop(): void**  
Kích hoạt chức năng kéo-thả trên lịch để di chuyển hoặc thay đổi thời lượng slot.
- **highlightAvailableSlots(): void**  
Làm nổi bật các slot thời gian còn trống trên lịch.
- **onSlotDelete(callback: (slotID: String) → void): void**  
Gán callback khi gia sư thực hiện hành động xóa một slot.

#### StudentTutorSelectionView

- **renderTutorCard(tutorID: String, name: String, expertise: String): void**  
Hiển thị thông tin của một gia sư dưới dạng thẻ (card), bao gồm ID, tên, và chuyên môn.
- **onTutorClick(callback: (tutorID: String) → void): void**  
Gán callback khi sinh viên nhấp vào thẻ của một gia sư (để xem chi tiết hoặc lịch).



- **renderSlotList(slots: List): void**

Hiển thị danh sách các slot thời gian (thường là của gia sư vừa được chọn).

- **disableBookedSlots(): void**

Vô hiệu hóa (làm mờ hoặc không cho phép nhập) các slot đã được đặt.

- **onSlotSelect(callback: (slotID: String) → void): void**

Gán callback khi sinh viên chọn một slot thời gian (để đặt lịch).

### MaterialUploadUI

- **chooseFile(): void**

Hiển thị cửa sổ/dialog để người dùng chọn tệp (file) từ máy tính.

- **fillForm(title: String, type: String, classId: String): void**

Cho phép người dùng điền thông tin mô tả cho tài liệu (tiêu đề, loại, ID lớp).

- **submitUpload(tutorId: int): void**

Gửi thông tin và tệp đã chọn lên hệ thống (qua API) để tải lên.

- **submitReviewRequest(materialId: int): void**

Gửi yêu cầu xem xét (review) cho một tài liệu đã tải lên.

- **listMyUploads(tutorId: String): void**

Hiển thị danh sách các tài liệu mà gia sư này đã tải lên.

- **showStatus(msg: String): void**

Hiển thị thông báo trạng thái (ví dụ: "Tải lên thành công", "Lỗi") cho người dùng.

### ListMeetingUI

- **fetchOfficialMeetings(studentId: int): void**

Lấy danh sách các buổi học của sinh viên từ.

- **formatMeetingTime(start: String, end: String): String**

Xử lý chuỗi thời gian thô thành định dạng hiển thị dễ đọc (Giờ - Ngày)

### FeedbackUI

- **fetchMeetingInfo(id: int): void**

Lấy thông tin chi tiết (chủ đề, thời gian) của buổi hẹn hiện tại dựa trên ID.



- fetchFeedbackHistory(meetingId: int): void  
Tải lịch sử các phản hồi trước đó của sinh viên cho buổi hẹn này.
- handleRatingChange(rating: int): void  
Cập nhật số sao đánh giá của sinh viên
- handleCommentChange(text: String): void  
Cập nhật nội dung nhận xét khi sinh viên nhập liệu vào ô văn bản.
- handleSubmitFeedback(): void  
Kiểm tra dữ liệu (bắt buộc rating) và gửi đánh giá lên server qua API POST.
- navigateBack(): void  
Điều hướng quay trở lại trang danh sách buổi hẹn trước đó.

### **CourseListUI**

- renderCourseList(courses)  
Hiển thị danh sách dạng card hoặc grid.
- showEnrollButton(courseID)  
Hiển thị nút đăng ký khi học viên chưa đăng ký.

### **CourseDetailUI**

- renderCourseDetail(course)  
Hiển thị tiêu đề, mô tả, tutor và số lượng sinh viên.
- displayTutorInfo(tutorID)  
Hiển thị thông tin chi tiết về giảng viên.

### **RequestFormUI**

- showForm(tutors, courses)  
Tải danh sách tutor và khóa học lên giao diện.
- onSubmit(requestDTO)  
Gửi yêu cầu khi người dùng nhấn nút Gửi.

### **RequestListUI**



- renderMyRequests(requests)

Liệt kê các yêu cầu đã gửi của sinh viên.

- onCancelRequest(requestID)

Cho phép hủy yêu cầu nếu trạng thái đang là PENDING.

### TutorRequestListView

- renderPendingRequests(requests)

Hiển thị danh sách yêu cầu cần duyệt.

- onAccept(requestID)

Chấp nhận yêu cầu và tạo buổi học mới.

- onReject(requestID)

Từ chối yêu cầu và gửi email thông báo lý do.

### NotificationUI

- load(): void

Mở màn hình thông báo và nạp dữ liệu ban đầu (danh sách thông báo gần đây, số lượng chưa đọc).

- render(): void

Cập nhật và hiển thị giao diện danh sách thông báo theo dữ liệu hiện có.

- handleManualSend(): void

Xử lý thao tác gửi thông báo thủ công.

- handleAutoSend(): void

Xử lý thao tác gửi thông báo tự động.

- markAsRead(userId: int, notificationId: int): void

Dánh dấu một thông báo của người dùng là “đã đọc”.

### UserProfileUI

- onOpenProfile(): void

Xử lý sự kiện mở trang cá nhân và gọi Controller lấy dữ liệu.



- **displayProfile(user: UserEntity): void**

Hiển thị thông tin chi tiết người dùng lên giao diện.

- **showErrorMessage(msg: String): void**

Hiển thị thông báo lỗi khi không tải được dữ liệu hồ sơ.

### **ReportUI**

- **handleManualSend(report: Report, requestedBy: Department): void**

Gửi báo cáo thủ công đến phòng ban yêu cầu.

- **handleAutoSend(report: Report, requestedBy: Department, periods: String): void**

Gửi báo cáo tự động theo chu kỳ đến phòng ban yêu cầu.

- **displayReports(reports: List<Report>): void**

Hiển thị danh sách các báo cáo đã tạo.

- **displayReport(report: Report): void**

Hiển thị chi tiết một báo cáo cụ thể.

- **downloadReport(reportId: int, format: ReportFormat): void**

- Tải về báo cáo ở định dạng mong muốn.

### **TutorViewFreeSlotsUI**

- **fetchDailySlots(date: String): void** – Lấy danh sách khung giờ rảnh chi tiết.

- **handleCalendarSelect(date: String): void** – Cập nhật state khi chọn ngày trên lịch.

- **MapsToSetupView(): void** – Chuyển sang màn hình "Thiết lập lịch rảnh".

### **TutorUpdateFreeSlotUI**

- **handleAddSlot(): void** – Thêm dòng nhập liệu mới.

- **handleRemoveSlot(index: int): void** – Xóa dòng nhập liệu tại vị trí index.

- **handleChangeTime(index: int, field: String, value: String): void** – Cập nhật giờ bắt đầu/kết thúc.



- handleSave(): void – Thu thập dữ liệu và gọi API POST để lưu lịch.
- MapsToCalendarView(): void – Quay lại màn hình xem lịch.

#### TutorRegistrationUI

- **render(): void**

Hiển thị giao diện đăng ký tutor cho người dùng.

- **onSelectTutor(tutor: any): void**

Xử lý sự kiện khi người dùng chọn một tutor từ danh sách.

- **onConfirm(): void**

Xử lý sự kiện khi người dùng xác nhận đăng ký, gửi request tới controller.

#### 3.5.2 Controller Method

##### TutorSchedulingAPI

- pending(tutorId: Long): List<Appointment>

Trả về danh sách các Appointment đang ở trạng thái PENDING của tutor.

- approve(id: Long, req: ApproveRequest): String

Tutor phê duyệt một Appointment theo ID. Trả về thông báo thành công hoặc lỗi.

- getAppointmentDetail(id: Long): Appointment

Xem chi tiết một Appointment theo ID.

- reject(id: Long, req: RejectRequest): String

Tutor từ chối Appointment kèm lý do. Lý do phải không rỗng.

- official(tutorId: Long): List<Meeting>

Trả về danh sách các Meeting đã được xác nhận (APPROVED) và còn hiệu lực của tutor.

- cancel(id: Long, req: CancelRequest): String

Tutor hủy một Meeting kèm lý do hủy.

- returnSlot(tutorId: Long, meetingId: Long): String

Tutor chọn trả hoặc không trả lại slot đã bị hủy vào lịch rảnh.



- detail(id: Long): Meeting  
Xem chi tiết một Meeting theo ID.

### **StudentSchedulingAPI**

- book(req: AppointmentRequest): String  
Sinh viên đặt lịch hẹn mới với tutor. Trả về thông báo thành công hoặc thất bại.
- getHistory(studentId: Long): List<Appointment>  
Trả về toàn bộ lịch sử Appointment của sinh viên.
- getCancelableMeetings(studentId: Long): List<Meeting>  
Trả về các Meeting sinh viên được phép hủy.
- cancel(id: Long, req: CancelRequest): String  
Sinh viên hủy một Meeting kèm lý do.
- getOfficial(studentId: Long): List<Meeting>  
Trả về danh sách Meeting đã được xác nhận và còn hiệu lực của sinh viên.
- viewSlots(tutorId: Long): List<FreeSlotResponse>  
Xem danh sách các slot rảnh của tutor để đặt lịch.
- detail(id: Long): Meeting  
Xem chi tiết một Meeting theo ID.

### **FreeSlotAPI**

- getDaily(date: LocalDate): ResponseEntity<FreeSlotResponse> – Lấy lịch rảnh chi tiết.
- getMonthly(month: int, year: int): ResponseEntity<List<FreeSlotResponse>> – Lấy lịch tóm tắt cả tháng.
- overwriteSchedule(request: FreeSlotRequest): ResponseEntity<?> – Ghi đè lịch rảnh.

### **RoomAPI**

- getAvailableRooms(date: Date, startTime: Time, endTime: Time): ResponseEntity<List<Room>>  
Trả về danh sách phòng trống theo yêu cầu từ UI.
- reserveRoom(Long roomId, ReserveRequest request): ResponseEntity<RoomBooking>  
Cập nhật trạng thái phòng.



- getRoomDetails(roomId: int): ResponseEntity<Room>  
Lấy thông tin chi tiết của một phòng.
- releaseRoom(roomId, meetingId): ResponseEntity<Void>  
Hủy đặt phòng theo mã cuộc họp.

### LearningAPI

- addCourse(courseData: Course): Course  
Thêm khóa học mới.
- getAllCourses(): List<CourseDTO>  
Trả về danh sách khóa học ở dạng DTO.
- getCourseById(courseId: int): CourseDetail  
Lấy thông tin chi tiết khóa học.
- registerCourse(studentId: int, courseId: int): Enrollment  
Cho phép sinh viên đăng ký khóa học.
- getEnrollmentsByStudent(studentId: int): List<Enrollment>  
Trả về danh sách đăng ký của sinh viên.
- submitRequest(requestData: Request): Request  
Xử lý gửi yêu cầu tư vấn.
- getMyRequests(userId: int): List<Request>  
Trả về danh sách yêu cầu của người dùng hiện tại.
- getRequestsByTutor(tutorId: int): List<Request>  
Trả về danh sách yêu cầu dành cho tutor.
- updateRequestStatus(requestId: int, newStatus: String): bool  
Cập nhật trạng thái yêu cầu.

### FeedbackAPI

- submitFeedback(meetingId: Long, rating: int, comment: String): Feedback  
Nhận phản hồi đánh giá buổi học từ sinh viên



- **getFeedbacks(meetingId: Long): List<Feedback>**  
Trả về danh sách đánh giá của một buổi học theo Id

### **LoginAPI**

- **login(username: String, password: String): void**  
Gọi SSOService để xác thực, tạo session và thông báo kết quả lên view.

### **ForgotPasswordAPI**

- **requestPasswordReset(bkId: String): void** – Kiểm tra user, tạo token, gửi email.
- **resetPassword(token: String, newPassword: String): void** – Xác thực token và cập nhật mật khẩu.

### **UserProfileManageAPI**

- **viewProfile(): void** – Lấy ID người dùng từ session và gọi Service.

### **LogoutAPI**

- **logout(): void**  
Vô hiệu hóa session và thông báo kết quả lên view.

### **MaterialAPI**

- **viewMaterialList(): void**  
Lấy danh sách tài liệu từ MaterialService và hiển thị.
- **viewMaterial(materialId: String): void**  
Lấy nội dung chi tiết tài liệu và hiển thị.
- **searchMaterials(criteria: SearchCriteria): void**  
Thực hiện tìm kiếm tài liệu theo tiêu chí và hiển thị kết quả.

### **SlotAPI**

- **addSlot(tutorID: String, slotDTO: SlotDTO): ResponseEntity**  
Xử lý yêu cầu HTTP (ví dụ: POST) để thêm một slot mới cho gia sư. Gọi đến AvailableSlotService.



- **updateSlot(tutorID: String, slotID: String, slotDTO: SlotDTO): ResponseEntity**

Xử lý yêu cầu HTTP (ví dụ: PUT) để cập nhật thông tin một slot đã có.

- **deleteSlot(tutorID: String, slotID: String): ResponseEntity**

Xử lý yêu cầu HTTP (ví dụ: DELETE) để xóa một slot.

- **getTutorSlots(tutorID: String): ResponseEntity**

Xử lý yêu cầu HTTP (ví dụ: GET) để lấy về danh sách các slot của một gia sư.

### **MaterialUploadAPI**

- **createDraft(tutorId: int, title: String, type: String, classId: String): Material**

Xử lý yêu cầu HTTP tạo một bản nháp tài liệu, gọi MaterialUploadService.

- **attachFile(materialId: int, fileName: String, file: File): Material**

Xử lý yêu cầu HTTP đính kèm tệp vào một bản nháp tài liệu đã có.

- **submitForReview(materialId: int): void**

Xử lý yêu cầu HTTP gửi tài liệu để xem xét.

- **listMyUploads(tutorId: int): List<Material>**

Xử lý yêu cầu HTTP lấy danh sách các tài liệu đã tải lên của gia sư.

### **NotificationAPI**

- **notifyScheduleRequested(studentId: int, tutorId: int, status: String): void**

Gửi thông báo về yêu cầu đặt lịch của sinh viên (tạo mới/được duyệt/bị từ chối) đến người nhận phù hợp.

- **notifyMeeting(meeting: Meeting, studentId: int, tutorId: int): void**

Thông báo liên quan tới buổi tư vấn: tạo mới, thay đổi thời gian/phòng hoặc hủy.

- **notifyMaterial(material: Material, tutorId: int, studentId: int, status: String): void**

Thông báo trạng thái tài liệu (đã duyệt/bị từ chối/được công bố) cho gia sư hoặc sinh viên tùy trường hợp.

- **notifyCourseUpdated(course: CourseEntity, tutorId: int): void**

Báo cho gia sư (và/hoặc lớp học liên quan) khi nội dung/kế hoạch khóa học được cập nhật.



- **getMyNotifications(userId: String): List<Notification>**

Lấy danh sách thông báo của người dùng để hiển thị trong hộp thư in-app.

- **markAsRead(userId: String, notificationId: String): void**

Dánh dấu một thông báo là đã đọc và cập nhật lại hiển thị/badge.

### ReportAPI

- **sendManual(file: File, requestedBy: Department): void**

Gửi báo cáo thủ công: nhận tệp đã chọn và gửi tới phòng ban được chỉ định (thường qua email).

- **sendAuto(file: File, requestedBy: Department, periods: String): void**

Gửi báo cáo tự động theo kỳ (ví dụ tháng/quý) đến phòng ban mặc định, dùng tham số kỳ để điền bộ lọc.

### UserProfileManageAPI

- **viewProfile(): void**

Lấy ID người dùng từ Session hiện tại, gọi Service lấy thông tin và trả về View.

### ProfileService

- **getUserProfile(userId: Long): UserEntity** – Lấy dữ liệu người dùng từ Repository.

### SessionManager

- **createSession(user: UserEntity): String** – Tạo session mới.

- **getSession(sessionId: String): SessionEntity** – Lấy thông tin session.

- **invalidateSession(sessionId: String): void** – Hủy session.

- **isValid(sessionId: String): Boolean** – Kiểm tra tính hợp lệ session.

### TutorRegistrationController

- **registerTutor(request: RegisterRequest): ResponseEntity**

Nhận request đăng ký tutor từ frontend và xử lý.

- **cancelRegistration(request: CancelRequest): ResponseEntity**

Nhận request hủy đăng ký và xử lý.



- **suggestTutors(subject: String): ResponseEntity**

Trả về danh sách tutor được gợi ý cho môn học cụ thể.

### 3.5.3 Service Method

#### StudentSchedulingService

- **bookAppointment(studentId: Long, tutorId: Long, date: LocalDateTime, startTime: LocalDateTime, endTime: LocalDateTime, topic: String): bool**

Tạo một lịch hẹn mới giữa sinh viên và tutor. Logic bao gồm: kiểm tra tutor có trống hay không trong khoảng thời gian yêu cầu, kiểm tra slot rảnh từ FreeSlotService, lưu Appointment vào Repository, và cất slot rảnh tương ứng. Trả về true nếu đặt thành công, false nếu không còn slot hoặc xảy ra lỗi.

- **viewOfficialMeetings(studentId: Long): List<Meeting>**

Lấy danh sách các Meeting chính thức của sinh viên, bao gồm Appointment đã duyệt (APPROVED) và các loại Meeting khác. Dữ liệu lấy từ MeetingRepository.

- **viewAppointmentHistory(studentId: Long): List<Appointment>**

Trả về lịch sử tất cả các Appointment của sinh viên, bao gồm PENDING, APPROVED, REJECTED, CANCELLED. Phục vụ mục xem lịch sử đặt lịch.

- **cancelMeeting(meetingId: Long, reason: String): bool**

Sinh viên yêu cầu hủy một Meeting. Kiểm tra Meeting tồn tại, chưa bị hủy, thuộc về sinh viên. Cập nhật trạng thái CANCELLED, lưu lý do hủy, và nếu Appointment đã được duyệt thì trả lại slot vào FreeSlotService. Trả về true nếu hủy thành công.

- **viewTutorAvailableSlots(tutorId: Long): List<FreeSlotResponse>**

Lấy danh sách các slot rảnh của tutor (dạng date + list of time ranges). Phục vụ UI để hiển thị lịch rảnh.

- **findCancellableMeetings(studentId: Long): List<Meeting>**

Trả về danh sách Meeting mà sinh viên có quyền hủy: là các Meeting APPROVED với startTime nằm trong tương lai.

- **viewMeetingDetails(meetingId: Long): Meeting**

Lấy chi tiết đầy đủ của một Meeting (bao gồm thông tin tutor, thời gian, online link, trạng thái...). Trả về null nếu MeetingId không tồn tại.



## TutorSchedulingService

- `viewPendingAppointments(tutorId: Long): List<Appointment>`

Lấy danh sách Appointment đang ở trạng thái PENDING của tutor. Dùng cho UI để hiển thị các yêu cầu chờ duyệt.

- `approveAppointment(appointmentId: Long, tutorId: Long): bool`

Tutor duyệt Appointment. Logic: kiểm tra quyền sở hữu, kiểm tra Appointment chưa được duyệt hoặc từ chối, cập nhật trạng thái thành APPROVED, tạo onlineLink, và lưu lại vào Repository. Trả về true nếu duyệt thành công.

- `rejectAppointment(appointmentId: Long, tutorId: Long, reason: String): bool`

Tutor từ chối Appointment. Kiểm tra quyền hợp lệ, cập nhật trạng thái thành REJECTED, lưu lý do từ chối. Trả về true nếu thao tác thành công.

- `viewOfficialMeetings(tutorId: Long): List<Meeting>`

Trả về Meetings chính thức của tutor, bao gồm các Appointment đã duyệt và các Meeting khác có trong Repository.

- `cancelMeeting(tutorId: Long, meetingId: Long, reason: String): bool`

Tutor hủy buổi Meeting. Kiểm tra Meeting thuộc tutor, chưa bị hủy, cập nhật trạng thái CANCELLED, lưu lý do. Trả về true nếu hủy thành công.

- `tutorReturnCancelledSlot(tutorId: Long, meetingId: Long): bool`

Sau khi hủy Meeting, tutor có tùy chọn trả slot vào lịch rảnh. Kiểm tra Meeting đúng tutor, đúng trạng thái CANCELLED, rồi gọi FreeSlotService để trả lại slot. Trả về true nếu slot được trả, false nếu tutor chọn không trả.

- `findCancellableMeetings(tutorId: Long): List<Meeting>`

Trả về các Meeting mà tutor có quyền hủy (APPROVED và xảy ra trong tương lai).

- `viewMeetingDetails(meetingId: Long): Meeting`

Lấy thông tin chi tiết của một Meeting theo ID. Trả về null nếu không tồn tại.

- `createOnlineLink(appointment: Appointment): String`

Tạo đường link học online cho Appointment (ví dụ: tạo link Google Meet). Được gọi khi tutor approve lịch. Trả về chuỗi URL.



- **viewAppointmentDetails(appointmentId: Long): Appointment**

Lấy thông tin chi tiết của một cuộc hẹn (Appointment). Nếu ID không tồn tại hoặc Meeting không phải là Appointment thì trả về null.

### FreeSlotService

- **getDailySchedule(tutorId: Long, date: LocalDate): FreeSlotResponse**

Lấy danh sách Entity từ Repository, chuyển đổi sang DTO để trả về Controller.

- **getMonthlySchedule(tutorId: Long, month: int, year: int): List<FreeSlotResponse>**

Lấy danh sách slot trong 1 tháng, gom nhóm theo ngày để trả về cấu trúc dữ liệu cho lịch tháng.

- **overwriteDailySchedule(tutorId: Long, request: FreeSlotRequest): List<String>**

Xóa lịch cũ, tạo Entity mới, lưu lịch mới, trả về cảnh báo validate nếu có.

- **reserveSlot(tutorId: Long, date: LocalDate, start: LocalTime, end: LocalTime): void**

Chuyển một slot từ trạng thái "Rảnh" sang "Đã đặt" khi có học viên đăng ký.

- **releaseSlot(tutorId: Long, date: LocalDate, start: LocalTime, end: LocalTime): void**

Hủy trạng thái "Đã đặt", trả slot về lại trạng thái "Rảnh".

- **getRawAvailableSlots(tutorId: Long, date: LocalDate): List<TutorSlot>**

Lấy trực tiếp danh sách Entity để phục vụ các xử lý nội bộ khác.

### RoomService

- **getAvailableRooms(date: Date, startTime: Time, endTime: Time): List<Room>**

Trả về danh sách phòng trống theo yêu cầu.

- **isRoomAvailable(roomId: int, date: Date, startTime: Time, endTime: Time): boolean**

Kiểm tra phòng có trống không.

- **reserveRoom(roomId: int, date: Date, startTime: Time, endTime: Time, meetingId: int): boolean**

Đặt phòng cho một cuộc họp. Trả về true nếu đặt thành công.



- releaseRoom(roomId: int, meetingId: int): boolean

Hủy đặt phòng theo mã cuộc họp.

- getRoomDetails(roomId: int): Room

Lấy thông tin chi tiết của phòng.

### CourseService

- addCourse(title: String, description: String, instructorID: int): bool

Dùng để thêm khóa học mới, có kiểm tra trùng tiêu đề và trả về **true** nếu thêm thành công.

- getAllCourses(): List<CourseEntity>

Trả về danh sách tất cả các CourseEntity trong cơ sở dữ liệu.

- getCourseById(courseID: int): CourseEntity

Dùng để tìm khóa học theo ID và trả về **null** nếu không tồn tại.

### EnrollmentService

- register(studentID: int, courseID: int): bool

Cho phép sinh viên đăng ký khóa học nếu chưa đăng ký trước đó.

- getEnrollmentsByStudent(studentID: int): List<Course>

Trả về danh sách các khóa học mà sinh viên đã đăng ký.

### RequestService

- createRequest(studentID: int, tutorID: int, courseID: int, topic: String): int

Tạo yêu cầu mới, gửi email đến tutor và trả về **requestID**.

- getRequestsByStudent(studentID: int): List<Request>

Trả về danh sách yêu cầu của sinh viên.

- getRequestsByTutor(tutorID: int): List<Request>

Trả về danh sách yêu cầu đang chờ tutor duyệt.

- updateStatus(requestID: int, status: String): bool

Cập nhật trạng thái yêu cầu từ PENDING sang APPROVED hoặc REJECTED.

### FeedbackService



- **addFeedback(meetingId: Long, rating: int, comment: String): Feedback**  
Kiểm tra và lưu phản hồi đánh giá buổi học của sinh viên
- **getFeedbackByMeeting(meetingId: Long): List<Feedback>**  
Lấy ra danh sách phản hồi đánh giá một buổi học theo ID buổi học

### SSOService

- **authenticate(username: String, password: String): UserEntity**  
Kiểm tra username/password, trả về UserEntity nếu thành công.
- **sendResetEmail(bkId: String): void**  
Gửi email reset password.
- **resetPassword(token: String, newPassword: String): void**  
Reset password dựa trên token.

### DatacoreService

- **getUserInfo(username: String): UserEntity**  
Lấy thông tin user từ Datacore.

### MailService

- **sendEmail(to: String, content: String): void**  
Gửi email đơn giản.
- **sendPasswordResetEmail(to: String, resetLink: String): void**  
Gửi link reset password.
- **sendNotificationEmail(to: String, subject: String, content: String): void**  
Gửi email thông báo.

### ProfileService

- **getUserProfile(userId: Long): UserEntity** – Lấy dữ liệu người dùng từ Repository.

### SessionManager

- **createSession(user: UserEntity): String**  
Tạo session mới và trả về sessionId.



- **getSession(sessionId: String): SessionEntity**

Lấy thông tin session.

- **invalidateSession(sessionId: String): void**

Vô hiệu hóa session.

- **isValid(sessionId: String): Boolean**

Kiểm tra session còn hợp lệ.

#### MaterialService

- **getMaterials(): List<MaterialEntity>**

Lấy danh sách tài liệu.

- **getMaterialContent(materialId: String): String**

Lấy nội dung chi tiết tài liệu.

- **searchMaterials(criteria: SearchCriteria): List<MaterialEntity>**

Tìm kiếm tài liệu theo tiêu chí.

#### AvailableSlotService

- **addAvailableSlot(tutorID: String, start: LocalDateTime, end: LocalDateTime): boolean**

Thêm một slot trống mới cho gia sư. Kiểm tra logic nghiệp vụ (ví dụ: không trùng lặp).

Trả về true nếu thành công.

- **updateAvailableSlot(tutorID: String, slotID: String, start: LocalDateTime, end: LocalDateTime): boolean**

Cập nhật thời gian bắt đầu/kết thúc của một slot. Trả về true nếu thành công.

- **removeAvailableSlot(tutorID: String, slotID: String): boolean**

Xóa một slot khỏi danh sách (ví dụ: gia sư hủy slot). Trả về true nếu thành công.

- **getAvailableSlotsByTutor(tutorID: String): List<AvailableSlot>**

Lấy tất cả các slot (cả trống và đã đặt) của một gia sư.

- **getAvailableSlotsByTutorAndRange(tutorID: String, start: LocalDateTime, end: LocalDateTime): List<AvailableSlot>**

Lấy các slot của gia sư trong một khoảng thời gian cụ thể.



- **findAvailable(tutorID: String, start: LocalDateTime, end: LocalDateTime): boolean**  
Kiểm tra xem gia sư có rảnh (available) trong khoảng thời gian được yêu cầu hay không.
- **markSlotAsBooked(tutorID: String, slotID: String): boolean**  
Đánh dấu một slot là đã được đặt (trạng thái: BOOKED). Trả về true nếu thành công.
- **markSlotAsAvailable(tutorID: String, slotID: String): boolean**  
Đánh dấu một slot là còn trống (trạng thái: AVAILABLE). Trả về true nếu thành công.

### MaterialUploadService

- **createDraft(tutorId: String, title: String, type: String, classId: String): Material**  
Logic nghiệp vụ tạo một bản nháp (draft) tài liệu trong cơ sở dữ liệu.
- **attachFile(materialId: String, fileName: String, bytes: byte[]): Material**  
Logic nghiệp vụ xử lý tệp (bytes) và liên kết nó với bản nháp tài liệu.
- **submitForReview(materialId: String): void**  
Logic nghiệp vụ cập nhật trạng thái tài liệu thành "Đang chờ xem xét" (Pending Review).
- **listMyUploads(tutorId: String): List<Material>**  
Truy vấn cơ sở dữ liệu (qua repository) để lấy danh sách tài liệu của gia sư.

### NotificationService

- **sendToUser(o: Object, userId: int, type: String, title: String, content: String): void**  
Tạo một thông báo và gửi trực tiếp cho *một* người dùng xác định bởi userId. Tham số type/title/content mô tả loại, tiêu đề và nội dung thông báo; o là dữ liệu ngữ cảnh đi kèm (nếu cần).
- **sendToUsers(o: Object, userIds: List<int>, type: String, title: String, content: String): void**  
Gửi cùng một thông báo đến *nhiều* người dùng trong danh sách userIds. Dùng cho các tình huống broadcast (ví dụ gửi cho cả lớp/khoa).

### ReportService

- **sendManual(report: Report, requestedBy: Department): void**  
Gửi báo cáo theo yêu cầu thủ công: nhận đối tượng report (tệp/siêu dữ liệu báo cáo đã chuẩn bị) và gửi đến phòng ban được chỉ định trong requestedBy.



- **sendAuto(report: Report, requestedBy: Department, periods: String): void**  
Gửi báo cáo tự động theo kỳ (ví dụ tháng/quý) đến phòng ban. Chuỗi periods mô tả khoảng thời gian cần áp dụng cho báo cáo.
- **generateReport(data: Map<key,value>, report: Report): File**  
Sinh tệp báo cáo từ dữ liệu đầu vào data theo mẫu/dịnh nghĩa trong report, trả về tệp để dùng cho việc gửi hoặc tải xuống.

#### TutorRegistrationService

- **createRequest(studentId: String, subject: String, tutorId: String): TutorRegistrationEntity**  
Tạo yêu cầu đăng ký tutor mới với trạng thái PENDING.
- **cancelRequest(registrationId: Long, studentId: String): boolean**  
Hủy yêu cầu đăng ký, kiểm tra quyền sở hữu trước khi hủy.
- **suggestTutor(subject: String): List<TutorSuggestion>**  
Lấy danh sách tutor phù hợp cho môn học (dựa trên rating, availability).
- **findPendingOlderThan(cutoff: LocalDateTime): List<TutorRegistrationEntity>**  
Tìm các yêu cầu PENDING cũ hơn một thời điểm để xử lý tự động.
- **approveRegistration(reg: TutorRegistrationEntity): void**  
Phê duyệt yêu cầu đăng ký (chuyển PENDING → APPROVED).

#### 3.5.4 Repository Method (Implementation)

##### RoomRepository

- **listAvailableRooms(date: Date, startTime: Time, endTime: Time): List<Room>**  
Trả về danh sách phòng trống trong khoảng thời gian nhất định.
- **updateRoomStatus(roomId: int, status: String): bool**  
Cập nhật trạng thái phòng (AVAILABLE, OCCUPIED, MAINTENANCE). Trả về true nếu thành công.
- **getRoomInfo(roomId: int): String**  
Lấy thông tin chi tiết của phòng theo roomId.



### RoomBookingRepository

- `findBookings(date: Date): List<RoomBooking>`  
Trả về tất cả các lịch đặt phòng theo ngày.
- `save(booking: RoomBooking): void`  
Lưu một lịch đặt phòng mới.
- `findConflicts(roomId: int, date: Date, startTime: Time, endTime: Time): boolean`  
Kiểm tra phòng có xung đột lịch hay không.

### MeetingRepository

- `findById(meetingId: Long): Meeting`  
Tìm và trả về Meeting theo `meetingId`. Trả về `null` nếu không tồn tại.
- `save(meeting: Meeting): void`  
Lưu một Meeting mới vào danh sách nội bộ.
- `update(meeting: Meeting): void`  
Cập nhật Meeting theo `meetingId`. Ném ngoại lệ nếu không tìm thấy.
- `findPendingAppointmentsByTutor(tutorId: Long): List<Appointment>`  
Trả về danh sách các Appointment ở trạng thái `PENDING` của tutor.
- `findApprovedAppointmentsByTutor(tutorId: Long): List<Appointment>`  
Trả về danh sách Appointment ở trạng thái `APPROVED` của tutor.
- `findAllAppointmentsByStudent(studentId: Long): List<Appointment>`  
Lấy toàn bộ Appointment của một student (mọi trạng thái).
- `findApprovedAppointmentsByStudent(studentId: Long): List<Appointment>`  
Trả về Appointment đã `APPROVED` của student.
- `findApprovedMeetingByStudent(studentId: Long): List<Meeting>`  
Trả về Meeting đã `APPROVED` của student.
- `findApprovedMeetingByTutor(tutorId: Long): List<Meeting>`  
Trả về Meeting đã `APPROVED` của tutor.



- **findOfficialMeetingsByStudent(studentId: Long): List<Meeting>**  
Trả về Meeting APPROVED và chưa bị hủy (`!isCancelled()`) của student.
- **findOfficialMeetingsByTutor(tutorId: Long): List<Meeting>**  
Trả về Meeting APPROVED và chưa bị hủy của tutor.

### UserRepository

- **findById(id: Long): UserEntity** – Lấy thông tin người dùng theo ID.
- **createSession(user: UserEntity): String** – Tạo session mới.
- **save(sessionId: String): void** – Lưu thay đổi session.
- **update(sessionId: String): Boolean** – Cập nhật trạng thái session.

### FreeSlotRepository

- **findAvailableByTutorIdAndDate(tutorId: Long, date: LocalDate): List<TutorSlot>**  
Lọc danh sách Mock để lấy các slot rảnh của gia sư trong ngày cụ thể.
- **findAvailableByTutorIdAndDateBetween(tutorId: Long, month: int, year: int): List<TutorSlot>**  
Tìm kiếm slot rảnh trong khoảng thời gian của tháng và năm.
- **saveAvailable(slots: List<TutorSlot>): void**  
Thêm danh sách các đối tượng TutorSlot mới vào kho lưu trữ.
- **findBookedByTutorIdAndDate(tutorId: Long, date: LocalDate): List<TutorSlot>**  
Lọc danh sách slot đã được học viên đặt.
- **saveBooked(slots: List<TutorSlot>): void**  
Lưu các slot đã chuyển sang trạng thái Booked vào kho lưu trữ.
- **deleteAllByTutorIdAndDate(tutorId: Long, date: LocalDate): void**  
Xóa toàn bộ slot của một ngày cụ thể để chuẩn bị cho việc ghi đè dữ liệu mới.

### CourseRepository

- **save(course: Course): Course**  
Dùng để lưu hoặc cập nhật khóa học.



- `findById(courseID: int): Course`

Tìm khóa học theo ID.

- `findAll(): List<Course>`

Trả về tất cả khóa học.

### **RequestRepository**

- `save(request: Request): Request`

Lưu yêu cầu.

- `findByStudentID(studentID: int): List<Request>`

Trả về danh sách yêu cầu của sinh viên.

- `findByTutorID(tutorID: int): List<Request>`

Trả về danh sách yêu cầu cần tutor duyệt.

- `updateStatus(requestID: int, status: String): bool`

Cập nhật trạng thái yêu cầu.

### **FeedbackRepository**

- `save(feedback: Feedback): Feedback`

Lưu phản hồi sau buổi học.

- `findByMeetingId(meetingId: Long): List<Feedback>`

Tìm danh sách phản hồi theo mã buổi học yêu cầu.

### **UserRepository**

- `getUserById(userId: String): UserEntity`

Lấy user theo ID.

- `saveUser(user: UserEntity): void`

Lưu user mới.

- `updateUser(user: UserEntity): void`

Cập nhật user.

### **ActivityRepository**



- **getActivitiesByUserId(userId: String): List<ActivityEntity>**

Lấy danh sách hoạt động theo userId.

- **saveActivity(activity: ActivityEntity): void**

Lưu hoạt động mới.

### AppointmentRepository

- **getAppointmentsByUserId(userId: String): List<AppointmentEntity>**

Lấy danh sách lịch hẹn theo userId.

- **saveAppointment(appointment: AppointmentEntity): void**

Lưu lịch hẹn mới.

### MaterialRepository (Implementation)

- **save(m: MaterialEntity): MaterialEntity**

Thực thi thao tác lưu tài liệu vào DB.

- **update(m: MaterialEntity): MaterialEntity**

Thực thi cập nhật dữ liệu tài liệu trong DB.

- **getAllMaterials(): List<MaterialEntity>**

Truy vấn toàn bộ bản ghi tài liệu từ DB.

- **getMaterialById(materialId: int): MaterialEntity**

Thực thi câu lệnh tìm tài liệu theo ID.

- **getMaterialByUploader(tutorId: int): List<MaterialEntity>**

Truy vấn danh sách tài liệu được tải lên bởi tutor.

- **searchMaterials(criteria: SearchCriteria): List<MaterialEntity>**

Thực thi tìm kiếm tài liệu theo bộ tiêu chí.

- **getMaterialsByStatus(status: MaterialStatus): List<MaterialEntity>**

Truy vấn tài liệu dựa vào trạng thái hiện tại.

### SessionRepository

- **createSession(user: UserEntity): String**

Tạo session mới.



- **getSession(sessionId: String): SessionEntity**

Lấy session theo ID.

- **save(sessionId: String): void**

Lưu session.

- **update(sessionId: String): Boolean**

Cập nhật session, trả về true nếu thành công.

### **AvailableSlotRepository**

- **findById(tutorId: String): List<AvailableSlot>**

Tìm và trả về danh sách các AvailableSlot từ cơ sở dữ liệu dựa trên tutorId.

- **save(slot: AvailableSlot): AvailableSlot**

Lưu (thêm mới hoặc cập nhật) một đối tượng AvailableSlot vào cơ sở dữ liệu.

### **NotificationRepository**

- **save(n: Notification): Notification**

Lưu một thông báo mới vào kho dữ liệu và trả về bản ghi đã lưu.

- **saveAll(list: List<Notification>): void**

Lưu hàng loạt thông báo (ví dụ gửi cho cả lớp/khoa) trong một lần thao tác.

- **getNotifications(userId: int): List<Notification>**

Lấy toàn bộ thông báo của một người dùng để hiển thị trong hộp thư.

- **markAsRead(userId: int, notificationId: int): void**

Dánh dấu một thông báo của người dùng là đã đọc.

### **ReportRepository**

- **fetchData(userId: int, key: List<String>): Map<key,value>**

Truy vấn dữ liệu phục vụ sinh báo cáo, dựa trên người yêu cầu và các khóa tham số (key) cụ thể.

- **exportPDF(data: Map<key,value>, report: Report): File**

Xuất dữ liệu đã render thành tệp PDF theo mẫu/định nghĩa của báo cáo, trả về tệp để đính kèm hoặc tải xuống.



- **exportCSV(data: Map<key,value>, report: Report): File**

Xuất dữ liệu thành tệp CSV (bảng dữ liệu), phù hợp cho phân tích hoặc nhập vào bảng tính.

#### TutorRegistrationRepository

- **save(entity: TutorRegistrationEntity): TutorRegistrationEntity**

Lưu hoặc cập nhật entity trong database.

- **findById(id: Long): Optional<TutorRegistrationEntity>**

Tìm đăng ký theo ID.

- **findByStatus(status: TutorRegistrationStatus): List<TutorRegistrationEntity>**

Tìm tất cả đăng ký theo trạng thái.

- **findByStudentIdAndStatus(studentId: String, status: TutorRegistrationStatus): List<TutorRegistrationEntity>**

Tìm đăng ký của một học sinh với trạng thái cụ thể.

- **existsByStudentIdAndStatusIn(studentId: String, statuses: List<TutorRegistrationStatus>): boolean**

Kiểm tra học sinh có đăng ký nào ở các trạng thái cho trước không.

- **findByRequestTimeBefore(cutoff: LocalDateTime): List<TutorRegistrationEntity>**

Tìm đăng ký có thời gian request trước thời điểm cutoff.

#### 3.5.5 Repository Interface Method

#### IRoomRepository

- **listAvailableRooms(date: Date, startTime: Time, endTime: Time): List<Room>**

Trả về danh sách phòng trống trong khoảng thời gian nhất định.

- **updateRoomStatus(roomId: int, status: String): bool**

Cập nhật trạng thái phòng (AVAILABLE, OCCUPIED, MAINTENANCE). Trả về true nếu thành công.

- **getRoomInfo(roomId: int): String**

Lấy thông tin chi tiết của phòng theo roomId.



### IRoomBookingRepository

- `findBookings(date: Date): List<RoomBooking>`  
Trả về tất cả các lịch đặt phòng theo ngày.
- `save(booking: RoomBooking): void`  
Lưu một lịch đặt phòng mới.
- `findConflicts(roomId: int, date: Date, startTime: Time, endTime: Time): boolean`  
Kiểm tra phòng có xung đột lịch hay không.
- `updateStatus(roomId: int, status: String): void`

### IMeetingRepository

- `findById(meetingId: Long): Meeting`  
Tìm và trả về Meeting theo `meetingId`. Trả về `null` nếu không tồn tại.
- `save(meeting: Meeting): void`  
Lưu một Meeting mới vào danh sách nội bộ.
- `update(meeting: Meeting): void`  
Cập nhật Meeting theo `meetingId`. Ném ngoại lệ nếu không tìm thấy.
- `findPendingAppointmentsByTutor(tutorId: Long): List<Appointment>`  
Trả về danh sách các Appointment ở trạng thái PENDING của tutor.
- `findApprovedAppointmentsByTutor(tutorId: Long): List<Appointment>`  
Trả về danh sách Appointment ở trạng thái APPROVED của tutor.
- `findAllAppointmentsByStudent(studentId: Long): List<Appointment>`  
Lấy toàn bộ Appointment của một student (mọi trạng thái).
- `findApprovedAppointmentsByStudent(studentId: Long): List<Appointment>`  
Trả về Appointment đã APPROVED của student.
- `findApprovedMeetingByStudent(studentId: Long): List<Meeting>`  
Trả về Meeting đã APPROVED của student.



- **findApprovedMeetingByTutor(tutorId: Long): List<Meeting>**  
Trả về Meeting đã APPROVED của tutor.
- **findOfficialMeetingsByStudent(studentId: Long): List<Meeting>**  
Trả về Meeting APPROVED và chưa bị hủy (!isCancelled()) của student.
- **findOfficialMeetingsByTutor(tutorId: Long): List<Meeting>**  
Trả về Meeting APPROVED và chưa bị hủy của tutor.

#### **IFreeSlotRepository / FreeSlotRepository**

- **findAvailableByTutorIdAndDate(...)** – Lấy slot rảnh trong ngày.
- **findAvailableByTutorIdAndDateBetween(...)** – Lấy slot rảnh trong tháng.
- **saveAvailable(...)** – Lưu slot rảnh mới.
- **findBookedByTutorIdAndDate(...)** – Lấy slot đã đặt.
- **saveBooked(...)** – Lưu slot đã đặt.
- **deleteAllByTutorIdAndDate(...)** – Xóa toàn bộ slot của ngày.

#### **IMaterialRepository**

- **save(m: MaterialEntity): MaterialEntity**  
Lưu tài liệu mới vào cơ sở dữ liệu.
- **update(m: MaterialEntity): MaterialEntity**  
Cập nhật nội dung hoặc thông tin của một tài liệu đã tồn tại.
- **getAllMaterials(): List<MaterialEntity>**  
Trả về danh sách toàn bộ tài liệu.
- **getMaterialById(materialId: int): MaterialEntity**  
Lấy một tài liệu dựa trên ID.
- **getMaterialByUploader(tutorId: int): List<MaterialEntity>**  
Trả về danh sách tài liệu do một tutor cụ thể tải lên.
- **searchMaterials(criteria: SearchCriteria): List<MaterialEntity>**  
Tìm kiếm tài liệu dựa trên tiêu chí (tên, lớp, loại, từ khóa...).



- **getMaterialsByStatus(status: MaterialStatus): List<MaterialEntity>**

Lấy danh sách tài liệu theo trạng thái (Draft, Pending, Published...).

### 3.5.6 Service Interface Method

#### **IFreeSlotService / FreeSlotService**

- **getDailySchedule(tutorId: Long, date: LocalDate): FreeSlotResponse** – Lấy Entity và chuyển sang DTO.
- **getMonthlySchedule(tutorId: Long, month: int, year: int): List<FreeSlotResponse>** – Gom nhóm slot theo ngày.
- **overwriteDailySchedule(tutorId: Long, request: FreeSlotRequest): List<String>** – Xóa, tạo và lưu slot mới.
- **reserveSlot(...)** – Chuyển slot từ Rảnh → Đã đặt.
- **releaseSlot(...)** – Trả slot về trạng thái Rảnh.
- **getRawAvailableSlots(...)** – Lấy danh sách Entity trực tiếp.

#### **ISSOServices**

- **authenticate(username: String, password: String): UserEntity**

Xác thực username/password, trả về UserEntity nếu thành công.

- **sendResetEmail(bkId: String): void**

Gửi email reset password.

- **resetPassword(token: String, newPassword: String): void**

Reset password dựa trên token.

#### **IMailService**

- **sendEmail(to: String, content: String): void**

Gửi email.

- **sendPasswordResetEmail(to: String, resetLink: String): void**

Gửi link reset password.



- **sendNotificationEmail(to: String, subject: String, content: String): void**

Gửi email thông báo.

#### IDatacoreService

- **getUserInfo(username: String): UserEntity**

Lấy thông tin user từ Datacore.

#### IProfileService

- **getProfile(userId: String): UserEntity**

Trả về thông tin user.

- **getActivityHistory(userId: String): List<ActivityEntity>**

Trả về danh sách hoạt động của user.

- **getAppointmentHistory(userId: String): List<AppointmentEntity>**

Trả về danh sách lịch hẹn của user.

#### ISessionManager

- **createSession(user: UserEntity): String**

Tạo session mới.

- **getSession(sessionId: String): SessionEntity**

Lấy session theo ID.

- **invalidateSession(sessionId: String): void**

Vô hiệu hóa session.

- **isValid(sessionId: String): Boolean**

Kiểm tra session còn hợp lệ.

#### IMaterialService

- **getMaterials(): List<MaterialEntity>**

Lấy danh sách tài liệu.

- **getMaterialContent(materialId: String): String**

Lấy nội dung chi tiết tài liệu.



- **searchMaterials(criteria: SearchCriteria): List<MaterialEntity>**

Tìm kiếm tài liệu theo tiêu chí.

#### **IMaterialUploadService**

- **createDraft(tutorId: String, title: String, type: String, classId: String): Material**

Tạo một tài liệu ở trạng thái nháp (draft) cho tutor trước khi upload nội dung.

- **attachFile(materialId: String, fileName: String, bytes: byte[]): Material**

Dính kèm file (PDF, PPT, DOCX, hình ảnh...) vào tài liệu nháp.

- **submitForReview(materialId: String): void**

Tutor gửi tài liệu sang trạng thái “chờ duyệt” để Admin/Tutor Leader xem và phê duyệt.

- **publish(materialId: String): void**

Xuất bản tài liệu đã được duyệt để sinh viên có thể xem và tải về.

- **listMyUploads(tutorId: String): List<Material>**

Trả về danh sách các tài liệu mà tutor đã tải lên (bao gồm draft, pending, published).

#### **IAvailableSlotRepository**

- **findByIdTutorId(id: String): List<AvailableSlot>**

Định nghĩa phương thức tìm kiếm các slot dựa trên tutorId.

- **save(slot: AvailableSlot): AvailableSlot**

Định nghĩa phương thức lưu (thêm mới hoặc cập nhật) một AvailableSlot.

#### **IStudentSchedulingService**

- **bookAppointment(studentId: Long, tutorId: Long, date: LocalDateTime, startTime: LocalDateTime, endTime: LocalDateTime, topic: String): bool**

Tạo một lịch hẹn mới giữa sinh viên và tutor. Logic bao gồm: kiểm tra tutor có trống hay không trong khoảng thời gian yêu cầu, kiểm tra slot rảnh từ FreeSlotService, lưu Appointment vào Repository, và cất slot rảnh tương ứng. Trả về true nếu đặt thành công, false nếu không còn slot hoặc xảy ra lỗi.

- **viewOfficialMeetings(studentId: Long): List<Meeting>**

Lấy danh sách các Meeting chính thức của sinh viên, bao gồm Appointment đã duyệt (APPROVED) và các loại Meeting khác. Dữ liệu lấy từ MeetingRepository.



- **viewAppointmentHistory(studentId: Long): List<Appointment>**  
Trả về lịch sử tất cả các Appointment của sinh viên, bao gồm PENDING, APPROVED, REJECTED, CANCELLED. Phục vụ mục xem lịch sử đặt lịch.
- **cancelMeeting(meetingId: Long, reason: String): bool**  
Sinh viên yêu cầu hủy một Meeting. Kiểm tra Meeting tồn tại, chưa bị hủy, thuộc về sinh viên. Cập nhật trạng thái thành CANCELLED, lưu lý do hủy, và nếu Appointment đã được duyệt thì trả lại slot vào FreeSlotService. Trả về true nếu hủy thành công.
- **viewTutorAvailableSlots(tutorId: Long): List<FreeSlotResponse>**  
Lấy danh sách các slot rảnh của tutor (dạng date + list of time ranges). Phục vụ UI để hiển thị lịch rảnh.
- **findCancellableMeetings(studentId: Long): List<Meeting>**  
Trả về danh sách Meeting mà sinh viên có quyền hủy: là các Meeting APPROVED với startTime nằm trong tương lai.
- **viewMeetingDetails(meetingId: Long): Meeting**  
Lấy chi tiết đầy đủ của một Meeting (bao gồm thông tin tutor, thời gian, online link, trạng thái...). Trả về null nếu MeetingId không tồn tại.

### ITutorSchedulingService

- **viewPendingAppointments(tutorId: Long): List<Appointment>**  
Lấy danh sách Appointment đang ở trạng thái PENDING của tutor. Dùng cho UI để hiển thị các yêu cầu chờ duyệt.
- **approveAppointment(appointmentId: Long, tutorId: Long): bool**  
Tutor duyệt Appointment. Logic: kiểm tra quyền sở hữu, kiểm tra Appointment chưa được duyệt hoặc từ chối, cập nhật trạng thái thành APPROVED, tạo onlineLink, và lưu lại vào Repository. Trả về true nếu duyệt thành công.
- **rejectAppointment(appointmentId: Long, tutorId: Long, reason: String): bool**  
Tutor từ chối Appointment. Kiểm tra quyền hợp lệ, cập nhật trạng thái thành REJECTED, lưu lý do từ chối. Trả về true nếu thao tác thành công.
- **viewOfficialMeetings(tutorId: Long): List<Meeting>**  
Trả về Meetings chính thức của tutor, bao gồm các Appointment đã duyệt và các Meeting khác có trong Repository.



- cancelMeeting(tutorId: Long, meetingId: Long, reason: String): bool  
Tutor hủy buổi Meeting. Kiểm tra Meeting thuộc tutor, chưa bị hủy, cập nhật trạng thái CANCELLED, lưu lý do. Trả về true nếu hủy thành công.
- tutorReturnCancelledSlot(tutorId: Long, meetingId: Long): bool  
Sau khi hủy Meeting, tutor có tùy chọn trả slot vào lịch rảnh. Kiểm tra Meeting đúng tutor, đúng trạng thái CANCELLED, rồi gọi FreeSlotService để trả lại slot. Trả về true nếu slot được trả, false nếu tutor chọn không trả.
- findCancellableMeetings(tutorId: Long): List<Meeting>  
Trả về các Meeting mà tutor có quyền hủy (APPROVED và xảy ra trong tương lai).
- viewMeetingDetails(meetingId: Long): Meeting  
Lấy thông tin chi tiết của một Meeting theo ID. Trả về null nếu không tồn tại.
- createOnlineLink(appointment: Appointment): String  
Tạo đường link học online cho Appointment (ví dụ: tạo link Google Meet). Được gọi khi tutor approve lịch. Trả về chuỗi URL.
- viewAppointmentDetails(appointmentId: Long): Appointment  
Lấy thông tin chi tiết của một cuộc hẹn (Appointment). Nếu ID không tồn tại hoặc Meeting không phải là Appointment thì trả về null.

### **IFeedbackService**

- addFeedback(meetingId: Long, rating: int, comment: String): Feedback  
Lưu phản hồi đánh giá buổi học
- getFeedbackByMeeting(meetingId: Long): List<Feedback>  
Lấy ra danh sách phản hồi đánh giá một buổi học theo ID.

### **ITutorRegistrationRepository**

- save(entity: TutorRegistrationEntity): TutorRegistrationEntity  
Lưu hoặc cập nhật entity trong database.
- findById(id: Long): Optional<TutorRegistrationEntity>  
Tìm đăng ký theo ID.



- **findByStatus(status: TutorRegistrationStatus): List<TutorRegistrationEntity>**  
Tìm tất cả đăng ký theo trạng thái.
- **findByStudentIdAndStatus(studentId: String, status: TutorRegistrationStatus): List<TutorRegistrationEntity>**  
Tìm đăng ký của một học sinh với trạng thái cụ thể.
- **existsByStudentIdAndStatusIn(studentId: String, statuses: List<TutorRegistrationStatus>): boolean**  
Kiểm tra học sinh có đăng ký nào ở các trạng thái cho trước không.
- **findByRequestTimeBefore(cutoff: LocalDateTime): List<TutorRegistrationEntity>**  
Tìm đăng ký có thời gian request trước thời điểm cutoff.

### 3.5.7 Entity Method

#### Meeting (abstract)

- **cancel(userId, reason): bool**  
Hủy buổi gặp mặt, lưu lý do.
- **updateStatus(): void**  
Cập nhật trạng thái theo thời gian (SCHEDULED → ONGOING → COMPLETED).
- Getter/Setter cho: meetingId, tutorId, date, startTime, endTime, topic, cancelled, cancellationReason.

#### Appointment (extends Meeting)

- **approve(tutorId): bool**  
Duyệt lịch hẹn PENDING → APPROVED.
- **reject(tutorId): bool**  
Từ chối lịch hẹn.
- **cancel(userId, reason): bool**  
Override phương thức từ Meeting.
- Getter/Setter cho: studentId, status (PENDING/APPROVED/REJECTED).



### Consultation (extends Meeting)

- register(studentId): bool

Thêm student vào participants.

- cancel(userId, reason): bool

Override phương thức từ Meeting.

- Getter/Setter cho: title, mode, room, onlineLink, maxParticipants, participants, status (SCHEDULED/ONGOING/COMPLETED/CANCELLED).

### Room

- getId(): int

Trả về mã phòng.

- getStatus(): String

Trả về trạng thái phòng.

- setStatus(status: String): void

Cập nhật trạng thái phòng.

### RoomBooking

- getRoomId(): int

Trả về mã phòng.

- getDate(): Date

Trả về ngày.

- getStartTime(): Time

Trả về giờ bắt đầu.

- getEndTime(): Time

Trả về giờ kết thúc.

- getMeetingId(): int

Trả về mã cuộc họp.

### User



- **login(): bool**

Thực hiện đăng nhập.

- **logout(): bool**

Thực hiện đăng xuất.

- **viewProfile(): void**

Xem thông tin cá nhân.

- **viewMaterials(): void**

Xem danh sách tài liệu.

- **forgotPassword(): void**

Yêu cầu đặt lại mật khẩu.

### Student

- **requestTutor(tutorId: int): bool**

Yêu cầu tư vấn từ tutor.

- **bookAppointment(tutorId: int, datetime): bool**

Đặt lịch hẹn với tutor.

- **giveFeedback(sessionId: int, rating: int, comment: String): bool**

Gửi phản hồi về buổi tư vấn.

- **registerCounselingSession(tutorId: int, datetime): bool**

Đăng ký buổi tư vấn với tutor.

### Tutor

- **acceptRequest(requestId: int): bool**

Chấp nhận yêu cầu tư vấn.

- **createCounselingSession(studentId: int, datetime): bool**

Tạo buổi tư vấn cho student.

- **cancelAppointment(appointmentId: int, reason: String): bool**

Hủy một buổi hẹn.



- **recordStudentProgress(studentId: int, summary: String): bool**  
Ghi lại tiến độ học tập của student.
- **setAvailability(startTime: datetime, endTime: datetime): bool**  
Cập nhật khung thời gian có sẵn.
- **uploadMaterials(materialId: int, content: String): bool**  
Tải lên tài liệu học tập.
- **deleteMaterials(materialId: int): bool**  
Xóa tài liệu học tập.
- **processAppointmentRequest(requestId: int): bool**  
Xử lý yêu cầu đặt lịch hẹn.

#### Feedback

- **getFeedbackId(): Long**  
Trả về mã đánh giá
- **getMeetingId(): Long**  
Trả về mã buổi học
- **getRating(): int**  
Trả về đánh giá (từ 1 đến 5)
- **getComment(): String**  
Trả về nhận xét đánh giá
- **getSubmitAt(): LocalDateTime**  
Trả về thời điểm đánh giá

#### AvailableSlot

- **getStatus(): SlotStatus**  
Trả về trạng thái hiện tại của slot (ví dụ: AVAILABLE, BOOKED).
- **setStatus(status: SlotStatus): void**  
Cập nhật trạng thái của slot.



- **isOverlapping(other: AvailableSlot): boolean**

Kiểm tra xem slot này có bị trùng lặp (chồng chéo) thời gian với một slot **other** hay không.

- **isAvailable(): boolean**

Kiểm tra nhanh xem trạng thái của slot có phải là AVAILABLE hay không.

### Notification

- **id: int**

ID thông báo.

- **userId: int**

ID người nhận thông báo.

- **type: String**

Loại thông báo, dùng để phân nhóm xử lý.

- **content: String**

Nội dung tóm tắt hiển thị cho người dùng.

- **createdAt: datetime**

Thời điểm hệ thống tạo thông báo.

- **readAt: datetime**

Thời điểm người dùng đánh dấu đã đọc.

### Report (Entity)

- **reportId: int**

ID báo cáo.

- **name: string**

Tên của mẫu báo cáo.

- **description: string**

Mô tả ngắn về nội dung của báo cáo.

- **format: ReportFormat**

Định dạng xuất (PDF, CSV).



- **title: string**

Tiêu đề hiển thị.

- **sendTo: Department**

Phòng ban nhận báo cáo.

- **createAt: datetime**

Thời điểm tạo báo cáo.

### **TutorSlot**

- **isOverlapping(other: TutorSlot): boolean** – Kiểm tra trùng lặp slot.

**UserEntity / SessionEntity** – Chứa thông tin người dùng và phiên làm việc.

## **3.6 Test case**



### 3.6.1 Đăng nhập

<b>Test case</b>	UC01-TC01 - Đăng nhập thành công qua HCMUT_SSO
<b>Test description</b>	User có tài khoản hợp lệ và có quyền sử dụng hệ thống, đăng nhập thành công thông qua HCMUT_SSO, phiên làm việc được tạo và được chuyển đến trang chính tương ứng với vai trò.
<b>Related screens</b>	- Trang chủ Tutor Support System với nút “Đăng nhập qua HCMUT_SSO”. - Trang đăng nhập HCMUT_SSO (form BKNetID + mật khẩu). - Trang chính sau đăng nhập (dashboard tương ứng role: Sinh viên / Tutor / Điều phối viên).
<b>Pre-conditions</b>	1. User có tài khoản hợp lệ trong hệ thống HCMUT_SSO (BKNetID + mật khẩu đúng). 2. Tài khoản thuộc nhóm được phép dùng Tutor Support System (role hợp lệ). 3. HCMUT_SSO và HCMUT_DATACORE đang hoạt động bình thường. 4. User chưa đăng nhập vào Tutor Support System trên trình duyệt hiện tại.
<b>Actions</b>	1. User truy cập trang chủ Tutor Support System. 2. User nhấn nút “Đăng nhập qua HCMUT_SSO”. 3. Hệ thống chuyển hướng sang trang đăng nhập HCMUT_SSO. 4. User nhập đúng BKNetID và mật khẩu, sau đó nhấn nút đăng nhập. 5. HCMUT_SSO xác thực thông tin đăng nhập thành công và trả về token xác thực cho Tutor Support System. 6. Tutor Support System dùng token truy vấn HCMUT_DATACORE để lấy thông tin cơ bản (họ tên, email, role, trạng thái tài khoản, ...). 7. Hệ thống kiểm tra role và trạng thái tài khoản, tạo phiên làm việc cho user. 8. Hệ thống chuyển hướng user đến trang chính tương ứng với vai trò (dashboard Sinh viên / Tutor / Điều phối viên).
<b>Inputs</b>	- BKNetID: tài khoản hợp lệ - Mật khẩu: mật khẩu tương ứng, hợp lệ.
<b>Expected Outputs</b>	- User được xác thực thành công từ HCMUT_SSO. - Dữ liệu người dùng được lấy thành công từ HCMUT_DATACORE. - Phiên làm việc (session) của user được tạo trong Tutor Support System. - User được chuyển tới đúng trang chính theo vai trò, không có thông báo lỗi.



Testing environment	Web
---------------------	-----



<b>Test case</b>	UC01-TC02 - Sai thông tin đăng nhập (mật khẩu/BKNetID không đúng)
<b>Test description</b>	User nhập sai BKNetID hoặc mật khẩu tại HCMUT_SSO, hệ thống hiển thị thông báo lỗi trên form và không đăng nhập vào Tutor Support System.
<b>Related screens</b>	- Trang chủ Tutor Support System. - Trang đăng nhập HCMUT_SSO (form đăng nhập).
<b>Pre-conditions</b>	1. HCMUT_SSO đang hoạt động bình thường. 2. User chưa đăng nhập vào Tutor Support System.
<b>Actions</b>	1. User truy cập trang chủ Tutor Support System và nhấn “Đăng nhập qua HCMUT_SSO”. 2. Hệ thống chuyển sang trang đăng nhập HCMUT_SSO. 3. User nhập BKNetID hoặc mật khẩu không chính xác, nhấn đăng nhập.
<b>Inputs</b>	- BKNetID(đúng) hoặc một mã bất kỳ. - Mật khẩu không khớp mật khẩu thật
<b>Expected Outputs</b>	- HCMUT_SSO hiển thị thông báo lỗi trực tiếp trên form, ví dụ: “Thông tin đăng nhập không chính xác.”. - User vẫn ở trên trang đăng nhập HCMUT_SSO, không được chuyển hướng về Tutor Support System. - Không có phiên làm việc nào được tạo ở Tutor Support System .
<b>Testing environment</b>	Web



<b>Test case</b>	UC01-TC03 - Tài khoản bị khóa / hết hạn / yêu cầu đổi mật khẩu
<b>Test description</b>	User nhập đúng BKNetID và mật khẩu nhưng tài khoản ở trạng thái bị khóa, hết hạn hoặc bắt buộc đổi mật khẩu; HCMUT_SSO hiển thị thông báo tương ứng và user không đăng nhập được vào Tutor Support System.
<b>Related screens</b>	Trang đăng nhập HCMUT_SSO (form đăng nhập).
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>Tài khoản BKNetID tồn tại nhưng đang ở trạng thái: bị khóa / hết hạn / yêu cầu đổi mật khẩu.</li><li>HCMUT_SSO hoạt động bình thường.</li><li>User chưa có phiên đăng nhập Tutor Support System.</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>User từ trang chủ Tutor Support System nhấn “Đăng nhập qua HCMUT_SSO” để mở trang đăng nhập SSO.</li><li>User nhập đúng BKNetID và mật khẩu của tài khoản đang bị khóa/hết hạn/yêu cầu đổi mật khẩu.</li><li>User nhấn đăng nhập.</li></ol>
<b>Inputs</b>	<ul style="list-style-type: none"><li>BKNetID: tài khoản tồn tại nhưng bị khóa/hết hạn.</li><li>Mật khẩu: hợp lệ với tài khoản đó.</li></ul>
<b>Expected Outputs</b>	<ul style="list-style-type: none"><li>HCMUT_SSO hiển thị thông báo lỗi tương ứng trên form đăng nhập (E2.1).</li><li>User không được chuyển hướng về Tutor Support System, không có token xác thực được trả về.</li><li>Use case kết thúc không thành công tại SSO, không tạo phiên làm việc ở Tutor Support System (E2.2).</li></ul>
<b>Testing environment</b>	Web



<b>Test case</b>	UC01-TC04- Đăng nhập bằng tài khoản không có quyền sử dụng Tutor Support System
<b>Test description</b>	User đăng nhập SSO thành công nhưng role không thuộc nhóm được phép truy cập Tutor Support System; hệ thống hiển thị thông báo “Bạn không có quyền truy cập hệ thống.” và không tạo phiên làm việc.
<b>Related screens</b>	- Trang chủ Tutor Support System. - Trang đăng nhập HCMUT_SSO. - Trang thông báo lỗi quyền truy cập của Tutor Support System.
<b>Pre-conditions</b>	1. User có tài khoản hợp lệ trong HCMUT_SSO. 2. Role của user <b>không</b> nằm trong tập role được phép dùng Tutor Support System. 3. HCMUT_SSO và HCMUT_DATACORE hoạt động bình thường.
<b>Actions</b>	1. User từ trang chủ Tutor Support System nhấn “Đăng nhập qua HC-MUT_SSO”. 2. User đăng nhập thành công trên HCMUT_SSO (đúng BKNetID và mật khẩu). 3. Tutor Support System nhận token, truy vấn HCMUT_DATACORE lấy thông tin user. 4. Hệ thống kiểm tra role, phát hiện user không thuộc nhóm được phép sử dụng.
<b>Inputs</b>	- BKNetID + mật khẩu của user có role không hợp lệ (ví dụ thuộc hệ thống khác).
<b>Expected Outputs</b>	- Tutor Support System hiển thị thông báo: “Bạn không có quyền truy cập hệ thống.” . - Phiên đăng nhập không được tạo; user không truy cập được vào bất kỳ trang chức năng nào. - Use case kết thúc không thành công, nhưng phiên SSO có thể vẫn tồn tại độc lập.
<b>Testing environment</b>	Web



<b>Test case</b>	UC01-TC05 - Lỗi kết nối đến HCMUT_SSO
<b>Test description</b>	Tutor Support System không thể kết nối đến dịch vụ HCMUT_SSO khi user yêu cầu đăng nhập; hệ thống báo lỗi và use case kết thúc không thành công.
<b>Related screens</b>	Trang chủ / trang đăng nhập của Tutor Support System (nơi hiển thị nút “Đăng nhập qua HCMUT_SSO”).
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>HCMUT_SSO bị ngừng hoạt động hoặc kết nối mạng từ Tutor Support System đến SSO bị lỗi.</li><li>User chưa đăng nhập vào Tutor Support System.</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>User truy cập trang chủ Tutor Support System.</li><li>User nhấn nút “Đăng nhập qua HCMUT_SSO”.</li><li>Hệ thống cố gắng chuyển hướng/kết nối tới HCMUT_SSO nhưng bị lỗi kết nối.</li></ol>
<b>Inputs</b>	Không có dữ liệu nhập từ user ngoài thao tác nhấn nút đăng nhập SSO.
<b>Expected Outputs</b>	<ul style="list-style-type: none"><li>Tutor Support System hiển thị thông báo: “Không thể kết nối đến dịch vụ xác thực, vui lòng thử lại sau.” .</li><li>Không có phiên làm việc được tạo, user vẫn ở trang login/landing của Tutor Support System.</li></ul>
<b>Testing environment</b>	Web.



<b>Test case</b>	UC01-TC06 - Lỗi đồng bộ dữ liệu từ HCMUT_DATACORE
<b>Test description</b>	User đăng nhập SSO thành công, nhưng Tutor Support System không lấy được thông tin người dùng từ HCMUT_DATACORE; hệ thống hiển thị thông báo lỗi, hủy phiên đăng nhập và không cho truy cập hệ thống.
<b>Related screens</b>	- Trang đăng nhập HCMUT_SSO. - Trang thông báo lỗi của Tutor Support System sau khi nhận token.
<b>Pre-conditions</b>	1. User có tài khoản hợp lệ và đăng nhập được vào HCMUT_SSO. 2. Dịch vụ HCMUT_DATACORE gặp lỗi (không truy cập được, trả về lỗi, hoặc dữ liệu không đầy đủ).
<b>Actions</b>	1. User từ trang chủ Tutor Support System chọn “Đăng nhập qua HCMUT_SSO” và đăng nhập thành công trên SSO. 2. Tutor Support System nhận token từ HCMUT_SSO. 3. Hệ thống dùng token truy vấn HCMUT_DATACORE để lấy thông tin người dùng. 4. Việc truy vấn thất bại hoặc trả về dữ liệu lỗi/thiếu.
<b>Inputs</b>	- BKNetID + mật khẩu hợp lệ (đăng nhập SSO thành công).
<b>Expected Outputs</b>	- Tutor Support System hiển thị thông báo: “Không thể tải thông tin người dùng, vui lòng thử lại sau.” . - Hệ thống hủy phiên đăng nhập hiện tại, không cho user truy cập bất kỳ chức năng nào . - Use case kết thúc không thành công (E5.3).
<b>Testing environment</b>	Web.



### 3.6.2 Đăng xuất

<b>Test case</b>	UC02-TC01 - Đăng xuất thành công
<b>Test description</b>	User đang đăng nhập và chủ động chọn chức năng Đăng xuất; hệ thống kết thúc phiên làm việc, vô hiệu hóa token và chuyển về trang đăng nhập với thông báo thành công.
<b>Related screens</b>	- Trang bất kỳ trong Tutor Support System có hiển thị menu người dùng / biểu tượng tài khoản. - Hộp thoại xác nhận “Bạn có chắc chắn muốn đăng xuất?”. - Trang đăng nhập Tutor Support System.
<b>Pre-conditions</b>	1. User (Sinh viên / Tutor / Điều phối viên) đã đăng nhập thành công vào hệ thống. 2. Phiên làm việc của user đang hoạt động, token xác thực còn hiệu lực.
<b>Actions</b>	1. User nhấn vào biểu tượng tài khoản hoặc menu người dùng ở góc trên. 2. Hệ thống hiển thị menu với tùy chọn “Đăng xuất”. 3. User chọn mục “Đăng xuất”. 4. Hệ thống hiển thị hộp thoại xác nhận “Bạn có chắc chắn muốn đăng xuất?” với hai nút: “Đăng xuất” và “Hủy”. 5. User chọn nút xác nhận Đăng xuất. 6. Hệ thống vô hiệu hóa token xác thực hiện tại. 7. Hệ thống xóa thông tin phiên làm việc (session, cookie liên quan). 8. Hệ thống chuyển hướng user về trang đăng nhập. 9. Hệ thống hiển thị thông báo “Đăng xuất thành công!” trên trang đăng nhập.
<b>Inputs</b>	Không có dữ liệu nhập dạng text; chỉ có thao tác bấm menu và xác nhận Đăng xuất.
<b>Expected Outputs</b>	- Phiên làm việc của user bị xóa; token xác thực bị vô hiệu hóa. - User được chuyển về trang đăng nhập, không thể truy cập lại các trang nội bộ nếu không đăng nhập lại. - Thông báo “Đăng xuất thành công!” được hiển thị rõ ràng.
<b>Testing environment</b>	Web



<b>Test case</b>	UC02-TC02 - Hủy thao tác đăng xuất
<b>Test description</b>	User chọn Đăng xuất nhưng bấm “Hủy” trong hộp thoại xác nhận; hệ thống đóng hộp thoại và user tiếp tục sử dụng hệ thống như bình thường.
<b>Related screens</b>	Trang bất kỳ trong hệ thống + hộp thoại xác nhận Đăng xuất.
<b>Pre-conditions</b>	1. User đã đăng nhập và phiên làm việc đang hoạt động.
<b>Actions</b>	1. User mở menu người dùng và chọn “Đăng xuất”. 2. Hệ thống hiển thị hộp thoại xác nhận “Bạn có chắc chắn muốn đăng xuất?”. 3. User chọn nút “Hủy” trên hộp thoại xác nhận.
<b>Inputs</b>	Thao tác bấm vào nút “Hủy” trên hộp thoại xác nhận Đăng xuất.
<b>Expected Outputs</b>	- Hệ thống đóng hộp thoại xác nhận (không còn hiển thị trên màn hình). - Phiên làm việc và token xác thực của user vẫn giữ nguyên, không bị xóa. - User vẫn ở lại trang hiện tại và tiếp tục sử dụng hệ thống bình thường.
<b>Testing environment</b>	Chrome / Windows 10; hệ thống backend hoạt động bình thường.



<b>Test case</b>	UC02-TC03 - Lỗi khi vô hiệu hóa token đăng nhập
<b>Test description</b>	Trong quá trình đăng xuất, hệ thống gặp lỗi khi gọi backend để vô hiệu hóa token; hệ thống ghi log lỗi, xóa thông tin phiên làm việc phía client và chuyển user về trang đăng nhập.
<b>Related screens</b>	- Trang bất kỳ trong hệ thống. - Hộp thoại xác nhận Đăng xuất. - Trang đăng nhập.
<b>Pre-conditions</b>	1. User đã đăng nhập và phiên làm việc đang hoạt động. 2. Môi trường kiểm thử được cấu hình mô phỏng lỗi ở bước vô hiệu hóa token trên server (backend trả lỗi).
<b>Actions</b>	1. User mở menu người dùng và chọn “Đăng xuất”. 2. Hệ thống hiển thị hộp thoại xác nhận, user chọn xác nhận Đăng xuất. 3. Hệ thống gửi yêu cầu vô hiệu hóa token đến backend và nhận lỗi.
<b>Inputs</b>	Thao tác xác nhận Đăng xuất; không có input text.
<b>Expected Outputs</b>	- Backend ghi lại log lỗi vô hiệu hóa token (E1.1). - Ở phía client, hệ thống vẫn xóa thông tin phiên làm việc hiện tại (session, cookie) . - User được chuyển hướng về trang đăng nhập; từ đó không thể truy cập lại trang nội bộ nếu chưa đăng nhập lại . - Có thể (tùy thiết kế) hiển thị thông báo chung “Đã đăng xuất, nhưng có lỗi nội bộ, vui lòng đăng nhập lại sau.”.
<b>Testing environment</b>	Web



<b>Test case</b>	UC02-TC04 - Phiên làm việc đã hết hạn trước khi người dùng chọn Đăng xuất
<b>Test description</b>	Phiên làm việc của user đã hết hạn do timeout; khi user thao tác Đăng xuất, hệ thống phát hiện session hết hạn và chuyển trực tiếp về trang đăng nhập với thông báo tương ứng.
<b>Related screens</b>	Trang bất kỳ trong hệ thống (có thể cũ) và trang đăng nhập.
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>User đã từng đăng nhập vào hệ thống.</li><li>Thời gian không hoạt động vượt quá thời gian timeout phiên; session trên server đã bị hủy.</li><li>Trình duyệt của user vẫn còn trang hệ thống mở (UI chưa refresh).</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>User trên trang cũ nhấn vào menu người dùng và chọn “Đăng xuất”.</li><li>Hệ thống gửi yêu cầu đăng xuất, nhưng phát hiện phiên làm việc đã hết hạn.</li></ol>
<b>Inputs</b>	Thao tác chọn “Đăng xuất” trên UI sau khi phiên đã hết hạn.
<b>Expected Outputs</b>	<ul style="list-style-type: none"><li>Hệ thống không xử lý như đăng xuất thông thường mà nhận diện phiên đã hết hạn.</li><li>User được chuyển hướng ngay về trang đăng nhập.</li><li>Trên trang đăng nhập hiển thị thông báo, ví dụ: “Phiên làm việc đã hết hạn.”.</li><li>User không còn truy cập được các trang nội bộ nếu không đăng nhập lại.</li></ul>
<b>Testing environment</b>	Web



### 3.6.3 Đăng ký Tutor

<b>Test case</b>	UC07-TC01 — Đăng ký Tutor thành công
<b>Test description</b>	Sinh viên đăng ký Tutor với đầy đủ thông tin hợp lệ
<b>Related screens</b>	Màn hình Đăng ký Tutor
<b>Pre-conditions</b>	1. Sinh viên đã đăng nhập 2. Chương trình Tutor đang mở đăng ký
<b>Actions</b>	1. Sinh viên mở chức năng “Đăng ký Tutor” 2. Hệ thống hiển thị giao diện chọn môn học/lĩnh vực 3. Sinh viên chọn môn học Nguyên lý ngôn ngữ lập trình và bấm “Tiếp theo” 4. Hệ thống hiển thị danh sách tutor phù hợp 5. Sinh viên chọn đăng ký tutor Nguyễn Thị B và bấm xác nhận 6. Hệ thống hiển thị màn hình Chờ duyệt 12h với nút “Hủy đăng ký”
<b>Inputs</b>	Môn học: Nguyên lý ngôn ngữ lập trình; Tutor: Nguyễn Thị B
<b>Expected Outputs</b>	1. Hệ thống hiển thị danh sách tutor phù hợp 2. Hệ thống hiển thị trạng thái Chờ duyệt 12h 3. Yêu cầu đăng ký được lưu trong hệ thống 4. Nút Hủy đăng ký khả dụng
<b>Testing environment</b>	Web – Windows 11



<b>Test case</b>	UC07-TC02 — Bỏ trống thông tin môn học
<b>Test description</b>	Sinh viên không chọn môn học ở bước 1
<b>Related screens</b>	Màn hình nhập thông tin
<b>Pre-conditions</b>	1. Sinh viên đã đăng nhập 2. Chương trình Tutor đang mở đăng ký
<b>Actions</b>	1. Sinh viên mở chức năng “Đăng ký Tutor” 2. Hệ thống hiển thị giao diện chọn môn học 3. Sinh viên bỏ trống môn học 4. Sinh viên Bấm “Tiếp theo”
<b>Inputs</b>	Môn học: (trống)
<b>Expected Outputs</b>	1. Hệ thống hiển thị thông báo lỗi “Vui lòng nhập đầy đủ thông tin” 2. Không chuyển sang màn hình hiển thị danh sách tutor.
<b>Testing environment</b>	Web – Windows 11



<b>Test case</b>	UC07-TC03 — Hủy đăng ký trong khi chờ duyệt và đăng ký mới
<b>Test description</b>	Sinh viên hủy yêu cầu đăng ký tutor đang chờ duyệt và ngay lập tức thực hiện đăng ký mới
<b>Related screens</b>	Màn hình Chờ duyệt 12h
<b>Pre-conditions</b>	1. Sinh viên đã đăng nhập 2. Yêu cầu đăng ký tutor đang ở trạng thái Chờ duyệt 12h
<b>Actions</b>	1. Sinh viên mở chức năng “Đăng ký Tutor” 2. Hệ thống hiển thị màn hình Chờ duyệt 12h 3. Nhấn nút Hủy đăng ký 4. Xác nhận hủy 5. Hệ thống hiển thị thông báo “Hủy thành công” 6. Sinh viên chọn Đăng kí mới 7. Hệ thống quay về hiển thị danh sách tutor phù hợp 8. Sinh viên chọn tutor mới và bấm “Xác nhận”
<b>Inputs</b>	Tutor: Nguyễn Thị B
<b>Expected Outputs</b>	1. Hệ thống hiển thị “Hủy thành công” 2. Cập nhật trạng thái slot tutor cũ 3. Nếu chọn đăng ký mới → hệ thống hiển thị danh sách tutor 4. Sinh viên có thể chọn tutor mới và hoàn tất đăng ký
<b>Testing environment</b>	Web – Windows 11



### 3.6.4 Đặt lịch hẹn

<b>Test case</b>	UC08-TC01 — Đặt lịch hẹn thành công
<b>Test description</b>	Sinh viên đặt lịch hẹn với tutor dựa trên lịch rảnh và nội dung hỗ trợ hợp lệ
<b>Related screens</b>	Màn hình Đặt lịch hẹn
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>Sinh viên đã đăng nhập</li><li>Sinh viên đã chọn tutor</li><li>Tutor đã cung cấp lịch rảnh</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>Sinh viên mở chức năng “Đặt lịch hẹn”</li><li>Hệ thống hiển thị lịch rảnh của tutor</li><li>Sinh viên chọn khoảng thời gian phù hợp và nhập nội dung hỗ trợ</li><li>Nhấn “Xác nhận”</li></ol>
<b>Inputs</b>	Tutor: Nguyễn Thị B, Ngày giờ: 15/11/2025 14:00-15:00, Nội dung: Hướng dẫn bài tập Nguyên lý ngôn ngữ lập trình
<b>Expected Outputs</b>	<ol style="list-style-type: none"><li>Hệ thống lưu yêu cầu lịch hẹn ở trạng thái Chờ duyệt</li><li>Tutor nhận thông báo yêu cầu mới</li></ol>
<b>Testing environment</b>	Web – Windows 11



<b>Test case</b>	UC08-TC02 — Đặt lịch hẹn thiếu thông tin
<b>Test description</b>	Sinh viên không nhập nội dung
<b>Related screens</b>	Màn hình Đặt lịch hẹn
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>Sinh viên đã đăng nhập</li><li>Sinh viên đã chọn tutor</li><li>Tutor đã cung cấp lịch rảnh</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>Sinh viên mở chức năng “Đặt lịch hẹn”</li><li>Không chọn thời gian hoặc không nhập nội dung</li><li>Nhấn “Xác nhận”</li></ol>
<b>Inputs</b>	Ngày giờ: 15/11/2025 14:00-15:00; Nội dung: (trống)
<b>Expected Outputs</b>	<ol style="list-style-type: none"><li>Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại</li><li>Quay lại màn hình nhập thông tin</li></ol>
<b>Testing environment</b>	Web – Windows 11



### 3.6.5 Phản hồi chất lượng buổi học

<b>Test case</b>	UC09-TC01 - Gửi phản hồi chất lượng buổi học thành công
<b>Test description</b>	Sinh viên gửi phản hồi hợp lệ cho một buổi học đã diễn ra, hệ thống lưu phản hồi, cập nhật lịch sử
<b>Related screens</b>	- Tab “Phản hồi chất lượng”. - Màn hình “Gửi phản hồi”.
<b>Pre-conditions</b>	1. Sinh viên đã có ít nhất một buổi hẹn đã tham gia. 2. Tab “Phản hồi chất lượng” đang hiển thị và có buổi hẹn mà sinh viên đã tham gia
<b>Actions</b>	1. Sinh viên nhấn nút “Phản hồi chất lượng” của buổi hẹn. 2. Hệ thống mở màn hình “Gửi phản hồi” cho buổi hẹn đó. 3. Sinh viên đánh giá sao 4. Sinh viên nhập nội dung vào ô “Nhận xét”, ví dụ: “Giải bài mẫu chi tiết, dễ hiểu.”. 5. Sinh viên nhấn nút “Gửi”. 6. Hệ thống kiểm tra dữ liệu hợp lệ (đã chọn sao và nhận xét không rỗng). 7. Hệ thống lưu phản hồi vào cơ sở dữ liệu. 8. Hệ thống cập nhật khung “Lịch sử phản hồi” với dòng phản hồi mới. 9. Hệ thống hiển thị thông báo gửi phản hồi thành công.
<b>Inputs</b>	- Mức đánh giá: 5/5 sao. - Nhận xét: “Giải bài mẫu chi tiết, dễ hiểu.”.
<b>Expected Outputs</b>	- Phản hồi mới được lưu đúng sinh viên, đúng buổi hẹn - Khung “Lịch sử phản hồi” hiển thị nội dung đã đánh giá, ví dụ: ngày 2025-10-26, điểm 5/5, nội dung “Giải bài mẫu chi tiết, dễ hiểu.”. - Sinh viên thấy thông báo gửi thành công, không có thông báo lỗi.
<b>Testing environment</b>	Web



<b>Test case</b>	UC09-TC02 - Gửi phản hồi thiếu nội dung nhận xét
<b>Test description</b>	Sinh viên chỉ chọn số sao, bỏ trống ô nhận xét rồi nhấn “Gửi”; hệ thống phát hiện thiếu thông tin, hiển thị thông báo lỗi và không lưu phản hồi.
<b>Related screens</b>	- Tab “Phản hồi chất lượng”. - Màn hình “Gửi phản hồi”.
<b>Pre-conditions</b>	1. Sinh viên đã có ít nhất một buổi hẹn đã tham gia. 2. Tab “Phản hồi chất lượng” đang hiển thị và có buổi hẹn mà sinh viên đã tham gia
<b>Actions</b>	1. Sinh viên nhấn “Phản hồi chất lượng” cho buổi hẹn. 2. Màn hình “Gửi phản hồi” hiển thị. 3. Sinh viên chọn số sao. 4. Sinh viên để trống ô “Nhận xét”. 5. Sinh viên nhấn nút “Gửi”.
<b>Inputs</b>	- Mức đánh giá: 4/5 sao. - Nhận xét: để trống.
<b>Expected Outputs</b>	- Hệ thống không lưu bất kỳ phản hồi nào vào cơ sở dữ liệu. - Hệ thống hiển thị thông báo lỗi, ví dụ: “Vui lòng nhập nội dung phản hồi trước khi gửi.”. - Màn hình “Gửi phản hồi” vẫn giữ nguyên: 4 sao vẫn được chọn, ô “Nhận xét” vẫn trống để sinh viên nhập lại. - Khung “Lịch sử phản hồi” không thay đổi.
<b>Testing environment</b>	Web



<b>Test case</b>	UC09-TC03 - Hủy gửi phản hồi chất lượng
<b>Test description</b>	Sinh viên mở màn hình “Gửi phản hồi” nhưng nhấn “Hủy” thay vì “Gửi”; hệ thống không lưu dữ liệu và quay về danh sách buổi hẹn.
<b>Related screens</b>	- Tab “Phản hồi chất lượng”. - Màn hình “Gửi phản hồi”.
<b>Pre-conditions</b>	1. Sinh viên đã có ít nhất một buổi hẹn đã tham gia. 2. Tab “Phản hồi chất lượng” đang hiển thị và có buổi hẹn mà sinh viên đã tham gia
<b>Actions</b>	1. Sinh viên nhấn “Phản hồi chất lượng” cho buổi hẹn. 2. Màn hình “Gửi phản hồi” hiển thị. 3. Sinh viên có thể chọn số sao và/hoặc nhập vài từ vào ô “Nhận xét” (tuỳ ý). 4. Sinh viên nhấn nút “Hủy” thay vì “Gửi”.
<b>Inputs</b>	Có thể đã nhập một số dữ liệu tạm (ví dụ chọn 3/5 sao, nhận xét ngắn), nhưng chưa thực hiện gửi.
<b>Expected Outputs</b>	- Hệ thống không lưu bất kỳ phản hồi nào vào cơ sở dữ liệu. - Hệ thống đóng màn hình “Gửi phản hồi” và quay lại tab “Phản hồi chất lượng” (danh sách buổi hẹn). - Khi mở lại màn hình “Gửi phản hồi” cho cùng buổi hẹn, khung “Lịch sử phản hồi” không có phản hồi mới.
<b>Testing environment</b>	Web



<b>Test case</b>	UC09-TC04 - Lỗi hệ thống khi lưu phản hồi
<b>Test description</b>	Trong lúc sinh viên gửi phản hồi hợp lệ, hệ thống gặp lỗi cơ sở dữ liệu hoặc lỗi backend; hệ thống báo thất bại, ghi log lỗi và không làm mất dữ liệu đã nhập trên form.
<b>Related screens</b>	- Tab “Phản hồi chất lượng”. - Màn hình “Gửi phản hồi”.
<b>Pre-conditions</b>	1. Tài khoản sinh viên tồn tại và đang hoạt động, đã đăng nhập. 2. Có buổi hẹn đã tham gia và hiển thị trong tab “Phản hồi chất lượng”. 3. Môi trường kiểm thử được cấu hình mô phỏng lỗi khi lưu vào cơ sở dữ liệu.
<b>Actions</b>	1. Sinh viên nhấn “Phản hồi chất lượng” cho buổi hẹn ngày 26/10/2025. 2. Màn hình “Gửi phản hồi” hiển thị. 3. Sinh viên chọn 5 sao. 4. Sinh viên nhập nhận xét, ví dụ: “Buổi học rất hữu ích.”. 5. Sinh viên nhấn nút “Gửi”. 6. Backend cố gắng lưu dữ liệu nhưng gặp lỗi cơ sở dữ liệu.
<b>Inputs</b>	- Mức đánh giá: 5/5 sao. - Nhận xét: “Buổi học rất hữu ích.”.
<b>Expected Outputs</b>	- Hệ thống không tạo bản ghi phản hồi mới trong cơ sở dữ liệu. - Hệ thống hiển thị thông báo lỗi, ví dụ: “Gửi phản hồi thất bại, vui lòng thử lại sau hoặc liên hệ bộ phận kỹ thuật.”. - Dữ liệu trên form không bị mất: 5 sao vẫn được chọn, nội dung nhận xét vẫn còn để sinh viên có thể thử gửi lại. - Lỗi được ghi log ở phía server để phục vụ việc theo dõi và khắc phục.
<b>Testing environment</b>	Web



<b>Test case</b>	UC09-TC05 - Gửi phản hồi không chọn mức sao đánh giá
<b>Test description</b>	Sinh viên chỉ nhập nội dung nhận xét, không chọn sao đánh giá rồi nhấn “Gửi”; hệ thống phát hiện thiếu mức đánh giá, hiển thị thông báo lỗi và không lưu phản hồi.
<b>Related screens</b>	- Tab “Phản hồi chất lượng”. - Màn hình “Gửi phản hồi”.
<b>Pre-conditions</b>	1. Tài khoản sinh viên tồn tại, đang hoạt động và đã đăng nhập. 2. Có ít nhất một buổi hẹn đã tham gia, hiển thị trong tab “Phản hồi chất lượng” với nút “Phản hồi chất lượng”.
<b>Actions</b>	1. Tại tab “Phản hồi chất lượng”, sinh viên nhấn nút “Phản hồi chất lượng” của buổi hẹn. 2. Hệ thống mở màn hình “Gửi phản hồi” cho buổi hẹn đó. 3. Sinh viên không chọn bất kỳ sao nào (mức đánh giá để trống). 4. Sinh viên nhập nhận xét vào ô “Nhận xét”, ví dụ: “Buổi học khá ổn.”. 5. Sinh viên nhấn nút “Gửi”.
<b>Inputs</b>	- Mức đánh giá: không chọn (trống). - Nhận xét: “Buổi học khá ổn.”.
<b>Expected Outputs</b>	- Hệ thống không lưu phản hồi vào cơ sở dữ liệu. - Hệ thống hiển thị thông báo lỗi, ví dụ: “Vui lòng chọn mức đánh giá trước khi gửi.”. - Màn hình “Gửi phản hồi” vẫn giữ lại nội dung đã nhập trong ô “Nhận xét” để sinh viên không phải gõ lại. - Sau lỗi, nếu quay lại tab “Phản hồi chất lượng”, khung “Lịch sử phản hồi” của buổi hẹn không có thêm phản hồi mới.
<b>Testing environment</b>	Web



### 3.6.6 Xem danh sách buổi gặp mặt

<b>Test case</b>	UC-10-TC01 — Xem danh sách buổi gặp mặt
<b>Test description</b>	Sinh viên hoặc tutor xem danh sách các buổi gặp mặt đã đăng ký thành công hoặc đã được duyệt
<b>Related screens</b>	Màn hình Danh sách buổi gặp mặt
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>Sinh viên/Tutor đã đăng nhập</li><li>Nếu là sinh viên thì phải đã đăng ký tutor</li><li>Có ít nhất một buổi gặp đã đăng ký / được tạo</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>Mở chức năng “Danh sách buổi gặp mặt”</li><li>Hệ thống hiển thị danh sách các buổi gặp</li><li>Nhấn vào một buổi để xem chi tiết</li><li>Hệ thống hiển thị thông tin chi tiết buổi đó</li><li>Người dùng nhấn Quay lại / Back</li><li>Hệ thống quay về danh sách buổi gặp mặt</li></ol>
<b>Inputs</b>	Chọn buổi gặp với Chủ đề: Công nghệ phần mềm
<b>Expected Outputs</b>	<ol style="list-style-type: none"><li>Hệ thống hiển thị chi tiết buổi gặp</li><li>Quay lại danh sách, các buổi gặp vẫn hiển thị đúng</li></ol>
<b>Testing environment</b>	Web – Windows 11



### 3.6.7 Hủy buổi gặp mặt

<b>Test case</b>	UC-11-TC01 — Hủy buổi gặp mặt với lý do
<b>Test description</b>	Actor (Sinh viên/Tutor) hủy buổi gặp đã lên lịch và nhập lý do hủy
<b>Related screens</b>	Màn hình hủy đăng ký buổi gặp mặt / hủy buổi gặp mặt
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>1. Actor đã đăng nhập</li><li>2. Buổi gặp chưa bắt đầu</li><li>3. Buổi gặp đã được đăng ký, lên lịch</li></ol>
<b>Actions</b>	<ol style="list-style-type: none"><li>1. Actor vào mục hủy đăng ký buổi gặp mặt / hủy buổi gặp mặt</li><li>2. Hệ thống hiển thị danh sách buổi gặp đã đăng ký / đã lên lịch</li><li>3. Actor chọn một buổi gặp cụ thể</li><li>4. Hệ thống hiển thị form nhập lý do hủy</li><li>5. Actor nhập lý do và bấm Xác nhận</li><li>6. Hệ thống cập nhật slot rảnh của Tutor và trạng thái buổi gặp</li><li>7. Hệ thống gửi thông báo xác nhận hủy:<ol style="list-style-type: none"><li>a. Nếu actor là Sinh viên → gửi thông báo cho Tutor</li><li>b. Nếu actor là Tutor → gửi thông báo cho Sinh viên và Tutor</li></ol></li></ol>
<b>Inputs</b>	Actor: Sinh viên/Tutor, Buổi gặp: 15/11/2025 14:00-15:00 , Lý do hủy: “Bận đột xuất”
<b>Expected Outputs</b>	<ol style="list-style-type: none"><li>1. Hệ thống hiển thị “Hủy thành công”</li><li>2. Cập nhật trạng thái slot:<ul style="list-style-type: none"><li>• Nếu actor là Sinh viên → mở slot Tutor, buổi gặp bị hủy cho sinh viên</li><li>• Nếu actor là Tutor → buổi gặp bị hủy cho cả Tutor và Sinh viên</li></ul></li><li>3. Gửi thông báo hủy cho actor liên quan</li></ol>
<b>Testing environment</b>	Web – Windows 11



### 3.6.8 Xử lý yêu cầu đặt lịch hẹn

<b>Test case</b>	UC-18-TC01 - Tutor đồng ý lịch hẹn
<b>Test description</b>	Tutor chấp nhận (đồng ý) một yêu cầu đặt lịch hẹn đang chờ xử lý từ sinh viên.
<b>Related screens</b>	<ul style="list-style-type: none"><li>- Giao diện danh sách yêu cầu lịch hẹn</li><li>- Giao diện xem chi tiết lịch hẹn</li></ul>
<b>Pre-conditions</b>	<ul style="list-style-type: none"><li>- Tutor đã đăng nhập vào hệ thống</li><li>- Giao diện danh sách lịch hẹn đang hiển thị</li><li>- Tồn tại ít nhất một lịch hẹn với trạng thái đang chờ xử lý</li></ul>
<b>Actions</b>	<ul style="list-style-type: none"><li>- Từ giao diện danh sách yêu cầu lịch hẹn, tutor chọn một lịch hẹn để xem chi tiết</li><li>- Hệ thống hiển thị thông tin chi tiết của lịch hẹn bao gồm họ tên sinh viên, nội dung, thời gian, và hình thức</li><li>- Tutor nhấn nút "Phê duyệt"</li><li>- Hệ thống xử lý yêu cầu và hiển thị thông báo "Yêu cầu đã được phê duyệt"</li><li>- Tutor chọn "Quay lại" hệ thống quay lại giao diện danh sách lịch hẹn</li></ul>
<b>Inputs</b>	<ul style="list-style-type: none"><li>- Danh sách yêu cầu lịch hẹn</li></ul>
<b>Expected Outputs</b>	Hệ thống xác nhận "Yêu cầu đã được phê duyệt" và gửi thông báo đến sinh viên đó
<b>Testing environment</b>	Giao diện web chạy trên trình duyệt, và vai trò người dùng là Tutor



<b>Test case</b>	UC-18-TC02 - Tutor từ chối lịch hẹn
<b>Test description</b>	Tutor từ chối một yêu cầu đặt lịch hẹn đang chờ xử lý từ sinh viên.
<b>Related screens</b>	<ul style="list-style-type: none"><li>- Giao diện danh sách yêu cầu lịch hẹn</li><li>- Giao diện xem chi tiết lịch hẹn</li><li>- Pop-up "Lý do từ chối"</li></ul>
<b>Pre-conditions</b>	<ul style="list-style-type: none"><li>- Tutor đã đăng nhập vào hệ thống</li><li>- Giao diện danh sách lịch hẹn đang hiển thị</li><li>- Tồn tại ít nhất một lịch hẹn với trạng thái đang chờ xử lý</li></ul>
<b>Actions</b>	<ul style="list-style-type: none"><li>- Từ giao diện danh sách yêu cầu lịch hẹn, tutor chọn một lịch hẹn để xem chi tiết</li><li>- Hệ thống hiển thị thông tin chi tiết của lịch hẹn bao gồm họ tên sinh viên, nội dung, thời gian, và hình thức</li><li>- Tutor nhấn nút "Từ chối"</li><li>- Hệ thống hiển thị Pop-up "Lý do từ chối", yêu cầu nhập lý do.</li><li>- Tutor nhập lý do vào ô text và chọn "Xác nhận từ chối"</li><li>- Hệ thống xử lý yêu cầu và thông báo "Đã từ chối lịch hẹn"</li><li>- Tutor chọn "Quay lại" hệ thống quay lại giao diện danh sách lịch hẹn</li></ul>
<b>Inputs</b>	<ul style="list-style-type: none"><li>- Danh sách yêu cầu lịch hẹn</li></ul>
<b>Expected Outputs</b>	Hệ thống xác nhận "Đã từ chối lịch hẹn" và gửi thông báo đến sinh viên đó
<b>Testing environment</b>	Giao diện web chạy trên trình duyệt, và vai trò người dùng là Tutor

## 4 Task 4

### 4.1 Hiện thực

Link hiện thực của dự án [tại đây](#).

Trong giai đoạn hiện thực hoá hệ thống, nhóm lựa chọn kết hợp hai công nghệ chính: **Java Spring Boot** cho phía backend và **ReactJS** cho phía frontend. Kiến trúc tổng thể của hệ thống được triển khai theo mô hình MVC kết hợp với layered architecture, giúp tách biệt giao diện, lớp điều khiển, nghiệp vụ và truy cập dữ liệu. Việc sử dụng các công nghệ hiện đại này giúp hệ thống phát triển theo hướng mở rộng, dễ bảo trì và phù hợp với yêu cầu xây dựng RESTful API của dự án.



#### 4.1.1 Spring Boot (Backend)

Spring Boot được nhóm lựa chọn vì khả năng hỗ trợ tạo ứng dụng web nhanh chóng, cấu hình tối giản và tích hợp tốt với các thư viện trong hệ sinh thái Spring.

##### **Ưu điểm:**

- Hỗ trợ mạnh mẽ cho mô hình phát triển kiến trúc theo tầng: controller–service–repository.
- Tự động cấu hình, giảm đáng kể thời gian setup môi trường.
- Tích hợp Spring Security và khả năng kết nối REST API tốt.
- Dễ mở rộng và dễ kiểm thử nhờ cấu trúc module rõ ràng.

##### **Nhược điểm:**

- Chi phí khởi động ban đầu cao hơn so với một số framework nhẹ (như NodeJS).
- Khó khăn cho người mới bắt đầu.
- Khi dự án lớn dần, cấu trúc Spring có thể phức tạp nếu không quản lý tốt.

#### 4.1.2 ReactJS (Frontend)

ReactJS được nhóm sử dụng để xây dựng giao diện người dùng. Việc chia nhỏ giao diện thành các component giúp hệ thống dễ phát triển, tái sử dụng mã nguồn và đồng thời tối ưu trải nghiệm người dùng. Kết hợp React với REST API từ Spring Boot cho phép hệ thống hoạt động linh hoạt và tách biệt frontend–backend rõ ràng.

##### **Ưu điểm:**

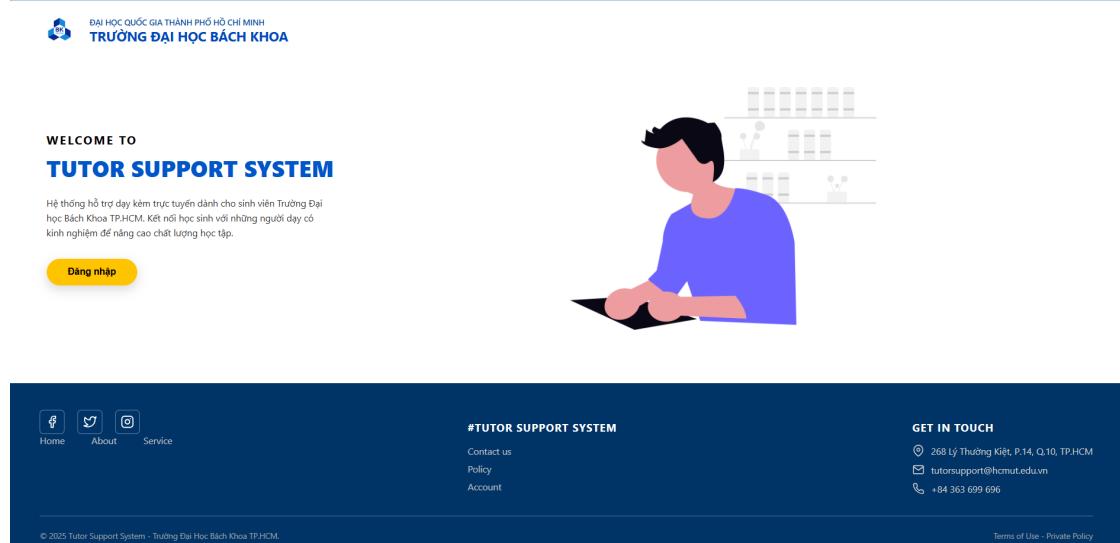
- Mô hình component giúp tái sử dụng và bảo trì giao diện dễ dàng.
- Nhiều thư viện hỗ trợ (Axios, React Router, v.v.)

##### **Nhược điểm:**

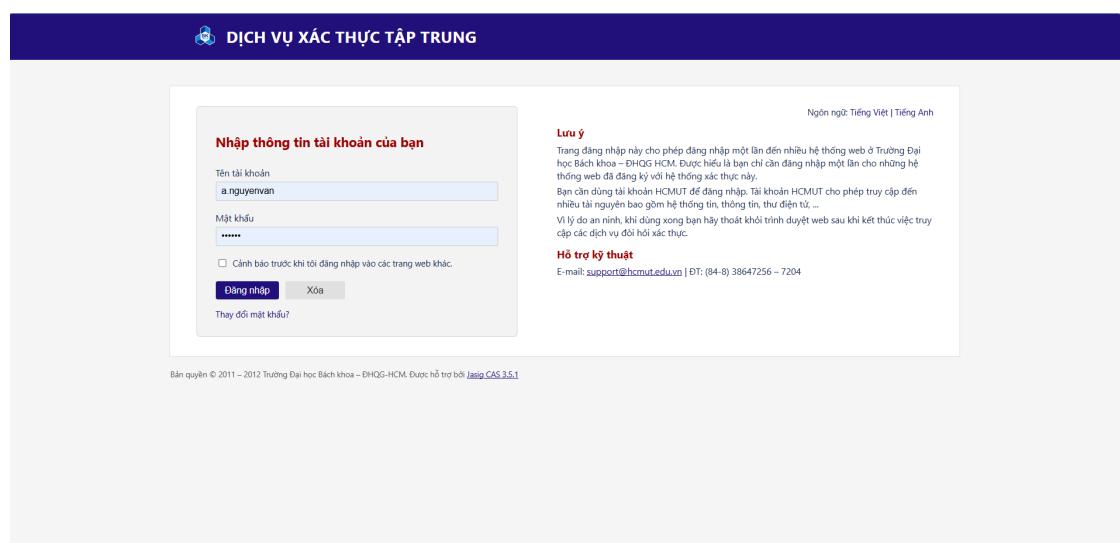
- Khá phụ thuộc vào các thư viện bên ngoài; dễ gây lỗi giao diện nếu thiếu thống nhất trong việc thiết kế từng thành phần.
- Khó khăn cho người mới bắt đầu.



## 4.2 Luồng hoạt động của hệ thống



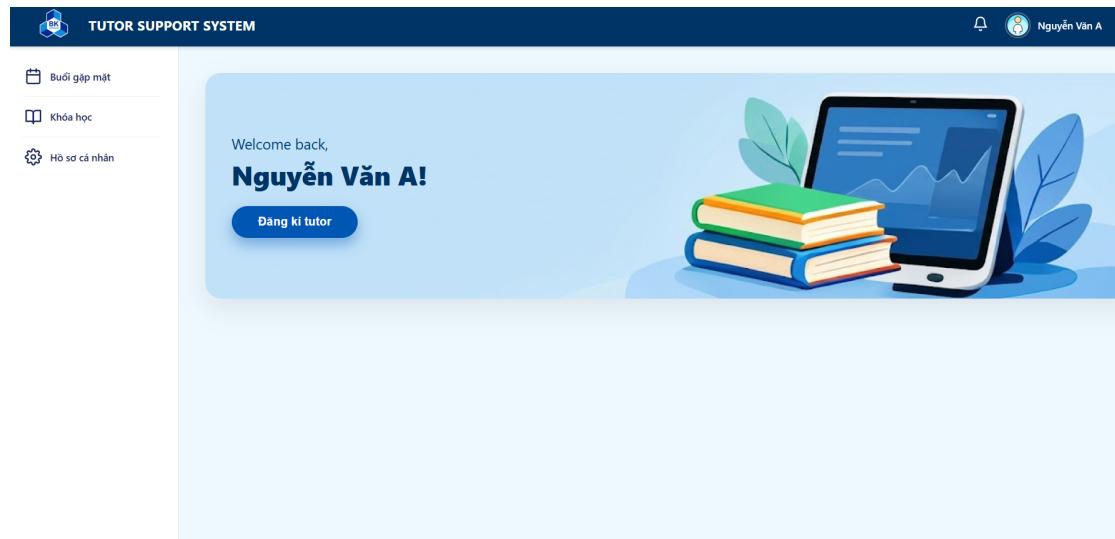
Hình 60: Giao diện chính của hệ thống



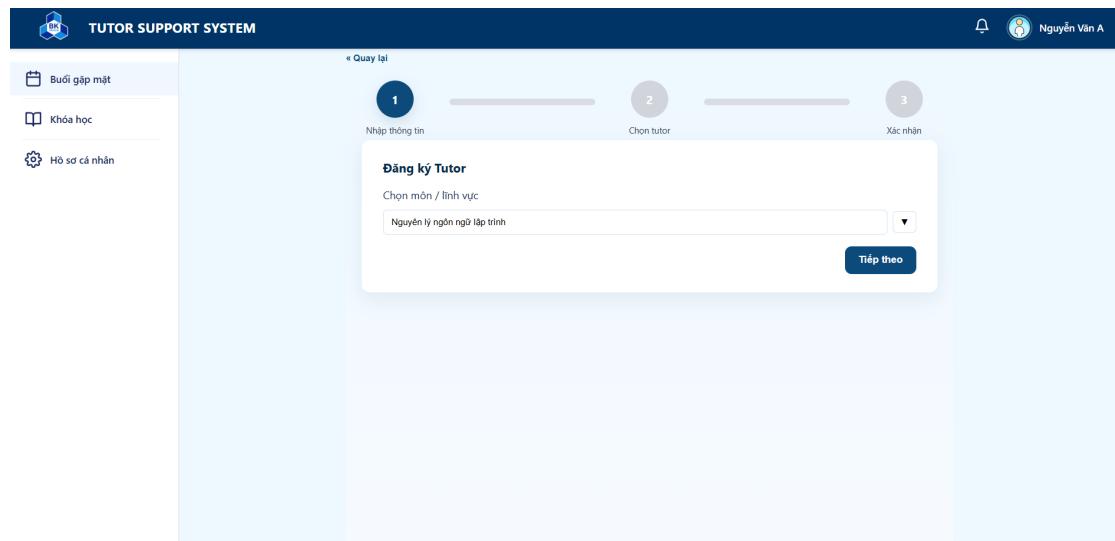
Hình 61: Giao diện đăng nhập qua HCMUT\_SSO



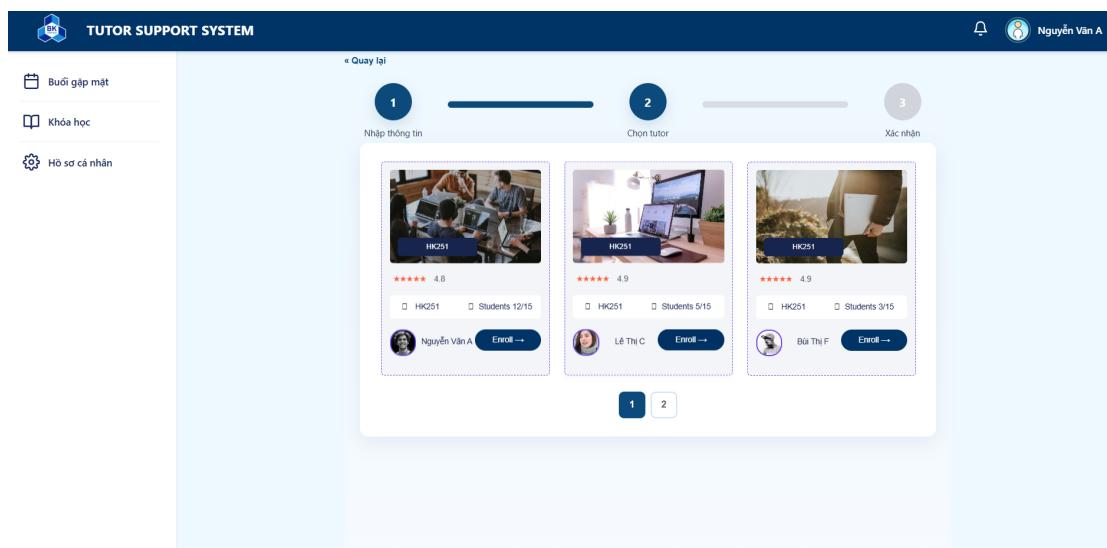
#### 4.2.1 Student



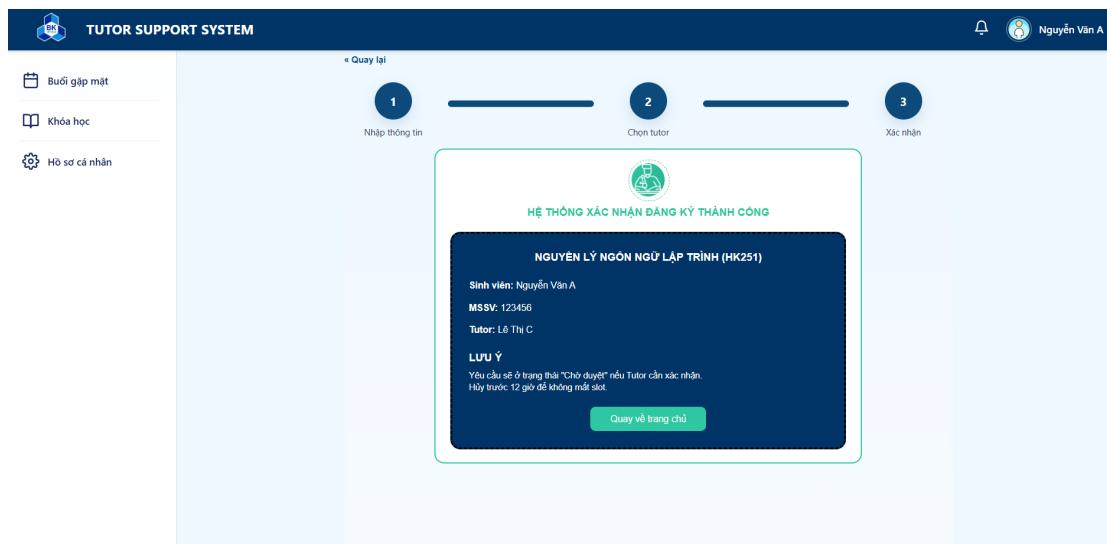
Hình 62: Trang chủ student



Hình 63: Đăng ký tutor - Chọn môn học



Hình 64: Đăng ký tutor - Chọn tutor



Hình 65: Đăng ký tutor thành công



The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. On the left sidebar, there are three main menu items: 'Buổi gặp mặt' (selected), 'Khóa học', and 'Hồ sơ cá nhân'. The main content area is titled 'Danh sách buổi gặp mặt' and contains three meeting entries:

- Chủ đề: Nguyễn lý ngôn ngữ lập trình**  
Thời gian: 08:00 - 10:00, ngày 30/11/2025  
Online: <https://meet.google.com/sample-link>  
Buttons: 'BUỔI HẸN' (green), 'Đã diễn ra' (orange)
- Chủ đề: Lập trình Web - ReactJS cơ bản**  
Thời gian: 14:00 - 16:00, ngày 1/12/2025  
Online: <https://meet.google.com/sample-link>  
Buttons: 'BUỔI HỘI THẢO' (green), 'Đã diễn ra' (orange)
- Chủ đề: Hội thảo AI trong giáo dục Đại học hiện nay**  
Thời gian: 09:30 - 11:30, ngày 12/12/2025  
Online: <https://meet.google.com/sample-link>  
Buttons: 'BUỔI HẸN' (green), 'Sắp diễn ra' (blue)

A red 'Hủy đăng ký' (Cancel registration) button is located at the top right of the list.

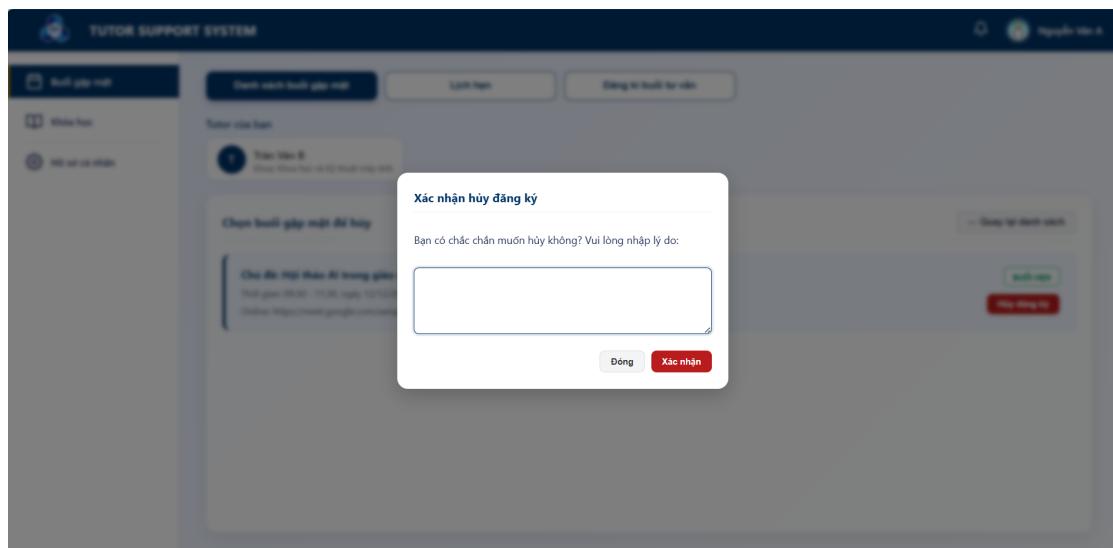
Hình 66: Danh sách buổi gặp mặt

This screenshot shows the same 'TUTOR SUPPORT SYSTEM' interface as above, but it has been expanded to show a modal dialog box. The dialog is titled 'Chọn buổi gặp mặt để hủy' (Select meeting to cancel) and displays the same third meeting entry from the list:

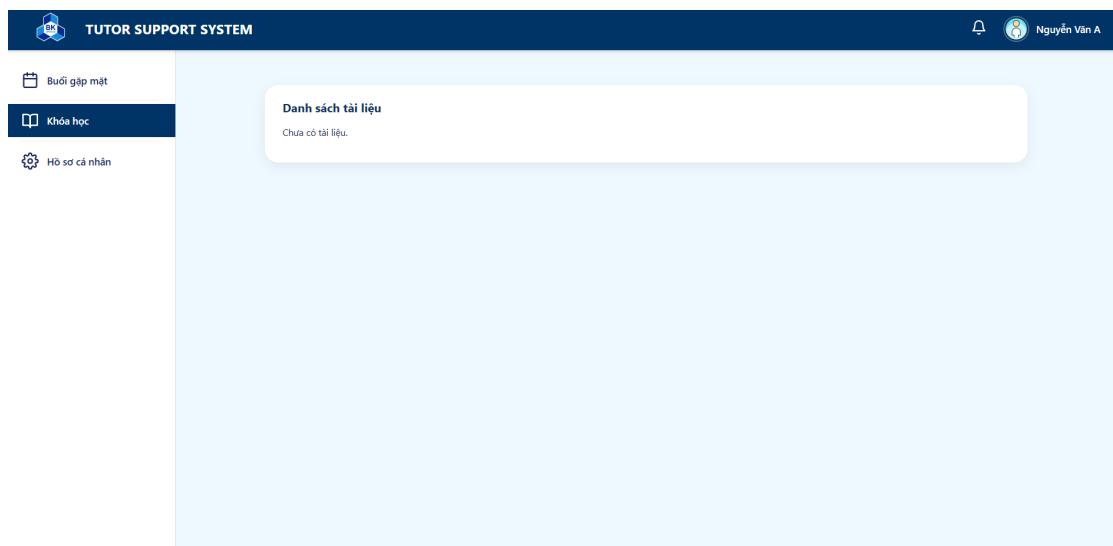
**Chủ đề: Hội thảo AI trong giáo dục Đại học hiện nay**  
Thời gian: 09:30 - 11:30, ngày 12/12/2025  
Online: <https://meet.google.com/sample-link>

At the bottom right of the dialog, there are two buttons: a green 'BUỔI HẸN' button and a red 'Hủy đăng ký' button.

Hình 67: Hủy buổi gặp mặt



Hình 68: Xác nhận hủy buổi gặp mặt



Hình 69: Danh sách tài liệu



The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. On the left sidebar, there are three buttons: 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), and 'Hồ sơ cá nhân' (Personal Profile). The main area is titled 'Đặt lịch hẹn' (Book Appointment) and displays a calendar for December 2025. A specific date, December 2nd, is highlighted. To the right of the calendar, there are fields for 'Ngày đã chọn' (Selected Date: 02-Dec-2025), 'Khung giờ ránh của tutor' (Tutor's availability: 07:00 - 12:45), 'Giờ hẹn bạn mong muốn' (Preferred appointment time: 07:00 AM - 09:00 AM), and 'Nội dung' (Content: Hỗ trợ BTL CNPM). At the bottom is a large blue button labeled 'Gửi yêu cầu' (Send Request).

Hình 70: Đặt lịch hẹn với tutor

The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. The sidebar has the same three buttons: 'Buổi gặp mặt', 'Khóa học', and 'Hồ sơ cá nhân'. The main area displays a profile for 'Nguyễn Văn A' (student). It includes a placeholder profile picture, the name 'Nguyễn Văn A' (student), and a 'Thêm ảnh' (Add Photo) button. Below this, there is a table with student information: Họ và tên (Name: Nguyễn Văn A), Mã sinh viên (Student ID: 123456), Khoa/Bộ môn (Faculty: Khoa học và Kỹ thuật máy tính), Chuyên ngành (Major: Khoa học máy tính), Email (Email: a.nguyenvan@hcmut.edu.vn), and Số điện thoại (Phone: 0999 999 999). At the bottom, there is a section titled 'Lịch sử đăng kí' (Registration History) showing two entries: Ngày (Date: 20-09-2025), Môn (Subject: Toán rời rạc), Tutor (Tutor: Nguyễn Thị B), and Trạng thái (Status: Hoàn tất); and Ngày (Date: 19-09-2025), Môn (Subject: Giải tích), Tutor (Tutor: Trần Văn C), and Trạng thái (Status: Hủy).

Hình 71: Hồ sơ cá nhân



The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. On the left sidebar, there are four main categories: 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), and 'Hồ sơ cá nhân' (Personal resume). The 'Khóa học' category is currently selected and highlighted in blue. In the main content area, there is a section titled 'Buổi hẹn đã tham gia' (Scheduled meetings). It lists three entries:

- Chủ đề: Học Reactjs**  
Thời gian: 07:00 - 09:00, ngày 1/12/2025  
Buttons: ✓ BUỔI HẸN and Phản hồi chất lượng
- Chủ đề: Báo cáo bài tập lớn**  
Thời gian: 07:00 - 09:00, ngày 2/12/2025  
Buttons: ✓ BUỔI HẸN and Phản hồi chất lượng
- Chủ đề: Hướng dẫn đồ án tốt nghiệp**  
Thời gian: 07:30 - 08:00, ngày 3/12/2025  
Buttons: ✓ BUỔI HẸN and Phản hồi chất lượng

Hình 72: Phản hồi

The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. The 'Khóa học' category is selected. In the main content area, there is a feedback form for a meeting:

**Nội dung**  
Học Reactjs

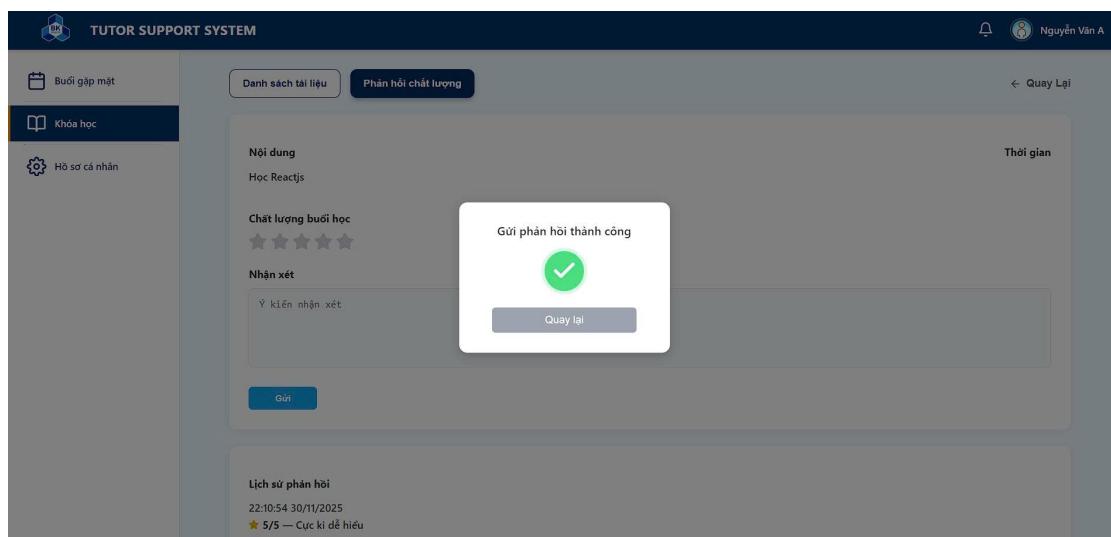
**Chất lượng buổi học**

**Nhận xét**  
Buổi học rất bổ ích

**Gửi**

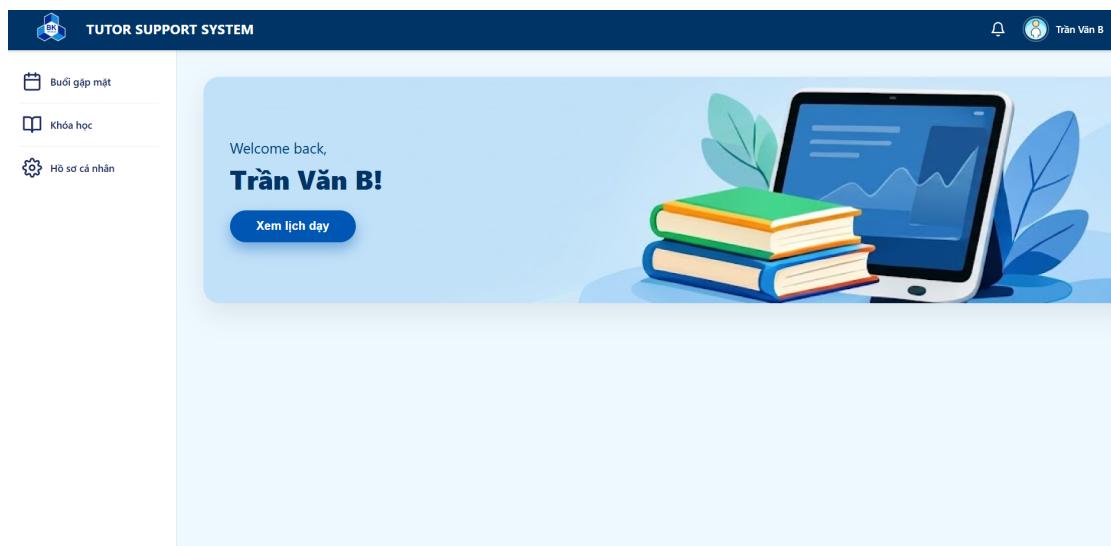
**Lịch sử phản hồi**  
22:09:28 30/11/2025  
★ 5/5 — Rất hay và thú vị

Hình 73: Phản hồi



Hình 74: Phản hồi

#### 4.2.2 Tutor



Hình 75: Trang chủ tutor



The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. At the top right, there is a user profile for 'Trần Văn B' with a notification icon. The left sidebar has three main menu items: 'Buổi gặp mặt' (selected), 'Khóa học', and 'Hồ sơ cá nhân'. The main content area is titled 'Buổi gặp mặt của tôi' (My meeting). It contains a message: 'Chưa có buổi gặp mặt nào.' (No meeting scheduled).

Hình 76: Danh sách buổi gặp mặt

The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. At the top right, there is a user profile for 'Trần Văn B' with a notification icon. The left sidebar has three main menu items: 'Buổi gặp mặt' (selected), 'Khóa học', and 'Hồ sơ cá nhân'. The main content area is titled 'Danh sách tài liệu' (List of documents). It contains a message: 'Chưa có tài liệu.' (No documents).

Hình 77: Danh sách tài liệu



The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. On the left sidebar, there are three menu items: 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), and 'Hồ sơ cá nhân' (Personal Profile). The main area features a search bar labeled 'Thông tin cá nhân' (Personal information) and a 'Lịch rảnh' (Free schedule) button. Below these are two buttons: '<< 2025-11-30 >>' and 'Chưa có lịch' (No schedule). A large calendar grid for November 2025 is displayed, with days from T2 to CN. The date '30' is highlighted in blue, indicating it is selected or the current date. A small button 'Thiết lập lịch rảnh' (Set free schedule) is located in the top right corner of the calendar area.

Hình 78: Thiết lập lịch rảnh - Chọn ngày

This screenshot shows the 'Thiết lập lịch rảnh' (Set free schedule) dialog box. It includes a message 'localhost:3000 says Đã lưu lịch thành công!' (Schedule saved successfully!) with an 'OK' button. The dialog has a 'Thứ' (Day) dropdown set to 'Thứ 7' (Saturday). Below it, a 'Ngày:' (Date:) input field shows '02-Jan-2026'. A time selection section with 'Bắt đầu' (Start) at 07:00 and 'Kết thúc' (End) at 09:00 is shown, along with a '+ Thêm thời gian' (Add more time) button. A red 'X' button is also present. The background of the dialog shows the 'Lịch rảnh' (Free schedule) section of the system.

Hình 79: Thiết lập lịch rảnh - Chọn giờ

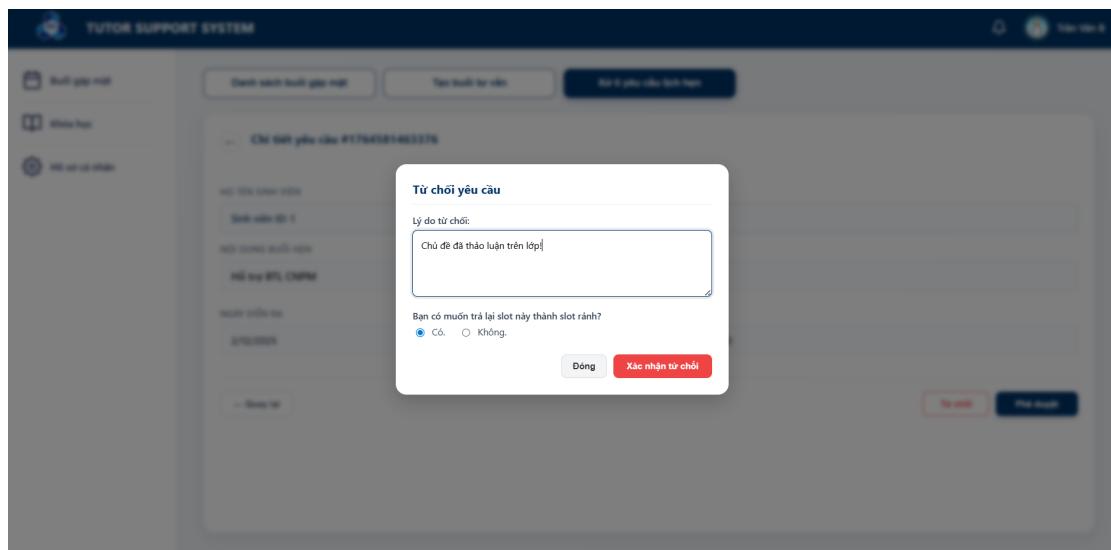


The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. On the left sidebar, there are three main menu items: 'Buổi gặp mặt' (Meeting), 'Khóa học' (Course), and 'Hồ sơ cá nhân' (Personal Profile). The 'Xử lý yêu cầu lịch hẹn' (Handle appointment requests) button is highlighted in blue. The main content area displays a list titled 'Danh sách yêu cầu chờ duyệt' (List of pending requests). A single item is listed: 'Hỗ trợ BTL CNPM' (Support for CNPM Head) with the date 'Ngày: 2/12/2025' and time 'Khung giờ: 07:00 - 09:00'. To the right of the list item are two buttons: 'PENDING' (orange) and 'Xem chi tiết' (blue).

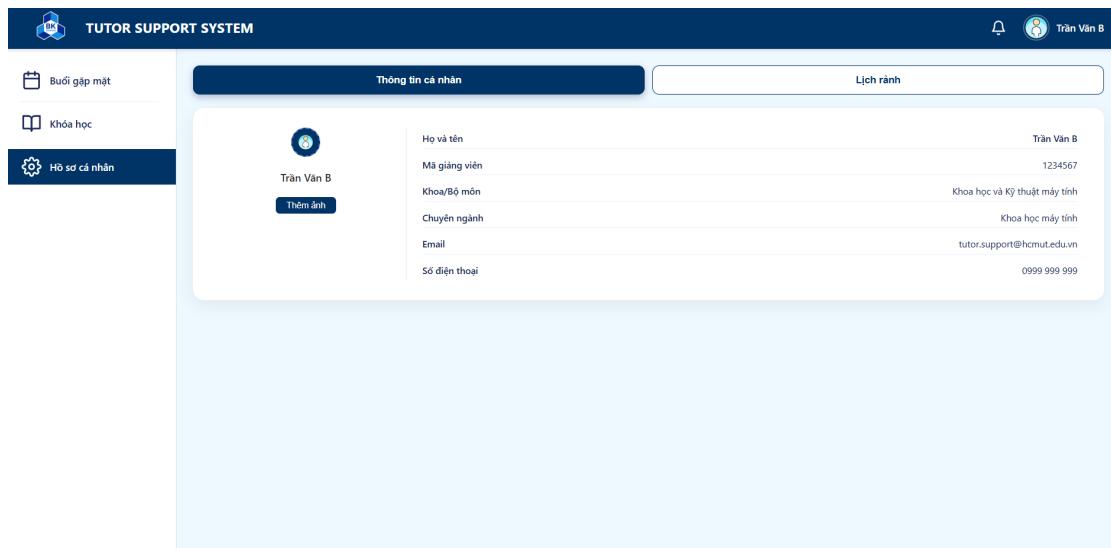
Hình 80: Xử lý yêu cầu đặt lịch hẹn - Danh sách lịch hẹn

The screenshot shows the 'TUTOR SUPPORT SYSTEM' interface. The left sidebar has the same three menu items: 'Buổi gặp mặt', 'Khóa học', and 'Hồ sơ cá nhân'. The 'Xử lý yêu cầu lịch hẹn' button is also highlighted in blue. The main content area shows a detailed view of a specific request under 'Chi tiết yêu cầu #1764581463376'. The details include the student's name ('Họ Tên Sinh Viên') as 'Sinh viên ID: 1', the purpose ('Nội dung buổi hẹn') as 'Hỗ trợ BTL CNPM', the date ('Ngày diễn ra') as '2/12/2025', and the time ('Khung giờ') as '07:00 - 09:00'. At the bottom of the detail view are two buttons: 'Từ chối' (Reject) in red and 'Phê duyệt' (Approve) in blue.

Hình 81: Xử lý yêu cầu đặt lịch hẹn - Thông tin chi tiết



Hình 82: Xử lý yêu cầu đặt lịch hẹn - Xác nhận từ chối yêu cầu



Hình 83: Hồ sơ cá nhân

### 4.3 Về việc sử dụng AI

Trong quá trình làm bài tập lớn, nhóm có sử dụng các công cụ AI tạo sinh (Generative AI) nhằm hỗ trợ phân tích yêu cầu, phát hiện lỗi trong thiết kế và đề xuất giải pháp kỹ thuật.

Đầu tiên, trong giai đoạn phân tích yêu cầu, AI được sử dụng để đổi chiều lại các yêu cầu chức năng và phi chức năng mà nhóm đã xác định. Công cụ AI hỗ trợ chỉ ra các điểm bất hợp lý



hoặc thiếu sót trong quá trình diễn giải yêu cầu so với mô tả đề bài. Nhờ đó, nhóm có thể điều chỉnh lại use-case, luồng hoạt động và các mối quan hệ giữa các tác nhân, giúp mô hình yêu cầu trở nên chặt chẽ và chính xác hơn.

Tiếp theo, trong quá trình thiết kế kiến trúc, AI được dùng để gợi ý cách tổ chức mô hình MVC theo tầng (layered architecture), làm rõ vai trò của từng lớp Controller, Service, Repository và Entity. Công cụ AI hỗ trợ đánh giá mức độ hợp lý của Class Diagram, gợi ý tách lớp, chuẩn hóa tên phương thức và điều chỉnh quan hệ giữa các lớp sao cho phù hợp với mô hình hướng đối tượng. Những gợi ý này giúp nhóm phát hiện sớm các mâu thuẫn thiết kế và chỉnh sửa sơ đồ UML kịp thời trước khi bước vào giai đoạn hiện thực hoá.

Trong giai đoạn triển khai mã nguồn, AI được sử dụng ở mức độ tham khảo, chủ yếu để xem ví dụ về cấu trúc REST Controller trong Spring Boot, cách viết tầng Service theo chuẩn DI (Dependency Injection), mẫu component và hook trong React. Những đoạn mã mẫu do AI cung cấp chỉ mang tính định hướng, nhóm sẽ triển khai lại theo phong cách lập trình và cấu trúc dự án của riêng mình.

## 5 Nhật ký thay đổi

- Cập nhật mô tả các use case cho phù hợp hơn với chức năng thực tế của hệ thống, đồng bộ với yêu cầu nghiệp vụ sau khi rà soát lại.
- Vẽ lại Use Case Diagram nhằm thể hiện đầy đủ các actor, luồng tương tác và phạm vi chức năng của hệ thống sau khi đã hoàn thiện phân tích yêu cầu.
- Cập nhật Mockup.
- Điều chỉnh Class Diagram theo kiến trúc MVC với 5 lớp (View, Controller, Service, Repository, Entity), thay đổi mô tả các phương thức trong Class Diagram cho khớp với phần hiện thực hoá bằng Spring Boot.
- Cập nhật lại Sequence Diagram để thể hiện đúng luồng xử lý từ giao diện React đến API Spring Boot.