

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



## CÔNG NGHỆ PHẦN MỀM (CO3001)

---

### BÀI TẬP LỚN

*Hệ thống hỗ trợ Tutor tại  
Trường Đại học Bách khoa  
Đại học Quốc gia - TP.HCM*

---

**Giáo viên hướng dẫn:** Phan Trung Hiếu, CSE-HCMUT

**Sinh viên - Nhóm 3:** Nguyễn Tấn Phát - 2352888 (CN01)  
Vũ Hà Như Ngọc - 2352818 (CN01)  
Lê Diệu Quỳnh - 2353036 (CN01)  
Mã Nhật Tiến - 2353178 (CN01)  
Bùi Phan Khánh Duy - 2352170 (CN01)  
Lương Đức Huy - 2352384 (CN01)  
Nguyễn Ngọc Phát - 2352887 (CN01)  
Văn Bá Trọng Khiêm - 2352546 (CN01)

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 9 NĂM 2025



## Mục lục

Danh sách Ký hiệu	3
Danh sách Từ viết tắt	3
Danh sách Hình ảnh	5
Danh sách Bảng	5
Danh sách thành viên & khối lượng công việc	5
<b>1. Tổng quan dự án</b>	<b>6</b>
<b>1.1. Giới thiệu dự án</b>	<b>6</b>
1.1.1. Mục đích . . . . .	6
1.1.2. Bối cảnh và lý do cần hệ thống . . . . .	6
1.1.3. Kỳ vọng và mục tiêu . . . . .	7
1.1.4. Sản phẩm bàn giao . . . . .	7
<b>1.2. Phạm vi dự án</b>	<b>8</b>
1.2.1. Trong phạm vi . . . . .	8
1.2.2. Ngoài phạm vi . . . . .	8
1.2.3. Ràng buộc hệ thống . . . . .	9
<b>1.3. Tài liệu tham khảo liên quan</b>	<b>9</b>
1.3.1. Các nền tảng tương tự . . . . .	10
1.3.2. Bài học rút ra . . . . .	10
1.3.3. Tính năng tổng hợp cho hệ thống . . . . .	11
<b>2. Phân tích yêu cầu dự án</b>	<b>12</b>
<b>2.1. Tác nhân</b>	<b>12</b>
2.1.1. Tác nhân chính . . . . .	12
2.1.1.1. Tutor . . . . .	12
2.1.1.2. Sinh viên . . . . .	13
2.1.2. Tác nhân phụ . . . . .	15
2.1.2.1. Khoa/Bộ môn . . . . .	15
2.1.2.2. Phòng Công tác sinh viên . . . . .	16
2.1.2.3. Phòng Đào tạo . . . . .	16
<b>2.2. Sơ đồ usecase toàn hệ thống</b>	<b>18</b>
<b>2.3. User Stories</b>	<b>19</b>
2.3.1. Module Quản lý Tài khoản và Hồ sơ . . . . .	19
2.3.2. Module Đăng ký chương trình học . . . . .	19
2.3.3. Module Ghép cặp Tutor – SV . . . . .	19
2.3.4. Module Quản lý lịch học . . . . .	19
2.3.5. Module Thông báo và nhắc nhở . . . . .	20



2.3.6. Module Quản lý buổi học và điểm danh . . . . .	20
2.3.7. Module Quản lý tài liệu học tập . . . . .	20
2.3.8. Module Đánh giá và phản hồi . . . . .	20
2.3.9. Module Thống kê và báo cáo . . . . .	20
2.3.9. Module Thống kê và báo cáo . . . . .	21
<b>2.4. Yêu cầu chức năng</b>	<b>22</b>
2.4.1. Module Quản lý Tài khoản và Hồ sơ . . . . .	22
2.4.2. Module Đăng ký chương trình học . . . . .	25
2.4.3. Module Ghép cặp Tutor – SV . . . . .	27
2.4.4. Module Quản lý lịch học . . . . .	31
2.4.5. Module Thông báo và nhắc nhở . . . . .	34
2.4.6. Module Quản lý buổi học và điểm danh . . . . .	36
2.4.7. Module Quản lý tài liệu học tập . . . . .	38
2.4.8. Module Đánh giá và phản hồi . . . . .	40
2.4.9. Module Thống kê và báo cáo . . . . .	43
2.4.10. Module Chương trình học thuật và phi học thuật . . . . .	46
<b>2.5. Yêu cầu phi chức năng</b>	<b>48</b>
2.5.1. Hiệu năng (Performance Requirements) . . . . .	48
2.5.2. Hiệu năng (Performance Requirements) . . . . .	49
2.5.3. Tính tin cậy & sẵn sàng (Reliability & Availability) . . . . .	49
2.5.4. Khả năng sử dụng (Usability) . . . . .	49
2.5.5. Tính bảo trì & mở rộng (Maintainability & Extensibility) . . . . .	50
2.5.6. Khả năng tương thích (Compatibility) . . . . .	50
2.5.7. Ràng buộc kỹ thuật (Technical Constraints) . . . . .	50



## Danh sách Ký hiệu

## Danh sách Từ viết tắt

Bảng 1: Danh sách Từ viết tắt

STT	Chữ viết tắt	Chữ viết đầy đủ
1	AI	Trí tuệ nhân tạo (Artificial Intelligence)
2	APP	Application (Ứng dụng)
3	BM	Bộ môn
4	DB	Database (Cơ sở dữ liệu)
5	DOCX	Microsoft Word Document (Tài liệu Microsoft Word)
6	DRP	Disaster Recovery Plan (Kế hoạch khôi phục sau thảm họa)
7	Excel	Microsoft Excel (Phần mềm bảng tính)
8	GPA	Điểm trung bình học tập (Grade Point Average)
9	HCMUT_DATACORE	Lỗi dữ liệu Đại học Bách khoa
10	HCMUT_LIBRARY	Thư viện Đại học Bách khoa
11	HTTPS	Hypertext Transfer Protocol Secure (Giao thức truyền tải siêu văn bản an toàn)
12	ID	Identifier (Mã định danh)
13	iOS	iPhone Operating System (Hệ điều hành của iPhone)
14	MB	Megabyte
15	MSSV	Mã số sinh viên
16	MVC	Model-View-Controller (Một mẫu kiến trúc phần mềm)
17	NCS	Nghiên cứu sinh
18	OTP	One-Time Password (Mật khẩu dùng một lần)
19	PCTSV	Phòng Công tác Sinh viên
20	PDF	Portable Document Format (Định dạng tài liệu di động)
21	PDT	Phòng Đào tạo
22	PPT	PowerPoint Presentation (Bài trình chiếu PowerPoint)
23	SDT	Số điện thoại
24	SMS	Short Message Service (Dịch vụ tin nhắn ngắn)
25	SV	Sinh viên
26	TLS	Transport Layer Security (Bảo mật tầng truyền tải)
27	UC	Use Case (Ca sử dụng)
28	US	User Story (Câu chuyện người dùng)
29	2FA	Two-Factor Authentication (Xác thực hai yếu tố)



## Danh sách Hình ảnh

1	Sơ đồ Usecase toàn hệ thống . . . . .	18
---	---------------------------------------	----

## Danh sách Bảng

1	Danh sách Từ viết tắt . . . . .	3
2	Danh sách thành viên & khối lượng công việc . . . . .	5
3	Bảng đặc tả chức năng đăng ký tài khoản . . . . .	22
4	Bảng đặc tả chức năng đăng nhập . . . . .	23
5	Bảng đặc tả chức năng cập nhật hồ sơ . . . . .	24
6	Bảng đặc tả chức năng đăng ký môn học . . . . .	25
7	Bảng đặc tả chức năng hủy đăng ký môn học . . . . .	26
8	Bảng đặc tả chức năng ghép thủ công (SV chọn Tutor cho từng môn học) . . . . .	28
9	Bảng đặc tả chức năng ghép tự động (hệ thống đề xuất Tutor theo từng môn học) . . . . .	30
10	Bảng đặc tả chức năng tạo lịch rảnh (Tutor) . . . . .	31
11	Bảng đặc tả chức năng đặt lịch học (SV) . . . . .	32
12	Bảng đặc tả chức năng hủy/đổi lịch học cố định . . . . .	33
13	Bảng đặc tả chức năng gửi thông báo lịch học . . . . .	34
14	Bảng đặc tả chức năng gửi nhắc nhở buổi học . . . . .	35
15	Bảng đặc tả chức năng điểm danh sinh viên . . . . .	36
16	Bảng đặc tả chức năng cập nhật trạng thái buổi học . . . . .	37
17	Bảng đặc tả chức năng quản lý tài liệu (Tutor) . . . . .	38
18	Bảng đặc tả chức năng sinh viên tải tài liệu . . . . .	39
19	Bảng đặc tả chức năng sinh viên đánh giá Tutor . . . . .	40
20	Bảng đặc tả chức năng Tutor đánh giá sinh viên . . . . .	41
21	Bảng đặc tả chức năng Khoa/BM tổng hợp đánh giá . . . . .	42
22	Bảng đặc tả chức năng báo cáo kết quả học tập sinh viên . . . . .	43
23	Bảng đặc tả chức năng báo cáo chất lượng Tutor . . . . .	44
24	Bảng đặc tả chức năng báo cáo tổng hợp (Khoa, PCTSV, PDT) . . . . .	45
25	Bảng đặc tả chức năng Tutor tạo chương trình học . . . . .	46
26	Bảng đặc tả chức năng SV đăng ký chương trình học thuật . . . . .	47
27	Bảng đặc tả chức năng SV đăng ký chương trình phi học thuật . . . . .	48



## Danh sách thành viên & khối lượng công việc

STT	Họ Tên	MSSV	Vai trò	% Hoàn thành
1	Nguyễn Tấn Phát	2352888	Scrum Master	100%
2	Vũ Hà Như Ngọc	2352818	Business Analyst	100%
3	Lê Diệu Quỳnh	2353036	Product Owner	100%
4	Mã Nhật Tiến	2353178	Quality Controller	100%
5	Bùi Phan Khánh Duy	2352170	Techlead + SysAdmin	100%
6	Lương Đức Huy	2352384	Developer	100%
7	Nguyễn Ngọc Phát	2352887	Developer	100%
8	Văn Bá Trọng Khiêm	2352546	Developer	100%

Bảng 2: Danh sách thành viên & khối lượng công việc

# 1. Tổng quan dự án

Dự án "Hệ thống hỗ trợ Tutor tại Trường Đại học Bách khoa – ĐHQG TP.HCM" là một sáng kiến công nghệ nhằm mục tiêu hiện đại hóa và nâng cao hiệu quả của chương trình Tutor/Mentor. Đây là một chương trình có ý nghĩa quan trọng, được nhà trường triển khai nhằm hỗ trợ sinh viên phát triển một cách toàn diện cả về tri thức học thuật lẫn các kỹ năng cần thiết. Báo cáo này sẽ trình bày chi tiết về quá trình, từ việc phân tích bối cảnh, xác định yêu cầu, đến thiết kế và xây dựng hệ thống.

## 1.1. Giới thiệu dự án

Trong bối cảnh giáo dục đại học đang không ngừng đổi mới, việc ứng dụng công nghệ để tối ưu hóa các hoạt động hỗ trợ sinh viên là một yêu cầu tất yếu. Dự án này ra đời nhằm xây dựng một nền tảng phần mềm chuyên biệt, đóng vai trò xương sống cho chương trình Tutor, qua đó tạo một môi trường học tập tương tác, hiệu quả và có hệ thống tại Trường Đại học Bách khoa TP.HCM.

### 1.1.1. Mục đích

Mục đích chính của dự án là phát triển một phần mềm quản lý tập trung, giúp vận hành chương trình Tutor một cách hiệu quả, hiện đại và có khả năng mở rộng. Hệ thống này sẽ là cầu nối vững chắc giữa Tutor (giảng viên, nghiên cứu sinh, sinh viên năm trên có thành tích tốt) và sinh viên cần hỗ trợ. Thông qua đó, dự án hướng đến việc nâng cao chất lượng học tập, tăng cường sự tương tác và gắn kết trong cộng đồng sinh viên, đồng thời góp phần vào việc phát triển kỹ năng mềm và định hướng nghề nghiệp cho người học.

### 1.1.2. Bối cảnh và lý do cần hệ thống

Trên thực tế, chương trình Tutor/Mentor tại Trường Đại học Bách khoa TP.HCM đã được triển khai và mang lại những lợi ích nhất định. Tuy nhiên, quy trình vận hành hiện tại vẫn còn phụ thuộc nhiều vào các phương pháp thủ công, dẫn đến một số thách thức và hạn chế đáng kể:

- **Về quản lý thông tin và kết nối:** Việc quản lý hồ sơ năng lực của Tutor và nhu cầu cụ thể của sinh viên còn khó khăn, khiến quá trình ghép cặp chưa đạt được hiệu quả tối ưu. Sinh viên thường gặp khó khăn trong việc chủ động tìm kiếm và kết nối với người hướng dẫn phù hợp nhất.
- **Về tổ chức và sắp xếp:** Công tác lên lịch, thay đổi hoặc hủy các buổi học phụ thuộc nhiều vào việc trao đổi cá nhân, tiềm ẩn nguy cơ nhầm lẫn, thiếu sót và tốn nhiều thời gian không cần thiết.
- **Về đo lường và cải tiến:** Việc thiếu một công cụ theo dõi và đánh giá bài bản đã tạo ra một khoảng trống trong việc đo lường tiến bộ của sinh viên cũng như chất lượng của các buổi học, gây khó khăn cho việc cải tiến và nâng cao hiệu quả chương trình.

Trước những bất cập đó, việc xây dựng một "Hệ thống hỗ trợ Tutor" là một giải pháp cấp thiết. Bằng cách tự động hóa các quy trình từ quản lý, ghép cặp, lên lịch cho đến đánh giá, hệ thống được kỳ vọng sẽ giải quyết hiệu quả những tồn tại này, đáp ứng các đòi hỏi thực tiễn của môi trường giáo dục đại học trong kỷ nguyên số.

### 1.1.3. Kỳ vọng và mục tiêu

Dự án được định hướng bởi những kỳ vọng và mục tiêu rõ ràng, hướng đến lợi ích của các bên liên quan:

**Kỳ vọng:**

- **Đối với Sinh viên:** Hệ thống được kỳ vọng sẽ trở thành một cổng thông tin thân thiện, giúp sinh viên dễ dàng tìm kiếm sự hỗ trợ học thuật, chủ động lựa chọn Tutor, linh hoạt sắp xếp lịch học và nhận được sự giúp đỡ kịp thời, đúng nhu cầu.
- **Đối với Tutor:** Cung cấp một bộ công cụ số hóa mạnh mẽ để quản lý thông tin cá nhân, sắp xếp lịch làm việc một cách khoa học, theo dõi và ghi nhận sự tiến bộ của sinh viên
- **Đối với Nhà trường:** Trao cho các Khoa và Phòng ban một công cụ quản lý tổng thể, cho phép giám sát, phân tích và đánh giá hiệu quả của chương trình. Dữ liệu thu thập được sẽ là cơ sở thực tiễn để tối ưu hóa việc phân bổ nguồn lực và đưa ra các quyết sách quan trọng, chẳng hạn như xét điểm rèn luyện hoặc học bổng.

**Mục tiêu:**

- Phát triển một nền tảng phần mềm hoàn chỉnh, bao quát các chức năng cốt lõi: quản lý hồ sơ, đăng ký, ghép cặp, lên lịch, thông báo và đánh giá.
- Tích hợp liền mạch và an toàn với hạ tầng công nghệ thông tin hiện có của trường, bao gồm dịch vụ xác thực tập trung (HCMUT\_SSO), cơ sở dữ liệu lõi (HCMUT\_DATACORE) và thư viện số (HCMUT\_LIBRARY).
- Đảm bảo giao diện người dùng (UI/UX) thân thiện, trực quan, dễ sử dụng trên nhiều nền tảng, đồng thời thiết kế kiến trúc hệ thống theo hướng mở, sẵn sàng cho việc mở rộng và tích hợp các tính năng nâng cao trong tương lai.

### 1.1.4. Sản phẩm bàn giao

Kết thúc dự án, nhóm sẽ bàn giao các sản phẩm sau:

- **Báo cáo phân tích yêu cầu phần mềm:** Tài liệu mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống, cùng với các biểu đồ Use-case.
- **Tài liệu thiết kế hệ thống:** Bao gồm thiết kế kiến trúc tổng quan, thiết kế chi tiết các module, thiết kế giao diện người dùng (UI), và các biểu đồ liên quan như biểu đồ tuần tự, biểu đồ lớp.
- **Mã nguồn của ứng dụng (MVP - Minimum Viable Product):** Một phiên bản phần mềm có thể hoạt động được, bao gồm các chức năng cốt lõi đã được thống nhất.
- **Tài liệu hướng dẫn sử dụng:** Hướng dẫn chi tiết cho các đối tượng người dùng khác nhau như sinh viên, Tutor và quản trị viên.
- **Slide thuyết trình và video demo sản phẩm:** Trình bày tổng quan về dự án, các chức năng chính của hệ thống và demo cách thức hoạt động.
- **Báo cáo cuối kỳ:** Tổng hợp toàn bộ quá trình thực hiện dự án, từ việc phân tích yêu cầu, thiết kế, triển khai cho đến kết quả đạt được và những bài học kinh nghiệm.



## 1.2. Phạm vi dự án

Để đảm bảo tính khả thi và sự tập trung của dự án, việc xác định rõ ràng ranh giới là vô cùng quan trọng. Phần này sẽ trình bày cụ thể các chức năng sẽ được xây dựng, những hạng mục nằm ngoài khuôn khổ, cùng các ràng buộc về kỹ thuật và nghiệp vụ mà hệ thống phải tuân thủ.

### 1.2.1. Trong phạm vi

Phạm vi của dự án được xác định rõ ràng, tập trung vào việc phát triển một bộ chức năng cốt lõi, đủ mạnh để quản lý và vận hành chương trình Tutor một cách toàn diện. Các nhóm chức năng chính của hệ thống bao gồm:

- **Nền tảng Quản lý Tài khoản và Hồ sơ:** Cung cấp một không gian tập trung để người dùng (Sinh viên, Tutor) khởi tạo và quản lý thông tin cá nhân. Chức năng này cho phép cập nhật hồ sơ, năng lực chuyên môn và các nhu cầu hỗ trợ cụ thể.
- **Module Đăng ký và Ghép cặp:** Hệ thống sẽ cung cấp một module linh hoạt, hỗ trợ sinh viên đăng ký tham gia các chương trình học và được ghép cặp với Tutor thông qua hai hình thức:
  - **Thủ công:** Cho phép sinh viên chủ động tìm kiếm, xem xét hồ sơ và gửi yêu cầu đến Tutor mà họ cho là phù hợp nhất.
  - **Tự động:** Cung cấp cơ chế gợi ý Tutor dựa trên các tiêu chí cơ bản như môn học và sự trùng khớp về lịch rảnh.
- **Hệ thống Quản lý Lịch và Buổi học:** Đây là chức năng quan trọng, giúp tự động hóa toàn bộ quy trình sắp xếp và quản lý các buổi học. Các tính năng bao gồm việc Tutor thiết lập lịch rảnh, sinh viên đặt lịch, hệ thống tự động gửi thông báo khi có thay đổi, và Tutor thực hiện các tác vụ như điểm danh, cập nhật trạng thái buổi học.
- **Không gian Quản lý Tài liệu:** Hỗ trợ Tutor tải lên và chia sẻ các tài liệu học tập liên quan (như slide, bài tập, đề cương) đến sinh viên trong nhóm của mình, đồng thời cho phép sinh viên dễ dàng truy cập và tải xuống.
- **Cơ chế Đánh giá, Phản hồi và Báo cáo:** Nhằm đảm bảo và không ngừng nâng cao chất lượng, hệ thống tích hợp một cơ chế thu thập phản hồi từ sinh viên sau mỗi buổi học. Dữ liệu này sẽ được tổng hợp thành các báo cáo đa chiều, phục vụ cho từng đối tượng người dùng, từ sinh viên, Tutor cho đến các cấp quản lý của nhà trường.

### 1.2.2. Ngoài phạm vi

Nhằm đảm bảo đội ngũ phát triển có thể tập trung vào các giá trị cốt lõi và hoàn thành dự án đúng tiến độ, phạm vi đã được giới hạn một cách có chủ đích. Các chức năng nâng cao sau đây sẽ được xem xét cho các giai đoạn phát triển trong tương lai và không nằm trong khuôn khổ của phiên bản hiện tại:

- **Các thuật toán Ghép cặp thông minh (AI-based Matching):** Việc ứng dụng Trí tuệ Nhân tạo để phân tích sâu hồ sơ và tự động tối ưu hóa việc ghép cặp sẽ không được triển khai.
- **Xây dựng Cộng đồng Trực tuyến:** Các tính năng mang tính mạng xã hội như diễn đàn (forum) hay nhóm trò chuyện (chat groups) để mở rộng tương tác sẽ không được phát triển.

- **Hỗ trợ các chương trình Tutor phi học thuật:** Hệ thống chỉ tập trung vào việc hỗ trợ học thuật. Các hoạt động tư vấn khác như hướng nghiệp, kỹ năng mềm hay tâm lý học đường đều nằm ngoài phạm vi.
- **Module cá nhân hóa:** Các tính năng sử dụng Trí tuệ Nhân tạo (AI) để phân tích quá trình học tập của từng sinh viên, từ đó tự động đề xuất lộ trình học tập riêng biệt hoặc gợi ý các tài liệu phù hợp sẽ không được triển khai. Lý do là các chức năng này đòi hỏi thuật toán phức tạp và một khối lượng lớn dữ liệu lịch sử để có thể hoạt động hiệu quả. Trong giai đoạn đầu, dự án sẽ ưu tiên tập trung vào việc hoàn thiện nền tảng cốt lõi là kết nối Tutor và sinh viên một cách hiệu quả nhất.

### 1.2.3. Ràng buộc hệ thống

Quá trình thiết kế và phát triển hệ thống phải tuân thủ nghiêm ngặt các ràng buộc quan trọng sau:

- **Tích hợp với Hạ tầng Công nghệ của HCMUT:** Đây là một ràng buộc mang tính nền tảng, đòi hỏi hệ thống phải có khả năng kết nối và giao tiếp hiệu quả với các dịch vụ hiện có của trường, cụ thể là:
  - Sử dụng dịch vụ xác thực tập trung HCMUT\_SSO cho việc đăng nhập
  - Đồng bộ dữ liệu cá nhân từ hệ thống lõi HCMUT\_DATACORE
  - Liên kết với kho tài nguyên của thư viện số HCMUT\_LIBRARY.
- **Cơ chế Phân quyền Dựa trên Vai trò Tập trung:** Hệ thống không tự định nghĩa vai trò người dùng mà phải tự động gán quyền truy cập dựa trên thông tin vai trò (sinh viên, tutor, điều phối viên...) được cung cấp từ hệ thống trung tâm của nhà trường.
- **Yêu cầu về Cơ sở dữ liệu cho Phiên bản MVP:** Theo đặc tả của đề bài, phiên bản MVP (Minimum Viable Product) không bắt buộc phải triển khai một hệ quản trị cơ sở dữ liệu hoàn chỉnh ở tầng backend. Dữ liệu có thể được mã hóa cứng (hard-coded) trong mã nguồn để phục vụ mục đích trình diễn.
- **Tuân thủ Quy định về sử dụng Generative AI:** Nhóm cam kết tuân thủ các quy định về đạo đức học thuật, đặc biệt là việc khai báo minh bạch và có trách nhiệm về mức độ và phạm vi sử dụng các công cụ AI tạo sinh trong suốt quá trình thực hiện dự án.
- **Khai báo sử dụng Generative AI:**
  - Hỗ trợ liệt kê các yêu cầu chức năng (Functional Requirements).
  - Đề xuất tên cho các usecase.
  - Hỗ trợ liệt kê các hành động chính cho từng Actors.

## 1.3. Tài liệu tham khảo

Để đảm bảo hệ thống được xây dựng không chỉ đáp ứng yêu cầu mà còn hiệu quả và phù hợp với thực tiễn, nhóm đã tiến hành một quá trình nghiên cứu và tham chiếu (benchmarking) các nền tảng hỗ trợ học tập và quản lý lịch hẹn tương tự đang vận hành thành công trên thị trường. Việc phân tích này giúp chúng tôi đúc kết những bài học giá trị, từ đó định hình các tính năng cốt lõi cho hệ thống.

### 1.3.1. Các nền tảng tương tự

Nhóm đã tập trung phân tích ba nhóm nền tảng chính, mỗi nhóm đại diện cho một khía cạnh quan trọng của hệ thống cần xây dựng:

- **Nền tảng Gia sư Trực tuyến (Online Tutoring Platforms):** Các dịch vụ thương mại như Chegg, TutorMe, và Preply là những ví dụ tiêu biểu. Chúng nổi bật với hệ thống tìm kiếm và bộ lọc mạnh mẽ, cho phép người dùng tìm kiếm gia sư theo môn học, chuyên ngành, khung giờ và mức đánh giá. Đặc biệt, hồ sơ (profile) của gia sư được xây dựng rất chi tiết, tạo sự tin cậy thông qua việc trình bày kinh nghiệm, trình độ học vấn và các nhận xét xác thực từ học viên.
- **Hệ thống Quản lý Sinh viên Chuyên dụng:** Các giải pháp phần mềm như Navigate và Starfish được nhiều trường đại học trên thế giới tin dùng để hỗ trợ sự thành công của sinh viên. Điểm mạnh cốt lõi của chúng là khả năng tích hợp sâu rộng với cơ sở dữ liệu của nhà trường, cho phép theo dõi toàn diện tiến trình học tập. Các hệ thống này không chỉ đơn thuần hỗ trợ đặt lịch hẹn với cố vấn học tập mà còn có khả năng gửi cảnh báo sớm khi nhận diện các sinh viên có dấu hiệu sa sút.
- **Nền tảng Đặt lịch hẹn Chuyên dụng:** Các công cụ như Calendly và Doodle là minh chứng cho hiệu quả của sự đơn giản và tập trung. Chúng giải quyết triệt để bài toán sắp xếp lịch hẹn bằng cách loại bỏ hoàn toàn các bước trao đổi thủ công. Người dùng chỉ cần thiết lập các khung giờ khả dụng và chia sẻ một liên kết duy nhất; hệ thống sẽ tự động xử lý việc đặt lịch, kiểm tra trùng lặp và gửi thông báo xác nhận, nhắc nhở.

### 1.3.2. Bài học rút ra

Từ việc phân tích các giải pháp trên, nhóm đã đúc kết được những bài học kinh nghiệm sâu sắc, đóng vai trò kim chỉ nam cho quá trình thiết kế hệ thống:

- **Trải nghiệm người dùng (UX) là yếu tố quyết định:** Quy trình từ tìm kiếm Tutor đến đặt lịch thành công phải được thiết kế tối giản, nhanh chóng và trực quan. Bất kỳ sự phức tạp nào trong luồng thao tác đều có thể làm giảm tỷ lệ sử dụng của sinh viên.
- **Hiệu quả của tìm kiếm và bộ lọc là giá trị cốt lõi:** Chức năng quan trọng nhất đối với sinh viên là khả năng tìm được đúng người hướng dẫn mình cần. Do đó, hệ thống bắt buộc phải có các bộ lọc cơ bản và hiệu quả như khoa, môn học và lịch rảnh.
- **Tự động hóa là chìa khóa của sự hiệu quả:** Toàn bộ quy trình đặt, hủy và nhắc lịch cần được tự động hóa. Điều này không chỉ giúp tiết kiệm thời gian cho cả Tutor và sinh viên mà còn giảm thiểu đáng kể các sai sót do con người.
- **Sự tin cậy được vun đắp từ cơ chế đánh giá:** Việc cho phép sinh viên để lại nhận xét và đánh giá sau mỗi buổi học là phương thức minh bạch và hiệu quả nhất để xây dựng một cộng đồng chất lượng, đồng thời cung cấp thông tin tham khảo giá trị cho những người dùng khác.
- **Quản lý tập trung tạo ra hiệu quả vận hành:** Thay vì các quy trình rời rạc, một bảng điều khiển (dashboard) quản lý tập trung sẽ cung cấp cho các bên liên quan một cái nhìn tổng thể, giúp việc theo dõi và đánh giá hiệu quả chương trình trở nên dễ dàng và chính xác hơn.

### 1.3.3. Tính năng tổng hợp cho hệ thống

Dựa trên những bài học đúc kết và các yêu cầu đặc thù của Trường Đại học Bách khoa TP.HCM, nhóm đã tổng hợp và đề xuất các nhóm tính năng trọng tâm cho hệ thống như sau:

- **Nền tảng Quản lý Người dùng và Hồ sơ chuyên nghiệp:** Mỗi người dùng sẽ sở hữu một không gian hồ sơ riêng, nơi Tutor có thể trình bày một cách chuyên nghiệp về chuyên môn, kinh nghiệm và thành tích của mình.
- **Module Tìm kiếm và Gợi ý Ghép cặp thông minh:** Cho phép sinh viên tìm kiếm Tutor theo các tiêu chí linh hoạt, đồng thời tích hợp chức năng gợi ý cơ bản dựa trên môn học và lịch trình tương thích.
- **Module Quản lý Lịch hẹn Tinh gọn:** Lấy cảm hứng từ sự đơn giản của Calendly, module này cho phép Tutor thiết lập lịch rảnh một cách trực quan, giúp sinh viên đặt lịch chỉ với vài thao tác. Toàn bộ quy trình thông báo xác nhận và nhắc nhở sẽ được tự động hóa.
- **Cơ chế Đánh giá và Phản hồi hai chiều:** Sau mỗi buổi học, hệ thống sẽ chủ động mời sinh viên đánh giá, từ đó tạo ra một nguồn dữ liệu quý giá, tạo một vòng lặp cải tiến liên tục cho chất lượng chương trình.
- **Module Báo cáo và Thống kê Trực quan:** Cung cấp các bảng điều khiển cho phép ban quản lý, các khoa và phòng ban dễ dàng theo dõi các chỉ số hiệu suất quan trọng (KPIs) như số lượng buổi học, tỷ lệ tham gia và mức độ hài lòng của sinh viên.
- **Khả năng Tích hợp sâu với Hạ tầng Công nghệ HCMUT:** Đây là yếu tố khác biệt và mang tính nền tảng, đảm bảo dữ liệu được đồng bộ, nhất quán và bảo mật thông qua việc kết nối chặt chẽ với các hệ thống lõi của nhà trường như HCMUT\_SSO và HCMUT\_DATA CORE.

## 2. Phân tích yêu cầu dự án

Trong chương này, nhóm em sẽ tiến hành xác định và phân tích các yêu cầu của hệ thống. Trước hết, việc nhận diện các tác nhân (Actors) tương tác với hệ thống là bước quan trọng để hiểu rõ các luồng chức năng và mục tiêu mà phần mềm cần đáp ứng.

### 2.1. Tác nhân

Tác nhân (Actor) là một thực thể bên ngoài, có thể là người dùng hoặc một hệ thống khác, tương tác trực tiếp với hệ thống để thực hiện một mục tiêu cụ thể. Dựa trên vai trò và cách thức tương tác, các tác nhân của "Hệ thống hỗ trợ Tutor" được phân loại thành tác nhân chính và tác nhân phụ.

#### 2.1.1. Tác nhân chính

##### 2.1.1.1. Tutor

Tutor có thể là sinh viên giỏi, NCS, hoặc giảng viên, được đăng ký vào hệ thống để hỗ trợ học tập. Họ có trách nhiệm quản lý hồ sơ, lịch rảnh, tổ chức buổi học và theo dõi tiến độ của sinh viên.

- **Tạo và cập nhật lịch rảnh:**
  - **Input:** Ngày, giờ, hình thức học (online/offline).
  - **Process:** Hệ thống lưu lại khung giờ rảnh, đồng bộ với module đặt lịch.
  - **Output:** Lịch rảnh hiển thị cho sinh viên để chọn.
  - **Constraints:** Không được nhập trùng khung giờ.
  - **Acceptance:** Sinh viên có thể đặt lịch học trong khung rảnh của Tutor.
  - **Error handling:** Nếu nhập sai định dạng hoặc trùng lịch thì hệ thống báo lỗi.
- **Mở (online/offline), hủy, đổi buổi học:**
  - **Input:** Yêu cầu mở buổi học, hoặc yêu cầu hủy/đổi.
  - **Process:** Hệ thống kiểm tra lịch đã có sinh viên đặt chưa, xử lý cập nhật.
  - **Output:** Buổi học được thêm/sửa/xoá trong hệ thống.
  - **Constraints:** Hủy/đổi  $\geq 3h$  trước giờ học
  - **Acceptance:** Sinh viên và Tutor đều nhận được thông báo cập nhật.
  - **Error handling:** Nếu yêu cầu đổi sát giờ hệ thống từ chối, báo lỗi.
- **Nhận thông báo và nhắc nhở giờ dạy:**
  - **Input:** Lịch học sắp diễn ra.
  - **Process:** Hệ thống gửi thông báo (noti/email).
  - **Output:** Tutor nhận được thông báo đúng hạn.
  - **Constraints:** Thông báo phải được gửi  $\geq 30$  phút trước giờ học.
  - **Acceptance:** Tutor xác nhận đã đọc thông báo.

- **Error handling:** Nếu gửi lỗi → hệ thống gửi lại lần 2 hoặc báo qua email dự phòng.

- **Theo dõi tiến bộ sinh viên:**

- **Input:** Điểm, nhận xét, đánh giá sau buổi học.
- **Process:** Tutor nhập vào form đánh giá, hệ thống lưu lại.
- **Output:** Báo cáo tiến bộ gắn với hồ sơ sinh viên.
- **Constraints:** Chỉ Tutor đã dạy sinh viên đó mới được nhập.
- **Acceptance:** Khoa/bộ môn có thể truy cập báo cáo.
- **Error handling:** Nếu buổi học chưa hoàn tất thì từ chối ghi nhận.

- **Điểm danh và record:**

- **Input:** ID sinh viên tham gia, mã buổi học.
- **Process:** Hệ thống điểm danh, ghi log tham dự.
- **Output:** Record buổi học (thời lượng, người tham gia).
- **Constraints:** Mỗi SV chỉ được điểm danh vào 1 buổi học tại 1 thời điểm.
- **Acceptance:** Log lưu thành công và hiển thị trong báo cáo.
- **Error handling:** Nếu trùng ID hệ thống từ chối, báo lỗi.

- **Cập nhật trạng thái buổi học:**

- **Input:** Trạng thái (hoàn thành, huỷ, đang diễn ra).
- **Process:** Tutor xác nhận trạng thái, hệ thống lưu lại.
- **Output:** Buổi học hiển thị trạng thái mới.
- **Constraints:** Trạng thái chỉ được thay đổi bởi Tutor của buổi học.
- **Acceptance:** Sinh viên và khoa/bộ môn nhìn thấy trạng thái chính xác.
- **Error handling:** Nếu cập nhật sai thì hệ thống cho phép sửa lại trong 24h.

- **Đăng nội dung bài học:**

- **Input:** File/tài liệu/note buổi học.
- **Process:** Upload vào hệ thống, lưu trữ trong HCMUT\_LIBRARY.
- **Output:** Sinh viên có thể tải xuống.
- **Constraints:** Dung lượng  $\leq 50\text{MB/file}$ .
- **Acceptance:** Nội dung hiển thị đúng với sinh viên liên quan.
- **Error handling:** Nếu file hỏng thì báo lỗi, yêu cầu upload lại.

### 2.1.1.2. Sinh viên

Sinh viên là đối tượng cần hỗ trợ, tham gia hệ thống để tìm Tutor, đặt lịch học và nhận hỗ trợ học tập.

- **Tạo tài khoản, hồ sơ cá nhân:**

- **Input:** Họ tên, MSSV, email, số điện thoại, thông tin học tập (GPA, môn cần hỗ trợ).

- **Process:** Hệ thống kiểm tra định dạng dữ liệu, đồng bộ với HCMUT\_DATACORE.
- **Output:** Hồ sơ cá nhân của SV được lưu và hiển thị trong hệ thống.
- **Constraints:** MSSV và email phải trùng khớp dữ liệu HCMUT.
- **Acceptance:** SV có thể đăng nhập và sử dụng các chức năng khác.
- **Error handling:** Nếu dữ liệu không hợp lệ → hệ thống báo lỗi, yêu cầu sửa.

• **Đăng ký chương trình học:**

- **Input:** Môn học hoặc lĩnh vực cần hỗ trợ, nguyện vọng học tập.
- **Process:** Hệ thống ghi nhận nhu cầu, đồng bộ với dữ liệu đào tạo và gợi ý Tutor phù hợp.
- **Output:** Hồ sơ SV được cập nhật với chương trình đã đăng ký.
- **Constraints:** Chỉ được đăng ký trong danh sách môn/lĩnh vực mà hệ thống hỗ trợ.
- **Acceptance:** SV thấy chương trình học hiển thị trong hồ sơ.
- **Error handling:** Nếu môn/lĩnh vực không hợp lệ thì hệ thống báo lỗi, yêu cầu chọn lại.

• **Lựa chọn Tutor / được ghép tự động:**

- **Input:** Nhu cầu hỗ trợ (môn, lịch, hình thức).
- **Process:**
  - \* **Thủ công:** SV chọn Tutor trong danh sách.
  - \* **Tự động:** Hệ thống so khớp theo khoa/ngành, lịch rảnh, AI ranking.
- **Output:** Ghép cặp Tutor – SV được xác lập.
- **Constraints:** Một SV chỉ có 1 Tutor chính tại một thời điểm.
- **Acceptance:** SV thấy thông tin Tutor trong hồ sơ.
- **Error handling:** Nếu lịch trùng thì yêu cầu chọn lại hoặc hệ thống gợi ý Tutor khác.

• **Đặt lịch học (cảnh báo trùng lịch):**

- **Input:** Ngày, giờ, môn học.
- **Process:** Hệ thống kiểm tra lịch rảnh của Tutor và lịch của SV.
- **Output:** Lịch học mới được thêm.
- **Constraints:** Không được đặt trùng với lịch học hoặc lịch thi chính thức.
- **Acceptance:** Lịch hiển thị trong tài khoản SV và Tutor.
- **Error handling:** Nếu trùng lịch thì cảnh báo, từ chối đặt.

• **Nhận thông báo và nhắc nhở giờ học:**

- **Input:** Lịch học sắp diễn ra.
- **Process:** Hệ thống gửi thông báo (noti/email).
- **Output:** SV nhận được thông báo.
- **Constraints:** Thông báo  $\geq 30$  phút trước giờ học.
- **Acceptance:** SV xác nhận thông báo trên hệ thống.

- **Error handling:** Nếu thông báo lỗi thì gửi lại qua email dự phòng.

- **Phản hồi và đánh giá chất lượng buổi học:**

- **Input:** Điểm (1–5 sao), bình luận nhận xét.
- **Process:** Hệ thống lưu đánh giá gắn với buổi học và Tutor.
- **Output:** Thông tin phản hồi hiển thị cho Tutor và khoa/bộ môn.
- **Constraints:** Chỉ được đánh giá sau khi buổi học hoàn thành.
- **Acceptance:** Đánh giá hiển thị trong báo cáo tổng hợp.
- **Error handling:** Nếu buổi học chưa hoàn tất → từ chối đánh giá.

## 2.1.2. Tác nhân phụ

### 2.1.2.1. Khoa/Bộ môn

- **Nhận đánh giá và tổng hợp kết quả của sinh viên:**

- **Input:** Đánh giá (điểm số, nhận xét) từ sinh viên sau buổi học.
- **Process:** Hệ thống tổng hợp các phản hồi, phân loại theo môn học/Tutor.
- **Output:** Báo cáo chất lượng buổi học theo lớp, môn, Tutor.
- **Constraints:** Chỉ sử dụng đánh giá từ các buổi học hợp lệ.
- **Acceptance:** Báo cáo được cập nhật định kỳ (theo tuần/tháng).
- **Error handling:** Nếu thiếu dữ liệu đánh giá thì hệ thống ghi chú “chưa có đủ dữ liệu”.

- **Quản lý chất lượng Tutor và SV:**

- **Input:** Hồ sơ Tutor, hồ sơ SV, số buổi học, đánh giá.
- **Process:** Khoa theo dõi, so sánh chất lượng giảng dạy và mức độ tiến bộ của SV.
- **Output:** Bảng xếp hạng/đánh giá Tutor và tổng kết tiến độ SV.
- **Constraints:** Dữ liệu phải dựa trên lịch sử buổi học và đánh giá chính thức.
- **Acceptance:** Báo cáo thể hiện chính xác tình hình giảng dạy – học tập.
- **Error handling:** Nếu dữ liệu không đồng bộ → hệ thống tự động cảnh báo để kiểm tra.

- **Theo dõi tiến độ học tập của sinh viên:**

- **Input:** GPA trước/sau, kết quả môn học, log buổi học.
- **Process:** Hệ thống đối chiếu tiến độ, xác định sự cải thiện.
- **Output:** Báo cáo cá nhân/tập thể về tiến bộ của SV.
- **Constraints:** Chỉ tính các SV tham gia tối thiểu X buổi học.
- **Acceptance:** Báo cáo có thể dùng làm cơ sở xét khen thưởng hoặc hỗ trợ.
- **Error handling:** Nếu thiếu GPA hoặc dữ liệu học tập → báo cáo đánh dấu “khuyết dữ liệu”.



#### 2.1.2.2. Phòng Công tác sinh viên

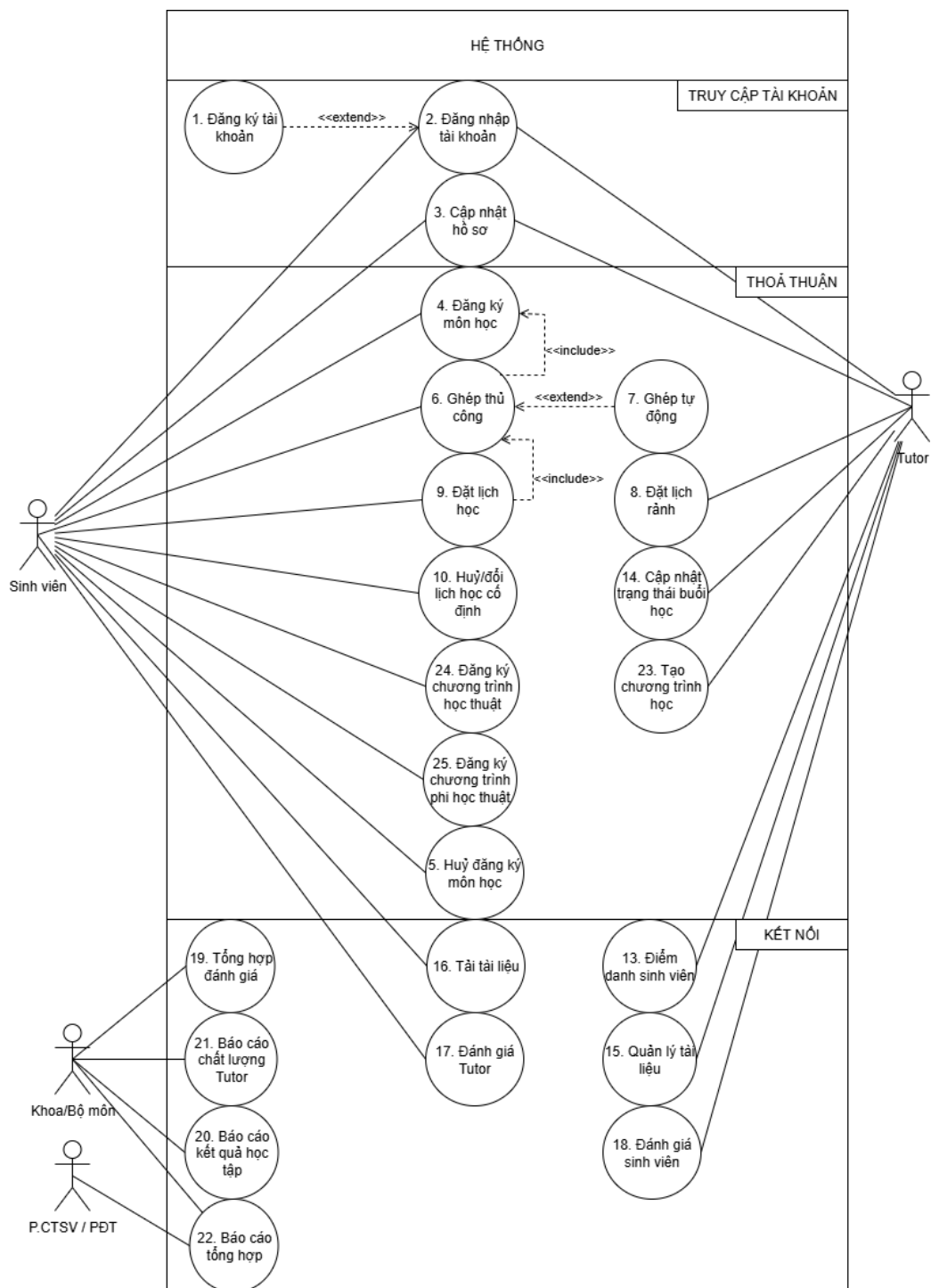
- **Nắm bắt GPA sinh viên trước và sau khi tham gia:**
  - **Input:** GPA ban đầu, GPA cập nhật sau kỳ học.
  - **Process:** Hệ thống tự động lấy dữ liệu từ HCMUT\_DATACORE, đối chiếu kết quả trước/sau.
  - **Output:** Báo cáo so sánh GPA từng sinh viên.
  - **Constraints:** Dữ liệu GPA phải đồng bộ chính xác từ hệ thống đào tạo.
  - **Acceptance:** PCTSV có thể tra cứu sự thay đổi kết quả học tập của SV.
  - **Error handling:** Nếu thiếu dữ liệu GPA thì hệ thống báo lỗi, yêu cầu đồng bộ lại.
- **Tổng hợp kết quả tham gia:**
  - **Input:** Danh sách SV, log số buổi học, đánh giá từ Tutor.
  - **Process:** Hệ thống thống kê tần suất tham gia và kết quả học tập.
  - **Output:** Báo cáo mức độ tham gia của SV
  - **Constraints:** hỉ tính những SV tham gia tối thiểu số buổi học quy định.
  - **Acceptance:** Báo cáo được xuất file (Excel/PDF) và tích hợp vào hệ thống quản lý SV.
  - **Error handling:** Nếu dữ liệu không đầy đủ thì báo cáo gắn cờ “chưa hoàn chỉnh”.
- **Ghi nhận kết quả tham gia để xét điểm rèn luyện / học bổng:**
  - **Input:** Báo cáo tổng hợp SV tham gia chương trình Tutor.
  - **Process:** PCTSV đối chiếu với quy chế điểm rèn luyện, học bổng.
  - **Output:** Điểm rèn luyện/học bổng của SV được cập nhật.
  - **Constraints:** Chỉ SV có tham gia hợp lệ, đủ số buổi quy định mới được ghi nhận.
  - **Acceptance:** Kết quả được tích hợp vào hệ thống xét điểm rèn luyện và học bổng.
  - **Error handling:** Nếu báo cáo thiếu dữ liệu thì đánh dấu “pending” cho đến khi bổ sung.

#### 2.1.2.3. Phòng Đào tạo

- **Quản lý và theo dõi hồ sơ Tutor:**
  - **Input:** Hồ sơ cá nhân, chuyên môn, lịch rảnh của Tutor.
  - **Process:** PDT xem, kiểm tra và xác nhận hồ sơ Tutor.
  - **Output:** Danh sách Tutor hợp lệ được duyệt.
  - **Constraints:** Chỉ Tutor đủ điều kiện (ví dụ GPA  $\geq 7.0$ , có chuyên môn rõ ràng) mới được phê duyệt.
  - **Acceptance:** Hồ sơ hiển thị trong hệ thống cho SV lựa chọn.
  - **Error handling:** Nếu hồ sơ không hợp lệ thì trả lại yêu cầu cập nhật.
- **Theo dõi số lượng buổi học:**
  - **Input:** Log buổi học từ hệ thống.

- **Process:** Hệ thống tổng hợp số buổi học theo Tutor, theo SV, theo môn.
  - **Output:** Báo cáo thống kê buổi học (ngày, giờ, trạng thái, số lượng).
  - **Constraints:** Chỉ tính các buổi học hợp lệ (có điểm danh).
  - **Acceptance:** Báo cáo hiển thị chính xác cho quản lý đào tạo.
  - **Error handling:** Nếu dữ liệu log thiếu thì hệ thống cảnh báo “incomplete data”.
- **Tối ưu phân bổ nguồn lực giữa Tutor và SV:**
    - **Input:** Danh sách Tutor, danh sách SV đăng ký, nhu cầu hỗ trợ.
    - **Process:** Hệ thống gợi ý phân bổ Tutor cho SV (theo ngành, lịch rảnh, số lượng tối đa).
    - **Output:** Bảng phân công Tutor – SV.
    - **Constraints:** Một Tutor chỉ nhận tối đa số SV theo quy định (ví dụ  $\leq 5$  SV).
    - **Acceptance:** Phân bổ hợp lý, không quá tải Tutor, đáp ứng nhu cầu SV.
    - **Error handling:** Nếu số SV vượt quá khả năng phân bổ thì hệ thống cảnh báo, yêu cầu thêm Tutor.

## 2.2. Sơ đồ usecase toàn hệ thống



Hình 1: Sơ đồ Usecase toàn hệ thống



## 2.3. User Stories

### 2.3.1. Module Quản lý Tài khoản và Hồ sơ

#### US-01: Đăng ký tài khoản

Là 1 user, tôi muốn đăng ký tài khoản trên hệ thống HCMUT SSO bằng số điện thoại /email mình.

#### US-02: Đăng nhập hệ thống

Là user đã có tài khoản, tôi muốn đăng nhập vào hệ thống một cách an toàn bằng email/SDT và mật khẩu, để truy cập dashboard phù hợp với vai trò của mình.

#### US-03: Cập nhật hồ sơ

Là 1 user, tôi muốn chỉnh sửa và cập nhật hồ sơ cá nhân (email, SDT, chuyên môn, GPA) để thông tin luôn chính xác và mới nhất.

### 2.3.2. Module Đăng ký chương trình học

#### US-04: Đăng ký chương trình học

Là một sinh viên, tôi muốn đăng kí môn học phù hợp chuyên ngành và tìm tutor phù hợp để có thể học tập và luyện thi.

#### US-05: Hủy đăng ký chương trình học

Là một sinh viên, tôi muốn hủy môn đã đăng ký chương trình học khi cảm thấy không còn nhu cầu.

### 2.3.3. Module Ghép cặp Tutor – SV

#### US-06: Ghép thủ công (SV chọn Tutor)

Là một sinh viên, tôi muốn chọn tutor từ danh sách đề xuất phù hợp chuyên ngành và thời gian biểu.

#### US-07: Ghép tự động (hệ thống đề xuất Tutor)

Là một sinh viên, tôi muốn hệ thống chọn tutor dựa chuyên môn và thời gian biểu của tôi.

### 2.3.4. Module Quản lý lịch học

#### US-08: Tạo lịch rảnh (Tutor)

Là tutor, tôi muốn tạo lịch rảnh để sinh viên có thể xem, sắp xếp và đăng ký.

#### US-09: Đặt lịch học (SV)

Là sinh viên, tôi muốn đặt lịch học với tutor phù hợp thời gian biểu của bản thân.



#### **US-10: Hủy/Đổi lịch học**

Là sinh viên, tôi muốn hủy hoặc đổi lịch học để phù hợp với thay đổi của bản thân.

### **2.3.5. Module Thông báo và nhắc nhở**

#### **US-11: Gửi thông báo lịch học**

Là user, tôi muốn nhận thông báo khi có lịch học mới được đặt hoặc thay đổi.

#### **US-12: Gửi nhắc nhở buổi học**

Là user, tôi muốn nhận thông báo nhắc nhở trước buổi học để có thể chuẩn bị.

### **2.3.6. Module Quản lý buổi học và điểm danh**

#### **US-13: Điểm danh sinh viên**

Là tutor, tôi muốn điểm danh sinh viên để có nắm rõ số lượng sinh viên tham gia buổi học.

#### **US-14: Cập nhật trạng thái buổi học**

Là tutor, tôi muốn cập nhật trạng thái buổi học để sinh viên có thể nắm rõ tình hình.

### **2.3.7. Module Quản lý tài liệu học tập**

#### **US-15: Tutor upload tài liệu**

Là tutor, tôi muốn đăng tải tài liệu học tập để sinh viên có thể sử dụng trong quá trình học tập.

#### **US-16: SV tải tài liệu**

Là sinh viên, tôi muốn tải xuống tài liệu học tập được tutor chia sẻ để ôn tập.

### **2.3.8. Module Đánh giá và phản hồi**

#### **US-17: Sinh viên đánh giá Tutor**

Là sinh viên, tôi muốn đánh giá tutor sau khi kết môn học để sinh viên khác biết về chất lượng giảng dạy.

#### **US-18: Tutor đánh giá sinh viên**

Là tutor, tôi muốn đánh thái độ học tập và mức độ tham gia của sinh viên để phản hồi cho khoa.

#### **US-19: Khoa/BM tổng hợp đánh giá**

Là Khoa, tôi muốn nhận tổng hợp đánh giá của sinh viên và tutor.

### **2.3.9. Module Thống kê và báo cáo**

#### **US-20: Báo cáo kết quả học tập SV**

Là khoa, tôi muốn tổng hợp và xác nhận kết quả học tập của sinh viên sau khi môn học kết thúc.



#### **US-21: Báo cáo chất lượng Tutor**

Là khoa, tôi muốn tổng hợp và xác nhận báo cáo chất lượng giảng dạy của tutor qua đánh giá của sinh viên và dữ liệu buổi học.

#### **US-22: Báo cáo tổng hợp (Khoa/ PCTSV/PDT)**

Là Khoa, Phòng Công tác Sinh viên (PCTSV) và Phòng Đào tạo (PDT), tôi muốn có thể xem báo cáo tổng hợp toàn hệ thống sau khi dữ liệu từ các báo cáo con (từ Sinh viên, Tutor) đã được xác nhận và tổng hợp.

### **2.3.10. Module Chương trình học thuật và phi học thuật**

#### **US-23: Tutor tạo chương trình học**

Là tutor, tôi muốn tạo thêm chương trình học tập mới(học thuật hoặc phi học thuật) để sinh viên có học hỏi và tìm hiểu thêm kiến thức mới.

#### **US-24: Sinh viên đăng ký chương trình học thuật**

Là sinh viên, tôi muốn đăng ký tham gia chương trình học thuật để có thể học tập và trao đổi thêm kiến thức.

#### **US-25: Sinh viên đăng ký chương trình phi học thuật**

Là sinh viên, tôi muốn đăng ký tham gia chương trình phi học thuật để có thể nâng cao kỹ năng mềm và tham gia các hoạt động ngoại khóa.

## 2.4. Yêu cầu chức năng

### 2.4.1. Module Quản lý Tài khoản và Hồ sơ

#### Use Case 01: Đăng ký tài khoản

<b>ID</b>	UC-01		
<b>Tên</b>	Đăng ký tài khoản		
<b>Mô tả</b>	Người dùng (SV hoặc Tutor) đăng ký tài khoản mới để tham gia hệ thống.		
<b>Actor chính</b>	Sinh viên, Tutor		
<b>Actor phụ</b>	Hệ thống xác thực OTP (Email), Admin PDT		
<b>Tiền điều kiện</b>	Người dùng chưa có tài khoản Có email hoặc số điện thoại hợp lệ		
<b>Hậu điều kiện</b>	Tài khoản hợp lệ được tạo, có ID duy nhất. Hồ sơ người dùng được lưu trong cơ sở dữ liệu		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Người dùng	Chọn chức năng “Đăng ký”.
	2	Người dùng	Nhập thông tin cá nhân (SV: MSSV, GPA, khoa; Tutor: chuyên môn, GPA $\geq 3.0$ hoặc giấy xác minh).
	3	Hệ thống	Kiểm tra định dạng dữ liệu.
	4	Hệ thống	Kiểm tra trùng MSSV/email. Nếu trùng $\rightarrow$ báo lỗi, dừng quy trình.
	5	Hệ thống	Gửi OTP xác thực qua email.
	6	Người dùng	Nhập OTP nhận được.
	7	Hệ thống	Kiểm tra OTP.
	8	Hệ thống	Lưu dữ liệu hợp lệ, gán UserID duy nhất.
	9	Hệ thống	Hiển thị thông báo “Đăng ký thành công”
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Người dùng	Nhập email không hợp lệ $\rightarrow$ Hệ thống báo lỗi, yêu cầu nhập lại (quay về bước 2).
	5a	Người dùng	Người dùng nhập OTP sai $\rightarrow$ Hệ thống báo lỗi, cho nhập lại tối đa 3 lần.
	7a	Hệ thống	Phát hiện MSSV/email đã tồn tại $\rightarrow$ hiển thị “Tài khoản đã có”.
<b>Ngoại lệ</b>	Hệ thống không gửi được OTP (lỗi server/email) $\rightarrow$ hiển thị “Vui lòng thử lại sau”. Hệ thống lỗi khi lưu dữ liệu vào DB $\rightarrow$ rollback, hiển thị “Thao tác thất bại”.		
<b>Business Rules</b>	Một MSSV/email chỉ được dùng cho 1 tài khoản. Tutor phải được xác minh bởi PDT trước khi tài khoản kích hoạt.		
<b>Data requirement</b>	Users(userID, role, email, password, MSSV, GPA, faculty) TutorProfile(tutorID, chuyên môn, giấy xác minh)		

Bảng 3: Bảng đặc tả chức năng đăng ký tài khoản



## Use Case 02: Đăng nhập

<b>ID</b>	UC-02		
<b>Tên</b>	Đăng nhập		
<b>Mô tả</b>	Người dùng đăng nhập để truy cập hệ thống bằng tài khoản đã đăng ký.		
<b>Actor chính</b>	SV, Tutor, Admin		
<b>Actor phụ</b>	Hệ thống xác thực đăng nhập		
<b>Tiền điều kiện</b>	Người dùng đã có tài khoản hợp lệ.		
<b>Hậu điều kiện</b>	Người dùng truy cập được dashboard theo đúng quyền		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Người dùng	Mở giao diện đăng nhập
	2	Người dùng	Nhập email/MSSV và mật khẩu.
	3	Hệ thống	Kiểm tra định dạng dữ liệu đầu vào
	4	Hệ thống	Kiểm tra thông tin đăng nhập trong DB.
	5	Hệ thống	Xác nhận thông tin đúng → mở phiên đăng nhập và chuyển đến giao diện chính.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Người dùng	Nhập sai mật khẩu → Hệ thống báo lỗi “Sai mật khẩu”, cho nhập lại tối đa 5 lần.
	4a	Hệ thống	Phát hiện tài khoản bị khóa → hiển thị “Tài khoản bị khóa, liên hệ Admin”..
<b>Ngoại lệ</b>	Gặp lỗi khi truy vấn DB → hiển thị “Không thể đăng nhập lúc này, vui lòng thử lại sau”.		
<b>Business Rules</b>	Nếu nhập sai mật khẩu quá 5 lần → Hệ thống tự động khóa tài khoản trong 30 phút. Admin có quyền thiết lập lại mật khẩu cho người dùng.		
<b>Data requirement</b>	Users(userID, email, password, role, status, lastLogin)		

Bảng 4: Bảng đặc tả chức năng đăng nhập





### Use Case 03: Cập nhật hồ sơ

<b>ID</b>	UC-03		
<b>Tên</b>	Cập nhật hồ sơ		
<b>Mô tả</b>	Người dùng cập nhật thông tin hồ sơ cá nhân để đảm bảo dữ liệu mới nhất.		
<b>Actor chính</b>	SV, Tutor		
<b>Actor phụ</b>	Hệ thống, Admin (có quyền xem/sửa/kiểm tra Tutor profile) (PĐT, Khoa)		
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập hệ thống.		
<b>Hậu điều kiện</b>	Hồ sơ cập nhật thành công trong cơ sở dữ liệu.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Người dùng	Đăng nhập, chọn “Cập nhật hồ sơ”.
	2	Người dùng	Chỉnh sửa thông tin (SDT, email, chuyên ngành, mô tả năng lực...).
	3	Hệ thống	Kiểm tra định dạng dữ liệu.
	4	Hệ thống	Lưu thông tin mới vào DB.
	5	Hệ thống	Hiển thị thông báo “Cập nhật thành công”.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Người dùng	Bỏ trống trường bắt buộc (email, SDT) → Hệ thống báo lỗi, yêu cầu nhập lại.
	3a	Người dùng	Nhập email/SDT sai định dạng → Hệ thống báo lỗi, yêu cầu nhập lại.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu dữ liệu vào DB → hiển thị “Cập nhật thất bại”.		
<b>Business Rules</b>	Tutor phải cập nhật thông tin học thuật theo mẫu do PĐT quy định. SV không được sửa MSSV.		
<b>Data requirement</b>	Users(userID, email, phone, faculty, GPA, updatedAt) TutorProfile(tutorID, chuyên môn, kinh nghiệm, updatedAt)		

Bảng 5: Bảng đặc tả chức năng cập nhật hồ sơ

## 2.4.2. Module Đăng ký chương trình học

### Use Case 04: Đăng ký môn học

<b>ID</b>	UC-04		
<b>Tên</b>	Đăng ký môn học		
<b>Mô tả</b>	Sinh viên đăng ký môn học/lĩnh vực cần được Tutor hỗ trợ. Mỗi lần đăng ký chỉ chọn 1 môn học, có thể lặp lại quy trình nhiều lần (tối đa 4 môn).		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Hệ thống, Khoa/Bộ môn		
<b>Tiền điều kiện</b>	SV đã đăng ký tài khoản, hồ sơ hợp lệ. Hệ thống đã có danh mục môn học/lĩnh vực hỗ trợ.		
<b>Hậu điều kiện</b>	Thông tin chương trình học được lưu trong hồ sơ SV. SV có thể tiếp tục đăng ký thêm môn khác, miễn là tổng số $\leq 4$ . SV sẵn sàng cho bước ghép cặp với Tutor.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập hệ thống
	2	Sinh viên	Chọn chức năng “Đăng ký môn học”.
	3	Hệ thống	Hiển thị danh mục môn học khả dụng trong kỳ.
	4	Sinh viên	Chọn 1 môn học.
	5	Hệ thống	Kiểm tra SV đã đăng ký bao nhiêu môn? ( $\leq 3$ trước đó $\rightarrow$ cho phép; $= 4 \rightarrow$ từ chối).
	6	Hệ thống	Lưu thông tin đăng ký vào DB
	7	Hệ thống	Hiển thị thông báo “Đăng ký thành công”.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Hệ thống	Nếu danh sách môn học khả dụng trống (chưa có Tutor nào đăng ký dạy) $\rightarrow$ Hệ thống hiển thị “Hiện chưa có môn học nào khả dụng để đăng ký”.
	5a	Sinh viên	Nếu SV đã đăng ký đủ 4 môn $\rightarrow$ Hệ thống hiển thị “Chỉ được đăng ký tối đa 4 môn”.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu vào DB $\rightarrow$ hiển thị “Đăng ký thất bại, thử lại sau”.		
<b>Business Rules</b>	SV chỉ có thể đăng ký tối đa 4 môn học cùng lúc. Hệ thống chỉ hiển thị các môn học đã có ít nhất 1 Tutor đăng ký dạy. SV có thể hủy môn học đã đăng ký (UC-05) để giải phóng slot trước khi đăng ký mới. Danh mục môn học khả dụng được Khoa/BM cập nhật theo từng kỳ học.		
<b>Data requirement</b>	ProgramRegistration(regID, studentID, subjectID, purpose, regDate, status)		

Bảng 6: Bảng đặc tả chức năng đăng ký môn học



Use Case 05: Hủy đăng ký môn học

<b>ID</b>	UC-05		
<b>Tên</b>	Hủy đăng ký môn học		
<b>Mô tả</b>	Sinh viên hủy môn học đã đăng ký nếu không còn nhu cầu hoặc muốn đổi môn khác.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Hệ thống		
<b>Tiền điều kiện</b>	Môn học chưa bắt đầu và còn ít nhất 1 tuần trước khi bắt đầu buổi học đầu tiên		
<b>Hậu điều kiện</b>	Thông tin đăng ký được cập nhật (status = Cancelled) trong DB. Slot đăng ký được giải phóng, SV có thể đăng ký môn khác thay thế (UC-04).		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập hệ thống
	2	Sinh viên	Chọn chức năng “Hủy đăng ký môn học”.
	3	Hệ thống	Hiển thị danh sách môn học mà sinh viên đã đăng ký
	4	Sinh viên	Chọn 1 môn học muốn hủy
	5	Hệ thống	Yêu cầu xác nhận thao tác
	6	Sinh viên	Xác nhận hủy
	7	Hệ thống	Cập nhật Database, đổi trạng thái đăng ký thành "Cancelled"
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Hệ thống	Phát hiện SV chưa đăng ký môn nào → hiển thị “Không có môn để hủy”.
	5a	Sinh viên	Bấm “Không đồng ý” khi xác nhận hủy → Hệ thống quay lại danh sách môn đã đăng ký. ( quay lại bước 3.)
<b>Ngoại lệ</b>	Hệ thống lỗi khi cập nhật DB → hiển thị “Hủy thất bại, thử lại sau”.		
<b>Business Rules</b>	SV chỉ có thể hủy môn học chưa bắt đầu trước 1 tuần. Sau khi hủy, SV có thể đăng ký môn khác miễn tổng số môn $\leq 4$ (UC-04).		
<b>Data requirement</b>	ProgramRegistration(regID, studentID, subjectID, purpose, regDate, status) (cập nhật status = “Cancelled”).		

Bảng 7: Bảng đặc tả chức năng hủy đăng ký môn học

### 2.4.3. Module Ghép cặp Tutor – SV

#### Use Case 06: Ghép thủ công (SV chọn Tutor)

<b>ID</b>	UC-06		
<b>Tên</b>	Ghép thủ công (SV chọn Tutor cho từng môn học)		
<b>Mô tả</b>	Sinh viên tự chọn Tutor từ danh sách đề xuất để tham gia chương trình học.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Tutor, hệ thống		
<b>Tiền điều kiện</b>	Sinh viên đã đăng ký ít nhất một môn học trong chương trình (UC-04) Có ít nhất 1 Tutor phù hợp trong hệ thống cho môn đó		
<b>Hậu điều kiện</b>	Cặp SV – Tutor được lưu trong DB với môn học cụ thể. Một SV có thể có nhiều bản ghi ghép cho nhiều môn khác nhau. Trạng thái ghép = “Đang hoạt động”.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn chức năng “Chọn Tutor thủ công”.
	2	Hệ thống	Hiển thị danh sách các môn học SV đã đăng ký.
	3	Sinh viên	Chọn 1 môn học để tìm Tutor.
	4	Hệ thống	Hiển thị danh sách Tutor phù hợp cho môn học đó (theo lịch rảnh, slot còn trống, chuyên môn).
	5	Sinh viên	Chọn một Tutor từ danh sách.
	6	Hệ thống	Kiểm tra slot của Tutor (còn chỗ / chưa vượt maxSV).
	7	Hệ thống	Nếu hợp lệ → Hệ thống lưu kết quả ghép (SV – Tutor – Môn học) vào DB.
	8	Hệ thống	Hiển thị thông báo “Ghép thành công” cho SV.
	9	Hệ thống	Gửi thông báo đến Tutor (Tutor chỉ nhận thông báo, không được từ chối).
	10	Sinh viên	Có thể lặp lại quy trình cho các môn khác nếu muốn.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Hệ thống	Không tìm thấy Tutor phù hợp cho môn học đã chọn → hiển thị “Chưa có Tutor khả dụng, vui lòng thử lại sau”.
	6a	Hệ thống	Tutor đã full slot → hệ thống hiển thị “Tutor đã đủ số lượng SV, vui lòng chọn Tutor khác”
	7a	Hệ thống	Nếu SV không hài lòng → chọn chức năng “Hủy ghép” (UC-10) cho môn học đó, rồi quay lại UC-06 để chạy lại matching.
<b>Ngoại lệ</b>	Hệ thống lỗi kết nối khi gửi yêu cầu đến Tutor → hiển thị “Thao tác thất bại, thử lại sau”.		



<b>Business Rules</b>	Một SV có thể có nhiều Tutor khác nhau cho nhiều môn học khác nhau. Với mỗi môn học, tại một thời điểm chỉ có 1 Tutor chính (thủ công hoặc tự động). Tutor chỉ nhận tối đa số SV theo quy định (VD: $\leq 10$ ).
<b>Data requirement</b>	Matching(matchID,studentID,tutorID,subjectID,status,createdAt) subjectID đảm bảo SV có thể được ghép nhiều lần (mỗi môn một Tutor).

Bảng 8: Bảng đặc tả chức năng ghép thủ công (SV chọn Tutor cho từng môn học)



Use Case 07: Ghép tự động (Hệ thống đề xuất Tutor)

<b>ID</b>	UC-07		
<b>Tên</b>	Ghép tự động (hệ thống đề xuất Tutor theo từng môn học)		
<b>Mô tả</b>	Hệ thống tự động tìm và ghép Tutor phù hợp với SV cho từng môn học dựa trên lịch rảnh của Tutor và ưu tiên mà SV chọn (VD: lịch 2-4-6, buổi tối). Một SV có thể được ghép với nhiều Tutor khác nhau, mỗi Tutor phụ trách một môn riêng.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Tutor, hệ thống		
<b>Tiền điều kiện</b>	SV đã đăng ký một hoặc nhiều môn học trong chương trình (UC-04). Tutor đã cập nhật lịch rảnh và hồ sơ môn học (UC-08).		
<b>Hậu điều kiện</b>	SV được ghép thành công với ít nhất 1 Tutor cho môn học đã chọn. Nếu SV học nhiều môn, hệ thống có thể ghép nhiều Tutor khác nhau (mỗi môn 1 Tutor). Kết quả lưu vào DB. SV có thể hủy ghép với 1 môn cụ thể và thử lại mà không ảnh hưởng các môn khác.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn chức năng “Ghép Tutor tự động”.
	2	Hệ thống	Hiển thị danh sách các môn học SV đã đăng ký.
	3	Sinh viên	Chọn 1 môn để thực hiện ghép Tutor
	4	Hệ thống	Hiển thị form để SV chọn ưu tiên cho môn đó: Khung lịch mong muốn (VD: 2-4-6, 3-5-7, buổi sáng/chiều/tối).
	5	Sinh viên	Xác nhận lựa chọn.
	6	Hệ thống	Lấy danh sách Tutor khả dụng từ DB theo tiêu chí: Có lịch rảnh trùng với ưu tiên. Có chuyên môn đúng môn học SV vừa chọn. Còn slot trống.
	7	Hệ thống	Chạy thuật toán so khớp → chọn Tutor phù hợp nhất cho môn học đó.
	8	Hệ thống	Lưu thông tin ghép cặp vào DB với subjectID + tutorID.
	9	Hệ thống	Gửi thông báo kết quả ghép cho SV và Tutor.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	6a	Hệ thống	Không tìm thấy Tutor phù hợp cho môn học → hiển thị thông báo “Chưa có Tutor phù hợp, vui lòng thử lại sau”.
	9a	Hệ thống	Nếu SV không hài lòng → chọn chức năng “Hủy ghép” (UC-10) cho môn học đó, rồi quay lại UC-07 để chạy lại matching.



<b>Ngoại lệ</b>	Hệ thống lỗi kết nối khi gửi yêu cầu đến Tutor → hiển thị “Thao tác thất bại, thử lại sau”.
<b>Business Rules</b>	Một SV có thể được ghép với nhiều Tutor khác nhau cho các môn khác nhau. Với mỗi môn học, tại 1 thời điểm chỉ có 1 Tutor (manual hoặc auto). Một SV không được ghép song song 2 Tutor cho cùng một môn (trừ khi hủy ghép trước đó). Tutor không thể vượt quá số lượng SV tối đa cho môn mình phụ trách. SV chỉ được chọn lịch nằm trong slot rảnh do Tutor khai báo.
<b>Data requirement</b>	Matching(matchID,studentID,tutorID,subjectID,method,status,createdAt ,preferences) method = "manual" hoặc "auto". subjectID cho phép 1 SV có nhiều bản ghi ghép (mỗi môn 1 bản ghi, mỗi bản ghi có 1 Tutor khác nhau).

Bảng 9: Bảng đặc tả chức năng ghép tự động (hệ thống đề xuất Tutor theo từng môn học)

## 2.4.4. Module Quản lý lịch học

### Use Case 08: Tạo lịch rảnh (Tutor)

ID	UC-08		
Tên	Tạo lịch rảnh (Tutor)		
Mô tả	Tutor khai báo thời gian rảnh để sinh viên có thể đặt lịch học.		
Actor chính	Tutor		
Actor phụ	Hệ thống		
Tiền điều kiện	Tutor đã đăng ký tài khoản và hồ sơ hợp lệ.		
Hậu điều kiện	Lịch rảnh được lưu trong DB và hiển thị cho SV lựa chọn.		
Luồng sự kiện	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Tutor	Đăng nhập, chọn chức năng “Tạo lịch rảnh”.
	2	Tutor	Nhập ngày, giờ, hình thức học (online/offline).
	3	Hệ thống	Kiểm tra định dạng dữ liệu và phát hiện trùng lặp lịch.
	4	Hệ thống	Lưu lịch rảnh vào DB.
Luồng thay thế	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Tutor	Bỏ trống ngày/giờ → Hệ thống yêu cầu nhập lại.
	3a	Hệ thống	Phát hiện lịch rảnh đã tồn tại → hiển thị lỗi, không lưu.
Ngoại lệ	Hệ thống lỗi DB khi lưu → hiển thị “Tạo lịch thất bại, thử lại sau”.		
Business Rules	Tutor chỉ có thể tạo lịch rảnh trong khung giờ hợp lệ của trường (ví dụ 7h – 21h).		
Data requirement	Availability(availlID, tutorID, date, startTime, endTime, mode, status)		

Bảng 10: Bảng đặc tả chức năng tạo lịch rảnh (Tutor)





### Use Case 09: Đặt lịch học (SV)

<b>ID</b>	UC-09		
<b>Tên</b>	Đặt lịch học (SV)		
<b>Mô tả</b>	Sinh viên đặt lịch học cố định với Tutor đã được ghép cho môn học. Lịch này được áp dụng xuyên suốt môn học cho đến khi kết thúc, không cần đăng ký lại từng buổi.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Hệ thống		
<b>Tiền điều kiện</b>	SV đã đăng ký chương trình học và đã được ghép Tutor cho môn học (UC-06 hoặc UC-07). Tutor đã khai báo lịch rảnh (UC-08).		
<b>Hậu điều kiện</b>	Một lịch học cố định được lưu vào DB, gắn với cặp SV – Tutor – Môn học. Lịch này được lặp lại tự động trong suốt thời gian môn học diễn ra.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn chức năng “Đặt lịch học”
	2	Hệ thống	Hiển thị danh sách môn học mà SV đã được ghép Tutor.
	3	Sinh viên	Chọn môn học.
	4	Hệ thống	Hiển thị danh sách slot rảnh của Tutor.
	5	Sinh viên	Chọn 1 slot cố định (ngày trong tuần + khung giờ) để học.
	6	Hệ thống	Kiểm tra: Slot còn chỗ ( $< \text{maxSV}/\text{slot}$ ). SV không có lịch học khác trùng giờ.
	7	Hệ thống	Lưu lịch cố định này vào DB (gắn với môn học + Tutor + SV).
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	4a	Hệ thống	Không có slot rảnh phù hợp → báo “Chưa có lịch khả dụng”.
	6a	Hệ thống	Slot đã full → báo “Lịch không khả dụng, vui lòng chọn slot khác”.
	6b	Hệ thống	SV có lịch học khác trùng giờ → báo lỗi, yêu cầu chọn lại.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu DB → hiển thị “Đặt lịch thất bại, thử lại sau”.		
<b>Business Rules</b>	Mỗi môn học, SV chỉ được đặt 1 lịch học cố định với Tutor. Lịch học cố định sẽ được lặp lại hàng tuần cho đến khi kết thúc môn. SV phải đặt lịch trước ít nhất 1 tuần trước buổi học đầu tiên. Tutor có thể dạy nhiều SV trong cùng slot, nhưng không vượt quá giới hạn maxSV.		
<b>Data requirement</b>	Schedule(scheduleID, tutorID, studentID, subjectID, date, startTime, endTime, status)		

Bảng 11: Bảng đặc tả chức năng đặt lịch học (SV)



### Use Case 10: Hủy/Đổi lịch học cố định

<b>ID</b>	UC-10		
<b>Tên</b>	Hủy/Đổi lịch học cố định		
<b>Mô tả</b>	Sinh viên có thể hủy hoặc đổi lịch học cố định của một môn đã đăng ký. Lịch học áp dụng xuyên suốt môn, nên hủy/đổi sẽ ảnh hưởng toàn bộ các buổi còn lại của môn đó. Tutor không được quyền hủy.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Tutor (chỉ nhận thông báo), Hệ thống		
<b>Tiền điều kiện</b>	SV đã đăng ký môn học và đã đặt lịch cố định (UC-09). Môn học chưa kết thúc.		
<b>Hậu điều kiện</b>	Lịch học được hủy hoặc thay đổi, thông tin được cập nhật trong DB. SV có thể đặt lại lịch mới (UC-09) nếu muốn.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn chức năng “Hủy/Đổi lịch học”.
	2	Hệ thống	hiển thị danh sách môn học mà SV đã có lịch cố định.
	3	Sinh viên	Chọn 1 môn học
	4	Sinh viên	Nếu hủy: Hệ thống cập nhật trạng thái lịch = “Cancelled” cho toàn bộ lịch cố định của môn đó. Hệ thống gửi thông báo đến Tutor và SV.
	5	Sinh viên	Nếu đổi: Hệ thống hiển thị danh sách rảnh khác của Tutor. SV chọn 1 slot mới. Hệ thống kiểm tra: Slot chưa full. Không trùng với lịch học khác của SV. Nếu hợp lệ → Hệ thống cập nhật slot mới cho toàn bộ môn học. Hệ thống gửi thông báo cho cả SV và Tutor.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Sinh viên	SV chọn môn học đã kết thúc → Hệ thống báo “Không thể hủy/đổi vì môn đã kết thúc”.
	5a	Sinh viên	SV chọn môn mới đã full hoặc trùng lịch → Hệ thống báo lỗi, yêu cầu chọn slot khác.
<b>Ngoại lệ</b>	Hệ thống lỗi DB khi cập nhật → hiển thị “Thao tác thất bại, thử lại sau”.		
<b>Business Rules</b>	SV chỉ được hủy/đổi lịch cố định trước ít nhất 1 tuần so với buổi học kế tiếp. Tutor không được quyền hủy lịch. Khi đổi lịch, toàn bộ buổi học trong môn đó sẽ chuyển sang slot mới (không đổi từng buổi riêng lẻ).		
<b>Data requirement</b>	Schedule(scheduleID, tutorID, studentID, subjectID, dayOfWeek, startTime, endTime, recurrence, status, updatedAt) recurrence = weekly (lặp hằng tuần). status = Active / Cancelled		

Bảng 12: Bảng đặc tả chức năng hủy/đổi lịch học cố định



## 2.4.5. Module Thông báo và nhắc nhở

### Use Case 11: Gửi thông báo lịch học

<b>ID</b>	UC-11		
<b>Tên</b>	Gửi thông báo lịch học		
<b>Mô tả</b>	Hệ thống gửi thông báo đến SV và Tutor khi có lịch học mới được đặt hoặc thay đổi.		
<b>Actor chính</b>	Hệ thống		
<b>Actor phụ</b>	Sinh viên, Tutor		
<b>Tiền điều kiện</b>	Lịch học đã được tạo hoặc có thay đổi (UC-09, UC-10). SV và Tutor đã kích hoạt tài khoản.		
<b>Hậu điều kiện</b>	SV và Tutor nhận được thông báo về lịch học.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Hệ thống	Phát hiện có lịch học mới/hủy/đổi.
	2	Hệ thống	Tạo thông báo gồm: ngày, giờ, môn học, trạng thái (mới/hủy/đổi).
	3	Hệ thống	Gửi thông báo qua email/app cho SV và Tutor.
	4	Người dùng (SV/Tutor)	Nhận và đọc thông báo.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	4a	Người dùng (SV/Tutor)	chưa mở app → Hệ thống lưu thông báo trong mục “Thông báo” để xem sau.
<b>Ngoại lệ</b>	Hệ thống lỗi khi gửi email → chỉ lưu thông báo trong app.		
<b>Business Rules</b>	Tất cả thay đổi về lịch học phải được thông báo trong vòng 5 phút kể từ khi cập nhật.		
	Nội dung thông báo phải có đủ thông tin: môn, ngày, giờ, hình thức học.		
<b>Data requirement</b>	Notification(notifID, userID, type, content, createdAt, status)		

Bảng 13: Bảng đặc tả chức năng gửi thông báo lịch học



#### Use Case 12: Gửi nhắc nhở buổi học

<b>ID</b>	UC-12		
<b>Tên</b>	Gửi thông báo lịch học		
<b>Mô tả</b>	Hệ thống gửi nhắc nhở cho SV và Tutor trước khi buổi học bắt đầu.		
<b>Actor chính</b>	Hệ thống		
<b>Actor phụ</b>	Sinh viên, Tutor		
<b>Tiền điều kiện</b>	Lịch học đã được xác nhận (UC-09). Buổi học chưa diễn ra.		
<b>Hậu điều kiện</b>	SV và Tutor nhận được nhắc nhở kịp thời.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Hệ thống	Quét DB tìm lịch học sắp diễn ra.
	2	Hệ thống	Tạo nhắc nhở (ngày, giờ, môn học, phòng học /link online).
	3	Hệ thống	Gửi nhắc nhở cho SV và Tutor.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu buổi học đã bị hủy → Hệ thống không gửi nhắc nhở.
	4a	Hệ thống	Nếu SV/Tutor tắt thông báo → Hệ thống chỉ lưu nhắc nhở trong mục “Thông báo” trên app.
<b>Ngoại lệ</b>	Hệ thống lỗi server vào thời điểm gửi → nhắc nhở không được gửi.		
<b>Business Rules</b>	Nhắc nhở gửi 2 lần: 24h trước và 1h trước buổi học. Người dùng có thể tùy chỉnh kênh nhận nhắc nhở (email, dashboard).		
<b>Data requirement</b>	Reminder(reminderID, scheduleID, userID, type, sendTime, status)		

Bảng 14: Bảng đặc tả chức năng gửi nhắc nhở buổi học

## 2.4.6. Module Quản lý buổi học và điểm danh

### Use Case 13: Điểm danh sinh viên

<b>ID</b>	UC-13		
<b>Tên</b>	Điểm danh sinh viên		
<b>Mô tả</b>	Tutor điểm danh sự tham gia của sinh viên trong buổi học.		
<b>Actor chính</b>	Sinh viên, Tutor		
<b>Actor phụ</b>	Hệ thống		
<b>Tiền điều kiện</b>	Buổi học đã được xác nhận và diễn ra đúng lịch. SV và Tutor đã đăng nhập hệ thống.		
<b>Hậu điều kiện</b>	Trạng thái điểm danh của SV được lưu trong DB.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Tutor	Đăng nhập và mở danh sách buổi học trong ngày.
	2	Tutor	Chọn buổi học cần điểm danh.
	3	Hệ thống	Hiển thị danh sách SV đã đăng ký buổi học.
	4	Tutor	Đánh dấu SV có mặt/vắng mặt./đi trễ
	5	Hệ thống	Lưu kết quả điểm danh vào DB.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Hệ thống	Nếu buổi học không có SV nào đăng ký → Hệ thống hiển thị “Không có SV tham gia”.
<b>Ngoại lệ</b>	DB lỗi khi lưu → hiển thị “Điểm danh thất bại, thử lại sau		
<b>Business Rules</b>	Chỉ Tutor phụ trách buổi học mới có quyền điểm danh. SV vắng mặt >2 lần liên tiếp → hệ thống thông báo cho Khoa. Điểm danh chỉ được thực hiện trong khoảng thời gian buổi học diễn ra. Tutor không thể thay đổi điểm danh sau khi buổi học kết thúc (trừ khi có xác nhận từ Khoa).		
<b>Data requirement</b>	Attendance(attID, scheduleID, studentID, status, attTime)		

Bảng 15: Bảng đặc tả chức năng điểm danh sinh viên

#### Use Case 14: Cập nhật trạng thái buổi học

<b>ID</b>	UC-14		
<b>Tên</b>	Cập nhật trạng thái buổi học		
<b>Mô tả</b>	Tutor hoặc hệ thống cập nhật trạng thái buổi học (chưa diễn ra, đang diễn ra, đã hoàn tất, bị hủy).		
<b>Actor chính</b>	Tutor, Hệ thống		
<b>Actor phụ</b>	Sinh viên		
<b>Tiền điều kiện</b>	Buổi học đã được xác nhận trong hệ thống.		
<b>Hậu điều kiện</b>	Trạng thái buổi học được cập nhật chính xác trong DB.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Hệ thống	Trước giờ học, gán trạng thái buổi học là “Sắp diễn ra”.
	2	Tutor	Chọn “Bắt đầu buổi học” → hệ thống cập nhật trạng thái = “Đang diễn ra”.
	3	Tutor	Chọn “Hoàn tất buổi học” → hệ thống cập nhật trạng thái = “Đã hoàn tất”.
	4	Hệ thống	Lưu log thay đổi trạng thái vào DB.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Tutor	Nếu Tutor quên cập nhật → hệ thống tự động đổi trạng thái sang “Đã hoàn tất” sau khi quá giờ học.
	3a	Tutor	Nếu buổi học bị hủy trước khi bắt đầu → Hệ thống cập nhật trạng thái = “Đã hủy”.
<b>Ngoại lệ</b>	Hệ thống lỗi khi cập nhật DB → hiển thị “Thao tác thất bại, thử lại sau”.		
<b>Business Rules</b>	Trạng thái buổi học chỉ có thể thay đổi theo luồng hợp lệ: “Sắp diễn ra” → “Đang diễn ra” → “Đã hoàn tất” hoặc “Sắp diễn ra” → “Đã hủy” SV chỉ xem được trạng thái, không được chỉnh sửa. Nếu buổi học bị hủy, Hệ thống phải gửi thông báo ngay cho SV. Trạng thái buổi học phải được cập nhật trong vòng 24h sau khi kết thúc.		
<b>Data requirement</b>	Session(sessionID, tutorID, studentID, subjectID, date, startTime, endTime, status, updatedAt)		

Bảng 16: Bảng đặc tả chức năng cập nhật trạng thái buổi học

## 2.4.7. Module Quản lý tài liệu học tập

### Use Case 15: Quản lý tài liệu (Tutor)

<b>ID</b>	UC-15		
<b>Tên</b>	Quản lý tài liệu (Tutor)		
<b>Mô tả</b>	Tutor quản lý tài liệu học tập (slide, đề cương, bài tập) cho môn học hoặc buổi học: upload, chỉnh sửa, xóa.		
<b>Actor chính</b>	Tutor		
<b>Actor phụ</b>	Sinh viên, Hệ thống		
<b>Tiền điều kiện</b>	Tutor đã đăng nhập hệ thống. Buổi học/chương trình học đã được xác nhận.		
<b>Hậu điều kiện</b>	Tài liệu được lưu trữ trong hệ thống và hiển thị cho SV liên quan.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Tutor	Đăng nhập và chọn “Quản lý tài liệu”.
	2	Tutor	Chọn môn học/buổi học liên quan.
	3	Tutor	Chọn thao tác: Upload: chọn file (PDF, PPT, DOCX), nhập mô tả → Hệ thống kiểm tra định dạng, dung lượng → lưu file + metadata → hiển thị “Upload thành công”. Chỉnh sửa: chọn tài liệu → cập nhật tên/mô tả/file mới → Hệ thống kiểm tra hợp lệ → cập nhật DB → hiển thị “Cập nhật thành công”. Xóa: chọn tài liệu → xác nhận → Hệ thống xóa bản ghi và file → hiển thị “Xóa thành công”.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Tutor	Không nhập mô tả khi upload → Hệ thống nhắc nhập lại.
	3b	Tutor	File trùng tên → yêu cầu đổi tên
	3c	Tutor	File sai định dạng → báo lỗi.
	3d	Tutor	File vượt dung lượng tối đa (ví dụ >50MB) → từ chối upload.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu file/DB → hiển thị “Thao tác thất bại, thử lại sau”.		
<b>Business Rules</b>	Tài liệu phải gắn với một môn học hoặc buổi học cụ thể. Tutor chỉ được chỉnh sửa/xóa tài liệu do chính mình upload.		
<b>Data requirement</b>	Materials(materialID, tutorID, subjectID, sessionID, fileName, filePath, fileSize, description, uploadedAt, updatedAt)		

Bảng 17: Bảng đặc tả chức năng quản lý tài liệu (Tutor)



Use Case 16: Sinh viên tải tài liệu

<b>ID</b>	UC-16		
<b>Tên</b>	SV tải tài liệu		
<b>Mô tả</b>	Sinh viên tải tài liệu học tập do Tutor chia sẻ để phục vụ học tập.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Tutor, Hệ thống		
<b>Tiền điều kiện</b>	SV đã đăng nhập hệ thống. Tutor đã upload ít nhất một tài liệu cho môn học. (UC-15).		
<b>Hậu điều kiện</b>	SV tải xuống và sử dụng tài liệu học tập.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn môn học.
	2	Hệ thống	Hiển thị danh sách tài liệu của môn học.
	3	Sinh viên	Chọn file muốn tải.
	4	Hệ thống	Kiểm tra quyền truy cập. (SV thuộc lớp/buổi học đó).
	5	Hệ thống	Gửi file để SV tải về.
	6	Sinh viên	Nhận và lưu file vào máy.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu chưa có tài liệu → Hệ thống hiển thị “Chưa có tài liệu được chia sẻ”.
	4a	Hệ thống	Nếu SV không thuộc buổi học → Hệ thống hiển thị “Bạn không có quyền tải tài liệu này”.
<b>Ngoại lệ</b>	Hệ thống lỗi server khi gửi file → hiển thị “Tải xuống thất bại”.		
<b>Business Rules</b>	SV chỉ tải tài liệu thuộc môn học mình đã đăng ký. Hệ thống ghi log mỗi lần SV tải tài liệu.		
<b>Data requirement</b>	DownloadLog(logID, studentID, materialID, downloadTime, status)		

Bảng 18: Bảng đặc tả chức năng sinh viên tải tài liệu



## 2.4.8. Module Đánh giá và phản hồi

### Use Case 17: Sinh viên đánh giá Tutor

<b>ID</b>	UC-17		
<b>Tên</b>	SV đánh giá Tutor		
<b>Mô tả</b>	Sau khi môn học kết thúc, sinh viên có thể đánh giá chất lượng Tutor để phản hồi về hiệu quả hỗ trợ.		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Tutor, Hệ thống, Khoa/BM		
<b>Tiền điều kiện</b>	Môn học đã hoàn tất. SV đã tham gia (có điểm danh $\geq 1$ buổi trong môn).		
<b>Hậu điều kiện</b>	Feedback được lưu vào hệ thống và gắn với hồ sơ Tutor.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập và chọn “Đánh giá Tutor”.
	2	Hệ thống	Hiển thị danh sách môn học đã hoàn tất.
	3	Sinh viên	Chọn môn học muốn đánh giá.
	4	Sinh viên	Nhập rating (1–5, bắt buộc) và nhận xét (tùy chọn).
	5	Hệ thống	Lưu đánh giá vào DB.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu không có môn học nào đã hoàn tất → hiển thị “Chưa có môn học để đánh giá”.
	4a	Sinh viên	Nếu SV bỏ trống rating → hệ thống nhắc nhập.
<b>Ngoại lệ</b>	Lỗi khi lưu DB → hiển thị “Đánh giá thất bại, thử lại sau”.		
<b>Business Rules</b>	Mỗi SV chỉ được đánh giá 1 lần cho mỗi môn. Rating bắt buộc, nhận xét không bắt buộc. Feedback SV gửi → Tutor & Khoa đều xem được. Không cho sửa sau khi gửi.		
<b>Data requirement</b>	Feedback(feedbackID, fromUserID, toUserID, subjectID, roleFrom, roleTo, rating, comment, createdAt) (roleFrom = "SV", roleTo = "Tutor")		

Bảng 19: Bảng đặc tả chức năng sinh viên đánh giá Tutor



#### Use Case 18: Tutor đánh giá sinh viên

<b>ID</b>	UC-18		
<b>Tên</b>	Tutor đánh giá sinh viên		
<b>Mô tả</b>	Sau khi kết thúc môn học, Tutor có thể đánh giá mức độ tham gia, thái độ học tập của SV để phản hồi cho Khoa/BM.		
<b>Actor chính</b>	Tutor		
<b>Actor phụ</b>	SV, Hệ thống, Khoa/BM		
<b>Tiền điều kiện</b>	Môn học đã hoàn tất.SV đã tham gia ít nhất 1 buổi.		
<b>Hậu điều kiện</b>	Feedback của Tutor được lưu trong DB, gắn với hồ sơ SV. Khoa/BM có thể xem dữ liệu này trong báo cáo tổng hợp.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Tutor	Đăng nhập, chọn “Đánh giá sinh viên”.
	2	Hệ thống	Hiển thị danh sách SV đã dạy.
	3	Tutor	Chọn SV cần đánh giá.
	4	Tutor	Nhập: Rating (1–5, bắt buộc) Nhận xét (tùy chọn) Tiêu chí (chuyên cần, thái độ, hợp tác–tùy chọn)
	5	Hệ thống	Lưu dữ liệu vào DB.
	6	Hệ thống	Hiển thị “Đánh giá thành công”.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu chưa có SV nào đã học xong → hiển thị “Không có sinh viên để đánh giá”.
	4a	Tutor	Nếu Tutor bỏ trống rating → hệ thống nhắc nhập.
<b>Ngoại lệ</b>	Lỗi DB → hiển thị “Đánh giá thất bại, thử lại sau”.		
<b>Business Rules</b>	Mỗi SV chỉ được Tutor đánh giá 1 lần/môn. Rating bắt buộc, nhận xét/tiêu chí không bắt buộc. Feedback của Tutor chỉ hiển thị cho Khoa/BM (không gửi cho SV). Dùng cho mục đích quản lý chất lượng.		
<b>Data requirement</b>	Feedback(feedbackID, fromUserID, toUserID, subjectID, roleFrom, roleTo, rating, comment, criteria, createdAt) (roleFrom = "Tutor", roleTo = "SV")		

Bảng 20: Bảng đặc tả chức năng Tutor đánh giá sinh viên



Use Case 19: Khoa/BM tổng hợp đánh giá

<b>ID</b>	UC-19		
<b>Tên</b>	Khoa/BM tổng hợp đánh giá		
<b>Mô tả</b>	Khoa/BM có thể xem báo cáo tổng hợp về đánh giá hai chiều: SV đánh giá Tutor và Tutor đánh giá SV		
<b>Actor chính</b>	Khoa/BM		
<b>Actor phụ</b>	Hệ thống, Tutor, SV		
<b>Tiền điều kiện</b>	Người dùng có quyền Khoa/BM hoặc Admin.		
<b>Hậu điều kiện</b>	Báo cáo tổng hợp được hiển thị trên giao diện hoặc xuất file.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Khoa/BM	Đăng nhập hệ thống.
	2	Khoa/BM	Chọn chức năng “Xem báo cáo đánh giá”.
	3	Hệ thống	Truy xuất dữ liệu feedback từ DB (theo từng Tutor và SV).
	4	Hệ thống	Tính toán các chỉ số: Đối với Tutor: điểm trung bình, số lượt đánh giá, % SV hài lòng. Đối với SV: điểm trung bình, số môn đã học, tiêu chí chuyên cần/thái độ/hợp tác.
	5	Hệ thống	Hiển thị báo cáo dạng bảng/biểu đồ.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	3a	Hệ thống	Nếu chưa có dữ liệu đánh giá (VD Tutor chưa từng dạy, SV chưa từng học) → Hệ thống hiển thị “Chưa có dữ liệu để báo cáo”.
<b>Ngoại lệ</b>	Hệ thống lỗi khi truy vấn DB → hiển thị “Không thể lấy dữ liệu, thử lại sau”		
<b>Business Rules</b>	Chỉ Khoa/BM hoặc Admin mới được xem báo cáo. Báo cáo phải hiển thị cả 2 chiều: Đánh giá Tutor (từ SV) và đánh giá SV (từ Tutor). Báo cáo cập nhật dữ liệu theo thời gian thực.		
<b>Data requirement</b>	TutorReport(tutorID, avgRating, totalFeedback, positiveRate, generatedAt) StudentReport(studentID, avgRating, totalFeedback, criteriaStats, generatedAt)		

Bảng 21: Bảng đặc tả chức năng Khoa/BM tổng hợp đánh giá

## 2.4.9. Module Thống kê và báo cáo

### Use Case 20: Báo cáo kết quả học tập SV

<b>ID</b>	UC-20		
<b>Tên</b>	Báo cáo kết quả học tập SV		
<b>Mô tả</b>	Khoa tổng hợp và xác nhận kết quả học tập của SV sau khi môn học kết thúc, sau đó hệ thống hiển thị cho SV.		
<b>Actor chính</b>	Khoa		
<b>Actor phụ</b>	Hệ thống, SV		
<b>Tiền điều kiện</b>	SV đã tham gia ít nhất 1 buổi học có điểm danh. Dữ liệu điểm danh, trạng thái buổi học, đánh giá Tutor đã được cập nhật.		
<b>Hậu điều kiện</b>	Báo cáo đã được xác nhận bởi Khoa và hiển thị cho SV.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Khoa	Đăng nhập, chọn “Tổng hợp kết quả học tập SV”.
	2	Hệ thống	Lấy dữ liệu điểm danh, trạng thái buổi học và đánh giá Tutor.
	3	Hệ thống	Tạo báo cáo sơ bộ: số buổi đăng ký, số buổi tham gia, số buổi vắng, % tham gia, phản hồi từ Tutor.
	4	Khoa	Kiểm tra và xác nhận báo cáo.
	5	Hệ thống	Hiển thị báo cáo cho SV và cho phép SV xuất file (PDF/Excel).
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu SV chưa tham gia buổi học nào → Hệ thống hiển thị “Chưa có dữ liệu để báo cáo”.
<b>Ngoại lệ</b>	Lỗi khi truy xuất DB → hiển thị “Không thể tạo báo cáo”.		
<b>Business Rules</b>	SV chỉ xem được báo cáo của chính mình. Khoa có quyền xem & xác nhận báo cáo cho SV trong phạm vi quản lý. Báo cáo phải bao gồm: số buổi đăng ký, số buổi tham gia, số buổi vắng, % tham gia.		
<b>Data requirement</b>	Attendance(attID, scheduleID, studentID, status) Session(sessionID, status) StudentReport(studentID, totalSessions, attendedSessions, attendanceRate, lastUpdated, approvedBy, approvedAt)		

Bảng 22: Bảng đặc tả chức năng báo cáo kết quả học tập sinh viên



### Use Case 21: Báo cáo chất lượng Tutor

<b>ID</b>	UC-21		
<b>Tên</b>	Báo cáo chất lượng Tutor		
<b>Mô tả</b>	Khoa tổng hợp và xác nhận báo cáo chất lượng giảng dạy của Tutor dựa trên feedback từ SV và dữ liệu buổi học.		
<b>Actor chính</b>	Khoa		
<b>Actor phụ</b>	Hệ thống, Tutor		
<b>Tiền điều kiện</b>	Tutor đã có ít nhất 1 buổi học hoàn tất. Có feedback từ SV.		
<b>Hậu điều kiện</b>	Báo cáo chất lượng được xác nhận và hiển thị cho Tutor.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Khoa	Đăng nhập, chọn “Báo cáo chất lượng Tutor”.
	2	Hệ thống	Lấy dữ liệu feedback, điểm danh, trạng thái buổi học.
	3	Hệ thống	Tổng hợp báo cáo sơ bộ (điểm trung bình, số SV đánh giá, tỷ lệ buổi thành công).
	4	Khoa	Kiểm tra và xác nhận báo cáo.
	5	Hệ thống	Hiển thị báo cáo cho Tutor và cho phép SV xuất file (PDF/Excel).
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu không có feedback → Hệ thống hiển thị “Chưa có dữ liệu đánh giá”.
<b>Ngoại lệ</b>	Lỗi khi truy xuất DB → hiển thị “Không thể tạo báo cáo”		
<b>Business Rules</b>	Tutor chỉ xem được báo cáo của chính mình. Khoa có quyền xem báo cáo của tất cả Tutor trong khoa. Báo cáo phải có trạng thái xác nhận trước khi công bố cho Tutor.		
<b>Data requirement</b>	Feedback(feedbackID, tutorID, rating, comment) TutorReport(tutorID, avgRating, totalFeedback, successRate, approvedBy, approvedAt, generatedAt)		

Bảng 23: Bảng đặc tả chức năng báo cáo chất lượng Tutor

### Use Case 22: Báo cáo tổng hợp (Khoa, PCTSV, PDT)

<b>ID</b>	UC-22		
<b>Tên</b>	Báo cáo tổng hợp (Khoa, PCTSV, PDT)		
<b>Mô tả</b>	Khoa, PCTSV và PDT có thể xem báo cáo tổng hợp toàn hệ thống sau khi dữ liệu từ các báo cáo con (SV, Tutor) đã được xác nhận.		
<b>Actor chính</b>	Khoa / PCTSV / PDT		
<b>Actor phụ</b>	Hệ thống		
<b>Tiền điều kiện</b>	Các báo cáo học tập SV (UC-20) và chất lượng Tutor (UC-21) đã được xác nhận. Người dùng có quyền xem báo cáo tổng hợp.		
<b>Hậu điều kiện</b>	Báo cáo tổng hợp hiển thị cho cấp quản lý, có thể xuất file.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Người dùng (Khoa/PCTSV/PDT)	Đăng nhập, chọn “Báo cáo tổng hợp”.
	2	Hệ thống	Lấy dữ liệu từ các báo cáo đã xác nhận (UC-20, UC-21).
	3	Hệ thống	Tổng hợp theo phạm vi: Khoa: theo môn/Tutor/SV trong khoa. PCTSV: tình hình hỗ trợ SV toàn trường. PDT: hiệu quả chương trình, đề xuất cải tiến.
	4	Hệ thống	Hiển thị báo cáo dạng bảng và biểu đồ.
	5	Người dùng	Có thể export file Excel/PDF.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Hệ thống	Nếu một số báo cáo chưa xác nhận → Hệ thống hiển thị cảnh báo nhưng vẫn hiển thị dữ liệu đã có.
<b>Ngoại lệ</b>	Lỗi khi truy xuất DB lớn → hiển thị “Không thể tạo báo cáo, thử lại sau”.		
<b>Business Rules</b>	Khoa chỉ thấy dữ liệu thuộc khoa. PCTSV/PDT thấy dữ liệu toàn trường. Báo cáo tổng hợp được cập nhật theo kỳ (tuần/tháng) sau khi dữ liệu đã được xác nhận.		
<b>Data requirement</b>	StudentReport(reportID, studentID, subjectID, totalSessions, attendedSessions, absentSessions, attendanceRate, tutorFeedback, lastUpdated, approvedBy, approvedAt) TutorReport(reportID, tutorID, totalSessions, completedSessions, cancelledSessions, avgRating, totalFeedback, positiveRate, successRate, lastUpdated, approvedBy, approvedAt ) SystemReport(reportID, type, scope, approvedBy, approvedAt, generatedAt, filePath)		

Bảng 24: Bảng đặc tả chức năng báo cáo tổng hợp (Khoa, PCTSV, PDT)

## 2.4.10. Module Chương trình học thuật và phi học thuật

### Use Case 23: Tutor tạo chương trình học

<b>ID</b>	UC-23		
<b>Tên</b>	Tutor tạo chương trình học		
<b>Mô tả</b>	Tutor tạo và công bố chương trình học mới (học thuật hoặc phi học thuật) để SV đăng ký.		
<b>Actor chính</b>	Tutor		
<b>Actor phụ</b>	Hệ thống, SV		
<b>Tiền điều kiện</b>	Tutor đã đăng nhập và được xác minh		
<b>Hậu điều kiện</b>	Chương trình học được hiển thị cho SV.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Tutor	Đăng nhập, chọn “Tạo chương trình học”.
	2	Tutor	Nhập thông tin chương trình: loại (học thuật/phi học thuật), tên, mô tả, thời lượng.
	3	Hệ thống	Kiểm tra thông tin hợp lệ
	4	Hệ thống	Lưu và hiển thị chương trình trên hệ thống.
	5	Sinh viên	Có thể thấy chương trình để đăng ký.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	2a	Tutor	Bỏ trống thông tin bắt buộc → Hệ thống nhắc nhập lại.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu → hiển thị “Tạo chương trình thất bại”.		
<b>Business Rules</b>	Tutor chỉ có thể tạo chương trình sau khi được Khoa phê duyệt. Mỗi chương trình phải thuộc một loại: học thuật hoặc phi học thuật.		
<b>Data requirement</b>	Program(programID, tutorID, type, name, description, duration, slots, createdAt, approvedBy, approvedAt, status)		

Bảng 25: Bảng đặc tả chức năng Tutor tạo chương trình học



Use Case 24: SV đăng ký chương trình học thuật

<b>ID</b>	UC-24		
<b>Tên</b>	SV đăng ký chương trình học thuật		
<b>Mô tả</b>	SV đăng ký tham gia chương trình học thuật (ôn tập, bổ trợ kiến thức, luyện thi).		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Hệ thống, Tutor		
<b>Tiền điều kiện</b>	Tutor đã tạo chương trình học thuật. SV đã có tài khoản hợp lệ.		
<b>Hậu điều kiện</b>	SV được thêm vào danh sách tham gia của chương trình.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn “Đăng ký chương trình học thuật”.
	2	Hệ thống	Hiển thị danh sách chương trình học thuật khả dụng.
	3	Sinh viên	Chọn chương trình muốn tham gia
	4	Hệ thống	Kiểm tra slot còn trống.
	5	Hệ thống	Lưu thông tin đăng ký và thêm SV vào danh sách.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	4a	Hệ thống	Phát hiện chương trình hết slot → hiển thị “Chương trình đã đủ người”.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu đăng ký → hiển thị “Thao tác thất bại, thử lại sau”.		
<b>Business Rules</b>	Mỗi SV có thể tham gia tối đa 2 chương trình học thuật cùng lúc.		
<b>Data requirement</b>	ProgramRegistration(regID, programID, studentID, registeredAt, status) Liên kết với Program (type = học thuật).		

Bảng 26: Bảng đặc tả chức năng SV đăng ký chương trình học thuật



### Use Case 25: SV đăng ký chương trình phi học thuật

<b>ID</b>	UC-25		
<b>Tên</b>	SV đăng ký chương trình phi học thuật		
<b>Mô tả</b>	SV đăng ký tham gia chương trình phi học thuật (kỹ năng mềm, hoạt động ngoại khóa).		
<b>Actor chính</b>	Sinh viên		
<b>Actor phụ</b>	Hệ thống, Tutor		
<b>Tiền điều kiện</b>	Tutor đã tạo chương trình phi học thuật. SV đã có tài khoản hợp lệ.		
<b>Hậu điều kiện</b>	SV được thêm vào danh sách tham gia của chương trình.		
<b>Luồng sự kiện</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	1	Sinh viên	Đăng nhập, chọn “Đăng ký chương trình phi học thuật”.
	2	Hệ thống	Hiển thị danh sách chương trình phi học thuật khả dụng.
	3	Sinh viên	Chọn chương trình muốn tham gia
	4	Hệ thống	Kiểm tra slot còn trống.
	5	Hệ thống	Lưu thông tin đăng ký và thêm SV vào danh sách.
<b>Luồng thay thế</b>	<b>Bước</b>	<b>Thực hiện bởi</b>	<b>Mô tả</b>
	4a	Hệ thống	Phát hiện chương trình hết slot → hiển thị “Chương trình đã đủ người”.
<b>Ngoại lệ</b>	Hệ thống lỗi khi lưu đăng ký → hiển thị “Thao tác thất bại, thử lại sau”.		
<b>Business Rules</b>	SV có thể tham gia nhiều chương trình phi học thuật, tùy vào lịch học.		
<b>Data requirement</b>	ProgramRegistration(regID, programID, studentID, registeredAt, status) Liên kết với Program (type = phi học thuật).		

Bảng 27: Bảng đặc tả chức năng SV đăng ký chương trình phi học thuật

## 2.5. Yêu cầu phi chức năng

Để xây dựng một hệ thống kết nối Tutor và Sinh viên thực sự hiệu quả và đáng tin cậy, việc đáp ứng các yêu cầu về chức năng là chưa đủ. Yếu tố quyết định trải nghiệm người dùng và sự thành công lâu dài của dự án nằm ở các Yêu cầu phi chức năng (Non-Functional Requirements). Các tiêu chí này đặt ra những chuẩn mực về tốc độ, bảo mật, độ ổn định và tính dễ sử dụng của hệ thống. Những ràng buộc và tiêu chuẩn dưới đây sẽ là những yếu tố để đảm bảo hệ thống không chỉ hoàn thiện mà còn mang lại sự hài lòng và tin tưởng tuyệt đối cho mọi người dùng, từ sinh viên, Tutor đến các cấp quản lý.

### 2.5.1. Hiệu năng (Performance Requirements)

- **Mô tả:** Hệ thống phải xử lý nhanh và ổn định cho nhiều người dùng đồng thời.
- **Constraints:**
  - Hỗ trợ tối thiểu 500 người dùng đồng thời.

- Thời gian phản hồi cho thao tác chính  $\leq 3$  giây.
- Thuật toán ghép cặp Tutor–SV chạy trong  $\leq 5$  giây.
- **Acceptance:** Kiểm thử tải (load test) cho thấy hệ thống đáp ứng  $\geq 95\%$  request trong 3 giây

### 2.5.2. Bảo mật (Security Requirements)

- **Mô tả:** Bảo vệ thông tin người dùng và dữ liệu hệ thống khỏi truy cập trái phép.
- **Constraints:**
  - Mã hóa toàn bộ giao tiếp bằng HTTPS (TLS 1.3).
  - Lưu mật khẩu bằng bcrypt/Argon2.
  - Xác thực 2FA áp dụng cho Tutor và Admin.
  - Khóa tài khoản sau 5 lần nhập sai mật khẩu.
  - Phân quyền theo role (SV, Tutor, Khoa, PCTSV, PDT, Admin).
- **Acceptance:** Thử nghiệm penetration test không phát hiện lỗ hổng nghiêm trọng.

### 2.5.3. Tính tin cậy & sẵn sàng (Reliability & Availability)

- **Mô tả:** Hệ thống phải đảm bảo tính liên tục và phục hồi khi có sự cố.
- **Constraints:**
  - Thời gian uptime  $\geq 99.5\%$ /tháng.
  - Backup dữ liệu hàng ngày, phục hồi  $\leq 2h$ .
  - Retry khi gửi thông báo thất bại.
  - Log toàn bộ giao dịch quan trọng.
- **Acceptance:** DRP (Disaster Recovery Plan) kiểm thử thành công, phục hồi dữ liệu  $\leq 2h$ .

### 2.5.4. Khả năng sử dụng (Usability)

- **Mô tả:** Giao diện thân thiện, dễ sử dụng cho tất cả loại người dùng.
- **Constraints:**
  - Hỗ trợ đa thiết bị (desktop, mobile, tablet).
  - Ngôn ngữ: Tiếng Việt (mặc định), Tiếng Anh (tùy chọn).
  - Người dùng mới có thể đăng ký, đặt lịch trong  $\leq 5$  phút.
  - Có màn hình trợ giúp/hướng dẫn.
- **Acceptance:** Khảo sát  $\geq 80\%$  người dùng đánh giá giao diện “dễ sử dụng”.

### 2.5.5. Tính bảo trì & mở rộng (Maintainability & Extensibility)

- **Mô tả:** Hệ thống dễ bảo trì, nâng cấp mà không ảnh hưởng đến chức năng hiện có.
- **Constraints:**
  - Tuân thủ mô hình MVC hoặc Microservices.
  - Code phải có comment, tuân thủ coding convention.
  - Thêm module mới không ảnh hưởng module cũ.
  - Bug critical sau release  $\leq 2\%$ .
- **Acceptance:** Regression test cho thấy chức năng cũ không bị ảnh hưởng sau khi thêm module mới.

### 2.5.6. Khả năng tương thích (Compatibility)

- **Mô tả:** Hệ thống chạy được trên nhiều nền tảng và dịch vụ tích hợp.
- **Constraints:**
  - Web chạy trên Chrome, Firefox, Edge, Safari (phiên bản mới nhất).
  - Mobile app chạy trên Android  $\geq 10$ , iOS  $\geq 13$ .
  - Tích hợp email server.
- **Acceptance:** Test cross-browser cho kết quả hiển thị đúng  $\geq 95\%$ .

### 2.5.7. Ràng buộc kỹ thuật (Technical Constraints)

- **Mô tả:** Các công nghệ, công cụ và nền tảng bắt buộc sử dụng.
- **Constraints:**
  - DB: MySQL hoặc PostgreSQL.
  - Backend: Java Spring Boot hoặc Node.js.
  - Frontend: ReactJS hoặc Angular.
  - Triển khai trên Docker/Kubernetes.
- **Acceptance:** Cấu hình hệ thống triển khai thành công trên môi trường staging/production.