

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
1. Tổng quan dự án	6
1.1. Giới thiệu dự án	6
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
1.2. Phạm vi dự án	8
1.2.1. Trong phạm vi	8
1.2.2. Ngoài phạm vi	8
1.2.3. Ràng buộc hệ thống	9
1.3. Tài liệu tham khảo liên quan	9
1.3.1. Các nền tảng tương tự	9
1.3.2. Bài học rút ra	10
1.3.3. Tính năng tổng hợp cho hệ thống	10
2. Phân tích yêu cầu dự án	12
2.1. Tác nhân	12
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
2.2. Sơ đồ usecase toàn hệ thống	17
2.3. User Stories	17
2.4. Yêu cầu chức năng	17



2.5. Yêu cầu phi chức năng	17
2.5.1. Hiệu năng (Performance Requirements)	17
2.5.2. Hiệu năng (Performance Requirements)	18
2.5.3. Tính tin cậy & sẵn sàng (Reliability & Availability)	18
2.5.4. Khả năng sử dụng (Usability)	18
2.5.5. Tính bảo trì & mở rộng (Maintainability & Extensibility)	18
2.5.6. Khả năng tương thích (Compatibility)	19
2.5.7. Ràng buộc kỹ thuật (Technical Constraints)	19
3. Mô hình hóa hệ thống	20
3.1. Sơ đồ hành vi	20
3.2. Sơ đồ lớp hệ thống	20
3.3. Các sơ đồ khác	20
4. Kết luận	21
4.1. Tổng kết	21
4.2. Hạn chế hiện tại	21
4.3. Định hướng phát triển	21



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	SĐT	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	2FA	Two-Factor Authentication (Xác thực hai yếu tố)



Danh sách Hình ảnh

Danh sách Bảng

1	Danh sách Từ viết tắt	3
2	Danh sách thành viên & khối lượng công việc	5



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.

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_DATACORE.

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 ≥ 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 ≥ 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 ≥ 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 ≤ 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

2.3. User Stories

2.4. Yêu cầu chức năng

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 ≤ 3 giây.
 - Thuật toán ghép cặp Tutor–SV chạy trong ≤ 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, PĐT, 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 ≤ 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 ≥ 10 , iOS ≥ 13 .
 - Tích hợp email server và SMS gateway.
- **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.



3. Mô hình hóa hệ thống

3.1. Sơ đồ hành vi

3.2. Sơ đồ lớp hệ thống

3.3. Các sơ đồ khác



4. Kết luận

4.1. Tổng kết

4.2. Hạn chế hiện tại

4.3. Định hướng phát triển