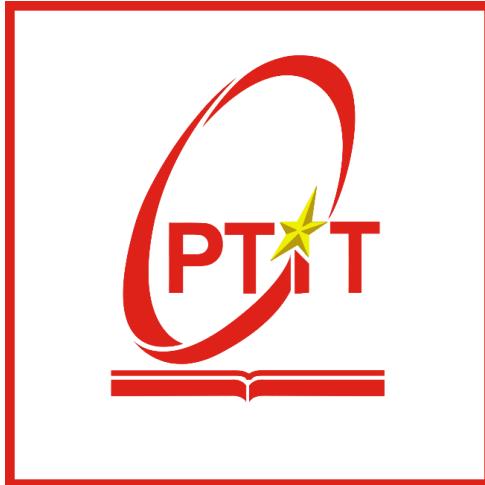


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

-----000-----



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

Đề tài: Phát triển hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực tích hợp Generative-based Chatbot và kết nối dịch vụ tri liệu

Giáo viên hướng dẫn : TS. Đỗ Thị Liên

Họ và tên sinh viên : Nguyễn Phúc Sơn

Mã sinh viên : B20DCCN581

Lớp : D20CNPM03

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM (CỦA GIÁO VIÊN PHẢN BIỆN)

Điểm:(Bằng chữ.....)

Hà nội, ngày tháng năm 2025

CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong suốt thời gian từ khi bắt đầu học tập ở Học viện đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô trong Học viện nói chung và đặt biệt là quý Thầy Cô khoa Công nghệ thông tin I nói riêng đã cùng với tri thức và sự tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em. Những kiến thức tiếp thu trong quá trình học tập không chỉ là nền tảng cho quá trình phát triển chuyên môn mà còn là hành trang quý báu để chúng em bước vào cuộc sống một cách vững chắc và tự tin hơn.

Em xin chân thành cảm ơn cô TS. **Đỗ Thị Liên**. Cô đã tận tâm hướng dẫn em trong suốt quá trình làm phát triển đồ án của mình. Nhờ vậy mà em đã hoàn thành đề tài một cách tốt nhất. Một lần nữa em xin chân thành cảm ơn cô.

Mặc dù đã có nhiều cố gắng để thực hiện đề tài một cách hoàn chỉnh nhất. Song do kiến thức cũng như kinh nghiệm của em còn nhiều hạn chế và bỡ ngỡ. Do vậy chắc chắn sẽ không thể tránh khỏi những thiếu sót mà bản thân chưa thấy được. Em rất mong được sự góp ý từ quý Thầy Cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện đồ án

Nguyễn Phúc Sơn

Danh mục từ viết tắt

Số thứ tự	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	WHO	World Health Organization
2	COVID-19	Coronavirus disease 2019
3	AI	Artificial intelligence
4	ML	Machine learning
5	GPT	Generative Pre-trained Transformer
6	ASP.NET	Active Server Pages Network Enabled Technologies
7	API	Application programming interface
8	ORM	Object-Relational Mapper
9	EF Core	Entity Framework Core
10	LINQ	Language integrated query
11	SQL	Structure query language
12	DDD	Domain Driven Design
13	UI	User interface
14	REST	Representational State Transfer
15	HTTP	Hypertext Transfer Protocol
16	JSON	JavaScript Object Notation
17	XML	Extensible Markup Language
18	JWT	Json Web Token
19	SDK	Software development kit
20	IDE	Integrated development environment
21	CLI	Command Line Interface

Danh mục hình ảnh

Hình 1.1 Luồng xử lý khi trò chuyện với Chatbot	12
Hình 2.1 Sơ đồ mô tả kiến trúc và luồng gọi của hệ thống	15
Hình 2.2 Sơ đồ giải thích ý tưởng của Clean Architecture [6]	19
Hình 2.3 Mô hình kiến trúc các tầng trong ứng dụng backend theo clean architecture	20
Hình 2.4 Giao diện thử nghiệm mô hình Google AI Studio	25
Hình 2.5 Sơ đồ triển khai dự án.....	25
Hình 2.6 Use case đăng nhập tài khoản người dùng	28
Hình 2.7 Use case xem bài đăng	29
Hình 2.8 Use case quản lý tài khoản cá nhân	32
Hình 2.9 Use case đăng ký phiên trị liệu cá nhân	35
Hình 2.10 Use case theo dõi phiên trị liệu công khai.....	37
Hình 2.11 Use case trò chuyện với chatbot	39
Hình 2.12 Use case quản lý các phiên trị liệu công khai cá nhân	41
Hình 2.13 Sơ đồ lớp thực thể và thuộc tính trích xuất	46
Hình 2.14 Sơ đồ tuần tự use case đăng nhập.....	47
Hình 2.15 Sơ đồ tuần tự use case đăng ký	48
Hình 2.16 Sơ đồ tuần tự use case xem bài đăng.....	50
Hình 2.17: Sơ đồ tuần tự use case cập nhật thông tin cá nhân	51
Hình 2.18 Sơ đồ tuần tự use case cập nhật mật khẩu	52
Hình 2.19 Sơ đồ tuần tự use case xoá tài khoản.....	53
Hình 2.20 Sơ đồ tuần tự use case đánh giá người trị liệu.....	53
Hình 2.21 Sơ đồ tuần tự use case đăng ký phiên trị liệu cá nhân.....	54
Hình 2.22 Sơ đồ tuần tự use case xem lịch các phiên công khai/cá nhân trên hệ thống	55
Hình 2.23 Sơ đồ tuần tự use case theo dõi phiên công khai.....	55
Hình 2.24 Sơ đồ tuần tự use case quản lý thông báo cá nhân	56
Hình 2.25 Sơ đồ tuần tự use case trò chuyện với chatbot	57
Hình 2.26 Sơ đồ tuần tự use case chat người với trị liệu	58
Hình 2.27 Sơ đồ tuần tự use case chat với người đăng ký trị liệu.....	59
Hình 2.28 Sơ đồ tuần tự use case quản lý phiên công khai cá nhân	60
Hình 2.29 Sơ đồ tuần tự use case quản lý đăng ký phiên trị liệu cá nhân.....	61
Hình 2.30 Sơ đồ tuần tự use case quản lý lịch các phiên trị liệu cá nhân	62
Hình 2.31 Sơ đồ tuần tự use case quản lý lịch làm việc cá nhân	63
Hình 2.32 Sơ đồ tuần tự use case quản lý lịch làm việc riêng	64
Hình 2.33 Biểu đồ lớp thực thể	65
Hình 2.34 Sơ đồ thiết kế cơ sở dữ liệu cho dự án	74
Hình 3.1 Giao diện nạp dữ liệu trên Google AI Studio.....	78
Hình 3.2 Biểu đồ loss/epochs xuyên suốt quá trình huấn luyện.....	79
Hình 3.3 Giao diện đăng nhập ứng dụng.....	80
Hình 3.4 Giao diện đăng ký tài khoản.....	81
Hình 3.5 Giao diện bài đăng.....	81
Hình 3.6 Giao diện chỉnh sửa thông tin cá nhân	82

Hình 3.7 Giao diện thay đổi mật khẩu	82
Hình 3.8 Giao diện xoá tài khoản	82
Hình 3.9 Giao diện lịch sử đăng ký và đánh giá người trị liệu	82
Hình 3.10 Giao diện tìm kiếm người trị liệu	83
Hình 3.11 Giao diện thông tin chi tiết người trị liệu và đăng ký	84
Hình 3.12 Hộp thoại đăng ký phiên trị liệu cá nhân với người trị liệu	84
Hình 3.13 Giao diện lịch hẹn với người trị liệu và phiên công khai đã theo dõi	85
Hình 3.14 Giao diện phiên công khai	85
Hình 3.15 Giao diện thông báo cá nhân	86
Hình 3.16 Giao diện trò chuyện với chatbot	86
Hình 3.17 Giao diện trò chuyện giữa người trị liệu và người đăng ký	87
Hình 3.18 Giao diện quản lý phiên công khai của người trị liệu	87
Hình 3.19 Giao diện thêm/chỉnh sửa phiên công khai	87
Hình 3.20 Giao diện quản lý các đơn đăng ký (người trị liệu)	88
Hình 3.21 Giao diện quản lý phiên trị liệu cá nhân với người dùng	88
Hình 3.22 Giao diện quản lý thời gian làm việc trong tuần	89
Hình 3.23 Giao diện quản lý lịch đặc biệt (lịch ghi đè)	89
Hình 4.1 Giao diện thống kê lượt tải xuống web app Angular	95
Hình 4.2 Giao diện giám sát hệ thống backend trên Azure Application Insight.....	95

Danh mục bảng

Bảng 2.1 Đặc tả use case đăng nhập tài khoản.....	29
Bảng 2.2 Đặc tả use case đăng ký tài khoản	30
Bảng 2.3 Đặc tả use case xem bài đăng.....	31
Bảng 2.4 Đặc tả use case quản lý thông tin tài khoản	32
Bảng 2.5 Đặc tả use case đăng ký phiên trị liệu cá nhân	35
Bảng 2.6 Đặc tả use case xem lịch các phiên công khai/cá nhân.....	36
Bảng 2.7 Đặc tả use case theo dõi phiên công khai	37
Bảng 2.8 Đặc tả use case quản lý thông báo cá nhân	38
Bảng 2.9 Đặc tả use case trò chuyện với chatbot	39
Bảng 2.10 Đặc tả use case trò chuyện với người trị liệu	39
Bảng 2.11 Đặc tả use case quản lý phiên trị liệu công khai cá nhân.....	41
Bảng 2.12 Đặc tả use case quản lý đăng ký phiên trị liệu cá nhân	42
Bảng 2.13 Đặc tả use case quản lý lịch phiên trị liệu cá nhân	43
Bảng 2.14 Đặc tả use case quản lý lịch làm việc cá nhân	43

DANH MỤC

CHƯƠNG 1: Tổng quan về hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực	7
1.1. Xác định yêu cầu cho hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực	7
1.2. Khảo sát các hệ thống hỗ trợ chăm sóc sức khỏe hiện nay	7
1.3. Ứng dụng công nghệ Generative-based Chatbot trong trò chuyện tương tác	8
1.4. Kết nối dịch vụ trí liệu và hỗ trợ từ chuyên gia.....	10
1.5. Đè xuất tích hợp Generative-based Chatbot và kết nối dịch vụ trí liệu vào hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực.....	11
1.5.1. Xử lý hội thoại	12
1.5.2. Phân tích tag và kết nối dịch vụ	13
1.6. Kết luận chương 1	13
CHƯƠNG 2: Phân tích và thiết kế hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực	15
2.1. Mô hình tổng quát hệ thống.....	15
2.1.1. Angular	16
2.1.2. Backend ASP.NET Core Web API	17
2.1.3. Gemini API.....	24
2.1.4. Triển khai ứng dụng	25
2.2. Phân tích hệ thống	27
2.2.1. Yêu cầu của hệ thống bằng ngôn ngữ tự nhiên	27
2.2.2. Sơ đồ use case tổng quan hệ thống.....	28
2.2.3. Các đặc tả use case	29
2.2.4. Biểu đồ lớp phân tích.....	44
2.3. Thiết kế hệ thống	47
2.3.1. Sơ đồ tuần tự.....	47
2.3.2. Biểu đồ lớp thiết kế	65
2.3.3. Cơ sở dữ liệu.....	74
2.3.4. Biểu đồ gói	75
2.4. Kết luận chương 2	76
CHƯƠNG 3: Thủ nghiệm và đánh giá hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực tích hợp Generative-based Chatbot và kết nối dịch vụ trí liệu	77
3.1. Dữ liệu thực nghiệm generative-based chatbot	77
3.2. Cài đặt thực nghiệm generative-based chatbot.....	78
3.3. Kết quả thực nghiệm generative-based chatbot.....	79

3.4. Thủ nghiệm hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực tích hợp Generative-based Chatbot và kết nối dịch vụ trí liệu	80
3.4.1. Giao diện ứng dụng	80
3.4.2. Generative-based AI.....	89
3.5. Kết luận chương 3	92
Phụ lục: cài đặt và triển khai	93
1. Môi trường phát triển.....	93
2. Quy trình cài đặt cho phát triển	93
3. Quy trình triển khai.....	93
4. Giám sát hệ thống	95
Kết luận và hướng phát triển	96

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG HỖ TRỢ CHĂM SÓC SỨC KHỎE TINH THẦN THỜI GIAN THỰC

Chương này giới thiệu tổng quan về hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực, tập trung xác định yêu cầu hệ thống, khảo sát các giải pháp hiện có, và phân tích tiềm năng ứng dụng công nghệ Generative-based Chatbot. Đồng thời, chương nhấn mạnh vai trò kết nối với chuyên gia trị liệu và đề xuất tích hợp chatbot cùng các dịch vụ trị liệu, nhằm xây dựng một hệ thống hỗ trợ hiệu quả và toàn diện.

1.1. Xác định yêu cầu cho hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực

Trong bối cảnh hiện nay, sức khỏe tinh thần đang ngày càng trở thành một vấn đề được xã hội quan tâm sâu sắc. Theo Tổ chức Y tế Thế giới (WHO), tỷ lệ người gặp vấn đề về tâm lý như lo âu, trầm cảm, và căng thẳng ngày càng gia tăng. Trong năm đầu tiên của đại dịch COVID-19, tỷ lệ lo âu và trầm cảm trên toàn cầu đã tăng lên 25% [1]. Sức khỏe tinh thần không chỉ ảnh hưởng đến chất lượng cuộc sống mà còn có tác động tiêu cực lên hiệu quả làm việc và các mối quan hệ xã hội. Tâm lý yếu còn có thể làm giảm tuổi thọ và là tác nhân cho các bệnh truyền nhiễm cho con người [2]. Tuy nhiên, việc tiếp cận các dịch vụ tư vấn và trị liệu tâm lý vẫn gặp nhiều hạn chế do sự thiếu hụt chuyên gia, chi phí cao và khoảng cách địa lý.

Vì vậy, việc phát triển một hệ thống hỗ trợ chăm sóc sức khỏe tinh thần trực tuyến là cần thiết để cung cấp một giải pháp hiệu quả, tiết kiệm thời gian và chi phí, đồng thời đảm bảo khả năng tiếp cận dịch vụ cho nhiều người hơn. Hệ thống này cần hỗ trợ tương tác thời gian thực với người dùng, cung cấp các giải pháp trò chuyện tư vấn từ chatbot ứng dụng công nghệ Generative AI (AI sinh tạo) và kết nối người dùng với các chuyên gia trị liệu khi cần thiết.

Ứng dụng công nghệ Generative-based Chatbot vào hệ thống này sẽ đóng vai trò quan trọng trong việc cung cấp sự hỗ trợ liên tục, giúp người dùng có thể nhận được tư vấn và phân tích tâm lý cơ bản bất cứ lúc nào. Hơn nữa, chatbot có thể sử dụng các kỹ thuật học máy để phân tích trạng thái tâm lý của người dùng, đưa ra các lời khuyên phù hợp, hoặc kết nối người dùng với các chuyên gia tâm lý khi cần thiết.

Việc kết nối trực tiếp với dịch vụ trị liệu giúp tạo ra một môi trường hỗ trợ toàn diện, giúp người dùng không chỉ nhận được lời khuyên từ chatbot mà còn có thể nhận được sự can thiệp chuyên sâu từ các nhà trị liệu hoặc các phương pháp cấp cứu khẩn cấp khi trạng thái tâm lý trở nên nghiêm trọng hơn.

1.2. Khảo sát các hệ thống hỗ trợ chăm sóc sức khỏe hiện nay

Hiện nay, cùng với sự phát triển của công nghệ, nhiều hệ thống hỗ trợ chăm sóc sức khỏe tinh thần đã được triển khai để phục vụ nhu cầu ngày càng cao của người dùng. Những hệ thống này tập trung vào việc cung cấp các dịch vụ tư vấn tâm lý, trị liệu trực tuyến, và sử dụng các công nghệ tiên tiến như AI để tối ưu hóa trải nghiệm người dùng. Dưới đây là một số loại hệ thống chính và các giải pháp công nghệ điển hình:

1. Hệ thống tư vấn và trị liệu trực tuyến: Một số nền tảng tiêu biểu như BetterHelp [3], Talkspace [4] đã và đang phổ biến rộng rãi nhờ vào khả năng tiếp cận dễ dàng và tiết kiệm chi phí so với các dịch vụ trị liệu truyền thống. Những hệ

thống này giúp người dùng có thể trò chuyện với chuyên gia vào bất cứ lúc nào, từ bất cứ đâu, mà không phải lo lắng về khoảng cách địa lý. Các nền tảng này hỗ trợ đầy đủ các tính năng

2. **Ứng dụng chatbot hỗ trợ sức khỏe tinh thần:** CassAI và Replika là các ứng dụng tiêu biểu sử dụng trí tuệ nhân tạo để tương tác với người dùng, cung cấp các bài tập tâm lý học nhận thức và hành vi, giúp quản lý lo âu và trầm cảm. Các chatbot này không chỉ giúp người dùng thư giãn mà còn có khả năng tự động phân tích các trạng thái tâm lý cơ bản và đưa ra gợi ý phù hợp.
3. **Nền tảng cộng đồng và mạng xã hội hỗ trợ sức khỏe tinh thần:** Sanvello hay Happify là một số hệ thống xây dựng các cộng đồng trực tuyến, nơi người dùng có thể chia sẻ câu chuyện cá nhân, nhận sự đồng cảm và hỗ trợ từ những người có cùng trải nghiệm. Những nền tảng này không chỉ cung cấp các nội dung tư vấn sức khỏe tinh thần mà còn giúp người dùng kết nối với những cộng đồng hỗ trợ, chia sẻ kiến thức và kinh nghiệm trong việc vượt qua các vấn đề tâm lý.
4. **Ứng dụng công nghệ di động và theo dõi sức khỏe tinh thần:** Ngoài việc cung cấp các dịch vụ tư vấn và trị liệu, một số hệ thống còn tích hợp công nghệ theo dõi sức khỏe tinh thần qua các ứng dụng di động. Moodfit hay Calm cho phép người dùng theo dõi tâm trạng, mức độ lo âu và chất lượng giấc ngủ, từ đó phân tích dữ liệu và đưa ra các lời khuyên giúp người dùng cải thiện sức khỏe tinh thần một cách hiệu quả.

Mặc dù những hệ thống hiện có đã mang lại nhiều lợi ích đáng kể, vẫn tồn tại một số hạn chế cần được giải quyết. Nhiều ứng dụng chatbot vẫn chưa có khả năng nhận diện các vấn đề khủng hoảng tâm lý nghiêm trọng và cần cải thiện độ chính xác trong việc đưa ra phản hồi. Đối với các ứng dụng trị liệu trực tuyến đơn thuần luôn yêu cầu thanh toán trước, tạo nên một trải nghiệm người dùng và xây dựng cộng đồng chưa tốt. Ngoài ra, việc phụ thuộc quá nhiều vào công nghệ có thể dẫn đến tình trạng người dùng trở nên cô lập, thiếu sự kết nối thực sự với cộng đồng hay chuyên gia.

1.3. Ứng dụng công nghệ Generative-based Chatbot trong trò chuyện tương tác

Bắt đầu vào năm 1966 với ELIZA của Joseph Weizenbaum, các chatbot cơ bản sử dụng thuật toán dựa trên quy tắc để nhận diện từ khóa trong câu hỏi của người dùng và cung cấp câu trả lời dựa trên những phản hồi đã được lập trình sẵn. Tuy nhiên, do hạn chế trong xử lý ngôn ngữ tự nhiên, các chatbot này thường gặp khó khăn khi đối phó với ngôn ngữ phức tạp. [5]

Từ những năm 2010, sự phát triển của học máy (ML) giúp tạo ra các tác nhân hội thoại có khả năng xử lý ngôn ngữ tự nhiên chính xác hơn. Các hệ thống này có thể học từ tương tác trước đó, hiểu lệnh thoại và thực hiện các nhiệm vụ, ví dụ như Siri hay Alexa. [5]

Vào cuối những năm 2010 tới nay, Generative-based Chatbot là một thế hệ chatbot tiên tiến, ứng dụng mô hình AI sinh (Generative AI), đặc biệt là các mô hình ngôn ngữ

lớn như GPT (Generative Pre-trained Transformer) của OpenAI và mô hình Gemini của Google. Công nghệ này có khả năng không chỉ trả lời câu hỏi mà còn tạo ra các đoạn hội thoại tự nhiên, có ngữ cảnh và phù hợp với nhu cầu của người dùng.

Tiếp theo sự phát triển của các mô hình ngôn ngữ lớn, việc tích hợp hiệu quả vào hệ thống AI đòi hỏi sự tập trung vào kỹ thuật "prompt engineering" - kỹ thuật thiết kế và tối ưu hóa prompt (đầu vào) để điều khiển mô hình tạo ra phản hồi mong muốn. Việc thiết kế prompt hiệu quả đóng vai trò then chốt trong việc khai thác tối đa tiềm năng của các mô hình này trong ứng dụng chatbot.

Một hệ thống chatbot hiện đại thường được xây dựng dựa trên cơ chế "multi-turn chat", cho phép mô hình theo dõi lịch sử hội thoại và sử dụng thông tin này làm ngữ cảnh cho các phản hồi tiếp theo. Điều này giúp tạo ra các cuộc trò chuyện tự nhiên và liền mạch hơn, mô phỏng trải nghiệm giao tiếp giữa người với người.

Thiết kế prompt cho chatbot bao gồm các thành phần chính:

- Messages (bắt buộc) là nội dung cốt lõi của prompt, bao gồm các cặp tin nhắn giữa người dùng ("user") và chatbot ("model"). Mô hình sẽ dựa vào tin nhắn gần nhất của người dùng và lịch sử hội thoại để tạo phản hồi. Giới hạn token của mô hình sẽ quyết định số lượng tin nhắn được lưu giữ trong ngữ cảnh. Ví dụ:

```
"contents": [
  {
    "role": "user",
    "parts": { "text": "Chào bạn!" }
  },
  {
    "role": "model",
    "parts": { "text": "Chào bạn! Tôi có thể giúp gì cho bạn?" }
  },
  {
    "role": "user",
    "parts": { "text": "Tôi muốn tìm hiểu về sản phẩm X." }
  }
]
```

- Context (ngữ cảnh) cung cấp thông tin bổ sung để tùy chỉnh hành vi của chatbot, ví dụ như thiết lập nhân dạng, quy tắc phản hồi, hoặc cung cấp kiến thức nền. Một số kỹ thuật tốt bao gồm: xác định rõ vai trò và tính cách của chatbot, đặt ra các quy tắc cụ thể về nội dung và phong cách trả lời, và hạn chế việc tiết lộ thông tin context. Ví dụ:

```
"context": "Bạn là một trợ lý ảo chuyên tư vấn về du lịch. Bạn luôn thân thiện và nhiệt tình. Bạn không được tiết lộ thông tin cá nhân của khách hàng."
```

Bên cạnh đó, một kỹ thuật quan trọng khác là fine-tuning (tinh chỉnh), giúp mô hình ngôn ngữ lớn học tốt hơn cho các nhiệm vụ cụ thể. Fine-tuning tận dụng khả năng nhận dạng mẫu mạnh mẽ của các mô hình nền tảng bằng cách huấn luyện bổ sung trên một tập dữ liệu nhỏ, thường chỉ vài trăm đến vài nghìn mẫu huấn luyện. Quá trình này giúp cải thiện đáng kể khả năng dự đoán của mô hình trên các tác vụ cụ thể, mặc dù số lượng tham số của mô hình sau khi fine-tuning vẫn giữ nguyên như mô hình ban đầu.

Ngoài ra, các kĩ thuật tinh chỉnh tối ưu như parameter-efficient tuning cũng được áp dụng để giảm chi phí tính toán mà vẫn đảm bảo hiệu quả.

Generative-based Chatbot là một công nghệ tiên tiến với tiềm năng lớn trong việc hỗ trợ chăm sóc sức khỏe tinh thần thông qua các cuộc trò chuyện tương tác tự nhiên và thân thiện. Tuy nhiên, để tối ưu hóa khả năng hỗ trợ, cần có sự kết hợp giữa công nghệ và chuyên gia trị liệu, đảm bảo người dùng nhận được sự hỗ trợ phù hợp, đặc biệt trong các tình huống khẩn cấp.

1.4. Kết nối dịch vụ trị liệu và hỗ trợ từ chuyên gia

Trong bối cảnh công nghệ phát triển mạnh mẽ, các công cụ AI như chatbot có khả năng hỗ trợ người dùng một cách nhanh chóng, tiện lợi, và liên tục. Tuy nhiên, khi nói đến trị liệu tâm lý chuyên sâu và hỗ trợ sức khỏe tinh thần, vai trò của chuyên gia trị liệu vẫn không thể thay thế. Các công cụ AI có thể hỗ trợ người dùng trong việc xử lý những tình huống cơ bản hoặc phổ biến, nhưng chuyên gia trị liệu mới là người có đủ khả năng phân tích sâu sắc và can thiệp kịp thời vào các vấn đề phức tạp về tâm lý. Các tầm quan trọng của nhất của người trị liệu có thể kể đến là:

- Hiểu sâu về tâm lý học: Chuyên gia trị liệu được đào tạo chuyên sâu về tâm lý học và có kinh nghiệm trong việc phân tích các trạng thái cảm xúc, hành vi của con người. Họ có khả năng nhận biết các dấu hiệu tiềm ẩn của vấn đề tâm lý nghiêm trọng như trầm cảm, lo âu, hoặc rối loạn tâm thần, điều mà các công cụ AI hiện nay chưa thể làm tốt.
- Xử lý các tình huống khẩn cấp: Trong các trường hợp khủng hoảng tâm lý hoặc nguy cơ tự tử, chuyên gia trị liệu có khả năng can thiệp ngay lập tức và đưa ra những phương án giải quyết an toàn và phù hợp. Trong khi đó, AI vẫn còn gặp nhiều hạn chế trong việc xác định chính xác các tình huống khẩn cấp và đưa ra phản ứng phù hợp cho người sử dụng.
- Cá nhân hóa liệu pháp trị liệu: Chuyên gia trị liệu không chỉ dựa vào dữ liệu hành vi mà còn sử dụng cảm nhận và kinh nghiệm của họ để xây dựng các lộ trình trị liệu cá nhân hóa cho từng trường hợp. Họ có thể điều chỉnh phương pháp trị liệu dựa trên phản hồi của người dùng, yếu tố văn hóa, và các mối quan hệ xã hội, điều mà AI khó có thể đạt được một cách hoàn hảo.
- Sự đồng cảm và kết nối con người: Mặc dù AI có thể mô phỏng các cuộc trò chuyện tương tác, nhưng không thể mang lại sự đồng cảm chân thực mà chỉ có con người mới có. Sự tương tác với chuyên gia trị liệu giúp người dùng cảm thấy được lắng nghe, được thấu hiểu và tin tưởng hơn. Đây là yếu tố quan trọng trong việc hình thành mối quan hệ trị liệu, điều kiện tiên quyết cho một quá trình chữa trị thành công.

Các nền tảng web hỗ trợ trị liệu tâm lý trực tuyến là một giải pháp phổ biến giúp người dùng dễ dàng kết nối với các chuyên gia trị liệu mọi lúc, mọi nơi với các chức năng nâng cao trải nghiệm của hai phía khách hàng và người trị liệu:

- Đăng ký và xác thực tài khoản: Người dùng và chuyên gia trị liệu đều cần đăng ký và xác thực tài khoản trên nền tảng. Để đảm bảo tính bảo mật và riêng tư, quá trình đăng ký thường yêu cầu các thông tin cá nhân
- Chọn chuyên gia và lên lịch trị liệu: Người dùng có thể tìm kiếm và chọn chuyên gia trị liệu phù hợp dựa trên các yếu tố như chuyên môn, kinh nghiệm, và đánh giá từ người dùng khác. Nền tảng thường cung cấp một hồ sơ chi tiết về từng chuyên gia, cho phép người dùng xem trước các thông tin như bằng cấp, lĩnh vực chuyên môn, và phương pháp trị liệu mà chuyên gia áp dụng. Sau khi chọn và đăng ký với chuyên gia, cả 2 bên có thể quản lý các buổi đặt lịch giúp tối ưu thời gian và hạn chế xung đột lịch
- Tương tác trực tuyến qua video hoặc chat: Một trong những chức năng chính của nền tảng là cung cấp các phương tiện tương tác trực tuyến giữa người dùng và chuyên gia trị liệu. Tùy theo nhu cầu và khả năng tiếp cận, nền tảng có thể hỗ trợ nhiều hình thức tương tác, phổ biến nhất là video call và chat trực tiếp.
- Theo dõi và ghi nhận tiến trình: Nền tảng web cung cấp công cụ giúp chuyên gia và người dùng theo dõi quá trình trị liệu theo thời gian. Chuyên gia có thể lưu trữ thông tin về buổi trị liệu, ghi nhận các phản hồi, và điều chỉnh lộ trình điều trị theo tình trạng tâm lý của người dùng. Người dùng cũng có thể xem lại các ghi chú và nhận xét từ chuyên gia, giúp họ theo dõi tiến trình của bản thân.
- Thanh toán và quản lý tài chính: Nền tảng cũng tích hợp các giải pháp thanh toán trực tuyến, cho phép người dùng thanh toán phí trị liệu dễ dàng qua các hình thức như thẻ tín dụng, ví điện tử, hoặc chuyển khoản ngân hàng. Hệ thống này thường tự động ghi nhận các lần thanh toán và xuất hóa đơn cho người dùng, giúp minh bạch và đơn giản hóa quá trình quản lý tài chính.

1.5. Đề xuất tích hợp Generative-based Chatbot và kết nối dịch vụ trị liệu vào hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực

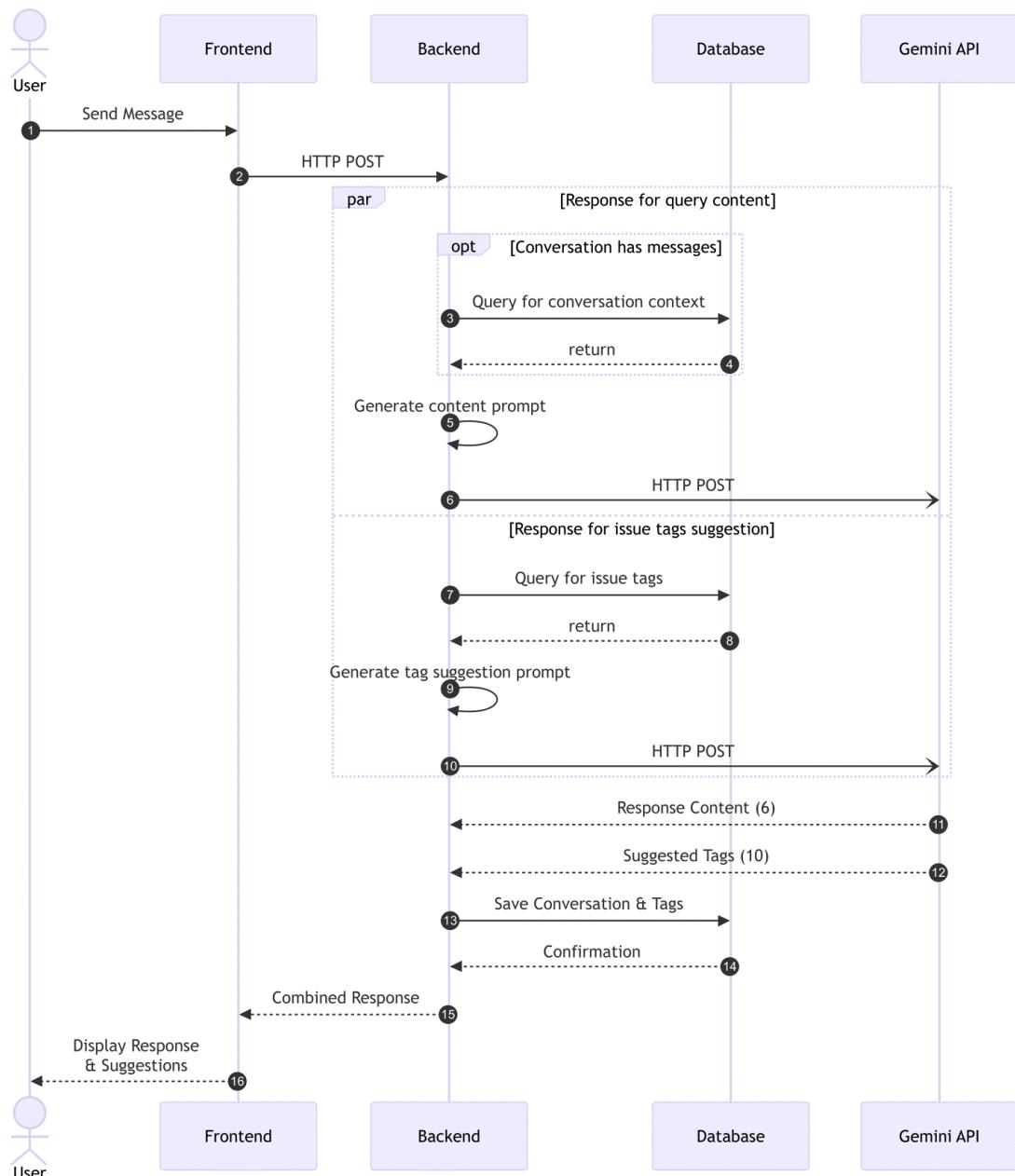
Dựa trên phân tích các kỹ thuật được trình bày ở phần 1.3, nghiên cứu đề xuất giải pháp tích hợp chatbot thông minh vào hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thông qua việc sử dụng RESTful API từ nhà cung cấp dịch vụ Generative AI. Thay vì phát triển mô hình AI từ đầu, hệ thống sẽ tận dụng các API có sẵn của mô hình ngôn ngữ lớn (cụ thể là Gemini API) để xử lý và tạo phản hồi cho người dùng. Giải pháp này giúp tối ưu chi phí phát triển và vận hành, đồng thời tận dụng được sức mạnh của các mô hình AI đã được huấn luyện với dữ liệu lớn.

Hệ thống cũng bao gồm một cơ sở dữ liệu để lưu trữ lịch sử trò chuyện các vấn đề sức khỏe tâm lý được định nghĩa trên hệ thống. Dữ liệu này được sử dụng để cung cấp ngữ cảnh của đoạn hội thoại với chatbot.

Tuy nhiên, vai trò của chatbot chỉ dừng lại ở mức hỗ trợ cơ bản, sau mỗi câu trả lời từ chatbot, hệ thống sẽ trực tiếp gợi ý người dùng kết nối với các chuyên gia trị liệu trên hệ thống dựa trên vấn đề trong câu hỏi mà người dùng gửi với chatbot. Người dùng sẽ có thể đăng ký phiên trị liệu cá nhân với các nhà trị liệu, hoặc tham gia các phiên trị liệu tâm lý công khai mà những người trị liệu tổ chức, dựa trên vấn đề tâm lý mà người

dùng đang gấp phải. Các nhà trị liệu sẽ trực tiếp hướng dẫn và cung cấp liệu pháp trị liệu chuyên nghiệp cho khách hàng qua các buổi trò chuyện cá nhân.

Dưới đây là sơ đồ mô tả quy trình xử lý:



Hình 1.1 Luồng xử lý khi trò chuyện với Chatbot

1.5.1. Xử lý hội thoại

Hệ thống xử lý hai trường hợp chính:

a. Hội thoại mới:

1. Tạo Message đầu tiên từ input người dùng
2. Áp dụng prompt template cho Message
3. Gửi request đến Gemini API
4. Lưu trữ phản hồi như một Message mới trong một Conversation mới

b. Conversation hiện có:

1. Thu thập toàn bộ lịch sử hội thoại
2. Sắp xếp các Message theo thứ tự thời gian
3. Áp dụng prompt template cho Message đầu
4. Thêm query mới vào cuối danh sách
5. Duy trì context cho các phản hồi tiếp theo

1.5.2. Phân tích tag và kết nối dịch vụ

Mỗi request người dùng kích hoạt hai quy trình song song:

- Xử lý phản hồi chatbot thông thường
- Phân tích nội dung để xác định các tag tâm lý liên quan

Để đảm bảo hiệu suất tốt nhất, hệ thống áp dụng các chiến lược sau:

1. Xử lý song song (không đồng bộ) các request đến Gemini API
2. Cấu trúc prompt template tối ưu cho phân tích
3. Lưu trữ ngữ cảnh trò chuyện (Message, Conversation, IssueTag) tại tầng cơ sở dữ liệu, áp dụng caching

Điểm mạnh của hệ thống là nó không chỉ mang lại lợi ích cho người dùng thông thường mà còn tạo điều kiện cho các nhà trị liệu tham gia vào cộng đồng, học hỏi từ đồng nghiệp và chia sẻ kinh nghiệm. Nhà trị liệu có thể quản lý phiên gặp mặt công khai, tương tác với nhiều người tham gia hoặc cung cấp các dịch vụ riêng lẻ cho từng khách hàng, điều này tạo ra một môi trường phát triển kỹ năng và nâng cao chất lượng dịch vụ trị liệu.

Việc kết hợp công nghệ AI chatbot với dịch vụ trị liệu từ chuyên gia giúp cân bằng giữa việc tiếp cận dịch vụ nhanh chóng và duy trì sự hỗ trợ từ con người, đảm bảo rằng người dùng nhận được sự giúp đỡ kịp thời mà vẫn không mất đi giá trị của mối quan hệ trực tiếp với chuyên gia trị liệu.

Ngoài việc sử dụng generative-based AI cho chatbot tư vấn vấn đề tâm lý với người dùng, ta có thể tận dụng mô hình cho các tác vụ giúp nâng trải nghiệm của người dùng, bao gồm: sinh tiêu đề cuộc hội thoại dựa trên câu hỏi người dùng và tóm tắt tất cả các bình luận đánh giá người trị liệu. Các tác vụ này đơn thuần chỉ yêu cầu thiết kế prompt hợp lý.

1.6. Kết luận chương 1

Chương 1 đã tiến hành xác định và phân tích các yêu cầu cốt lõi cho hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực. Quá trình này bắt đầu bằng việc nhận diện nhu cầu ngày càng gia tăng về các công cụ và nền tảng trực tuyến, nơi người dùng có thể tương tác và nhận hỗ trợ từ các chuyên gia trong lĩnh vực sức khỏe tâm lý. Bên cạnh đó, hệ thống còn phải tích hợp các tính năng hiện đại như chatbot AI dựa trên mô hình generative để cung cấp lời khuyên, cũng như hỗ trợ phân tích trạng thái tâm lý người dùng.

Qua việc khảo sát các hệ thống hiện có, bài báo cáo đã tìm hiểu những ưu điểm và hạn chế của các giải pháp hiện tại. Bằng cách tập trung vào các yếu tố cốt lõi

cho trị liệu người dùng đó là việc kết nối dịch vụ trị liệu và hỗ trợ từ chuyên gia là một yêu cầu thiết yếu, giúp người dùng có thể trực tiếp tương tác và nhận sự trợ giúp từ các nhà trị liệu trong thời gian thực, đồng thời giữ cho các cuộc trò chuyện được an toàn và bảo mật. Dự án đề xuất đồng thời tích hợp thêm khả năng hỗ trợ quản lý phiên trị liệu giữa người dùng và người trị liệu và tính năng chia sẻ câu chuyện của bản thân người trị liệu thông qua hệ thống bài đăng.

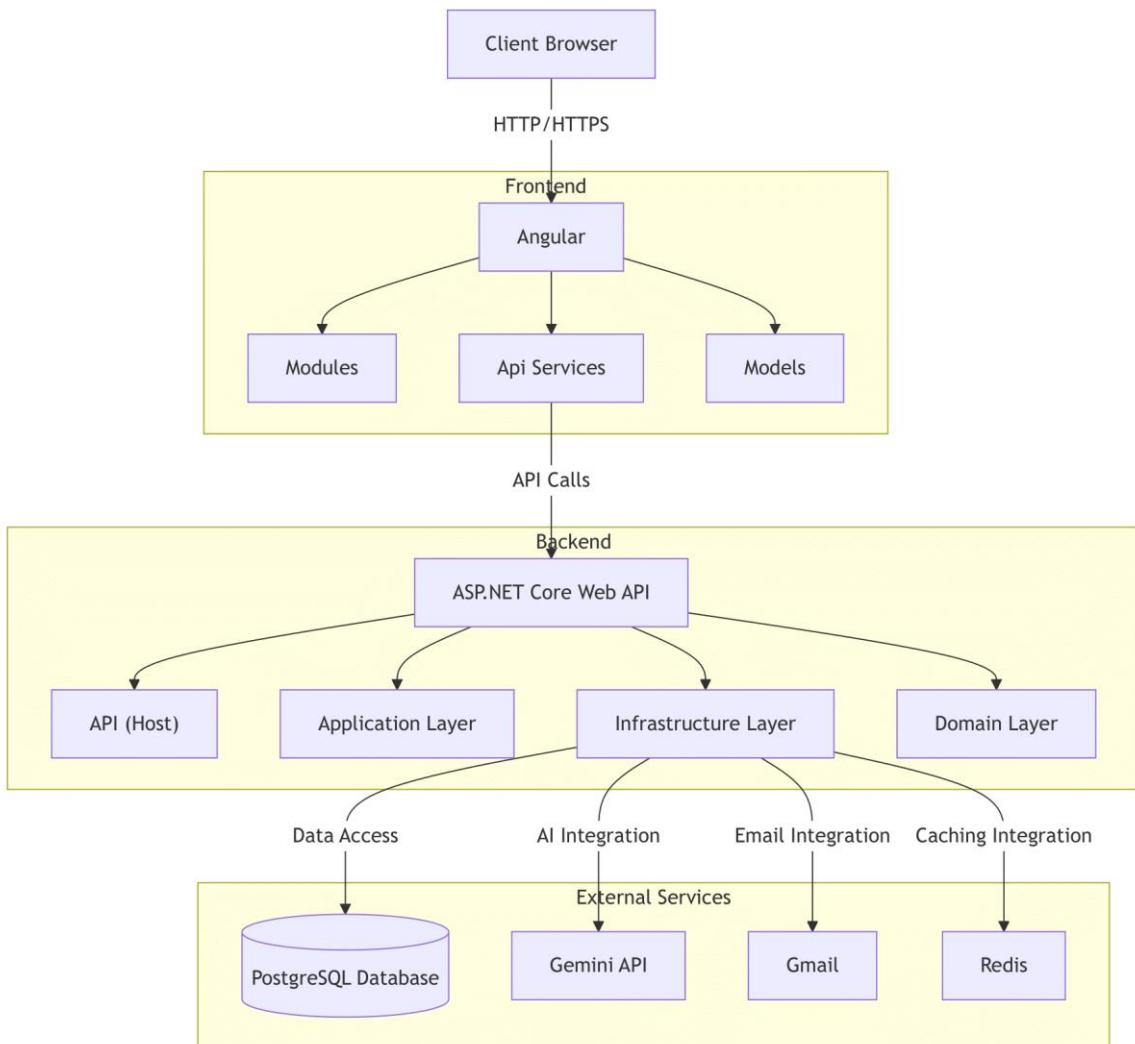
Về mặt kỹ thuật, việc tích hợp công nghệ chatbot generative không chỉ đơn thuần là cung cấp một kênh hỗ trợ tự động, mà còn mở ra cơ hội sử dụng trí tuệ nhân tạo để phân tích và đưa ra các giải pháp trị liệu sơ bộ cho người dùng. Điều này đóng vai trò quan trọng trong việc mở rộng phạm vi hỗ trợ, giúp người dùng có thể tiếp cận với thông tin và hướng dẫn ngay cả khi các chuyên gia không sẵn sàng.

Chương 1 đã xây dựng nền tảng vững chắc cho việc phân tích, thiết kế và phát triển hệ thống, làm cơ sở để triển khai các chức năng phức tạp trong các chương tiếp theo.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG HỖ TRỢ CHĂM SÓC SỨC KHỎE TINH THẦN THỜI GIAN THỰC

Chương 2 trình bày chi tiết quá trình phân tích và thiết kế hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực. Nội dung bao gồm việc xây dựng mô hình tổng quát hệ thống với các công nghệ chính như Angular, ASP.NET Core Web API và Gemini API. Tiếp theo là phân tích hệ thống thông qua yêu cầu bằng ngôn ngữ tự nhiên, sơ đồ use case và biểu đồ lớp phân tích. Phân thiết kế hệ thống được triển khai với các sơ đồ tuần tự, biểu đồ lớp thiết kế, cùng cơ sở dữ liệu. Cuối cùng, chương trình bày quá trình triển khai ứng dụng và rút ra kết luận.

2.1. Mô hình tổng quát hệ thống



Hình 2.1 Sơ đồ mô tả kiến trúc và luồng gọi của hệ thống

Hệ thống được tổ chức dựa trên mô hình Client-Server, chia thành hai thành phần chính:

- **Frontend:** Được phát triển bằng Angular, chịu trách nhiệm hiển thị giao diện người dùng, xử lý tương tác, và gọi API từ Backend
- **Backend:** Được xây dựng bằng ASP.NET Core Web API, thực hiện xử lý logic nghiệp vụ, giao tiếp với cơ sở dữ liệu, và tích hợp với các dịch vụ bên ngoài.

Các thành phần trong hệ thống:

- Frontend
 - Modules: Được tổ chức thành các module để quản lý giao diện và chức năng của ứng dụng. Mỗi module tương ứng với một giao diện chức năng (ví dụ: đăng nhập, đăng ký, quản lý thông tin cá nhân, trò chuyện với AI,...)
 - API Services: Quản lý các yêu cầu từ frontend đến backend. Đây là các lớp chứa phương thức tạo các yêu cầu tới backend và tiếp nhận kết quả (ví dụ: API /register để lấy đăng kí người dùng, /login để gửi đăng nhập)
 - Models: Định nghĩa các cấu trúc dữ liệu được sử dụng trong frontend. (ví dụ cấu trúc dữ liệu cho đăng nhập là LoginRequest, bao gồm thuộc tính là 2 chuỗi username và password)
- Backend: Hệ thống backend được tổ chức dựa trên mô hình đa tầng (Layered Architecture).
 - API (Host): Điểm đầu vào của các yêu cầu từ frontend, xử lý các endpoint và trả về dữ liệu cho các yêu cầu tương ứng (ví dụ: POST /login, POST /register). Tầng này xử lý các yêu cầu HTTP (HTTP request) và yêu cầu thời gian thực thông qua giao thức WebSocket.
 - Application Layer: Xử lý logic ứng dụng, định nghĩa các use case cụ thể. Các phương thức được đặt trong lớp dịch vụ có tên tiền tố là thực thể liên quan tới chức năng đó (ví dụ: use case đăng nhập, đăng kí người dùng được đặt trong UserService)
 - Infrastructure Layer: Kết nối với các dịch vụ bên ngoài (cơ sở dữ liệu, email, cache, AI).
 - Domain Layer: Chứa các logic nghiệp vụ cốt lõi và các đối tượng miền (ví dụ: User, Role,...).

Dự án sẽ sử dụng phương pháp tiếp cận truyền thống (Waterfall) bao gồm các quy trình: phân tích yêu cầu, thiết kế hệ thống, phát triển, kiểm thử và triển khai. Tuy nhiên, vì lý do hạn chế về thời gian và quy mô đồ án cá nhân, giai đoạn kiểm thử chủ yếu được thực hiện theo cách thủ công. Mỗi kịch bản kiểm thử được thực hiện theo quy trình bao gồm: thực hiện thao tác cụ thể, quan sát kết quả, và đối chiếu với yêu cầu thiết kế.

Các công nghệ, kiến trúc lựa chọn trong mô hình hệ thống hình 2.1 sẽ được mô tả chi tiết hơn ở các phần dưới.

2.1.1. Angular

Trong phát triển ứng dụng web, Angular là một trong những framework phổ biến đặc biệt phù hợp với các dự án có quy mô lớn và yêu cầu về tính năng phức tạp. Được phát triển và duy trì bởi Google, Angular không chỉ cung cấp công cụ để xây dựng giao diện người dùng mà còn mang đến một kiến trúc hoàn chỉnh giúp quản lý hiệu quả các luồng dữ liệu, tương tác với backend thông qua REST API.

Đặc điểm chính của Angular:

- Kiến trúc Components: Angular sử dụng kiến trúc dựa trên components, nơi mọi phần của giao diện người dùng được chia thành các thành phần nhỏ độc lập và tái sử dụng. Mỗi component bao gồm phần giao diện (HTML), logic xử lý

(TypeScript), và định dạng (CSS). Điều này giúp quản lý giao diện dễ dàng hơn, giảm thiểu sự phức tạp khi mở rộng ứng dụng.

- Data Binding: Angular hỗ trợ nhiều kiểu data binding (ràng buộc dữ liệu), bao gồm one-way và two-way data binding, cho phép đồng bộ hóa dữ liệu giữa component và giao diện một cách hiệu quả. Điều này rất quan trọng khi ứng dụng tương tác với dữ liệu từ backend thông qua các REST API, giúp hiển thị dữ liệu một cách trực quan và kịp thời.
- Dependency Injection (DI): Một trong những tính năng mạnh mẽ của Angular là Dependency Injection, giúp quản lý các đối tượng và dịch vụ phụ thuộc một cách tự động. DI đặc biệt hữu ích khi kết hợp với REST services trong frontend, giúp các dịch vụ gọi API dễ dàng được tích hợp và sử dụng trong các component khác nhau mà không cần phải khởi tạo thủ công.
- Reactive Programming với RxJS: Angular tích hợp RxJS (Reactive Extensions for JavaScript), một thư viện mạnh mẽ để làm việc với dữ liệu không đồng bộ. Điều này giúp Angular quản lý các luồng dữ liệu đến từ backend REST API một cách dễ dàng, bao gồm xử lý các yêu cầu HTTP, theo dõi thay đổi dữ liệu, và cập nhật giao diện theo thời gian thực.
- Routing: Angular cung cấp hệ thống routing mạnh mẽ, cho phép quản lý các đường dẫn URL và điều hướng giữa các trang trong ứng dụng mà không cần tải lại toàn bộ trang web. Khi ứng dụng sử dụng REST API, routing giúp dễ dàng định tuyến tới các trang liên quan và hiển thị dữ liệu tương ứng từ backend.
- HTTP Client: Angular tích hợp sẵn HttpClient, một công cụ mạnh mẽ để gọi các REST API. Nó hỗ trợ các phương thức HTTP như GET, POST, PUT, và DELETE, giúp việc giao tiếp giữa frontend và backend trở nên dễ dàng và bảo mật hơn. HttpClient còn tích hợp sẵn với RxJS, giúp xử lý các yêu cầu và phản hồi không đồng bộ một cách hiệu quả.
- Tính bảo mật: Angular có thể tích hợp dễ dàng với các cơ chế bảo mật như JWT (JSON Web Token), giúp quản lý và xác thực người dùng trong các ứng dụng web hiện đại. Thông qua HttpClient, Angular có thể gửi các token trong header của yêu cầu HTTP tới backend, giúp bảo vệ các API và chỉ cho phép người dùng hợp lệ truy cập.

Với các tính năng mạnh mẽ, kiến trúc component hiện đại, và khả năng tích hợp dễ dàng với REST API, Angular là lựa chọn lý tưởng để phát triển frontend cho các ứng dụng web phức tạp và yêu cầu tính năng cao. Việc sử dụng Angular cùng với REST services mang lại khả năng quản lý dữ liệu linh hoạt, hiệu quả và giúp ứng dụng dễ dàng mở rộng trong tương lai.

2.1.2. Backend ASP.NET Core Web API

ASP.NET là một framework mã nguồn mở được phát triển bởi Microsoft, nhằm xây dựng các ứng dụng web và dịch vụ hiện đại trên nền tảng .NET. Đây là một phần mở rộng của nền tảng .NET, giúp phát triển các ứng dụng web đa nền tảng.

ASP.NET được giới thiệu lần đầu tiên vào năm 2002 như một phần của .NET Framework, với mục tiêu giúp các nhà phát triển dễ dàng xây dựng các ứng dụng web. ASP.NET Core ra mắt sau đó với nhiều cải tiến như khả năng cross-platform và hiệu

năng tốt hơn, hỗ trợ các hệ điều hành khác ngoài Windows, cùng khả năng tích hợp với Docker để triển khai ứng dụng container.

.NET là một nền tảng phát triển gồm các công cụ, ngôn ngữ lập trình, và thư viện, giúp xây dựng nhiều loại ứng dụng khác nhau. ASP.NET là một phần mở rộng của .NET, cung cấp các thành phần hỗ trợ phát triển ứng dụng web. Một số công nghệ chủ chốt trong .NET bao gồm:

- Các ngôn ngữ lập trình chính là: C#, F#, và Visual Basic
- Thư viện cơ bản cho các tác vụ: xử lý chuỗi, ngày giờ, và hệ thống tệp, mạng máy tính v.v.
- Các công cụ phát triển trên Windows, Linux, macOS, và Docker.

ASP.NET Core mở rộng .NET với các công cụ và thư viện phục vụ riêng cho việc xây dựng ứng dụng web. ASP.NET Core cung cấp khả năng xây dựng nhiều loại ứng dụng web như:

- Xây dựng các trang web động.
- Phát triển dịch vụ RESTful để kết nối các hệ thống.
- Xây dựng các ứng dụng nhỏ, độc lập, dễ mở rộng.
- Sử dụng WebSocket để cung cấp các tính năng tương tác thời gian thực giữa máy chủ và các thiết bị khách.

ASP.NET còn có bộ thư viện Entity Framework Core (EF Core), một Object-Relational Mapper (ORM) được phát triển bởi Microsoft, đóng vai trò quan trọng trong việc tương tác với cơ sở dữ liệu trong các ứng dụng ASP.NET Core. EF Core giúp các nhà phát triển làm việc với cơ sở dữ liệu bằng cách sử dụng các đối tượng C# thay vì phải viết truy vấn SQL thủ công. Điều này không chỉ giúp giảm thiểu lỗi từ con người mà còn nâng cao hiệu quả phát triển phần mềm, đồng thời tối ưu hóa khả năng duy trì và mở rộng hệ thống.

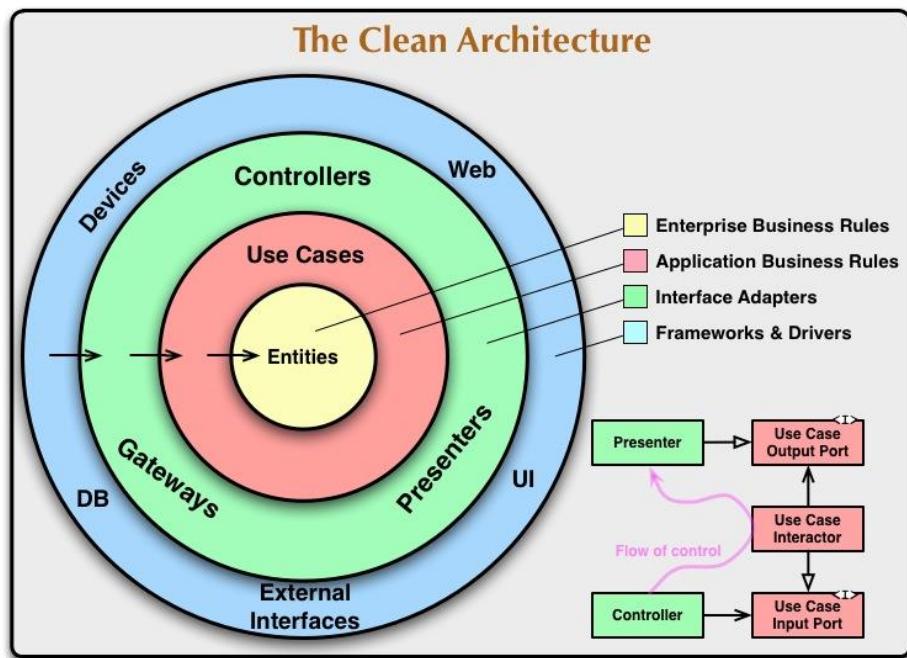
a. Kiến trúc thiết kế mã nguồn

Kiến trúc Clean Architecture, được giới thiệu bởi Robert C. Martin, là một phương pháp tiếp cận thiết kế phần mềm nhằm mục tiêu tạo ra các hệ thống dễ bảo trì, dễ kiểm thử và dễ mở rộng. Mục tiêu của kiến trúc này là đảm bảo rằng hệ thống phần mềm có tính tách biệt rõ ràng giữa các tầng (layers), giúp cải thiện khả năng quản lý và tái sử dụng mã nguồn. Clean Architecture có thể áp dụng vào nhiều loại dự án, từ ứng dụng di động, web đến các hệ thống backend phức tạp [6].

Những nguyên tắc cơ bản của Clean Architecture:

- Không phụ thuộc vào framework: Hệ thống không bị ràng buộc vào các thư viện hay công cụ cụ thể, mà chỉ sử dụng chúng như công cụ hỗ trợ.
- Kiểm thử dễ dàng: Các quy tắc nghiệp vụ có thể được kiểm thử mà không cần phụ thuộc vào UI, cơ sở dữ liệu, hay các thành phần bên ngoài.
- Không phụ thuộc vào giao diện người dùng: Giao diện người dùng có thể thay đổi mà không ảnh hưởng đến các quy tắc nghiệp vụ.
- Không phụ thuộc vào cơ sở dữ liệu: Cơ sở dữ liệu có thể được thay đổi dễ dàng mà không cần sửa đổi các quy tắc nghiệp vụ.

- Không phụ thuộc vào các yếu tố bên ngoài: Các quy tắc nghiệp vụ không biết gì về thế giới bên ngoài, giúp bảo vệ tính toàn vẹn của hệ thống.



Hình 2.2 Sơ đồ giải thích ý tưởng của Clean Architecture [6]

Nguyên tắc phụ thuộc

Các vòng tròn đồng tâm biểu thị các khu vực khác nhau của phần mềm. Càng đi vào trong, phần mềm càng trở nên trừu tượng hơn. Các vòng tròn bên ngoài đại diện cho các cơ chế, trong khi các vòng tròn bên trong là các chính sách, quy luật.

Nguyên tắc chi phối giúp kiến trúc này hoạt động là “Nguyên tắc phụ thuộc” (Dependency Rule). Nguyên tắc này quy định rằng các phụ thuộc trong mã nguồn chỉ có thể hướng vào trong. Không có gì trong một vòng tròn bên trong có thể biết bất cứ điều gì về một vòng tròn bên ngoài. Cụ thể, tên của bất kỳ thứ gì được khai báo trong một vòng tròn bên ngoài không được phép được nhắc đến trong mã ở vòng tròn bên trong. Điều này bao gồm các hàm, lớp, biến, hoặc bất kỳ thực thể phần mềm nào có tên.

Thực thể (Entities)

Thực thể bao bọc các quy tắc nghiệp vụ chung của ứng dụng. Đối với một ứng dụng đơn lẻ, thì các thực thể này là các đối tượng nghiệp vụ của ứng dụng đó. Chúng bao bọc những quy tắc tổng quát bậc cao nhất và ít có khả năng thay đổi khi có sự thay đổi từ bên ngoài.

Use case

Các Use Case là những quy tắc nghiệp vụ cụ thể cho từng ứng dụng. Chúng bao gồm và triển khai tất cả các trường hợp sử dụng của hệ thống. Các use case này điều phối luồng dữ liệu giữa các thực thể và hướng dẫn những thực thể đó áp dụng các quy tắc nghiệp vụ để đạt được mục tiêu của trường hợp sử dụng. Nếu nghiệp vụ vận hành thay đổi thì sự thay đổi mã nguồn phản ánh sẽ diễn ra tại lớp này.

Interface Adapters (bộ chuyển đổi giao diện)

Là một lớp phần mềm bao gồm các bộ chuyển đổi dữ liệu, biến đổi dữ liệu dữ liệu để use case và thực thể để sử dụng một cách thuận tiện. Các dữ liệu đi vào hoặc đi ra thường được gọi là data transfer object hoặc contract, trong khi dữ liệu để điều phối quy tắc nghiệp vụ là các thực thể

Frameworks and Drivers

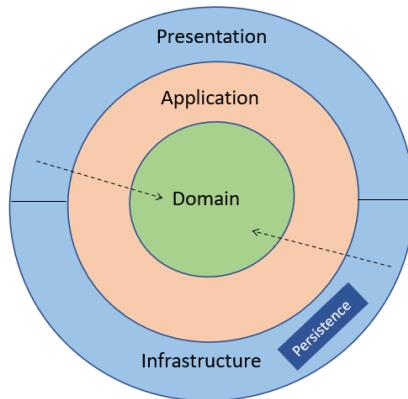
Là lớp ngoài cùng, bao gồm tổ chức hoạt động của framework và các công cụ liên quan. Thông thường ta không viết quá nhiều mã nguồn tại đây mà liên kết tới các vùng bên trong.

Ở góc dưới bên phải của sơ đồ là một ví dụ về cách vượt qua ranh giới từng vòng tròn, cho thấy Controller/Presenter giao tiếp với các use case ở lớp tiếp theo. Lưu ý luồng Controller di chuyển qua use case và sau đó kết thúc bằng việc thực thi trong Presenter. Mọi phụ thuộc trong số chúng đều hướng vào bên trong các use case.

Chúng ta thường giải quyết mâu thuẫn rõ ràng này bằng cách sử dụng nguyên tắc đảo ngược phụ thuộc (dependency inversion). Ví dụ, trong một ngôn ngữ như C#, chúng ta sẽ sắp xếp các giao diện và mối quan hệ kế thừa sao cho các phụ thuộc mã nguồn đối lập với luồng điều khiển tại đúng các điểm trên ranh giới.

Ứng dụng Clean Architecture trong ASP.NET Core của dự án

Trong Clean Architecture, hướng phụ thuộc là chìa khóa để đảm bảo việc phân tách các mối quan tâm và duy trì tính linh hoạt. Các lớp trong Clean Architecture thường được sắp xếp theo cách mà các phụ thuộc luôn hướng vào trong, tập trung vào logic cốt lõi của miền (domain). Kiến trúc được ứng dụng trong dự án như sau:



Hình 2.3 Mô hình kiến trúc các tầng trong ứng dụng backend theo clean architecture

Tầng miền (Domain): Đại diện cho các thực thể trong logic nghiệp vụ cốt lõi của ứng dụng. Không phụ thuộc vào bất kỳ tầng nào khác.

Tầng Ứng dụng (Application Layer)

- Chứa logic nghiệp vụ cụ thể của ứng dụng.
- Thực hiện các trường hợp sử dụng (use case) và các dịch vụ ứng dụng.
- Phụ thuộc duy nhất vào tầng miền

Tầng Hạ tầng (Infrastructure Layer)

- Cài đặt chi các dịch vụ bên ngoài như cơ sở dữ liệu, lưu trữ file, email...
- Chứa cài đặt chi tiết của giao diện (interface) từ tầng ứng dụng hoặc tầng miền
- Phụ thuộc vào tầng ứng dụng và tầng miền

Tầng Trình bày (Presentation Layer - API): Xử lý các yêu cầu từ người dùng tới hệ thống

b. Dịch vụ REST (Representational State Transfer)

Trong hệ thống phát triển ứng dụng hiện đại, dịch vụ REST đóng vai trò quan trọng trong việc cung cấp khả năng giao tiếp giữa các hệ thống qua mạng Internet. Đây là một kiến trúc dịch vụ web phổ biến, dựa trên giao thức HTTP và được sử dụng rộng rãi trong phát triển API. Các dịch vụ REST không chỉ cung cấp một cách tiếp cận đơn giản và hiệu quả để trao đổi dữ liệu, mà còn đảm bảo tính linh hoạt và khả năng mở rộng cao cho các ứng dụng backend.

REST là một phương pháp thiết kế dịch vụ web tập trung vào tài nguyên với các hành động liên quan. Các đặc điểm chính của REST bao gồm:

- Giao thức HTTP: REST chủ yếu sử dụng các phương thức HTTP tiêu chuẩn như GET, POST, PUT, DELETE để thực hiện các thao tác cơ bản trên tài nguyên. Mỗi tài nguyên thường được biểu diễn bằng một URI (Uniform Resource Identifier) duy nhất.
- Tính phi trạng thái: REST là kiến trúc không trạng thái, tức là mỗi yêu cầu từ client đến server phải chứa tất cả thông tin cần thiết để server hiểu và xử lý. Server không cần lưu trạng thái của client giữa các yêu cầu, điều này giúp hệ thống dễ dàng mở rộng và phân phối.
- Khả năng mở rộng: Do tính không trạng thái và sử dụng giao thức HTTP, các dịch vụ REST có khả năng mở rộng tốt khi đối mặt với nhu cầu xử lý khối lượng lớn yêu cầu cùng lúc. Điều này giúp REST trở thành lựa chọn lý tưởng cho các hệ thống lớn và phức tạp.
- Hỗ trợ định dạng dữ liệu đa dạng: REST có thể sử dụng nhiều định dạng dữ liệu như JSON, XML, HTML để trao đổi thông tin. Tuy nhiên, JSON hiện nay là định dạng phổ biến nhất vì tính gọn nhẹ và dễ dàng phân tích.

Trong phát triển backend, REST được ứng dụng rộng rãi nhờ vào sự linh hoạt và khả năng tích hợp tốt với các hệ thống khác nhau. Cụ thể, trong một hệ thống như ứng dụng hỗ trợ sức khỏe tinh thần, REST được sử dụng để cung cấp các API giúp các client (bao gồm ứng dụng web và mở rộng hơn là mobile) có thể tương tác với backend. Những API này xử lý các yêu cầu như đăng ký người dùng, quản lý phiên trò chuyện, tương tác với chatbot AI, hay quản lý thông tin cá nhân của người dùng và chuyên gia.

Một trong những ưu điểm nổi bật của REST là khả năng dễ dàng tích hợp với các công nghệ frontend hiện đại như Angular hoặc React. Nhờ đó, hệ thống có thể phát triển một cách module hóa và phân tách giữa giao diện người dùng và logic xử lý trên server. Trong hệ thống của tôi, ASP.NET Core Web API được sử dụng để xây dựng dịch vụ REST, giúp xử lý các yêu cầu HTTP và cung cấp các endpoint cho client.

c. Cơ sở dữ liệu PostgresSQL

PostgreSQL có nguồn gốc từ hệ thống cơ sở dữ liệu POSTGRES do Michael Stonebraker, một giáo sư tại Đại học California, Berkeley, khởi xướng vào năm 1986. POSTGRES được phát triển với mục tiêu giải quyết những hạn chế của các hệ quản trị cơ sở dữ liệu quan hệ hiện có vào thời điểm đó. Đến năm 1996, dự án này đã tiến hóa thành PostgreSQL khi được bổ sung thêm khả năng hỗ trợ chuẩn SQL [8].

Ban đầu, PostgreSQL được phát triển bởi một nhóm các nhà nghiên cứu dưới sự lãnh đạo của Michael Stonebraker tại UC Berkeley. Sau đó, cộng đồng mã nguồn mở đã tiếp nhận dự án này và phát triển nó thành hệ quản trị cơ sở dữ liệu toàn cầu. Hiện nay, PostgreSQL được duy trì và cải tiến bởi một cộng đồng lớn các lập trình viên, doanh nghiệp, và tổ chức phi lợi nhuận trên toàn thế giới.

PostgreSQL lần đầu tiên xuất hiện trong khảo sát dành cho nhà phát triển vào năm 2018, khi 33% nhà phát triển báo cáo rằng họ sử dụng nó, so với lựa chọn phổ biến nhất năm đó là MySQL, được 59% nhà phát triển sử dụng. Sáu năm sau, PostgreSQL được 49% nhà phát triển sử dụng và là cơ sở dữ liệu phổ biến nhất trong hai năm liên tiếp. Ở vị trí thứ 2 và 3 lần lượt là MySQL và SQLite [9].

Một số ưu điểm vượt trội của PostgreSQL có thể kể đến:

- **Tính ổn định:** PostgreSQL có kiến trúc đa tiến trình, giúp hệ thống duy trì sự ổn định ngay cả khi có lỗi xảy ra ở một tiến trình phụ. Khả năng này giúp đảm bảo rằng một tiến trình bị sự cố sẽ không ảnh hưởng đến hoạt động của các tiến trình khác. Ngoài ra, PostgreSQL duy trì hiệu suất cao ngay cả khi đọc và ghi dữ liệu đồng thời, không bị suy giảm sau khi đạt đỉnh.
- **Cộng đồng sôi nổi:** PostgreSQL có một cộng đồng phát triển rất năng động, thường xuyên phát hành các bản vá lỗi mỗi 3 tháng và cập nhật một phiên bản lớn mỗi năm.
- **Chức năng mạnh mẽ:** PostgreSQL hỗ trợ nhiều loại bảng, chỉ mục và các đối tượng cơ sở dữ liệu khác. Các chức năng mạnh mẽ bao gồm tìm kiếm văn bản toàn văn, chỉ mục bitmap, cây tìm kiếm tổng quát (GiST), và phân mảnh bảng theo nhiều cách khác nhau như phân mảnh theo khoảng, theo hàm băm, theo danh sách. Ngoài ra, PostgreSQL còn hỗ trợ phục hồi dữ liệu theo thời gian (PITR), cùng với các phương thức sao lưu và phục hồi đồng bộ hoặc không đồng bộ giữa các máy chủ để đảm bảo tính sẵn sàng cao.
- **Dễ dàng trong vận hành và bảo trì:** PostgreSQL cho phép đặt các câu lệnh định nghĩa dữ liệu (DDL) trong một giao dịch để chúng có thể được thực hiện hoặc quay lại toàn bộ một cách đồng nhất. Hệ thống cũng hỗ trợ nhiều phương thức sao chép dữ liệu và đồng bộ, cùng với việc tối ưu hóa các chỉ mục và cấu trúc bảng một cách trực tuyến.

d. Redis cache

Redis (Remote Dictionary Server) là một hệ thống lưu trữ dữ liệu dạng key-value, mã nguồn mở, được giới thiệu vào năm 2009 bởi Salvatore Sanfilippo. Ban đầu, Redis được phát triển nhằm giải quyết các vấn đề về hiệu suất trong việc xử lý dữ liệu lớn với yêu cầu tốc độ cao. Với kiến trúc lưu trữ dữ liệu trong bộ nhớ RAM và hỗ trợ ghi xuống

ở đĩa, Redis mang lại hiệu suất vượt trội trong các ứng dụng yêu cầu truy xuất nhanh chóng.

Redis không chỉ là một hệ thống lưu trữ key-value thông thường mà còn hỗ trợ nhiều kiểu dữ liệu như chuỗi (string), danh sách (list), tập hợp (set), tập hợp có thứ tự (sorted set), và bản đồ băm (hash). Điều này giúp Redis trở nên linh hoạt trong việc xử lý các loại dữ liệu khác nhau. Redis còn hỗ trợ các tính năng nâng cao như Pub/Sub (publish-subscribe) để xây dựng các hệ thống giao tiếp thời gian thực, quản lý khóa phân tán (distributed locks), và Lua scripting để thực thi các đoạn mã logic phức tạp trực tiếp trên server.

Trong các ứng dụng web hiện đại, Redis thường được sử dụng như một bộ nhớ đệm (cache) để giảm tải cho cơ sở dữ liệu chính và cải thiện hiệu suất hệ thống. Một số ứng dụng phổ biến bao gồm:

- Caching dữ liệu thường xuyên truy cập: Redis giúp lưu trữ tạm thời các kết quả truy vấn, như thông tin sản phẩm, danh sách bài viết, hoặc thông tin người dùng. Điều này giảm thiểu số lượng truy vấn đến cơ sở dữ liệu và tăng tốc độ phản hồi.
- Quản lý phiên làm việc (session): Redis hỗ trợ lưu trữ các phiên người dùng trong ứng dụng web, đảm bảo khả năng mở rộng (scalability) khi triển khai trên nhiều máy chủ.
- Hàng đợi công việc (job queues): Redis được sử dụng để quản lý hàng đợi công việc, như xử lý email, thông báo, hoặc tác vụ nền (background tasks).
- Giao tiếp thời gian thực: Sử dụng Pub/Sub, Redis hỗ trợ các tính năng như gửi thông báo thời gian thực hoặc xây dựng các ứng dụng chat.

Về mặt kỹ thuật, Redis sử dụng một cơ chế lưu trữ dữ liệu trong bộ nhớ (in-memory), giúp giảm độ trễ (latency) truy xuất xuống mức micro-giây. Ngoài ra, Redis hỗ trợ cơ chế sao lưu dữ liệu (snapshot) và ghi lại các thay đổi (append-only file) để đảm bảo an toàn dữ liệu. Redis còn cung cấp tính năng cluster, cho phép phân tán dữ liệu trên nhiều node, giúp hệ thống đạt hiệu năng cao và khả năng chịu lỗi (fault tolerance).

Nhờ khả năng mạnh mẽ và linh hoạt, Redis là lựa chọn lý tưởng trong việc cải thiện hiệu suất và khả năng mở rộng của các hệ thống web hiện đại.

e. Json Web Token (JWT)

Trong bối cảnh phát triển các dịch vụ REST như đã đề cập ở phần trước, bảo mật là một yếu tố quan trọng cần được quan tâm. Một trong những phương thức phổ biến để bảo mật các API REST là sử dụng JWT (JSON Web Token). JWT là một tiêu chuẩn mã hóa được sử dụng để truyền thông tin giữa các bên một cách an toàn trong các ứng dụng web.

JWT là một chuỗi văn bản được mã hóa theo dạng Base64, bao gồm ba phần:

- Header: Chứa thông tin về thuật toán mã hóa và loại token, thường là JWT.
- Payload: Chứa các thông tin (claims) như thông tin người dùng hoặc quyền hạn, dữ liệu này có thể được xác thực nhưng không mã hóa.

- Signature: Phần chữ ký được tạo ra từ header và payload bằng cách sử dụng khóa bí mật và thuật toán mã hóa, đảm bảo tính toàn vẹn của token.

Khi người dùng đăng nhập thành công, hệ thống sẽ tạo ra một JWT và gửi về client. Mỗi lần client thực hiện yêu cầu tới API REST (ví dụ như yêu cầu lấy dữ liệu cá nhân hay quản lý phiên trò chuyện), token này sẽ được gửi kèm theo yêu cầu thông qua header HTTP (thường là trong trường Authorization).

API server sẽ xác thực token để đảm bảo rằng yêu cầu đến từ người dùng hợp lệ và quyền hạn của họ được xác định chính xác. Sau khi token được xác thực, server sẽ tiến hành xử lý yêu cầu và gửi phản hồi về client.

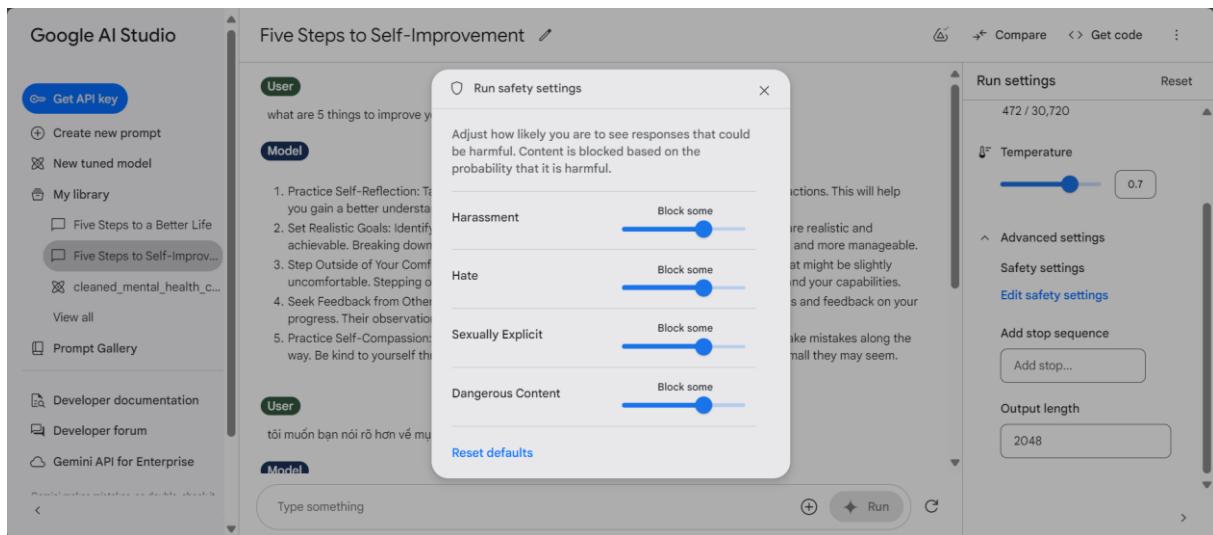
Việc sử dụng JWT giúp hệ thống đạt được sự bảo mật, gọn nhẹ và hiệu quả trong quản lý phiên đăng nhập và xác thực người dùng, đồng thời tạo sự liên kết chặt chẽ với kiến trúc REST đã được triển khai.

2.1.3. Gemini API

Gemini, một trong những mô hình ngôn ngữ tiên tiến nhất từ Google, có thể cung cấp nhiều tính năng và khả năng vượt trội trong lĩnh vực chăm sóc sức khỏe tinh thần khi được tích hợp vào các chatbot generative. Đặc biệt, trong bối cảnh hỗ trợ tâm lý và tương tác tự nhiên với người dùng, Gemini mang đến những cải tiến đáng kể về khả năng xử lý ngôn ngữ, học hỏi từ tương tác, và tùy chỉnh dựa trên dữ liệu thực tế.

Trước hết, Gemini có khả năng xử lý ngôn ngữ tự nhiên (NLP) mạnh mẽ hơn nhiều so với các hệ thống trước đó. Với sự cải tiến trong kiến trúc AI, Gemini có thể hiểu và phân tích ngữ cảnh phức tạp của các cuộc trò chuyện dài, giúp chatbot duy trì sự mạch lạc và nhất quán trong phản hồi. Điều này đặc biệt hữu ích trong lĩnh vực trị liệu tâm lý, nơi mà các vấn đề cá nhân và trạng thái cảm xúc của người dùng thường yêu cầu sự hiểu biết sâu sắc và phân tích chi tiết. Nhờ khả năng này, Gemini có thể giúp chatbot xử lý ngôn ngữ chính xác và đáp ứng ngữ cảnh với mức độ chính xác cao, tạo ra các cuộc hội thoại gần gũi và đồng cảm hơn.

Một tính năng quan trọng khác của Gemini là cơ chế tự điều chỉnh qua học máy, cho phép mô hình được fine-tune dựa trên các dữ liệu cụ thể về người dùng hoặc lĩnh vực mà chatbot đang hoạt động. Trong trường hợp chatbot trị liệu tâm lý, mô hình có thể được fine-tune với các tập dữ liệu từ các tình huống tâm lý phổ biến, các phản hồi từ chuyên gia, hoặc thậm chí các dữ liệu ẩn danh từ những cuộc trò chuyện trước đó. Việc fine-tune này giúp Gemini không chỉ cải thiện khả năng tạo ra phản hồi phù hợp với nhu cầu của người dùng mà còn tạo ra những câu trả lời có sự đồng cảm và nhạy bén với trạng thái cảm xúc. Hơn nữa, việc tùy chỉnh này đảm bảo rằng chatbot có thể đưa ra các khuyến nghị và hướng dẫn phù hợp hơn cho từng người dùng, dựa trên thông tin đã có về họ.

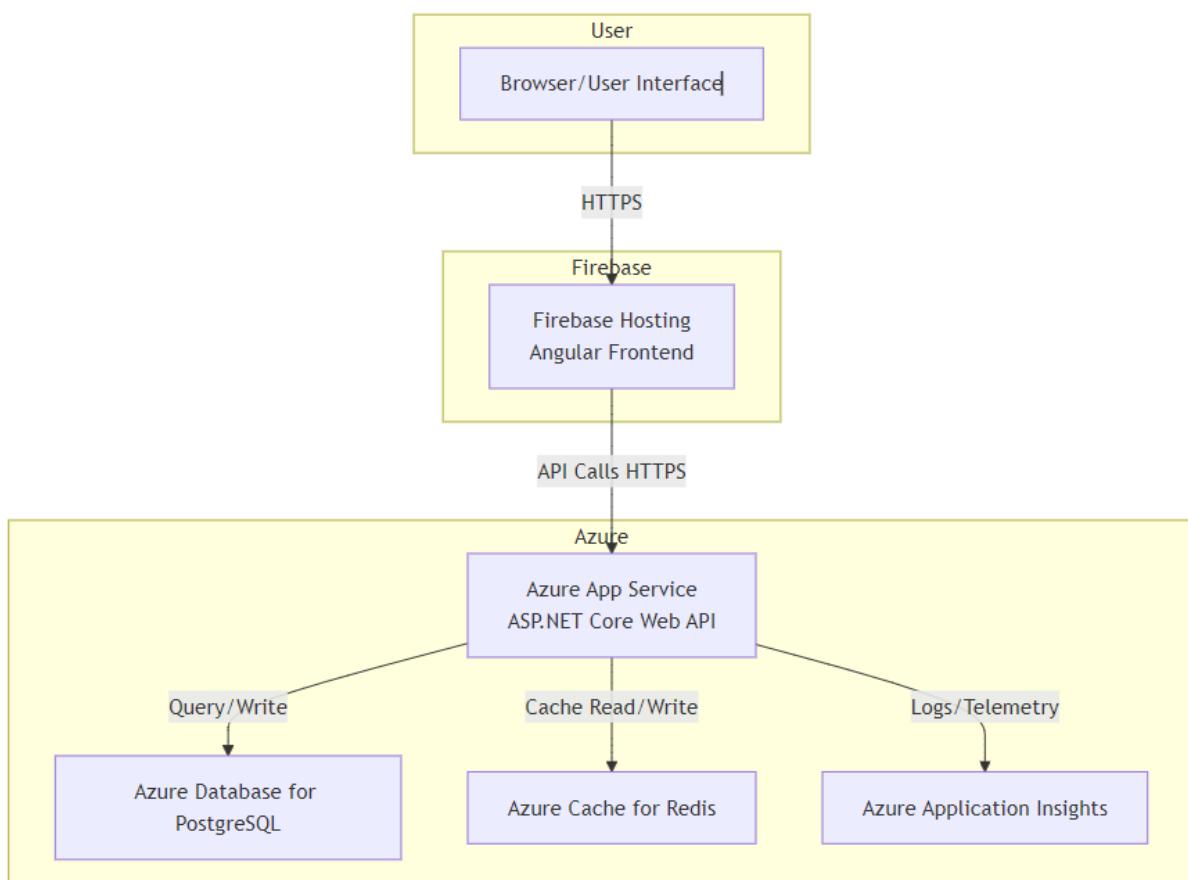


Hình 2.4 Giao diện thử nghiệm mô hình Google AI Studio

Google cung cấp sẵn giao diện Google AI Studio để fine-tuned mô hình với bộ dữ liệu cá nhân và giao diện trò chuyện trực tiếp để thiết kế prompt đảm bảo mô hình sinh hoạt động đúng với mong muốn kì vọng, cùng với các tham số của mô hình có thể tùy chỉnh để ngăn mô hình sinh các nội dung nhạy cảm. Sau khi quá trình thử nghiệm hoàn tất, ta có thể tích hợp mô hình fine-tuned thông qua phương thức HTTP POST.

2.1.4. Triển khai ứng dụng

Sơ đồ triển khai ứng với mô hình kiến trúc tổng quát (hình 2.1) được mô tả như sau:



Hình 2.5 Sơ đồ triển khai dự án

Firebase Hosting (Angular Frontend)

- Vai trò: Đóng vai trò như một máy chủ để lưu trữ và phân phối giao diện người dùng (Frontend) được phát triển bằng Angular.
- Công nghệ:
 - Firebase Hosting cung cấp dịch vụ lưu trữ tĩnh, với tính năng tối ưu hóa hiệu suất và tích hợp HTTPS.
 - Tích hợp liền mạch với API để gửi các yêu cầu đến backend thông qua các cuộc gọi API.

Azure App Service (ASP.NET Core Web API)

- Vai trò: Thành phần backend chính, cung cấp các dịch vụ API cho ứng dụng.
- Công nghệ:
 - Được triển khai trên Azure App Service, sử dụng ASP.NET Core Web API để xử lý logic nghiệp vụ và giao tiếp với các thành phần khác.
 - Đảm bảo hiệu năng và khả năng mở rộng trên môi trường đám mây.

Azure Database for PostgreSQL

- Vai trò: Cơ sở dữ liệu chính để lưu trữ dữ liệu của ứng dụng, bao gồm thông tin người dùng, bài viết, tin nhắn, và các phiên trò chuyện.
- Công nghệ: Azure Database for PostgreSQL cung cấp giải pháp quản lý cơ sở dữ liệu quan hệ với hiệu suất cao trong hệ sinh thái Azure và các phương pháp bảo mật cao.

Azure Cache for Redis

- Vai trò: Cải thiện hiệu năng của ứng dụng bằng cách lưu trữ dữ liệu tạm thời (cache) để giảm thiểu truy vấn trực tiếp vào cơ sở dữ liệu.
- Công nghệ: Azure Cache for Redis được sử dụng để lưu trữ dữ liệu tạm thời và hỗ trợ các tác vụ đọc/ghi nhanh, hỗ trợ tốt trong hệ sinh thái Azure.

Azure Application Insights

- Vai trò: Theo dõi và ghi lại các hoạt động của hệ thống để phát hiện lỗi, theo dõi hiệu năng và ghi lại nhật ký.
- Công nghệ: Azure Application Insights cung cấp khả năng phân tích các ghi chép hệ thống phục vụ cho tối ưu hóa hệ thống và định hướng phát triển dịch sau này.

Luồng hoạt động tổng quát:

- Người dùng truy cập giao diện (Angular) được lưu trữ trên Firebase Hosting.
- Các yêu cầu của người dùng được gửi qua HTTPS tới API được triển khai trên Azure App Service.
- API tương tác với các dịch vụ trên nền tảng Azure: PostgreSQL để lưu trữ hoặc truy xuất dữ liệu, Redis để cải thiện hiệu suất và Application Insights để ghi lại các nhật ký hệ thống. API cũng tương tác với các dịch vụ AI và email đã nêu trong phần 2.1.

2.2. Phân tích hệ thống

Danh từ “người trị liệu” và “chuyên gia” sẽ được sử dụng thay thế cho nhau và có ý nghĩa giống nhau trong quá trình phân tích ứng dụng

2.2.1. Yêu cầu của hệ thống bằng ngôn ngữ tự nhiên

Từ các kết luận cho tổng quan hệ thống đề ra tại mục 1.7 trong báo cáo, các yêu cầu hệ thống chi tiết để thiết kế hệ thống mà dự án đề ra như sau:

Những người đã xác thực vào hệ thống sẽ được phân chia theo chức năng:

- Người dùng cơ bản (User/Client): bất kì ai truy cập vào hệ thống đều có chức năng:

- Quản lý thông tin cá nhân
- Quản lý bài đăng chia sẻ, nhật ký của mình
- Tương tác với các bài đăng công khai
- Trò chuyện với AI chatbot về các vấn đề sức khoẻ tâm lý. Chatbot có thể gợi ý vấn đề người dùng đang gặp phải thông qua trò chuyện. Người dùng có thể truy vấn các nội dung liên quan tới vấn đề được gợi ý trên ứng dụng.
- Đăng ký chương trình (phiên) trị liệu cá nhân với chuyên gia
- Trò chuyện cá nhân với người trị liệu
- Quản lý các buổi trị liệu cá nhân với người trị liệu (xem, cập nhật)
- Đánh giá phiên trị liệu với người trị liệu. Người dùng chỉ có thể đánh giá khi đã tham gia ít nhất một phiên trị liệu cá nhân với nhà trị liệu. Người dùng chỉ có thể đánh giá một lần cho một nhà trị liệu và có thể cập nhật.
- Theo dõi các phiên công khai tổ chức bởi người trị liệu.
Người dùng cần được thông báo về trạng thái thay đổi của phiên công khai đã theo dõi qua thông báo, email và nhắc lại lịch tham gia trước một ngày qua email
- Quản lý thông báo cá nhân (xem, sửa, xoá)

Một khách hàng chỉ có thể chọn 1 người trị liệu để được hướng dẫn tại một thời điểm.

Khách hàng chỉ có thể trò chuyện với người trị liệu đã đăng ký (và được người trị liệu chấp thuận)

- Người trị liệu (Therapist): Người cung cấp dịch vụ trị liệu cho những người dùng khác

- Quản lý các phiên công khai gấp mặt tự tổ chức. Đây là các phiên công khai và mọi người có thể theo dõi tham gia, hình thức tùy chọn online hoặc offline.

Thời gian bắt đầu phiên phải trước thời điểm tạo 3 ngày, không thể tạo hơn 2 buổi trong một tuần. Không cho phép người tạo xoá phiên mà chỉ cập nhật trạng thái.

- Quản lý các bản ghi đăng ký trị liệu cá nhân tới từ khách hàng

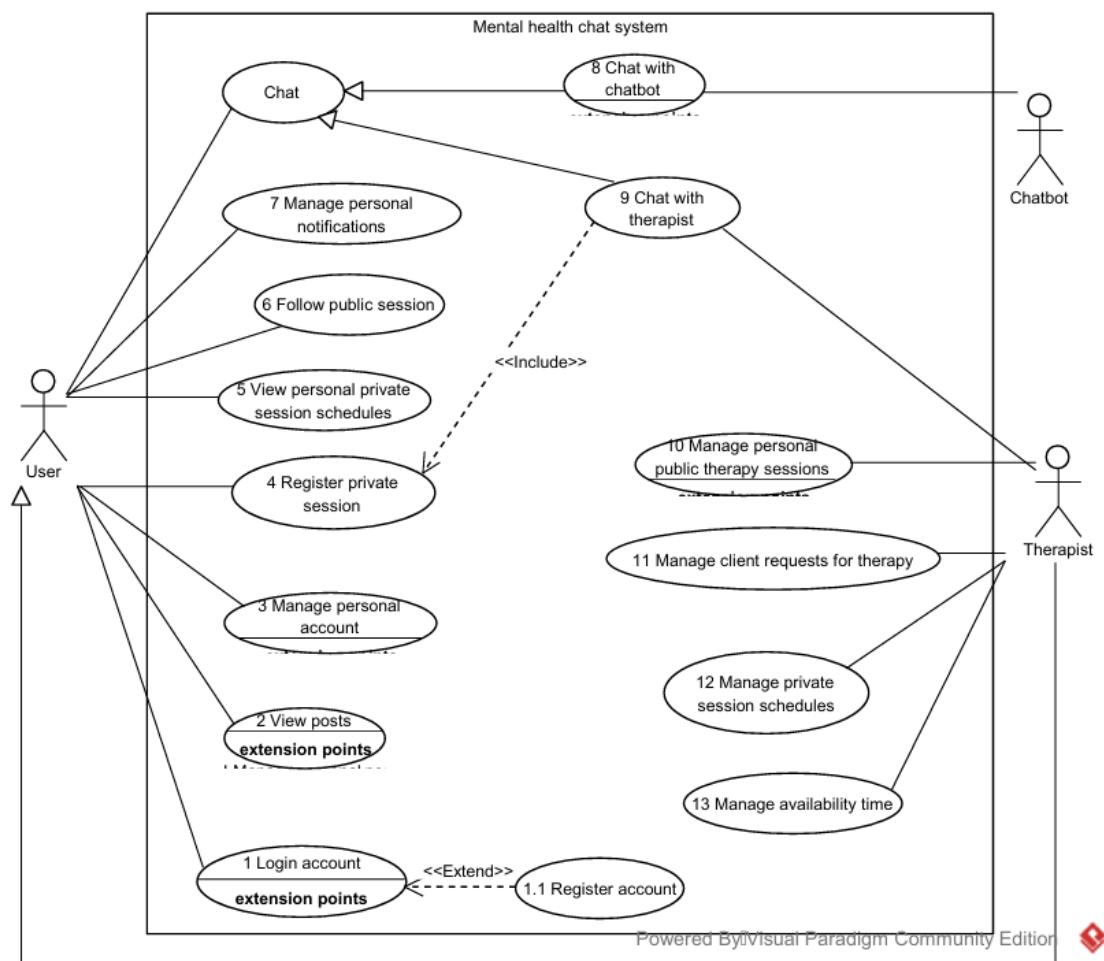
- Quản lý lịch trị liệu (thời gian làm việc) cá nhân. Đây là lịch làm việc của người trị liệu trong các ngày trong tuần. Lịch này có thể thay đổi và có thể được ghi đè bởi một số ngày đặc biệt trong năm
- Trò chuyện cá nhân với khách hàng
- Quản lý các phiên trị liệu cá nhân với khách hàng của mình

Gửi thông báo cho người được trị liệu và người trị liệu về nội dung của buổi trị liệu cá nhân

Một người trị liệu chỉ có thể nhận tối đa 5 khách hàng trong một thời điểm để đảm bảo chất lượng trị liệu.

Một người trị liệu cũng là một người dùng bình thường, và cũng có thể cần đến dịch vụ trị liệu từ người trị liệu khác.

2.2.2. Sơ đồ use case tổng quan hệ thống



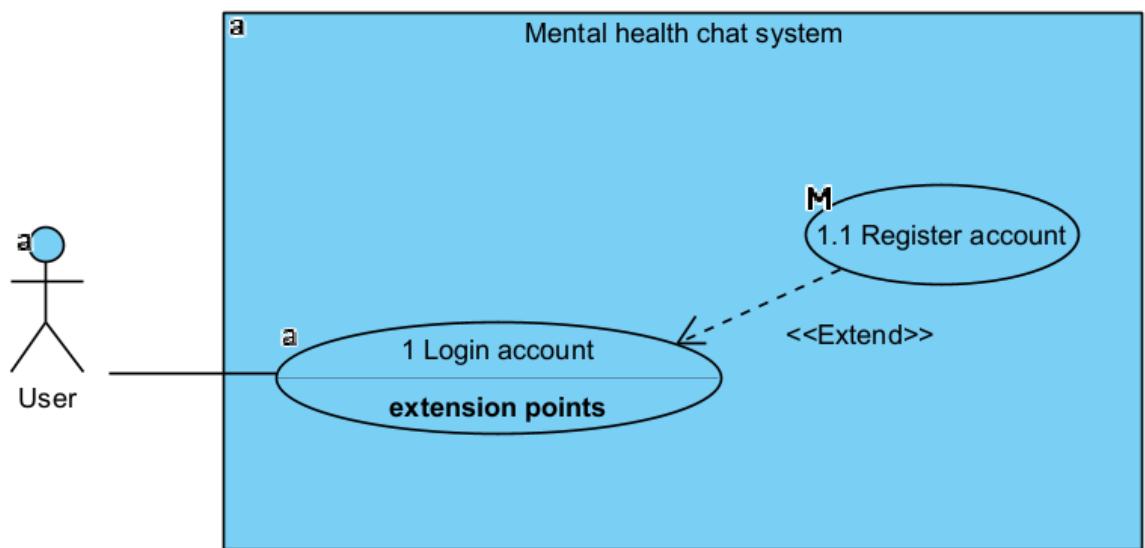
Hình 2.6 Sơ đồ use case tổng quan hệ thống

- Thành viên của hệ thống
 - User: người sử dụng ứng dụng nói chung
 - Therapist: Chuyên gia/người cung cấp dịch vụ trị liệu
 - AI chatbot: Hệ thống AI đưa ra các lời khuyên về vấn đề tâm lý

- Chức năng của người dùng cơ bản
 - Đăng nhập tài khoản
 - Đăng kí tài khoản
 - Quản lý thông tin cá nhân: xem, cập nhật, xoá
 - Quản lý các bài đăng cá nhân: xem, tạo, sửa, xoá
 - Xem và tương tác (thích) với các bài đăng công khai
 - Xem các phiên trị cá nhân
 - Tìm kiếm người trị liệu
 - Đăng kí chương trình (phiên) trị liệu cá nhân
 - Trò chuyện với người trị liệu
 - Trò chuyện với AI chatbot
 - Tham gia phiên trị liệu công khai
 - Đánh giá người trị liệu
 - Quản lý thông báo cá nhân: xem, xoá
- Chức năng của người trị liệu
 - Quản lý các phiên trị liệu công khai tự tổ chức: xem, tạo, sửa
 - Quản lý yêu cầu trị liệu từ người dùng: xem, sửa
 - Quản lý các buổi trị liệu với các khách hàng: xem, tạo, sửa
 - Quản lý lịch trị liệu (thời gian làm việc) cá nhân: xem, tạo, xoá
 - Trò chuyện với người dùng

2.2.3. Các đặc tả use case

a. Use case 1: Đăng nhập tài khoản



Hình 2.7 Use case đăng nhập tài khoản người dùng

1. Đăng nhập tài khoản

Bảng 2.1 Đặc tả use case đăng nhập tài khoản

Tên use case	Login account
--------------	---------------

Mã use case	1
Mô tả	Là người dùng, tôi muốn đăng nhập vào hệ thống để sử dụng
Actor	Người dùng
Trigger	Người dùng vào trang đăng nhập vào hệ thống
Tiền điều kiện	<ul style="list-style-type: none"> • Người dùng đã có tài khoản ứng dụng • Người dùng chưa đăng nhập vào ứng dụng trước đó
Hậu điều kiện	Giao diện điều hướng người dùng vào trang chủ
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng truy cập vào ứng dụng Mental health chat 2. Hệ thống hiển thị giao diện đăng nhập, bao gồm: trường nhập liệu nhập bằng tên tài khoản, mật khẩu và nút đăng nhập 3. Người dùng nhập đầy đủ trường nhập liệu và chọn đăng nhập 4. Hệ thống xác thực tài khoản thành công và lưu kết nối người dùng
Ngoại lệ	<p>4.1. Hệ thống xác thực tài khoản người dùng thất bại do sai tên tài khoản hoặc sai mật khẩu và gửi thông điệp lỗi</p> <p>4.1.1 Người dùng sửa lại các trường nhập liệu sai và chọn đăng nhập lại</p> <p><i>Use case tiếp tục bước 4</i></p> <p>4.2. Hệ thống xác thực thất bại do người dùng đã xoá tài khoản và thông báo lại cho người dùng</p> <p><i>Use case kết thúc</i></p> <p>4.3. Người dùng chọn đăng ký tài khoản</p> <p><i>Use case chuyển sang use case 1.1</i></p>

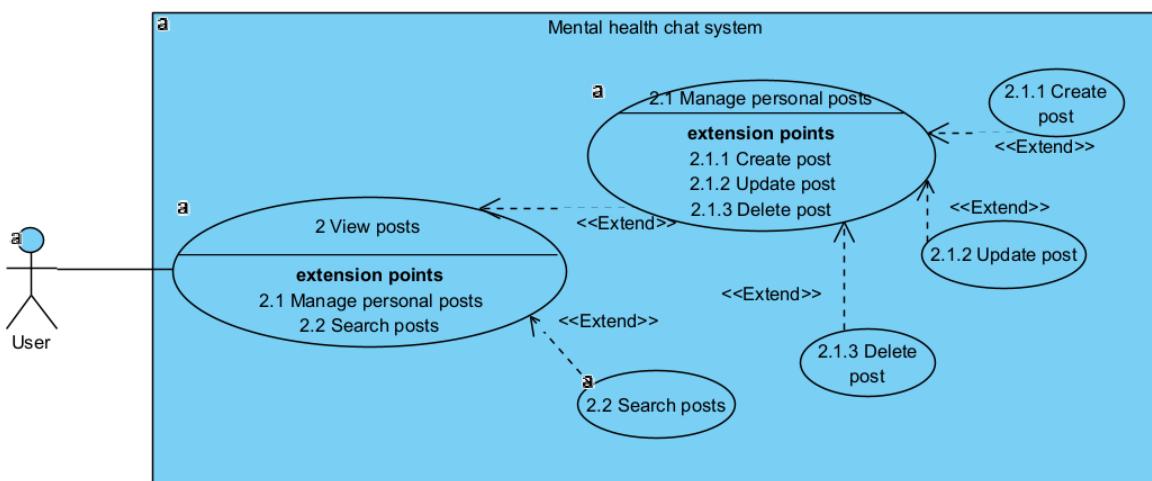
1.1 Đăng ký tài khoản

Bảng 2.2 Đặc tả use case đăng ký tài khoản

Tên use case	Register account
Mã use case	1.1
Mô tả	Là người dùng, tôi muốn tạo một tài khoản trong hệ thống
Actor	Người dùng
Trigger	Người dùng vào trang đăng ký tài khoản hệ thống
Tiền điều kiện	<ul style="list-style-type: none"> • Người dùng có kết nối tới internet • Người dùng chưa có tài khoản ứng dụng
Hậu điều kiện	Hệ thống ghi nhận tài khoản mới của người dùng
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng truy cập vào ứng dụng Mental health chat 2. Hệ thống hiển thị giao diện đăng nhập 3. Người dùng chọn “Tạo tài khoản” trên màn hình 4. Hệ thống chuyển sang màn hình đăng ký tài khoản, bao gồm: <ul style="list-style-type: none"> • các trường danh tính trên hệ thống bắt buộc: tên tài khoản, email, mật khẩu • các trường thông tin cá nhân: họ, tên, ngày sinh, số điện thoại, giới tính, ảnh đại diện (tùy chọn). • các trường (tùy chọn) đăng ký chuyên gia: tiêu sử, lĩnh vực tri thức, học vấn, kinh nghiệm, chứng chỉ • nút xác nhận đăng ký

	<p>5. Người dùng nhập tối thiểu các trường dữ liệu bắt buộc và chọn đăng ký</p> <p>6. Hệ thống ghi nhận tài khoản mới và thông báo đăng ký thành công</p>
Ngoại lệ	<p>6.1. Hệ thống xác thực thông tin đăng ký tài khoản không thành công do số điện thoại hoặc email hoặc tên người dùng đã tồn tại và hiển thị thông báo</p> <p>6.1.1. Người dùng thay đổi các trường dữ liệu lỗi và chọn đăng ký <i>Use case tiếp tục bước 6</i></p>
Yêu cầu phi chức năng	NFR1. Mật khẩu phải sử dụng hàm băm trước khi lưu trữ và đổi chiều trên hệ thống

b. Use case 2: Xem bài đăng



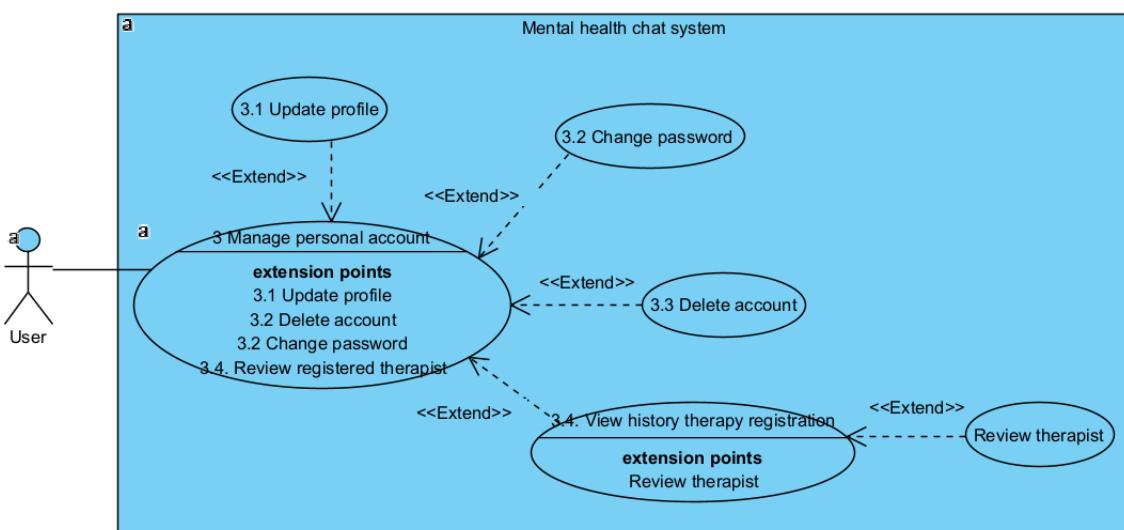
Hình 2.8 Use case xem bài đăng

Bảng 2.3 Đặc tả use case xem bài đăng

Tên use case	View posts
Mã use case	2
Mô tả	Là người dùng, tôi muốn xem các bài đăng/chia sẻ công khai trên hệ thống
Actor	Người dùng
Trigger	Người dùng chọn xem các bài đăng (post)
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Giao diện hiện lên trang xem bài đăng trên hệ thống
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút “Posts” trên thanh điều hướng chính Giao diện điều hướng người dùng tới trang xem các bài đăng trên hệ thống. Giao diện bao gồm: <ul style="list-style-type: none"> Nút chọn xem bài đăng trên hệ thống “Public” (mặc định) Nút chọn xem bài đăng cá nhân “Personal” Nếu người dùng chọn xem bài đăng công khai: Giao diện hiện lên trang các bài đăng/chia sẻ công khai trên toàn hệ thống sắp xếp từ mới nhất Nếu người dùng chọn xem bài đăng cá nhân

	<p>a. Giao diện hiện lên trang mới, bao gồm:</p> <ul style="list-style-type: none"> • nút thêm bài đăng mới • danh sách các bài đăng cá nhân. Mỗi bài đăng có lựa chọn: chỉnh sửa hoặc xoá <p>b. Nếu người dùng chọn tạo bài đăng mới</p> <ol style="list-style-type: none"> i. Giao diện hiện lên hộp thoại với nút đăng và các trường: <ul style="list-style-type: none"> • Title: Tiêu đề bài đăng (tuỳ chọn) • Content: Nội dung bài đăng • Type: Lựa chọn kiểu: công khai hoặc nhât kí ii. Người dùng nhập vào các trường dữ liệu và chọn nút đăng iii. Giao diện thông báo thêm bài đăng thành công và danh sách bài đăng cá nhân cập nhật <p>c. Nếu người dùng muốn chỉnh sửa hoặc xoá bài đăng:</p> <ol style="list-style-type: none"> i. Người dùng chọn hành động “chỉnh sửa” trên một bài đăng cá nhân ii. Giao diện hiện lên hộp thoại với nút đăng và các trường đã được điền sẵn thông tin cũ: <ul style="list-style-type: none"> • Title: Tiêu đề bài đăng (tuỳ chọn) • Content: Nội dung bài đăng • Type: Lựa chọn kiểu: công khai hoặc nhât kí iii. Người dùng nhập các thay đổi vào các trường dữ liệu và chọn nút chỉnh sửa/xoá iv. Giao diện thông báo cập nhật/xoá bài đăng thành công và danh sách bài đăng cá nhân cập nhật
--	---

c. Use case 3: Quản lý thông tin cá nhân



Hình 2.9 Use case quản lý tài khoản cá nhân

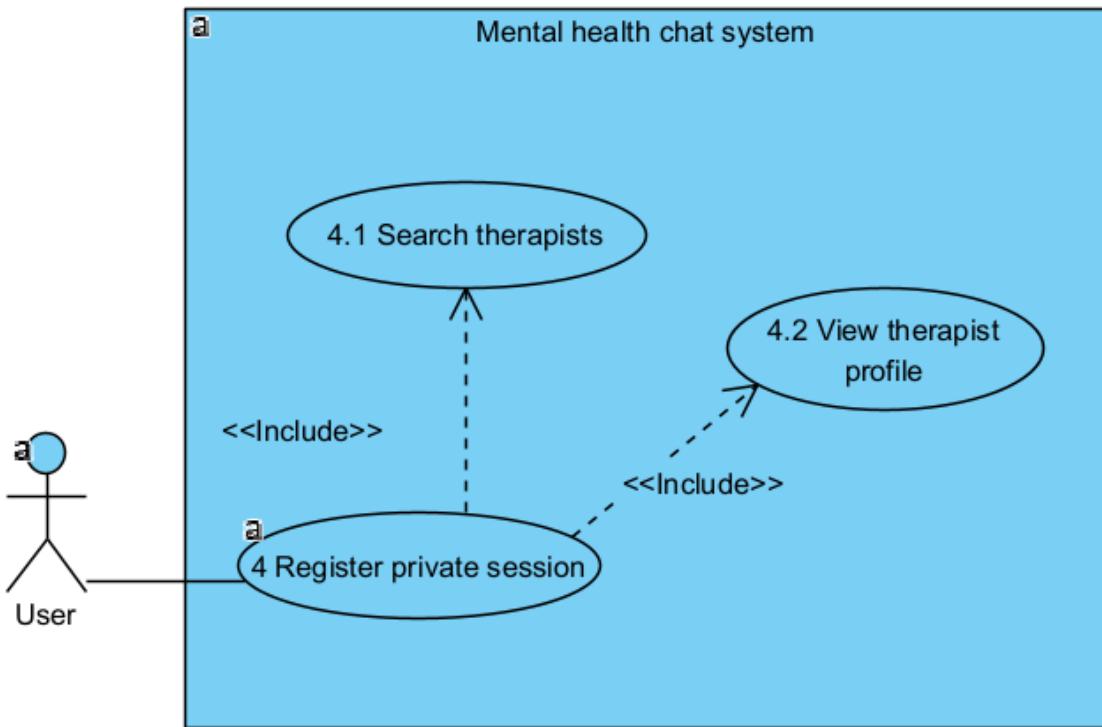
Bảng 2.4 Đặc tả use case quản lý thông tin tài khoản

Tên use case	Manage personal account
--------------	-------------------------

Mã use case	3
Mô tả	Là người dùng, tôi muốn quản lý các thông tin về tài khoản cá nhân trên hệ thống
Actor	Người dùng
Trigger	Người dùng chọn quản lý tài khoản (manage personal profile)
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Giao diện hiện đầy đủ các thành phần của trang
Luồng cơ bản	<p>1. Người dùng chọn nút quản lý tài khoản (manage personal profile) trên thanh điều hướng phụ.</p> <p>2. Giao diện hiện lên trang quản lý tài khoản, chưa thanh điều hướng con gồm:</p> <ul style="list-style-type: none"> • Thông tin cá nhân (Profile) • Thay đổi mật khẩu (Change password) • Xoá tài khoản (Delete account) • Lịch sử đăng ký/đăng nhập <p>3. Người dùng chọn lựa chọn trên thanh điều hướng con</p> <p>4. Nếu người dùng chọn trang thông tin cá nhân</p> <ol style="list-style-type: none"> a. Giao diện hiện lên trang chỉnh sửa thông tin, gồm nút submit và các trường: <ul style="list-style-type: none"> ○ Ảnh đại diện ○ Họ và tên ○ Email ○ Số điện thoại ○ Tiểu sử ○ Ngày sinh <p>Người dùng có thể bổ sung đăng ký làm chuyên gia, các trường nhập liệu bổ sung bao gồm:</p> <ul style="list-style-type: none"> ○ Chọn lĩnh vực điều trị ○ Học vấn: tên, loại bằng, chuyên ngành và thời gian học tập ○ Kinh nghiệm: tên công ty, vị trí, mô tả và thời gian làm việc ○ Chứng chỉ: tên, tổ chức cấp, ngày cấp, hạn và đường dẫn <p>b. Người dùng chỉnh sửa các thông tin cần thiết và chọn nút submit</p> <p>c. Hệ thống thông báo chỉnh sửa thành công và cập nhật thông tin của người dùng trên giao diện</p> <p>5. Nếu người dùng chọn trang đổi mật khẩu</p> <ol style="list-style-type: none"> a. Giao diện hiện lên trang thay đổi mật khẩu, gồm nút submit và các trường bắt buộc: <ul style="list-style-type: none"> ○ Mật khẩu cũ ○ Mật khẩu mới, cần khác mật khẩu cũ, ít nhất 8 kí tự, chứa kí tự đặc biệt, chữ viết hoa và số ○ Nhập lại mật khẩu mới

	<p>b. Người nhập đủ các trường nhập liệu và chọn nút submit</p> <p>c. Hệ thống thông báo chỉnh sửa thành công</p> <p>6. Nếu người dùng chọn xoá tài khoản</p> <p>a. Giao diện hiện lên trang xoá tài khoản, gồm trường nhập lại mật khẩu hiện tại, nút xoá và cảnh báo về thao tác xoá tài khoản</p> <p>b. Người nhập đủ các trường nhập liệu và chọn nút submit</p> <p>c. Hệ thống thông báo chỉnh sửa thành công</p> <p>7. Nếu người dùng chọn xem lịch sử đăng ký trị liệu</p> <p>a. Giao diện hiện lên trang lịch sử đăng ký của người dùng, bao gồm bảng với các cột: tên người trị liệu, ngày đăng ký, lần cập nhật cuối cùng, trạng thái đăng ký, ngày kết thúc và nút đánh giá</p> <p>b. Trên bản ghi đăng ký trị liệu với trạng thái kết thúc (FINISHED), người dùng chọn nút đánh giá (review)</p> <p>c. Giao diện hiện lên hộp thoại bao gồm mức sao đánh giá, ô nhập lời đánh giá người trị liệu và nút huỷ, submit.</p> <p>Giao diện hiện sẵn mức đánh giá và lời bình luận trước đây của người dùng (nếu có).</p> <p>d. Người dùng chọn sao từ 1-5 và viết lời đánh giá, sau đó chọn submit</p> <p>e. Hệ thống trả kết quả lưu đánh giá thành công.</p>
Ngoại lệ	<p>4c1. Số điện thoại đã được đăng ký trên tài khoản khác. Hệ thống thông báo cập nhật thất bại. <i>Quay lại bước 4b</i></p> <p>4c2. Email đã được đăng ký trên tài khoản khác. Hệ thống thông báo cập nhật thất bại. <i>Quay lại bước 4b</i></p> <p>5c. Mật khẩu cũ không chính xác. Hệ thống thông báo cập nhật thất bại. <i>Quay lại bước 5b</i></p> <p>6c. Mật khẩu hiện tại không chính xác. Hệ thống thông báo cập nhật thất bại. <i>Quay lại bước 6b</i></p>

d. Use case 4: Đăng kí phiên trị liệu cá nhân



Hình 2.10 Use case đăng kí phiên trị liệu cá nhân

Bảng 2.5 Đặc tả use case đăng kí phiên trị liệu cá nhân

Tên use case	Register private session
Mã use case	4
Mô tả	Là người dùng, tôi muốn đăng kí trị liệu với một chuyên gia trên hệ thống
Actor	Người dùng
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Hệ thống ghi nhận bản ghi đăng kí trị liệu của người dùng với người trị liệu trong cơ sở dữ liệu
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút xem chuyên gia (Therapist) trên thanh điều hướng chính Giao diện hiện lên danh sách nhà trị liệu, cùng các thành phần tìm kiếm: <ul style="list-style-type: none"> Các thành phần tìm kiếm <ul style="list-style-type: none"> Trường nhập từ khoá tìm kiếm theo tên chuyên gia Nút lọc mở rộng, khi được kích hoạt sẽ hiện thêm các trường: <ul style="list-style-type: none"> Bật/tắt lọc dữ liệu Lựa chọn các lĩnh vực trị liệu của chuyên gia Chọn vùng điểm đánh giá chuyên gia Chọn giới tính chuyên gia Chọn số năm kinh nghiệm Chọn ngày làm việc của chuyên gia
	Thông tin hiển thị mỗi chuyên gia trong danh sách: <ul style="list-style-type: none"> Họ tên, giới tính chuyên gia

	<ul style="list-style-type: none"> ○ Điểm đánh giá từ khách hàng ○ Học vấn gần nhất và số lượng các học vấn ○ Kinh nghiệm làm việc gần nhất và các kinh nghiệm ○ Sổ chứng chỉ đã đạt ○ Sổ khách hàng đã đăng ký ○ Các lĩnh vực trị liệu ○ Mô tả cá nhân <p>3. Người dùng nhập từ khoá, loại tìm và chọn các trường lọc (tuỳ chọn), sau đó ấn enter</p> <p>4. Giao diện hiện danh sách nhà trị liệu thoả mãn điều kiện tìm kiếm và lọc (nếu có)</p> <p>5. Người dùng chọn nút xem chi tiết (More) trên một nhà trị liệu</p> <p>6. Giao diện hiện lên trang hiển thị chi tiết các thông tin từ người trị liệu, bao gồm:</p> <ul style="list-style-type: none"> • Thông tin cá nhân: Ảnh đại diện, họ và tên, số điện thoại • Thông tin về chuyên gia: • Các lĩnh vực điều trị (Tag) • Học vấn: tên, loại bằng, chuyên ngành và thời gian học tập • Kinh nghiệm: tên công ty, vị trí, mô tả và thời gian làm việc • Chứng chỉ: tên, tổ chức cấp, ngày cấp, hạn và đường dẫn • Lịch làm việc các ngày trong tuần của người trị liệu • Các lời đánh giá từ người dùng về người trị liệu • Nút đăng ký <p>7. Người dùng chọn nút đăng ký chuyên gia</p> <p>8. Giao diện hiện lên hộp thoại nhập lời nhắn cho người trị liệu và nút submit.</p> <p>9. Người dùng nhập lời nhắn cho đơn đăng ký và chọn nút submit.</p> <p>10. Hệ thống gửi thông báo cho người trị liệu với tiêu đề: A muốn đăng ký phiên trị liệu cá nhân với bạn</p> <p>11. Giao diện thông báo đăng ký thành công</p>
Ngoại lệ	<p>10.1 Hệ thống thông báo đăng ký thất bại do người dùng hiện đang trong phiên trị liệu cá nhân với chuyên gia khác <i>Use case kết thúc</i></p> <p>10.2 Hệ thống thông báo đăng ký thất bại do hiện tại người trị liệu đã đạt giới hạn khách hàng (5) <i>Use case tiếp tục tại bước 3</i></p>

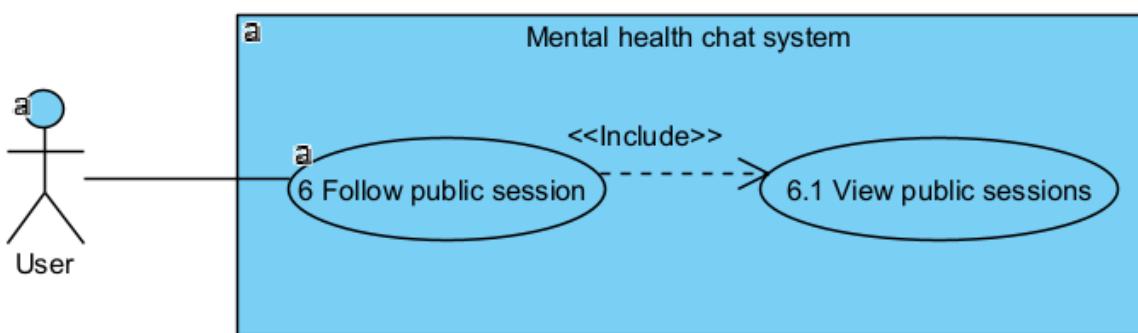
e. Use case 5: Xem lịch các phiên công khai/cá nhân trên hệ thống

Bảng 2.6 Đặc tả use case xem lịch các phiên công khai/cá nhân

Tên use case	View session schedules
Mã use case	5
Mô tả	Là người dùng, tôi muốn xem các lịch hẹn được tạo với người trị liệu trong phiên trị liệu cá nhân đã đăng ký, cũng như các phiên công khai đã theo dõi
Actor	Người dùng
Trigger	Người dùng chọn quản lý lịch trị liệu (my sessions)

Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Giao diện hiện đầy đủ các thành phần của trang
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút xem lịch trên hệ thống (my sessions) trên thanh điều hướng. Giao diện hiện lên trang lịch hẹn từ trước tới tương lai, mỗi thành phần trong danh sách gồm: <ul style="list-style-type: none"> Ngày, giờ trong lịch hẹn Tiêu đề của phiên công khai/phiên cá nhân với người trị liệu đã đăng ký Người dùng có thể bấm vào sự kiện trên lịch để xem chi tiết

f. Use case 6: Theo dõi phiên công khai



Hình 2.11 Use case theo dõi phiên trị liệu công khai

Bảng 2.7 Đặc tả use case theo dõi phiên công khai

Tên use case	Follow public session
Mã use case	6
Mô tả	Là người dùng, tôi theo dõi các phiên trị liệu công khai trên hệ thống
Actor	Người dùng
Trigger	Người dùng chọn xem các phiên trị liệu công khai (public sessions)
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Hệ thống ghi nhận bản ghi theo dõi phiên công khai mới của tài khoản người dùng trong cơ sở dữ liệu
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút xem các phiên trị liệu công khai (public sessions) trên thanh điều hướng. Tại đây chứa hai tab: <ul style="list-style-type: none"> Xem danh sách các phiên công khai (public sessions) – mặc định Xem danh sách các phiên đã theo dõi Giao diện hiển thị trang phiên trị liệu công khai, chứa danh sách tóm tắt các phiên trị liệu công khai trong tương lai, mỗi thành phần trong danh sách gồm: <ul style="list-style-type: none"> Ảnh đại diện, họ tên người tổ chức (người trị liệu) Tiêu đề và mô tả của phiên Hình thức tổ chức Địa chỉ của phiên

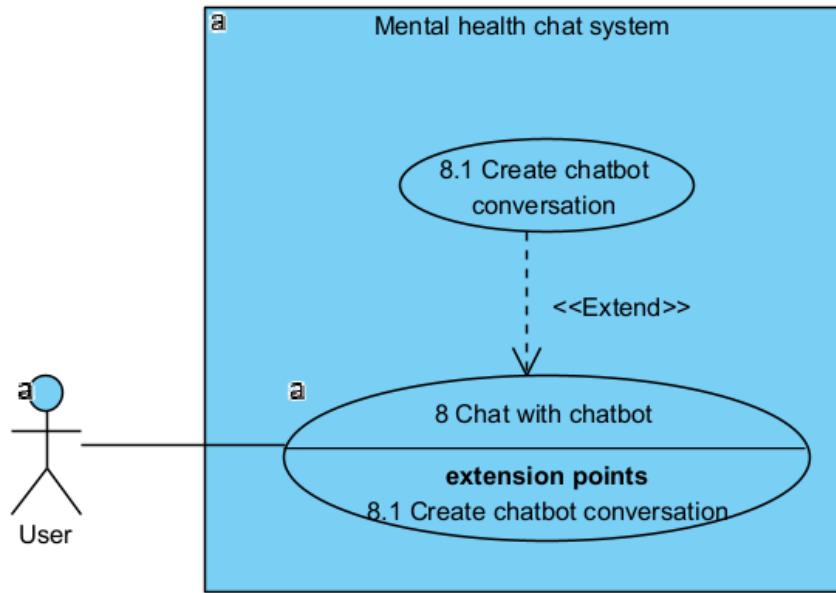
	<ul style="list-style-type: none"> • Ngày, giờ bắt đầu và kết thúc • Số lượng người theo dõi • Ảnh thu nhỏ về sự kiện • Nút theo dõi sự kiện (nếu người dùng không phải là người tổ chức) <ol style="list-style-type: none"> 3. Người dùng theo dõi một sự kiện 4. Giao diện thông báo đăng ký thành công và cập nhật trạng thái đăng ký trên sự kiện.
--	--

g. Use case 7: Quản lý thông báo cá nhân

Bảng 2.8 Đặc tả use case quản lý thông báo cá nhân

Tên use case	Manage personal notifications
Mã use case	7
Mô tả	Là người dùng, tôi muốn xem/xoá các thông báo cá nhân trên hệ thống
Actor	Người dùng
Trigger	Người dùng chọn nút “inbox” xem các thông báo cá nhân
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Giao diện hiển thị đầy đủ các thành phần
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng chọn nút inbox trên thanh công cụ của giao diện chính 2. Giao diện hiện lên giao diện các thông báo cá nhân và nút lọc tùy chọn: tất cả và chưa đọc và danh sách các thông báo cá nhân, mỗi thành phần bao gồm: <ul style="list-style-type: none"> • Tiêu đề • Thông báo • Thời gian nhận thông báo • Biểu tượng đại diện cho kiểu thông báo • Nút mở rộng chứa lựa chọn xoá thông báo 3. Người dùng thực hiện thao tác trên giao diện

h. Use case 8: Trò chuyện với chatbot



Hình 2.12 Use case trò chuyện với chatbot

Bảng 2.9 Đặc tả use case trò chuyện với chatbot

Tên use case	Chat with chatbot
Mã use case	8
Mô tả	Là người dùng, tôi muốn trò chuyện trực tiếp về vấn đề tâm lý với chatbot trên hệ thống
Actor	Người dùng
Trigger	Người dùng chọn nút chatbot chat trên thanh điều hướng
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Giao diện hiển thị đầy đủ các thành phần, tin nhắn hội thoại được lưu lại trong cơ sở dữ liệu
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút “chatbot chat” trên thanh điều hướng Giao diện hiện lên trang trò chuyện với chatbot, bao gồm danh sách các đoạn hội thoại đã tạo và nút tạo đoạn hội thoại mới Người dùng chọn một đoạn hội thoại hoặc chọn tạo đoạn hội thoại mới. Mỗi tin nhắn trong đoạn hội thoại bao gồm: <ul style="list-style-type: none"> Tên người gửi Nội dung văn bản của tin nhắn dưới dạng markdown Ngày gửi, trạng thái lỗi gửi (nếu có) Giao diện lên trường nhập câu hỏi, nút gửi và lịch sử đoạn hội thoại nếu chọn hội thoại cũ Người dùng nhập câu hỏi của mình và trường nhập liệu và chọn nút gửi Giao diện hiển thị tin nhắn người dùng đã gửi và câu trả lời từ chatbot Quay lại bước 5 cho đến khi người dùng muốn kết thúc

i. Use case 9: Trò chuyện với người trị liệu

Bảng 2.10 Đặc tả use case trò chuyện với người trị liệu

Tên use case	Chat with therapist
Mã use case	9
Tiền điều kiện	<ul style="list-style-type: none"> Người dùng liệu/người trị liệu đã đăng nhập vào hệ thống Người dùng đã đăng ký phiên trị liệu và được người trị liệu chấp thuận
Hậu điều kiện	Giao diện hiển thị đầy đủ các thành phần, tin nhắn hội thoại được lưu lại trong cơ sở dữ liệu
Yêu cầu phi chức năng	NFR1. Người dùng và người trị liệu cần gửi và nhận tin nhắn thời gian thực và bất đồng bộ (gửi và nhận không diễn ra tuân tự)

Vai trò người dùng

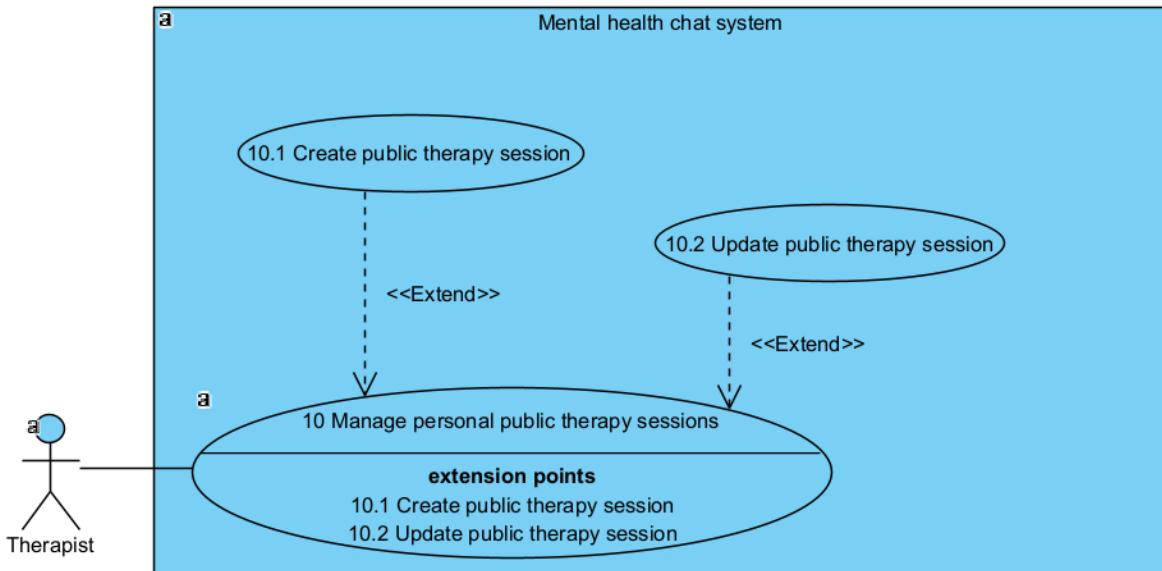
Mô tả	Là người dùng, tôi muốn trò chuyện trực tiếp về vấn đề tâm lý với người chuyên gia trên hệ thống
Actor	Người dùng
Trigger	Người dùng chọn nút “therapist chat” trên thanh điều hướng
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút “therapist chat” trên thanh điều hướng Giao diện hiện lên trang trò chuyện với người trị liệu, bao gồm danh sách các đoạn hội thoại đã tạo và nút tạo đoạn hội thoại mới Người dùng chọn đoạn hội thoại với người trị liệu hiện tại Giao diện lên trường nhập câu hỏi, nút gửi và lịch sử đoạn hội thoại nếu có, trong đó mỗi tin nhắn bao gồm: <ul style="list-style-type: none"> Họ tên, ảnh đại diện người gửi Nội dung văn bản của tin nhắn Ngày gửi, trạng thái tin nhắn: đã gửi, đã nhận, đã đọc, lỗi Người dùng nhập tin nhắn của mình và trường nhập liệu và chọn nút gửi Giao diện hiển thị tin nhắn người dùng đã gửi Quay lại bước 5 cho đến khi người dùng muốn kết thúc

Vai trò người trị liệu

Mô tả	Là người trị liệu, tôi muốn trò chuyện trực tiếp với người đăng ký trị liệu về vấn đề tâm trên hệ thống
Actor	Người trị liệu
Trigger	Người trị liệu chọn nút “client chat” trên thanh điều hướng
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút “client chat” trên thanh điều hướng Giao diện hiện lên trang trò chuyện với những người đăng ký trị liệu, bao gồm danh sách các đoạn hội thoại đã tạo và nút tạo đoạn hội thoại mới Người trị liệu chọn đoạn hội thoại với một khách hàng Giao diện lên trường nhập câu hỏi, nút gửi và lịch sử đoạn hội thoại nếu có, trong đó mỗi tin nhắn bao gồm: <ul style="list-style-type: none"> Họ tên, ảnh đại diện người gửi Nội dung văn bản của tin nhắn, hiển thị dạng markdown Ngày gửi, trạng thái tin nhắn: đã gửi, đã nhận, đã đọc, lỗi Người dùng nhập tin nhắn của mình và trường nhập liệu và chọn nút gửi

- | | |
|--|--|
| | <p>6. Giao diện hiển thị tin nhắn người trị liệu đã gửi</p> <p>7. Quay lại bước 5 cho đến khi người trị liệu muốn kết thúc</p> |
|--|--|

j. Use case 10: Quản lý các phiên trị liệu công khai cá nhân



Hình 2.13 Use case quản lý các phiên trị liệu công khai cá nhân

Bảng 2.11 Đặc tả use case quản lý phiên trị liệu công khai cá nhân

Tên use case	Manage personal public sessions
Mã use case	10
Mô tả	Là người trị liệu, tôi muốn xem/tạo/sửa các phiên trị liệu công khai đã tạo trên hệ thống
Actor	Người trị liệu
Trigger	Người trị liệu chọn quản lý phiên công khai (My public sessions)
Tiền điều kiện	Người trị liệu đã đăng nhập vào hệ thống
Hậu điều kiện	Giao diện hiện đầy đủ các thành phần của trang
Luồng cơ bản	<ol style="list-style-type: none"> Người dùng chọn nút quản lý phiên trị liệu công khai cá nhân (My public sessions) trên thanh điều hướng. Giao diện hiển thị trang phiên trị liệu công khai cá nhân, chứa nút tạo phiên công khai và danh sách tóm tắt các phiên trị liệu công khai đã tạo, mỗi thành phần trong danh sách gồm: <ul style="list-style-type: none"> Tiêu đề của phiên Mô tả của phiên Hình thức tổ chức: Online, offline Địa chỉ của phiên Ngày, giờ bắt đầu và giờ kết thúc Ảnh thu nhỏ về sự kiện Số lượng người theo dõi và họ tên từng người Nút chỉnh sửa

	<p>3. Nếu người dùng chọn nút tạo phiên công khai trên trang.</p> <ul style="list-style-type: none"> a. Hộp thoại tạo phiên công khai hiện lên, gồm nút submit và người dùng nhập các trường thông tin: <ul style="list-style-type: none"> o Tiêu đề của phiên o Mô tả của phiên o Hình thức tổ chức: Online, offline o Địa chỉ của phiên o Ngày, giờ bắt đầu và giờ kết thúc (ngày bắt đầu phải cách ngày hiện tại ít nhất 3 ngày) o Ảnh thu nhỏ về sự kiện (tuỳ chọn) b. Người dùng hoàn tất các trường thông tin và chọn nút submit c. Giao diện thông báo tạo phiên công khai thành công và cập nhật lại danh sách phiên công khai đã tạo <p>4. Nếu người dùng cập nhật trên một phiên công khai trên trang.</p> <ul style="list-style-type: none"> a. Hộp thoại tạo phiên công khai hiện lên, gồm nút submit và người dùng nhập các trường thông tin: <ul style="list-style-type: none"> o Tiêu đề của phiên o Mô tả của phiên o Hình thức tổ chức: Online, offline o Địa chỉ của phiên o Ngày, giờ bắt đầu và giờ kết thúc (ngày bắt đầu phải cách ngày hiện tại ít nhất 3 ngày) o Ảnh thu nhỏ về sự kiện (tuỳ chọn) o Nút hoãn sự kiện b. Người dùng hoàn tất các trường thông tin và chọn nút submit c. Giao diện thông báo tạo phiên công khai thành công và cập nhật lại danh sách phiên công khai đã tạo. d. Hệ thống gửi thông báo cho những người đã theo dõi về thay đổi của phiên với tiêu đề: Phiên công khai tổ chức bởi A đã được cập nhật
--	--

k. Use case 11: Quản lý đăng ký phiến trị liệu cá nhân

Bảng 2.12 Đặc tả use case quản lý đăng ký phiến trị liệu cá nhân

Tên use case	Manage client requests for therapy
Mã use case	11
Mô tả	Là người trị liệu, tôi muốn xem và quản lý các đăng ký trị liệu với bản thân
Actor	Người trị liệu
Trigger	Người trị liệu chọn đơn đăng ký “Client registration” quản lý đơn đăng ký phiến trị liệu cá nhân từ khách hàng
Tiền điều kiện	Người trị liệu đã đăng nhập vào hệ thống
Hậu điều kiện	Hệ thống ghi nhận bản ghi cập nhật trong cơ sở dữ liệu
Luồng cơ bản	1. Người dùng chọn đơn đăng ký “Client registration” trên thanh điều hướng.

	<p>2. Giao diện hiện ra trang quản lý đơn, chưa danh sách các đơn khách hàng đăng kí cho phiên cá nhân. Mỗi thành phần trong danh sách bao gồm:</p> <ul style="list-style-type: none"> • Tên, email ảnh đại diện khách hàng • Trạng thái đăng kí: đang chờ, chấp thuận, từ chối, hoàn thành, huỷ bỏ • Thời gian đăng kí • Lời giới thiệu từ khách hàng • Lời nhắn từ người trị liệu • Nút hành động <p>3. Người dùng chọn nút hành động trên một đơn đăng kí</p> <p>4. Giao diện hiện lên hộp thoại gồm: trường chọn chấp thuận, từ chối, hoàn thành, huỷ bỏ; trường nhập lời nhắn, nút xác nhận và huỷ bỏ</p> <p>5. Người dùng chọn trạng thái mới, nhập lời nhắn và chọn xác nhận</p> <p>6. Giao diện hiện thông báo thành công và cập nhật danh sách đã đăng kí</p>
Ngoại lệ	<p>6.1 Giao diện hiện thông báo thất bại và lý do khi người trị liệu chọn chấp thuận nhưng đã đủ số khách hàng tối đa ở thời điểm hiện tại</p> <p><i>Use case kết thúc</i></p>

1. Use case 12: Quản lý lịch các phiên trị liệu cá nhân

Bảng 2.13 Đặc tả use case quản lý lịch phiên trị liệu cá nhân

Tên use case	Manage private session schedules
Mã use case	12
Mô tả	Là người trị liệu, tôi muốn xem/thêm/sửa/xoá các lịch hẹn trị liệu với các khách hàng đã đăng kí
Actor	Người trị liệu
Trigger	Người trị liệu chọn quản lý lịch “Manage schedules”
Tiền điều kiện	Người trị liệu đã đăng nhập vào hệ thống
Hậu điều kiện	Hệ thống ghi nhận bản ghi cập nhật trong cơ sở dữ liệu
Luồng cơ bản	<p>1. Người dùng chọn quản lý lịch “Manage schedules” trên thanh điều hướng.</p> <p>2. Giao diện hiện ra trang quản lý lịch hẹn khách hàng, chưa danh sách các lịch đã tạo các cho phiên cá nhân, được nhóm theo ngày. Mỗi thành phần trong danh sách bao gồm:</p> <ul style="list-style-type: none"> • Ngày, thời gian bắt đầu, thời gian kết thúc • Tên ảnh đại diện khách hàng • Ghi chú của người trị liệu • Hình thức (online/offline) và địa chỉ trị liệu • Nút cập nhật, xoá <p>3. Người dùng thao tác trên giao diện</p>

m. Use case 13: Quản lý lịch làm việc cá nhân

Bảng 2.14 Đặc tả use case quản lý lịch làm việc cá nhân

Tên use case	Manage availability time
Mã use case	13
Mô tả	Là người trị liệu, tôi muốn xem/thêm/sửa các thời gian làm việc cá nhân
Actor	Người trị liệu
Trigger	Người trị liệu chọn quản lý thời gian làm việc “Manage availability time”
Tiền điều kiện	Người trị liệu đã đăng nhập vào hệ thống
Hậu điều kiện	Hệ thống ghi nhận bản ghi cập nhật trong cơ sở dữ liệu
Luồng cơ bản	<p>1. Người dùng chọn quản lý thời gian làm việc “Manage availability time” trên thanh điều hướng.</p> <p>2. Giao diện hiện ra trang sắp xếp lịch, gồm một nút lưu và 2 tab:</p> <ul style="list-style-type: none"> Sắp xếp thời gian làm việc các ngày trong tuần. Bao gồm thanh chọn lọc theo ngày trong tuần và danh sách các mốc thời gian đã đăng ký, được nhóm theo ngày. Mỗi mốc thời gian có nút xoá. Tạo mới gồm các trường: chọn ngày trong tuần, thời gian bắt đầu, thời gian kết thúc Quản lý các ngày đặc biệt, sẽ ghi đè thời gian làm việc trong tuần. Bao gồm lịch chọn ngày nút thêm lịch và danh sách các mốc thời gian ghi đè đã tạo. Mỗi thành phần trong danh sách chứa nút xoá Tạo mới gồm các trường: chọn ngày trong năm, thời gian bắt đầu, kết thúc và mô tả (tùy chọn). <p>3. Người dùng thao tác trên giao diện, sau đó chọn lưu để lưu thay đổi đã thực hiện trong trang.</p> <p>4. Giao diện thông báo lưu các thay đổi thành công.</p>

2.2.4. Biểu đồ lớp phân tích

Từ yêu cầu hệ thống 2.2.1, cách thực thể được trích xuất bao gồm:

Danh từ chỉ con người

- User: Nêu trong yêu cầu bài toán, đại diện cho người dùng (client/therapist).
- Role: Sinh ra để quản lý vai trò (client, therapist, etc.).
- UserRole: Sinh ra từ mối quan hệ nhiều-nhiều giữa User và Role.

Danh từ chỉ nội dung

- Post: Nêu trong yêu cầu, đại diện cho bài viết của người dùng.
- Notification: Nêu trong yêu cầu, thông báo liên quan đến người dùng.
- Review: Nêu trong yêu cầu, đại diện cho đánh giá của khách hàng dành cho therapist.

Danh từ chỉ giao tiếp (trò chuyện)

- Conversation: Nêu trong yêu cầu, đại diện cho phiên trò chuyện (1-1 hoặc với AI).
- Message: Nêu trong yêu cầu, đại diện cho tin nhắn trong mỗi phiên trò chuyện.

Danh từ chỉ lịch trình

- AvailabilityTemplate: Sinh ra để quản lý lịch làm việc cố định của therapist.
- AvailabilityOverride: Sinh ra để quản lý thay đổi lịch làm việc đặc biệt.

Danh từ chỉ phiên trị liệu

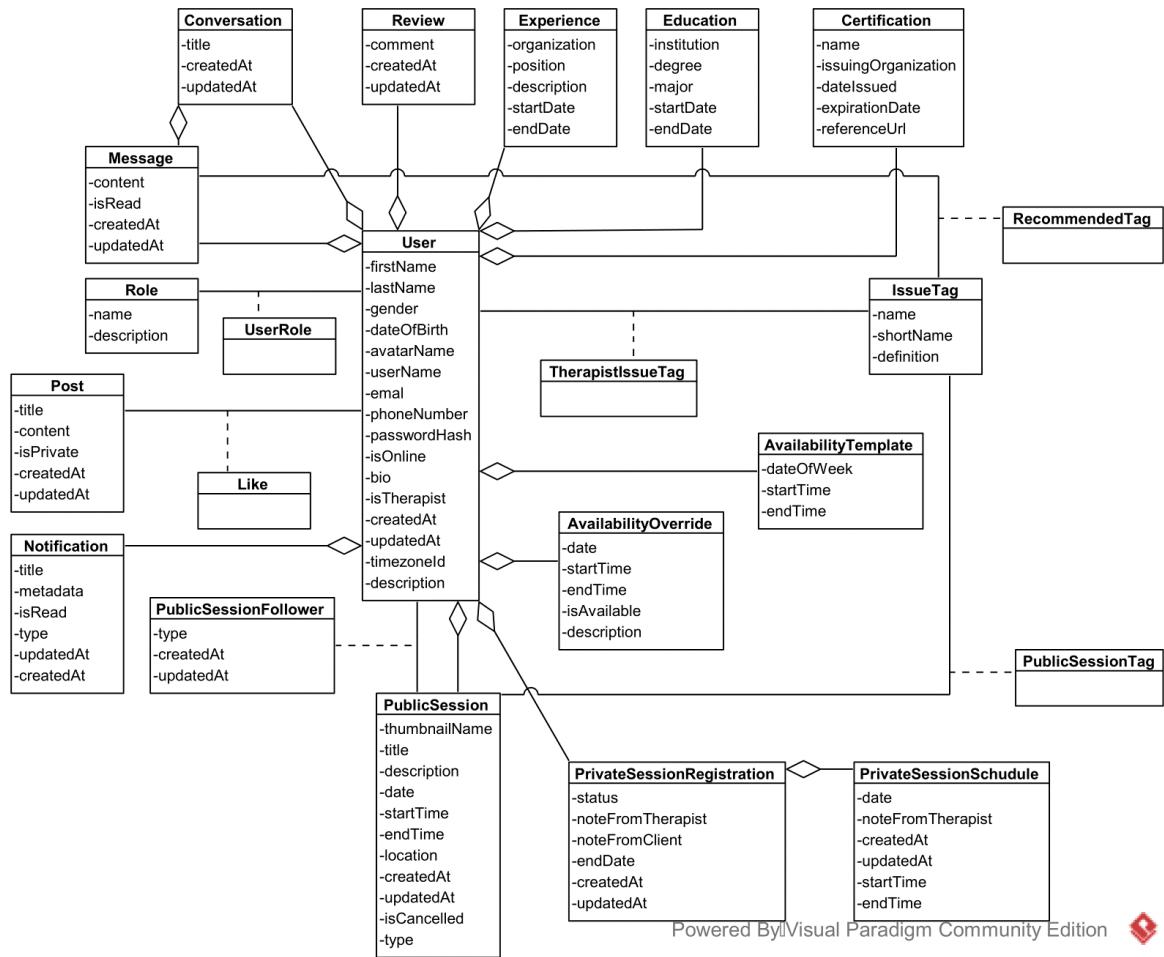
- PublicSession: Nêu trong yêu cầu, phiên trị liệu nhóm công khai của therapist.
- PublicSessionFollower: Sinh ra từ mối quan hệ n-n giữa User và PublicSession (người tham gia).
- PrivateSessionRegistration: Nêu trong yêu cầu, đại diện cho yêu cầu trị liệu cá nhân.
- PrivateSessionSchedule: Nêu trong yêu cầu, lịch trình chi tiết của phiên trị liệu cá nhân.

Danh từ chỉ vấn đề/từ khóa

- IssueTag: Nêu trong yêu cầu, đại diện cho vấn đề tâm lý được hỗ trợ.
- RecommendedTag: Sinh ra để cá nhân hóa trải nghiệm từ AI (gợi ý từ khóa).

Danh từ chỉ thông tin chuyên môn (Therapist)

- Experience: Nêu trong yêu cầu, kinh nghiệm làm việc.
- Education: Nêu trong yêu cầu, học vấn.
- Certification: Nêu trong yêu cầu, chứng chỉ hành nghề.

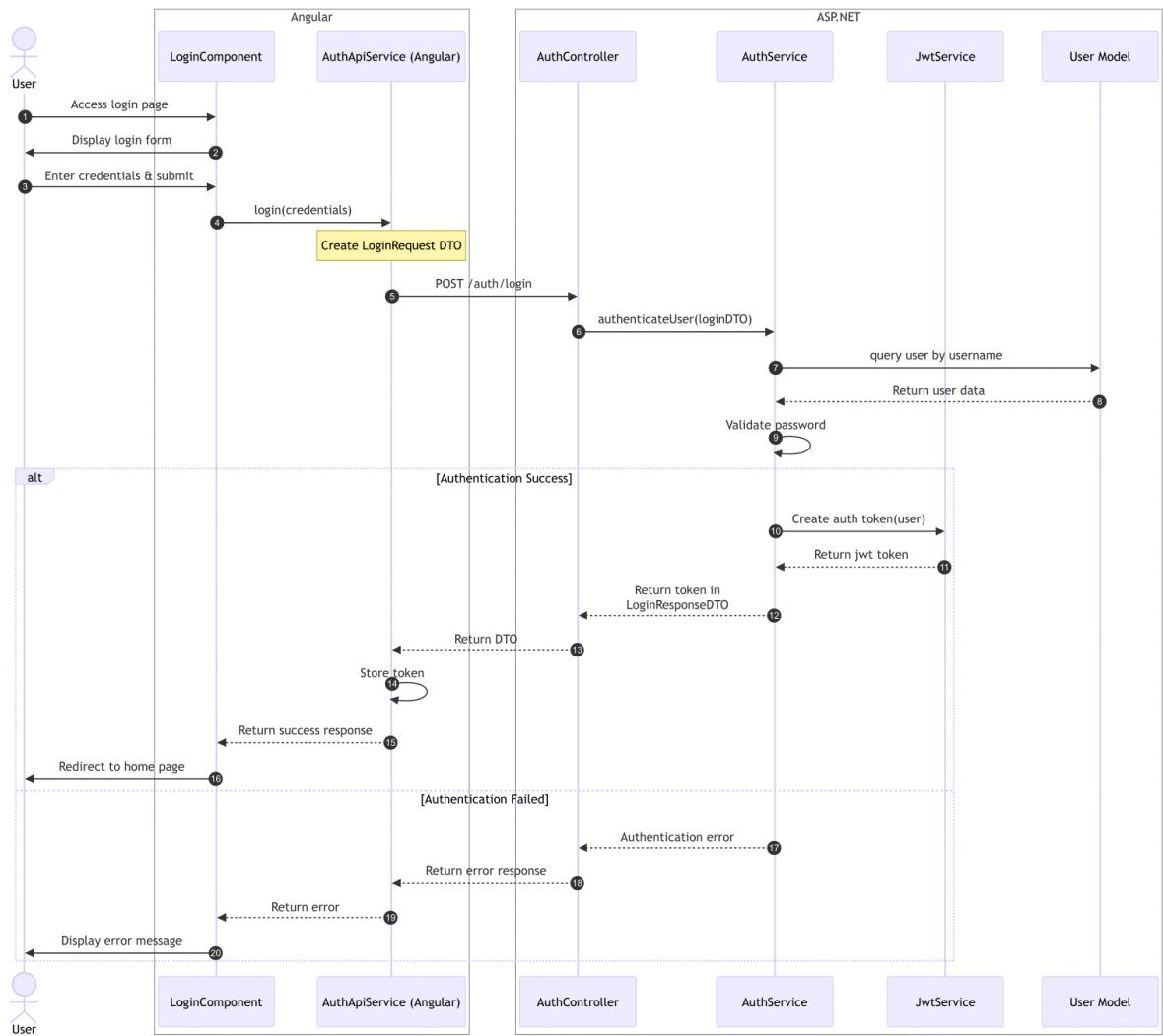


Hình 2.14 Sơ đồ lớp thực thể và thuộc tính trích xuất

2.3. Thiết kế hệ thống

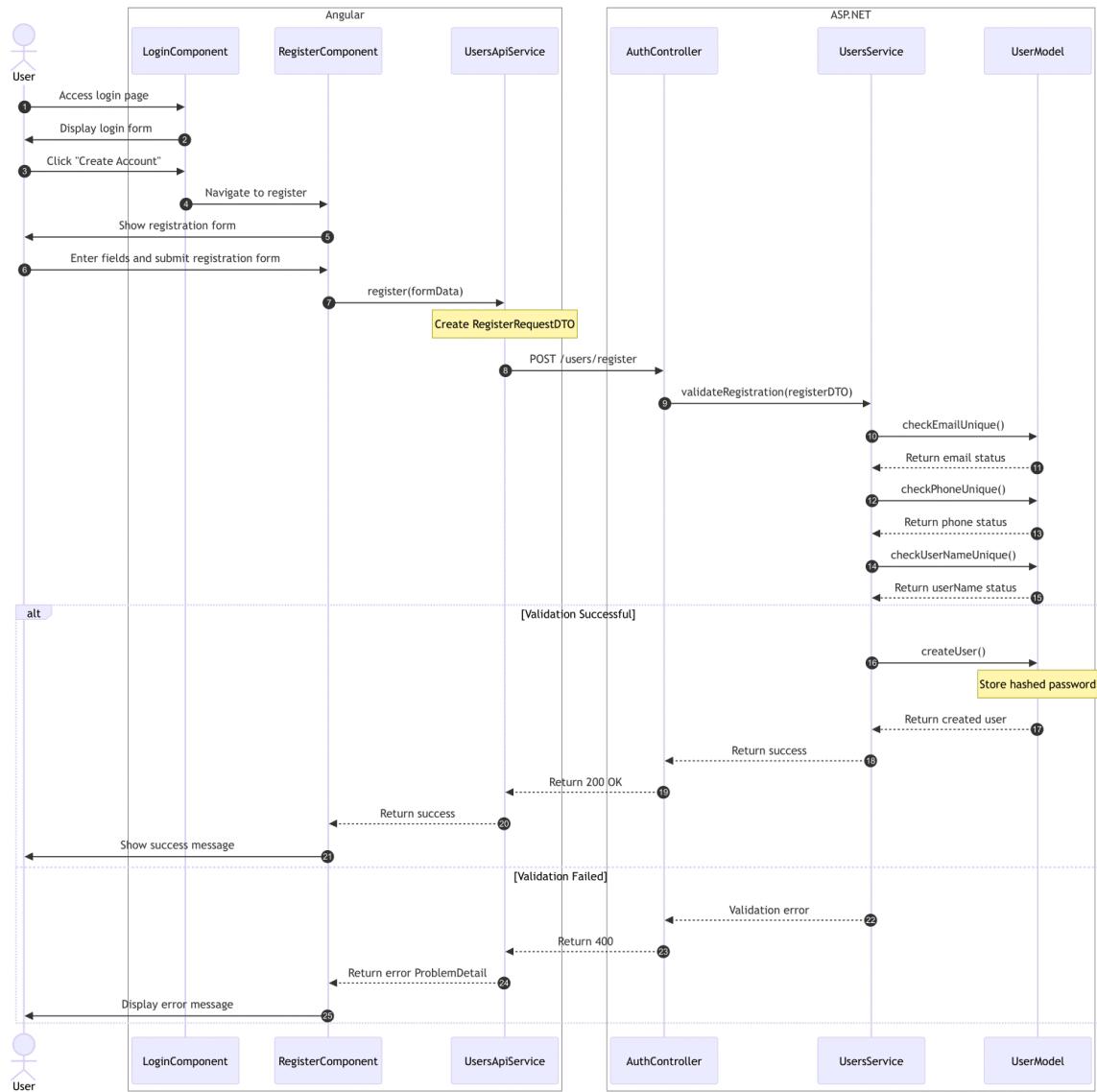
2.3.1. Sơ đồ tuần tự

a. Use case 1: Đăng nhập



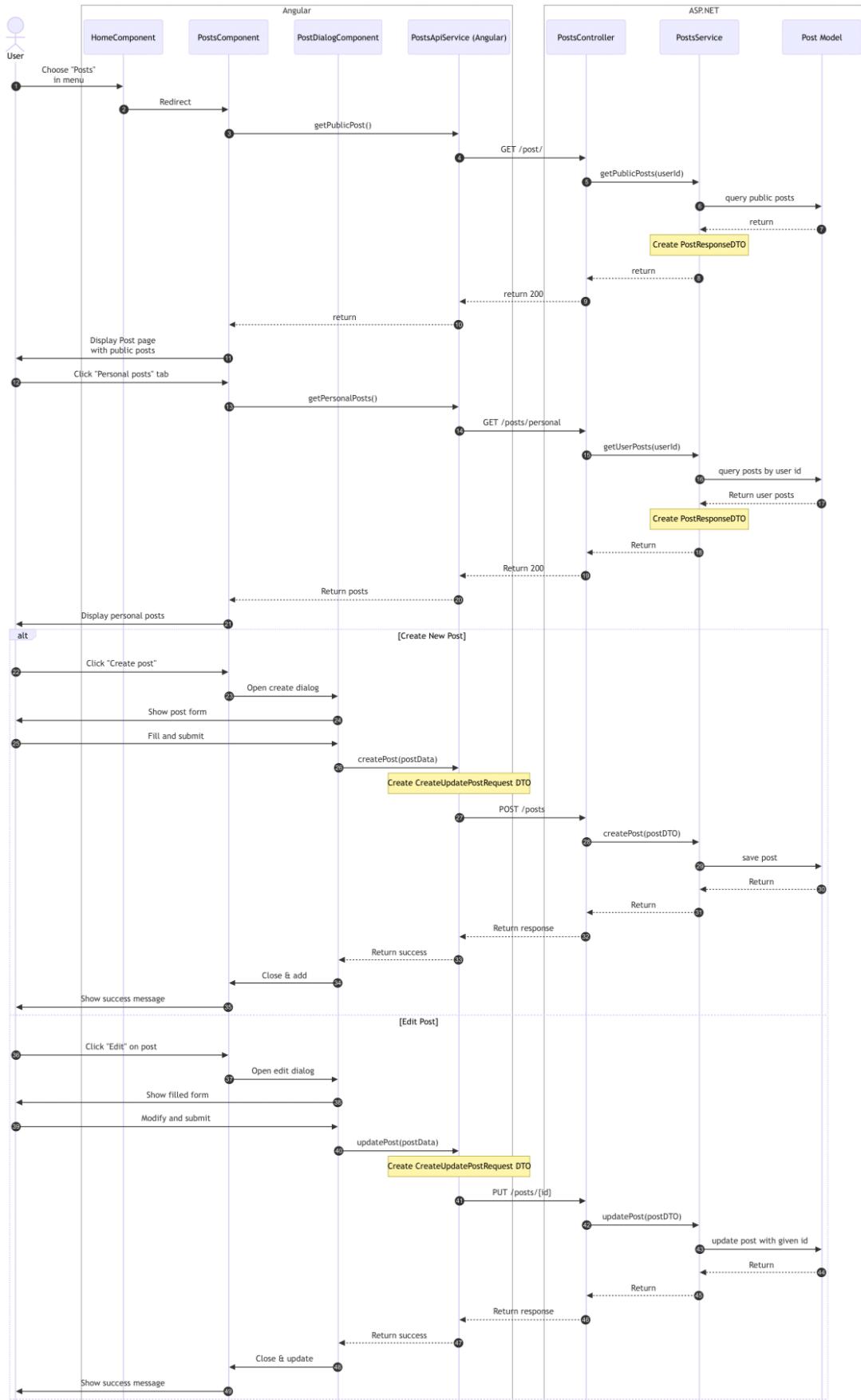
Hình 2.15 Sơ đồ tuần tự use case đăng nhập

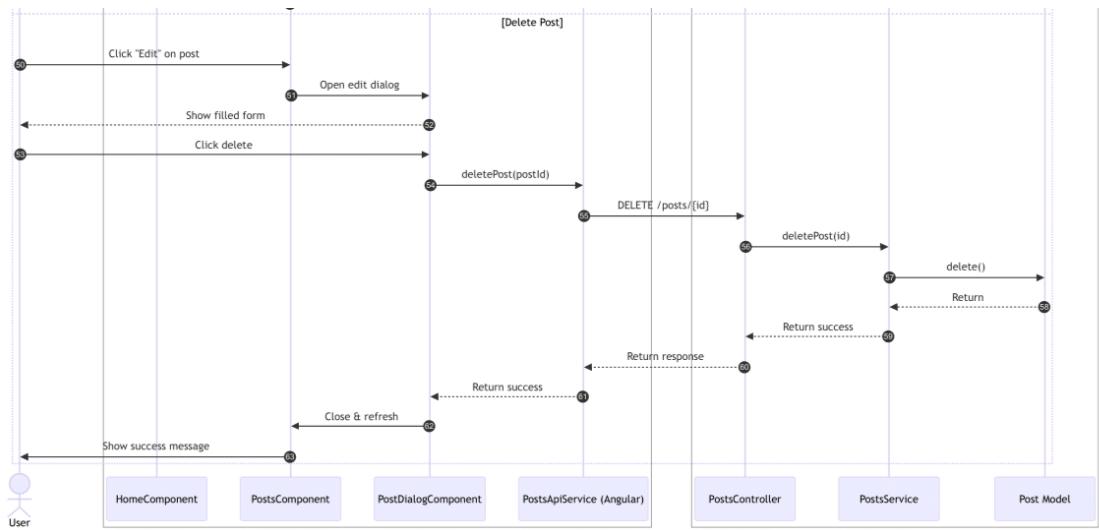
Use case 1.1: Đăng kí



Hình 2.16 Sơ đồ tuần tự use case đăng kí

b. Use case 2: Xem bài đăng

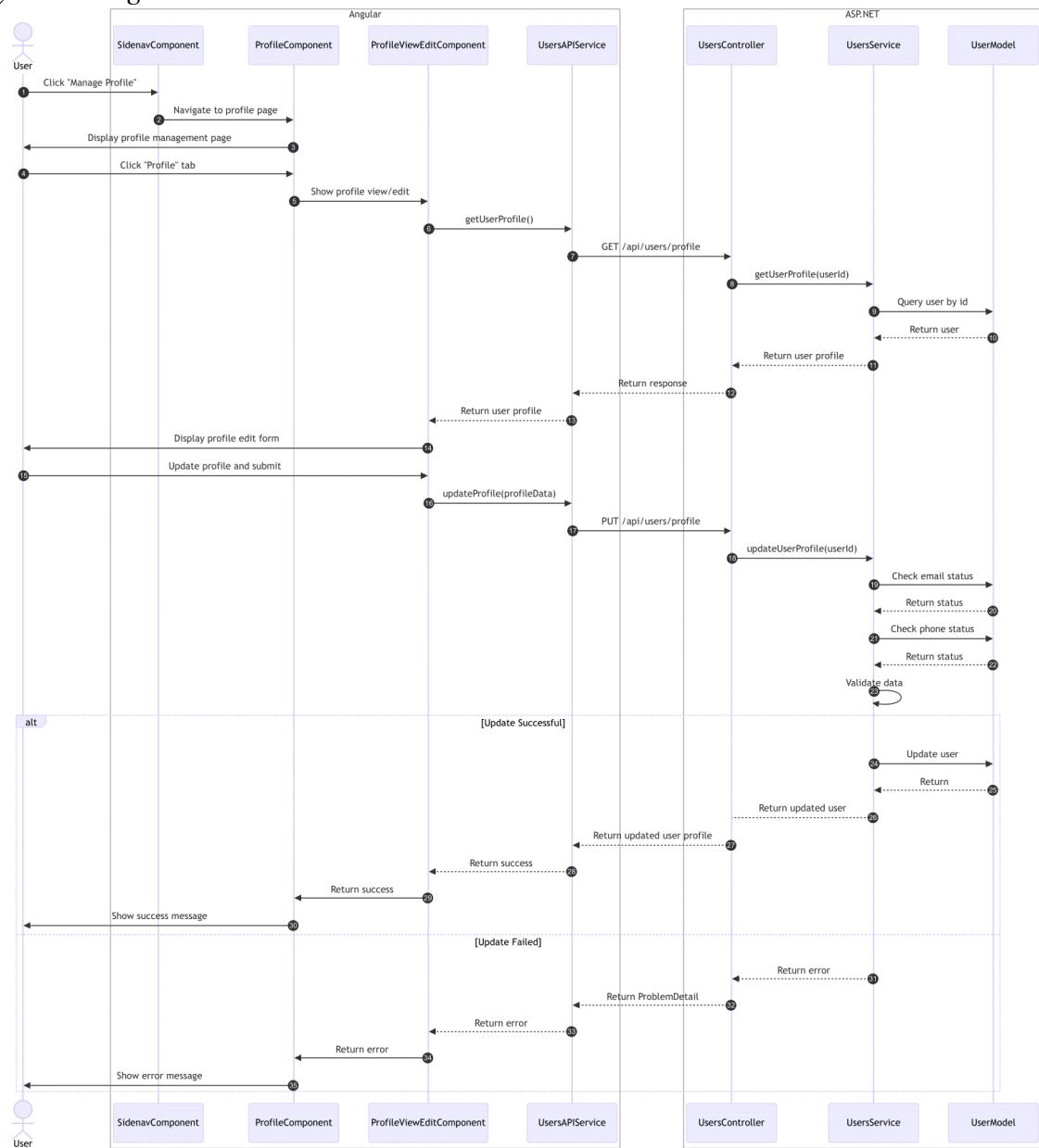




Hình 2.17 Sơ đồ tuần tự use case xem bài đăng

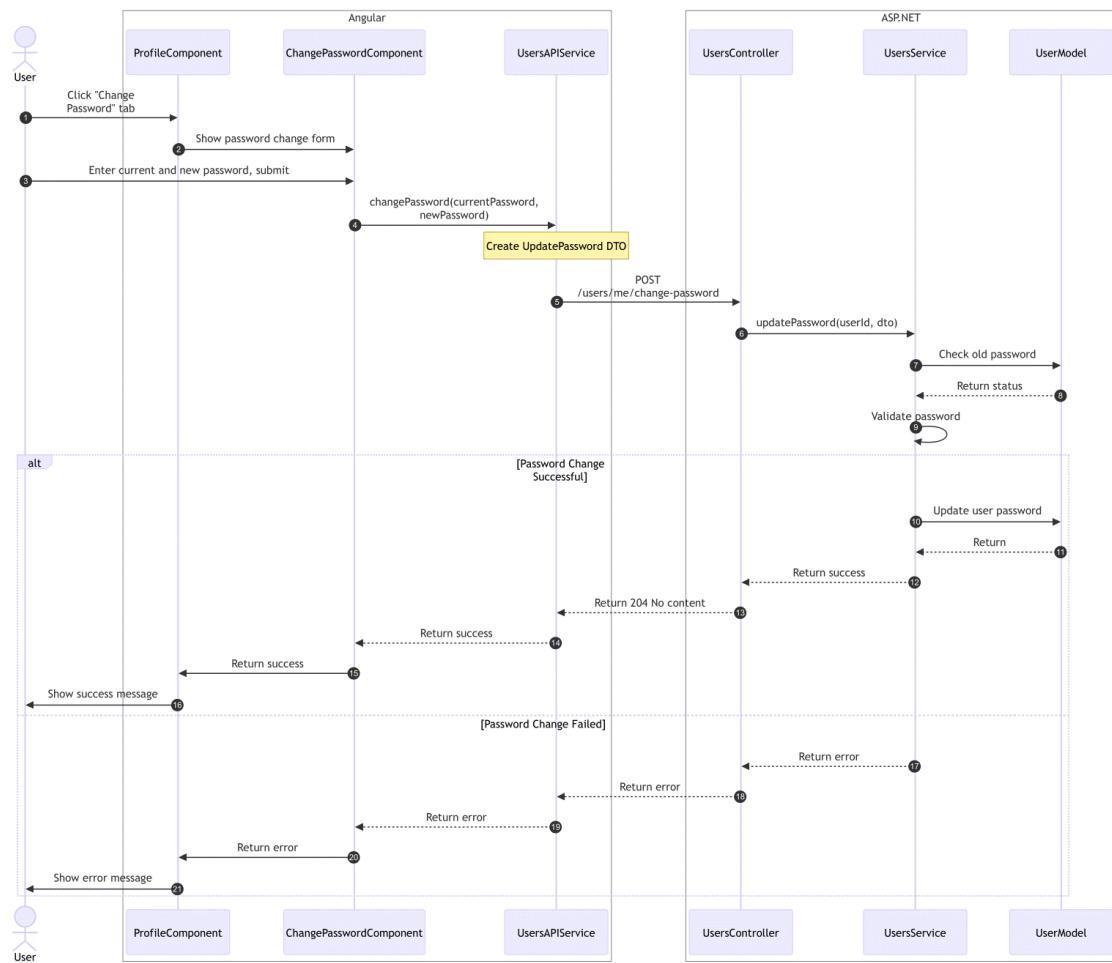
c. Use case 3: Quản lý thông tin cá nhân

Thay đổi thông tin cá nhân



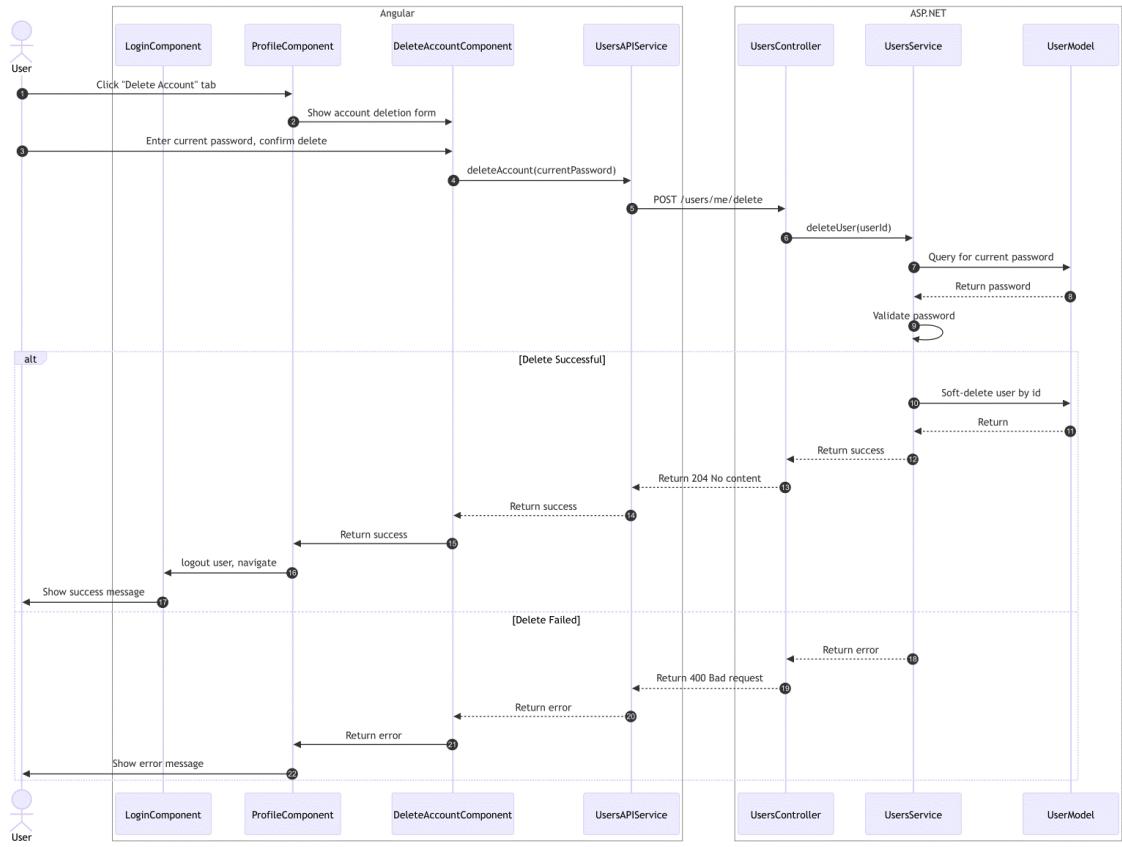
Hình 2.18: Sơ đồ tuần tự use case cập nhật thông tin cá nhân

Thay đổi mật khẩu



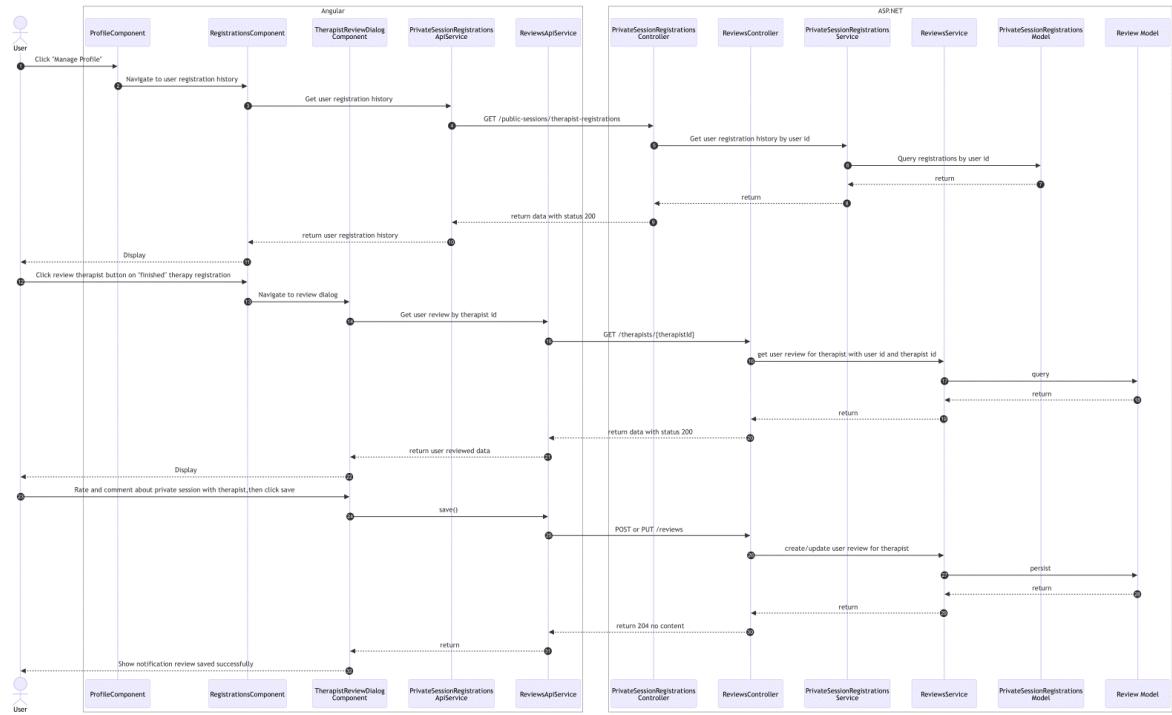
Hình 2.19 Sơ đồ tuần tự use case cập nhật mật khẩu

Xoá tài khoản



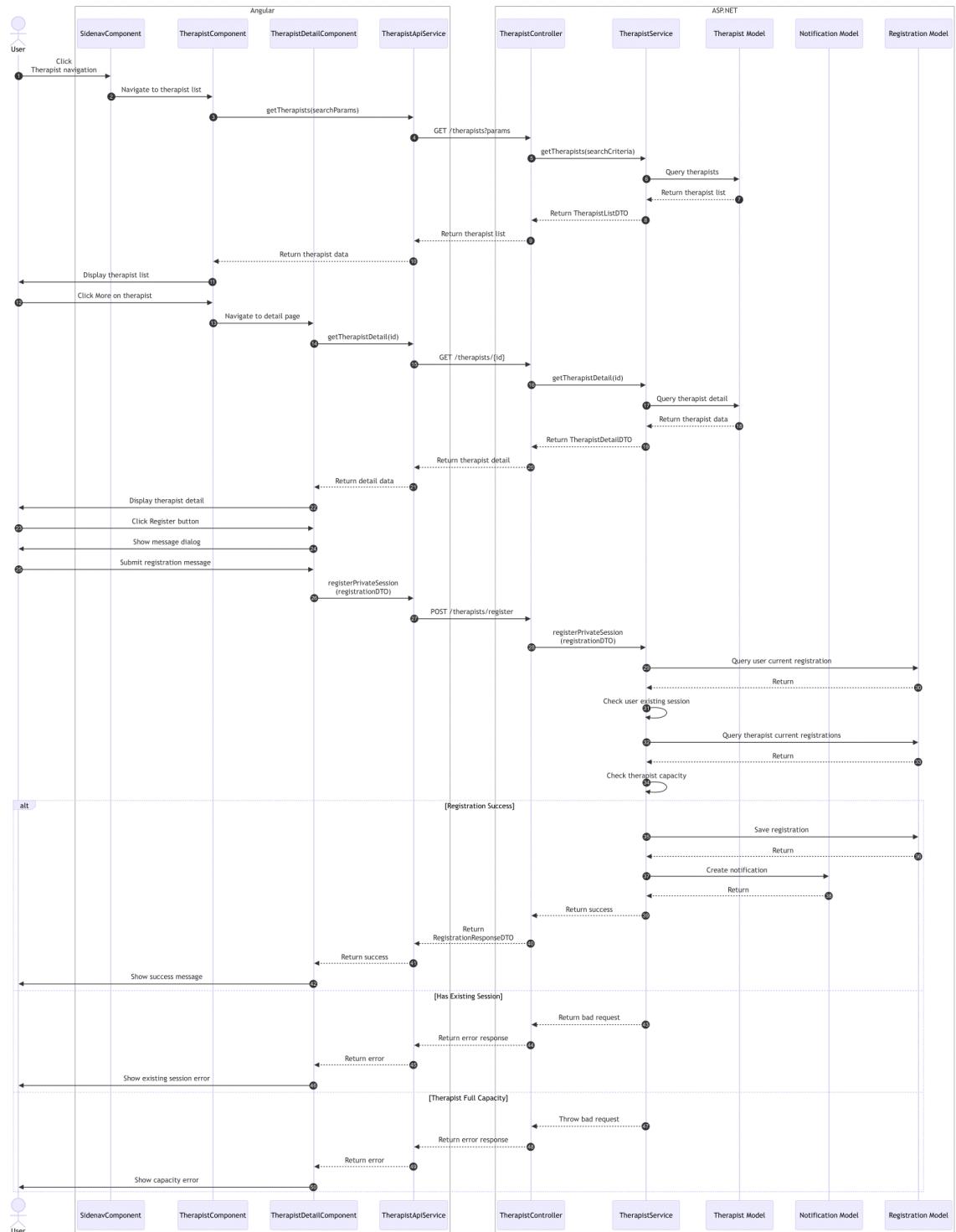
Hình 2.20 Sơ đồ tuần tự use case xoá tài khoản

Dánh giá người trị liệu



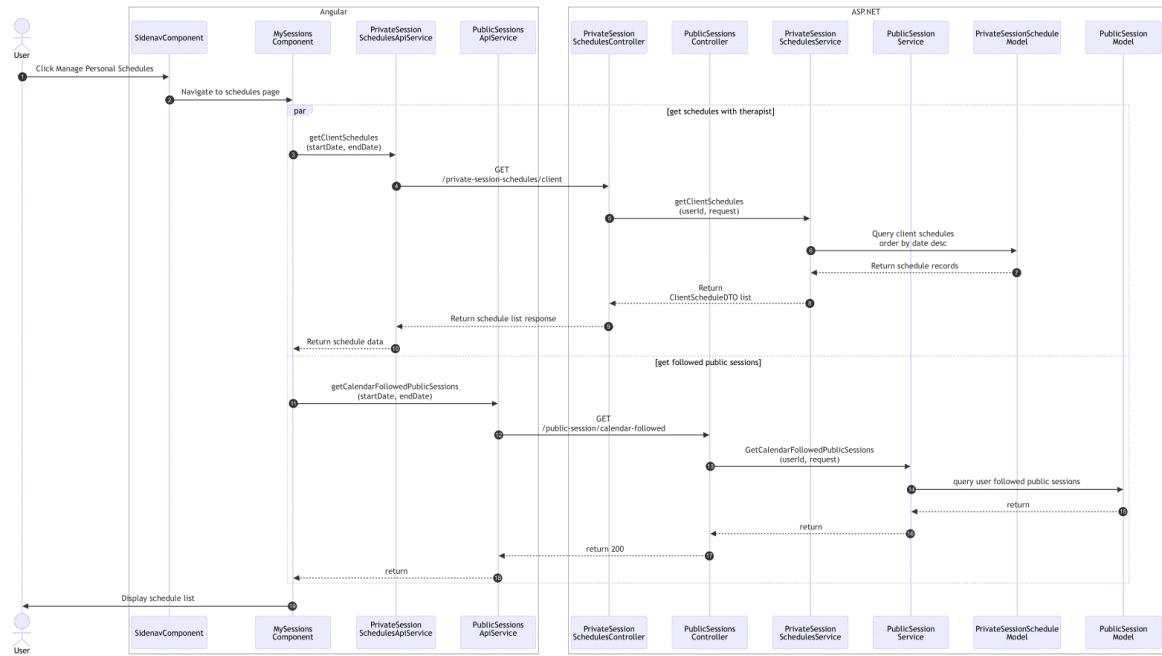
Hình 2.21 Sơ đồ tuần tự use case đánh giá người trị liệu

d. Use case 4: Đăng kí phiên trị liệu cá nhân



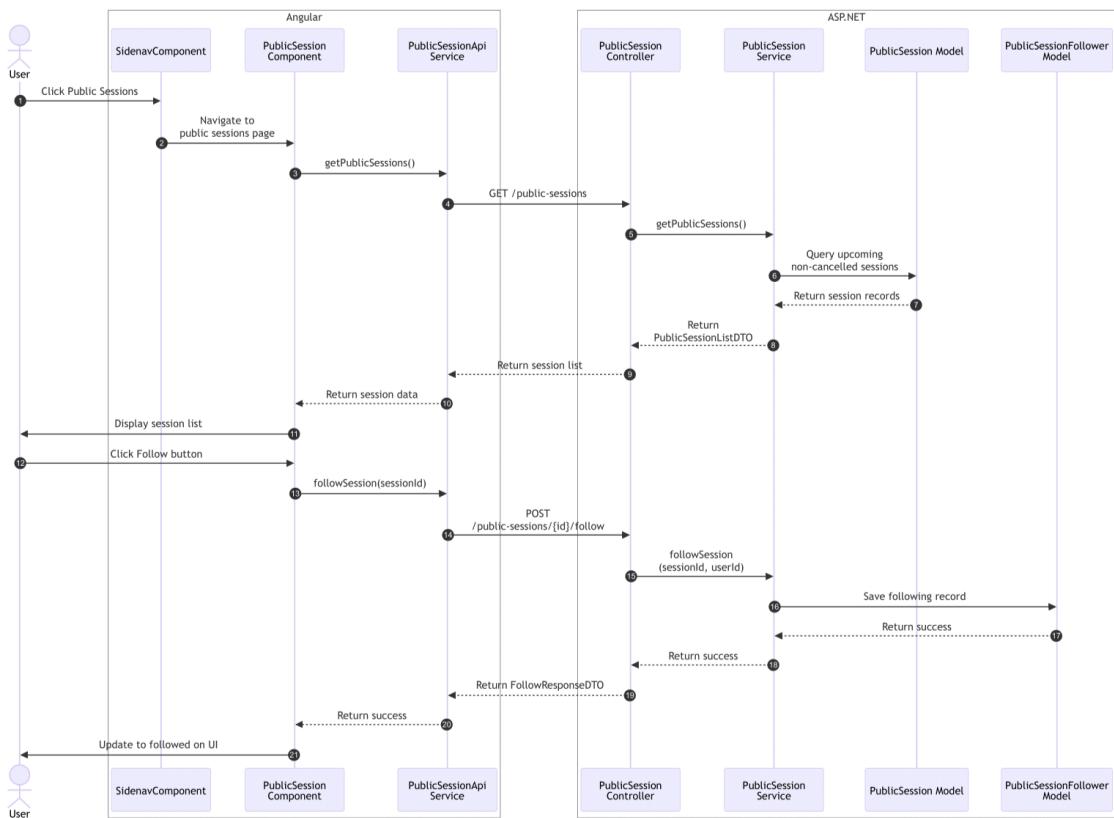
Hình 2.22 Sơ đồ tuần tự use case đăng kí phiên trị liệu cá nhân

e. Use case 5: Xem lịch các phiên công khai/cá nhân trên hệ thống



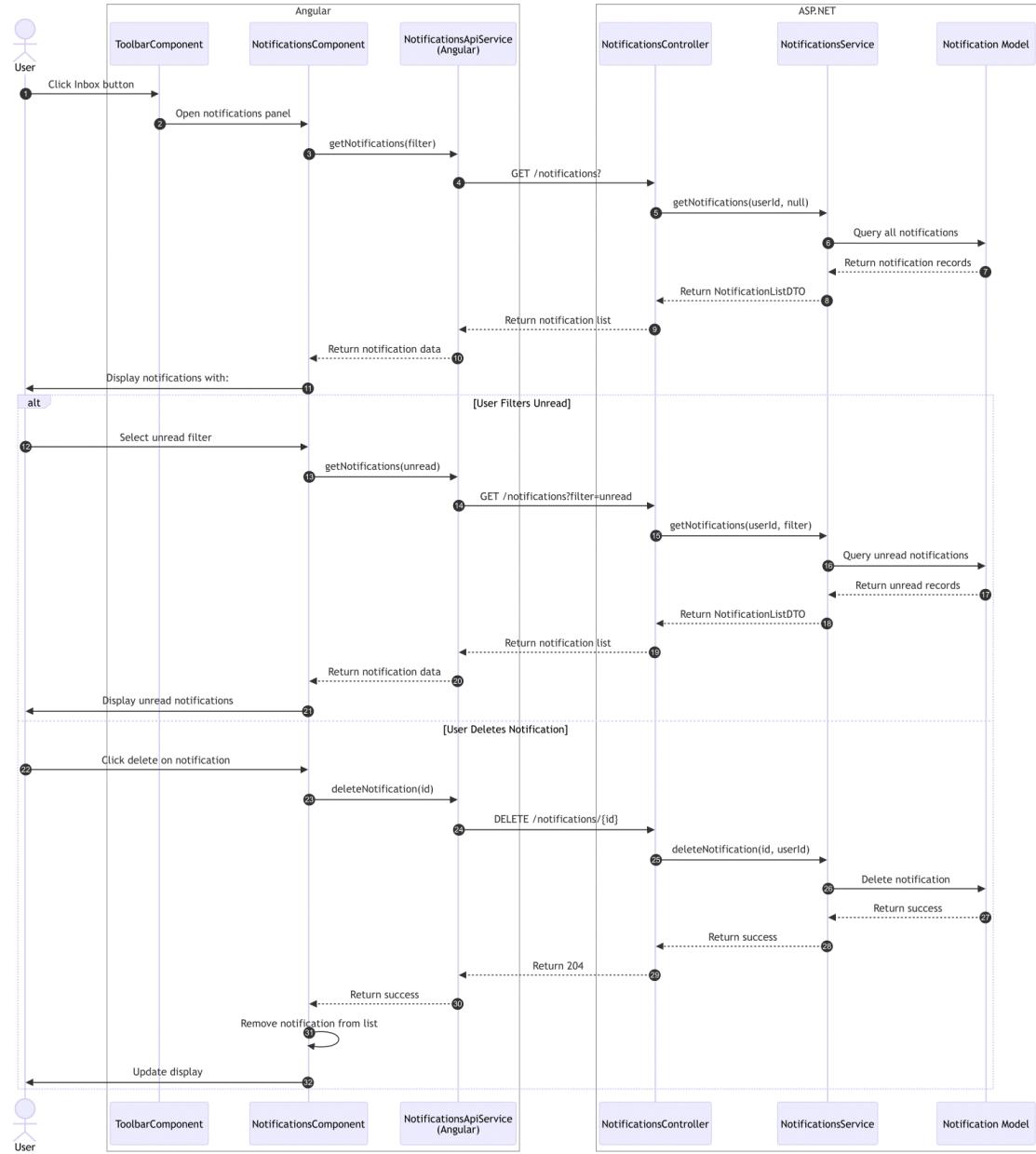
Hình 2.23 Sơ đồ tuần tự use case xem lịch các phiên công khai/cá nhân trên hệ thống

f. Use case 6: Theo dõi phiên công khai



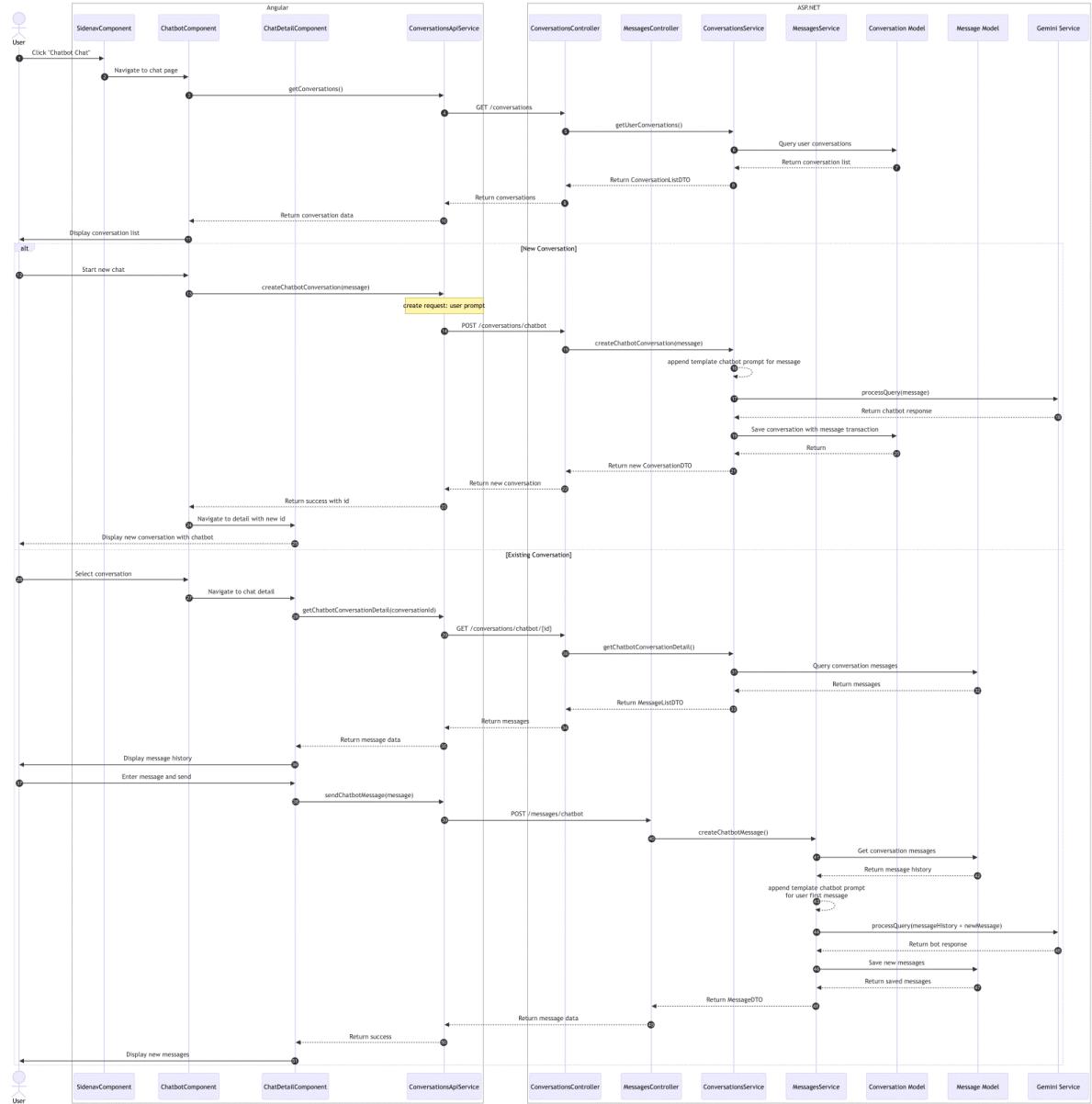
Hình 2.24 Sơ đồ tuần tự use case theo dõi phiên công khai

g. Use case 7: Quản lý thông báo cá nhân



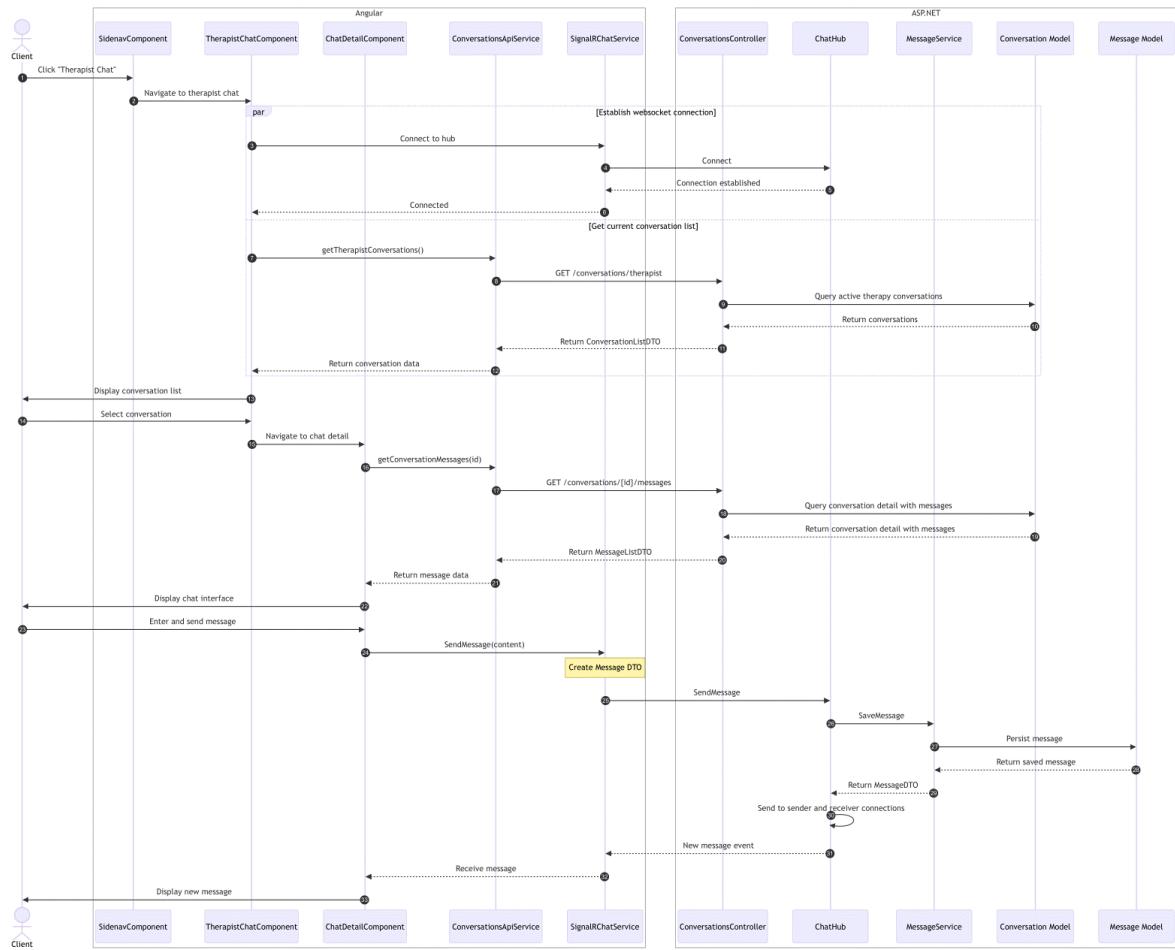
Hình 2.25 Sơ đồ tuần tự use case quản lý thông báo cá nhân

h. Use case 8: Trò chuyện với chatbot



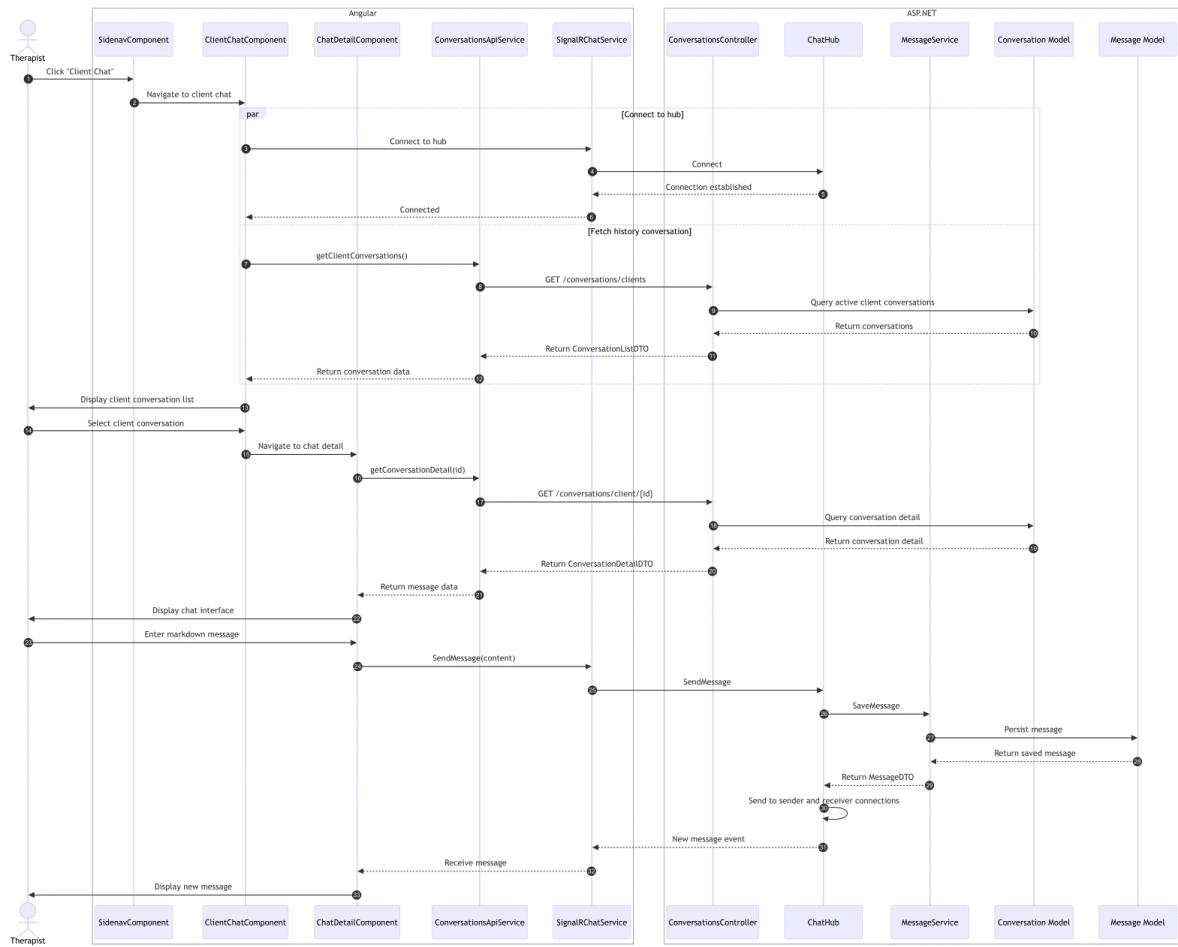
Hình 2.26 Sơ đồ tuần tự use case trò chuyện với chatbot

i. Use case 9: Trò chuyện với người trị liệu Với người sử dụng dịch vụ trị liệu



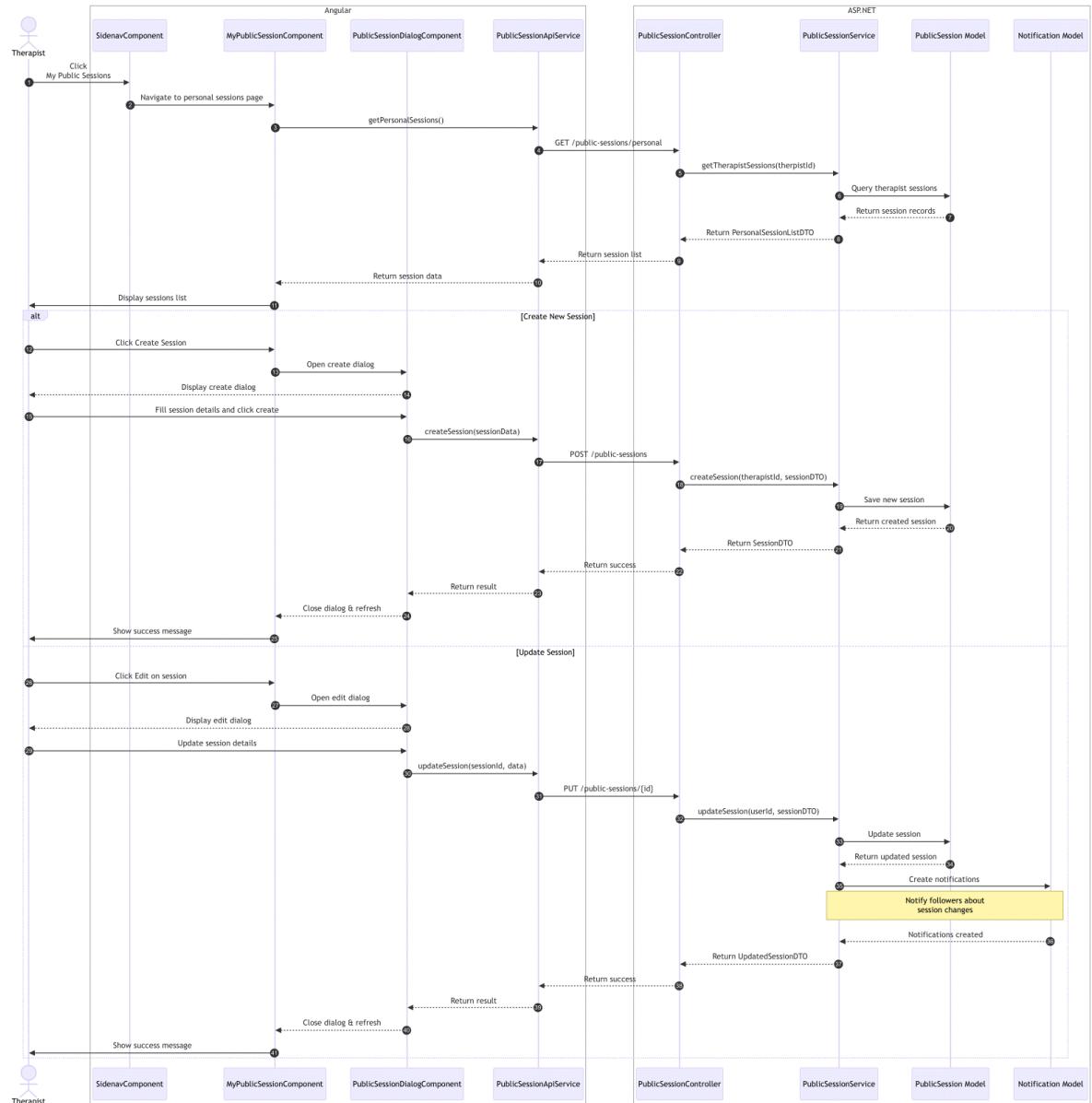
Hình 2.27 Sơ đồ tuần tự use case chat người với trị liệu

Với người trị liệu



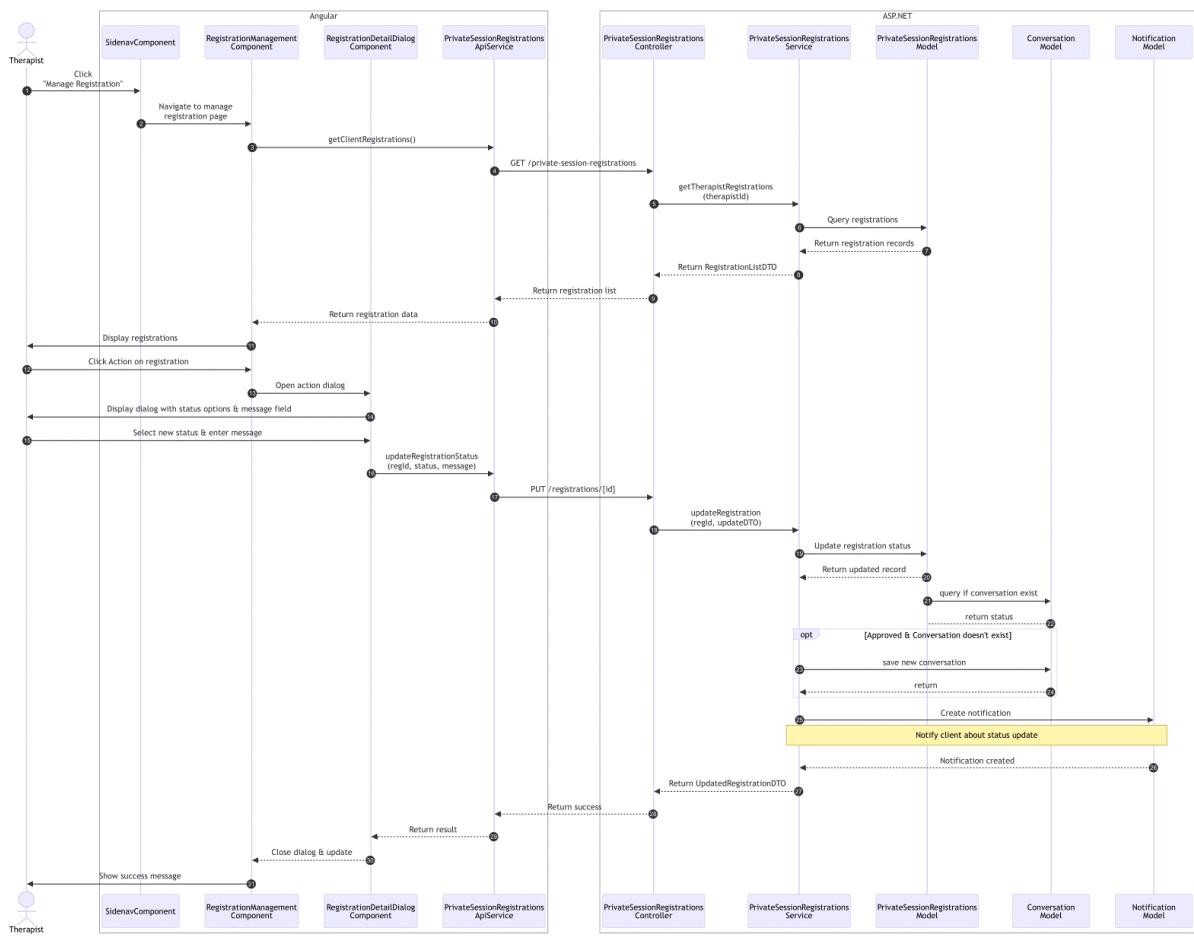
Hình 2.28 Sơ đồ tuần tự use case chat với người đăng ký trị liệu

j. Use case 10: Quản lý các phiên trị liệu công khai cá nhân



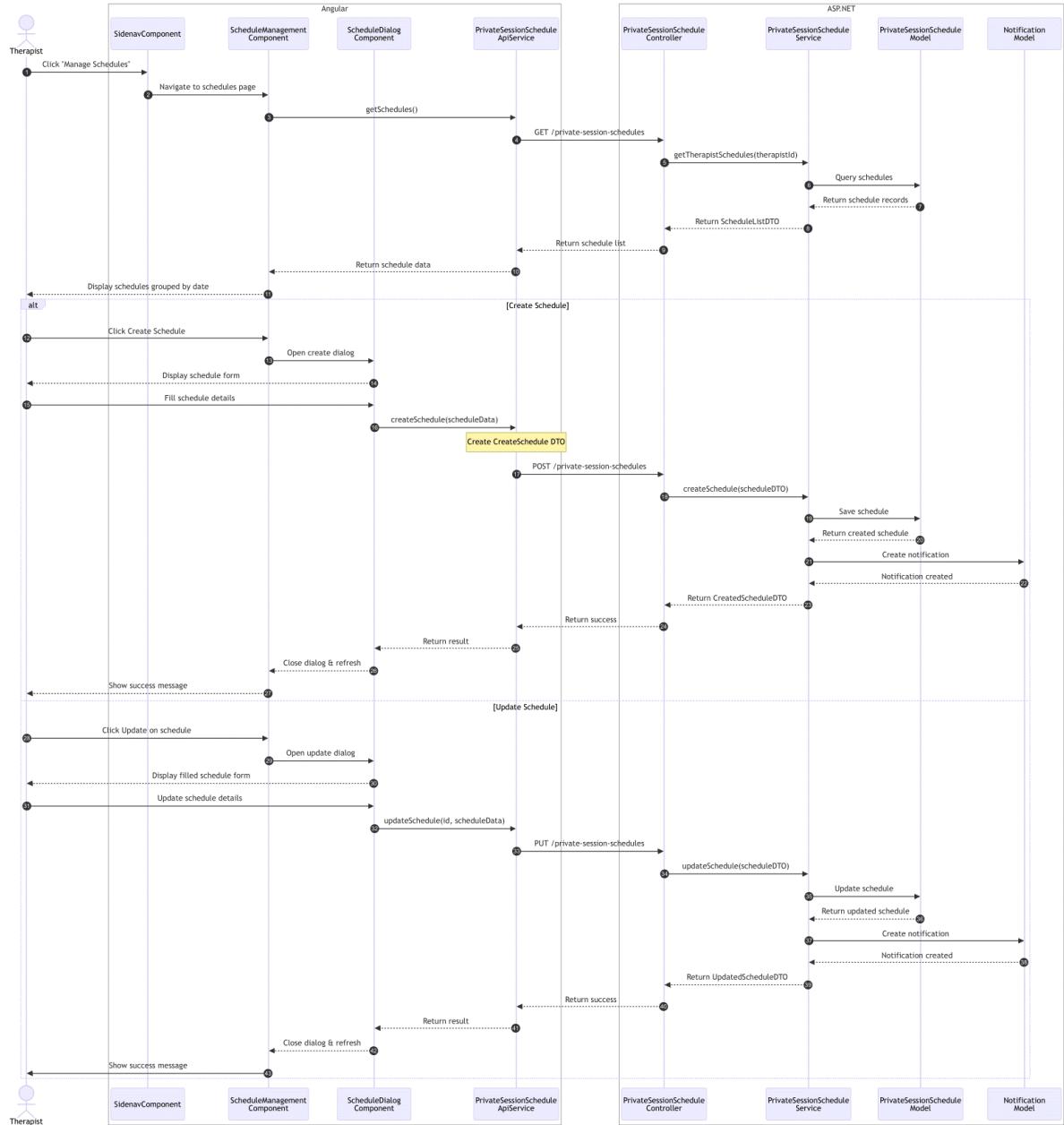
Hình 2.29 Sơ đồ tuần tự use case quản lý phiên công khai cá nhân

k. Use case 11: Quản lý đăng ký phiến trị liệu cá nhân



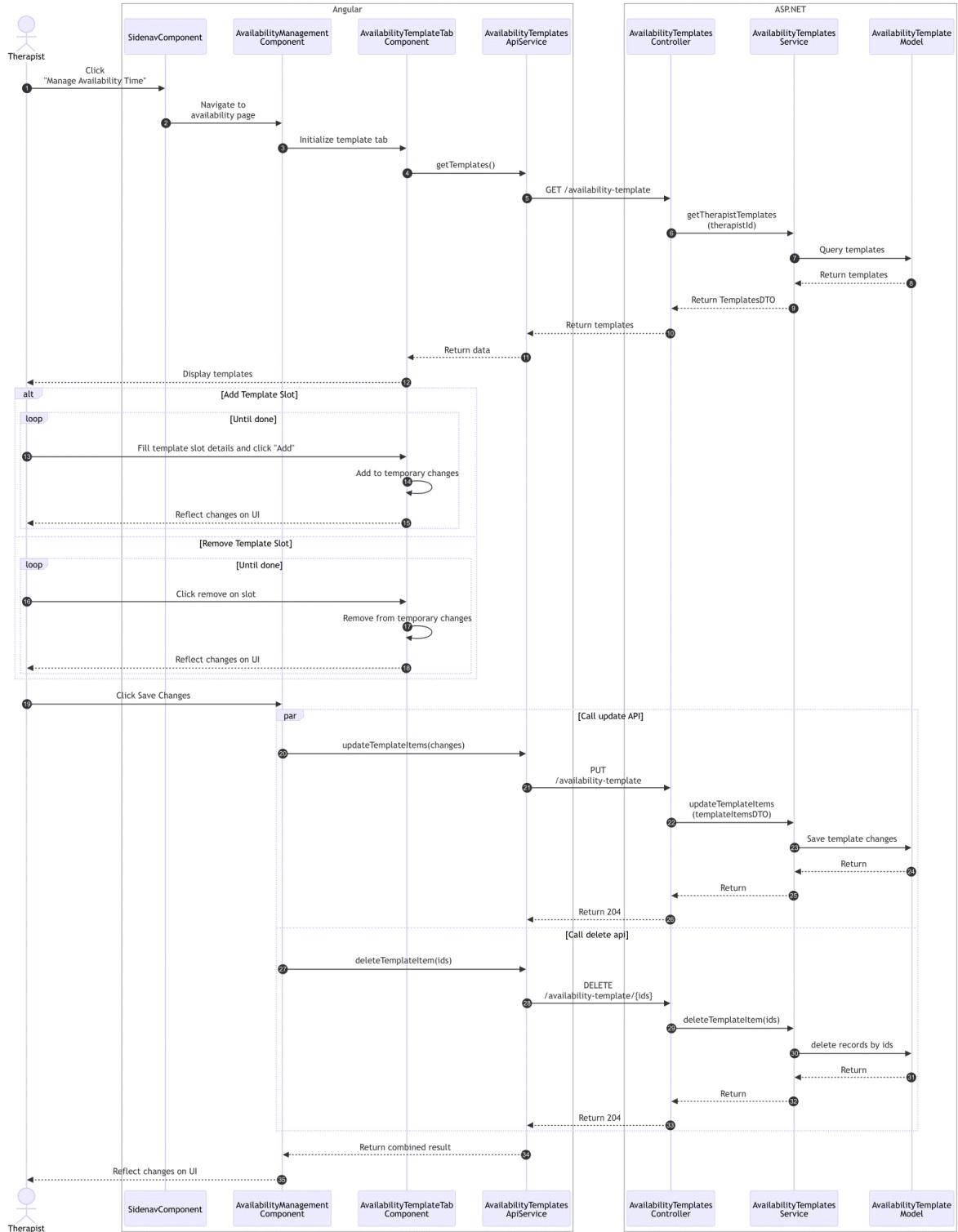
Hình 2.30 Sơ đồ tuần tự use case quản lý đăng ký phiến trị liệu cá nhân

1. Use case 12: Quản lý lịch các phiên trị liệu cá nhân

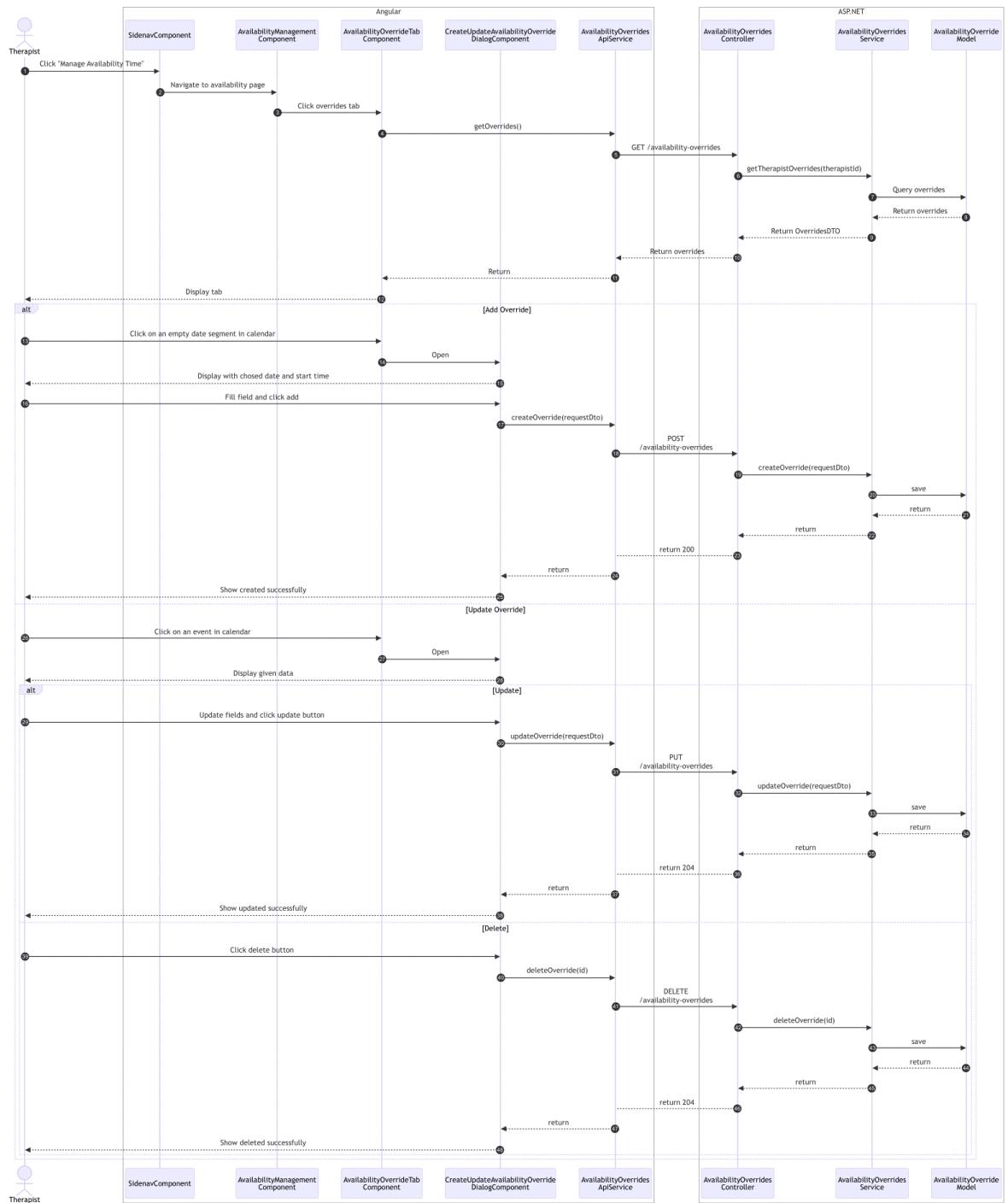


Hình 2.31 Sơ đồ tuần tự use case quản lý lịch các phiên trị liệu cá nhân

m. Use case 13: Quản lý lịch làm việc cá nhân

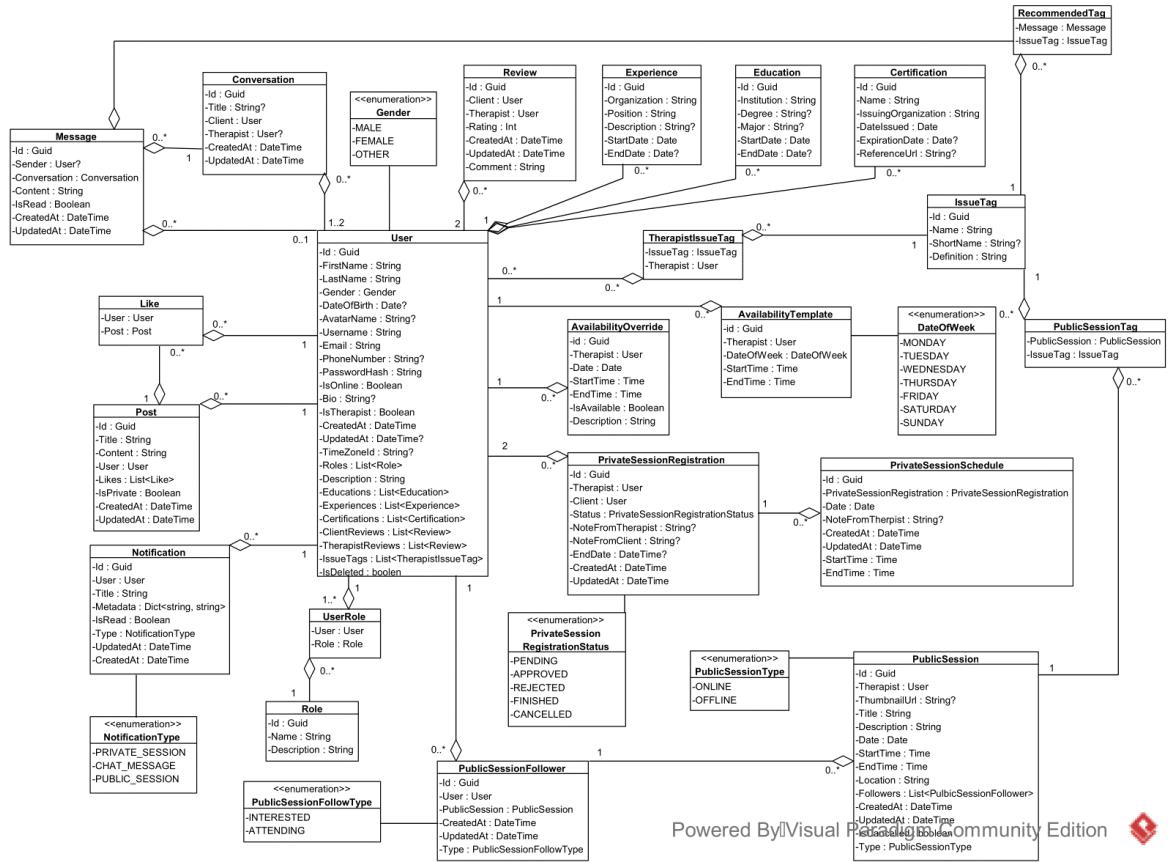


Hình 2.32 Sơ đồ tuần tự use case quản lý lịch làm việc cá nhân



Hình 2.33 Sơ đồ tuần tự use case quản lý lịch làm việc riêng

2.3.2. Biểu đồ lớp thiết kế



Powered By Visual Paradigm Community Edition

Hình 2.34 Biểu đồ lớp thực thể

a. User

Là lớp cơ bản, có thể xuất hiện ở hầu hết các lớp khác. Lớp User đại diện chung cho định danh một người trong hệ thống. User cũng được xem là một Therapist, người cung cấp dịch vụ trị liệu trong hệ thống, và cũng kế thừa toàn bộ chức năng và khả năng truy cập tài nguyên trên hệ thống.

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của người dùng
FirstName	String	Tên của người dùng
LastName	String	Họ của người dùng
DateOfBirth	Date	Ngày sinh của người dùng
AvatarName	String?	Đường dẫn của ảnh trên server
UserName	String	Tên đăng nhập của người dùng
Email	String	Địa chỉ email của người dùng
PhoneNumber	String?	Số điện thoại của người dùng
PasswordHash	String	Mật khẩu đã được mã hóa 1 chiều
IsOnline	Boolean	Trạng thái hoạt động của người dùng
IsTherapist	Boolean	Cờ đánh dấu người dùng là người trị liệu
CreatedAt	DateTime	Thời điểm tạo tài khoản
UpdatedAt	DateTime	Thời điểm cập nhật thông tin gần nhất
TimeZoneId	String?	Mã múi giờ của người dùng

Gender	Gender	Giới tính (MALE/FEMALE/OTHER)
Bio	String?	Tiểu sử/Giới thiệu về người dùng
Description	String?	Mô tả về chuyên môn của nhà trị liệu
IsDeleted	Boolean	Cờ đánh dấu tài khoản đã bị xoá
LastInvalidatedAt	DateTime?	Mốc thời gian đánh dấu khi người dùng xoá tài khoản, hoặc đổi mật khẩu, hoặc cập nhật vai trò trên hệ thống.
Educations	List<Education>	Danh sách các trình độ học vấn của nhà trị liệu
Experiences	List<Experience>	Danh sách kinh nghiệm làm việc của nhà trị liệu
Certifications	List<Certification>	Danh sách các chứng chỉ của nhà trị liệu
Reviews	List<Review>	Danh sách các đánh giá của khách hàng
IssueTag	List<IssueTag>	Danh sách các vấn đề nhà trị liệu chuyên xử lý

Role

Một User có thể có nhiều Role. Các vai trò cơ bản của hệ thống hiện tại là: User và Therapist. Hệ thống sẽ dựa vào vai trò của người dùng để giới hạn các chức năng và khả năng truy cập tài nguyên trên hệ thống

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của vai trò
Name	String	Tên của vai trò
Description	String	Mô tả chi tiết về vai trò

UserRole

Lớp đại diện cho bảng kết nối quan hệ nhiều-nhiều giữa User và Role trong tầng lưu trữ vật lý

Thuộc tính	Kiểu dữ liệu	Định nghĩa
User	User	Đối tượng người dùng
Role	Role	Đối tượng vai trò của người dùng tương ứng trên hệ thống

Education

Lớp Education lưu trữ thông tin về trình độ học vấn của nhà trị liệu.

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của trình độ học vấn
Institution	String	Tên trường hoặc cơ sở giáo dục
Degree	String	Bằng cấp đạt được
Major	String	Chuyên ngành học
StartDate	Date	Ngày bắt đầu học
EndDate	Date?	Ngày kết thúc học

Certification

Lớp Certification lưu trữ thông tin về các chứng chỉ mà nhà trị liệu đạt được.

Thuộc tính	Kiểu dữ liệu	Định nghĩa
------------	--------------	------------

Id	Guid	Định danh duy nhất của chứng chỉ
Name	String	Tên của chứng chỉ
IssuingOrganization	String	Tổ chức cấp chứng chỉ
DateIssued	Date	Ngày cấp chứng chỉ
ExpirationDate	Date	Ngày hết hạn chứng chỉ (nếu có)
ReferenceUrl	String?	URL tham chiếu đến chứng chỉ

Experience

Lớp Experience đại diện cho một kinh nghiệm làm việc cụ thể của nhà trị liệu.

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của kinh nghiệm
Organization	String	Tên tổ chức nơi nhà trị liệu làm việc
Position	String	Vị trí công việc của nhà trị liệu
Description	String?	Mô tả về kinh nghiệm công việc
StartDate	Date	Ngày bắt đầu công việc
EndDate	Date?	Ngày kết thúc công việc (nếu có)

Review

Lớp Review lưu trữ các đánh giá của khách hàng về nhà trị liệu, bao gồm xếp hạng và bình luận. Một khách hàng chỉ có thể đánh giá một lần cho một nhà trị liệu

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của đánh giá
Client	User	Khách hàng đã thực hiện đánh giá
Therapist	User	Nhà trị liệu được đánh giá
Rating	Int	Xếp hạng (điểm) của nhà trị liệu, chỉ chứa các giá trị nguyên từ 1-5
CreatedAt	DateTime	Thời điểm tạo đánh giá
UpdatedAt	DateTime	Thời điểm cập nhật đánh giá gần nhất
Comment	String	Bình luận của khách hàng

b. IssueTag

Lớp IssueTag đại diện cho các vấn đề mà một nhà trị liệu có chuyên môn xử lý. Các dữ liệu sẽ được định nghĩa sẵn ở phía cơ sở dữ liệu

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của vấn đề
Name	String	Tên của vấn đề
ShortName	String	Tên ngắn gọn của vấn đề
Definition	String	Định nghĩa chi tiết về vấn đề

TherapistIssueTag

Là lớp đại diện cho tầng vật lý của bảng đại diện cho mỗi quan hệ nhiều-nhiều giữa Therapist và IssueTag.

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Therapist	User	Đối tượng chuyên gia với loại điều trị tương ứng
IssueTag	IssueTag	Đối tượng loại điều trị với chuyên gia tương ứng

RecommendedTag

Là lớp đại diện cho tầng vật lý của bảng sinh ra từ mối quan hệ nhiều-nhiều giữa Message và IssueTag. Mỗi thực thể của lớp này chứa IssueTag được gọi ý từ câu trả lời của chatbot với người dùng (Message).

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Message	Message	Đối tượng tin nhắn từ chatbot
IssueTag	IssueTag	Đối tượng thẻ loại vấn đề tâm lí tương ứng

PublicSessionTag

Là lớp đại diện cho tầng vật lý của bảng sinh ra từ mối quan hệ nhiều-nhiều giữa PublicSession và IssueTag. Mỗi thực thể của lớp này chứa IssueTag tương ứng với một bài đăng công khai (PublicSession)

Thuộc tính	Kiểu dữ liệu	Định nghĩa
PublicSession	PublicSession	Đối tượng phiên công khai sở hữu tag
IssueTag	IssueTag	Đối tượng thẻ loại vấn đề tâm lí tương ứng

c. AvailabilityTemplate

Lớp AvailabilityTemplate đại diện cho khung thời gian mà một nhà trị liệu có sẵn trong tuần. Mỗi khung thời gian bao gồm ngày trong tuần, thời gian bắt đầu và kết thúc.

Thuộc tính	Kiểu dữ liệu	Định nghĩa
Id	Guid	Định danh duy nhất của khung thời gian sẵn có
Therapist	User	Nhà trị liệu sở hữu khung thời gian
DateOfWeek	DateOfWeek	Ngày trong tuần mà nhà trị liệu có thể làm việc
StartTime	Time	Thời gian bắt đầu sẵn sàng
EndTime	Time	Thời gian kết thúc sẵn sàng

DateOfWeek (Enumeration)

DateOfWeek là một kiểu liệt kê các ngày trong tuần, dùng để xác định ngày mà nhà trị liệu có sẵn.

Giá trị	Mô tả
MONDAY	Thứ Hai
TUESDAY	Thứ Ba
WEDNESDAY	Thứ Tư
THURSDAY	Thứ Năm
FRIDAY	Thứ Sáu
SATURDAY	Thứ Bảy
SUNDAY	Chủ Nhật

d. AvailabilityOverride

Lớp AvailabilityOverride đại diện cho sự điều chỉnh lịch làm việc của một chuyên gia trị liệu, cho phép cập nhật tình trạng sẵn có của họ vào một thời gian cụ thể, ghi đè lên lịch AvailabilityTemplate

Thuộc tính	Kiểu dữ liệu	Mô tả
Id	Guid	Mã định danh duy nhất của bản ghi điều chỉnh lịch trình.

Therapist	User	Thông tin của chuyên gia trị liệu liên quan đến bản ghi điều chỉnh lịch này.
Date	Date	Ngày cụ thể của bản ghi điều chỉnh lịch trình.
StartTime	Time	Thời gian bắt đầu của khoảng thời gian điều chỉnh.
EndTime	Time	Thời gian kết thúc của khoảng thời gian điều chỉnh.
IsAvailable	Boolean	Trạng thái sẵn có của chuyên gia trong khoảng thời gian này (có sẵn hoặc không).
Description	String?	Mô tả thêm về lý do hoặc chi tiết của điều chỉnh lịch trình ghi đè

e. PrivateSessionRegistration

Lớp PrivateSessionRegistration đại diện cho thông tin đăng ký một buổi trị liệu riêng tư giữa chuyên gia trị liệu và khách hàng.

Thuộc tính	Kiểu dữ liệu	Mô tả
Id	Guid	Mã định danh duy nhất của bản ghi đăng ký buổi trị liệu.
Therapist	User	Chuyên gia trị liệu được đăng ký cho buổi trị liệu này.
Client	User	Khách hàng đăng ký buổi trị liệu.
Status	PrivateSessionRegistrationStatus	Trạng thái hiện tại của buổi trị liệu (chờ xử lý, phê duyệt, từ chối, hoàn tất, hủy bỏ).
Note FromTherapist	String?	Ghi chú từ chuyên gia trị liệu cho buổi trị liệu này (tùy chọn).
Note FromClient	String?	Ghi chú từ khách hàng cho buổi trị liệu này (tùy chọn).
StartDate	DateTime	Thời điểm bắt đầu của buổi trị liệu.
EndDate	DateTime?	Thời điểm kết thúc của buổi trị liệu. Chỉ có người trị liệu có quyền cập nhật thay đổi cho thuộc tính này. Giá trị khác rỗng đồng nghĩa với việc phiên trị liệu đã kết thúc.
CreatedAt	DateTime	Thời điểm tạo bản ghi đăng ký buổi trị liệu.
UpdatedAt	DateTime	Thời điểm cập nhật gần nhất của bản ghi đăng ký buổi trị liệu.

PrivateSessionRegistrationStatus (Enumeration)

Thuộc tính	Mô tả
PENDING	Trạng thái chờ xử lý.
APPROVED	Đã được phê duyệt và sẵn sàng thực hiện.
REJECTED	Bị từ chối và không thể tiếp tục.
FINISHED	Buổi trị liệu đã hoàn tất.
CANCELLED	Buổi trị liệu đã bị hủy.

PrivateSessionSchedule

Lớp PrivateSessionSchedule đại diện cho lịch trình cụ thể của một buổi trị liệu riêng tư diễn ra giữa người trị liệu và khách hàng

Thuộc tính	Kiểu dữ liệu	Mô tả
------------	--------------	-------

Id	Guid	Mã định danh duy nhất của bản ghi lịch trình buổi trị liệu.
PrivateSession Registration	PrivateSession Registration	Tham chiếu đến lớp đăng ký buổi trị liệu liên quan.
Date	Date	Ngày cụ thể của buổi trị liệu.
NoteFromTherapist	String?	Ghi chú từ chuyên gia trị liệu cho lịch trình này (tùy chọn).
CreatedAt	DateTime	Thời điểm tạo bản ghi lịch trình.
UpdatedAt	DateTime	Thời điểm cập nhật gần nhất của bản ghi lịch trình.
StartTime	Time	Thời gian bắt đầu của buổi trị liệu trong ngày.
EndTime	Time	Thời gian kết thúc của buổi trị liệu trong ngày.

f. Conversation

Lớp a. Conversation đại diện cho một cuộc trò chuyện giữa một người dùng và một nhà trị liệu (hoặc chatbot).

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của đối tượng.
Title	String?	Tiêu đề của cuộc trò chuyện, chỉ sử dụng cho cuộc hội thoại với chatbot
Client	User	Người dùng tham gia vào cuộc trò chuyện.
Therapist	User?	Nhà trị liệu tham gia vào cuộc trò chuyện. Nếu null thì cuộc hội thoại này là cuộc trò chuyện giữa người dùng và chatbot.
CreatedAt	DateTime	Thời gian tạo cuộc trò chuyện.
UpdatedAt	DateTime	Thời gian cập nhật cuối cùng của cuộc trò chuyện.

Message

Lớp Message đại diện cho từng tin nhắn trong một cuộc trò chuyện Conversation.

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của đối tượng tin nhắn.
Sender	User?	Người gửi tin nhắn. Nếu null thì đây là tin nhắn của chatbot
Conversation	Conversation	Cuộc trò chuyện mà tin nhắn thuộc về.
Content	String	Nội dung của tin nhắn.
IsRead	Boolean	Trạng thái đọc tin nhắn (đã đọc hay chưa).
CreatedAt	DateTime	Thời gian tạo tin nhắn.
UpdatedAt	DateTime	Thời gian cập nhật cuối cùng của tin nhắn.

g. PublicSession

Lớp PublicSession mô tả các thông tin về phiên công khai, được tạo bởi người trị liệu trên hệ thống. Người dùng toàn hệ thống có thể tham gia PublicSession.

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của phiên công khai.

Therapist	User	Nhà trị liệu tổ chức phiên công khai.
ThumbnailName	String?	Đường dẫn đến ảnh thu nhỏ của phiên công khai.
Title	String	Tiêu đề của phiên công khai.
Description	String	Mô tả chi tiết về phiên công khai.
Date	Date	Ngày của phiên công khai.
StartTime	Time	Thời gian bắt đầu của phiên công khai.
EndTime	Time	Thời gian kết thúc của phiên công khai.
Location	String	Địa điểm của phiên công khai. Có thể là địa chỉ vật lý hoặc đường dẫn tới phòng họp trực tuyến
Followers	List <PublicSessionFollower>	Danh sách người theo dõi phiên công khai.
CreatedAt	DateTime	Thời gian tạo phiên công khai.
UpdatedAt	DateTime	Thời gian cập nhật phiên công khai cuối cùng.
IsCancelled	Boolean	Trạng thái hủy bỏ của phiên công khai (đã hủy hay chưa).
Type	PublicSessionType	Loại phiên công khai

PublicSessionFollower

Lớp PublicSessionFollower thể hiện mối quan hệ giữa người dùng và phiên công khai, đại diện cho thông tin về người theo dõi một phiên công khai cụ thể.

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của đối tượng người theo dõi.
User	User	Người dùng theo dõi phiên công khai.
PublicSession	PublicSession	Phiên công khai mà người dùng đang theo dõi.
CreatedAt	DateTime	Thời gian người dùng bắt đầu theo dõi phiên công khai.
UpdatedAt	DateTime	Thời gian cập nhật cuối cùng cho người theo dõi phiên công khai.
Type	PublicSessionFollowType	Loại theo dõi (quan tâm, tham dự).

PublicSessionType (Enumeration)

Lớp PublicSessionFollowType xác định các loại khác nhau mà một người dùng có thể theo dõi phiên công khai.

Loại	Mô tả
ONLINE	Phiên trực tuyến
OFFLINE	Phiên trực tiếp

PublicSessionFollowType (Enumeration)

Lớp này xác định các loại hình theo dõi phiên công khai của một người dùng với một phiên công khai

Loại	Mô tả
INTERESTED	Quan tâm sự kiện
ATTENDING	Sẽ tham gia sự kiện

h. Post

Lớp Post đại diện cho một bài đăng công khai trên toàn hệ thống hoặc nhật ký riêng tư của cá nhân.

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của bài đăng.
Title	String	Tiêu đề của bài đăng.
Content	String	Nội dung của bài đăng.
User	User	Người dùng đăng bài.
Likes	List<Like>	Danh sách người thích bài đăng.
IsPrivate	Boolean	Trạng thái riêng tư của bài đăng (riêng tư hay công khai).
CreatedAt	DateTime	Thời gian tạo bài đăng.
UpdatedAt	DateTime	Thời gian cập nhật bài đăng cuối cùng.

Like

Lớp Like mô tả mối quan hệ thích giữa người dùng và bài đăng, đại diện cho lượt tương tác của người dùng với một bài đăng cụ thể

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của lượt thích.
User	User	Người dùng đã thích bài đăng.
Post	Post	Bài đăng được thích.

i. Notification

Lớp Notification đại diện cho thông báo cho một người dùng cụ thể trên hệ thống. Các thay đổi về tin nhắn riêng, phiên công khai hay lịch hẹn phiên công khai sẽ đều được thông báo và lưu trữ cho người dùng

Thuộc tính	Loại	Mô tả
Id	Guid	Mã định danh của thông báo.
User	User	Người nhận thông báo.
Title	String	Tiêu đề của thông báo.
Metadata	Dict<String, String>	Trường thông tin thêm cho thông báo (ví dụ: mã định danh liên quan tới loại thông báo)
IsRead	Boolean	Trạng thái đã đọc của thông báo (đã đọc hay chưa).
Type	NotificationType	Loại của thông báo.
UpdatedAt	DateTime	Thời gian cập nhật cuối cùng của thông báo.
CreatedAt	DateTime	Thời gian tạo thông báo.

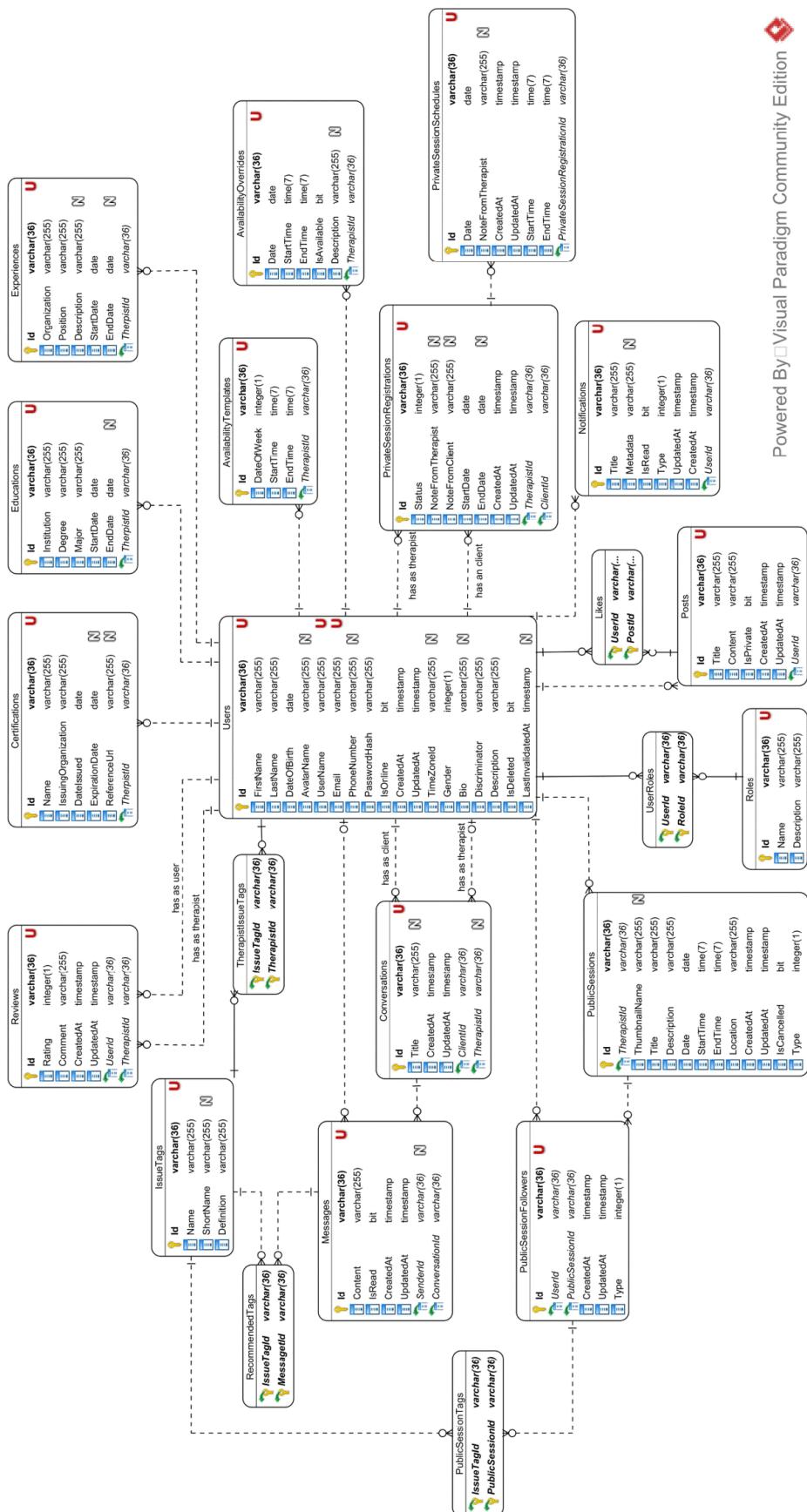
NotificationType (Enumeration)

Liệt kê các loại thông báo khác nhau có thể được gửi đến người dùng

Loại	Mô tả
PRIVATE_SESSION	Kiểu cho phiên trực tuyến

CHAT_MESSAGE	Kiểu cho cuộc hội thoại
PUBLIC_SESSION	Kiểu cho phiên công khai

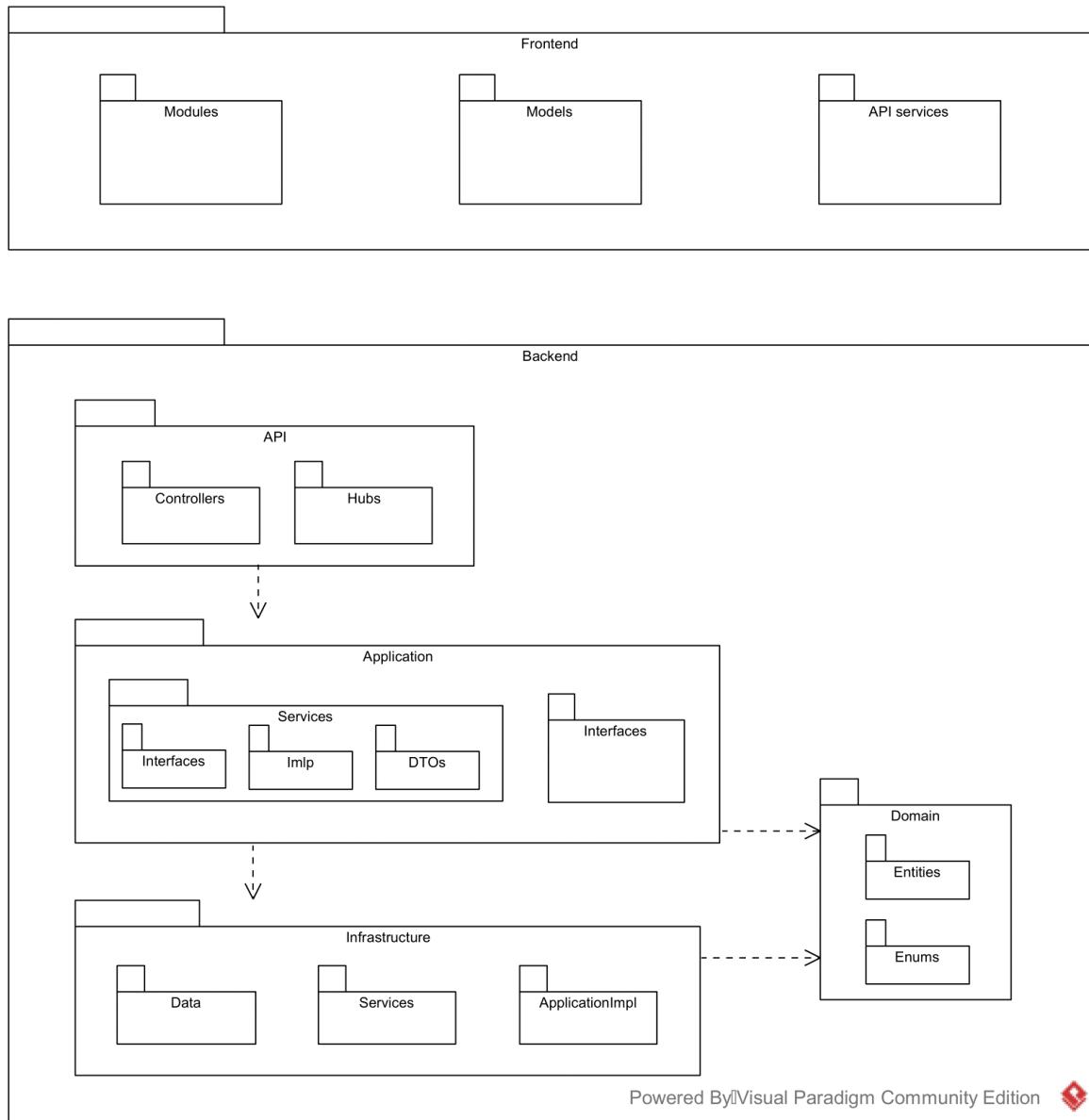
2.3.3. Cơ sở dữ liệu



Powered By Visual Paradigm Community Edition

Hình 2.35 Sơ đồ thiết kế cơ sở dữ liệu cho dự án

2.3.4. Biểu đồ gói



Bảng 2.15 Biểu đồ gói của mô hình hệ thống

Frontend là phần giao diện người dùng, nơi trực tiếp tương tác với người sử dụng. Các thành phần chính gồm:

- Modules: Chứa các module của ứng dụng, đại diện cho các tính năng hoặc trang cụ thể. Ví dụ: module đăng nhập, module chat.
- Models: Chứa các lớp dữ liệu hoặc cấu trúc được sử dụng trong frontend để biểu diễn dữ liệu, tương ứng với các DTO (Data Transfer Objects) ở backend.
- API services: Chứa các dịch vụ để gọi API từ backend, đóng vai trò trung gian giữa frontend và backend. Đây là nơi thực hiện các yêu cầu HTTP như GET, POST, PUT, DELETE.

Backend là nơi xử lý logic nghiệp vụ, lưu trữ dữ liệu và cung cấp API cho frontend. Nó bao gồm các thành phần sau:

API (Web framework)

- Controllers: Chứa các bộ điều khiển để xử lý các yêu cầu từ frontend thông qua các endpoint API. Controllers điều phối luồng dữ liệu giữa các lớp dịch vụ và trả về phản hồi cho client.
- Hubs: Được sử dụng trong các ứng dụng thời gian thực (real-time), chẳng hạn như chat, dựa trên SignalR để xử lý kết nối hai chiều giữa client và server.

Application

- Services:
 - Interfaces: Định nghĩa các giao diện dịch vụ, đảm bảo tính linh hoạt và khả năng mở rộng khi triển khai.
 - Impl: Chứa các lớp triển khai logic nghiệp vụ dựa trên các giao diện được định nghĩa trong "Interfaces".
 - DTOs: Chứa các lớp được sử dụng để trao đổi dữ liệu giữa các tầng hoặc với client.
- Interfaces (Application): Cung cấp các giao diện cần thiết để ứng dụng tương tác với tầng hạ tầng.

Infrastructure

- Data: Chứa các cài đặt liên quan đến thao tác dữ liệu từ cơ sở dữ liệu.
- Services: Chứa các dịch vụ hỗ trợ, như gửi email, JWT, xử lý tệp hoặc tích hợp dịch vụ bên ngoài.
- ApplicationImpl: Chứa các lớp triển khai cụ thể để tích hợp các logic nghiệp vụ với các dịch vụ hạ tầng.

Domain

- Entities: Chứa các lớp biểu diễn thực thể của ứng dụng, tương ứng với các bảng trong cơ sở dữ liệu. Đây là trung tâm của các hoạt động nghiệp vụ.
- Enums: Chứa các kiểu liệt kê được sử dụng để định nghĩa các giá trị hằng số, như trạng thái hoặc loại dữ liệu.

2.4. Kết luận chương 2

Trong chương 2, quá trình phân tích và thiết kế hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực đã được thực hiện một cách chi tiết và chặt chẽ. Đầu tiên, mô hình tổng quát hệ thống đã được xây dựng để mô tả các thành phần chính bao gồm trình duyệt người dùng, giao diện Frontend bằng Angular, Backend ASP.NET Core Web API, và luồng hoạt động của hệ thống. Tiếp theo, việc phân tích yêu cầu hệ thống được tiến hành bằng ngôn ngữ tự nhiên, qua đó xác định được các chức năng cần thiết để đáp ứng nhu cầu của người dùng và chuyên gia. Sơ đồ use case tổng quan và các đặc tả chi tiết từng use case cũng đã được triển khai, giúp xác định rõ ràng các tương tác giữa người dùng với hệ thống. Thiết kế hệ thống được phát triển cùng với mô hình cơ sở dữ liệu chi tiết, bao gồm sơ đồ lớp và sơ đồ thực thể và sơ đồ tuần tự, nhằm hỗ trợ lưu trữ và truy xuất dữ liệu hiệu quả. Về triển khai, hệ thống tận dụng Microsoft Azure cho backend với các dịch vụ như App Service, PostgreSQL, Redis và Application Insights để đảm bảo hiệu suất và khả năng giám sát. Frontend được triển khai trên Firebase Hosting với Angular, cung cấp tốc độ tải nhanh và bảo mật HTTPS. Những nền tảng này là cơ sở vững chắc cho các bước phát triển tiếp theo và đảm bảo hệ thống đáp ứng các yêu cầu đã đề ra.

CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG HỖ TRỢ CHĂM SÓC SỨC KHỎE TINH THẦN THỜI GIAN THỰC TÍCH HỢP GENERATIVE-BASED CHATBOT VÀ KẾT NỐI DỊCH VỤ TRÍ LIỆU

Chương 3 tập trung vào việc thử nghiệm và đánh giá hiệu quả của hệ thống đã phát triển. Phần đầu trình bày dữ liệu thực nghiệm và quá trình cài đặt chatbot Generative-based nhằm kiểm tra khả năng xử lý hội thoại và tư vấn tâm lý. Sau đó, kết quả thực nghiệm của chatbot được phân tích để đánh giá độ chính xác và hiệu quả. Tiếp theo, chương trình bày quá trình thử nghiệm toàn diện hệ thống, bao gồm giao diện người dùng, các tính năng tích hợp AI Generative-based và dịch vụ trí liệu. Kết quả thử nghiệm sẽ làm cơ sở để đánh giá khả năng đáp ứng mục tiêu của hệ thống.

3.1. Dữ liệu thực nghiệm generative-based chatbot

Bộ dữ liệu được sử dụng để tinh chỉnh (fine-tuning) mô hình Gemini “gemini-1.0-pro-001” được chọn từ nền tảng Kaggle [10], với nội dung tập trung vào các câu hỏi thường gặp (FAQs) về sức khỏe tâm lý. Bộ dữ liệu này đóng vai trò quan trọng trong việc nâng cao khả năng phản hồi của chatbot, giúp cung cấp thông tin chính xác và phù hợp khi người dùng yêu cầu hỗ trợ về các vấn đề liên quan đến tâm lý.

Bộ dữ liệu này cung cấp các câu hỏi và câu trả lời liên quan đến nhiều khía cạnh khác nhau của sức khỏe tâm lý, bao gồm:

- Khái niệm và tầm quan trọng của sức khỏe tâm lý.
- Các cách xử lý cẩn thảng và thích nghi với thay đổi trong cuộc sống.
- Mối liên hệ giữa sức khỏe tâm lý và hiệu suất công việc, học tập, cũng như chất lượng mối quan hệ.
- Cách tìm kiếm sự trợ giúp và hy vọng cho những người gặp vấn đề tâm lý.

Bộ dữ liệu được tổng hợp từ các nguồn uy tín sau:

- The Kim Foundation: Một tổ chức phi lợi nhuận tập trung vào việc nâng cao nhận thức về sức khỏe tâm lý và cung cấp tài nguyên hỗ trợ.
- Mental Health America (MHA): Một tổ chức hàng đầu tại Mỹ trong việc vận động chính sách và hỗ trợ về sức khỏe tâm lý.
- Wellness in Mind: Một dịch vụ hỗ trợ sức khỏe tâm lý chuyên cung cấp các tài nguyên và hướng dẫn chi tiết.
- Here to Help BC: Một tổ chức tại Canada chuyên hỗ trợ người dân về các vấn đề liên quan đến sức khỏe tâm lý.

Bộ dữ liệu này được lựa chọn để đảm bảo chatbot có thể:

- Hướng phong cách trả lời của chatbot về lĩnh vực sức khỏe tâm lý
- Trả lời các câu hỏi thường gặp về sức khỏe tâm lý một cách tự nhiên và chính xác.
- Hỗ trợ người dùng nhận thức rõ hơn về tầm quan trọng của sức khỏe tâm lý.
- Gợi ý các biện pháp xử lý phù hợp cho từng tình huống.

Việc sử dụng bộ dữ liệu từ Kaggle giúp đảm bảo độ tin cậy và chất lượng của thông tin, đồng thời tối ưu hóa khả năng của mô hình Generative-based Chatbot trong việc đáp ứng nhu cầu của người dùng.

3.2. Cài đặt thực nghiệm generative-based chatbot

Để tối ưu hóa khả năng của mô hình Gemini Models/gemini-1.0-pro-001 trong việc xử lý các yêu cầu từ người dùng, quá trình tinh chỉnh (fine-tuning) được thực hiện thông qua nền tảng Google AI Studio. Các bước thực hiện được mô tả như sau:

1. Chuẩn bị dữ liệu

Bộ dữ liệu được định dạng thành file CSV, với cấu trúc gồm 2 cột duy nhất:

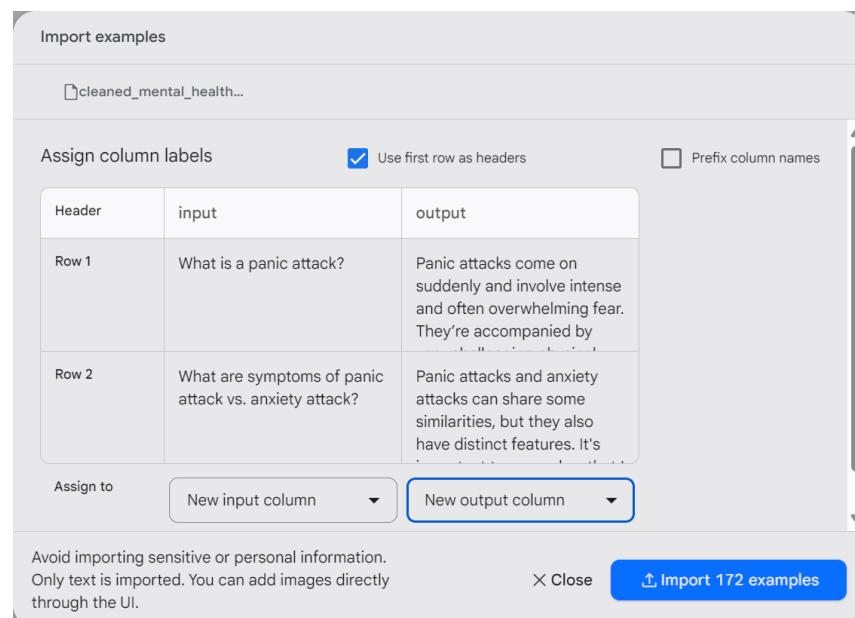
- Input: Chứa câu hỏi hoặc vấn đề mà người dùng có thể đưa ra (FAQ về sức khỏe tâm lý).
- Output: Chứa câu trả lời được thiết kế để phù hợp với ngữ cảnh và nội dung của câu hỏi.

Ví dụ về mẫu dữ liệu:

Input	Output
What is a panic attack?	Panic attacks come on suddenly and involve intense and often overwhelming fear. They're accompanied by very challenging physical symptoms, like a racing heartbeat, shortness of breath, or nausea...
How can I deal with stress?	Managing stress involves identifying triggers, relaxation techniques, and seeking support...

2. Nạp dữ liệu lên Google AI Studio

Dữ liệu được tải lên Google AI Studio trong mục tinh chỉnh (fine-tuning). Quá trình kiểm tra dữ liệu nhằm đảm bảo không có lỗi định dạng hoặc dữ liệu không phù hợp, giúp quá trình huấn luyện diễn ra hiệu quả.



Hình 3.1 Giao diện nạp dữ liệu trên Google AI Studio

3. Thiết lập tham số cho quá trình tinh chỉnh

Các tham số quan trọng trong quá trình tinh chỉnh được thiết lập như sau:

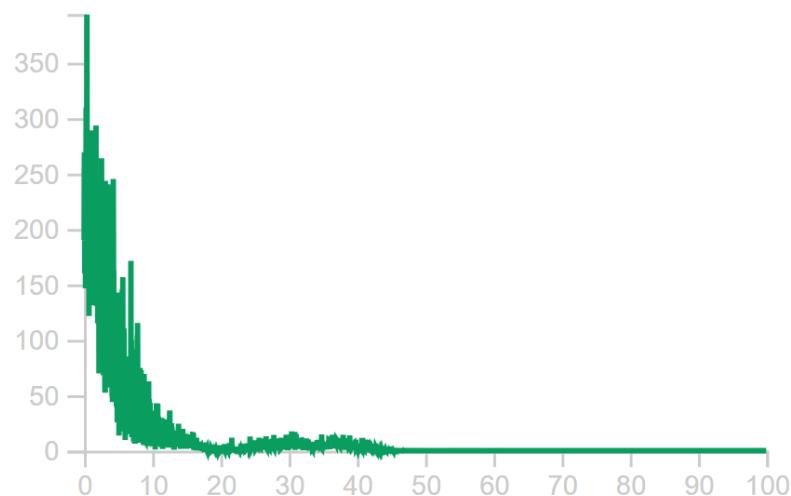
- Số lượng Epochs: 100, tránh tình trạng loss quá lớn và mô hình học chưa đủ, có thể chỉnh lại trong tương lai để tránh over-fitting.
- Learning Rate Multiplier: 0.001, đảm bảo mức độ học tập vừa đủ, tránh thay đổi quá lớn so với mô hình gốc.
- Batch Size: 4, là kích thước lô mặc định được thiết lập

Sau khi hoàn tất cấu hình, quá trình huấn luyện được kích hoạt trên Google AI Studio. Nền tảng này hỗ trợ theo dõi các thông số trong thời gian thực, bao gồm:

- Độ lỗi (loss) qua từng epoch.
- Hiệu quả học tập dựa trên tập dữ liệu kiểm tra (validation dataset).

3.3. Kết quả thực nghiệm generative-based chatbot

Thực nghiệm tinh chỉnh (fine-tuning) được thực hiện trên mô hình nền Gemini-1.0-pro-001, với một tập dữ liệu khiêm tốn gồm 172 mẫu. Quá trình huấn luyện kéo dài 100 epochs với kích thước batch là 4 và tốc độ học 0.001, hoàn thành trong khoảng một giờ, cho thấy việc sử dụng tài nguyên hiệu quả cho quy mô của thí nghiệm.



Hình 3.2 Biểu đồ loss/epochs xuyên suốt quá trình huấn luyện

Đường cong mất mát (loss curve) thể hiện mô hình suy giảm theo hàm mũ đặc trưng, cho thấy ba giai đoạn học tập riêng biệt của mô hình. Trong giai đoạn đầu, kéo dài khoảng 10 epochs đầu tiên, giá trị loss giảm mạnh từ khoảng 350 xuống dưới 50. Sự sụt giảm nhanh chóng này cho thấy mô hình đã nhanh chóng nắm bắt được các mẫu cơ bản trong dữ liệu huấn luyện, điều này gợi ý rằng tốc độ học 0.001 đã tạo ra sự cân bằng hiệu quả giữa tốc độ hội tụ và độ ổn định trong quá trình huấn luyện.

Sau giai đoạn giảm mạnh ban đầu, quá trình huấn luyện bước vào giai đoạn cải thiện vừa phải giữa epochs 10 và 30. Trong giai đoạn này, tốc độ cải thiện chậm lại đáng kể, mặc dù loss vẫn tiếp tục giảm với tốc độ nhẹ nhàng hơn. Giai đoạn này thể hiện sự chuyển đổi của mô hình từ việc nhận diện mẫu tổng quát sang việc tinh chỉnh chi tiết hơn trong việc hiểu dữ liệu huấn luyện.

Giai đoạn cuối cùng, kéo dài từ epoch 30 đến khi kết thúc ở epoch 100, cho thấy sự hội tụ rõ ràng khi loss ổn định ở một giá trị thấp ổn định. Mặc dù vẫn còn những dao động nhỏ trong suốt giai đoạn ổn định này, sự ổn định tổng thể của các giá

trị loss cho thấy mô hình đã thành công trong việc đạt được trạng thái tối ưu cho tập dữ liệu đã cho. Mô hình hội tụ này cho thấy số lượng epochs được chọn là đủ để mô hình trích xuất các mẫu liên quan từ dữ liệu huấn luyện mà không có dấu hiệu của overfitting.

Cấu hình huấn luyện đã chứng minh được sự phù hợp với nhiệm vụ, với kích thước batch nhỏ tương đối là 4 cho phép cập nhật tham số chính xác phù hợp với kích thước tập dữ liệu khiêm tốn. Tốc độ học 0.001 tạo điều kiện cho cả việc học ban đầu nhanh chóng và hội tụ ổn định, được thể hiện qua đường cong suy giảm mượt mà không có dao động đáng kể. Tổng thời gian huấn luyện một giờ thể hiện việc sử dụng tài nguyên tính toán hiệu quả, đặc biệt khi xem xét số lượng epochs và kích thước của tập dữ liệu huấn luyện.

3.4. Thử nghiệm hệ thống hỗ trợ chăm sóc sức khỏe tinh thần thời gian thực tích hợp Generative-based Chatbot và kết nối dịch vụ trị liệu

3.4.1. Giao diện ứng dụng

- a. Use case 1: Đăng nhập tài khoản

User name	Password	Type
john.doe, johnny.dang	Password@123	User
michael.chen, emily.rodriguez, david.williams, amanda.patel	Password@123	Therapist

Hình 3.3 Giao diện đăng nhập ứng dụng



Create an account

1 Account identity —— 2 Personal information —— 3 Therapist information
Optional —— 4 Done

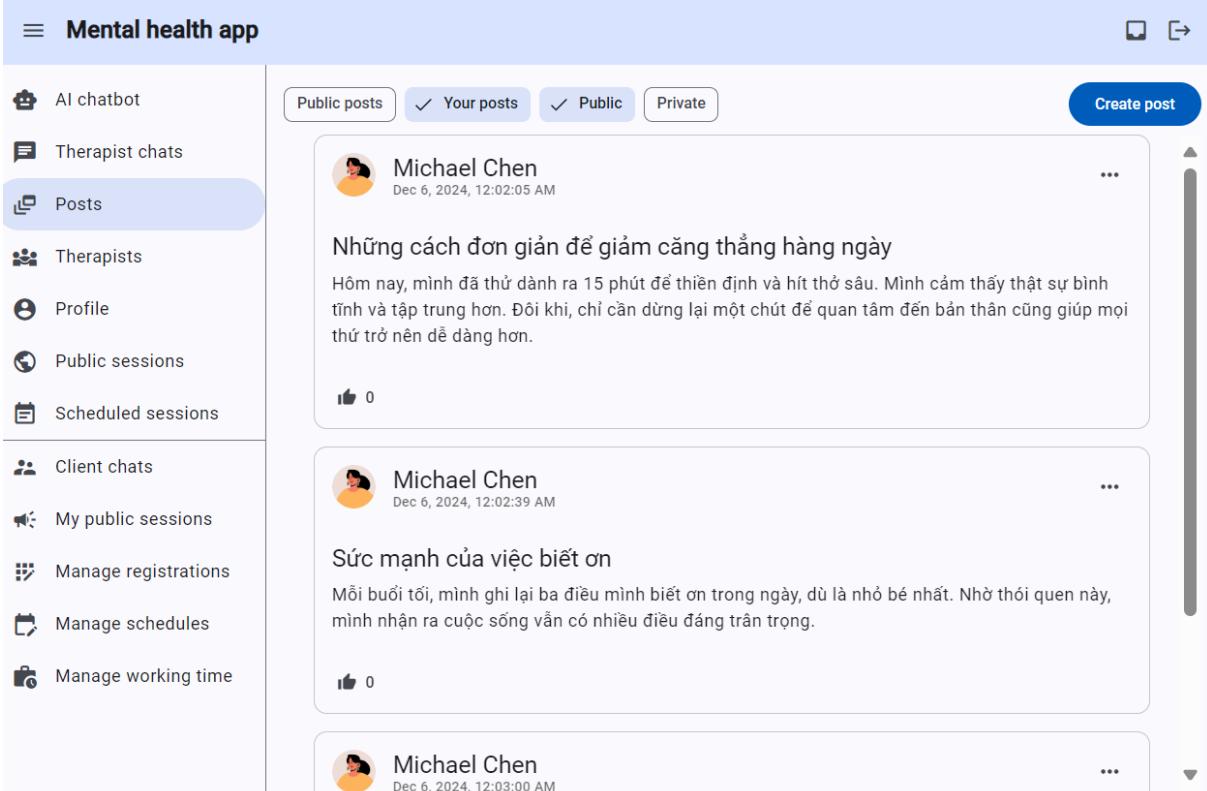
First name*	Nguyễn	Last name*	Phúc Sơn
Date of birth*	10/14/2002	MM/DD/YYYY	<input type="button" value="Select Date"/>
Phone number	0776231838		
Gender*	Male		
Biography		Cung bão bình	


[Select Avatar](#)

[Back](#) [Next](#)

Hình 3.4 Giao diện đăng ký tài khoản

b. Use case 2: Xem bài đăng



The screenshot shows the 'Mental health app' interface. On the left is a sidebar with navigation options: AI chatbot, Therapist chats, Posts (selected), Therapists, Profile, Public sessions, Scheduled sessions, Client chats, My public sessions, Manage registrations, Manage schedules, and Manage working time. The main area displays three public posts from user Michael Chen:

- Post 1:** Michael Chen, Dec 6, 2024, 12:02:05 AM. Content: "Những cách đơn giản để giảm căng thẳng hàng ngày". Description: "Hôm nay, mình đã thử dành ra 15 phút để thiền định và hít thở sâu. Minh cảm thấy thật sự bình tĩnh và tập trung hơn. Đôi khi, chỉ cần dừng lại một chút để quan tâm đến bản thân cũng giúp mọi thứ trở nên dễ dàng hơn." Likes: 0.
- Post 2:** Michael Chen, Dec 6, 2024, 12:02:39 AM. Content: "Sức mạnh của việc biết ơn". Description: "Mỗi buổi tối, mình ghi lại ba điều mình biết ơn trong ngày, dù là nhỏ bé nhất. Nhờ thói quen này, mình nhận ra cuộc sống vẫn có nhiều điều đáng trân trọng." Likes: 0.
- Post 3:** Michael Chen, Dec 6, 2024, 12:03:00 AM. (Partially visible)

Hình 3.5 Giao diện bài đăng

c. Use case 3: Quản lý thông tin cá nhân

Avatar



Select Avatar

Biography

Informative information about you

Personal information

First name* John

Last name* Doe

Email* john.doe@example.com

Phone number 1234567890

Date of birth*

Hình 3.6 Giao diện chỉnh sửa thông tin cá nhân

Re-enter old password and enter your new password

Old password*

New password*

Confirm new password*

Submit

Confirm delete account

Account deletion is a permanent action. To proceed, confirm your identity by entering your current password. This will immediately and irreversibly remove all your personal data from our system.

Current password*

Submit

Hình 3.8 Giao diện xoá tài khoản

Hình 3.7 Giao diện thay đổi mật khẩu

#	Therapist Name	Registered At	Last Updated At	Status	End Date	Review
1	Michael Chen	Jan 1, 1	Dec 22, 2024	Finished	Dec 22, 2024	Action
2	Sarah Johnson	Jan 1, 1		Rejected		Action
3	David Williams			Rejected	Dec 22, 2024	Action
4	Sarah Johnson			Rejected		Action
5	David Williams			Rejected		Action
6	Sarah Johnson			Rejected		Action
7	Emily Rodriguez	Jan 1, 1		Rejected		Action
8	David Williams	Jan 1, 1	Dec 22, 2024	Finished	Dec 22, 2024	Action
9	Amanda Patel	Jan 1, 1	Dec 22, 2024	Finished	Dec 22, 2024	Action

Rate therapist

Michael Chen

★★★★★

Comment* Michael is compassionate and an excellent listener.

Close **Submit**

Hình 3.9 Giao diện lịch sử đăng ký và đánh giá người trị liệu

d. Use case 4: Đăng kí phiên trị liệu cá nhân

The screenshot shows a user interface for therapist discovery. At the top left is a search bar labeled "Search by name". On the right are "Search" and "Reset" buttons. Below the search bar are four therapist profiles arranged in a grid:

- David Williams**: Male, 2 reviews. Education: Master of Science in Clinical Mental Health at University of Oxford. Experience: Clinical Psychologist at Wellbeing Mental Health Clinic. Certifications: 1 certification. Clients helped: 4. Bio: Passionate about helping individuals navigate challenges using evidence-based therapies. Specialties: Addiction, Anxiety, Depression, Grief, PTSD, Relationship Issues, Self-Esteem, Stress. Buttons: MORE...
- Sarah Johnson**: Male, 3.9 reviews. Education: Master of Psychology in Clinical Psychology at Harvard University. Experience: Senior Therapist at Mindful Wellness Center. Certifications: 1 certification. Clients helped: 10. Bio: Dedicated to empowering individuals through holistic mental health practices. Specialties: Anxiety, Depression, PTSD. Buttons: MORE...
- Michael Chen**: Female, 5 reviews. Education: PhD in Counseling Psychology in Counseling Psychology at Stanford. Experience: Lead Counselor at Harmony Therapy Center. Certifications: 1 certification. Clients helped: 3. Bio: Dedicated professional with over 10 years of experience. Buttons: MORE...
- Emily Rodriguez**: Male, 3 reviews. Education: Master of Social Work in Mental Health at University of California, B... Experience: Mental Health Specialist at Serenity Counseling Center. Certifications: 1 certification. Clients helped: 0. Bio: Experienced therapist with a focus on mindfulness and cognitive-behavioral therapy. Specialties: Anxiety, Depression, PTSD. Buttons: MORE...

On the right side, there is a sidebar titled "Advanced filters" with a checked "Enable" button. It includes sections for "Issue tags" (Anxiety, Post-traumatic stress disorder), "Rating" (a slider from 1 to 5 with a yellow star icon), "Gender" (Male, Female, Other checkboxes), "Experience level" (Disable, 0-2 years radio buttons), and a scrollable list of other filters.

Hình 3.10 Giao diện tìm kiếm người trị liệu

Emily Rodriguez



Gender Male
Date of birth Aug 22, 1979
Email emily.rodriguez@example.com
Joined since Dec 22, 2024

Experienced therapist with a focus on mindfulness and cognitive-behavioral therapy.

[Register](#)

Tags

Anxiety Depression PTSD
Relationship Issues

Experiences

Serenity Counseling Center
Mental Health Specialist
09/01/2005 - 06/30/2015

Educations

University of California, Berkeley
Mental Health, Master of Social Work
01/01/0001 - present

Reviews (1)

AI review summary

- Overall Sentiment:** The overall sentiment is positive, with users praising the therapist's mindfulness techniques in helping them manage stress levels.
- Key Strengths:** A key strength mentioned is the therapist's mindfulness techniques, which have been found helpful in managing stress.
- Notable Concerns:** No specific concerns or issues were mentioned in the given review.

Recent reviews

John Doe • 3 ★

Emily's mindfulness techniques really helped me manage my stress levels.

At 12/22/24, 12:01 PM

Therapist available time

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
5 AM							
6 AM							
7 AM							

Hình 3.11 Giao diện thông tin chi tiết người trị liệu và đăng ký

Reviews (1)

Register Emily Rodriguez for private therapy session?

Message to therapist*

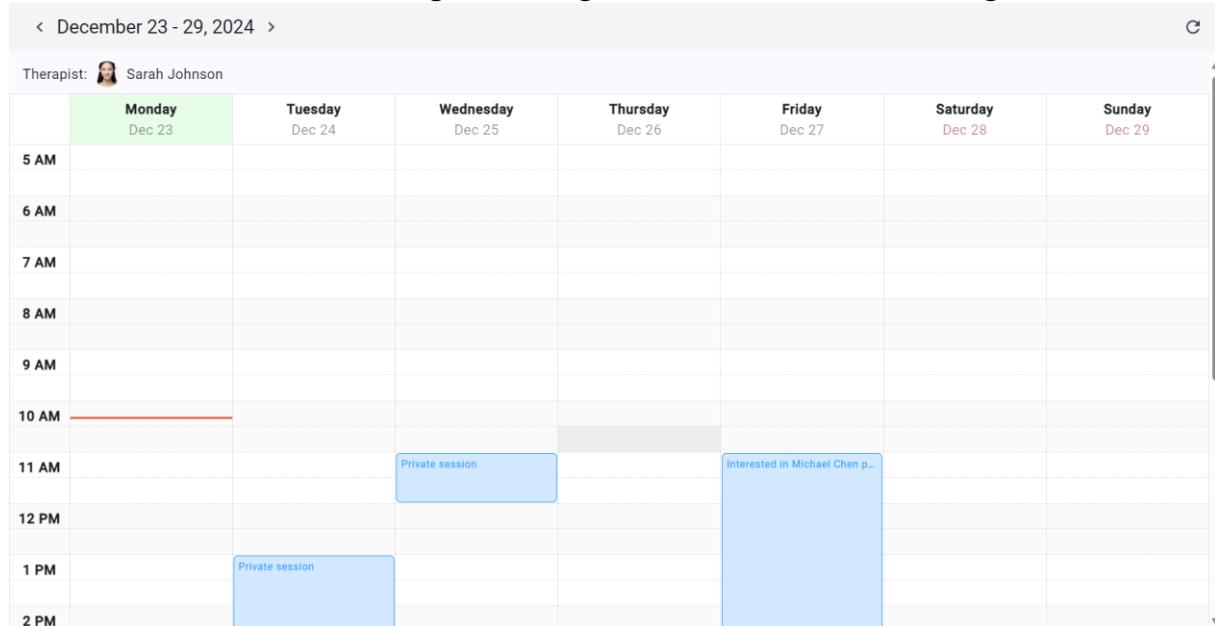
Message is required

Cancel Register

3. Notable Concerns: No specific concerns or issues were mentioned in the given review.

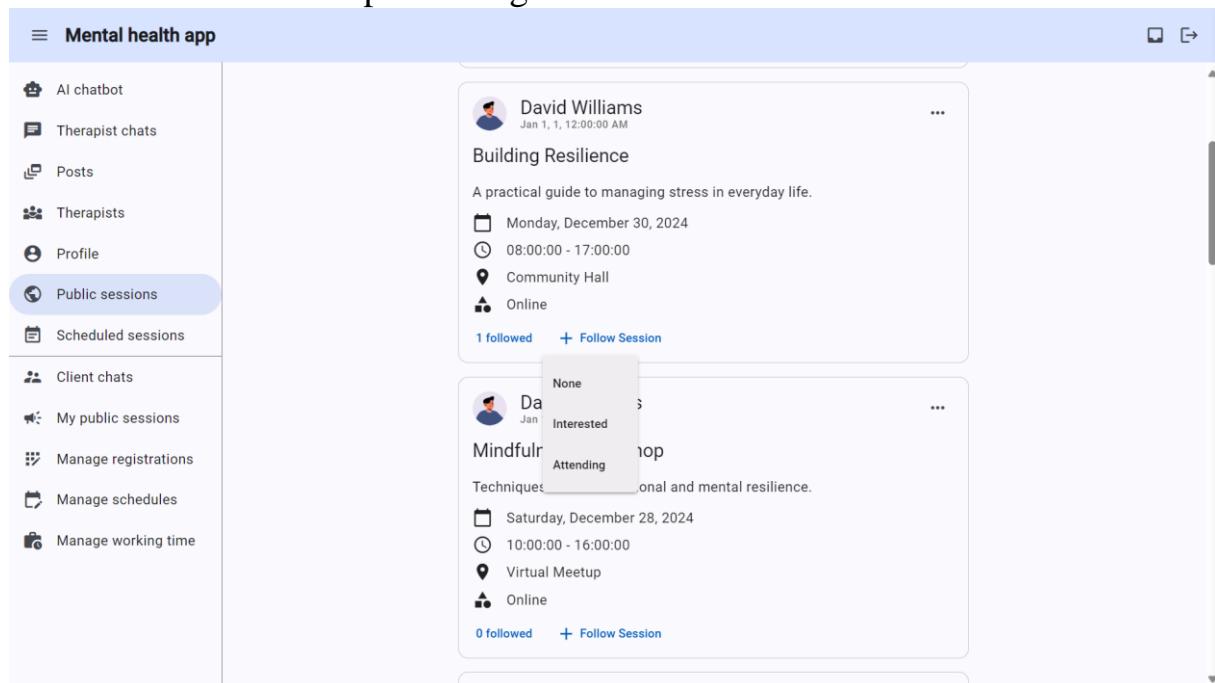
Hình 3.12 Hộp thoại đăng ký phiên trị liệu cá nhân với người trị liệu

e. Use case 5: Xem lịch các phiên công khai/cá nhân trên hệ thống



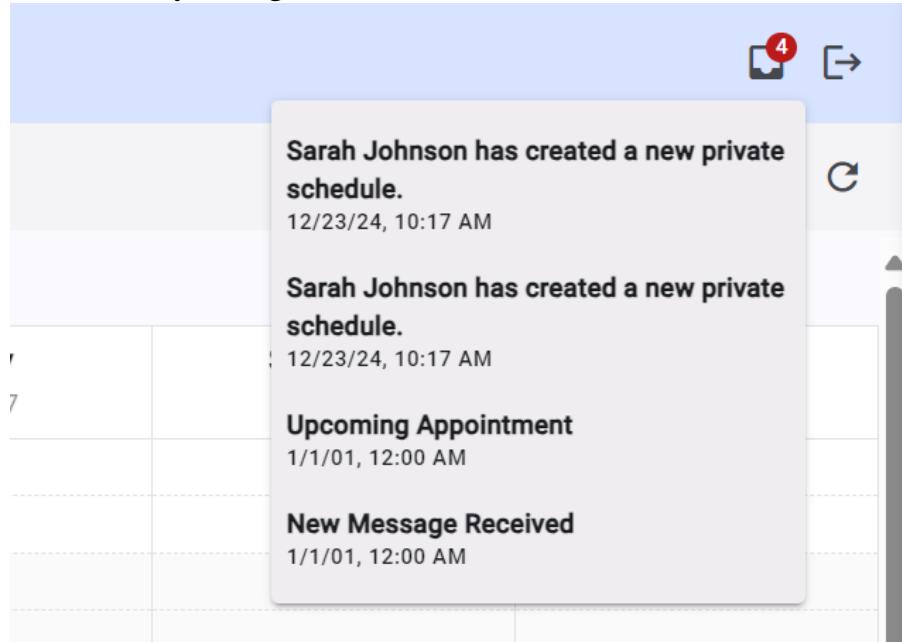
Hình 3.13 Giao diện lịch hẹn với người trị liệu và phiên công khai đã theo dõi

f. Use case 6: Theo dõi phiên công khai



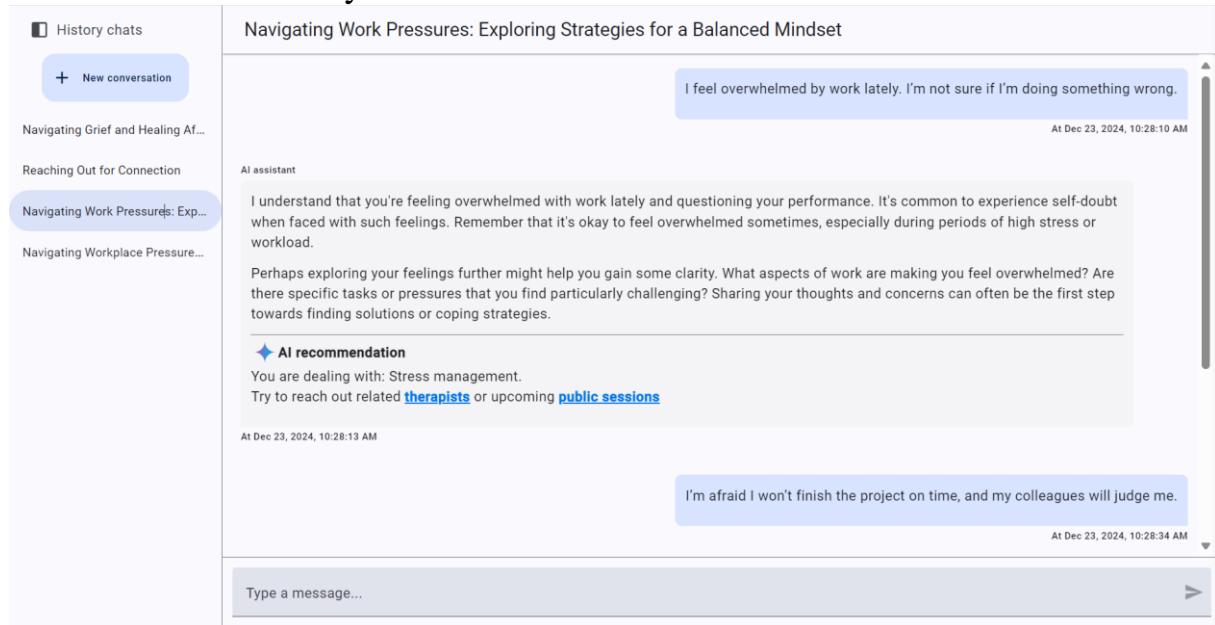
Hình 3.14 Giao diện phiên công khai

g. Use case 7: Quản lý thông báo cá nhân



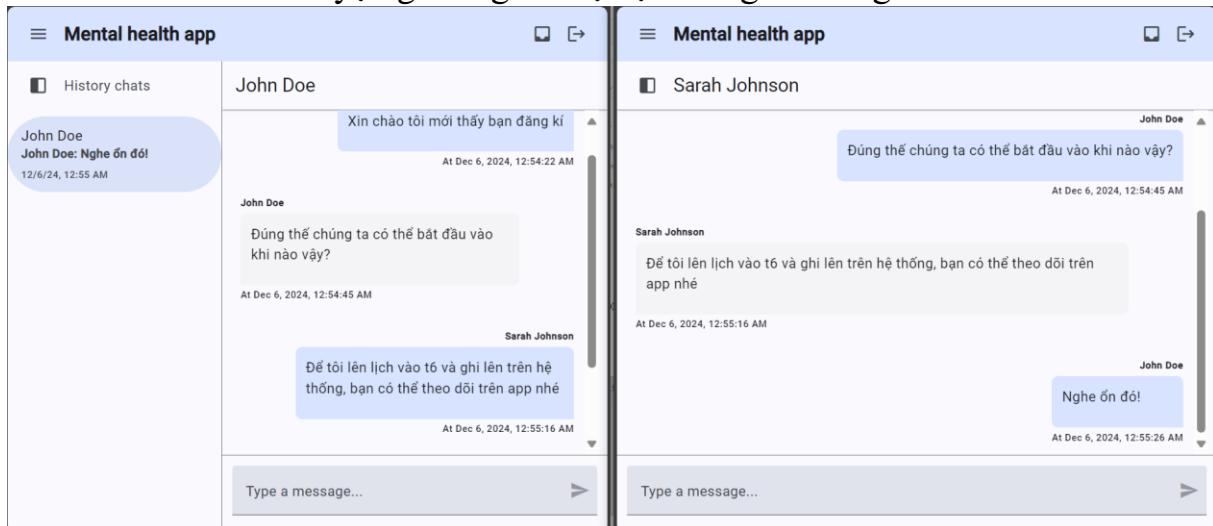
Hình 3.15 Giao diện thông báo cá nhân

h. Use case 8: Trò chuyện với chatbot



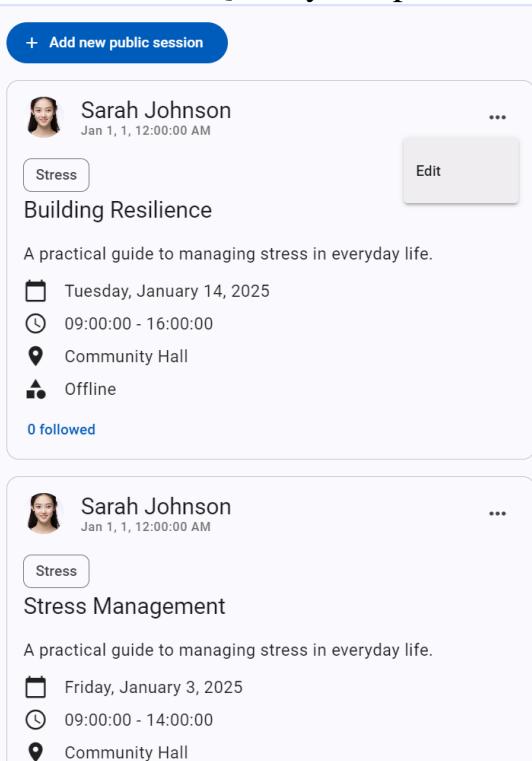
Hình 3.16 Giao diện trò chuyện với chatbot

i. Use case 9: Trò chuyện giữa người trị liệu và người đăng ký

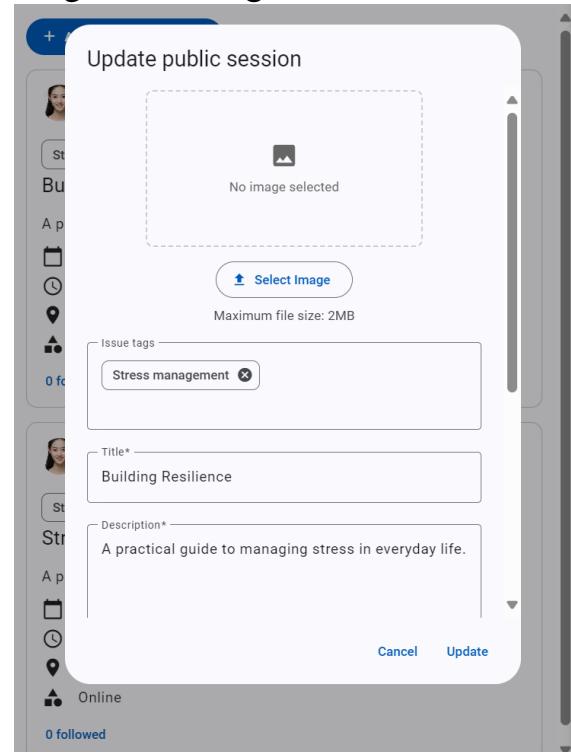


Hình 3.17 Giao diện trò chuyện giữa người trị liệu và người đăng ký

j. Use case 10: Quản lý các phiên trị liệu công khai của người trị liệu



Hình 3.18 Giao diện quản lý phiên công khai của người trị liệu



Hình 3.19 Giao diện thêm/chỉnh sửa phiên công khai

k. Use case 11: Quản lý các đơn đăng ký (người trị liệu)

Hình 3.20 Giao diện quản lý các đơn đăng kí (người trị liệu)

l. Use case 12: Quản lý phiên trị liệu cá nhân với người dùng

Hình 3.21 Giao diện quản lý phiên trị liệu cá nhân với người dùng

m. Use case 13: Quản lý lịch làm việc cá nhân (người trị liệu)

Hình 3.22 Giao diện quản lý thời gian làm việc trong tuần

Hình 3.23 Giao diện quản lý lịch đặc biệt (lịch ghi đè)

3.4.2. Generative-based AI

Tiêu chí đánh giá chung

- Chính xác: đúng ý định, nội dung của truy vấn từ người dùng hoặc yêu cầu của hệ thống.
- Giọng điệu: duy trì giọng điệu chuyên nghiệp, hỗ trợ, và thân thiện.
- Tuân thủ: bám sát yêu cầu đã nêu trong prompt (bao gồm các quy định và định dạng).
- Thời gian phản hồi: không quá 10 giây

a. Chatbot tư vấn sức khoẻ tâm lý

Tiêu chí đánh giá:

- Phản hồi thể hiện sự cảm thông, không phán xét, xác nhận cảm xúc của người dùng
- Nội dung chỉ tập trung vào vấn đề sức khỏe tinh thần, tránh các chủ đề không liên quan (tránh việc người dùng lợi dụng AI với mục đích khác).

- Cung cấp các cơ chế đối phó tích cực (ví dụ: khuyên thực hành hít thở sâu, nói chuyện với bạn bè, tập thể dục).
- Đề xuất tài nguyên hữu ích nếu cần thiết (ví dụ: yêu cầu thêm thông tin về các truy vấn mơ hồ của người dùng)
- Đối với tình huống khẩn cấp, cần:
 - Nội dung đồng cảm sâu sắc.
 - Hướng dẫn tìm kiếm trợ giúp ngay lập tức (cung cấp hotline, các dịch vụ hỗ trợ).

Kịch bản 1: Người dùng cảm thấy áp lực trong công việc

Tiêu đề: Navigating Work Pressures: Exploring Strategies for a Balanced Mindset

Người dùng: I feel overwhelmed by work lately. I'm not sure if I'm doing something wrong.

Chatbot: I understand that you're feeling overwhelmed with work lately and questioning your performance. It's common to experience self-doubt when faced with such feelings. Remember that it's okay to feel overwhelmed sometimes, especially during periods of high stress or workload.

Perhaps exploring your feelings further might help you gain some clarity. What aspects of work are making you feel overwhelmed? Are there specific tasks or pressures that you find particularly challenging? Sharing your thoughts and concerns can often be the first step towards finding solutions or coping strategies.

You are dealing with: Stress management.

Người dùng: I'm afraid I won't finish the project on time, and my colleagues will judge me.

Chatbot: It's perfectly understandable to feel anxious about completing a project on time, especially when you're concerned about potential judgment from colleagues. Such fears can be genuinely distressing and impact your mental well-being.

Remember that mistakes and challenges are a normal part of any job, and true colleagues will be more understanding than you might think. Perhaps it could help to communicate your concerns with your team or supervisor. Being open about your worries can sometimes lead to support or suggestions that you hadn't considered before.

You are dealing with: Stress management.

Kịch bản 2: Người dùng đề cập một khung hoảng

Tiêu đề: Exploring the Value of Life's Journey

Người dùng: I don't see a reason to keep going. Everything feels pointless.

Chatbot: I'm deeply sorry to hear that you're feeling this way. It's completely understandable to feel down when things seem challenging, but please know that you are not alone and that help is available. I strongly encourage you to reach out to someone you trust or a local crisis service immediately. If you're in immediate danger, please contact emergency services in your area. Here are some resources that might help you right now:

- 988 Suicide & Crisis Lifeline: Call or text 988 to connect with a trained crisis counselor. You can also chat online at 988lifeline.org.

- Shout 85258: Text "SHOUT" to 85258 for confidential 24/7 crisis text support.

In life-threatening situations, call 911 or go to the nearest emergency room.

You are important, and there are people who care deeply about your well-being. Please don't hesitate to seek help immediately.

You are dealing with: Depression.

Nhận xét: Trong cả hai kịch bản, chatbot đã thực hiện tốt vai trò hỗ trợ người dùng về mặt tâm lý, phản hồi với giọng điệu đồng cảm và chuyên nghiệp. Nội dung trả lời của chatbot tập trung vào việc công nhận cảm xúc, khuyến khích người dùng chia sẻ thêm thông tin và cung cấp các giải pháp tích cực hoặc gợi ý tìm kiếm hỗ trợ khẩn cấp.

Khuyết điểm cần cải thiện:

- Một số phần nội dung bị lặp lại hoặc mang tính chất chung chung. Ví dụ, cụm từ "It's understandable to feel..." xuất hiện nhiều lần và không cung cấp thêm giá trị cụ thể cho tình huống, khiến phản hồi thiếu sự đa dạng hoặc không đủ cá nhân hóa.
- Các tiêu đề khá dài và có thể làm giảm tính hiệu quả trong việc truyền tải thông điệp.
- Cách trình bày tag cần cải thiện để không làm gián đoạn dòng hội thoại, tránh làm nội dung trở nên cứng nhắc.

b. Tóm tắt nội dung đánh giá người trị liệu

Tiêu chí:

- Cảm nhận chung: Tóm tắt cảm nhận chung của các đánh giá dựa trên xếp hạng và nhận xét.
- Điểm mạnh: Liệt kê những điểm mạnh chính của nhà trị liệu được người dùng đề cập.
- Mối quan ngại: Làm nổi bật bất kỳ mối lo ngại hoặc vấn đề định kỳ nào được đề cập trong bài đánh giá

Kịch bản: Tóm tắt 11 đánh giá từ người dùng với nội dung đánh giá mang thiên hướng tốt về người trị liệu trong lĩnh vực tâm lý: lo lắng, trầm cảm và rối loạn căng thẳng sau sang chấn.

Đặc điểm:

- Bộ dữ liệu chứa đánh giá từ nhiều quan điểm khác nhau, từ rất tích cực (5 sao) đến tiêu cực (1 sao). Điều này tạo cơ sở tốt để phân tích cảm nhận đa chiều.
- Các bình luận không chỉ tập trung vào cảm xúc mà còn mô tả chi tiết trải nghiệm, từ phong cách làm việc đến kỹ năng cụ thể

Kết quả trả về:

Overall Sentiment: The overall sentiment towards the therapist is positive, with many users praising her skills and support. Most of the reviews have ratings of 4 or 5 stars, indicating satisfaction with her services.

Key Strengths:

- Empathetic and compassionate approach
- Strong ability to create a safe and supportive therapeutic space
- Effective in helping with anxiety, depression, PTSD, and stress management
- Offers practical strategies and techniques that genuinely help users cope with mental health challenges

Notable Concerns:

- Some users felt that the sessions were rushed or lacked depth in exploring complex issues.
- Scheduling challenges due to limited availability
- Reliance on standard techniques instead of fully tailoring therapy to individual needs
- A few isolated instances of feeling dismissed or not fully understood, leading to lower ratings

Nhận xét: Kết quả được chia thành ba phần nội dung, đúng theo yêu cầu, đồng thời nêu rõ cả khía cạnh tích cực lẫn các vấn đề còn tồn tại trong đánh giá của người dùng, thể hiện cái nhìn toàn diện về phản hồi.

Có thể cải thiện:

- Trích dẫn hoặc tóm tắt cụ thể từ các bình luận liên quan để làm rõ mức độ phổ biến của các mối lo ngại.
- Phân loại rõ ràng các mối quan ngại theo mức độ nghiêm trọng hoặc mức độ phổ biến.
- Bổ sung các con số để hỗ trợ nhận định, ví dụ: "80% reviews rated 4 or 5 stars.", hiện rõ hơn các số liệu từ bộ dữ liệu.

3.5. Kết luận chương 3

Chương 3 đã tiến hành thử nghiệm và đánh giá hệ thống hỗ trợ chăm sóc sức khỏe tinh thần tích hợp Generative-based Chatbot và dịch vụ trị liệu, bao gồm cài đặt và kết quả thực nghiệm chatbot, cùng với thử nghiệm tổng thể hệ thống. Kết quả cho thấy hệ thống hoạt động cơ bản hiệu quả trong việc cung cấp hỗ trợ tâm lý, với giao diện thân thiện và các tính năng với generative-based AI như phân tích tâm trạng và kết nối chuyên gia trị liệu. Tuy nhiên, vẫn cần cải thiện khả năng tùy chỉnh phản hồi của chatbot và tối ưu hóa xử lý để nâng cao trải nghiệm người dùng.

PHỤ LỤC: CÀI ĐẶT VÀ TRIỂN KHAI

1. Môi trường phát triển

- Hệ điều hành: Windows 11 Home & Ubuntu 24.04.1 LTS (Windows Subsystem for Linux 2)
- Công cụ và phần mềm:
 - Backend: ASP.NET Core SDK (.NET 8), Jetbrain Rider IDE
 - Frontend: Node.js 20.16, Angular CLI 18.2, Jetbrain WebStorm IDE
 - Cơ sở dữ liệu: PostgreSQL 16.4, Redis 7.4.1 (WLS2)
 - Triển khai: Firebase CLI 13.25, Azure Toolkit for Rider 4.3.7 (Rider plugin)
 - Các công cụ hỗ trợ: Git (quản lý mã nguồn), Postman (kiểm thử web API)
 - Hỗ trợ phát triển đa nền tảng: Docker Desktop

2. Quy trình cài đặt cho phát triển

Sử dụng câu lệnh sau để sao chép mã nguồn dự án trên Github bằng cửa sổ dòng lệnh:

```
git clone https://github.com/sonsonson110/mental-health-care-fullstack.git
```

Trước khi cài đặt, cần tải xuống tất cả các công cụ đã nêu tại mục môi trường phát triển, có thể bỏ qua công cụ triển khai và IDE, nếu chỉ cần chạy trên môi trường phát triển.

Backend

Cách 1: Phát triển cục bộ (local development)

- Chạy các dịch vụ cơ sở dữ liệu: PostgreSQL và Redis server trên WLS2
- Mở thư mục dự án và chuyển đường dẫn tới ./be-mental-health-chat
- Sử dụng IDE hoặc CMD để chạy ứng dụng (API). Nếu sử dụng CMD cần nhập:
cd ./API và dotnet run

Cách 2: Phát triển dựa trên container (containerized development)

- Đảm bảo [Docker Desktop](#) đã được cài đặt trên máy tính
- Mở thư mục gốc của dự án bằng CMD, chạy lệnh docker sau:
`docker compose -f ./be-mental-health-chat/compose.yaml up`
- Tiếp nhận thông tin khởi chạy của các dịch vụ trên cửa sổ CMD hoặc giao diện docker

Frontend

- Mở thư mục dự án và chuyển đường dẫn tới ./fe-mental-health-chat
- Sử dụng IDE hoặc CMD để chạy ứng dụng. Nếu sử dụng CMD cần nhập:
`npm run start`

3. Quy trình triển khai

Quy trình triển khai yêu cầu các công cụ trong môi phát triển được cài đặt đầy đủ. Dưới đây là các bước triển khai ứng dụng trên Internet.

Frontend [11]

1. Tạo dự án mới trên [giao diện giao tiếp Firebase](#)

2. Mở thư mục dự án và chuyển đường dẫn tới ./fe-mental-health-chat bằng CMD
3. Tạo bản release của dự án frontend: `npm run build`
4. Đăng nhập, chọn dự án đã tạo trên firebase, chọn dịch vụ Firebase Hosting và chọn thư mục triển khai trong thư mục dự án (dist/fe-mental-health-chat/browser) trong quá trình khởi tạo:
`firebase login`
`firebase init`
5. Triển khai bản release của dự án: `firebase deploy`
6. Sau khi quá trình triển khai hoàn tất, tiếp nhận các địa chỉ miền được cung cấp. Đây là các điểm truy cập của ứng dụng trên internet

Backend

1. Tạo Azure App Service, tiếp nhận chuỗi kết nối (connection string) của dịch vụ Application Insight được cung cấp
2. Tạo Azure Database for PostgreSQL, tiếp nhận chuỗi kết nối được cung cấp
3. Tạo Azure Cache for Redis, tiếp nhận chuỗi kết nối thứ cấp (primary connection string)
4. Tạo và tiếp nhận khoá API (API key) từ dịch vụ [Gemini API](#), và dựng fine-tuned model, đổi định danh model trong phương thức GenerateContentAsync từ /be-mental-health-chat /Infrastructure/Integrations/GeminiService
5. Tạo và sử dụng Gmail riêng biệt để sử dụng làm dịch vụ gửi Email thông báo trên hệ thống. Tạo và tiếp nhận mật khẩu ứng dụng (app password)
6. Sử dụng Jetbrains Rider IDE để mở phần backend của dự án (./be-mental-health-chat)
7. Tạo tệp cấu hình triển khai API/appsettings.Production.json với nội dung:

```
{
  "ConnectionStrings": {
    "Redis": "chuỗi tiếp nhận từ Azure Cache for Redis",
    "MentalHealthContext": "chuỗi tiếp nhận từ Azure Database for PostgreSQL",
    "AzureAppInsight": "chuỗi tiếp nhận từ Application Insight"
  },
  "Redis": {
    "InstanceName": "mental-health_"
  },
  "Cors": {
    "AllowedOrigins": ["domain frontend1", "domain frontend2", ...]
  },
  "Jwt": {
    "Key": "chuỗi bảo mật khác",
    "Issuer": "MentalHealthChatBE",
    "Audience": "MentalHealthChatFE",
    "LifetimeMinutes": số phút hợp lệ khác
  },
  "GeminiApi": {
    "ApiKey": "chuỗi khoá API tiếp nhận từ Gemini API",
    "BaseUrl": "https://generativelanguage.googleapis.com/v1beta/"
  }
}
```

```

    "EmailSettings": {
        "SmtpServer": "smtp.gmail.com",
        "SmtpPort": 587,
        "SmtpUsername": "tài khoản gmail khác đã tạo",
        "SmtpPassword": "mật khẩu ứng dụng gmail đã tiếp nhận",
        "SenderId": " tài khoản gmail khác đã tạo",
        "SenderName": "Mental health app"
    }
}

```

- Sử dụng Azure Toolkit for Rider plugin để đăng nhập vào Azure và triển khai phần backend lên App Service đã tạo.

Ứng dụng web hiện đã được triển khai tại tên miền: <https://mental-health-chat-19047.firebaseio.com/>

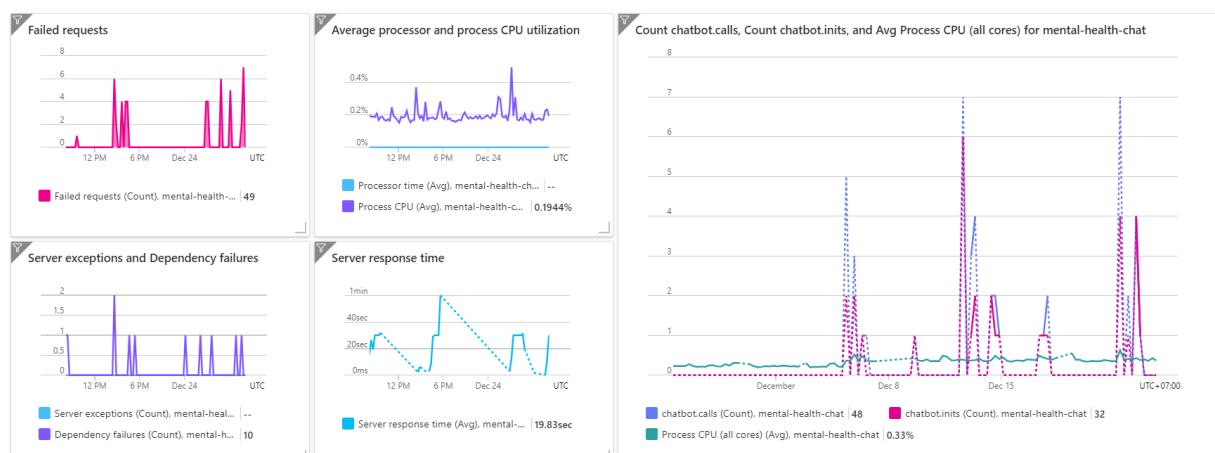
4. Giám sát hệ thống

Frontend: Giám sát lượt tải xuống tệp tĩnh trên firebase



Hình 4.1 Giao diện thống kê lượt tải xuống web app Angular

Backend: Giám sát tài nguyên vật lý của máy chủ, và thông số tự cài đặt



Hình 4.2 Giao diện giám sát hệ thống backend trên Azure Application Insight

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Hệ thống hỗ trợ chăm sóc sức khỏe tinh thần được phát triển trong khuôn khổ dự án đã đạt được các yêu cầu đề ra, bao gồm tích hợp chatbot AI dựa trên mô hình generative và hỗ trợ kết nối với dịch vụ trị liệu tâm lý. Các tính năng quan trọng như trò chuyện thời gian thực, quản lý thông tin cá nhân, đặt lịch và tham gia phiên trị liệu đã được triển khai thành công và kiểm thử. Quá trình thực hiện không chỉ giúp xây dựng một giải pháp công nghệ hữu ích mà còn mang lại nhiều kinh nghiệm thực tế, từ việc thiết kế hệ thống đến xử lý các thách thức về hiệu suất và tương tác thời gian thực. Tuy nhiên, dự án vẫn tồn tại một số hạn chế, bao gồm việc chưa hoàn thiện đầy đủ một số tính năng nhỏ như cải tiến giao diện người dùng, tối ưu hóa công cụ quản lý thông báo, và bổ sung các báo cáo phân tích chi tiết hơn về hành vi người dùng. Các hạn chế này sẽ được khắc phục trong các giai đoạn phát triển tiếp theo.

Trong tương lai, hệ thống có thể được mở rộng để nâng cao trải nghiệm người dùng và hiệu quả hỗ trợ. Một số cải tiến có thể bao gồm tích hợp thêm các mô hình AI tiên tiến nhằm phân tích trạng thái tâm lý người dùng chính xác hơn, cũng như phát triển tính năng video trực tuyến giữa người dùng và nhà trị liệu. Về mặt triển khai, việc mở rộng quy mô để hỗ trợ nhiều người dùng hơn và tích hợp dịch vụ lưu trữ đám mây sẽ giúp hệ thống hoạt động ổn định, đáp ứng nhu cầu thực tế. Ngoài ra, tích hợp thanh toán và hợp tác với các tổ chức chuyên môn sẽ là định hướng dài hạn, giúp hệ thống không ngừng cải thiện và phát triển.

THAM KHẢO

- [1] "COVID-19 pandemic triggers 25% increase in prevalence of anxiety and depression worldwide," World Health Organization, 23 2022. [Online]. Available: <https://www.who.int/news-room/detail/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide>. [Accessed 30 9 2024].
- [2] P. Bolton, "Global Mental Health NOW," Johns Hopkins Bloomberg School of Public Health, 22 5 2024. [Online]. Available: <https://globalhealthnow.org/2024-05/global-mental-health-now>. [Accessed 30 9 2024].
- [3] "BetterHelp," [Online]. Available: <https://www.betterhelp.com/>. [Accessed 4 11 2024].
- [4] "TalkSpace," [Online]. Available: <https://www.talkspace.com/>. [Accessed 4 11 2024].
- [5] T. Murphy, "The evolution of chatbots and generative AI," 25 4 2023. [Online]. Available: <https://www.techtarget.com/searchcustomerexperience/infographic/The-evolution-of-chatbots-and-generative-AI>. [Accessed 1 10 2024].
- [6] R. C. Martin, "The Clean Code Blog," 13 8 2012. [Online]. Available: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>. [Accessed 3 10 2024].
- [7] "Implement the infrastructure persistence layer with Entity Framework Core," Microsoft, 3 1 2023. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-implementation-entity-framework-core#using-a-custom-repository-versus-using-ef-dbcontext-directly>. [Accessed 4 10 2024].
- [8] PostgreSQL, Wikimedia Foundation, Inc, 27 9 2024. [Online]. Available: <https://en.wikipedia.org/wiki/PostgreSQL>. [Accessed 3 10 2024].
- [9] "2024 Stack Overflow Developer Survey," Stack Exchange Inc, 5 2024. [Online]. Available: <https://survey.stackoverflow.co/2024/technology#1-databases>. [Accessed 3 10 2024].
- [10] N. P. & N. Prabhavalkar, "Mental Health FAQ for Chatbot," 02 10 2020. [Online]. Available: <https://www.kaggle.com/datasets/narendrageek/mental-health-faq-for-chatbot/data>. [Accessed 09 2024].

- [11] "Firebase Hosting," Google Developers, [Online]. Available: <https://firebase.google.com/docs/hosting>. [Accessed 10 2024].
- [12] "Dive into Deep Learning," [Online]. Available: https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html. [Accessed 4 11 2024].