

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**XÂY DỰNG TRANG WEB ĐẶT PHÒNG KHÁM VÀ**  
**ĐIỀU TRỊ TẠI BỆNH VIỆN TÍCH HỢP CHATBOT**  
**BẰNG SPRING BOOT VÀ ANGULAR**

**Giáo viên hướng dẫn** : TS. Nguyễn Mạnh Cường  
**Sinh viên thực hiện** : Vương Trí Tín  
**Mã sinh viên** : 2021603785

Hà Nội - 2025

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

---



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**XÂY DỰNG TRANG WEB ĐẶT PHÒNG KHÁM VÀ**  
**ĐIỀU TRỊ TẠI BỆNH VIỆN TÍCH HỢP CHATBOT**  
**BẰNG SPRING BOOT VÀ ANGULAR**

<b>Giáo viên hướng dẫn</b>	<b>: TS. Nguyễn Mạnh Cường</b>
<b>Sinh viên thực hiện</b>	<b>: Vương Trí Tín</b>
<b>Mã sinh viên</b>	<b>: 2021603785</b>

Hà Nội - 2025

## MỤC LỤC

DANH MỤC HÌNH ẢNH.....	5
DANH MỤC BẢNG BIỂU .....	6
LỜI CẢM ƠN .....	7
LỜI NÓI ĐẦU .....	8
CHƯƠNG 1: KHẢO SÁT HỆ THỐNG .....	11
1.1. Khảo sát nhu cầu đặt và lên lịch khám hiện nay .....	11
1.2. Xác định phạm vi và quyền hạn của người dùng trong hệ thống .....	13
1.3. Xác định các yêu cầu của hệ thống .....	15
1.4. Tổng kết chương 1 .....	18
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	19
2.1. Biểu đồ Use case .....	19
2.1.1. Các Use case chính .....	19
2.1.2. Quan hệ giữa các Use case .....	19
2.2. Mô tả chi tiết các Use case .....	25
2.2.1. Mô tả use case: Đăng nhập .....	25
2.2.2. Mô tả usecase: Đăng ký .....	26
2.2.3. Mô tả use case: Trợ lý ảo .....	27
2.2.4. Mô tả Use case: Đặt lịch .....	28
2.2.5. Mô tả use case: Cập nhật thông tin cá nhân .....	29
2.2.6. Mô tả use case: Quản lý lịch khám .....	29
2.2.7. Mô tả use case: Tìm kiếm .....	30
2.2.8. Mô tả use case: Xem chuyên gia y tế .....	31
2.2.9. Mô tả use case: Lấy lại mật khẩu .....	32
2.2.10. Mô tả use case: Quản lý người dùng .....	33
2.2.11. Mô tả use case: Quản lý tin tức .....	34
2.2.12. Mô tả use case: Quản lý bình luận .....	35
2.2.13. Mô tả use case: Quản lý trang chủ .....	36
2.3. Các biểu đồ trình tự Use case .....	38
2.3.1. Biểu đồ đăng nhập .....	38
2.3.2. Biểu đồ đăng ký .....	39
2.3.3. Biểu đồ lấy lại mật khẩu .....	40
2.3.4. Biểu đồ cập nhật thông tin cá nhân .....	41
2.3.5. Biểu đồ đặt lịch khám .....	42
2.3.6. Biểu đồ Tư vấn trợ lý ảo .....	43
2.3.7. Biểu đồ tìm kiếm chuyên gia theo phòng ban .....	44
2.3.8. Biểu đồ thêm bình luận .....	45

2.3.9. Biểu đồ xem tin tức .....	46
2.3.10. Biểu đồ tìm kiếm .....	47
2.3.11. Biểu đồ xem chuyên gia .....	47
2.3.12. Biểu đồ quản lý lịch khám .....	48
2.3.13. Biểu đồ quản lý tài khoản người dùng .....	49
2.3.14. Biểu đồ quản lý tài khoản bác sĩ .....	50
2.3.15. Biểu đồ quản lý tin tức .....	51
2.3.16. Biểu đồ quản lý quảng cáo trên trang chủ .....	53
2.4. Biểu đồ lớp .....	55
2.5. Thiết kế cơ sở dữ liệu .....	56
2.5.1. Mô hình quan hệ dữ liệu .....	56
2.5.2. Chi tiết các bảng .....	56
2.6. Xây dựng kiến trúc hệ thống .....	68
2.6.1. Kiến trúc hệ thống của node RAG .....	69
2.6.2. Kiến trúc hệ thống của Node Sql .....	71
CHƯƠNG 3. CÀI ĐẶT HỆ THỐNG VÀ MỘT SỐ KẾT QUẢ .....	72
3.1. Công nghệ đã sử dụng .....	72
3.1.1. Phía backend .....	72
3.1.2. Phía Front end .....	73
3.1.3. Phía Database .....	74
3.1.4. Phía Trợ lí ảo .....	76
3.2. Môi trường đã sử dụng .....	76
3.3. Một số kết quả đạt được .....	77
KẾT LUẬN .....	86
TÀI LIỆU THAM KHẢO .....	88

## DANH MỤC HÌNH ẢNH

Hình 2.1. Biểu đồ use case mức tổng quát .....	19
Hình 2.2. Quan hệ của use case quản lý bình luận .....	20
Hình 2.3. Quan hệ use case quản lý người dùng .....	20
Hình 2.4. Quan hệ use case quản lý tin tức .....	21
Hình 2.5. Quan hệ use case quản lý Quảng cáo .....	22
Hình 2.6. Quan hệ use case quản lý lịch khám .....	23
Hình 2.7. Quan hệ use case quản lý chuyên khoa .....	24
Hình 2.8. Quan hệ use case quản lý bác sĩ .....	24
Hình 2.9. Biểu đồ trình tự use case đăng nhập .....	38
Hình 2.10. Biểu đồ trình tự use case đăng ký .....	39
Hình 2.11. Biểu đồ trình tự use case lấy lại mật khẩu .....	40
Hình 2.12. Biểu đồ trình tự use case cập nhật thông tin cá nhân .....	41
Hình 2.13. Biểu đồ trình tự use case đặt lịch khám .....	42
Hình 2.14. Biểu đồ trình tự use case tư vấn trợ lý ảo .....	43
Hình 2.15. Biểu đồ trình tự use case tìm kiếm chuyên gia theo phòng ban .....	44
Hình 2.16. Biểu đồ trình tự use case thêm bình luận .....	45
Hình 2.17. Biểu đồ trình tự use case xem tin tức .....	46
Hình 2.18. Biểu đồ trình tự use case tìm kiếm .....	47
Hình 2.19. Biểu đồ trình tự use case xem chuyên gia .....	47
Hình 2.20. Biểu đồ trình tự use case xem lịch sử khám .....	48
Hình 2.21. Biểu đồ trình tự use case cập nhật lịch khám .....	48
Hình 2.22. Biểu đồ trình tự use case kích hoạt hoặc cấm khách hàng .....	49
Hình 2.23. Biểu đồ trình tự use case cấp quyền Bác sĩ .....	49
Hình 2.24. Biểu đồ trình tự use case cấm tài khoản bác sĩ .....	50
Hình 2.25. Biểu đồ trình tự use case xóa bài báo .....	51
Hình 2.26. Biểu đồ trình tự use case thêm bài báo .....	52
Hình 2.27. Biểu đồ trình tự use case thêm mới quảng cáo .....	53
Hình 2.28. Biểu đồ trình tự use case xóa quảng cáo .....	54
Hình 2.29. Biểu đồ lớp chính của hệ thống .....	55
Hình 2.30. Mô hình dữ liệu của hệ thống .....	56
Hình 2.31. Mô hình kiến trúc chính hệ thống AI .....	68
Hình 2.32. Mô hình kiến trúc hệ thống RAG .....	69
Hình 3.23. Mô hình kiến trúc hệ thống truy vấn Sql .....	71
Hình 3.24. Giao diện đăng nhập .....	75
Hình 3.25. Giao diện trang quản lý chung của Admin .....	76
Hình 3.26. Giao diện quản lý người dùng của Admin .....	77

Hình 3.27. Giao diện quản lý Bác sĩ của Admin .....	77
Hình 3.28. Giao diện quản lý tin tức của Admin .....	77
Hình 3.29. Giao diện trang quản lý lịch khám của Admin .....	71
Hình 3.30. Giao diện trang lịch khám cá nhân của người dùng .....	71
Hình 3.31. Giao diện trang quản lý lịch khám sắp tới của Bác sĩ .....	79
Hình 3.32. Giao diện trang chủ .....	79
Hình 3.33. Giao diện trang giới thiệu về Bệnh viện .....	80
Hình 3.34. Giao diện lựa chọn chuyên khoa phù hợp .....	80
Hình 3.35. Giao diện đặt lịch .....	81
Hình 3.36. Giao diện theo dõi các dịch vụ của Bệnh viện .....	81
Hình 3.37. Giao diện tư vấn với trợ lý ảo .....	82
Hình 3.38. Giao diện xem tin tức Y khoa .....	82
Hình 3.39. Giao diện cập nhật hồ sơ cá nhân .....	83
Hình 3.40. Giao diện đăng ký .....	83
Hình 3.41. Mô hình kiến trúc hệ thống truy vấn Sql .....	84

## DANH MỤC BẢNG BIỂU

Bảng 2.1. Chi tiết bảng user .....	55
Bảng 2.2. Chi tiết bảng bảng token .....	57
Bảng 2.3. Chi tiết bảng bảng role .....	58
Bảng 2.4. Chi tiết bảng bảng review .....	59
Bảng 2.5. Chi tiết bảng bảng doctor .....	60
Bảng 2.6. Chi tiết bảng bảng department .....	61
Bảng 2.7. Chi tiết bảng bảng appointment .....	62
Bảng 2.8. Chi tiết bảng bảng EmailVerification .....	64
Bảng 2.9. Chi tiết bảng bảng Article .....	65

## LỜI CẢM ƠN

Lời đầu tiên cho phép em gửi lời cảm ơn sâu sắc tới các thầy cô trong Trường Đại học Công nghệ thông tin và Truyền thông - Đại học Công Nghiệp Hà Nội, những người đã hết mình truyền đạt và chỉ dẫn cho em những kiến thức, những bài học quý báu và bổ ích. Đặc biệt em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên Nguyễn Mạnh Cường người trực tiếp hướng dẫn, chỉ bảo em trong suốt quá trình học tập, nghiên cứu và hoàn thành được đồ án. Sau nữa, em xin gửi tình cảm sâu sắc tới gia đình và bạn bè vì đã luôn bên cạnh khuyến khích, động viên, giúp đỡ cả về vật chất lẫn tinh thần cho em trong suốt quy trình học tập để em hoàn thành tốt việc học tập của bản thân.

Trong quá trình nghiên cứu và làm đề tài, do thời gian có hạn và năng lực, kiến thức, trình độ bản thân em còn hạn hẹp nên không tránh khỏi những thiếu sót và em mong nhận được sự thông cảm và những góp ý từ quý thầy cô cũng như các bạn trong lớp.

Em xin trân trọng cảm ơn!

**Sinh viên thực hiện**

Vương Trí Tín



## LỜI NÓI ĐẦU

Hiện nay, nhu cầu đặt lịch khám và điều trị tại bệnh viện ngày càng gia tăng. Tuy nhiên, quá trình tìm kiếm, đăng ký lịch khám vẫn gặp nhiều khó khăn do thiếu thông tin chính xác, minh bạch và được cập nhật thường xuyên. Người bệnh thường phải xếp hàng chờ đợi lâu, gặp khó khăn trong việc lựa chọn bác sĩ, cũng như không thể chủ động sắp xếp thời gian khám bệnh hợp lý. Xuất phát từ thực tế đó, em quyết định thực hiện đề tài "Xây dựng trang web đặt phòng khám và điều trị tại bệnh viện tích hợp Chatbot bằng Spring Boot và Angular" với mục tiêu tạo ra một nền tảng trực tuyến giúp việc tìm kiếm, đặt lịch khám và điều trị trở nên dễ dàng, nhanh chóng và hiệu quả hơn.

Website sẽ cung cấp đầy đủ thông tin về các dịch vụ khám chữa bệnh, lịch làm việc của bác sĩ, chi phí điều trị, cùng với hệ thống đặt lịch trực tuyến giúp bệnh nhân chủ động sắp xếp thời gian khám. Ngoài ra, hệ thống còn hỗ trợ bệnh viện trong việc quản lý lịch hẹn, tối ưu hóa quy trình tiếp nhận bệnh nhân, giảm tải áp lực cho nhân viên y tế. Một điểm nổi bật của website là tích hợp Chatbot hỗ trợ tư vấn, giải đáp thắc mắc của bệnh nhân một cách tự động, giúp nâng cao trải nghiệm người dùng.

Bên cạnh đó, hệ thống sẽ tích hợp các tính năng nâng cao như tìm kiếm bác sĩ theo chuyên khoa, đánh giá chất lượng dịch vụ từ bệnh nhân, và gợi ý lịch khám phù hợp dựa trên tình trạng sức khỏe. Việc triển khai hệ thống không chỉ giúp bệnh nhân tiết kiệm thời gian mà còn góp phần số hoá quy trình khám chữa bệnh, nâng cao chất lượng dịch vụ y tế.

Chính vì những lý do trên, đề tài này không chỉ có ý nghĩa thực tiễn cao mà còn là cơ hội để nhóm áp dụng các kiến thức đã học vào thực tế, đồng thời tiếp cận với các công nghệ hiện đại trong phát triển web.

Báo cáo này trình bày toàn bộ kết quả đạt được trong suốt quá trình thực hiện đề tài "Xây dựng trang web đặt phòng khám và điều trị tại bệnh viện tích

hợp Chatbot bằng Spring Boot và Angular". Nội dung báo cáo bao gồm ba chương chính và phần kết luận.

## **Chương 1: Khảo sát hệ thống**

Trong chương đầu, em sẽ giới thiệu bối cảnh và sự cần thiết của việc xây dựng một website hỗ trợ đặt lịch khám bệnh. Em sẽ xác định bài toán đặt ra, phân tích nhu cầu thực tế của bệnh nhân trong việc đặt lịch khám, cũng như những khó khăn thường gặp khi sử dụng các phương pháp truyền thống như gọi điện đặt lịch, đăng ký trực tiếp tại bệnh viện. Đồng thời, chương này cũng sẽ đề cập đến mục tiêu và phạm vi của đề tài, cùng với các giải pháp đã được áp dụng trước đây.

## **Chương 2: Phân tích và thiết kế hệ thống**

Từ những dữ liệu đã thu thập qua khâu khảo sát, em sẽ tiến hành vẽ biểu đồ chức năng của hệ thống ở mức tổng quát, mô tả chi tiết các chức năng, vẽ biểu đồ trình tự, biểu đồ lớp,... Từ biểu đồ lớp em thiết kế cơ sở dữ liệu phù hợp cho trang web. Chương này cũng sẽ hướng dẫn cách thiết kế, xây dựng một mô hình Chatbot từ giai đoạn phát triển đến khi có thể hoạt động thực tế trên website.

## **Chương 3: Cài đặt hệ thống và một số kết quả**

Đến đây, từ những kết quả thu được ở bước phân tích và thiết kế hệ thống, em sẽ tập trung vào việc cài đặt, triển khai hệ thống đã được đưa ra. Ứng dụng các công nghệ như Spring boot, Angular, Tailwindcss, Mysql, MongoDB, LangChain, LangGraph.

## **Kết luận**

Phần này sẽ tổng hợp những kết quả đã đạt được, đánh giá hiệu quả của hệ thống và rút ra bài học kinh nghiệm trong quá trình thực hiện đề tài. Đồng thời, em cũng sẽ đề xuất các hướng phát triển mở rộng trong tương lai, như tích

hợp trí tuệ nhân tạo để dự đoán lịch khám phù hợp, triển khai ứng dụng di động hoặc bổ sung các tính năng hỗ trợ y tế trực tuyến.

Việc thực hiện đề tài này không chỉ giúp em củng cố kiến thức về lập trình web, phát triển hệ thống mà còn mang lại giá trị thực tiễn, góp phần nâng cao chất lượng dịch vụ y tế và cải thiện trải nghiệm đặt lịch khám cho bệnh nhân.

## CHƯƠNG 1: KHẢO SÁT HỆ THỐNG

### 1.1. Khảo sát nhu cầu đặt và lên lịch khám hiện nay

Trong bối cảnh nhu cầu khám và điều trị bệnh ngày càng tăng cao, việc đặt lịch khám tại bệnh viện trở thành một vấn đề cấp thiết. Hiện nay, nhiều bệnh nhân phải đối mặt với các trở ngại như thời gian chờ đợi lâu, quá trình đăng ký không minh bạch, cũng như thiếu thông tin cập nhật về lịch khám và các dịch vụ y tế cần thiết. Tình trạng quá tải tại các bệnh viện, đặc biệt là ở các tuyến trung ương và thành phố lớn như Hà Nội và TP.HCM, đã trở thành vấn đề nan giải trong nhiều năm qua. Bệnh nhân thường phải chờ đợi hàng giờ để được khám, trong khi thời gian thực tế được bác sĩ thăm khám lại rất ngắn, đôi khi chỉ dưới một phút. Điều này không chỉ gây mệt mỏi cho người bệnh mà còn ảnh hưởng đến chất lượng dịch vụ y tế và sự hài lòng của bệnh nhân. Nguyên nhân của tình trạng này bao gồm sự thiếu hụt về nhân lực y tế, cơ sở vật chất hạn chế, và quy trình đăng ký khám bệnh còn nhiều bất cập. Nhiều bệnh viện vẫn chưa áp dụng hiệu quả công nghệ thông tin trong việc quản lý và điều phối lịch khám, dẫn đến việc người bệnh phải xếp hàng dài chờ đợi mà không có thông tin rõ ràng về thời gian khám của mình.

#### ❖ Thực trạng đặt và lên lịch khám hiện nay của các bệnh viện truyền thống

- Người bệnh phải chờ máy bận, không biết trước lịch trống, dễ bị lừa thông tin.
- Phải xếp hàng, điền tờ khai tay, phụ thuộc giờ hành chính; không chủ động điều chỉnh khi có thay đổi đột xuất.
- Cập nhật không liên tục, giao diện hạn chế tính năng tìm kiếm/filter, khó xem trước lịch bác sĩ và dịch vụ.
- Thông tin chưa đầy đủ, thiếu cập nhật kịp thời.
- Quy trình đăng ký phức tạp, dễ xảy ra nhầm lẫn.

- Thiếu công cụ hỗ trợ tư vấn tự động, khiến người dùng không nhận được sự hỗ trợ cần thiết trong quá trình đặt lịch.
- Lễ tân vẫn phải thao tác trên sổ sách hoặc file Excel, dễ sai sót và mất nhiều thời gian.
- Khó theo dõi số lượng lịch hẹn, tỷ lệ “nhỡ hẹn” hay báo cáo doanh thu theo ngày/tháng.
- Thông tin trao đổi giữa lễ tân và bác sĩ, giữa bệnh viện và bệnh nhân không được lưu trữ tập trung, khiến quy trình xử lý khiếu nại, đổi lịch kém hiệu quả.

Chính vì vậy với sự phát triển nhanh chóng của công nghệ thông tin đã mở ra nhiều cơ hội xây dựng phát triển các ứng dụng trong lĩnh vực y tế giúp cải thiện dịch vụ và chất lượng cho các bệnh viện truyền thống. Việc số hoá các quy trình quản lý và đặt lịch khám giúp tối ưu hoá thời gian và nguồn lực cho cả bệnh viện và người bệnh.

❖ Mặc dù các nền tảng trực tuyến đã xuất hiện, nhưng vẫn tồn tại một số hạn chế:

- Khi bác sĩ thay ca hoặc có sự cố, người bệnh không được thông báo kịp thời.
- Nhiều ứng dụng còn cồng kềnh, thiếu hướng dẫn rõ ràng, gây bối rối cho người không rành công nghệ.
- Ít nền tảng cho phép bệnh nhân để lại phản hồi, đánh giá chất lượng khám chữa, nên người dùng khó tham khảo.

Trước tình hình đó, đề tài “Xây dựng trang web đặt phòng khám và điều trị tại bệnh viện tích hợp Chatbot bằng Spring Boot và Angular” được đề xuất nhằm tạo ra một giải pháp công nghệ tiên tiến, giúp kết nối bệnh nhân với các cơ sở y tế một cách nhanh chóng, hiệu quả và tiện lợi.

❖ Mục tiêu của website

- Cung cấp thông tin minh bạch về bác sĩ, chuyên khoa, khung giờ trống và chi phí dịch vụ.
- Hỗ trợ tìm kiếm và lọc theo chuyên khoa, bác sĩ, ngày giờ, gói khám hoặc nơi khám.
- Kết nối trực tiếp giữa bệnh nhân và cơ sở y tế, đảm bảo giao dịch nhanh gọn, rõ ràng.
- Tích hợp đánh giá và phản hồi từ bệnh nhân trước, giúp người dùng có cái nhìn khách quan.
- Tự động hóa tư vấn và nhắc nhở qua Chatbot, giảm tải khâu hành chính và nâng cao trải nghiệm.
- Công cụ quản lý cho bệnh viện: Dashboard trực quan, thống kê lịch hẹn, nhớ hẹn, doanh thu và dự báo công suất khám.

Với nhu cầu đặt lịch khám trực tuyến ngày càng tăng cao, đặt ra yêu cầu cần có một nền tảng chuyên nghiệp, minh bạch và hiệu quả hơn. Việc phát triển một website đặt lịch thông minh không chỉ giúp người bệnh dễ dàng tiếp cận thông tin mà còn hỗ trợ người bệnh tiếp cận các dịch vụ dễ dàng và tiện lợi, giúp tối ưu hóa quá trình, nâng cao chất lượng phục vụ, hiệu quả vận hành của bệnh viện. Với bài tập lớn này, em nghiên cứu mong muốn xây dựng một hệ thống web tối ưu, giúp cải thiện chất lượng đặt lịch khám, góp phần tạo ra một môi trường an toàn và tiện lợi hơn cho cộng đồng.

## **1.2. Xác định phạm vi và quyền hạn của người dùng trong hệ thống**

### **➤ Khách (Guest) - Người chưa đăng nhập**

Đây là những người truy cập website mà chưa có tài khoản hoặc chưa đăng nhập vào hệ thống. Mặc dù chưa đăng nhập, họ vẫn có thể sử dụng một số chức năng cơ bản như:

- Xem danh sách các chuyên gia y tế: Người dùng có thể duyệt qua danh sách các chuyên gia được đăng tải trên hệ thống.
- Hỏi đáp tư vấn của trợ lý ảo.
- Xem chi tiết từng phòng ban: Khách có thể nhấn vào từng phòng ban của từng lĩnh vực để xem các chuyên gia.
- Đọc các bài báo, tin tức được công bố liên quan đến các lĩnh vực y tế.
- Tham khảo các dịch vụ hiện có của bệnh viện.
- Đăng ký tài khoản: Có thể đăng ký để trở thành người dùng chính thức và sử dụng thêm nhiều tính năng khác.
- Hạn chế: Không thể tương tác (bình luận, đánh giá, đặt lịch), không xem được lịch sử đặt lịch khám.

➤ Người dùng (User) - Người dùng đã đăng ký tài khoản

Người dùng là những người đã đăng ký và đăng nhập thành công vào hệ thống. Quyền hạn bao gồm:

- Tất cả chức năng của khách.
- Đặt lịch khám chữa bệnh: Người dùng có thể lựa chọn từng bác sĩ với các thành tựu trong từng lĩnh vực phù hợp với nhu cầu bản thân.
- Bình luận và đánh giá: Có thể để lại nhận xét hoặc đánh giá về từng bác sĩ.
- Gửi yêu cầu liên hệ với phía hỗ trợ khách hàng: Gửi thông tin yêu cầu đến phía hỗ trợ để được nhận tư vấn hoặc đặt lịch.
- Quản lý thông tin cá nhân: Cập nhật họ tên, số điện thoại, địa chỉ, ảnh đại diện,...
- Quản lý danh sách lịch sử các lần khám chữa bệnh: Xem lại các phòng đã được lưu yêu thích trong tài khoản.

### ➤ Bác sĩ (Doctor) - Người trực tiếp khám bệnh

Bác sĩ là người đăng nhập vào hệ thống với mục đích xem lịch biểu và quản lý cập nhật bệnh lý cho người bệnh đến khám. Họ có đầy đủ các quyền của khách, cộng thêm các quyền nâng cao liên quan đến quản lý lịch khám bệnh:

- Quản lý danh sách lịch khám bệnh nhân gửi đến: cập nhật bệnh lý vào hồ sơ bệnh án của bệnh nhân(user).
- Theo dõi các lịch khám được đặt online sắp tới

### ➤ Quản trị viên (Admin)

Quản trị viên là người điều hành toàn bộ hệ thống. Họ có toàn quyền trên website và có thể can thiệp vào mọi thành phần. Các quyền bao gồm:

- Quản lý toàn bộ người dùng: Xem danh sách, chỉnh sửa, khóa tài khoản hoặc phân quyền cho người dùng khác.
- Quản lý tất cả các bình luận / đánh giá: Xem hoặc xóa bình luận khi cần thiết.
- Quản lý các lịch đặt khám: cập nhật trạng thái, hủy bỏ lịch khám....
- Xem thống kê hệ thống: Theo dõi các báo cáo như số lượng người dùng, lượt đặt lịch, lượt bình luận, ...
- Cấu hình nội dung website: Chỉnh sửa các phần như logo, liên hệ, chính sách bảo mật, giới thiệu, ...
- Admin có quyền can thiệp toàn diện để đảm bảo hệ thống hoạt động ổn định, hiệu quả và đúng định hướng.

## 1.3. Xác định các yêu cầu của hệ thống

### ➤ Yêu cầu phi chức năng

STT	Yêu cầu phi chức năng	Mô tả
-----	-----------------------	-------



1	Dễ sử dụng	Giao diện thân thiện, dễ hiểu, hỗ trợ cả người mới sử dụng, tương thích trên máy tính và thiết bị di động
2	Khả năng tương thích	Website phải hoạt động tốt trên các nền tảng trình duyệt phổ biến như Chrome, Firefox, Edge, Safari, ...
3	Khả năng mở rộng	Có thể mở rộng để phục vụ nhiều người dùng đồng thời mà không ảnh hưởng đến hiệu năng.
4	Tính sẵn sàng	Hệ thống cần hoạt động liên tục 24/7, hạn chế tối đa thời gian bảo trì và ngắt kết nối.

### ➤ Yêu cầu chức năng

STT	Yêu cầu chức năng	Mô tả
1	Đăng ký tài khoản	Cho phép người dùng có thể đăng ký và tạo tài khoản bằng thông tin cá nhân (mật khẩu, số điện thoại...).
2	Đăng nhập / Đăng xuất	Người dùng có thể đăng nhập vào hệ thống bằng thông tin tài khoản đã đăng ký, và đăng xuất khi không sử dụng.

3	Bình luận và đánh giá	Người dùng có thể đánh giá từng bác sĩ
4	Quản lý thông tin cá nhân	Người dùng có thể quản lý, cập nhật các thông tin cá nhân như ảnh đại diện, họ và tên
5	Tìm kiếm các bác sĩ phù hợp	Người dùng có thể tìm kiếm theo tên hoặc phòng ban
6	Xem chi tiết bác sĩ	Người dùng có thể xem chi tiết các thông tin của bác sĩ như: kinh nghiệm làm việc, chuyên khoa, đánh giá của bệnh nhân khác
7	Đặt lịch khám	Người dùng có thể lựa chọn bác sĩ, thời gian, phòng ban để lựa chọn được bác sĩ phù hợp với bản thân
8	Cập nhật hồ sơ bệnh án (Bác sĩ)	Bác sĩ có thể cập nhật hồ sơ bệnh án của bệnh nhân đang khám
9	Quản lý người dùng (Admin)	Admin có thể xem xét danh sách, khoá tài khoản vi phạm hoặc thay đổi quyền hạn người dùng.
10	Quản lý nội dung trang (Admin)	Admin có thể toàn quyền chỉnh sửa các nội dung tĩnh của website như giới thiệu, chính sách, liên hệ, ...

11	Quản lý lịch đặt khám (Admin)	Admin có thể huỷ bỏ hoặc chấp thuận đã hoàn thành đặt lịch
12	Hỏi đáp trợ lý ảo	Người dùng có thể hỏi đáp với trợ lý ảo
13	Xem tin tức y khoa	Người dùng có thể xem tin tức các vấn đề về y khoa
14	Theo dõi lịch khám	Người dùng và bác sĩ có thể theo dõi thông tin các lịch khám sắp tới
15	Lấy lại mật khẩu đã mất	Cho phép người dùng lấy lại mật khẩu đã mất thông qua xác nhận email
16	Xuất file PDF kết quả khám (Admin)	Admin có thể xuất ra file kết quả khám sau khi lịch khám đã hoàn thành các thủ tục

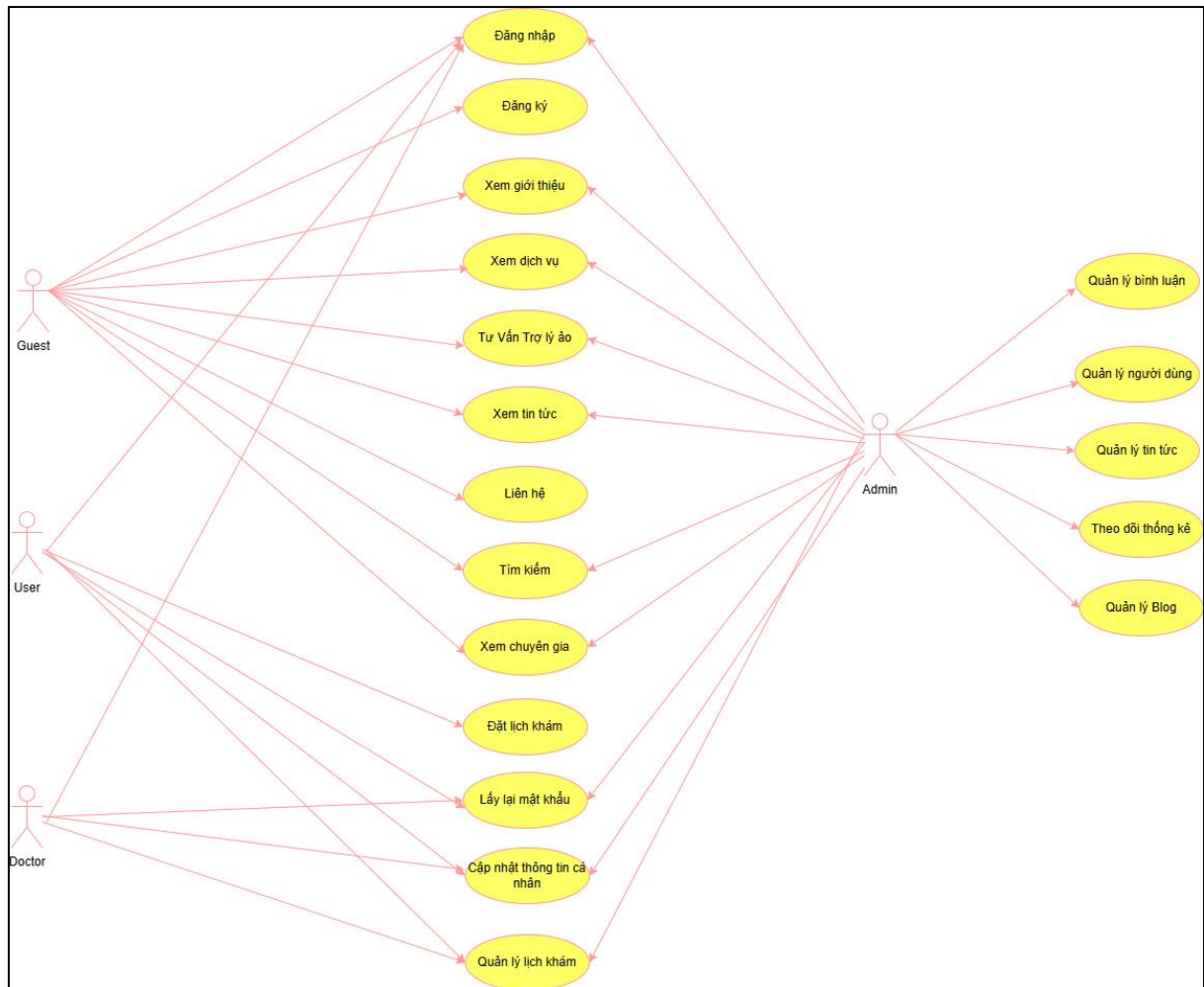
#### 1.4. Tổng kết chương 1

Chương khảo sát hệ thống đã giúp xác định rõ ràng bối cảnh, nhu cầu thực tế cũng như các vấn đề gặp phải trong quy trình đặt lịch khám hiện tại. Qua đó, đề tài được hình thành với mục tiêu phát triển một hệ thống đặt lịch khám và điều trị trực tuyến tích hợp Chatbot, nhằm giải quyết các hạn chế của các phương pháp truyền thống, tối ưu hóa quy trình làm việc và nâng cao trải nghiệm của người dùng. Những phân tích trong chương này là cơ sở để xây dựng các giải pháp kỹ thuật cụ thể được trình bày trong các chương tiếp theo của báo cáo.

## CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Biểu đồ Use case

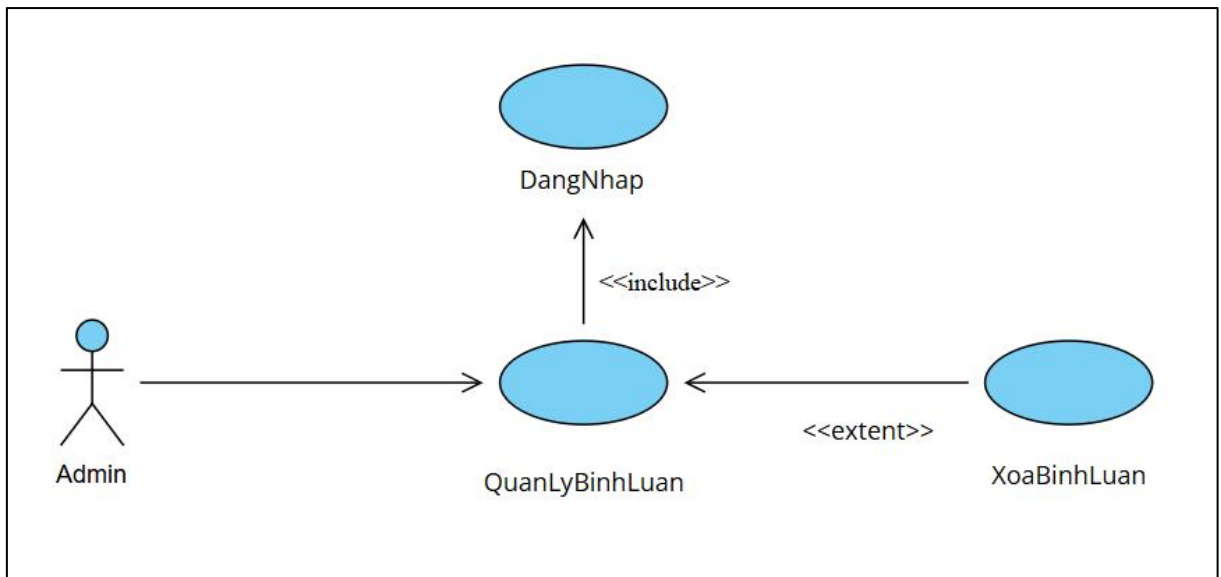
#### 2.1.1. Các Use case chính



Hình 2.1. Biểu đồ use case mức tổng quát

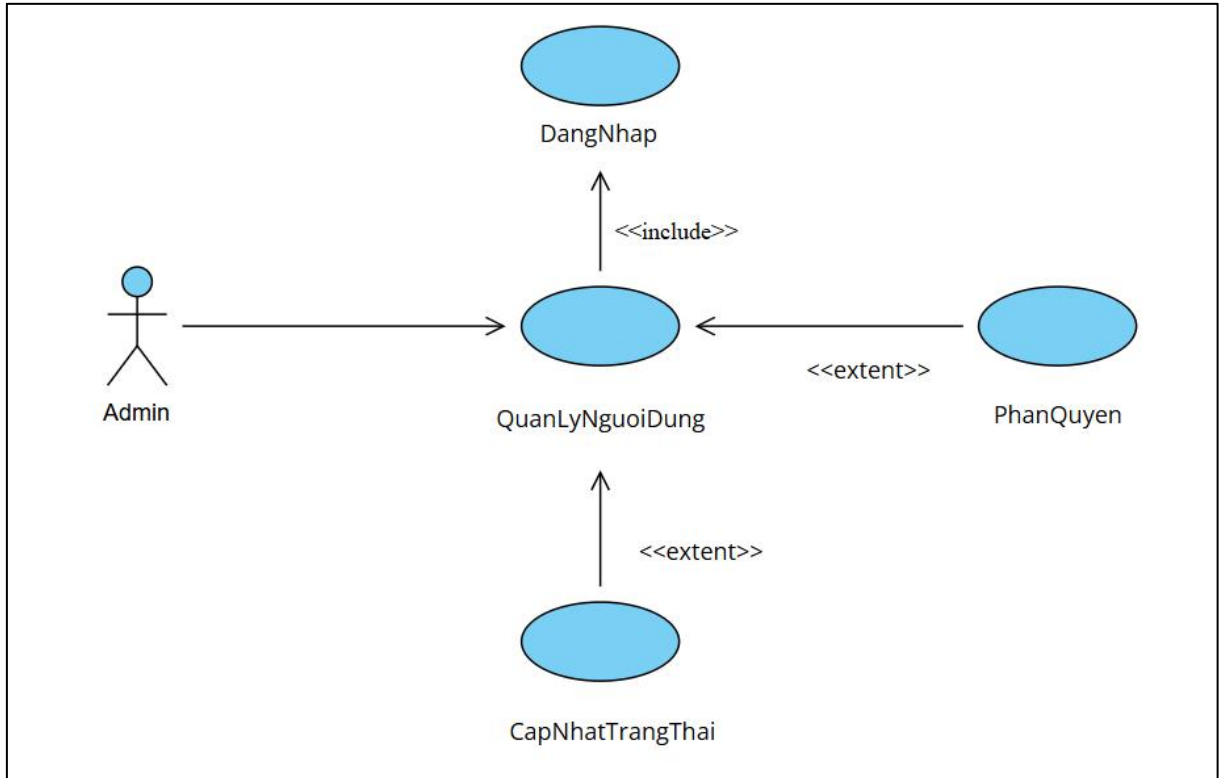
#### 2.1.2. Quan hệ giữa các Use case

## Quản lý bình luận



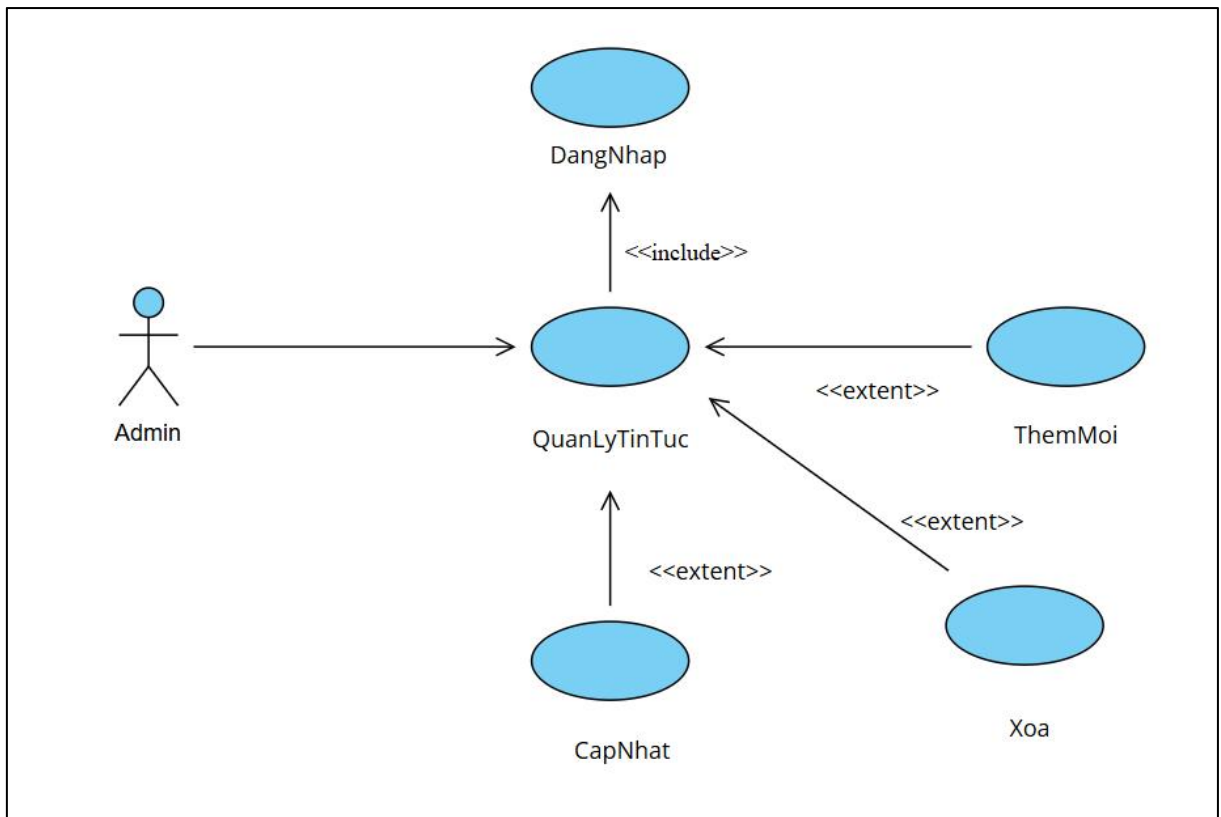
Hình 2.2. Quan hệ của use case quản lý bình luận

## Quản lý người dùng



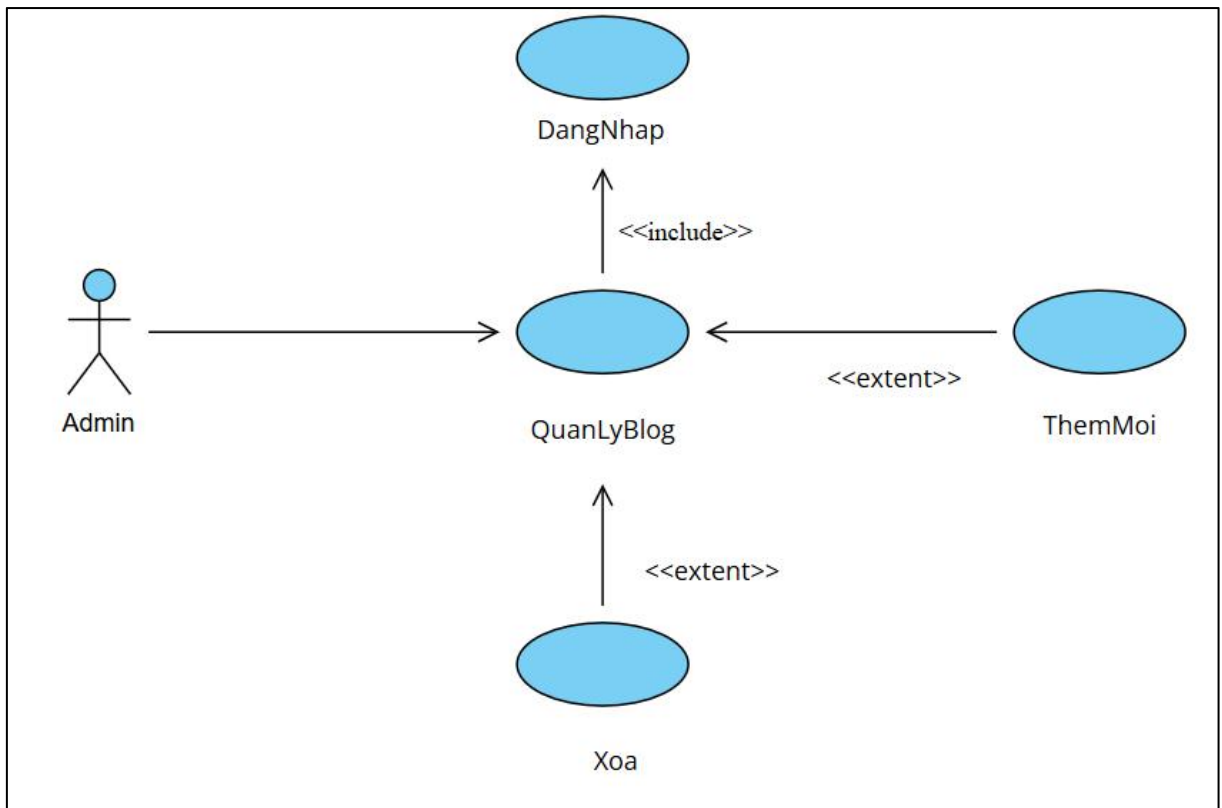
Hình 2.3. Quan hệ use case quản lý người dùng

## Quản lý tin tức



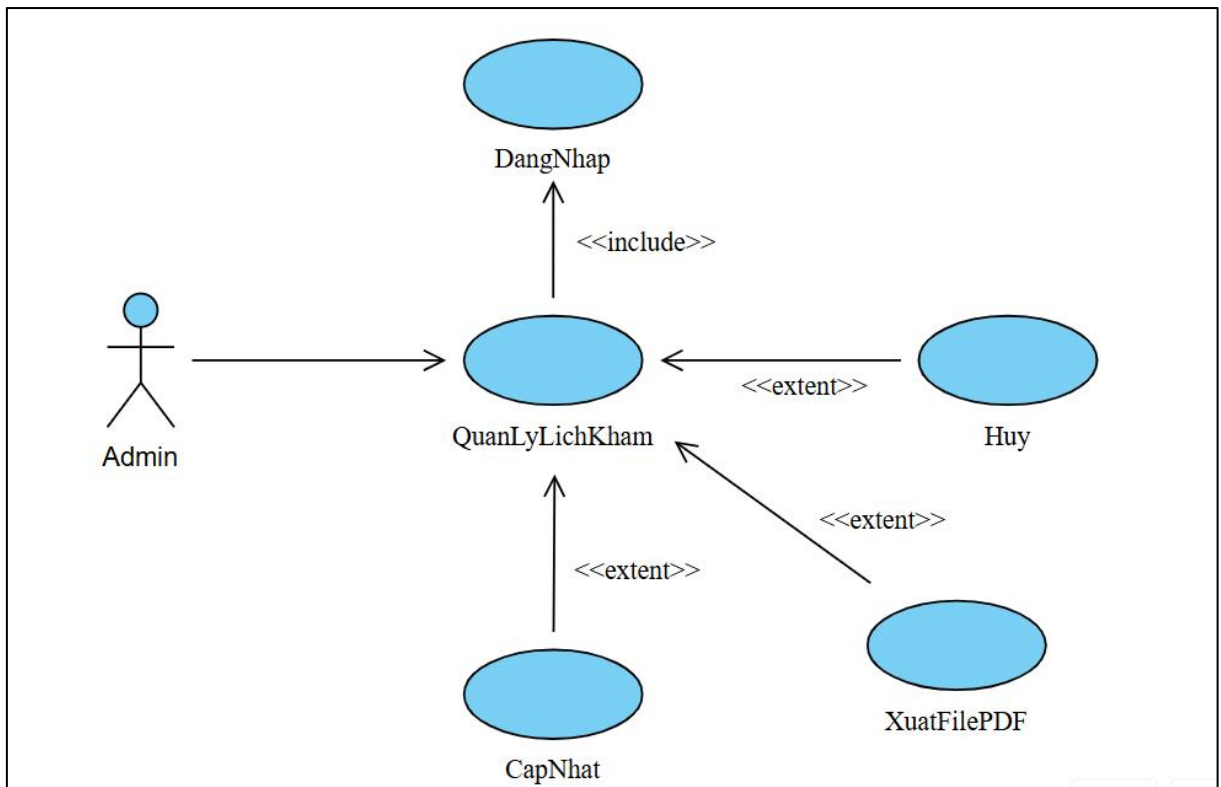
Hình 2.4. Quan hệ use case quản lý tin tức

## Quản lý Quảng cáo



Hình 2.5. Quan hệ use case quản lý Quảng cáo

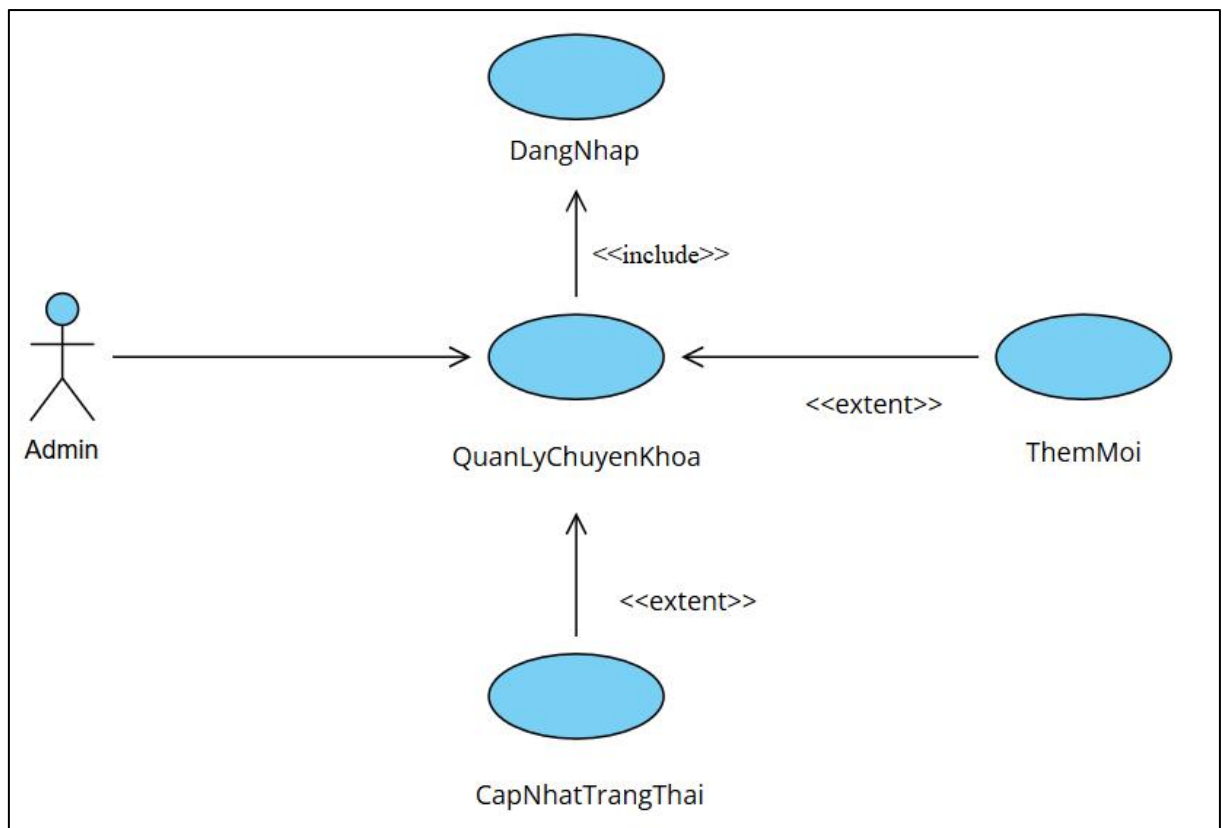
## Quản lý Lịch khám



Hình 2.6. Quan hệ use case quản lý lịch khám

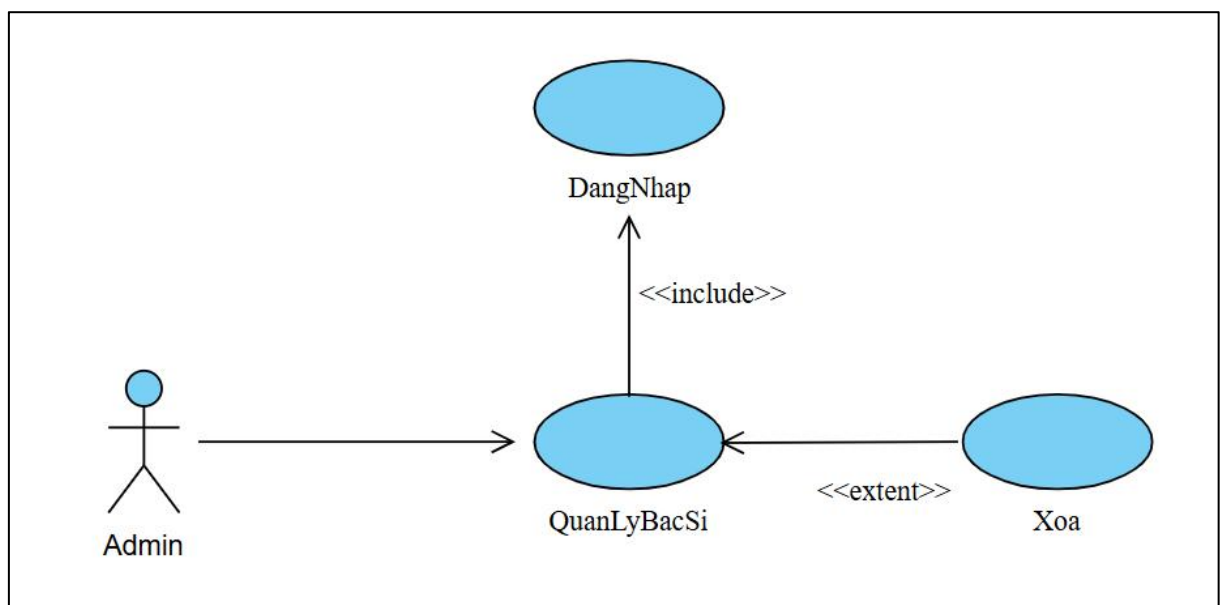


## Quản Lý Chuyên Khoa



Hình 2.7. Quan hệ use case quản lý chuyên khoa

## Quản Lý Bác Sĩ



Hình 2.8. Quan hệ use case quản lý bác sĩ

## 2.2. Mô tả chi tiết các Use case

### 2.2.1. Mô tả use case: Đăng nhập

1. **Tên use case:** Đăng nhập
2. **Mô tả vắn tắt:** Use case này cho phép người dùng đăng nhập vào hệ thống để thực hiện các chức năng.
3. **Luồng các sự kiện:**
  - a) **Luồng cơ bản:**
    - i. Use case này bắt đầu khi người dùng bắt đầu nhấn vào nút “Đăng nhập” trên màn hình. Hệ thống sẽ hiển thị thông tin đăng nhập gồm: tài khoản và mật khẩu.
    - ii. Người dùng sẽ ấn nút “Đăng nhập” và nhập các thông tin cơ bản. Hệ thống sẽ kiểm tra thông tin và cho khách hàng đăng nhập vào hệ thống. Use case kết thúc.
  - b) **Luồng rẽ nhánh:**
    - i. Khi người dùng nhập sai tài khoản hoặc nhập sai mật khẩu thì hệ thống sẽ thông báo thông tin đăng nhập không chính xác và yêu cầu người dùng nhập lại.
    - ii. Nếu tài khoản bị chặn thì khi đăng nhập hệ thống sẽ hiển thị thông báo lỗi rằng tài khoản đã bị cấm và use case kết thúc.
    - iii. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi và use case kết thúc.
4. **Các yêu cầu đặc biệt:** Không có.
5. **Tiền điều kiện:** Không có.
6. **Hậu điều kiện:** Người dùng đã đăng nhập thành công và có thể sử dụng chức năng tương ứng của hệ thống ứng với mỗi quyền tài khoản.
7. **Điểm mở rộng:** Không có.

### 2.2.2. Mô tả usecase: Đăng ký

1. **Tên usecase:** Đăng ký
2. **Mô tả vắn tắt:** Use case này cho phép người dùng tạo tài khoản mới để đăng nhập vào website
3. **Luồng các sự kiện:**
  - a) **Luồng cơ bản:**
    - i. Use case này bắt đầu khi người dùng bắt đầu nhấn vào nút “Đăng ký” trên màn hình. Hệ thống sẽ hiển thị thông tin đăng ký bao gồm: Tài khoản email, mật khẩu và mật khẩu nhập lại.
    - ii. Hệ thống sẽ gửi 1 mã xác nhận về email người dùng. Sau đó người dùng sẽ nhập mã lên để xác nhận đăng ký tài khoản.
    - iii. Khi người dùng sẽ ấn nút “Xác thực” và nhập các thông tin cơ bản. Hệ thống sẽ lưu thông tin người dùng vào bảng Users trong cơ sở dữ liệu quan hệ và sẽ có 1 thông báo đăng ký thành công hiện lên. Use case kết thúc.
  - b) **Luồng rẽ nhánh:**
    - i. Khi người dùng nhập sai định dạng email hoặc nhập khác giữa mật khẩu và mật khẩu nhập lại thì hệ thống sẽ thông báo thông tin đăng ký không chính xác và yêu cầu người dùng nhập lại.
    - ii. Khi người dùng đăng ký tài khoản đã tồn tại thì hệ thống sẽ hiện thông báo lỗi và yêu cầu người dùng chọn tài khoản khác.
    - iii. Nếu người dùng nhập sai mã xác nhận thì hệ thống sẽ thông báo người dùng nhập lại.
    - iv. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và Use case kết thúc.

4. **Các yêu cầu đặc biệt:** Không có.
5. **Tiền điều kiện:** Không có.
6. **Hậu điều kiện:** Khách hàng đăng ký tài khoản thành công.
7. **Điểm mở rộng:** Không có.

### 2.2.3. Mô tả use case: Trợ lý ảo

1. **Tên use case:** Trợ lý ảo
2. **Mô tả vắn tắt:** Use case này cho phép người dùng hỏi đáp thông tin với trợ lý ảo
3. **Luồng các sự kiện:**
  - a) Luồng cơ bản:
    - i. Use case này bắt đầu khi người dùng bắt đầu nhấn vào nút “Tư vấn” trên thanh công cụ. Hệ thống sẽ hiển thị giao diện trò chuyện.
    - ii. Khi người dùng sẽ ấn nút “Gửi” và nhập câu hỏi. Hệ thống sẽ phân tích câu hỏi và đưa ra câu trả lời phù hợp dựa trên bộ dữ liệu có sẵn. Use case kết thúc.
  - b) Luồng rẽ nhánh:
    - i. Nếu không kết nối được với phía backend thì hệ thống sẽ hiện thông báo có lỗi xảy ra. Use case kết thúc.
4. **Các yêu cầu đặc biệt:** Không có.
5. **Tiền điều kiện:** Không có.
6. **Hậu điều kiện:** Không có.
7. **Điểm mở rộng:** Không có.

#### 2.2.4. Mô tả Use case: Đặt lịch

1. **Tên use case:** Đặt lịch
2. **Mô tả vắn tắt:** Use case này cho phép người dùng lựa đặt lịch khám phù hợp với nhu cầu của bản thân.
3. **Luồng các sự kiện:**
  - a) **Luồng cơ bản:**
    - i. Use case này bắt đầu khi người dùng nhấn vào nút “Đặt lịch” trên giao diện web sau khi đã lựa chọn được bác sĩ phù hợp. Hệ thống sẽ hiển thị phiếu nhập thông tin gồm: Họ và tên, số điện thoại, địa chỉ, ảnh đại diện, ngày khám, giờ khám, triệu chứng.
    - ii. Khi người dùng sẽ ấn nút “Đặt lịch” và đã kiểm tra các thông tin cơ bản. Hệ thống sẽ khởi tạo 1 lịch hẹn và lưu vào cơ sở dữ liệu. Use case kết thúc.
  - b) **Luồng rẽ nhánh:**
    - i. Nếu người dùng chưa nhập đầy đủ thông tin cá nhân thì hệ thống sẽ yêu cầu cung cấp đầy đủ thông tin.
    - ii. Nếu hệ thống kiểm tra dưới cơ sở dữ liệu và thấy thời gian chọn đã đầy thì hệ thống sẽ thông báo người dùng chọn thời gian khác.
    - iii. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.
4. **Các yêu cầu đặc biệt:** Tài khoản không bị cấm.
5. **Tiền điều kiện:** Đã đăng nhập tài khoản.
6. **Hậu điều kiện:** Người dùng đặt lịch thành công.
7. **Điểm mở rộng:** Không có.

### 2.2.5. Mô tả use case: Cập nhật thông tin cá nhân

1. **Tên use case:** Cập nhật thông tin cá nhân
2. **Mô tả vắn tắt:** Use case này cho phép người dùng kiểm tra và cập nhật lại thông tin cá nhân.
3. **Luồng các sự kiện:**
  - a) **Luồng cơ bản:**
    - i. Use case này bắt đầu khi người dùng nhấn vào nút “Hồ sơ” trên thanh công cụ. Hệ thống sẽ hiển thị 1 form thông tin của người dùng gồm: Ảnh đại diện, họ và tên, số điện thoại, giới tính, ngày sinh, địa chỉ, mật khẩu mới.
    - ii. Khi người dùng sẽ ấn nút “Cập nhật” và hệ thống sẽ kiểm tra lại thông tin. Nếu hợp lệ thì hệ thống sẽ cập nhật lại thông tin người dùng vào bảng Users trong cơ sở dữ liệu và thông báo cho người dùng. Use case kết thúc.
  - b) **Luồng rẽ nhánh:**
    - i. Nếu người dùng nhập thông tin không hợp lệ thì hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.
    - ii. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.
4. **Các yêu cầu đặc biệt:** Tài khoản không bị cấm.
5. **Tiền điều kiện:** Đã đăng nhập tài khoản.
6. **Hậu điều kiện:** Người dùng cập nhật thành công tài khoản.
7. **Điểm mở rộng:** Không có.

### 2.2.6. Mô tả use case: Quản lý lịch khám

1. **Tên use case:** Quản lý lịch khám

**2. Mô tả vắn tắt:** Use case này cho phép người dùng theo dõi, cập nhật lại trạng thái của lịch khám.

**3. Luồng các sự kiện:**

a) Luồng cơ bản:

- i. Use case này bắt đầu khi người dùng nhấn vào “Quản lý” trên màn hình.
- ii. Hệ thống sẽ kiểm tra vai trò của người dùng và đưa ra các chức năng phù hợp:
  - Nếu người dùng có vai trò là user, doctor, admin thì sẽ có thể theo dõi trạng thái và xem chi tiết lịch khám. Use case kết thúc
  - Nếu người dùng có vai trò doctor hoặc Admin thì có thể cập nhật cho lịch khám như cập nhật trần đoán, cập nhật trạng thái lịch khám. Use case kết thúc.

b) Luồng rẽ nhánh:

- i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc

**4. Các yêu cầu đặc biệt:** Tài khoản không bị cấm.

**5. Tiền điều kiện:** Đã đăng nhập tài khoản.

**6. Hậu điều kiện:** Không có.

**7. Điểm mở rộng:** Không có.

**2.2.7. Mô tả use case: Tìm kiếm**

**1. Tên use case:** Tìm kiếm

**2. Mô tả vắn tắt:** Use case này cho phép người dùng tìm kiếm các chuyên gia y tế phù hợp theo nhu cầu cá nhân.

### 3. **Luồng các sự kiện:**

#### a) **Luồng cơ bản:**

- i. Use case này bắt đầu khi người dùng trở con chuột vào ô tìm kiếm trên giao diện website, người dùng sẽ nhập thông tin muốn tìm kiếm. Khi người dùng ấn nút tìm kiếm thì hệ thống sẽ lấy thông tin của các bác sĩ rồi hiển thị lên màn hình. Use case kết thúc.

#### b) **Luồng rẽ nhánh:**

- i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

### 4. **Các yêu cầu đặc biệt:** Không có.

### 5. **Tiền điều kiện:** Không có.

### 6. **Hậu điều kiện:** Không có.

### 7. **Điểm mở rộng:** Không có.

## 2.2.8. **Mô tả use case: Xem chuyên gia y tế**

### 1. **Tên use case:** Xem chuyên gia y tế

### 2. **Mô tả vắn tắt:** Use case này cho phép người dùng xem thông tin của các chuyên gia y tế.

### 3. **Luồng các sự kiện:**

#### a) **Luồng cơ bản:**

- i. Use case này bắt đầu khi người dùng ấn vào ảnh chuyên gia hoặc kích vào tên chuyên gia ở trang đội ngũ chuyên gia.
- ii. Hệ thống hiển thị thông tin chi tiết của chuyên gia. Use case kết thúc.

#### b) **Luồng rẽ nhánh:**



- i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

- 4. **Các yêu cầu đặc biệt:** Không có.
- 5. **Tiền điều kiện:** Không có.
- 6. **Hậu điều kiện:** Không có.
- 7. **Điểm mở rộng:** Không có.

#### 2.2.9. Mô tả use case: Lấy lại mật khẩu

- 1. **Tên use case:** Lấy lại mật khẩu
- 2. **Mô tả vắn tắt:** Use case này cho phép người dùng lấy lại mật khẩu của bản thân.
- 3. **Luồng các sự kiện:**

##### c) Luồng cơ bản:

- i. Use case này bắt đầu khi người dùng ấn vào nút “Quên mật khẩu”. Hệ thống sẽ hiển thị nhập email người dùng.
- ii. Người dùng nhập email xong. Hệ thống sẽ gửi 1 mã xác nhận đến email của người dùng. Hệ thống hiển thị form nhập mã xác thực.
- iii. Người dùng nhập mã xác thực. Hệ thống sẽ hiển thị form đổi mật khẩu. Người dùng thực hiện mật khẩu mới, nhập lại mật khẩu và ấn đổi mật khẩu. Hệ thống sẽ hiển thị thông báo đổi mật khẩu thành công. Use case kết thúc

##### d) Luồng rẽ nhánh:

- i. Nếu người dùng nhập sai mã xác nhận hoặc mật khẩu nhập lại khác mật khẩu mới thì hệ thống sẽ hiển thị thông báo yêu cầu người dùng kiểm tra lại.

- ii. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

4. **Các yêu cầu đặc biệt:** Không có.
5. **Tiền điều kiện:** Không có.
6. **Hậu điều kiện:** Người dùng lấy lại mật khẩu thành công.
7. **Điểm mở rộng:** Không có.

#### 2.2.10. Mô tả use case: Quản lý người dùng

1. **Tên use case:** Quản lý người dùng
2. **Mô tả vắn tắt:** Use case này cho phép admin quản lý danh sách người dùng trên hệ thống.
3. **Luồng các sự kiện:**

a) Luồng cơ bản:

- i. Use case này bắt đầu khi Admin kích vào nút “Admin Dashboard” trên thanh công cụ và kích vào quản lý người dùng. Hệ thống sẽ hiển thị danh sách người dùng lên giao diện từ bảng user.
- ii. Thêm Bác sĩ: use case này bắt đầu khi Admin kích nút “add Doctor” bên phía người dùng. Hệ thống sẽ cập nhật người dùng này lên thành Bác sĩ và lưu vào cơ sở dữ liệu cùng 1 thông báo thành công. Use case kết thúc.
- iii. Ẩn Bác sĩ: use case này bắt đầu khi Admin kích nút “Ẩn” phía Bác sĩ. Hệ thống sẽ chuyển quyền người này về User. Hệ thống sẽ cập nhật lại quyền người dùng xuống cơ sở dữ liệu và hiện thông báo thành công. Use case kết thúc.

iv. **Cấm người dùng:** use case này bắt đầu khi Admin kích nút “Deactive” phía người dùng. Hệ thống sẽ cập nhật lại trạng thái người dùng xuống cơ sở dữ liệu và hiện thông báo thành công. Use case kết thúc.

b) **Luồng rẽ nhánh:**

i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

4. **Các yêu cầu đặc biệt:** Không có.

5. **Tiền điều kiện:** Đăng nhập với quyền admin.

6. **Hậu điều kiện:** Admin cập nhật thành công.

7. **Điểm mở rộng:** Không có.

#### 2.2.11. Mô tả use case: Quản lý tin tức

1. **Tên use case:** Quản lý tin tức

2. **Mô tả vắn tắt:** Use case này cho phép Admin quản lý danh sách tin tức trên hệ thống.

3. **Luồng các sự kiện:**

a) **Luồng cơ bản:**

i. Use case này bắt đầu khi Admin kích vào nút “Admin Dashboard” trên thanh công cụ và kích vào quản lý tin tức. Hệ thống sẽ hiển thị danh sách tin tức lên giao diện từ bảng article.

ii. Thêm bài báo: use case này bắt đầu khi Admin kích nút “Đăng bài” và đã điền form khởi tạo. Hệ thống sẽ tạo 1 bài báo mới và lưu vào cơ sở dữ liệu cùng 1 thông báo thành công. Use case kết thúc.

iii. Xóa bài báo: use case này bắt đầu khi Admin kích nút “Xóa”. Hệ thống sẽ xóa bài báo với id liên quan. Hệ thống sẽ cập nhật lại quyền người dùng xuống cơ sở dữ liệu và hiện thông báo thành công. Use case kết thúc.

iv. Cập nhật: use case này bắt đầu khi admin kích nút “Cập nhật”. Hệ thống sẽ lấy và hiện lên thông tin nội dung của bài báo, sau khi cập nhật lại nội dung thì sẽ được lưu xuống cơ sở dữ liệu và hiện thông báo thành công. Use case kết thúc.

b) Luồng rẽ nhánh:

i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

4. **Các yêu cầu đặc biệt:** Không có.

5. **Tiền điều kiện:** Đăng nhập với quyền Admin.

6. **Hậu điều kiện:** Admin cập nhật thành công.

7. **Điểm mở rộng:** Không có.

#### 2.2.12. Mô tả use case: Quản lý bình luận

1. **Tên use case:** Quản lý bình luận

2. **Mô tả vắn tắt:** Use case này cho phép admin quản lý bình luận(theo dõi, xóa) của người dùng trên hệ thống.

3. **Luồng các sự kiện:**

a) Luồng cơ bản:

i. Use case này bắt đầu khi người dùng đăng nhập vào hệ thống với vai trò “Admin”.

ii. Xóa bình luận: use case này bắt đầu khi Admin xem các bình luận và kích nút “Xóa”. Hệ thống sẽ xóa bỏ bình luận này trong cơ sở dữ liệu và hiện thông báo thành công. Use case kết thúc.

iii. Theo dõi thống kê: use case này bắt đầu khi Admin kích nút “Trang Admin”. Hệ thống sẽ hiện thống kê số lượng bình luận trong 6 tháng gần đây. Use case kết thúc.

b) Luồng rẽ nhánh:

i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

4. **Các yêu cầu đặc biệt:** Không có.

5. **Tiền điều kiện:** Đăng nhập với quyền admin.

6. **Hậu điều kiện:** Admin cập nhật thành công.

7. **Điểm mở rộng:** Không có.

### 2.2.13. Mô tả use case: Quản lý trang chủ

1. **Tên use case:** Quản lý trang chủ

2. **Mô tả vắn tắt:** Use case này cho phép Admin quản lý trang chủ của website trên hệ thống.

3. **Luồng các sự kiện:**

a) Luồng cơ bản:

i. Use case này bắt đầu khi Admin kích vào nút “Admin Dashboard” trên thanh công cụ và kích vào Trang Admin. Hệ thống sẽ hiển thị danh sách các quảng cáo trên trang chủ.

ii. Xóa quảng cáo: use case này bắt đầu khi Admin kích nút “Xóa”. Hệ thống sẽ xóa bỏ bình luận này trong cơ sở dữ liệu và hiện thông báo thành công. Use case kết thúc.

iii. Thêm quảng cáo: use case này bắt đầu khi Admin kích nút “Thêm”. Hệ thống sẽ hiện ra 1 form để chọn ảnh, sau khi ấn “Thêm” thì hệ thống sẽ lưu ảnh và hiện thông báo thành công. Use case kết thúc.

b) Luồng rẽ nhánh:

i. Nếu hệ thống không kết nối được với cơ sở dữ liệu thì sẽ có 1 thông báo lỗi cơ sở dữ liệu và use case kết thúc.

4. **Các yêu cầu đặc biệt:** Không có.

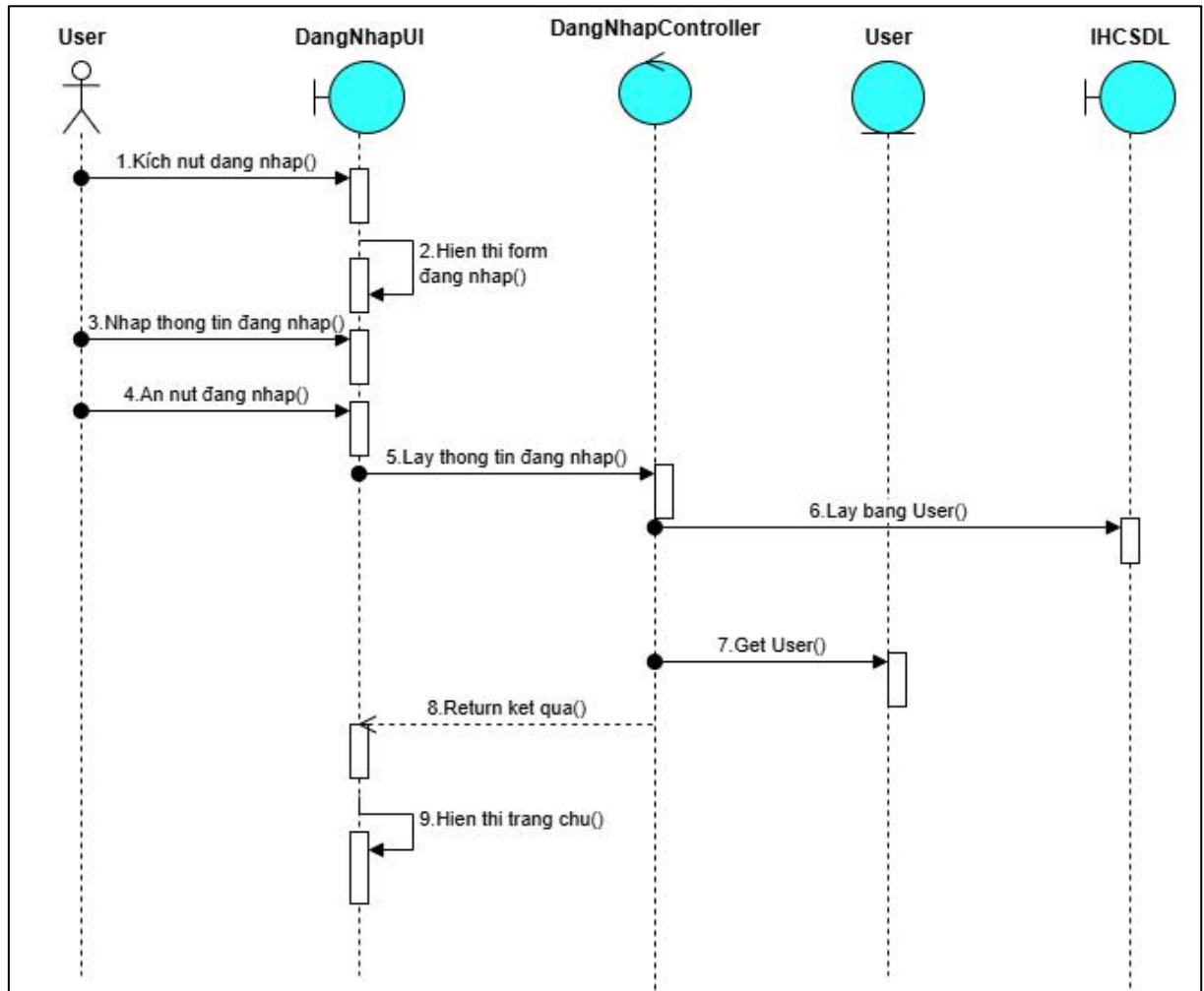
5. **Tiền điều kiện:** Đăng nhập với quyền Admin.

6. **Hậu điều kiện:** Admin cập nhật thành công.

7. **Điểm mở rộng:** Không có.

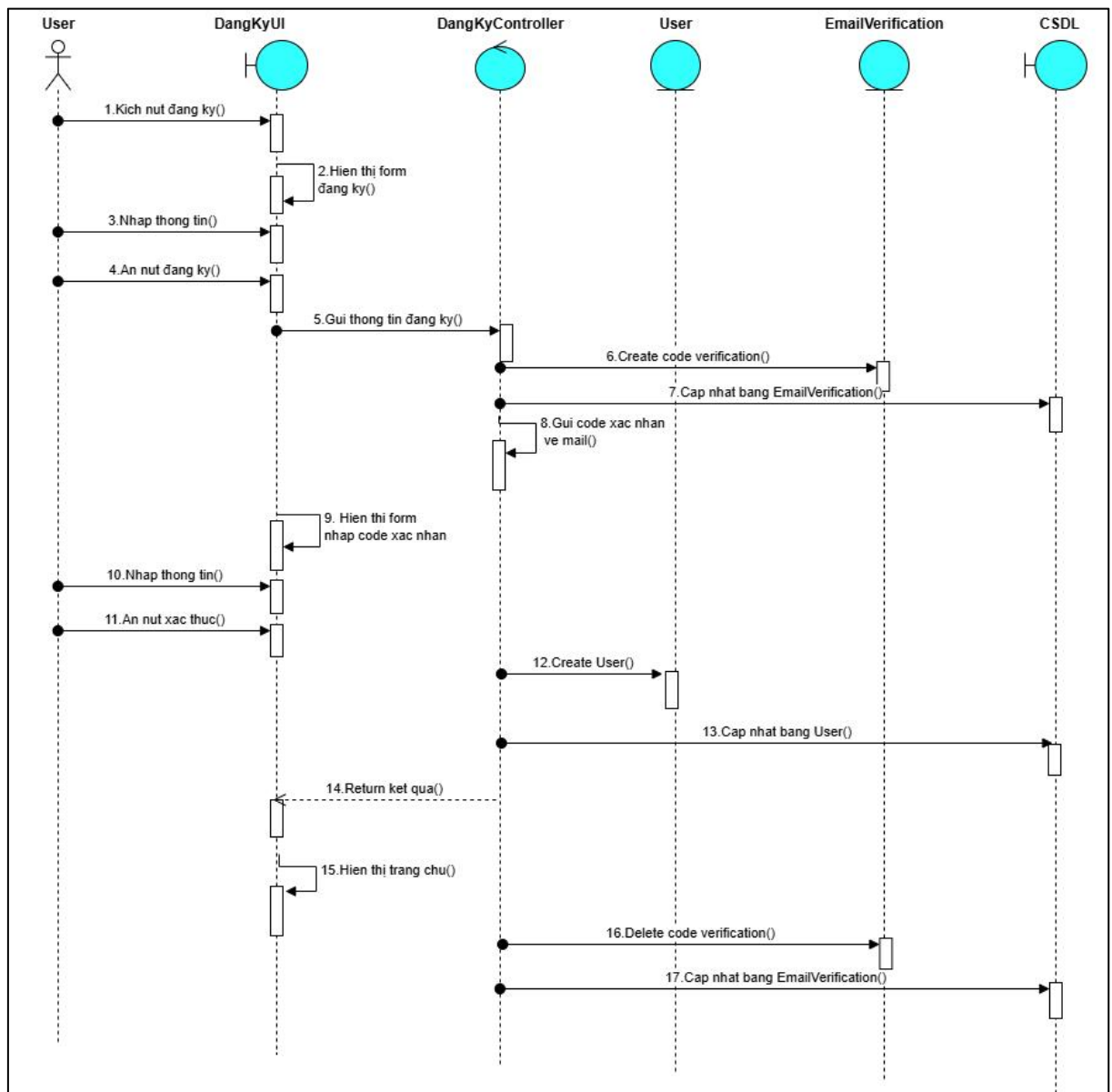
## 2.3. Các biểu đồ trình tự Use case

### 2.3.1. Biểu đồ đăng nhập



Hình 2.9. Biểu đồ trình tự use case đăng nhập

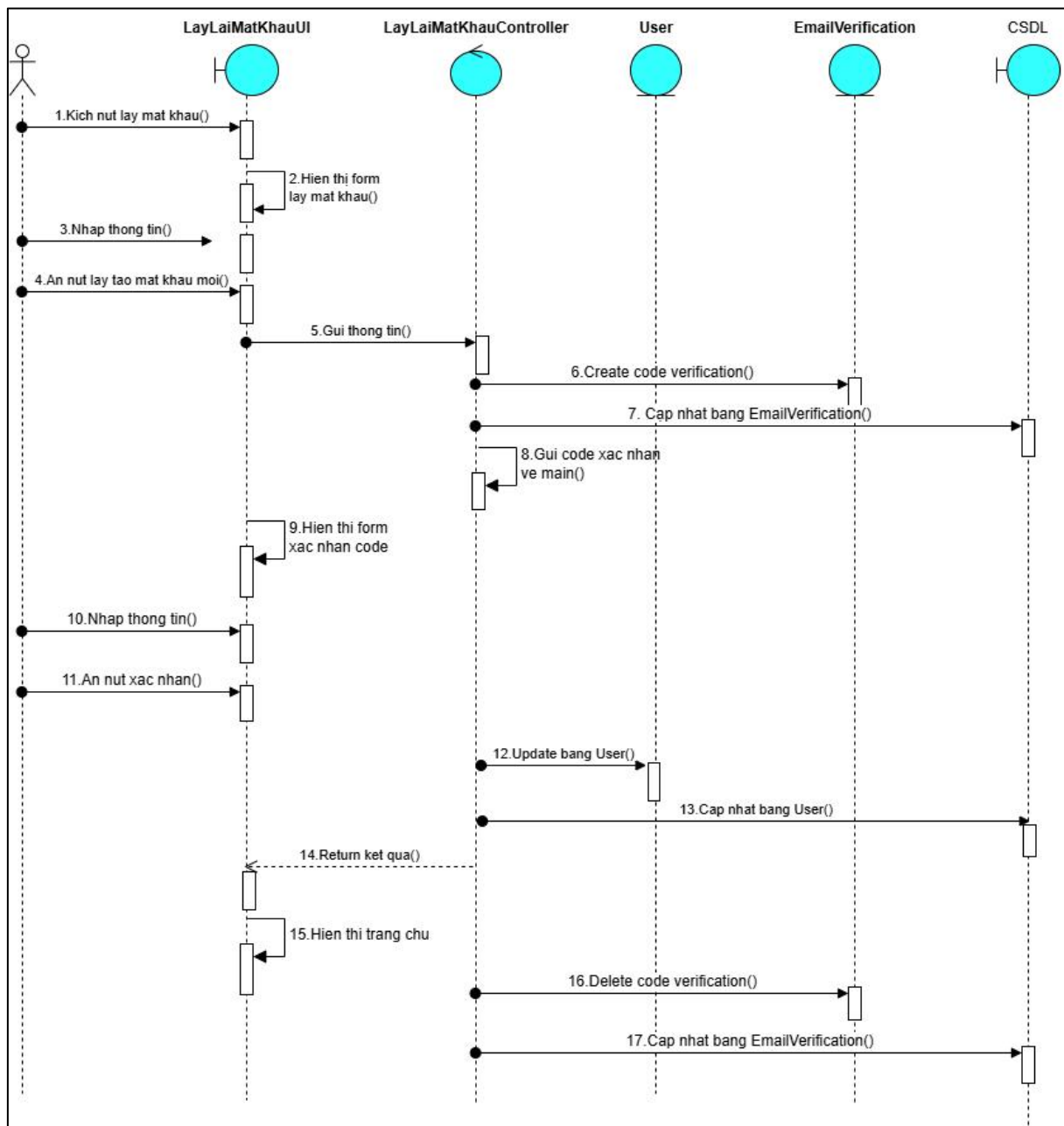
### 2.3.2. Biểu đồ đăng ký



Hình 2.10. Biểu đồ trình tự use case đăng ký

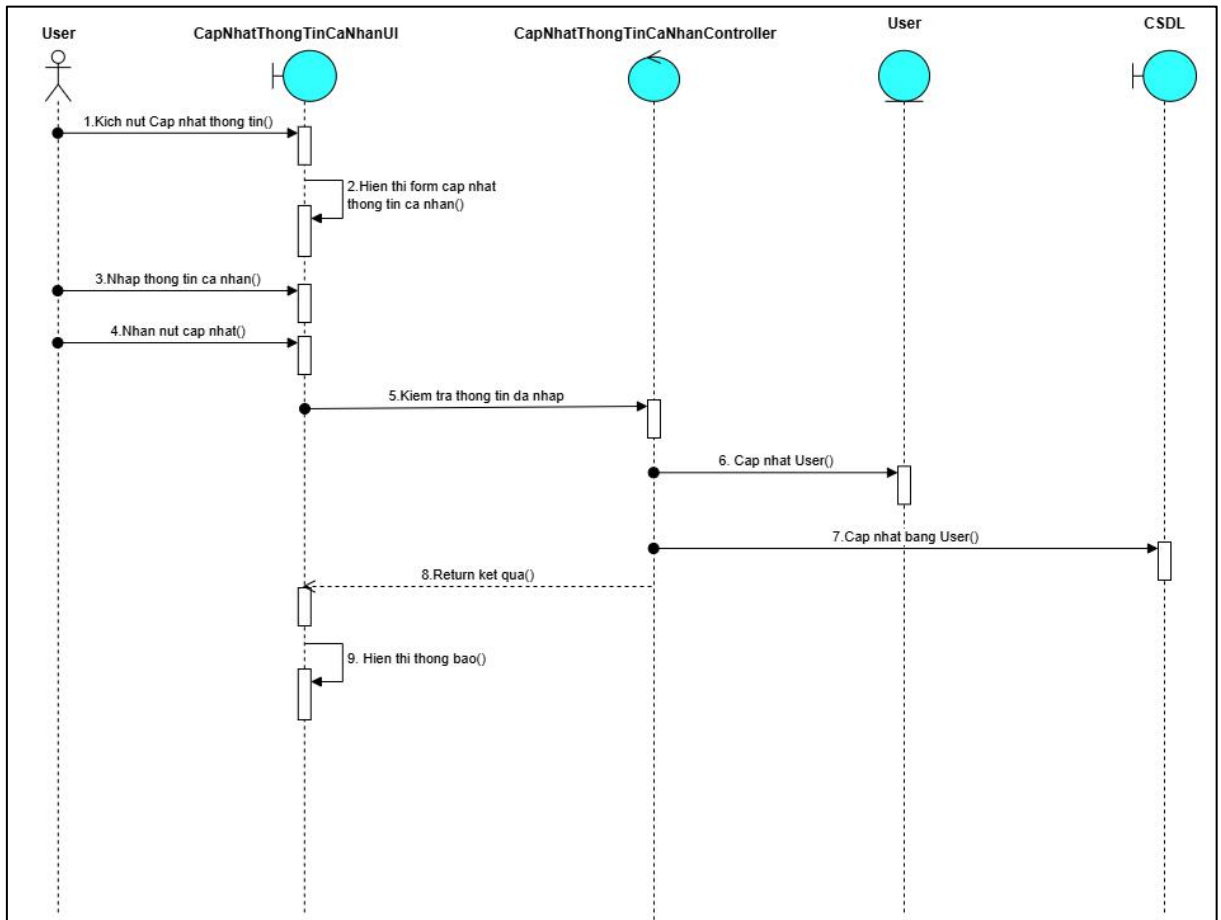


### 2.3.3. Biểu đồ lấy lại mật khẩu



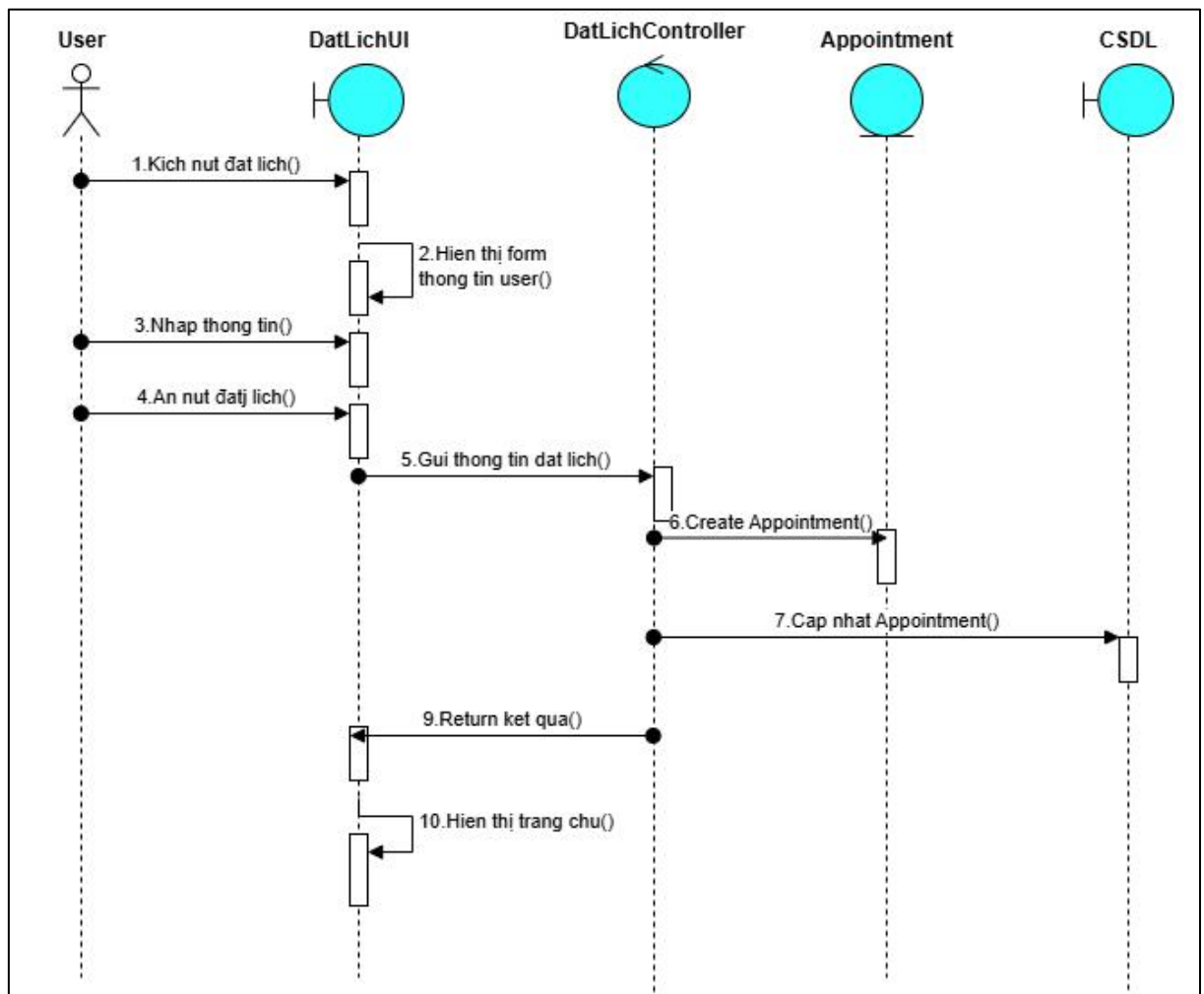
Hình 2.11. Biểu đồ trình tự use case lấy lại mật khẩu

### 2.3.4. Biểu đồ cập nhật thông tin cá nhân



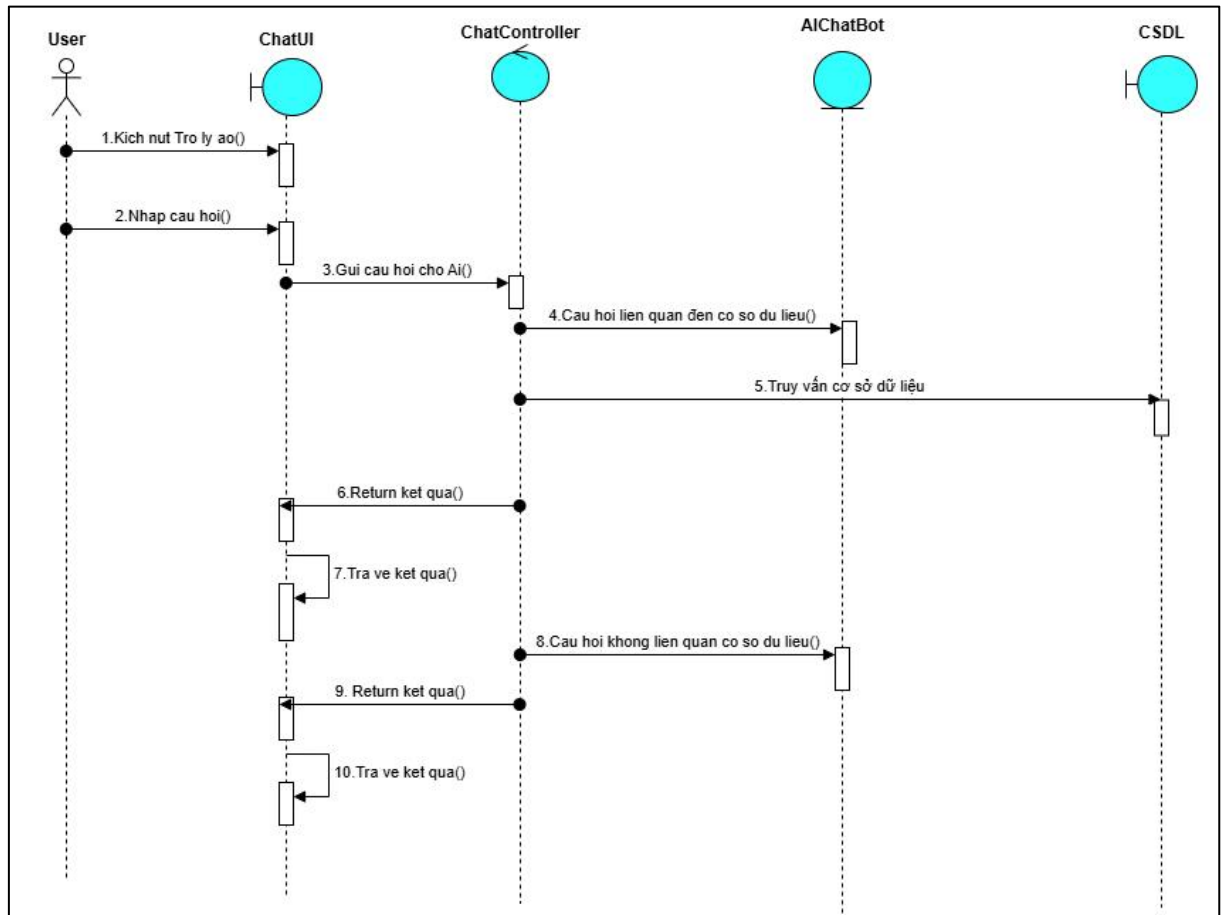
Hình 2.12. Biểu đồ trình tự use case cập nhật thông tin cá nhân

### 2.3.5. Biểu đồ đặt lịch khám



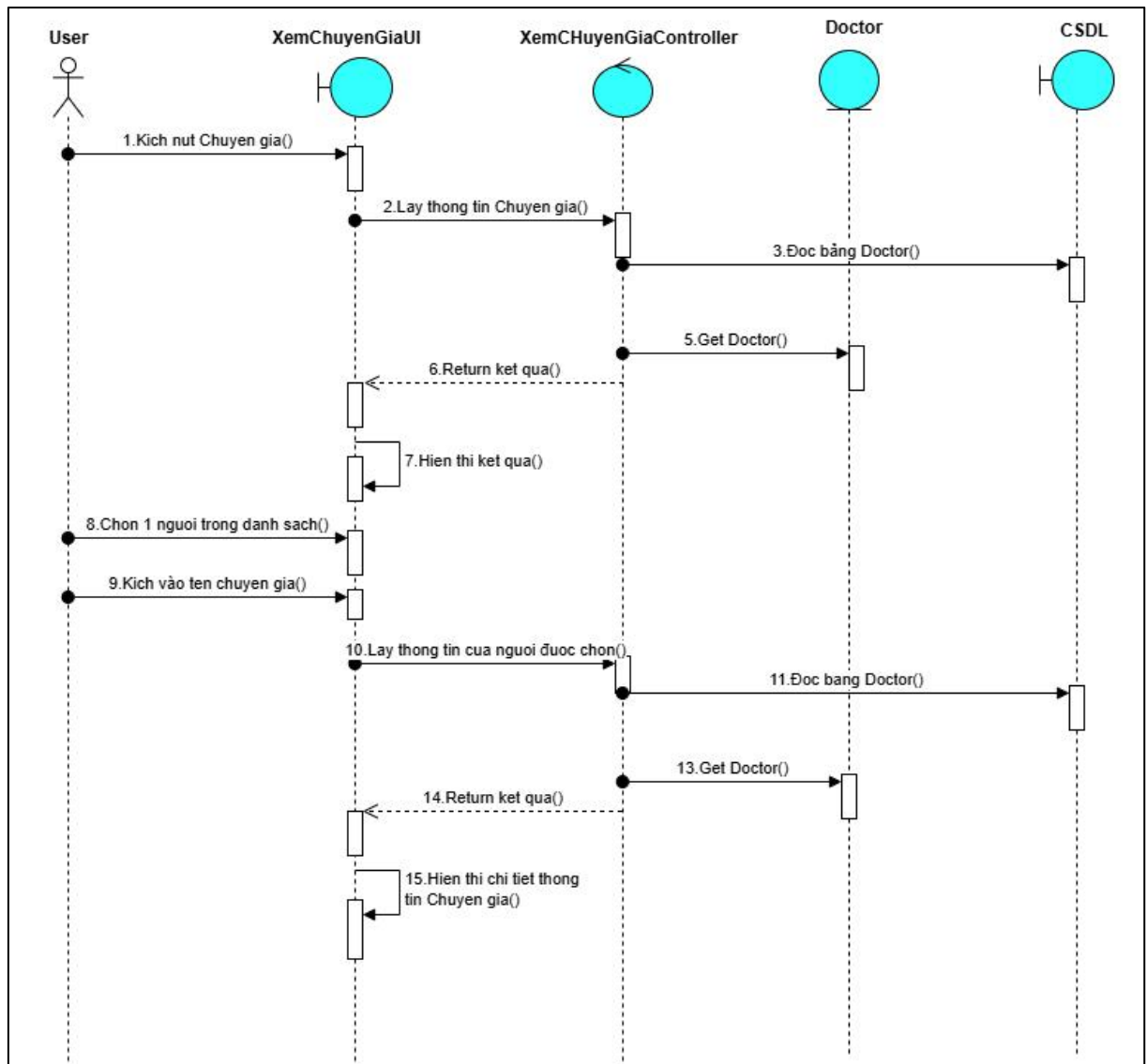
Hình 2.13. Biểu đồ trình tự use case đặt lịch khám

### 2.3.6. Biểu đồ Tư vấn trợ lý ảo



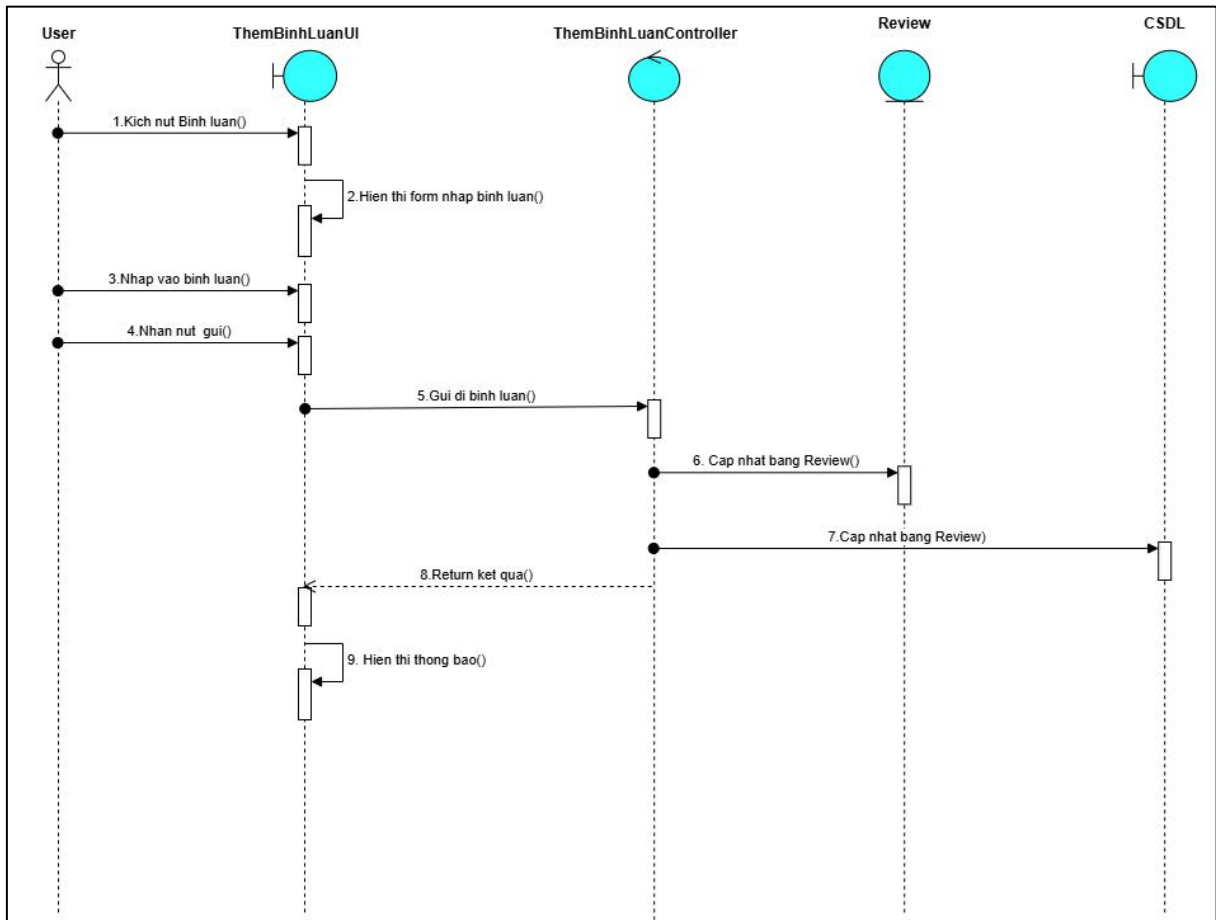
Hình 2.14. Biểu đồ trình tự use case tư vấn trợ lý ảo

### 2.3.7. Biểu đồ tìm kiếm chuyên gia theo phòng ban



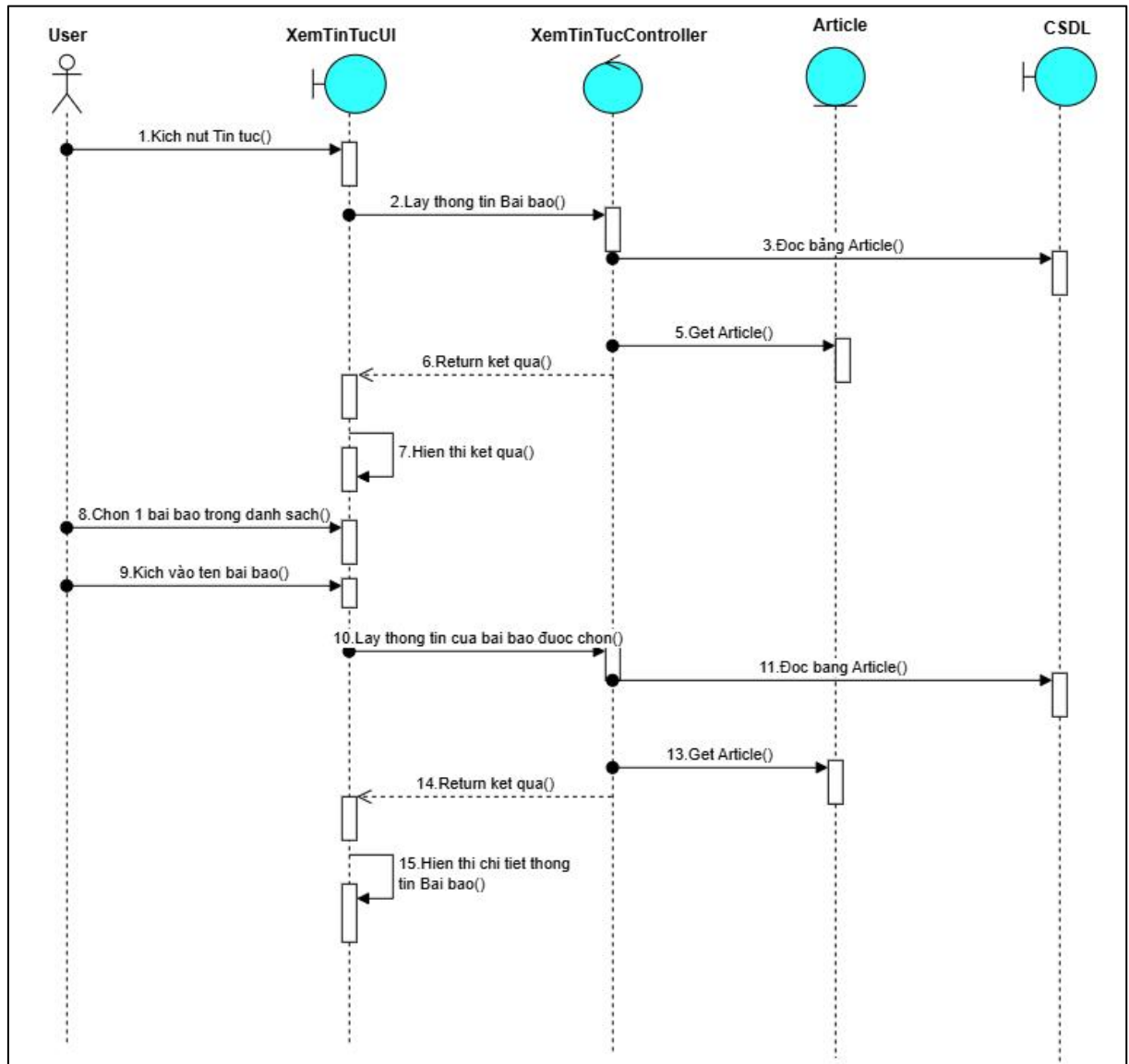
Hình 2.15. Biểu đồ trình tự use case tìm kiếm chuyên gia theo phòng ban

### 2.3.8. Biểu đồ thêm bình luận



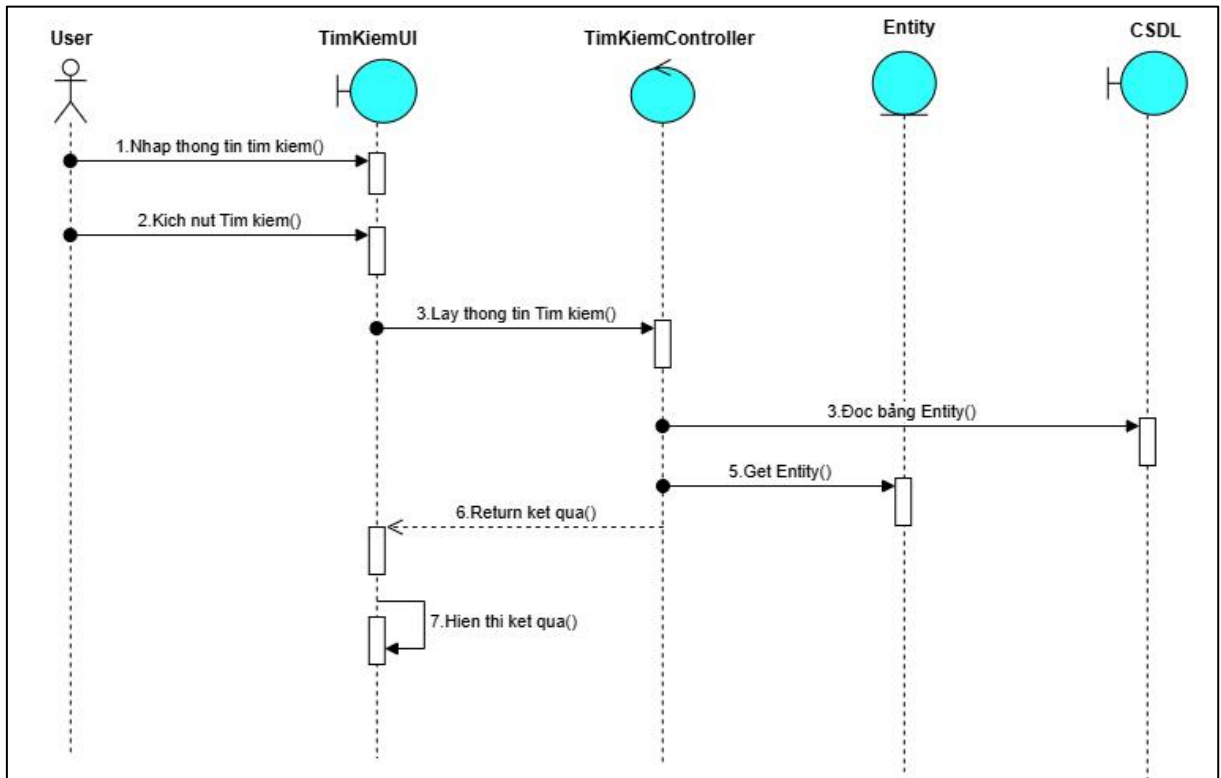
Hình 2.16. Biểu đồ trình tự use case thêm bình luận

### 2.3.9. Biểu đồ xem tin tức



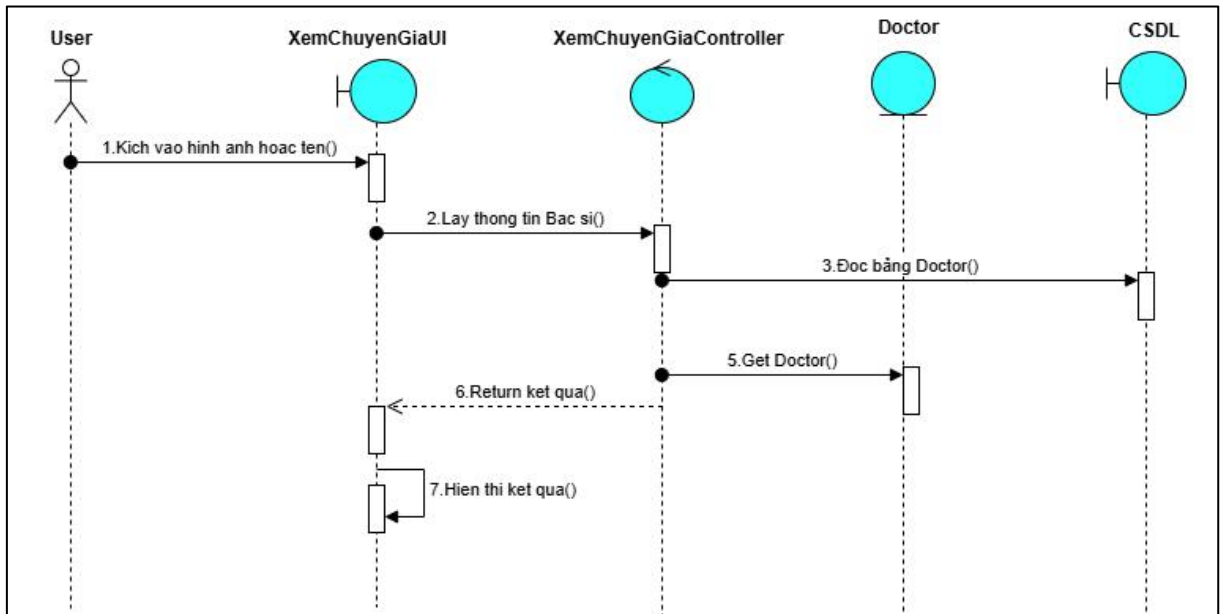
Hình 2.17. Biểu đồ trình tự use case xem tin tức

### 2.3.10. Biểu đồ tìm kiếm



Hình 2.18. Biểu đồ trình tự use case tìm kiếm

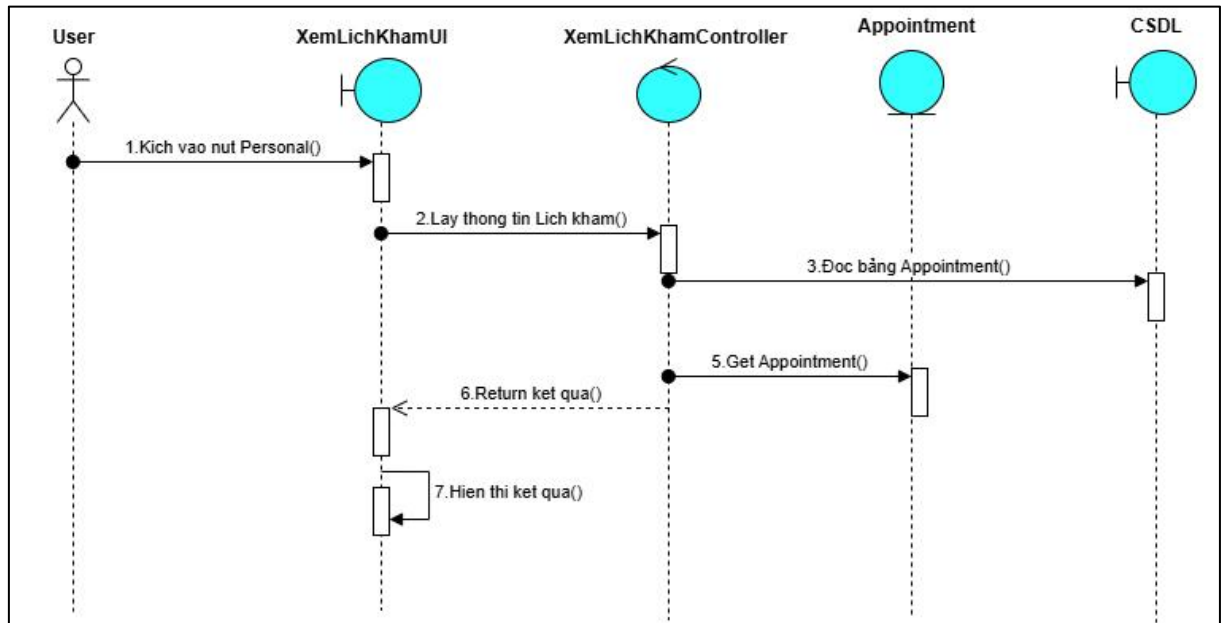
### 2.3.11. Biểu đồ xem chuyên gia



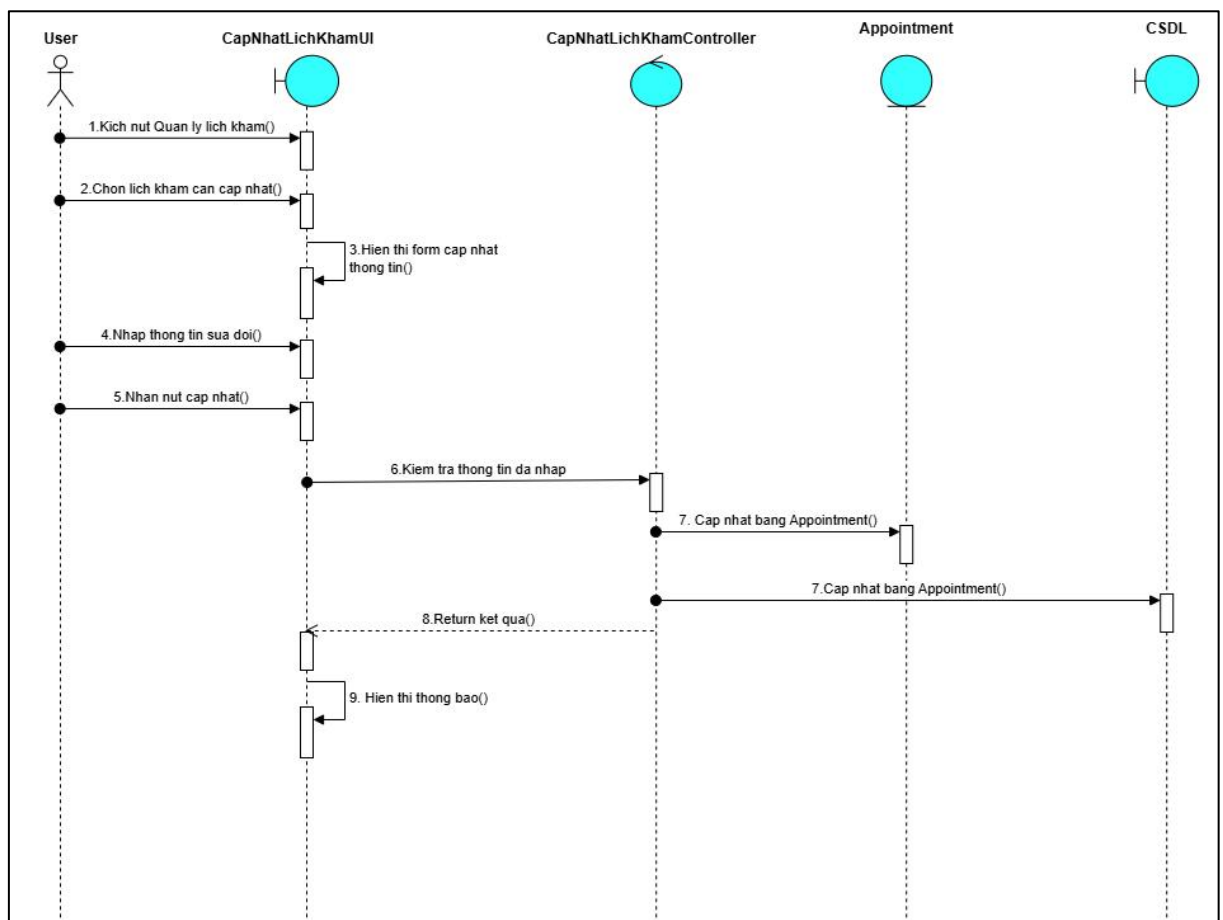
Hình 2.19. Biểu đồ trình tự use case xem chuyên gia



### 2.3.12. Biểu đồ quản lý lịch khám

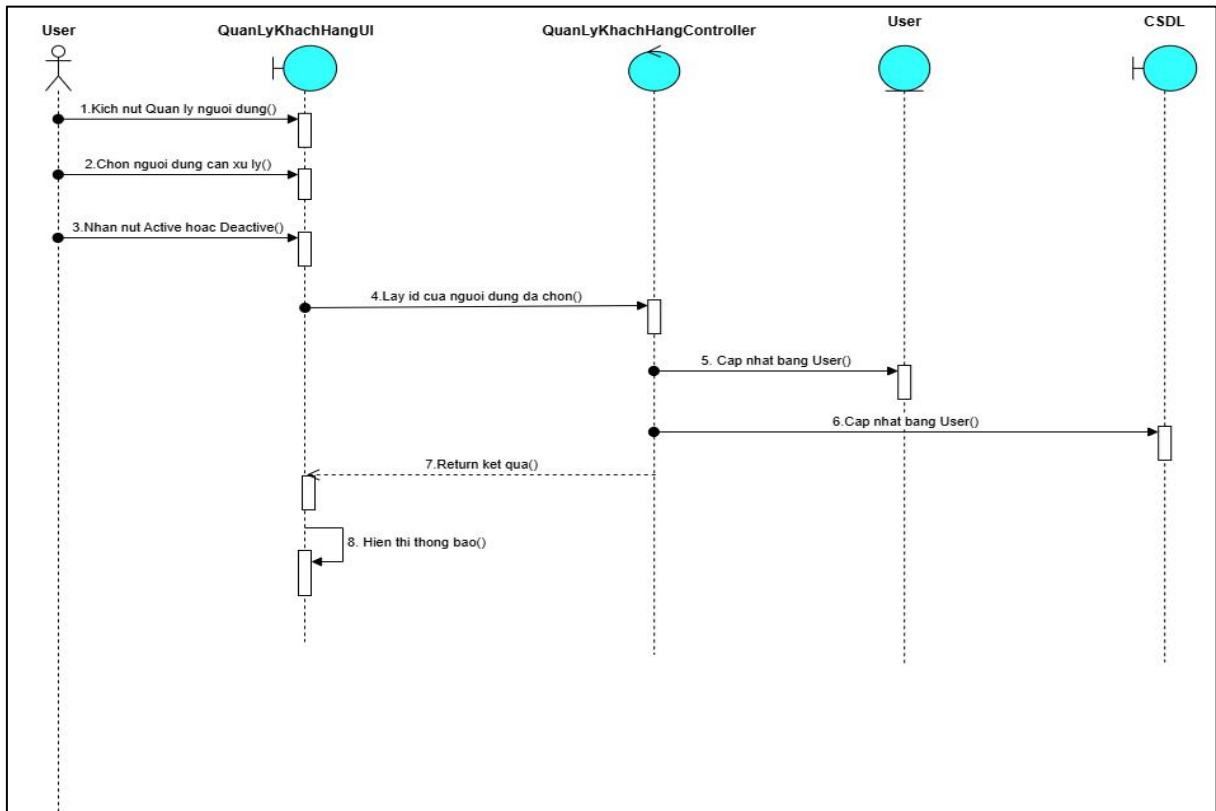


Hình 2.20. Biểu đồ trình tự use case xem lịch sử khám

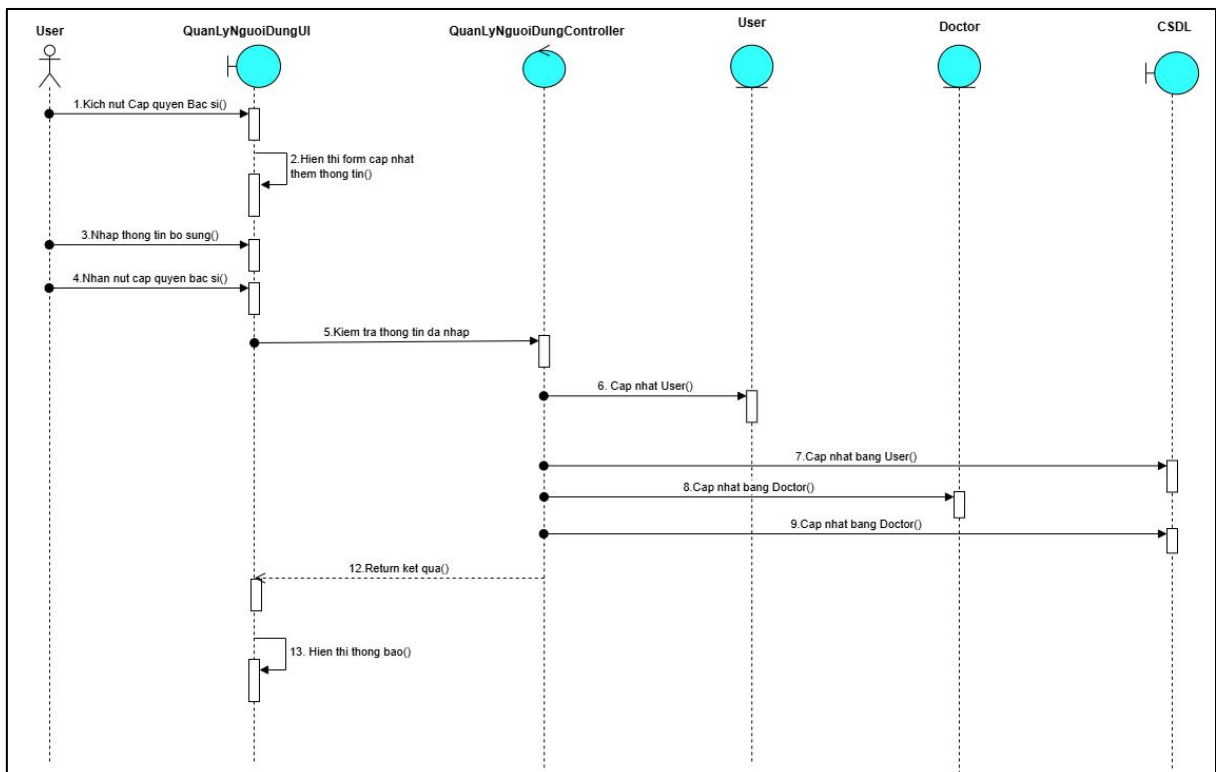


Hình 2.21. Biểu đồ trình tự use case cập nhật lịch khám

### 2.3.13. Biểu đồ quản lý tài khoản người dùng

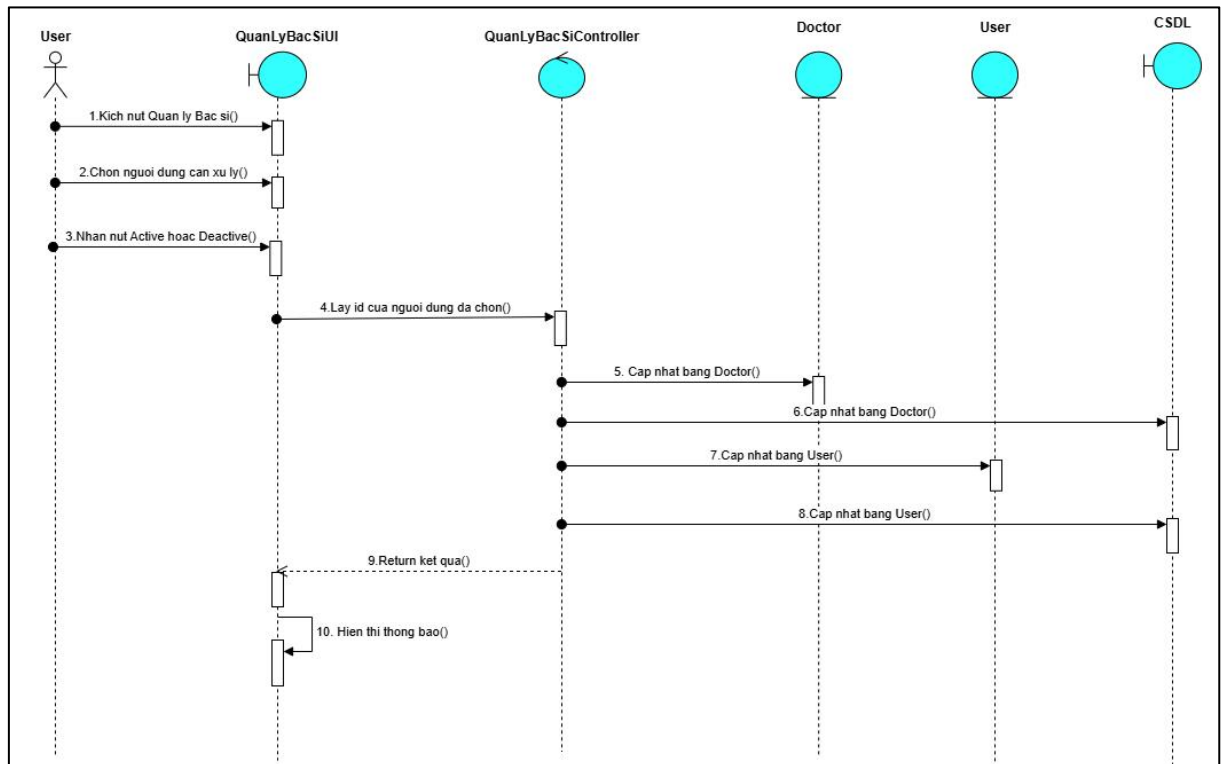


Hình 2.22. Biểu đồ trình tự use case kích hoạt hoặc cấm khách hàng



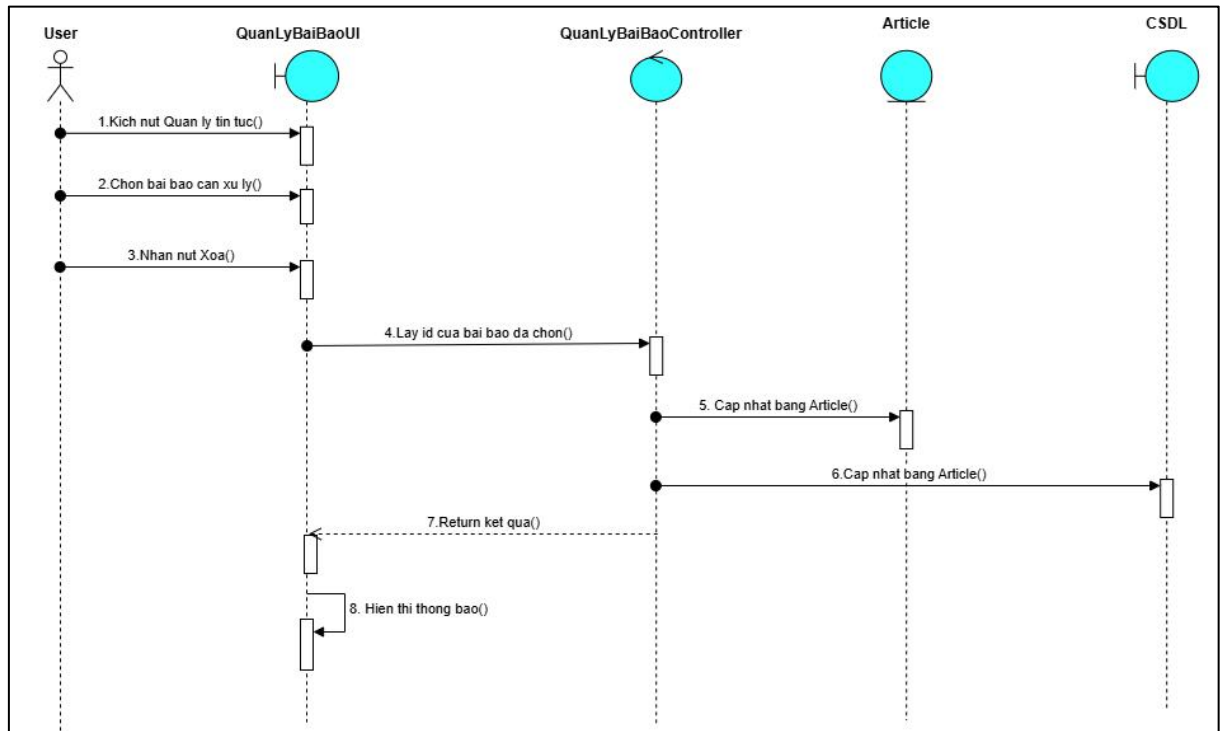
Hình 2.23. Biểu đồ trình tự use case cấp quyền Bác sĩ

### 2.3.14. Biểu đồ quản lý tài khoản bác sĩ

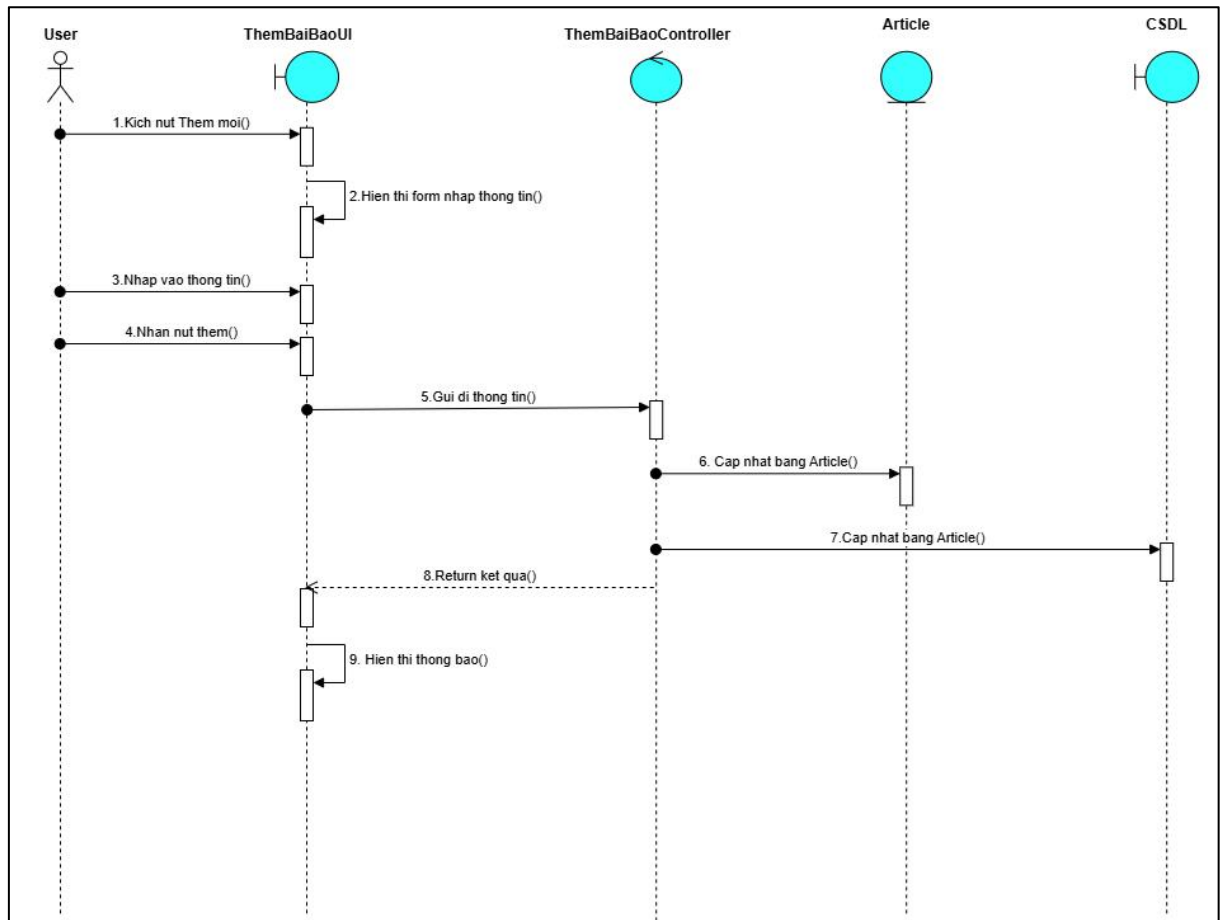


Hình 2.24. Biểu đồ trình tự use case cấm tài khoản bác sĩ

### 2.3.15. Biểu đồ quản lý tin tức

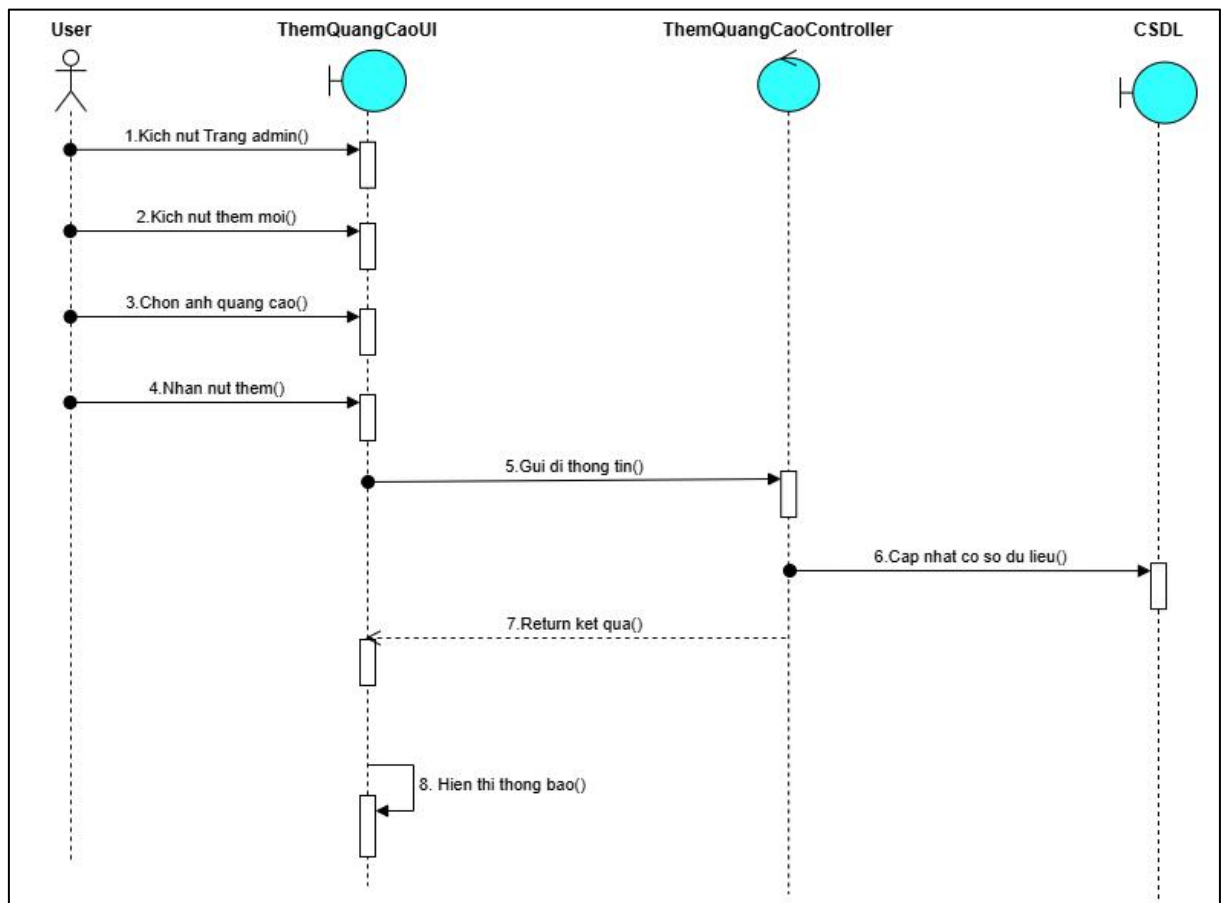


Hình 2.25. Biểu đồ trình tự use case xóa bài báo

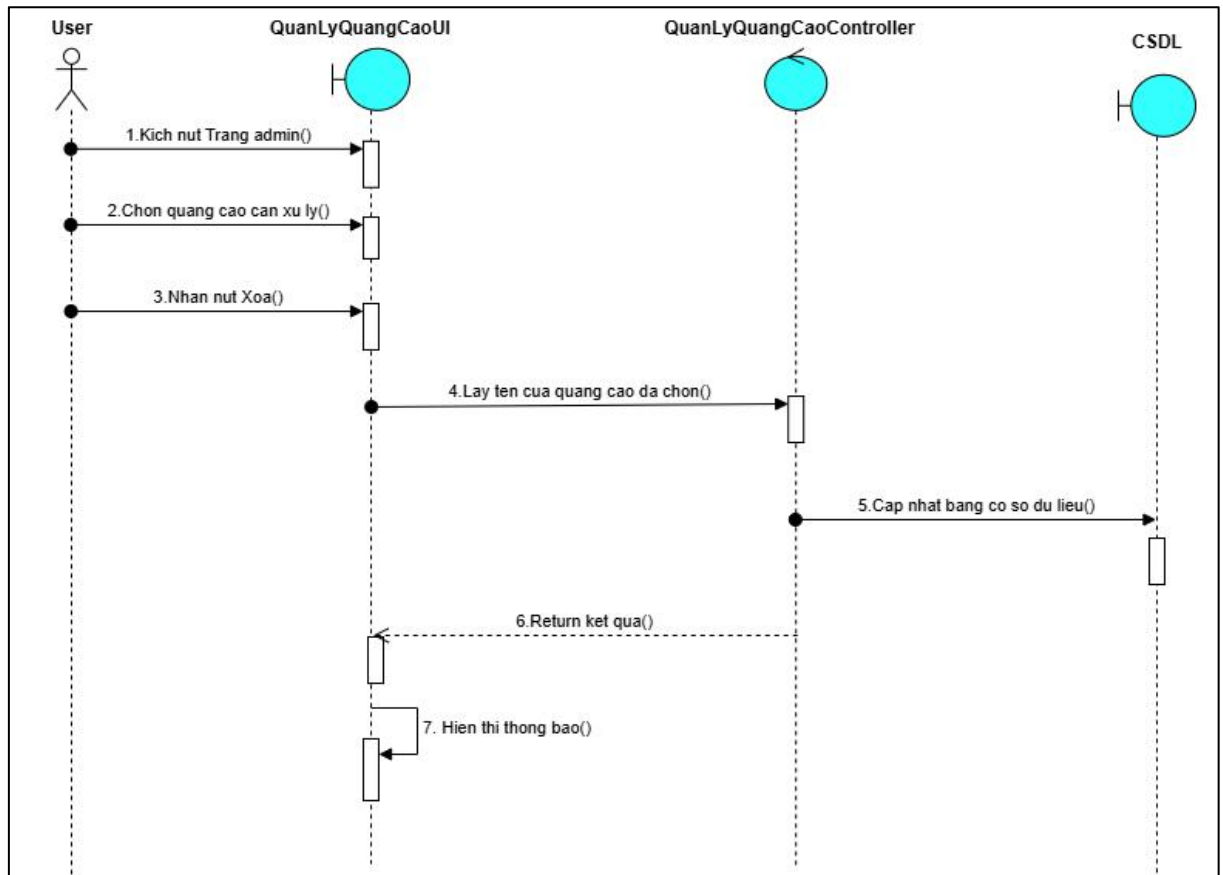


Hình 2.26. Biểu đồ trình tự use case thêm bài báo

### 2.3.16. Biểu đồ quản lý quảng cáo trên trang chủ

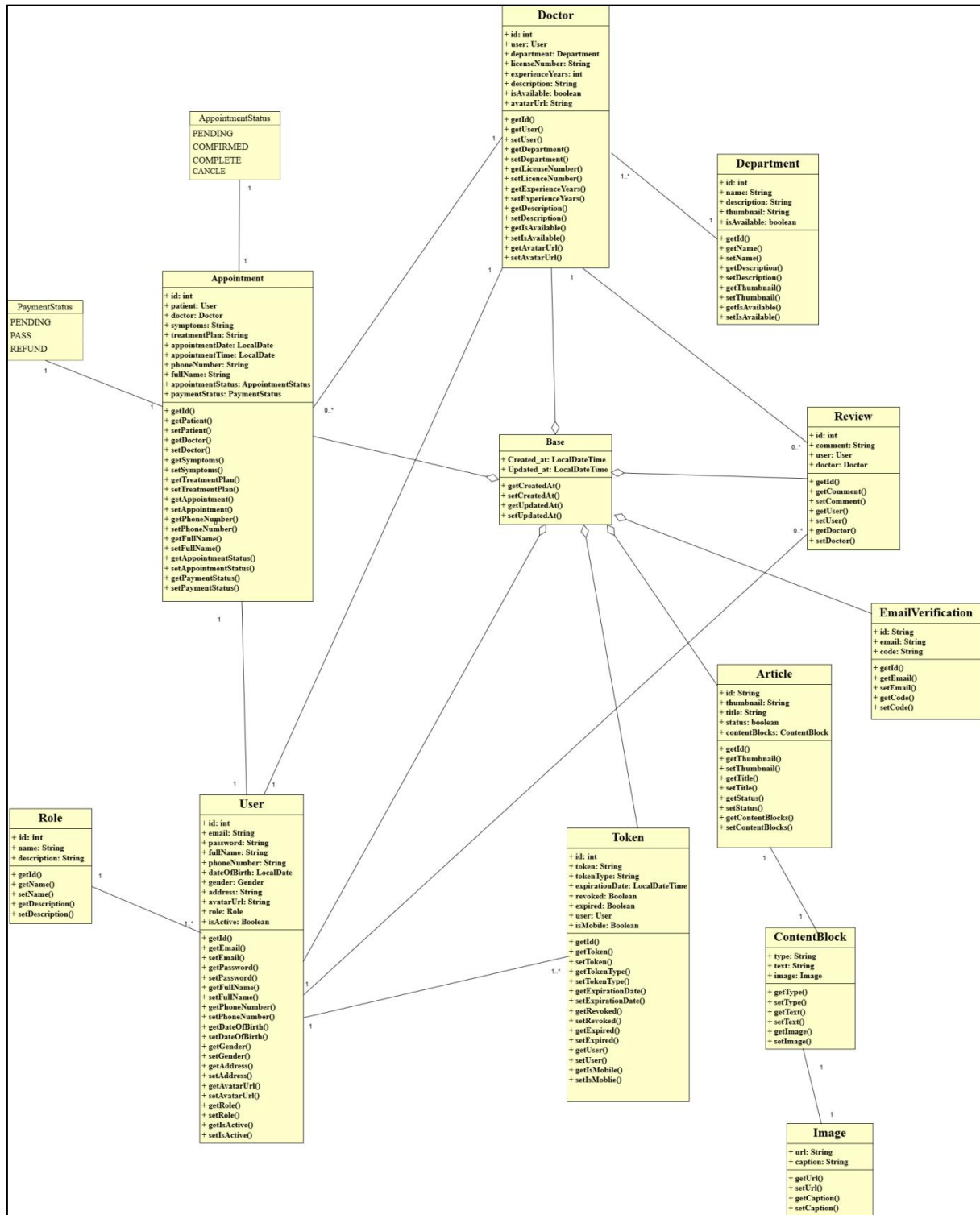


Hình 2.27. Biểu đồ trình tự use case thêm mới quảng cáo



Hình 2.28. Biểu đồ trình tự use case xóa quảng cáo

## 2.4. Biểu đồ lớp

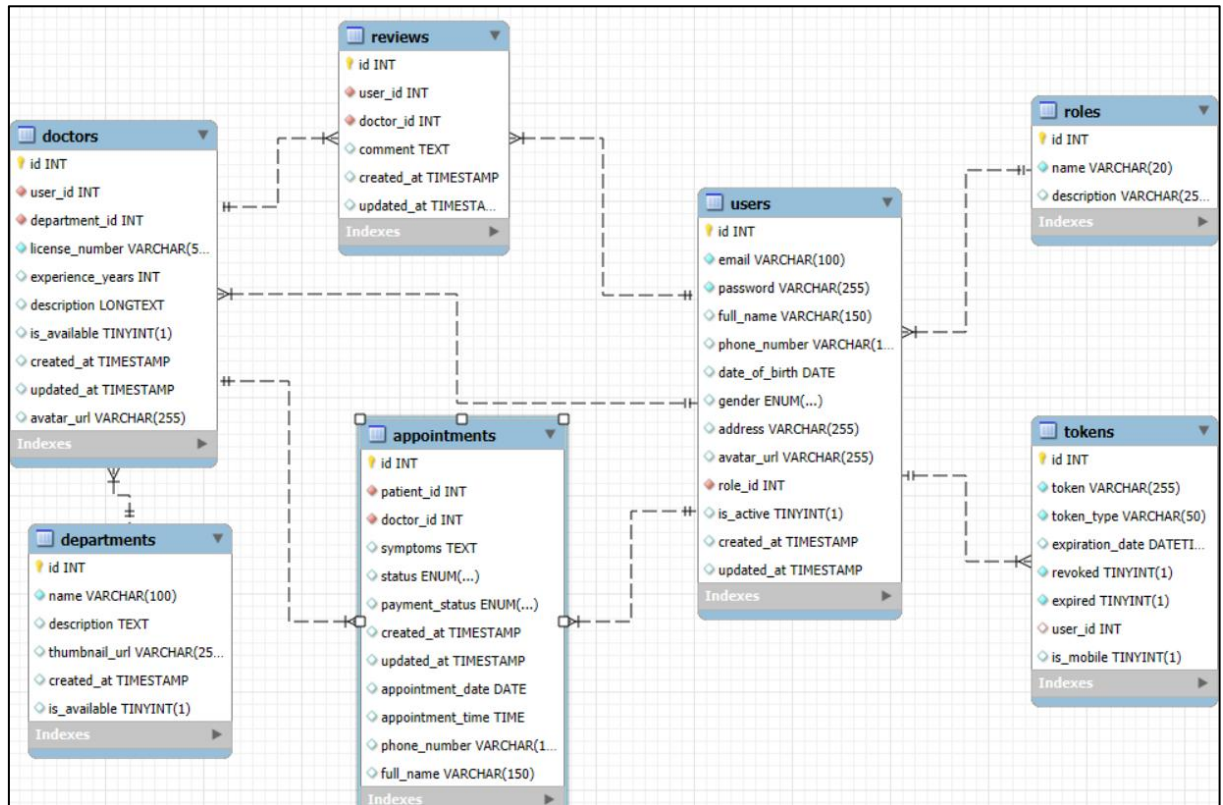


Hình 2.29. Biểu đồ lớp chính của hệ thống



## 2.5. Thiết kế cơ sở dữ liệu

### 2.5.1. Mô hình quan hệ dữ liệu



Hình 2.30. Mô hình dữ liệu của hệ thống

### 2.5.2. Chi tiết các bảng

#### 2.5.2.1. Bảng dữ liệu có quan hệ

Bảng 2.1. Chi tiết bảng user

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	Primary key, auto increase	Mã người dùng

email	varchar(100)	No	No	Email người dùng
password	varchar(255)	No	No	Mật khẩu
full_name	varchar(250)	Yes	No	Họ và tên
phone_number	varchar(15)	Yes	No	Số điện thoại
date_of_birth	date	Yes	No	Ngày sinh
gender	enum('MALE', 'FEMALE', 'OTHER')	Yes	No	Giới tính
address	varchar(255)	Yes	No	Địa chỉ
avatar_url	varchar(255)	Yes	No	Ảnh đại diện

role_id	int	No	Foreign key	Mã vai trò
is_active	boolean	No	No	Trạng thái tài khoản
create_at	timestamp	No	No	Ngày tạo
update_at	timestamp	No	No	Ngày cập nhật

*Bảng 2.2. Chi tiết bảng token*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	Primary key, auto increase	Mã token
token	varchar(255)	No	No	Nội dung token

expitation_date	datetime	No	No	Ngày hết hạn
revoked	tinyint(1)	No	No	Đã thu hồi
expired	tinyint(1)	No	No	Đã hết hạn
user_id	int	No	No	Mã người dùng
is_moblie	tinyint(1)	No	default 0	Có phải điện thoại

*Bảng 2.3. Chi tiết bảng role*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	Primary key, auto increase	Mã vai trò

name	varchar(20)	No	No	Tên vai trò
description	varchar(255)	No	No	Miêu tả

*Bảng 2.4. Chi tiết bảng review*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	Primary key, auto increase	Mã comment
user_id	int	No	Foreign key	Mã người dùng
doctor_id	int	No	Foreign key	Mã bác sĩ
comment	text	No	No	Bình luận

created_at	timestamp	No	default current_timestamp	Ngày tạo
updated_at	timestamp	No	default current_timestamp	Ngày cập nhật

*Bảng 2.5. Chi tiết bảng doctor*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	Primary key, auto increase	Mã bác sĩ
user_id	int	No	Foreign key	Mã người dùng
department_id	int	No	Foreign key	Mã chuyên khoa
license_number	varchar(50)	No	No	Mã chứng chỉ nghề

experience_year	int	No	default 0	Năm kinh nghiệm
is_available	boolean	No	default true	Trạng thái
avatar_url	varchar(255)	No	No	Ảnh đại diện
created_at	timestamp	No	default current_timestamp	Ngày tạo
updated_at	timestamp	No	default current_timestamp	Ngày cập nhật

*Bảng 2.6. Chi tiết bảng department*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	primary key, auto increase	Mã phòng ban

name	varchar(100)	No	No	Tên phòng ban
description	text	No	No	Miêu tả
thumbnail_url	varchar(255)	No	No	Ảnh đại diện
created_at	timestamp	No	default current_timestamp	Ngày tạo

*Bảng 2.7. Chi tiết bảng appointment*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	int	No	Primary key, auto increase	Mã lịch khám
patient_id	int	No	Foreign key	Mã bệnh nhân



doctor_id	int	No	Foreign key	Mã bác sĩ
symptoms	text	Yes	No	Triệu chứng
treatment_plan	text	Yes	No	Phương án điều trị
appointment_date	date	No	No	Ngày khám
appointment_time	time	No	No	Giờ khám
phone_number	varchar(15)	No	No	Số điện thoại
full_name	varchar(150)	No	No	Họ và tên
status	enum('PENDING', 'CONFIRMED', 'COMPLETED',	No	default Pending	Trạng thái khám

	'CANCELLED')			
payment_status	enum('PENDING', 'PAID', 'REFUNDED')	No	Default Pending	Trạng thái thanh toán
created_at	timestamp	No	default current_timestamp	Ngày tạo
updated_at	timestamp	No	default current_timestamp	Ngày cập nhật

#### 2.5.2.2. Bảng dữ liệu không quan hệ

Bảng 2.8. Chi tiết bảng EmailVerification

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	ObjectId	No	Primary key, auto increase	Khóa chính
email	String	Yes	No	Tài khoản email

code	String	Yes	No	Mã xác thực
created_at	Date	Yes	No	Ngày tạo

*Bảng 2.9. Chi tiết bảng Article*

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
id	ObjectId	No	Primary key, auto increase	Khóa chính
thumbnail	String	Yes	No	Ảnh đại diện
title	String	Yes	No	Tiêu đề
status	Boolean	Yes	No	Trạng thái

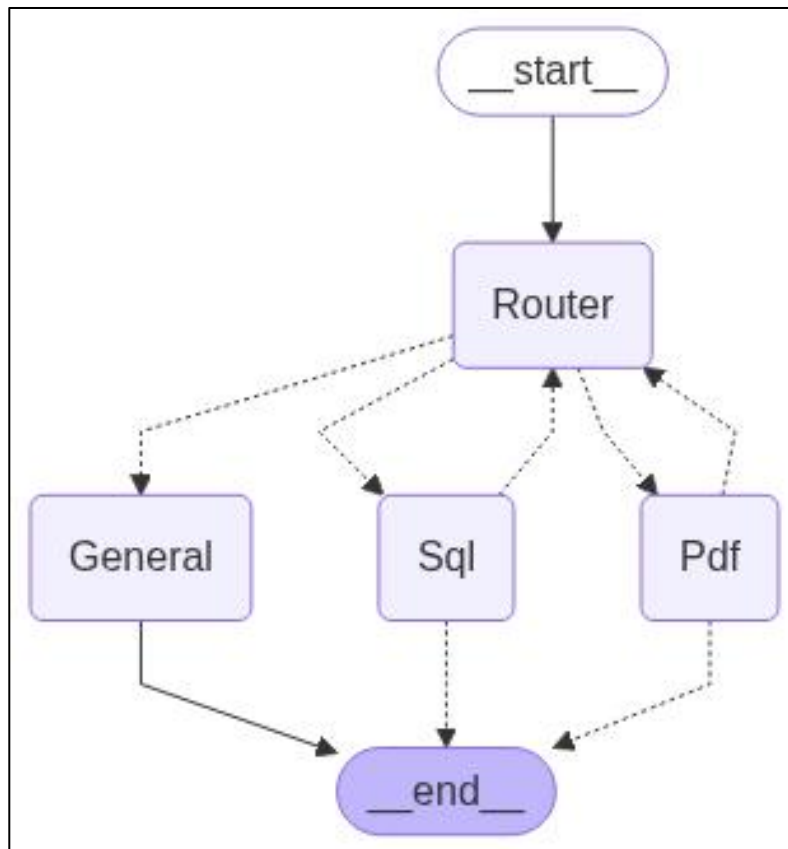
contentBlock	Array	Yes	No	Khối nội dung
contentBlock.type	String	Yes	No	Kiểu nội dung
contentBlock.text	String	Yes	No	Nội dung văn bản
contentBlock.image	String	Yes	No	Khối nội dung hình ảnh
contentBlock.image.url	String	Yes	No	Địa chỉ hình ảnh
contentBlock.image.caption	String	Yes	No	Tiêu đề hình ảnh
created_at	Date	Yes	No	Ngày tạo
updated_at	Date	Yes	No	Ngày cập nhật

**Chú thích:** Trong bảng dữ liệu không quan hệ Article:

- contentBlock: là 1 khối dữ liệu để hiển thị các khối nội dung của bài báo gồm các trường type, text, image. Kiểu type để xác định trong khối này text hay image.
- image: là 1 khối lưu trữ thông tin của ảnh hiển thị trên bài báo, gồm có 2 trường chính là url và caption.

## 2.6. Xây dựng kiến trúc hệ thống

Với xu hướng phát triển liên tục của các mô hình ngôn ngữ lớn thì nhu cầu các website được tích hợp các trợ lý ảo ngày càng cần thiết. Đối mặt với vấn đề đó, trong đề tài này tôi đã thiết kế một mô hình kiến trúc hệ thống hỏi đáp với trợ lý ảo như sau:

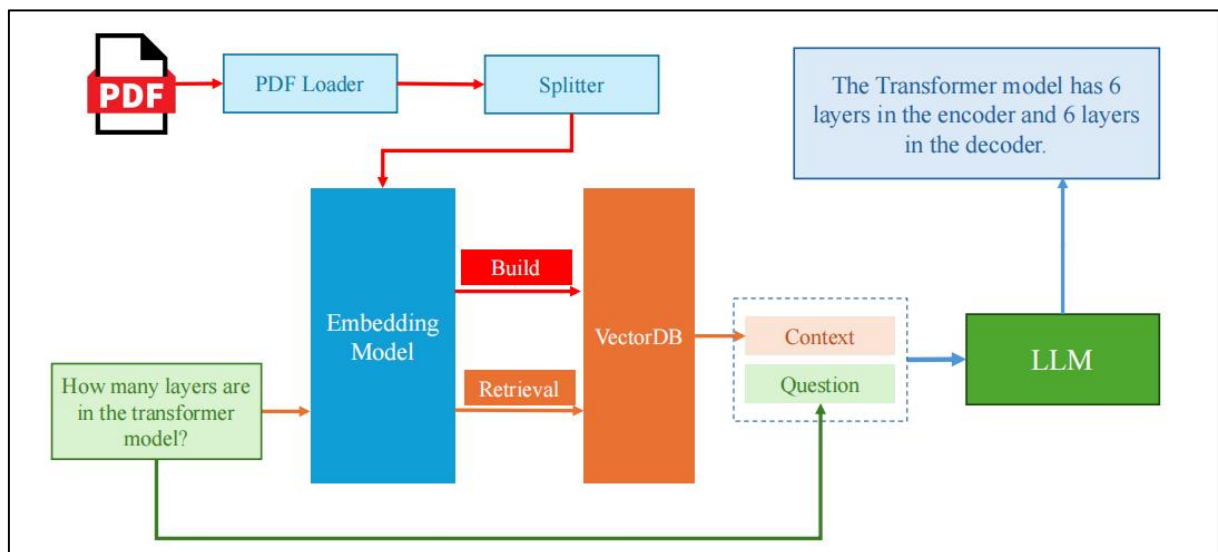


Hình 2.31. Mô hình kiến trúc chính hệ thống AI

### Chú thích:

- Node `_start_` và `_end_` là 2 node mặc định của chương trình có tác vụ bắt đầu và kết thúc của hệ thống.
- Node router có tác vụ từ câu hỏi đầu vào nhằm chọn ra 1 trong 3 node tiếp theo với các vai trò riêng để phù hợp với nhu cầu của người hỏi.
- Node General có nhiệm vụ phản hồi những câu hỏi bình thường, không liên quan đến bệnh viện và lĩnh vực y tế.
- Node Sql có nhiệm vụ phản hồi những câu hỏi về đánh giá của người dùng với bác sĩ.
- Node Pdf có nhiệm vụ trích xuất thông tin lĩnh vực y tế từ tài liệu đã được kiểm chứng và chuẩn bị sẵn lưu dưới cơ sở dữ liệu.

#### 2.6.1. Kiến trúc hệ thống của node RAG



Hình 2.32. Mô hình kiến trúc hệ thống RAG

### Theo đó:

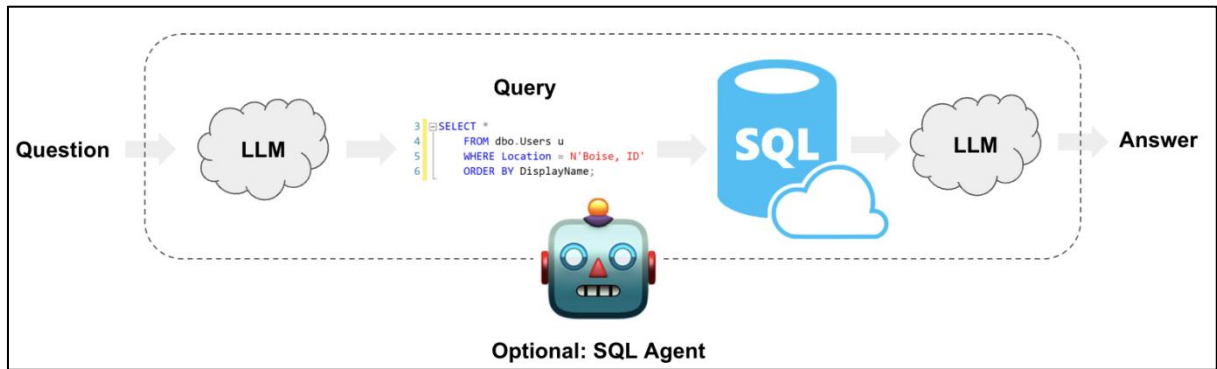
- Từ danh sách các bài báo y tế, ta tách thành các văn bản nhỏ. Từ đó, xây dựng một hệ cơ sở dữ liệu vector với một embedding model.

- Bên cạnh câu hỏi đầu vào (question), mô hình sẽ truy vấn các mẫu văn bản có liên quan đến câu hỏi, dùng làm ngữ cảnh (context) trong câu prompt. Đây là nguồn thông tin mà LLMs có thể dựa vào để trả lời câu hỏi.
- Đưa câu prompt vào mô hình (question và context) để nhận câu trả lời từ mô hình.

### **Bộ dữ liệu được sử dụng để huấn luyện RAG cho AI:**

- **WHO-UCN-GMP-2023.01-Rev.1-eng:** Đây là tài liệu từ Tổ chức Y tế Thế giới (WHO), cung cấp hướng dẫn và chính sách về sức khỏe toàn cầu. Mục tiêu là hỗ trợ các quốc gia cải thiện sức khỏe cộng đồng và đối phó với thách thức y tế thông qua chiến lược dựa trên bằng chứng.
- **STI-Guidelines-2021:** Tài liệu này đưa ra hướng dẫn cập nhật về quản lý và phòng ngừa bệnh lây truyền qua đường tình dục (STIs). Mục tiêu là trang bị cho chuyên gia y tế công cụ để chẩn đoán, điều trị và ngăn chặn STIs, bảo vệ sức khỏe cộng đồng.
- **Nesse-Dawkins-EvolMed-OUPMedTextbook-2010best:** Đây là sách giáo khoa về Y học Tiến hóa, nghiên cứu ảnh hưởng của tiến hóa đến sức khỏe và bệnh tật. Mục tiêu là cung cấp góc nhìn mới về y học, giúp hiểu nguồn gốc bệnh và phát triển phương pháp điều trị hiệu quả.
- **5\_national-clinical-guideline-of-dengue-timor-leste\_clean\_final-12-dec-2022:** Đây là hướng dẫn lâm sàng quốc gia về quản lý sốt xuất huyết dengue ở Timor-Leste. Mục tiêu là nâng cao chất lượng chăm sóc bệnh nhân và giảm tác động của dịch bệnh thông qua khuyến nghị điều trị tốt nhất.

### 2.6.2. Kiến trúc hệ thống của Node Sql



Hình 3.23. Mô hình kiến trúc hệ thống truy vấn Sql

Từ câu hỏi đầu vào mô hình sẽ kết hợp với kết quả truy vấn được lấy từ database lên để phản hồi lại người dùng



## CHƯƠNG 3. CÀI ĐẶT HỆ THỐNG VÀ MỘT SỐ KẾT QUẢ

### 3.1. Công nghệ đã sử dụng

#### 3.1.1. Phía backend

Mặc dù hiện nay có rất nhiều công nghệ được sử dụng để phát triển và thiết kế backend cho website như: PHP, ASP, ASP.NET, JSP...

Tuy nhiên Java là một loại mã nguồn mở, hiện đang được khá nhiều người dùng lựa chọn để thiết kế website nhờ những ưu điểm nổi bật như sau.

**Khả năng bảo mật cao:** Java được đánh giá là có độ bảo mật cao hơn khá nhiều so với các ngôn ngữ lập trình khác. Do đó khi sử dụng mã nguồn này thì bạn sẽ không phải lo ngại nhiều về vấn đề bảo mật hay tình trạng bị hack thông tin trên web.

**Hoạt động mạnh mẽ:** Ưu điểm tiếp theo khiến cho Java được yêu thích đó chính là có khả năng mang lại một website hoạt động mạnh mẽ trên bất kỳ nền tảng nào. Đây là ưu điểm mà không phải bất kỳ ngôn ngữ nào cũng có thể đảm bảo được.

**Thông dịch:** Ngôn ngữ lập trình của Java vừa có khả năng biên dịch, vừa có khả năng thông dịch. Theo đó, chương trình nguồn viết bằng Java có đuôi \*.java đầu tiên được biên dịch thành tập tin có đuôi \*.class, sau đó sẽ tiếp tục được thông dịch thành mã máy.

**Độc lập nền, khả năng chuyên:** Các chương trình được viết bởi ngôn ngữ Java có khả năng chạy trên nhiều máy tính, nhiều hệ điều hành khác nhau, chỉ cần ở đó có cài đặt máy ảo Java Virtual Machine. Tóm lại, ưu điểm cực lớn của Java chính là “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).

**Hướng đối tượng:** Hướng đối tượng trong ngôn ngữ lập trình Java khá giống với C++, tuy nhiên lại là ngôn ngữ lập trình hướng đối tượng hoàn toàn.

**Đa nhiệm - đa luồng (MultiTasking - Multithreading):** Ngôn ngữ lập trình này hỗ trợ lập trình đa nhiệm, đa luồng nên cho phép chạy song song nhiều tiến trình, tiểu trình trong cùng một thời điểm và tạo tương tác với nhau.

**Hỗ trợ mạnh cho việc phát triển ứng dụng:** Một trong những ưu điểm không thể không kể đến của Java chính là cung cấp nhiều công cụ, thư viện lập trình, từ đó hỗ trợ phong phú cho việc phát triển nhiều loại hình ứng dụng khác nhau như:

- J2SE (Java 2 Standard Edition): hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server.
- J2EE (Java 2 Enterprise Edition): hỗ trợ phát triển các ứng dụng thương mại.
- J2ME (Java 2 Micro Edition): hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây,..

Chính vì những ưu điểm của việc phát triển ứng dụng web bằng Java như vậy nên em quyết định sử dụng công nghệ Java: Spring MVC, Spring JPA, Spring Security, Spring Boot để ứng dụng vào đề tài thực tập tốt nghiệp lần này.

### 3.1.2. Phía Front end

Bên cạnh các framework phổ biến cho phát triển giao diện như React, Vue.js thì Angular là một lựa chọn phù hợp thân thiện cho người dùng springboot với backend bởi cách thiết kế và quản lý file của 2 framework có tính tương đồng là đều sử dụng kỹ thuật Dependency Injection (DI) để quản lý.

#### **Các tính năng nổi bật của Angular:**

- Component và model: Ứng dụng angular được chia thành các component và module rõ ràng, giúp tổ chức ứng dụng rõ ràng, dễ quản lý.
- Two-way data binding: Hỗ trợ kết nối 2 chiều dữ liệu giữa mô hình với giao diện, giúp thuận tiện và giảm thiểu mã cần viết để đồng bộ thông tin.

- RxJs & Observables: Sử dụng các thư viện RxJs cho lập trình phản ứng (reactive programming), giúp xử lý các tác vụ bất đồng bộ và sự kiện dễ dàng hơn.

### 3.1.3. Phía Database

#### ❖ Cơ sở dữ liệu Mysql

**Hệ quản trị quan hệ mã nguồn mở:** MySQL là RDBMS phổ biến hàng đầu, hỗ trợ lưu trữ dữ liệu theo các bảng có schema rõ ràng. Nó hoạt động ổn định trên nhiều nền tảng (Linux, Windows, macOS...) và có phiên bản Community miễn phí cài đặt dễ dàng.

**Tính năng ACID và transaction:** MySQL hỗ trợ đầy đủ cơ chế giao dịch ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo mọi thay đổi dữ liệu đều nhất quán và an toàn kể cả khi hệ thống gặp sự cố. Nhờ vậy, MySQL thích hợp cho các ứng dụng cần độ chính xác cao như hệ thống tài chính, giao dịch bán hàng

**Hiệu năng cao, xử lý đồng thời tốt:** MySQL được tối ưu để xử lý khối lượng lớn kết nối đồng thời và các truy vấn đọc-ghi thường xuyên. Ví dụ, MySQL có thể xử lý nhiều kết nối cùng lúc với hiệu suất tốt. Ngoài ra, MySQL cũng hỗ trợ nhiều cơ chế lưu trữ (InnoDB, MyISAM...) để tùy chỉnh theo mục đích sử dụng.

**Hỗ trợ tính năng hiện đại:** Mặc dù là hệ cơ sở dữ liệu quan hệ truyền thống, MySQL hiện đã bổ sung khả năng lưu trữ dữ liệu JSON và các kiểu dữ liệu phức tạp khác, giúp tích hợp tốt hơn với các ứng dụng hiện đại. MySQL cũng hỗ trợ full-text index, các trigger, stored procedure cơ bản, đáp ứng tốt nhu cầu phát triển.

**Dễ sử dụng và triển khai:** Cú pháp SQL của MySQL đơn giản, dễ học với người mới; cấu hình và cài đặt dễ dàng, đặc biệt là trong các môi trường LAMP/PHP/Java. MySQL cũng được các nhà cung cấp đám mây lớn (AWS,

Google Cloud, Azure) hỗ trợ đầy đủ, giúp triển khai linh hoạt và quản lý thuận tiện

### ❖ **Cơ sở dữ liệu MongoDB:**

**Cơ sở dữ liệu theo mô hình document:** MongoDB lưu trữ dữ liệu dưới dạng tài liệu BSON (gần giống JSON), cho phép mỗi bản ghi linh hoạt lưu các trường khác nhau mà không cần schema tĩnh. Thiết kế này giúp mô hình dữ liệu gần gũi với ngôn ngữ lập trình hướng đối tượng, giảm bớt lớp ánh xạ ORM phức tạp.

**Mở rộng ngang và sẵn sàng cao:** MongoDB tích hợp sẵn cơ chế phân mảnh (sharding) và sao chép (replica set) giúp tự động chia tải và đảm bảo dữ liệu luôn có bản sao dự phòng. Nhờ vậy, MongoDB quy mô từ các hệ thống nhỏ đến phân tán toàn cầu đều có khả năng mở rộng linh hoạt và độ chịu lỗi cao.

**Truy vấn linh hoạt, tốc độ cao:** MongoDB hỗ trợ ngôn ngữ truy vấn MQL mạnh mẽ, cho phép thực hiện truy vấn đa dạng trên các tài liệu với điều kiện phức tạp và hỗ trợ nhiều loại chỉ mục (compound, text, geo, TTL,...). Điều này giúp truy xuất và phân tích dữ liệu theo thời gian thực nhanh chóng. Ngoài ra, MongoDB có khả năng chủ yếu lấy dữ liệu từ RAM để tăng hiệu suất, đẩy nhanh việc trả kết quả truy vấn.

**Hỗ trợ giao dịch ACID đa tài liệu:** Bắt đầu từ phiên bản 4.0, MongoDB đã bổ sung khả năng giao dịch ACID trên nhiều tài liệu, cho phép thực hiện các thay đổi phức tạp một cách toàn vẹn tương tự như cơ sở dữ liệu quan hệ. Tính năng này khắc phục hạn chế trước đây và giúp MongoDB có thể đảm bảo tính nhất quán cao khi cần.

**Hệ sinh thái phong phú:** MongoDB có nhiều công cụ, thư viện và dịch vụ đi kèm (như Compass, Atlas) hỗ trợ phát triển. Nó hỗ trợ nhiều ngôn ngữ lập trình phổ biến (JavaScript, Python, Java, C#, ...) và có cộng đồng người dùng rộng lớn, giúp dễ tìm tài liệu và hỗ trợ kỹ thuật.

### 3.1.4. Phía Trợ lý ảo

Một framework khá phổ biến những thời gian gần đây là Langgraph, mã nguồn mở cho phép bạn xây dựng các ứng dụng AI phức tạp dựa trên multi-agent system và graph-based workflow, đặc biệt phù hợp cho các mô hình ngôn ngữ lớn (LLM) như GPT, Claude, Mistral, v.v. Nói một cách dễ hiểu LangGraph là sự kết hợp giữa LangChain và Graphs. Nếu LangChain chỉ giúp tạo luồng logic từng bước, thì LangGraph cho phép xây dựng “máy trạng thái” (state machine) có vòng lặp, nhánh rẽ, và agent phức tạp hơn rất nhiều.

#### Tính năng nổi bật của Langgraph

- Hỗ trợ stateful execution để lưu và cập nhật trạng thái từng bước
- Dễ thiết kế các hệ thống multi-agent có giao tiếp phức tạp
- Chạy song song, xử lý theo hướng asynchronous tốt
- Tích hợp sẵn với LangChain components: retriever, memory, tools, chains, agent...
- Có thể deploy dưới dạng microservice workflow hoặc chạy local

### 3.2. Môi trường đã sử dụng

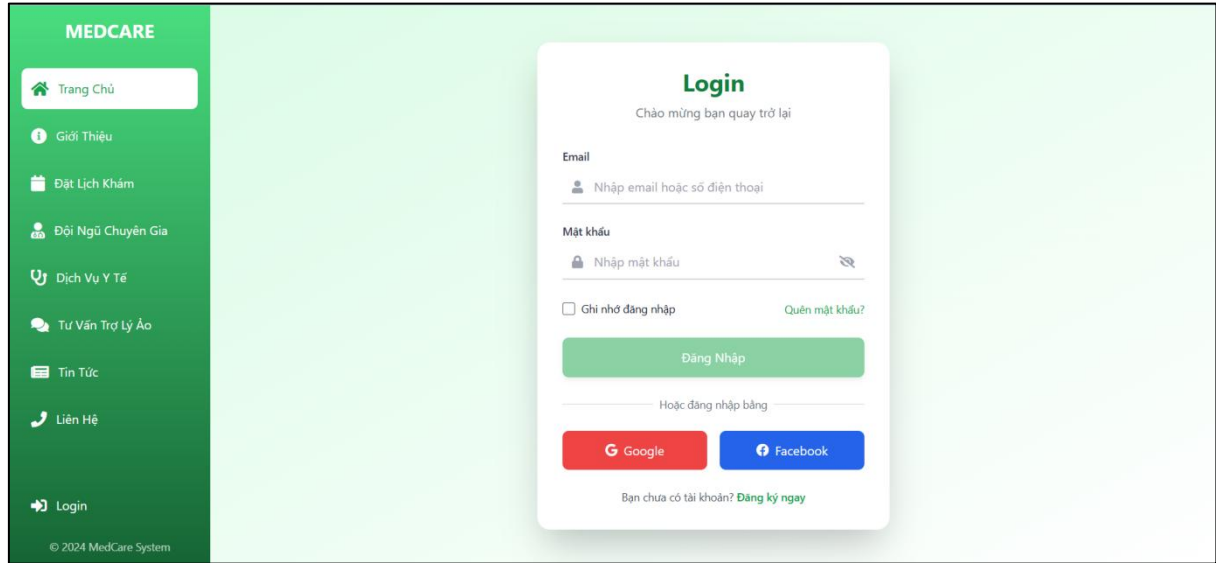
Visual Studio Code là một trình soạn thảo mã nguồn mở, nhẹ, đa nền tảng (Windows/Linux/Mac) do Microsoft phát triển (MIT License).

VSCode sử dụng công nghệ web (Electron – bao gồm Chromium và Node.js) làm nền tảng, với kiến trúc phân lớp (core base, platform, Monaco editor, workbench) và Extension Host chạy tách biệt để đảm bảo tính ổn định.

Nhờ tính mở và cộng đồng lớn, VSCode hỗ trợ hàng trăm ngôn ngữ lập trình thông qua extension (các ngôn ngữ phổ biến như JavaScript, TypeScript, HTML, CSS... có sẵn hoặc dễ dàng cài thêm).

Theo khảo sát StackOverflow 2024, VSCode là môi trường phát triển được dùng nhiều nhất (có tới 74% lập trình viên chuyên nghiệp sử dụng) và đứng đầu danh sách “mong muốn dùng”.

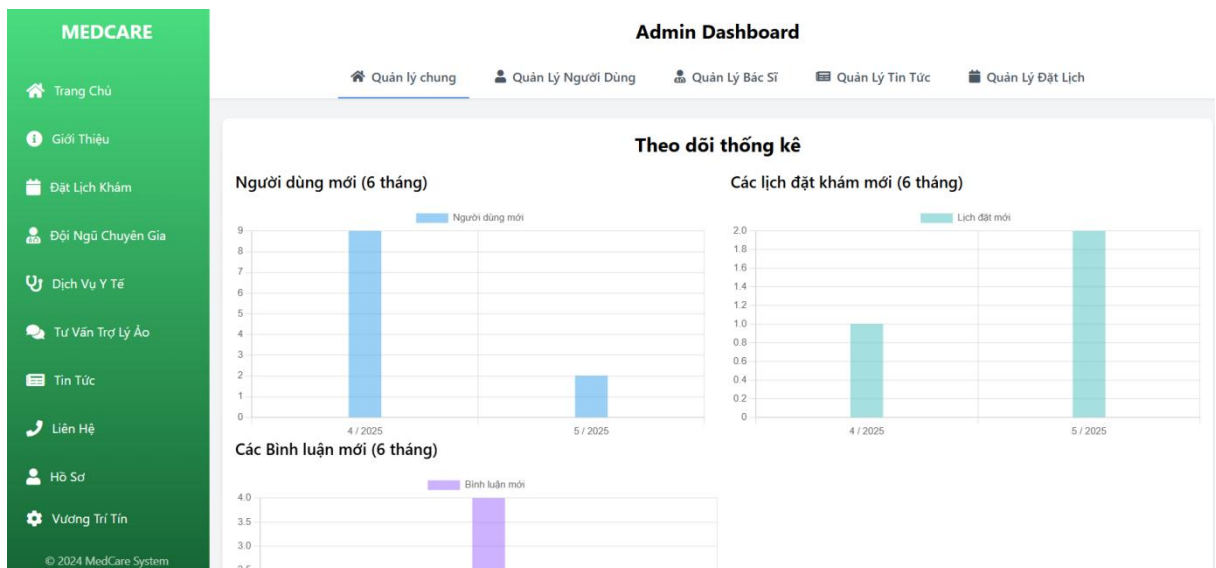
### 3.3. Một số kết quả đạt được



Hình 3.24. Giao diện đăng nhập

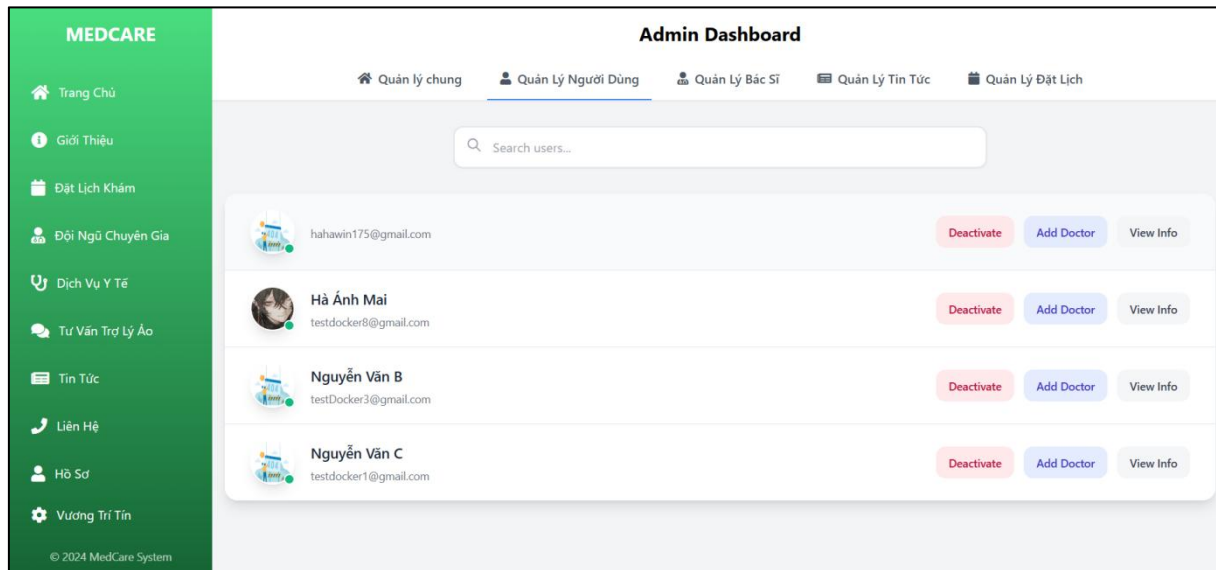
Người dùng sẽ nhập tài khoản và mật khẩu vào để đăng nhập. Nếu đăng nhập sai thì hệ thống sẽ phản hồi là đăng nhập tài khoản hoặc mật khẩu không chính xác. Còn với trường hợp đăng nhập thành công thì người dùng sẽ được đưa đến trang tương ứng với vai trò tài khoản.

Dưới đây là giao diện khi người dùng đăng nhập thành công với vai trò Admin, Hệ thống sẽ xuất hiện các trang quản lý của website sau:



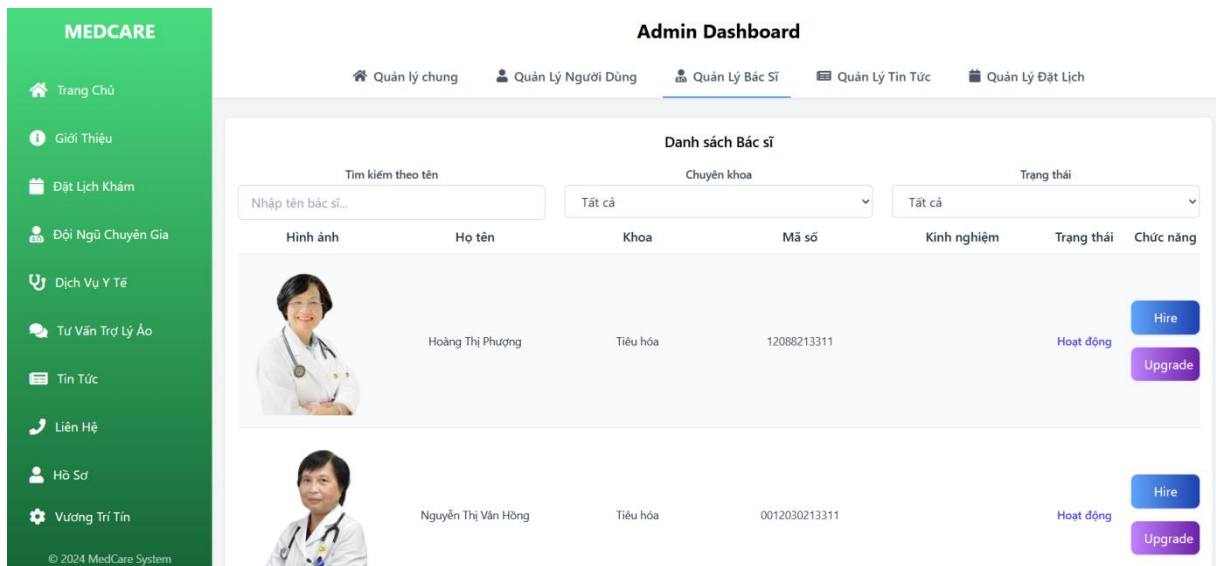
Hình 3.25. Giao diện trang quản lý chung của Admin

Với trang quản lý chung này Admin có thể theo dõi và thống kê các hoạt động chính của website và tùy chỉnh quảng cáo trên trang chủ.



Hình 3.26. Giao diện quản lý người dùng của Admin

Với trang Quản lý người dùng thì Admin có thể tùy ý theo dõi, tùy chỉnh trạng thái người dùng, phong người dùng lên vài trò bác sĩ khi bác sĩ đã tạo thành công tài khoản hoặc theo dõi thông tin chi tiết của người dùng.



Hình 3.27. Giao diện quản lý Bác sĩ của Admin

Với trang quản lý Bác sĩ, Admin tùy chỉnh đưa bác sĩ ẩn bác sĩ đi hoặc cập nhật lại thông tin của bác sĩ.

**MEDCARE**

**Admin Dashboard**

Quản lý chung Quản Lý Người Dùng Quản Lý Bác Sĩ Quản Lý Tin Tức Quản Lý Đặt Lịch

### Thêm mới bài viết

Ảnh đại diện  
Choose File No file chosen

Tiêu đề bài viết  
Nhập tiêu đề...

Thêm Text Thêm Image

Đăng bài

### Theo dõi bài báo

Nhập tên bài báo...

© 2024 MedCare System

Hình 3.28. Giao diện quản lý tin tức của Admin

Trang quản lý tin tức giúp dễ dàng thêm mới, xóa hoặc biên soạn 1 bài báo y tế mới lên website.

**MEDCARE**

**Admin Dashboard**

Quản lý chung Quản Lý Người Dùng Quản Lý Bác Sĩ Quản Lý Tin Tức Quản Lý Đặt Lịch

Bệnh nhân: Tìm theo tên bệnh nhân  
Bác sĩ: Tìm theo tên bác sĩ  
Trạng thái: Tất cả  
Thanh toán: Tất cả

NGÀY & GIỜ	BỆNH NHÂN	BÁC SĨ	TRIỆU CHỨNG	TRẠNG THÁI	THANH TOÁN	CHỨC NĂNG
2025-05-16 16:00	Nguyễn Văn C	Hoàng Thị Phương		PENDING	PENDING	
2025-05-09 09:00	Nguyễn Văn B	Nguyễn Thị Văn Hồng	hurt head	COMPLETED	PAID	Xuất PDF
2025-04-28 08:00	Hà Ánh Mai	Nguyễn Thị Văn Hồng		PENDING	PENDING	

← Trước 1 / 1 Sau →

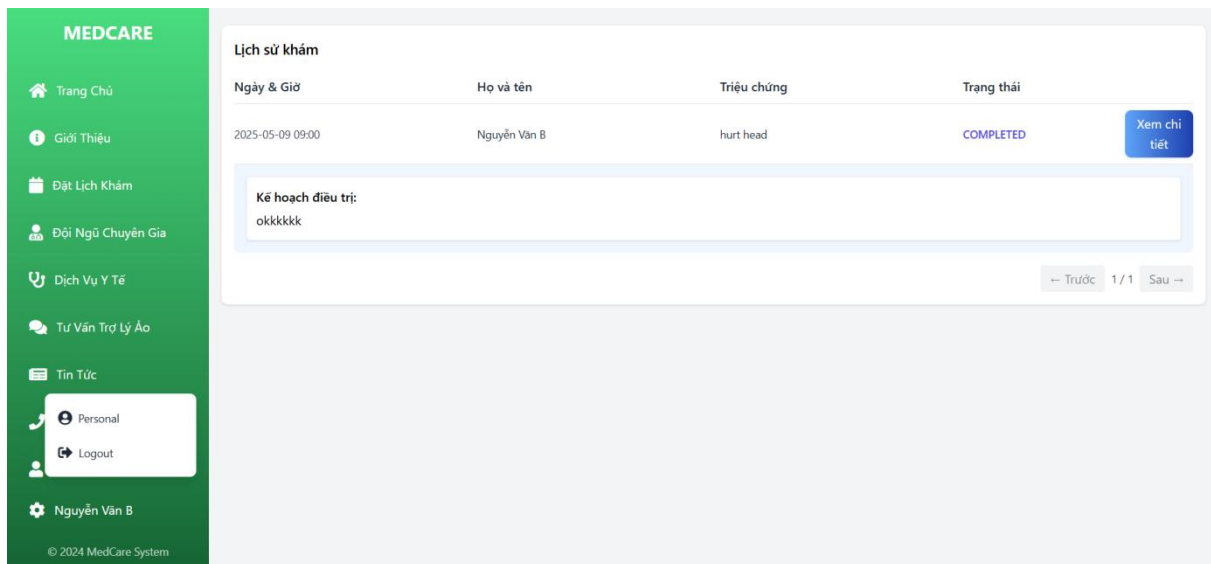
© 2024 MedCare System

Hình 3.29. Giao diện trang quản lý lịch khám của Admin

Trang quản lý giao dịch thì giúp Admin có thể theo dõi, cập nhật trạng thái nếu người dùng đã thanh toán viện phí khám và xuất ra file báo cáo kết quả khám.

Nếu người dùng đăng nhập thành công với vai trò User thì sẽ có thêm trang sau:

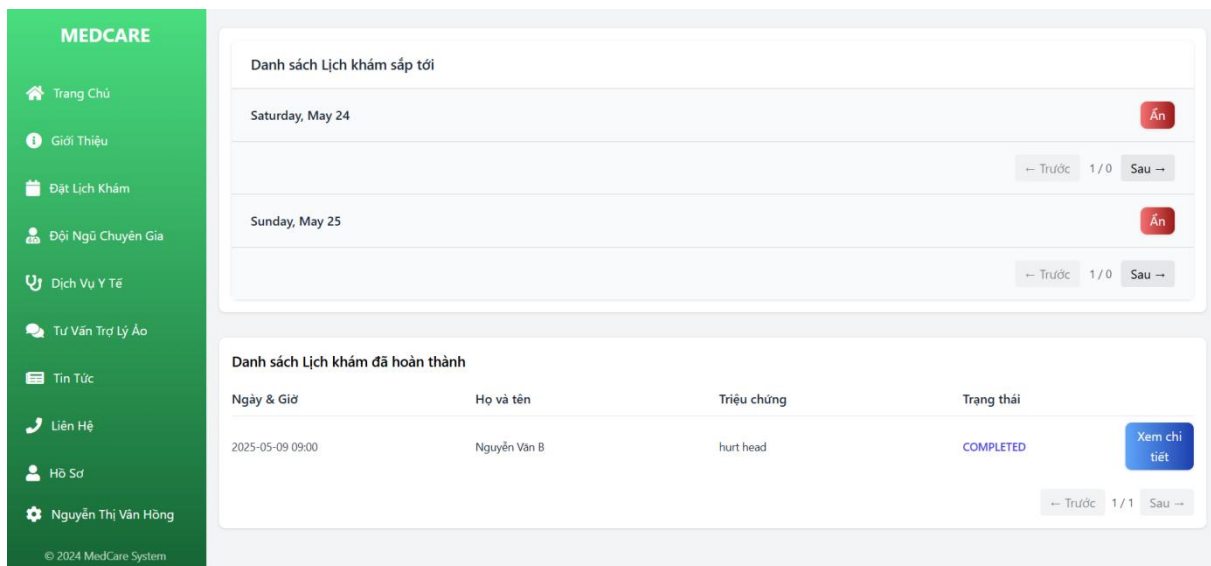




Hình 3.30. Giao diện trang lịch khám cá nhân của người dùng

Tại đây người dùng có thể theo dõi các lịch khám mình đã đặt, nếu trạng thái đã hoàn thành thì có thể theo dõi lại kết quả khám.

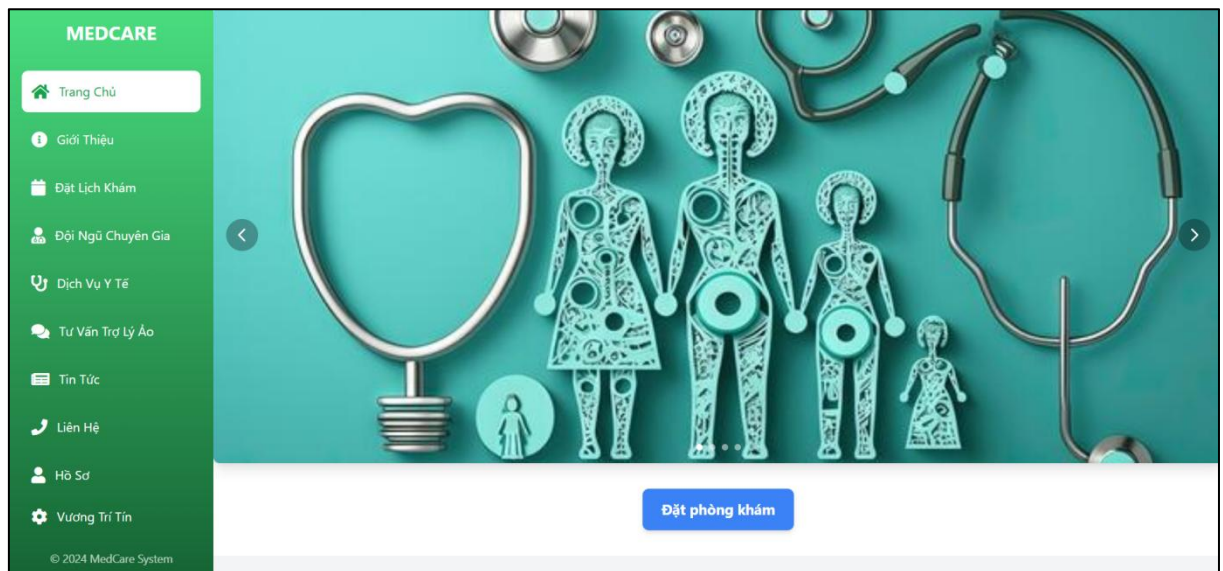
Nếu người dùng đăng nhập thành công với vai trò Doctor thì sẽ có thêm trang chức năng theo dõi các lịch khám sắp tới trong tuần:



Hình 3.31. Giao diện trang quản lý lịch khám sắp tới của Bác sĩ

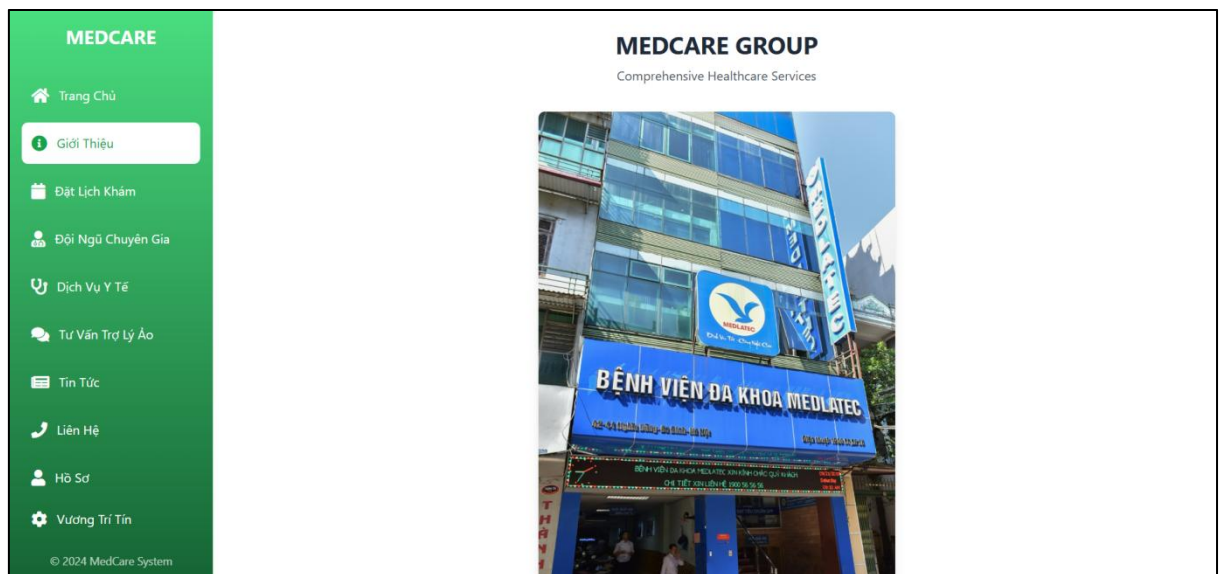
Tại trang này Bác sĩ có thể theo dõi lịch khám sắp tới và xem lại các lịch khám đã hoàn thành, đặc biệt có thể cập nhật phương án điều trị vào lịch khám đã đặt của bệnh nhân sau khi chẩn đoán xong.

Dưới đây là các trang chung khác hàng có thể truy cập để xem:



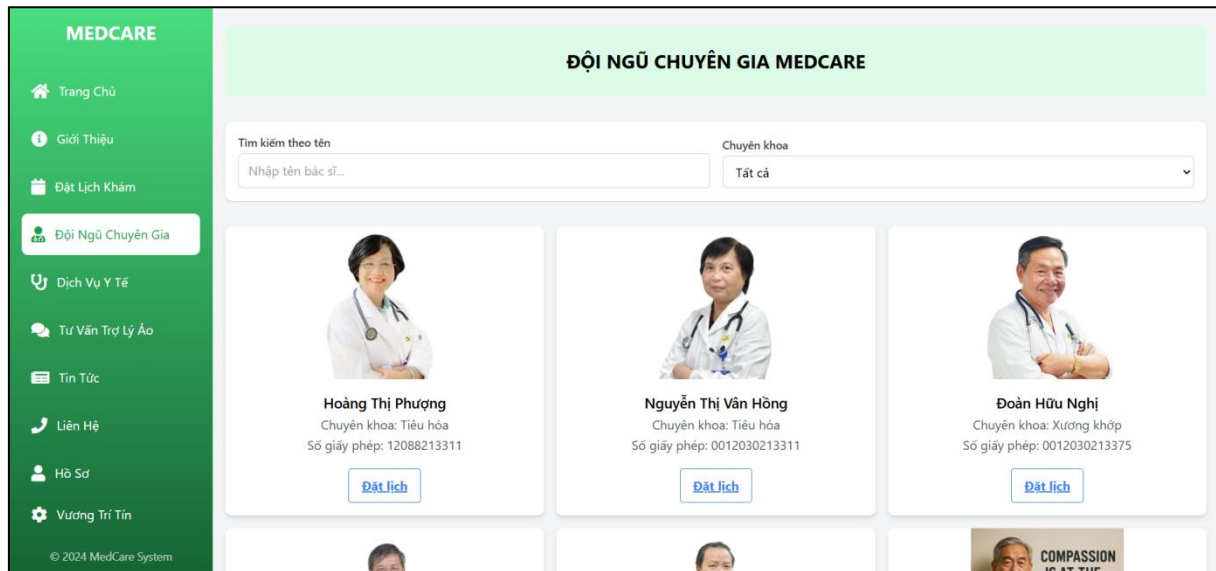
Hình 3.32. Giao diện trang chủ

Tại trang chủ người dùng có thể theo dõi các hoạt động nổi bật của bệnh viện, từ đó lựa chọn dịch vụ tốt cho bản thân, còn nếu muốn xem chi tiết các dịch vụ thì có thể ghé qua bên trang Dịch vụ.



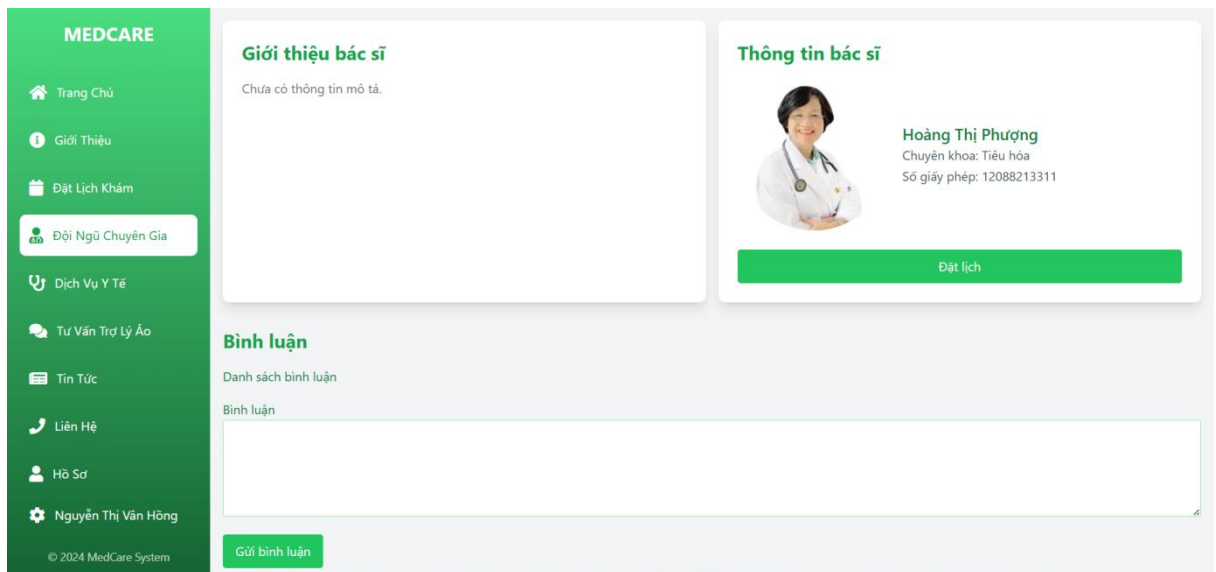
Hình 3.33. Giao diện trang giới thiệu về Bệnh viện

Trang này chỉ giới thiệu đôi nét về lịch sử hình thành và phát triển của Bệnh viện.



Hình 3.34. Giao diện lựa chọn chuyên khoa phù hợp

Ở đây, người dùng có thể dễ dàng tìm kiếm lựa chọn chuyên khoa, bác sĩ với nhu cầu. Đặc biệt khi lựa chọn được một bác sĩ thì sẽ được đưa tới giao diện đặt lịch, gồm chi tiết các thông tin của Bác sĩ gồm sơ yếu lý lịch, năm kinh nghiệm.



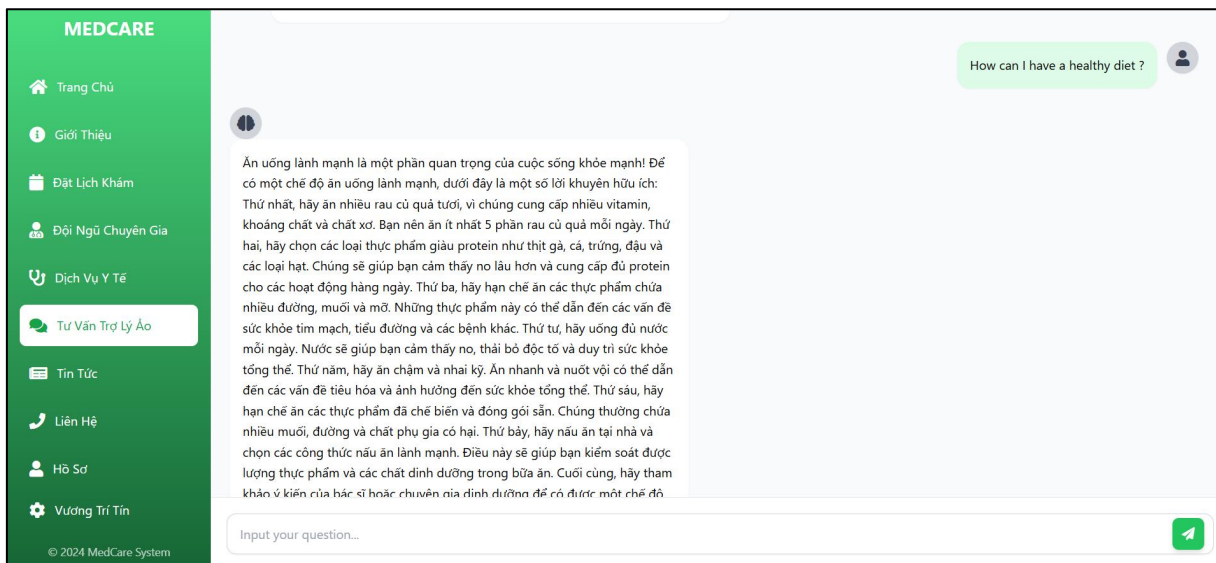
Hình 3.35. Giao diện đặt lịch

Tại đây người dùng cũng có thể để lại bình luận đánh giá về vị chuyên gia này hoặc đặt lịch khám, đặc biệt để đặt lịch khám thì người dùng phải có tài khoản.



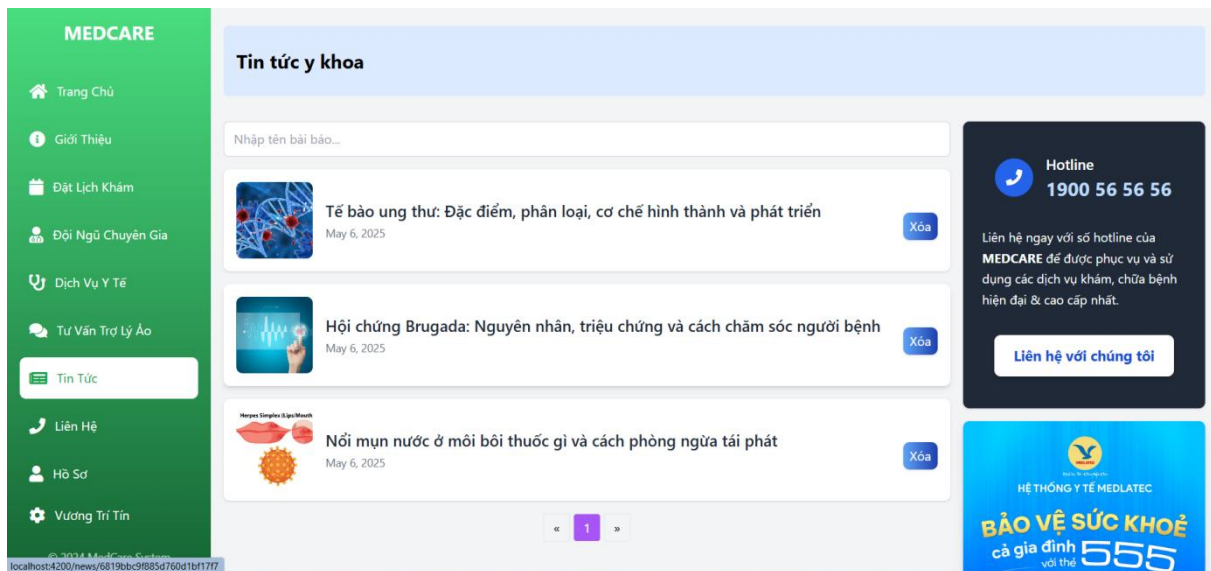
Hình 3.36. Giao diện theo dõi các dịch vụ của Bệnh viện

Tại đây người dùng có thể dễ dàng theo dõi các dịch vụ của Bệnh viện đang có.



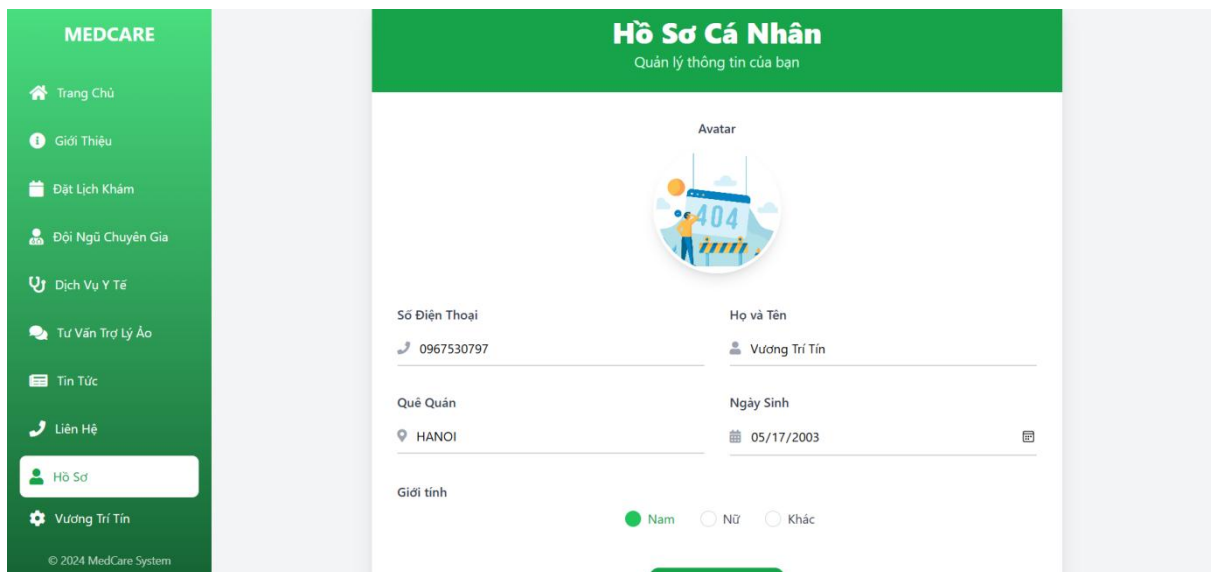
Hình 3.37. Giao diện tư vấn với trợ lý ảo

Giao diện này cho phép người dùng tương tác trò chuyện với trợ lý ảo thông minh, cho phép trò chuyện, hỏi đáp về lĩnh vực y tế hay đánh giá của bệnh nhân về bác sĩ.



Hình 3.38. Giao diện xem tin tức Y khoa

Giao diện dễ dàng giúp người đọc, tìm kiếm các bài báo khác nhau về lĩnh vực y khoa.



Hình 3.39. Giao diện cập nhật hồ sơ cá nhân

Tại trang này người dùng có thể dễ dàng tùy chỉnh lại các thông tin cá nhân của bản thân.



Hình 3.40. Giao diện đăng ký

Giao diện giúp người dùng tạo tài khoản, tại đây người dùng sẽ tạo tài khoản và nhập mã xác nhận được gửi về email.

Hình 3.41. Giao diện lấy mật khẩu mới

Nếu người dùng quên mất mật khẩu thì có thể lấy lại tại đây, người dùng sẽ nhập thông tin tài khoản và mật khẩu mới, để kích hoạt thì sẽ lấy mã xác nhận được gửi về email để nhập vào.

## KẾT LUẬN

Qua quá trình làm đồ án tốt nghiệp em đã rút ra cho mình được rất nhiều kinh nghiệm, kiến thức bổ ích phục vụ cho công việc sau khi ra trường. Với đề tài “Xây dựng trang web đặt phòng khám và điều trị tại bệnh viện tích hợp Chatbot bằng Spring boot và Angular” em đã tiếp cận và học thêm được các kiến thức cũng như kỹ năng:

- Sử dụng được ngôn ngữ Java và nắm được các kiến thức cơ bản về Spring MVC, Spring Data JPA, Spring Security, Spring Boot.
- Nắm được các kiến thức về HTML, CSS, Typescript, Angular, Tailwinds và có thể vận dụng cơ bản vào website.
- Ứng dụng được các câu lệnh trong MySQL, MongoDB để thực hiện truy vấn ra những kết quả cần thiết, từ đó hiển thị dữ liệu từ Database lên giao diện của website.
- Xây dựng được 1 hệ thống trợ lý ảo, hỗ trợ người dùng trong quá trình lựa chọn chuyên gia phù hợp cho bản thân.
- Nắm bắt được các chức năng cơ bản mà một website cần có và cách thức một website hoạt động.

Do thời gian thực hiện đồ án có hạn nên em chưa hoàn thiện được đầy đủ một ứng dụng website đặt lịch. Trang web cũng đã có những chức năng cơ bản như:

- Phía người dùng đã hiển thị được danh sách chuyên gia, có thể tìm kiếm chuyên gia theo phòng ban, có thể đặt lịch khám, tìm kiếm, theo dõi, xem lịch sử khám, đánh giá chuyên gia, liên hệ hỗ trợ viên, theo dõi để cập nhật các bài báo về lĩnh vực y tế, hỏi đáp với trợ lý ảo...
- Phía quản trị đã có thể quản lý được chuyên gia, người dùng, tài khoản, lịch khám, bài báo, phản hồi của người dùng, bảo trì trang chủ, xem báo cáo thống kê, phê duyệt lịch khám, ...

Trong thời gian tới em sẽ thêm một số chức năng để trang web được hoàn thiện hơn như:

- Thêm mục “yêu thích” chuyên gia cho khách hàng để khách hàng có thể thêm bác sĩ vào danh sách yêu thích của mình.
- Thêm nhiều chức năng lịch khám hơn.
- Thêm chức năng đăng nhập bằng Facebook, Google.
- Mở rộng khả năng chịu tải của website.



## TÀI LIỆU THAM KHẢO

- [1].Nguyễn Văn Thành - Lập trình hướng đối tượng Java, 2013.
- [2].Phạm Hữu Khang - Lập trình cơ sở dữ liệu tập II: thiết kế cơ sở dữ liệu, tạo các thủ tục xử lý, 2005.
- [3]. Web site hướng dẫn viết code bằng Java, URL:  
[Http://www.youtube.com/watch?v=d\\_G\\_beKu1Jc&list=PLWBrqglNjNl1Tuc2KdLO5kte3hSm8voV](Http://www.youtube.com/watch?v=d_G_beKu1Jc&list=PLWBrqglNjNl1Tuc2KdLO5kte3hSm8voV). Lần truy cập gần nhất: 30/12/2024