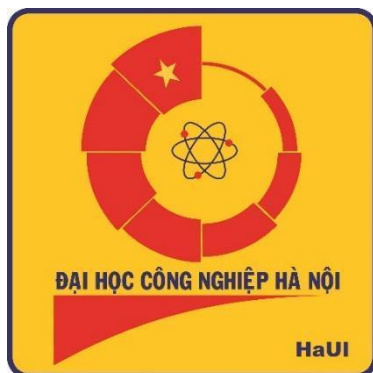


TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
HỌC PHẦN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM
NGHIÊN CỨU VỀ MÔ HÌNH THÁC NƯỚC VÀ ỨNG
DỤNG TRONG PHẦN MỀM THĂM KHÁM Y TẾ VINMEC

Giảng viên hướng dẫn: T.S Vũ Đình Minh

Nhóm – Lớp : 12 – 20232IT6082007

Sinh viên thực hiện: Trần An – 2022600916

Vũ Trí Khoa – 2022604805

Trần Văn Nhã – 2022603089

Nguyễn Văn Hoàng – 2022123456

Hà Nội - Năm 2024

LỜI CẢM ƠN

Trước hết chúng em xin bày tỏ tình cảm và lòng biết ơn đối với thầy Vũ Đình Minh – Khoa Công Nghệ Thông Tin – Đại học Công Nghiệp Hà Nội, người đã dành rất nhiều thời gian, trực tiếp hướng dẫn tận tình, chỉ bảo chúng em trong suốt quá trình làm bài tiểu luận.

Chúng em chân thành cảm ơn thầy đã truyền đạt những kiến thức quý báu trong suốt thời gian 5 tuần học trực tiếp, và những tuần học trực tuyến sau đó để chúng em hiểu rõ hơn các vấn đề mình nghiên cứu.

Tuy có nhiều cố gắng trong quá trình học tập, nhưng trong quá trình làm bài tiểu luận không thể tránh khỏi những thiếu sót về mặt kiến thức, khả năng tiếp thu thực tế, chúng em rất mong được sự góp ý quý báu của thầy để bài tiểu luận của chúng em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Hà Nội, ngày 12 tháng 04 năm 2024

Nhóm 12

MỤC LỤC

LỜI CẢM ƠN	2
MỤC LỤC	3
DANH MỤC HÌNH VẼ	5
DANH MỤC BẢNG BIỂU	6
MỞ ĐẦU	1
1. Lý do chọn chủ đề/đề tài.....	1
2. Mục tiêu nghiên cứu	1
3. Đối tượng và phạm vi nghiên cứu	1
4. Kết quả mong muốn đạt được của đề tài	2
5. Cấu trúc của báo cáo.....	2
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	3
1.1. Giới thiệu chung.....	3
1.2. Giới thiệu dự án phần mềm	3
1.3. Công cụ, kỹ thuật và phương pháp phát triển phần mềm.	4
1.3.1. Mô hình quy trình thác nước	4
1.3.2. Công cụ, kỹ thuật và phương pháp.	5
CHƯƠNG 2: PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM	9
2.1. Quy trình kỹ thuật yêu cầu.....	9
2.1.1. Giới thiệu quy trình kỹ thuật yêu cầu	9
2.1.2. Các hoạt động trong quy trình kỹ thuật yêu cầu	9
2.2. Đặc tả yêu cầu phần mềm Vinmec	10
2.2.1. Giới thiệu	10
2.2.2. Yêu cầu chức năng.....	11
2.2.3. Yêu cầu phi chức năng.....	29
2.2.4. Kết luận chương 2.....	31
CHƯƠNG 3 ĐẶC TẢ THIẾT KẾ PHẦN MỀM	32
3.1. Tổng quan về thiết kế phần mềm.....	32
3.1.1. Vai trò và các nhiệm vụ trong hoạt động thiết kế phần mềm	32
3.1.2. Chiến lược thiết kế phần mềm	33

3.1.3. Công cụ hỗ trợ thiết kế phần mềm.....	34
3.2. Tài liệu thiết kế phần mềm	35
3.2.1. Giới thiệu	35
3.2.2. Thiết kế kiến trúc phần mềm	36
3.2.3. Thiết kế cấu trúc phần mềm	40
3.2.4. Thiết kế dữ liệu	57
3.2.5. Mô hình cơ sở dữ liệu mức vật lý	62
3.2.5. Thiết kế giao diện người sử dụng	66
KẾT LUẬN	72
TÀI LIỆU THAM KHẢO	73

DANH MỤC HÌNH VẼ

Hình 1.1 Mô hình quy trình thác nước	5
Hình 1.2: Trình biên tập Visual Studio Code	8
Hình 1.3: Chương trình tạo máy chủ Xampp.....	8
Hình 2.1: Sơ đồ Use Case tổng quan	12
Hình 2.2. Sơ đồ phân rã Use Case	12
Hình 2.3: Biểu đồ minh họa Use Case đăng ký tài khoản khách hàng	13
Hình 2.4: Biểu đồ minh họa Use Case đăng nhập (Admin)	15
Hình 2.5: Biểu đồ minh họa Use Case quản lý tài khoản khách hàng	17
Hình 2.6: Biểu đồ minh họa Use Case đăng ký lịch khám	21
Hình 2.7: Biểu đồ minh họa Use Case quản lý đơn khám	23
Hình 2.8: Biểu đồ minh họa Use Case quản lý chuyên khoa.....	25
Hình 2.9: Biểu đồ minh họa Use Case quản lý phòng khám	27
Hình 3.1: Component diagram.....	37
Hình 3.2: Deployment diagram	38
Hình 3.3: Biểu đồ minh họa Use Case đăng nhập	40
Hình 3.4: Biểu đồ minh họa Use Case quản lý tài khoản	44
Hình 3.5: Biểu đồ minh họa Use Case đăng ký lịch khám	51
Hình 3.6: Biểu đồ minh họa Use Case quản lý chuyên khoa.....	54
Hình 3.7: Lớp thực thể người dùng	57
Hình 3.8: Lớp thực thể chuyên khoa	58
Hình 3.9: Lớp thực thể tài khoản	59
Hình 3.10: Lớp thực thể quản lý nhân sự	60
Hình 3.11: Lớp thực thể bài viết y tế	61
Hình 3.12: Lớp thực thể lịch khám	62
Hình 3.13: Giao diện màn hình đăng nhập	67
Hình 3.14: Giao diện màn hình đăng nhập (Rẽ nhánh)	68
Hình 3.15: Giao diện màn hình quản lý tài khoản	68
Hình 3.16: Giao diện màn hình xem lịch khám	69
Hình 3.17: Giao diện màn hình xem các chuyên khoa	69
Hình 3.15: Giao diện màn hình bảo trì chuyên khoa	70
Hình 3.16: Giao diện màn hình quản lý lịch khám	70

DANH MỤC BẢNG BIỂU

Bảng 2.1 Danh sách các tác nhân hệ thống.....	11
Bảng 2.2: Biểu đồ minh họa quy trình đăng ký tài khoản khách hàng.....	13
Bảng 2.3: Đặc tả chức năng đăng ký tài khoản khách hàng	14
Bảng 2.4: Biểu đồ minh họa quy trình đăng nhập (Admin)	15
Bảng 2.5: Đặc tả chức năng đăng nhập (Admin).....	16
Bảng 2.6: Biểu đồ minh họa quy trình xem danh sách tài khoản	17
Bảng 2.7: Biểu đồ minh họa quy trình thêm tài khoản	18
Bảng 2.8: Biểu đồ minh họa quy trình sửa tài khoản.....	18
Bảng 2.9: Biểu đồ minh họa quy trình xóa tài khoản	19
Bảng 2.10: Biểu đồ minh họa quy trình tìm kiếm tài khoản.....	19
Bảng 2.11: Đặc tả chức năng quản lý tài khoản khách hàng	20
Bảng 2.12: Biểu đồ minh họa quy trình đăng ký lịch khám	22
Bảng 2.13: Đặc tả chức năng đăng ký lịch khám	22
Bảng 2.14: Biểu đồ minh họa quy trình quản lý đơn khám	24
Bảng 2.15: Đặc tả chức năng quản lý đơn khám	24
Bảng 2.16: Biểu đồ minh họa quy trình quản lý chuyên khoa.....	26
Bảng 2.17: Đặc tả chức năng quản lý chuyên khoa.....	26
Bảng 2.18: Biểu đồ minh họa quy trình quản lý phòng khám	28
Bảng 2.19: Đặc tả Use Case quản lý phòng khám.....	28
Bảng 3.1: Biểu đồ trình tự chức năng đăng nhập	41
Bảng 3.2: Biểu đồ minh họa quy trình đăng nhập	42
Bảng 3.3: Biểu đồ lớp chức năng Đăng nhập	43
Bảng 3.4: Biểu đồ các lớp thực thể chức năng Đăng Nhập.....	43
Bảng 3.5: Biểu đồ trình tự chức năng quản lý tài khoản	45
Bảng 3.6: Biểu đồ minh họa quy trình xem danh sách tài khoản	46
Bảng 3.7: Biểu đồ minh họa quy trình thêm tài khoản	46
Bảng 3.8: Biểu đồ minh họa quy trình sửa tài kho	47
Bảng 3.9: Biểu đồ minh họa quy trình xóa tài khoản	47
Bảng 3.10: Biểu đồ minh họa quy trình tìm kiếm tài khoản.....	48

Bảng 3.11: Biểu đồ lớp chức năng quản lý tài khoản.....	48
Bảng 3.12: Biểu đồ các lớp thực thể chức năng quản lý tài khoản.....	49
Bảng 3.13: Biểu đồ trình tự chức năng đăng ký lịch khám	51
Bảng 3.14: Biểu đồ minh họa quy trình đăng ký lịch khám	52
Bảng 3.15: Biểu đồ lớp chức năng đăng ký lịch k.....	53
Bảng 3.16: Biểu đồ trình tự chức năng quản lý chuyên khoa.....	54
Bảng 3.17: Biểu đồ minh họa quy trình quản lý chuyên khoa.....	55
Bảng 3.18: Biểu đồ lớp chức năng quản lý chuyên khoa	56
Bảng 3.19: Tác giả.....	62
Bảng 3.20: Bài viết y tế	62
Bảng 3.21: Diễn đàn	63
Bảng 3.22: Bác sĩ.....	63
Bảng 3.23: Bệnh vi	63
Bảng 3.24: Quản lý nhân sự.....	63
Bảng 3.25: Chuyên khoa.....	64
Bảng 3.26: Dịch vụ chăm sóc	64
Bảng 3.27: Quản lý khoa	64
Bảng 3.28: Dịch vụ bệnh viện	64
Bảng 3.29: Lịch khám.....	65
Bảng 3.3: Bảng người dùng	65
Bảng 3.31: Bảng tài khoản.....	65
Bảng 3.32: Biểu đồ thực thể liên kết mức vật lý	66
Bảng 3.33: Mô tả tổng quan giao diện người dùng	66

MỞ ĐẦU

1. Lý do chọn chủ đề/đề tài

Với sự phát triển vượt bậc và ứng dụng rộng rãi của Công Nghệ Thông Tin vào đời sống ngày nay, lĩnh vực về y tế đã được cải thiện rất nhiều. Tuy nhiên vẫn phải đối mặt với nhiều vấn đề về mất mát thông tin, khó truy cập, khả năng bảo mật kém,... Việc sử dụng hồ sơ bệnh nhân bằng giấy đã trở nên lỗi thời, lạc hậu.

Mô hình Waterfall, với sự phân chia quy trình làm việc thành từng phần rõ ràng và tuần tự, đã được sử dụng trong rất nhiều lĩnh vực để quản lý một dự án một cách có hệ thống và hiệu quả (như: xây dựng, thiết kế, quản lý dự án, sản xuất và chuỗi cung ứng, ...). Đặc biệt, việc áp dụng mô hình này vào trong lĩnh vực y tế, cụ thể là trong việc đặt lịch khám đòi hỏi một sự hiểu biết sâu rộng, từ đó sẽ cho chúng em cơ hội để học hỏi với một dự án thực tế, rút ra bài học kinh nghiệm.

Để làm rõ hơn về vấn đề này, nên nhóm chúng em lựa chọn đề tài “Nghiên cứu mô hình thác nước và ứng dụng trong phần mềm thăm khám y tế Vinmec”, với mong muốn sau khi kết thúc học phần “Nhập môn công nghệ phần mềm” (IT6082), chúng em có thể thiết kế một phần mềm có thể quản lý hồ sơ bệnh án. Giúp ích cho các bệnh nhân không cần phải làm những thủ tục gian nan mà vẫn có thể đặt trước lịch khám, xem bệnh án của mình trực tiếp qua Internet, cũng như tăng tính tiện lợi, truy cập dữ liệu với các y bác sĩ.

2. Mục tiêu nghiên cứu

Trong thời buổi công nghệ lên ngôi, chúng ta thường có sự ưu tiên hơn với một thiết bị thông minh như laptop, điện thoại để phục vụ nhu cầu hằng ngày. Do đó, chúng em mong muốn phát triển một hệ thống đặt lịch hẹn với bác sĩ thông qua ứng dụng “Vinmec”, qua đó giúp người dùng có thể dễ dàng hơn so với việc đặt một lịch khám trực tiếp tại bệnh viện.

3. Đối tượng và phạm vi nghiên cứu

Sẽ có 3 đối tượng sử dụng: bệnh nhân, bác sĩ và quản trị viên:

- Bệnh nhân: Có thể đặt lịch hẹn khám với bác sĩ của mình. Trong giao diện người dùng, bệnh nhân có thể chọn các bác sĩ có sẵn lịch khám vào một ngày cụ thể. Họ có thể đặt một thời gian cụ thể phù hợp với mình. Dashboard (Bảng điều khiển) cho biết bệnh nhân đã đến gặp bác sĩ hay chưa. Bệnh nhân có thể cập nhật được hồ sơ của mình.
- Quản trị viên: Có thể tạo và cập nhật thông tin của bác sĩ. Quản trị viên có thể lọc các bệnh nhân và xem ai đã đặt lịch hẹn vào một ngày cụ thể và chuyển đổi trạng thái của bệnh nhân là đã khám / không đến khám. Quản trị viên có thể tạo các phòng ban, cập nhật và xóa chúng. Quản trị viên cũng có thể lấy thông tin về hệ thống trên trang dashboard của mình (gồm: số lượng bác sĩ đang làm việc, phòng ban, bệnh nhân đã đặt lịch hẹn,...)
- Bác sĩ: Có thể tạo lịch hẹn cho một ngày cụ thể, kiểm tra số lượng bệnh nhân đã đặt lịch hẹn bằng cách lọc qua ngày đó. Ngoài ra họ có còn thể kê đơn thuốc cho bệnh nhân sau khi khám.

4. Kết quả mong muốn đạt được của đề tài

- Hiểu rõ về quy trình phát triển phần mềm và cách áp dụng nó vào một dự án cụ thể (Quản lý thăm khám y tế trên website vinmec.com)
- Có thể cung cấp phân tích chi tiết về các bước của một quy trình phát triển phần mềm cụ thể.
- Có những kiến thức thực tiễn để đề xuất các chiến lược triển khai mô hình phát triển và phần mềm.

5. Cấu trúc của báo cáo

Ngoài phần Mở đầu và Kết luận, nội dung quyền báo cáo còn bao gồm 3 chương:

- Chương 1: Tổng quan về đề tài
- Chương 2: Phân tích và đặc tả yêu cầu phần mềm.
- Chương 3: Đặc tả thiết kế phần mềm.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu chung

Bệnh viện đa khoa quốc tế Vinmec một là bệnh viện tư nhân đa ngành tại Việt Nam, có trụ sở chính đặt tại Phường Vĩnh Tuy, Quận Hai Bà Trưng, Hà Nội, đảm bảo chuyên môn và cơ sở hạ tầng để đảm nhận chức năng thăm khám tại địa phương.

Để linh hoạt, thuận tiện trong việc khám, chữa của bệnh nhân, bệnh viện đã có ý tưởng xây dựng một phần mềm thăm khám y tế trực tuyến trên nền tảng ứng dụng (Được gọi là Vinmec). Ngoài ra, hệ thống này còn cho phép bệnh nhân có thể đăng ký thăm khám thông qua ứng dụng web. Điều này giúp dễ dàng hơn cho bệnh nhân trong việc khai báo thông tin, lấy số chờ gọi tên truyền thống, và thuận tiện cho người quản lý trong việc sắp xếp hồ sơ, đặt lịch hẹn. Từ đó giúp bệnh viện giảm bớt đi những quy trình thừa thãi, tăng hiệu suất làm việc, độ chính xác, giảm số lượng giấy tờ và khiến bệnh viên trở thành một cơ sở y tế mạnh mẽ, nổi bật trên thị trường.

1.2. Giới thiệu dự án phần mềm

- Ngày nay, mỗi con người chúng ta đều có nhu cầu sở khám/chữa bệnh khác nhau nhưng hiện nay việc xếp hàng chờ khám bệnh tại các cơ sở y tế hiện nay là khá khó khăn với cả bệnh nhân và bác sĩ. Hiểu được những trở ngại đó mà các ứng dụng đặt lịch hẹn trực tuyến ra đời nhằm giúp người bệnh có thể đăng ký giờ khám, lựa chọn bác sĩ và thanh toán một cách nhanh chóng, hiệu quả.
- Y tế là một lĩnh vực rất thiết yếu đối với đời sống con người ở bất kỳ quốc gia nào. Thế nhưng ở Việt Nam, đất nước 100 triệu dân không có quá nhiều dự án về y tế, bởi đây được coi là ngành đặc thù, nhạy cảm, ảnh hưởng trực tiếp đến sức khỏe con người, có nhiều thách thức.
- Một thực tế khác là có không ít những bệnh viện và phòng khám đang phải giải quyết một bài toán về quá tải bệnh nhân xếp hàng chờ khám. Đứng từ góc độ một bệnh nhân thì bài toán đó cũng gây ra không ít phiền

toái và lãng phí không đáng có. Tổng hợp lại những vấn đề mà các bệnh viện, cơ sở y tế, bác sĩ và bệnh nhân đang gặp phải:

- Các bệnh viện và cơ sở y tế nhiều nơi đang trong tình trạng quá tải nhất là giờ cao điểm.
 - Bác sĩ không chủ động được lịch khám.
 - Bệnh nhân phải chờ và xếp hàng rất lâu để có thể đăng ký khám.
 - Bệnh nhân bị động trong việc sắp xếp quỹ thời gian của mình cho việc đi khám, chữa bệnh.
 - Lãng phí thời gian, tiền bạc, công sức của xã hội
 - Bệnh nhân không được biết thông tin về bác sĩ sẽ khám cho mình.
- Thay vì việc phải xếp hàng chờ đợi để tới lượt khám chữa bệnh thì công nghệ ngày nay có thể giúp chúng ta dễ dàng kết nối với bác sĩ thông qua việc đặt lịch khám trực tuyến và có thể kết nối được với bác sĩ đúng chuyên khoa chúng ta cần. Chúng em tin rằng bệnh nhân được gặp đúng thầy, đúng thuốc, đúng phương pháp sẽ mang lại lợi ích cho người bệnh tốt hơn, không chỉ là việc giảm thời gian chờ khám. Đó cũng chính là ý tưởng bắt nguồn để nhóm em quyết định thực hiện đề tài “Xây dựng ứng dụng thăm khám y tế Vinmec”.

1.3. Công cụ, kỹ thuật và phương pháp phát triển phần mềm.

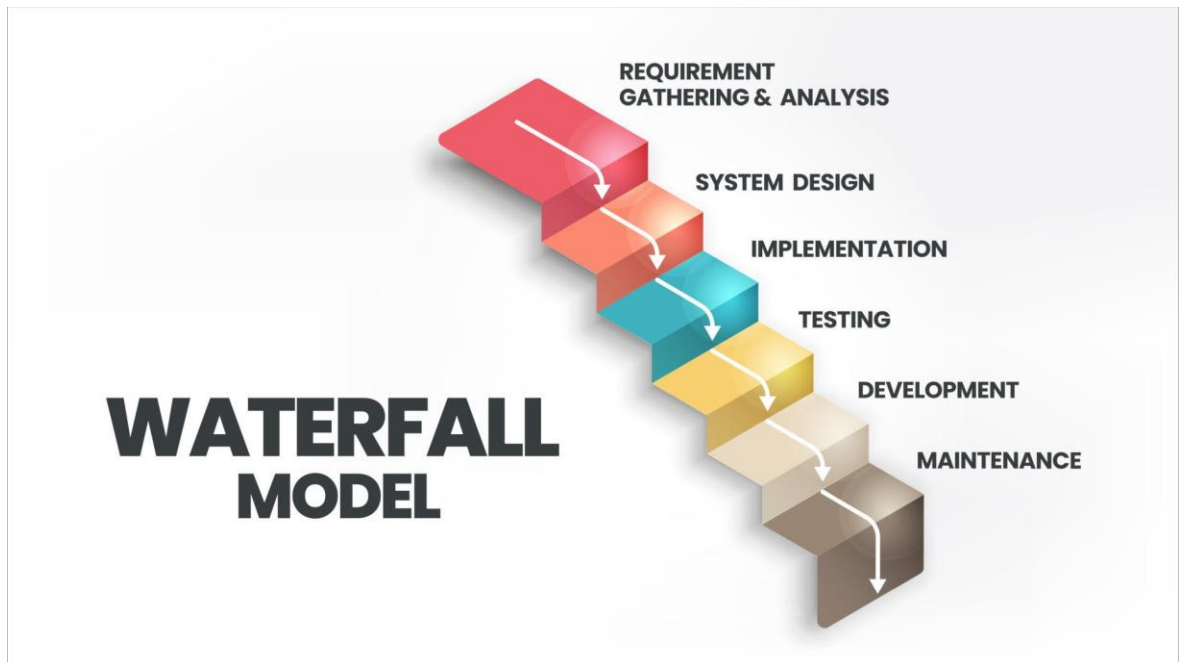
1.3.1. Mô hình quy trình thác nước

Mô hình thác nước (Waterfall model) là mô hình vòng đời tuần tự tuyến tính. Waterfall là mô hình quy trình đầu tiên và nổi tiếng nhất được giới thiệu, với cách tiếp cận Vòng đời phát triển hệ thống (SDLC) lâu đời nhất, được sử dụng để phát triển phần mềm.

Mô hình thác nước minh họa quá trình phát triển phần mềm theo dòng

tuần tự tuyến tính. Điều này có nghĩa là bất kỳ giai đoạn nào trong quá trình phát triển chỉ bắt đầu nếu giai đoạn trước đó hoàn thành. Trong mô hình thác nước, các pha không chồng lên nhau.

- Thu thập và phân tích yêu cầu (Requirement Analysis)
- Thiết kế hệ thống (System Design)
- Thực hiện (Implementation)
- Tích hợp và Kiểm thử (Integration and Testing)
- Triển khai hệ thống (Deployment of system)
- Bảo trì (Maintenance)



Hình 1.1 Mô hình quy trình thác nước

1.3.2. Công cụ, kỹ thuật và phương pháp.

1.3.2.1. Công nghệ sử dụng

(1) Bootstrap 4

- Bootstrap là một framework front-end miễn phí giúp phát triển các ứng dụng web nhanh và dễ dàng hơn. Bootstrap bao gồm các mẫu thiết kế dựa trên HTML và CSS như typography, form, button, table, navigation,

modal, image carousels cũng như các plugins JavaScript tùy ý. Cung cấp khả năng thiết kế web responsive một cách dễ dàng.

- Ưu điểm Bootstrap 4:
 - Dễ dàng sử dụng: Chỉ cần lượng kiến thức cơ bản về HTML và CSS, bất cứ ai cũng có thể sử dụng Bootstrap.
 - Các tính năng đáp ứng: Responsive CSS của Bootstrap điều chỉnh cho điện thoại, máy tính bảng và máy tính để bàn.
 - Cách tiếp cận Mobile-first: Trong Bootstrap, mobile-first styles là một phần của core framework.
 - Khả năng tương thích trình duyệt: Bootstrap 4 tương thích với tất cả các trình duyệt hiện nay (Chrome, Firefox, Internet Explorer 10+, Edge, Safari và Opera)

(2) Reactjs

- ReactJS đã được cải tiến và phát triển qua nhiều phiên bản khác nhau kể từ khi nó được giới thiệu lần đầu vào năm 2013. Trong quá trình phát triển, ReactJS đã giới thiệu các tính năng mới như React Hooks, React Router, Redux, Next.js, ... để giúp người lập trình xây dựng các ứng dụng web phức tạp một cách dễ dàng hơn.
- Ngoài ra, ReactJS cũng có một cộng đồng lớn, hoạt động tích cực trên các diễn đàn, blog và các trang web chia sẻ kiến thức về lĩnh vực phát triển web. Cộng đồng này liên tục đóng góp ý tưởng, công cụ và các giải pháp mới để hỗ trợ cho việc phát triển ứng dụng ReactJS.
- Với những lợi thế trên, hiện nay ReactJS đã trở thành một công cụ phát triển web rất phổ biến được sử dụng để xây dựng các ứng dụng web phức tạp, từ các trang thương mại điện tử cho đến các ứng dụng xã hội, giúp cho việc phát triển ứng dụng web nhanh hơn và dễ dàng hơn.

(3) Express

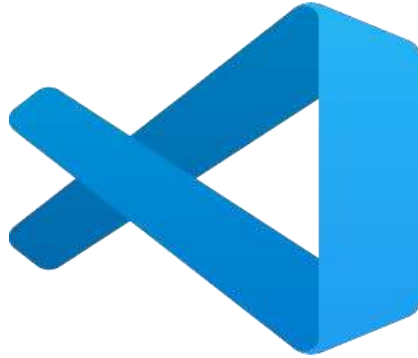
- Express là một framework cho Node.js, được sử dụng để xây dựng các ứng dụng web và API RESTful. Nó giúp xây dựng các ứng dụng Node.js nhanh chóng và dễ dàng hơn bằng cách cung cấp các chức năng tiện ích để xử lý các yêu cầu HTTP, quản lý định tuyến, phân tích và xử lý yêu cầu từ phía người dùng và nhiều chức năng khác.
- Express có thể được sử dụng để tạo ra các ứng dụng web đơn giản hoặc phức tạp, bao gồm các ứng dụng đa người dùng, API RESTful, ứng dụng thời gian thực, ứng dụng chat, v.v... Là một trong những framework phổ biến nhất cho Node.js và được sử dụng rộng rãi trong cộng đồng phát triển web.
- Một số tính năng của Express bao gồm:
 - Hỗ trợ định tuyến: Express cung cấp các phương thức để quản lý các tuyến đường (route) trong ứng dụng web.
 - Có nhiều middleware được tích hợp sẵn giúp xử lý các yêu cầu HTTP, xử lý và phân tích các thông tin từ phía người dùng.
 - Hỗ trợ nhiều template engine như Pug, EJS, Handlebars giúp tạo ra các trang web động với dữ liệu được tải từ máy chủ.
 - Xử lý tĩnh và động: Express có khả năng xử lý các file tĩnh (static files) như CSS, JavaScript, ảnh,.. cũng như tạo ra các trang web động với dữ liệu được tải từ máy chủ.
 - Phát triển API RESTful: Express hỗ trợ phát triển các API RESTful để giao tiếp giữa các ứng dụng.

1.3.2.2. Công cụ sử dụng

(1) Visual Studio Code

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông

minh, snippets, và cải tiến mã nguồn.



Hình 1.2: Trình biên tập Visual Studio Code

(2) Xampp – PhpMyAdmin

Xampp là chương trình tạo máy chủ Web được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Không như Appserv, Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.



Hình 1.3: Chương trình tạo máy chủ Xampp

CHƯƠNG 2: PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM

2.1. Quy trình kỹ thuật yêu cầu

2.1.1. Giới thiệu quy trình kỹ thuật yêu cầu

Quy trình kỹ thuật yêu cầu (requirement engineering process) là một phần quan trọng trong phát triển phần mềm và hệ thống. Nó bao gồm các bước để xác định, ghi nhận, và quản lý các yêu cầu từ khách hàng và các bên liên quan để đảm bảo rằng sản phẩm cuối cùng đáp ứng được các nhu cầu và mong đợi đặt ra.

2.1.2. Các hoạt động trong quy trình kỹ thuật yêu cầu

Thu thập yêu cầu:

- **Khảo sát và Phỏng vấn:** Gặp gỡ các bên liên quan để hiểu nhu cầu và mong đợi của họ.
- **Phiên thảo luận (Workshops):** Tổ chức các buổi làm việc nhóm để thu thập yêu cầu.
- **Quan sát:** Theo dõi cách mà người dùng hiện tại tương tác với hệ thống để phát hiện các yêu cầu ngầm.
- **Tài liệu hiện có:** Xem xét các tài liệu hiện có để lấy thông tin về yêu cầu.

Phân tích yêu cầu:

- **Xác định tính khả thi:** Đánh giá tính khả thi của các yêu cầu đã thu thập.
- **Phân loại và ưu tiên yêu cầu:** Sắp xếp các yêu cầu theo mức độ quan trọng và ưu tiên.
- **Làm rõ và làm chi tiết yêu cầu:** Làm rõ những yêu cầu còn mơ hồ và chi tiết hóa chúng.

Đặc tả yêu cầu:

- **Tạo tài liệu yêu cầu:** Viết tài liệu mô tả chi tiết các yêu cầu hệ thống.
- **Sử dụng các công cụ mô hình hóa:** Dùng các biểu đồ và mô hình như UML (Unified Modeling Language) để biểu diễn yêu cầu một cách trực quan.

Xác minh và Thẩm định yêu cầu:

- Xem xét tài liệu yêu cầu: Đảm bảo rằng tài liệu yêu cầu chính xác và đầy đủ.
- Thẩm định yêu cầu với các bên liên quan: Đảm bảo rằng các yêu cầu được chấp nhận và đồng thuận bởi các bên liên quan.
- Kiểm tra yêu cầu: Đảm bảo có thể kiểm tra được và không có mâu thuẫn.

Quản lý yêu cầu:

- Theo dõi yêu cầu: Giám sát sự thay đổi của các yêu cầu trong suốt vòng đời dự án.
- Quản lý thay đổi yêu cầu: Đưa ra quy trình quản lý và xử lý các thay đổi về yêu cầu.
- Ghi nhận và lưu trữ yêu cầu: Lưu trữ các yêu cầu và tài liệu liên quan để dễ dàng truy xuất và tham chiếu sau này.

2.2. Đặc tả yêu cầu phần mềm Vinmec

2.2.1. Giới thiệu

2.2.1.1. Mục đích tài liệu

Tài liệu này mô tả chi tiết về thiết kế phần mềm Vinmec phục vụ hoạt động đặt lịch khám trực tuyến của bệnh nhân.

2.2.1.2. Phạm vi tài liệu

Tài liệu này là cơ sở giao tiếp của các thành viên trong đội phát triển dự án phần mềm Vinmec và cũng là căn cứ trong hoạt động kiểm thử, vận hành và bảo trì phần mềm.

2.2.1.3. Tài liệu tham khảo

- Giáo trình Nhập môn công nghệ phần mềm ĐHCNHN(2016) - Hoàng Quang Huy, Phùng Đức Hoà, Trịnh Bá Quý
- <https://www.studocu.com/vn/course/truong-dai-hoc-cong-nghiep-ha-noi/nhap-mon-cong-nghe-phan-mem/5274241>

2.2.1.4. Bố cục tài liệu thiết kế phần mềm

Ngoài phần giới thiệu, cấu trúc Tài liệu đặc tả yêu cầu phần mềm Vinmec còn gồm các nội dung chính sau:

- Phần 1: Trình bày tổng quan về tài liệu đặc tả yêu cầu phần mềm.
- Phần 2 Trình bày chi tiết về các yêu cầu chức năng của phần mềm, bao gồm yêu cầu chức nghiệp vụ và yêu cầu chức năng hệ thống.
- Phần 3: Trình bày chi tiết về các yêu cầu phi chức năng của phần mềm.

2.2.2. Yêu cầu chức năng

2.2.2.1. Các tác nhân hệ thống

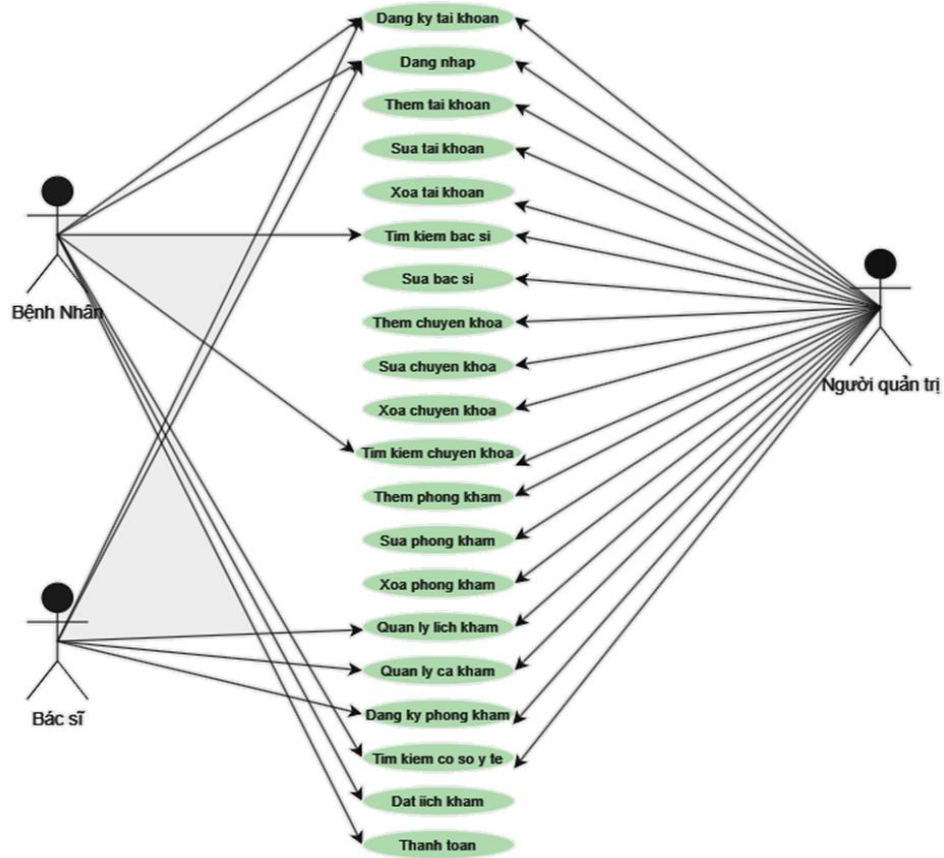
Các tác nhân hệ thống là những đối tượng tương tác với hệ thống, bao gồm người dùng và các hệ thống khác. Dưới đây là các tác nhân chính trong hệ thống

Bảng 2.1 Danh sách các tác nhân hệ thống

STT	Tác nhân	Chức năng
1	Admin	Quản lý toàn bộ website
2	Bác sĩ	Đăng ký các ca khám bệnh, quản lý kế hoạch khám bệnh
3	Bệnh nhân	Đặt lịch hẹn bác sĩ, xem thông tin tất cả bác sĩ, xem chuyên khoa, thông tin dịch vụ,...

2.2.2.2. Các yêu cầu chức năng

2.2.2.2.1. Sơ đồ Use Case tổng quan



Hình 2.1: Sơ đồ Use Case tổng quan

2.2.2.2.2. Sơ đồ phân rã Use Case

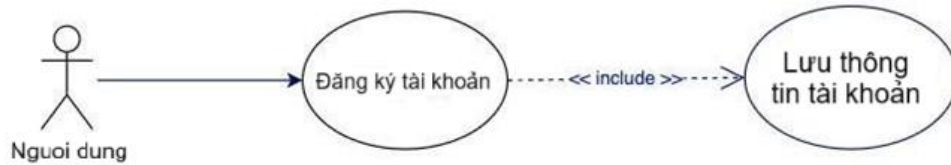


Hình 2.2. Sơ đồ phân rã Use Case

2.2.2.2.3. Quy trình sử dụng phần mềm

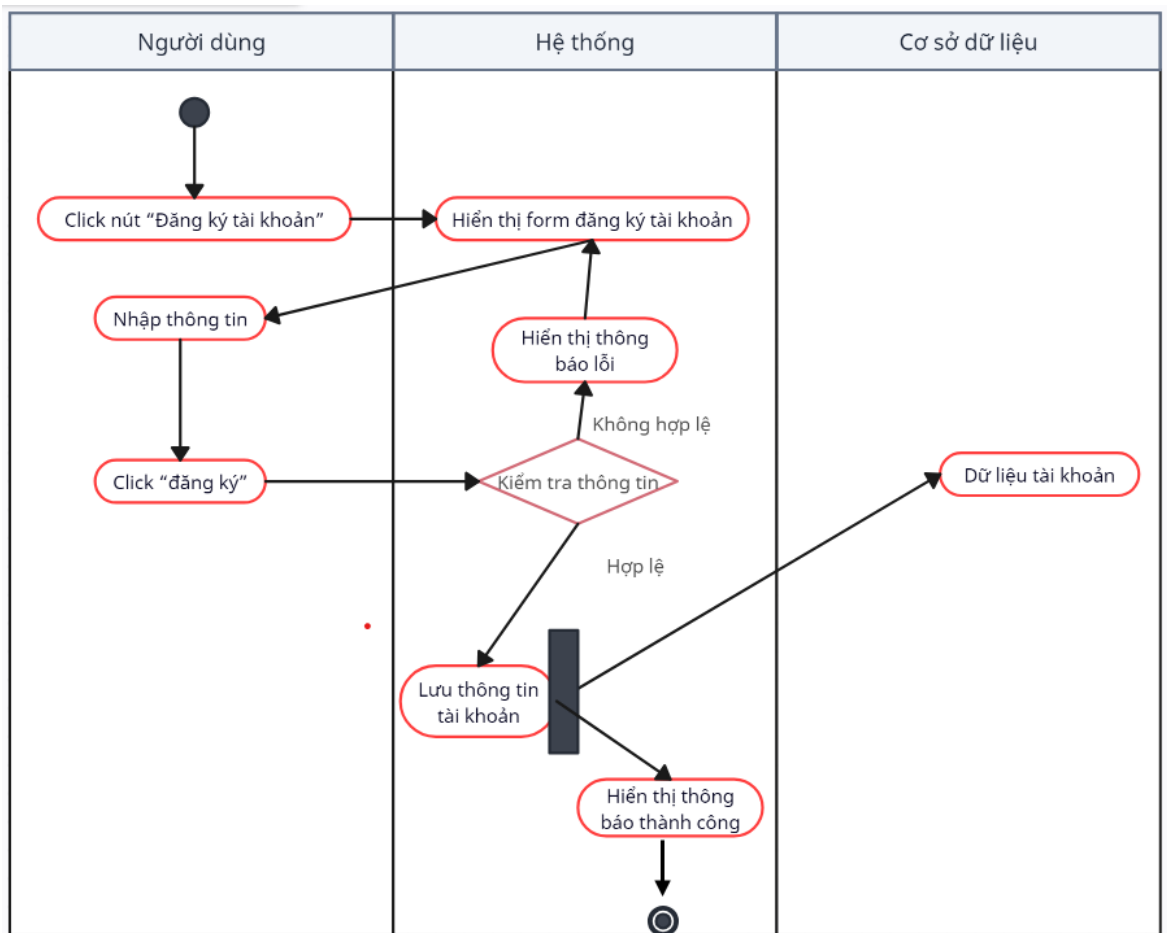
(1) Quy trình đăng ký tài khoản khách hàng

Use case này cho phép khách hàng tạo tài khoản người dùng để đăng nhập vào ứng dụng thăm khám y tế Vinmec.



Hình 2.3: Biểu đồ minh họa Use Case đăng ký tài khoản khách hàng

Bảng 2.2: Biểu đồ minh họa quy trình đăng ký tài khoản khách hàng

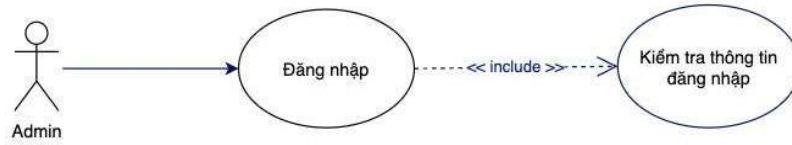


Bảng 2.3: Đặc tả chức năng đăng ký tài khoản khách hàng

Mã use case: F1	Đăng ký tài khoản		
Tác nhân	Người sử dụng		
Mô tả	Cho phép người dùng đăng ký tài khoản		
Sự kiện kích hoạt	Use case bắt đầu khi người dùng bấm vào nút “Đăng ký” ở trang chủ làm việc. Hệ thống sẽ hiển thị form đăng ký tài khoản. Sau đó người dùng nhập các thông tin cần thiết.		
Tiền điều kiện	Không		
Luồng sự kiện	#	Thực hiện bởi	Hành động
	1	Hệ thống	Đưa ra giao diện đăng ký
	2	Người dùng	Nhập thông tin
	3	Người dùng	Nhấn nút đăng ký
	4	Hệ thống	Nhận xử lý thông tin,
	5	Hệ thống	Nếu thông tin hợp lệ, đưa ra thông báo thành công, đưa đến trang chính
	6	Hệ thống	Nếu thông tin không hợp lệ, đưa ra thông báo và yêu cầu nhập lại
Luồng sự kiện rẽ nhánh	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc		
Hậu điều kiện	Không		

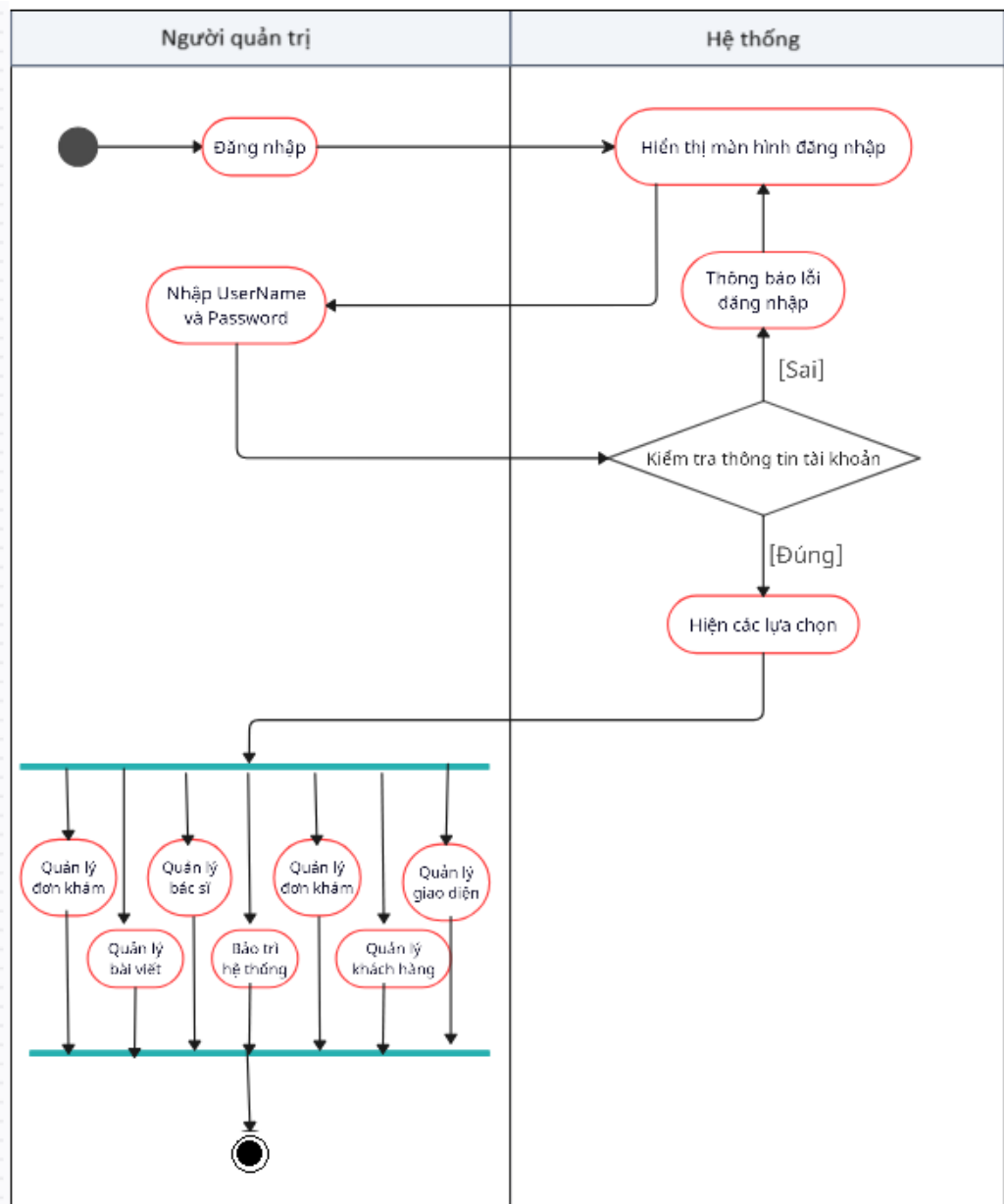
(2) Quy trình đăng nhập (Admin)

Use case này cho phép người quản trị đăng nhập vào hệ thống để thực hiện các chức năng quản lý, bảo trì.



Hình 2.4: Biểu đồ minh họa Use Case đăng nhập (Admin)

Bảng 2.4: Biểu đồ minh họa quy trình đăng nhập (Admin)

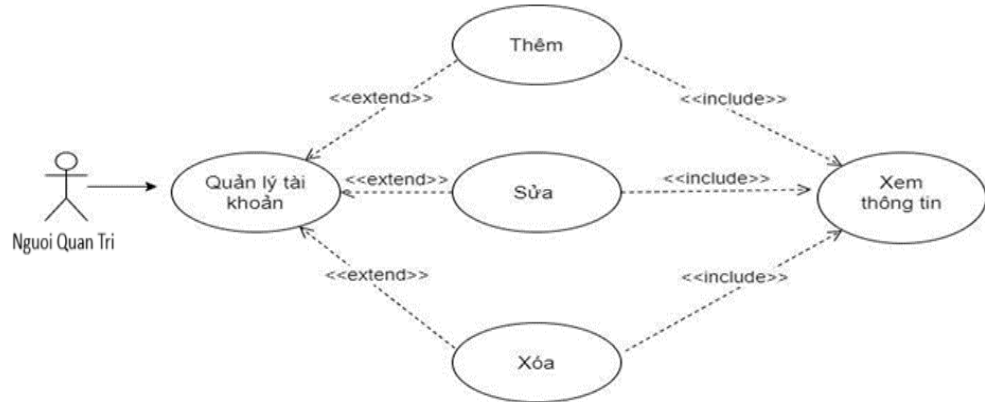


Bảng 2.5: Đặc tả chức năng đăng nhập (Admin)

Mã use case: F2	Tên UC: Đăng nhập		
Tác nhân	Người quản trị		
Mô tả	Cho phép người dùng truy cập vào hệ thống		
Sự kiện kích hoạt chức năng	Use case bắt đầu khi người dùng bấm vào nút đăng nhập ở trang chủ làm việc. Hệ thống sẽ hiển thị lên màn hình form đăng nhập, sau đó người dùng nhập thông tin tài khoản, mật khẩu.		
Tiền điều kiện	Đã có tài khoản quản trị viên trước đó		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Người quản trị	Chọn chức năng đăng nhập
	2	Hệ thống	Hiển thị giao diện đăng nhập
	3	Người quản trị	Nhập tài khoản, mật khẩu vào giao diện đăng nhập
	4	Hệ thống	Xử lý thông tin. Nếu nhập đúng thì hệ thống sẽ chuyển tới trang quản lý.
Luồng sự kiện rẽ nhánh	<ul style="list-style-type: none"> Tại bước 3, nếu người quản trị nhập sai tài khoản hoặc mật khẩu, hệ thống sẽ thông báo đăng nhập không thành công và quay lại bước 2. Nếu người quản trị chọn quay lại, use case kết thúc. Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc 		
Hậu điều kiện	Không		

(3) Quy trình quản lý tài khoản khách hàng

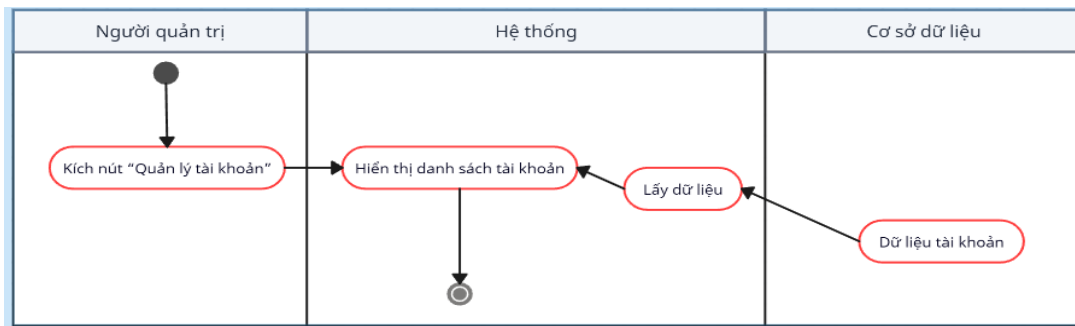
Use case này cho phép người quản trị thêm tài khoản, xóa tài khoản, sửa thông tin tài khoản, và tìm kiếm tài khoản khách hàng.



Hình 2.5: Biểu đồ minh họa Use Case quản lý tài khoản khách hàng

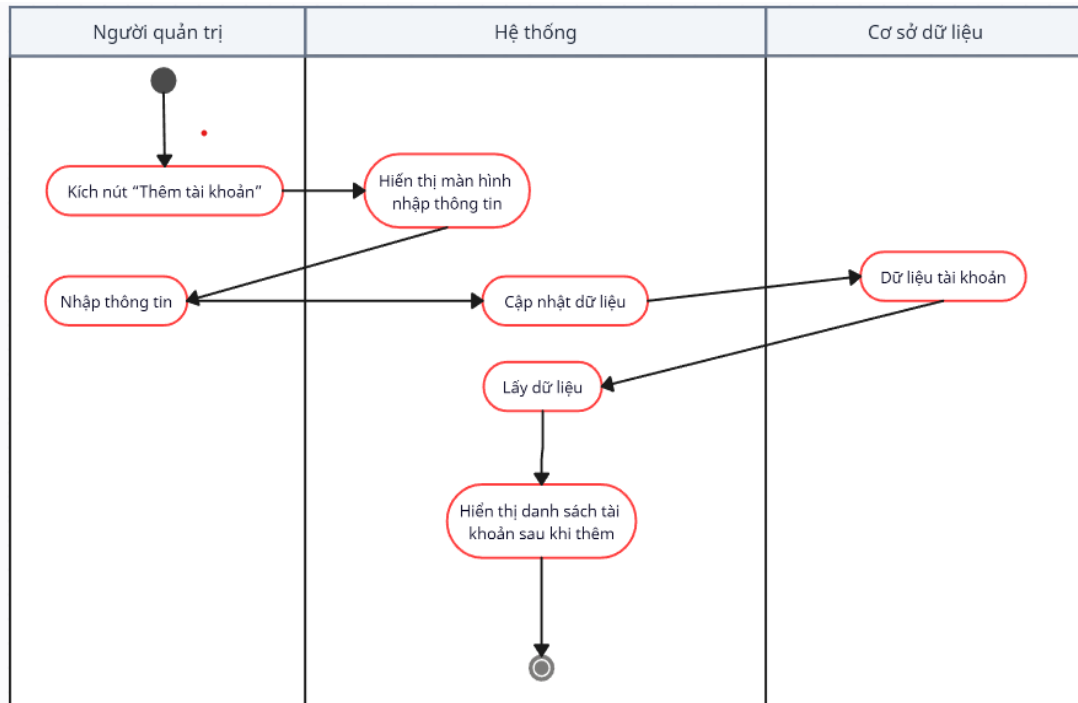
2.3.5.1. Xem danh sách tài khoản

Bảng 2.6: Biểu đồ minh họa quy trình xem danh sách tài khoản



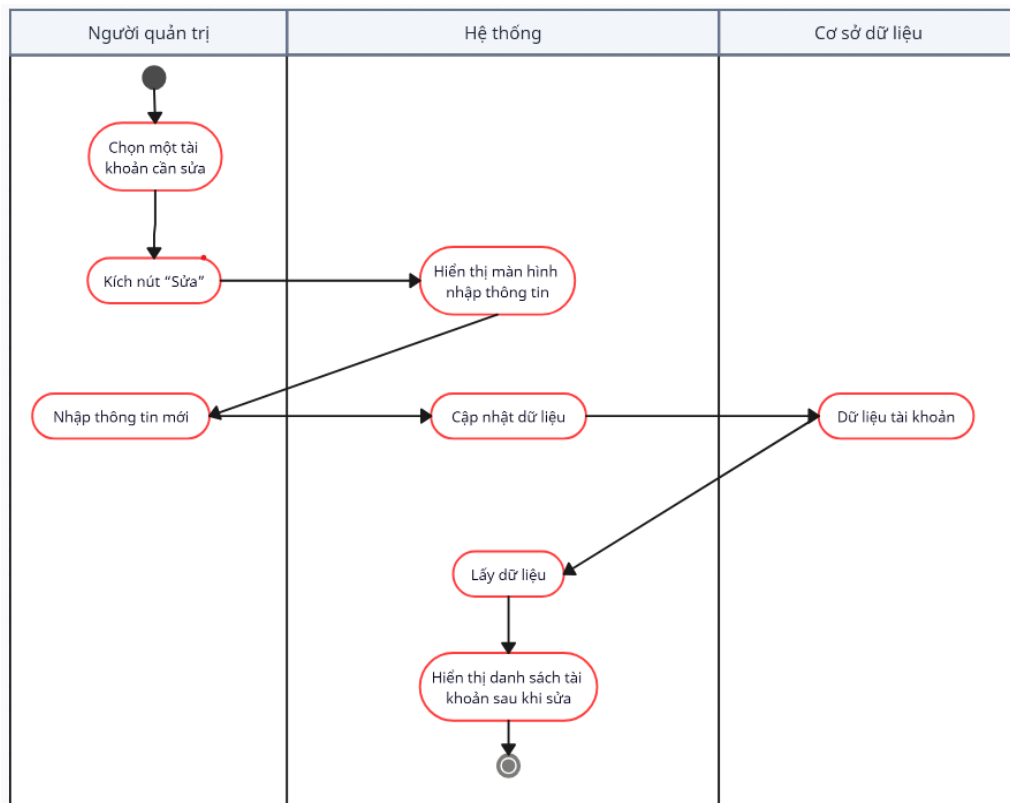
2.3.5.2. Thêm tài khoản khách hàng

Bảng 2.7: Biểu đồ minh họa quy trình thêm tài khoản



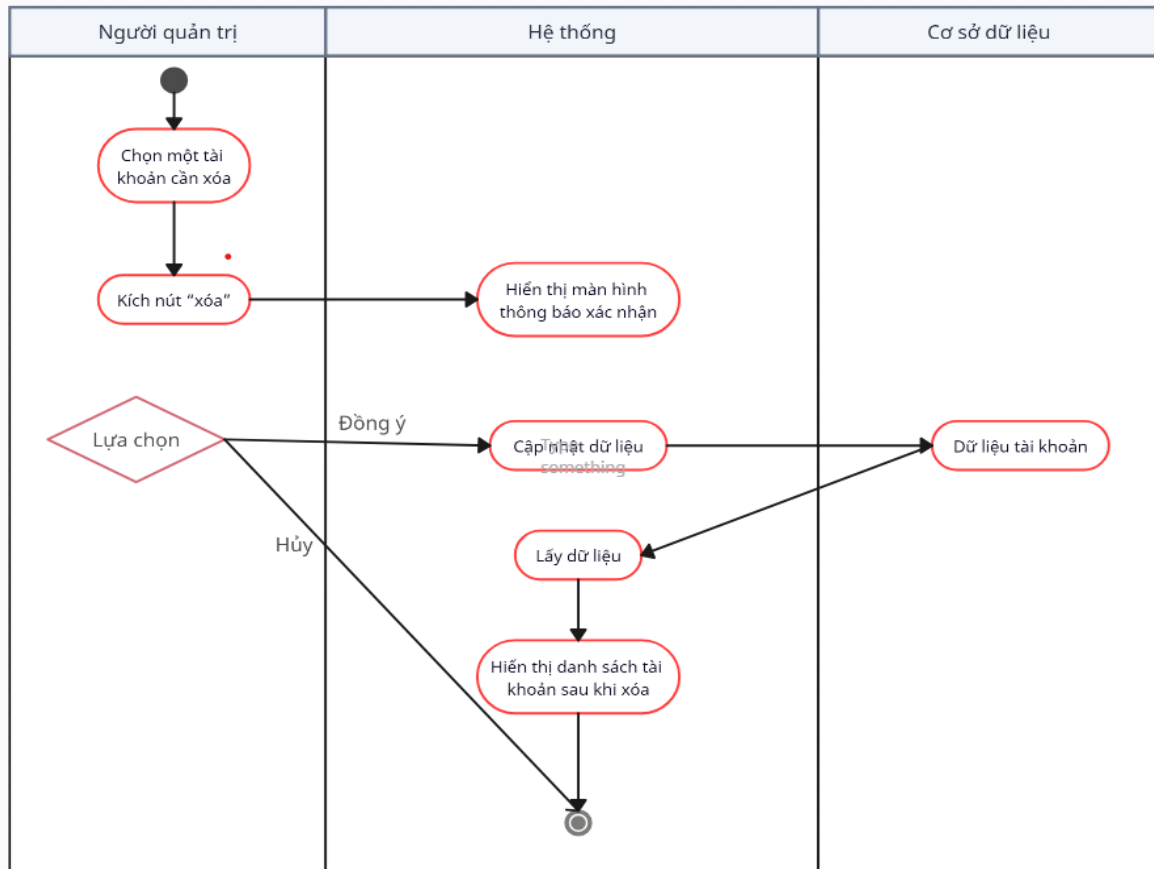
2.3.5.3. Sửa tài khoản khách hàng

Bảng 2.8: Biểu đồ minh họa quy trình sửa tài khoản



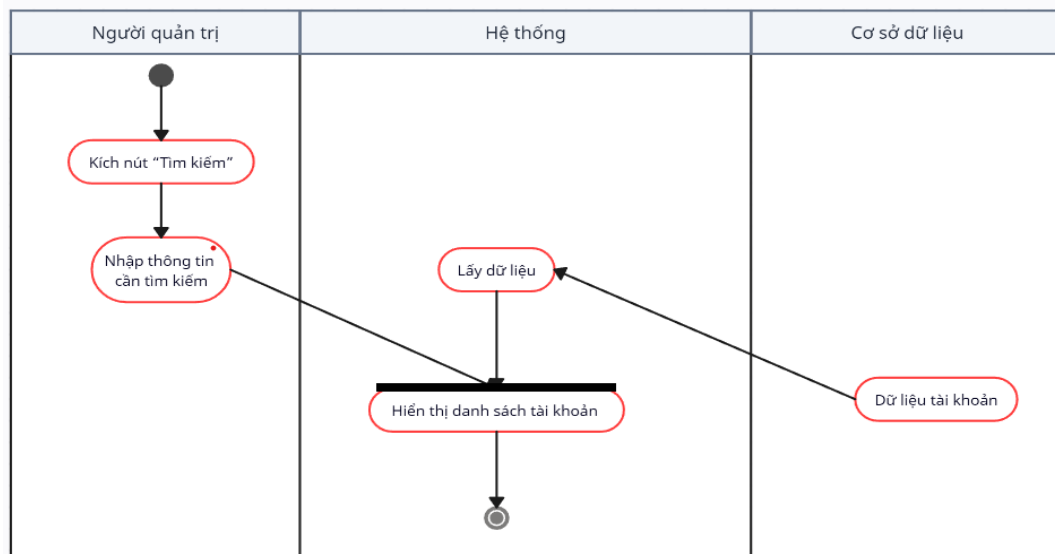
2.3.5.4. Xóa tài khoản khách hàng

Bảng 2.9: Biểu đồ minh họa quy trình xóa tài khoản



2.3.5.5. Tìm kiếm tài khoản

Bảng 2.10: Biểu đồ minh họa quy trình tìm kiếm tài khoản



*Chú ý: Các quy trình Thêm, sửa, xóa, tìm kiếm chỉ có thể được sử dụng khi người quản trị đã kích vào nút “Quản lý tài khoản”.

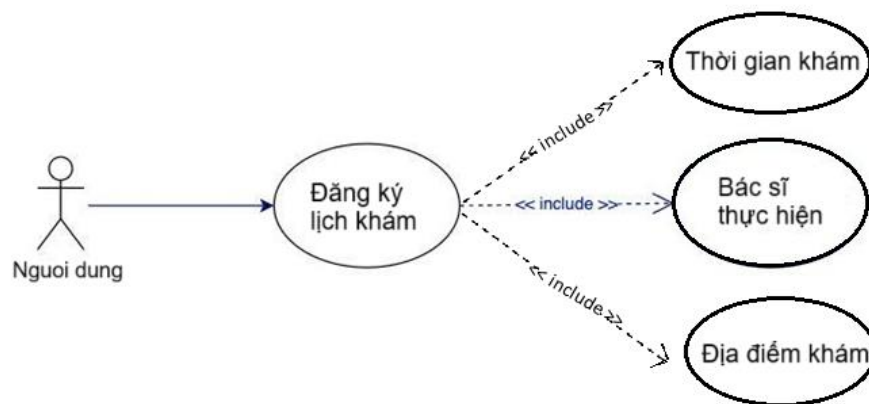
Bảng 2.11: Đặc tả chức năng quản lý tài khoản khách hàng

Mã UC: F3	Tên Use Case: Quản lý tài khoản khách hàng		
Tác nhân	Người quản trị		
Mô tả	Use Case này cho phép người quản trị quản lý tài khoản khách hàng		
Sự kiện kích hoạt chức năng	Use Case bắt đầu khi người quản trị click vào “Quản lý tài khoản” . Hệ thống sẽ lấy thông tin khách hàng và hiển thị lên màn hình		
Tiền điều kiện	Đăng nhập vào hệ thống với quyền quản trị		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Người quản trị	Click vào thông tin khách hàng
	2	Hệ thống	Chuyển sang thông tin chi tiết khách hàng
	3	Người quản trị	<ul style="list-style-type: none"> Thêm người dùng vào hệ thống bằng cách điền thông tin như email , số điện thoại Sửa đổi thông tin người dùng bằng cách sửa đổi các thông tin liên quan Xóa tài khoản người dùng khỏi hệ thống Tìm kiếm người dùng theo tên , email
	4	Hệ thống	Hiển thị form xác nhận
	5	Người quản trị	Nhấn xác nhận
	6	Hệ thống	Xác nhận thông tin, lưu vào hệ thống, trở về trang chủ
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động

	1	Người quản trị	Nếu người quản trị nhập thông tin không hợp lệ hoặc thiếu thông tin bắt buộc, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại thông tin.
	2	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc
Hậu điều kiện	Nếu Use case kết thúc thành công thì thông tin được cập nhật sẽ được lưu trong cơ sở dữ liệu		

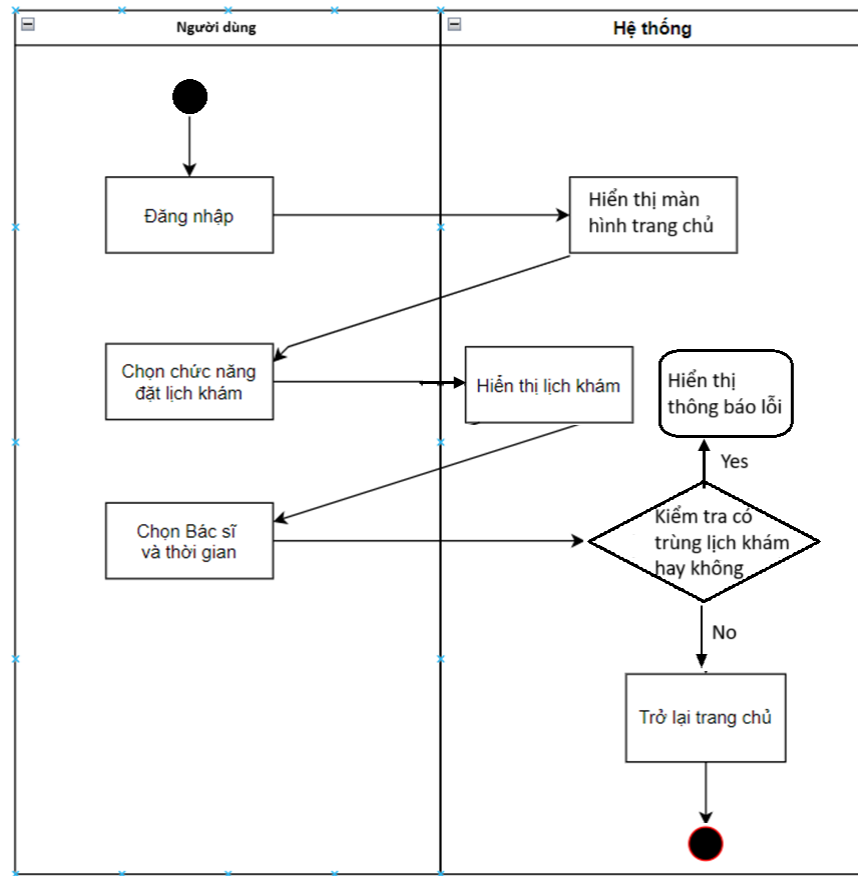
(4) Quy trình đăng ký lịch khám

Use case này cho phép bệnh nhân đặt lịch khám bệnh.



Hình 2.6: Biểu đồ minh họa Use Case đăng ký lịch khám

Bảng 2.12: Biểu đồ minh họa quy trình đăng ký lịch khám



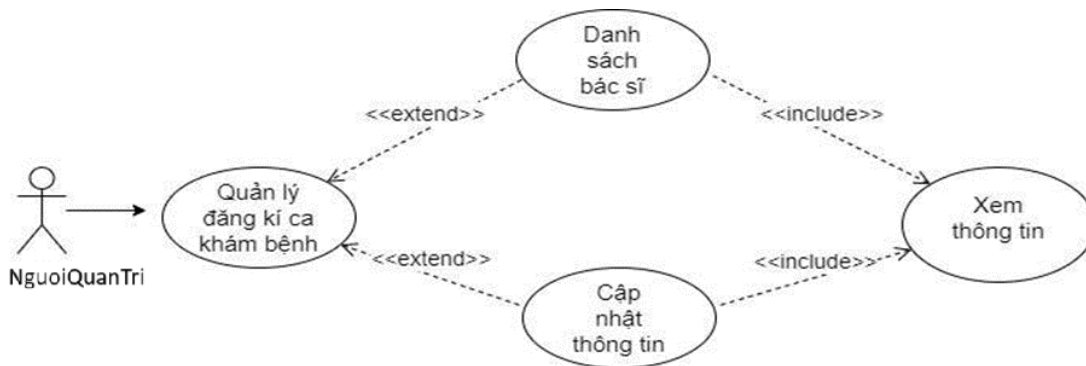
Bảng 2.13: Đặc tả chức năng đăng ký lịch khám

Mã UC: F4	Tên Use Case: Đăng ký lịch khám		
Tác nhân	Bệnh nhân		
Mô tả	Cho phép bệnh nhân đặt lịch khám trực tuyến với bác sĩ. Bệnh nhân có thể chọn ngày giờ khám phù hợp và gửi yêu cầu đặt lịch.		
Sự kiện kích hoạt chức năng	Bệnh nhân chọn chức năng đặt lịch khám từ danh sách bác sĩ hoặc trang cá nhân của bác sĩ. Hệ thống hiển thị lịch trình khả dụng của bác sĩ và yêu cầu bệnh nhân nhập thông tin cần thiết. Bệnh nhân điền thông tin và chọn ngày giờ khám. Hệ thống xác nhận thông tin và gửi yêu cầu đặt lịch đến bác sĩ.		
Tiền điều kiện	Bệnh nhân đã đăng nhập vào ứng dụng Vinmec.		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động

	1	Bệnh nhân	Chọn chức năng đặt lịch khám từ danh sách bác sĩ hoặc trang cá nhân của bác sĩ.
	2	Hệ thống	Hệ thống hiển thị lịch trình khả dụng của bác sĩ và yêu cầu bệnh nhân nhập thông tin cần thiết (ngày, giờ, chuyên khoa, v.v.).
	3	Bệnh nhân	Bệnh nhân điền thông tin và chọn ngày giờ khám
	4	Hệ thống	Xác nhận thông tin và gửi yêu cầu đặt lịch đến bác sĩ.
Luồng sự kiện rẽ nhánh			
	#	Thực hiện bởi	Hành động
	1	Hệ thống	Nếu không kết nối được với cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case, hệ thống sẽ hiển thị một thông báo lỗi.
Hậu điều kiện	Nếu use case kết thúc thành công, lịch khám sẽ được cập nhật trong cơ sở dữ liệu.		

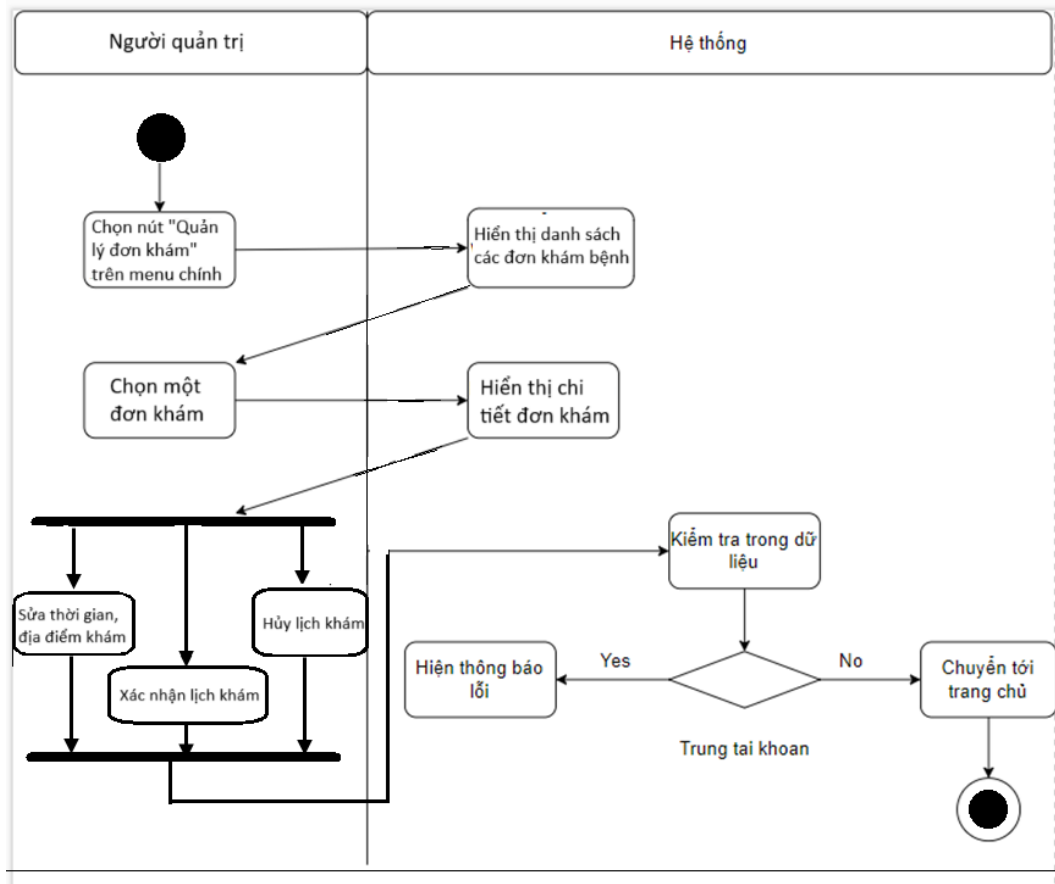
(5) Quy trình quản lý đơn khám bệnh

Use case này cho phép người quản trị quản lý đơn khám.



Hình 2.7: Biểu đồ minh họa Use Case quản lý đơn khám

Bảng 2.14: Biểu đồ minh họa quy trình quản lý đơn khám

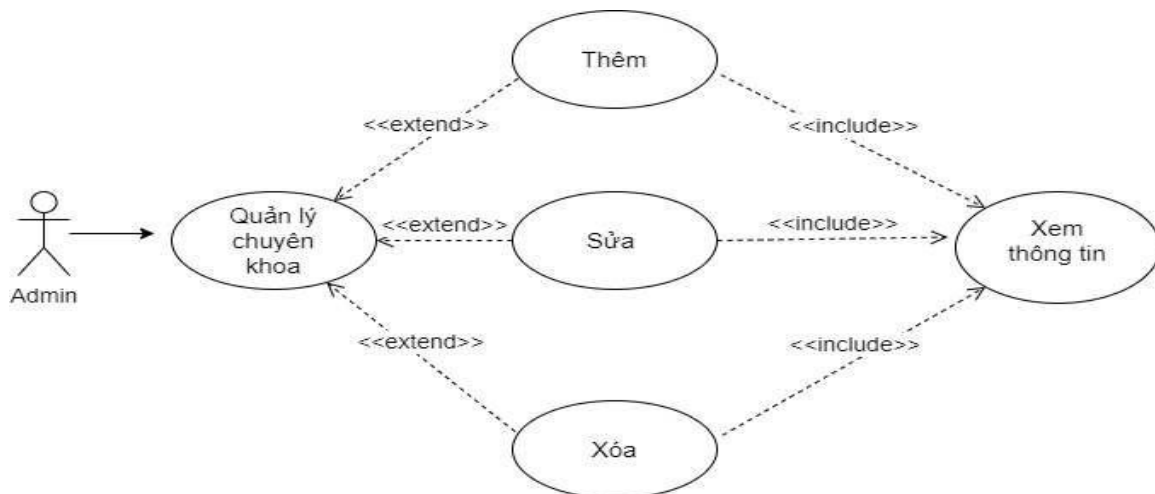


Bảng 2.15: Đặc tả chức năng quản lý đơn khám

Mã UC: F5	Tên UC: Quản Lý đơn Khám		
Tác nhân	Bác sĩ, Người quản trị		
Mô tả	Cho phép bác sĩ và người quản trị tế cập nhật và quản lý đơn khám của bệnh nhân. Hệ thống thông báo cho bệnh nhân về mọi thay đổi trong lịch khám.		
Sự kiện kích hoạt chức năng	Bác sĩ hoặc nhân viên y tế truy cập vào hệ thống quản lý đơn khám.		
Tiền điều kiện	Bác sĩ hoặc Người quản trị đã đăng nhập vào ứng dụng Vinmec.		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Bác sĩ/ Người	Truy cập vào hệ thống quản lý đơn khám.

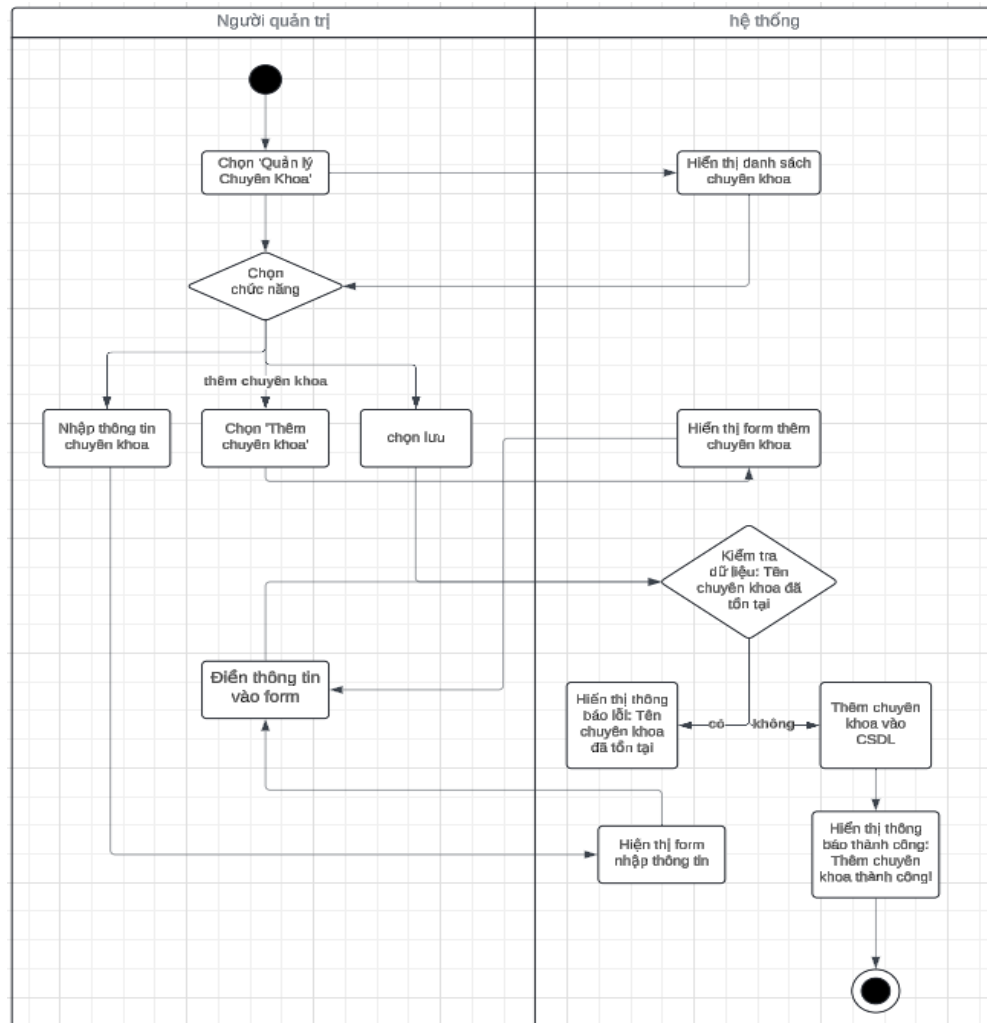
		quản trị	
	2	Hệ thống	Hiển thị danh sách đơn khám hiện tại và cho phép chỉnh sửa (thêm, sửa, xóa đơn khám).
	3	Bác sĩ/ Người quản trị	Cập nhật thời gian khám hoặc thông tin liên quan
	4	Hệ thống	Lưu thay đổi và thông báo cho bệnh nhân qua email hoặc tin nhắn.
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động
	1	Hệ thống	Nếu không kết nối được với cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case, hệ thống sẽ hiển thị một thông báo lỗi.
Hậu điều kiện	Nếu use case kết thúc thành công, đơn khám sẽ được cập nhật trong cơ sở dữ liệu.		

(6) Quy trình quản lý chuyên khoa



Hình 2.8: Biểu đồ minh họa Use Case quản lý chuyên khoa

Bảng 2.16: Biểu đồ minh họa quy trình quản lý chuyên khoa

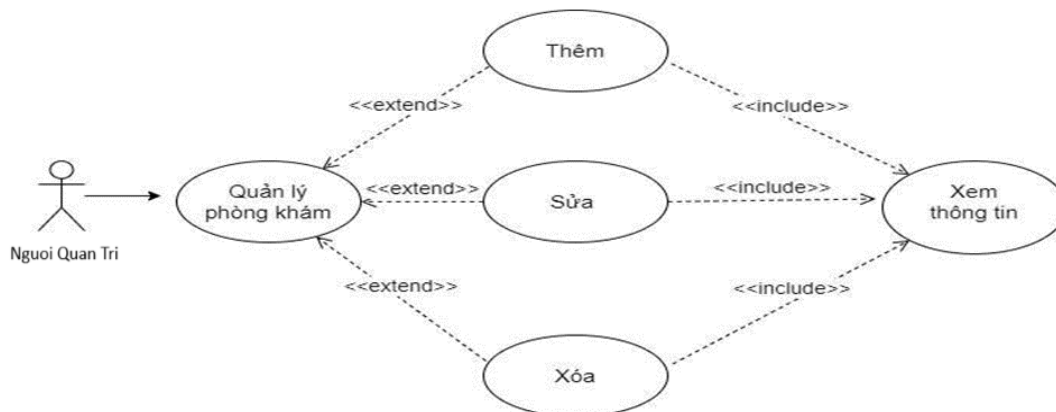


Bảng 2.17: Đặc tả chức năng quản lý chuyên khoa

Mã UC: F6	Tên UC: Quản lý chuyên khoa								
Tác nhân	Người quản trị								
Mô tả	UC này cho phép người quản trị thêm và thay đổi thông tin của chuyên khoa								
Sự kiện kích hoạt chức năng	UC bắt đầu khi người quản trị click vào “Quản lý chuyên khoa” . Hệ thống sẽ lấy thông tin các chuyên khoa và hiển thị lên màn hình								
Tiền điều kiện	Người quản trị đã đăng nhập vào hệ thống								
Luồng sự kiện cơ bản	<table><tr><td>#</td><td>Thực hiện bởi</td><td>Hành động</td></tr><tr><td> </td><td> </td><td> </td></tr></table>			#	Thực hiện bởi	Hành động			
#	Thực hiện bởi	Hành động							

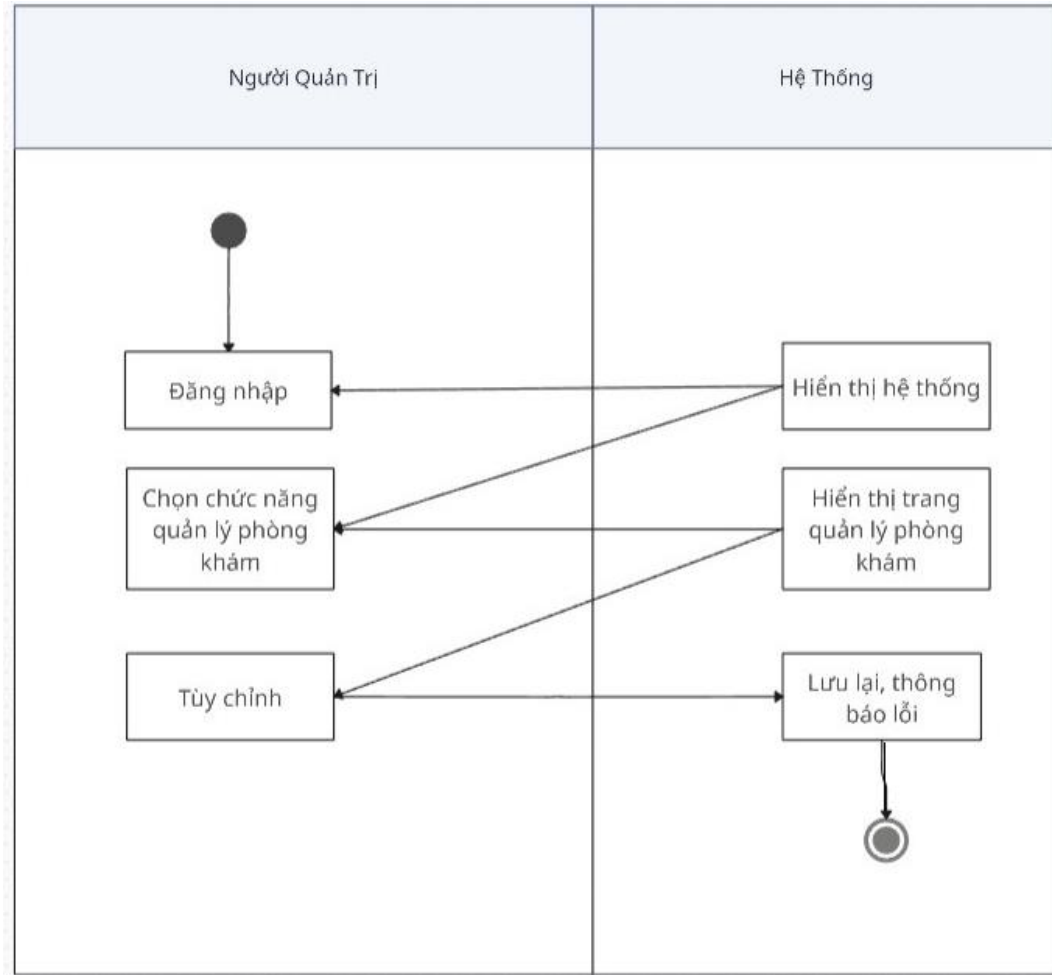
	1	Người quản trị	Click vào nút “Chuyên khoa”
	2	Hệ thống	Hiển thị danh sách các chuyên khoa của bệnh viện
	3	Người quản trị	Chọn một chuyên khoa bất kì
	4	Người quản trị	<ul style="list-style-type: none"> Thêm chuyên khoa vào hệ thống bằng cách điền những thông tin cần thiết. Sửa đổi thông tin chuyên khoa Xóa thông tin về chuyên khoa
	5	Người quản trị	Nhấn xác nhận
	6	Hệ thống	Xác nhận thông tin, lưu vào hệ thống, trở về trang chủ
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động
	1	Người quản trị	Nếu người quản trị nhập thông tin không hợp lệ hoặc thiếu thông tin bắt buộc, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại thông tin.
	2	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc
Hậu điều kiện	Nếu Use case kết thúc thành công thì thông tin được cập nhật sẽ được lưu trong cơ sở dữ liệu		

(7) Quy trình quản lý phòng khám



Hình 2.9: Biểu đồ minh họa Use Case quản lý phòng khám

Bảng 2.18: Biểu đồ minh họa quy trình quản lý phòng khám



Bảng 2.19: Đặc tả Use Case quản lý phòng khám

Mã UC: F7	Tên UC: Quản lý phòng khám		
Tác nhân	Người quản trị		
Mô tả	Cho phép người quản trị thêm và thay đổi thông tin của phòng khám.		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người quản trị click vào “Phòng khám”, hệ thống sẽ truy cập cơ sở dữ liệu và lấy thông tin của các phòng khám và hiển thị lên màn hình.		
Tiền điều kiện	Người quản trị đã đăng nhập vào hệ thống		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động

	1	Người quản trị	Chọn một phòng khám
	2	Hệ thống	Hiển thị thông tin của phòng khám
	3	Người quản trị	Thêm, sửa, xóa thông tin của phòng khám
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động
	1	Hệ thống	Nếu người dùng nhập thông tin mới cho phòng khám không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và quay trở lại bước 2.
	2	Hệ thống	Nếu không kết nối được với cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case, hệ thống sẽ hiển thị một thông báo lỗi.
Hậu điều kiện	Nếu use case kết thúc thành công, thông tin về phòng khám sẽ được thay đổi và hiển thị trên giao diện.		

2.2.3. Yêu cầu phi chức năng

Khi phát triển phần mềm thăm khám y tế, các yêu cầu phi chức năng đặc biệt quan trọng để đảm bảo tính hiệu quả, bảo mật và tin cậy của hệ thống. Dưới đây là những yêu cầu phi chức năng cần được chú ý:

(1) Hiệu năng:

- Thời gian phản hồi: Ứng dụng cần phản hồi nhanh chóng đối với các thao tác của người dùng, đặc biệt là trong việc tìm kiếm hồ sơ bệnh nhân, đặt lịch hẹn, và truy cập thông tin y tế.
- Khả năng mở rộng: Ứng dụng phải có khả năng mở rộng để hỗ trợ số lượng lớn người dùng đồng thời mà không ảnh hưởng đến hiệu suất.

(2) Tính sẵn sàng:

- Thời gian hoạt động: Ứng dụng cần phải có thời gian hoạt động tối đa, đảm bảo sẵn sàng 24/7, đặc biệt trong các trường hợp khẩn cấp.
- Khả năng khôi phục: Cần có các biện pháp dự phòng và khôi phục nhanh

chóng sau sự cố để đảm bảo dịch vụ không bị gián đoạn.

(3) Bảo mật:

- Bảo vệ dữ liệu: Thông tin y tế cá nhân (PHI) phải được mã hóa khi lưu trữ và truyền tải để bảo vệ khỏi truy cập trái phép.
- Xác thực và quyền truy cập: Cần có các cơ chế xác thực mạnh mẽ và quản lý quyền truy cập để đảm bảo chỉ những người có thẩm quyền mới có thể truy cập và chỉnh sửa thông tin y tế.
- Ghi nhật ký hoạt động: Hệ thống phải ghi lại mọi hoạt động truy cập và thao tác trên dữ liệu để có thể theo dõi và xử lý các hành vi đáng ngờ.

(4) Tính khả dụng:

- Độ tin cậy cao: Ứng dụng phải đảm bảo hoạt động ổn định và giảm thiểu tối đa các lỗi hệ thống.
- Khả năng chịu lỗi: Ứng dụng cần có khả năng chịu lỗi và tự động khôi phục trong trường hợp xảy ra sự cố.

(5) Tính tương thích:

- Tương thích đa hệ thống: Ứng dụng phải tương thích với các hệ thống khác như hệ thống bảo hiểm, quản lý dược phẩm, và các thiết bị y tế.
- Hỗ trợ đa nền tảng: Ứng dụng cần hỗ trợ trên nhiều thiết bị và nền tảng khác nhau (máy tính, máy tính bảng, điện thoại di động).

(6) Khả năng bảo trì:

- Dễ dàng cập nhật: Ứng dụng cần được thiết kế để dễ dàng cập nhật và sửa lỗi mà không ảnh hưởng đến hoạt động của người dùng.
- Tài liệu và hướng dẫn: Cung cấp tài liệu chi tiết và hướng dẫn sử dụng để hỗ trợ việc bảo trì và nâng cấp.

(7) Khả năng sử dụng:

- Giao diện người dùng: Giao diện phải thân thiện, trực quan và dễ sử dụng, giúp người dùng dễ dàng thực hiện các thao tác cần thiết.

- Hỗ trợ người dùng: Cần có các tài liệu hướng dẫn và hỗ trợ kỹ thuật để giúp người dùng giải quyết các vấn đề phát sinh.

(8) Tuân thủ quy định:

- Tuân thủ pháp luật: Ứng dụng phải tuân thủ các quy định pháp luật liên quan đến bảo mật thông tin y tế và quyền riêng tư của bệnh nhân (ví dụ: HIPAA ở Mỹ, GDPR ở Châu Âu).
- Chứng nhận: Ứng dụng cần được chứng nhận và kiểm tra định kỳ để đảm bảo tuân thủ các tiêu chuẩn quy định trong y tế.

2.2.4. Kết luận chương 2:

Vậy qua chương II, chúng em đã đặc tả được các use case cơ bản của ứng dụng thăm khám y tế Vinmec bằng việc sử dụng bảng. Bạn đọc có thể biết được các tác nhân, các mô tả, sự kiện kích hoạt chức năng, các luồng cơ bản, luồng rẽ nhánh, hậu điều kiện có trong một use case.

CHƯƠNG 3

ĐẶC TẢ THIẾT KẾ PHẦN MỀM

Trong Chương 3 này, nhóm chúng em sẽ trình bày về các hoạt động trong thiết kế phần mềm. Trong đó sẽ nói đến các loại thiết kế phần mềm: Thiết kế kiến trúc và thiết kế cấu trúc phần mềm. Ngoài ra các chiến lược thiết kế cũng được đề cập đến trong phần này. Để thực hiện được những nội dung đó, chúng em sẽ dùng đến một số công cụ như: Rational rose, Star UML, Draw.io, ...

3.1. Tổng quan về thiết kế phần mềm

3.1.1. Vai trò và các nhiệm vụ trong hoạt động thiết kế phần mềm

3.1.1.1. Khái niệm

Là quá trình chuyển hóa các đặc tả yêu cầu phần mềm thành một biểu diễn thiết kế của hệ thống phần mềm cần xây dựng, sao cho người lập trình có thể ánh xạ nó thành một chương trình.

3.1.1.2. Một số hoạt động chính:

- Nghiên cứu để hiểu vấn đề.
- Chọn một số giải pháp thiết kế và xác định các đặc điểm thô của nó.
- Mô tả trừu tượng cho mỗi giải pháp thiết kế, các sai sót cần phát hiện và chỉnh sửa trước khi lập tài liệu thiết kế chính thức.

3.1.1.3. Vai trò:

- Là cách duy nhất để chuyển hóa một cách chính xác các yêu cầu của khách hàng thành mô hình thiết kế hệ thống phần mềm cuối cùng làm cơ sở cho việc triển khai chương trình phần mềm.
- Là công cụ giao tiếp giữa các nhóm cùng tham gia phát triển sản phẩm, quản lý rủi ro, đạt được phần mềm hiệu quả.
- Là tài liệu cung cấp đầy đủ thông tin cần thiết cho để bảo trì hệ thống.
- Nếu không có thiết kế thì hệ thống không tin cậy, nguy cơ thất bại cao.
- Thiết kế tốt là chìa khóa làm cho phần mềm trở nên hữu hiệu.

3.1.1.4. Các hoạt động và sản phẩm thiết kế:

- Thiết kế kiến trúc: Xác định các hệ con tạo nên hệ thống tổng thể và mối quan hệ giữa chúng
- Đặc tả trừu tượng: Mô tả trừu tượng các dịch vụ của hệ con
- Thiết kế giao diện thành phần
- Thiết kế cấu trúc dữ liệu
- Thiết kế hệ thống giao diện người dùng
- Thiết kế các thành phần
- Thiết kế thủ tục

3.1.1.5. Biểu diễn thiết kế.

Có ba hình thức biểu diễn thiết kế:

- Các biểu đồ: Biểu diễn các mối quan hệ giữa các thành phần của hệ thống, vừa là mô hình mô tả thế giới thực
- Ngôn ngữ mô tả chương trình: Dùng để kiểm tra và cấu trúc các cơ cấu thiết kế dựa trên các cấu trúc của một ngôn ngữ lập trình
- Dạng văn bản không hình thức hóa: Mô tả các thông tin không thể hình thức hóa được như thông tin phi chức năng bên cạnh cách mô tả khác

3.1.1.6. Các giai đoạn thiết kế:

Nhìn theo các khía cạnh khác nhau: Nhà quản lý, người phát triển, mức độ hình thức hóa.

3.1.2. Chiến lược thiết kế phần mềm

3.1.2.1. Thiết kế hướng chức năng

- Là một phương pháp thiết kế phần mềm tập trung vào việc xác định và mô tả các chức năng chính của hệ thống. Tập trung vào cách hệ thống hoạt động và tương tác với người dùng và các thành phần khác
- Trước khi bắt đầu quá trình thiết kế hướng chức năng, các yêu cầu chức năng đã được phân tích và mô tả chi tiết trong giai đoạn phân tích hệ

thống. Thiết kế hướng chức năng tạo ra một cấu trúc và mô hình cho việc triển khai các chức năng này

- Trong quá trình thiết kế hướng chức năng, các chức năng chính được xác định và mô tả chi tiết. Các thành phần, module của hệ thống được thiết kế để thực hiện các chức năng này. Mô hình dữ liệu và luồng xử lý được xác định để đảm bảo tính logic và lung thông tin chính xác
- Một số phương pháp thường được sử dụng trong thiết kế hướng chức năng: biểu đồ UML (Biểu đồ Use Case, Activity, Sequence) để mô tả các chức năng và luồng xử lý.

3.1.2.2. Thiết kế hướng đối tượng

- Là một phương pháp thiết kế phần mềm dựa trên các đối tượng và mối quan hệ giữa chúng. Tập trung vào việc mô hình hóa thế giới thực thành các đối tượng trong phần mềm và thiết kế các lớp, phương thức và thuộc tính để đảm bảo tính linh hoạt, dễ mở rộng và tái sử dụng của hệ thống.
- Các đối tượng được xác định dựa trên các đặc điểm và hành vi của chúng. Các đối tượng được tổ chức thành các lớp, mỗi lớp chứa các thuộc tính và phương thức tương ứng. Mối quan hệ giữa các đối tượng được mô tả bằng các khái niệm như kế thừa, giao tiếp và liên kết.
- Trong quá trình thiết kế hướng đối tượng, ta sử dụng các nguyên tắc thiết kế SOLID (Single Responsibility, Open-Closed, Dependency Inversion) để tạo ra một thiết kế linh hoạt và dễ bảo trì

3.1.3. Công cụ hỗ trợ thiết kế phần mềm

3.1.3.1. Rational Rose:

- Rational Rose là một công cụ CASE (Computer-Aided Software Engineering) từ IBM.
- Công cụ được sử dụng để mô hình hóa phần mềm bằng các biểu đồ UML (Unified Modeling Language): biểu đồ lớp, biểu đồ tuần tự, biểu đồ hoạt động

- Giúp các nhà phát triển phần mềm hiểu rõ hơn về cấu trúc và quan hệ giữa các phần của hệ thống, giúp cải thiện quá trình phát triển và bảo trì phần mềm.

3.1.3.2. Star UML:

Là một công cụ mã nguồn mở và miễn phí được sử dụng để tạo và quản lý các biểu đồ UML. UML là một ngôn ngữ tiêu chuẩn được sử dụng rộng rãi trong lĩnh vực phát triển phần mềm để mô tả, thiết kế và trình bày các yếu tố trong một hệ thống phần mềm.

3.1.3.3. Draw.io:

- Draw.io là một công cụ vẽ sơ đồ trực tuyến, miễn phí và mã nguồn mở.
- Công cụ được sử dụng để tạo ra các biểu đồ và sơ đồ cho nhiều mục đích khác nhau, bao gồm sơ đồ luồng công việc, sơ đồ UML, sơ đồ mạng, và nhiều loại biểu đồ khác.
- Draw.io rất linh hoạt và dễ sử dụng, có khả năng tích hợp với nhiều dịch vụ lưu trữ đám mây như Google Drive, OneDrive, và Dropbox.

3.2. Tài liệu thiết kế phần mềm

3.2.1. Giới thiệu

3.2.1.1. Mục đích tài liệu

Tài liệu này mô tả chi tiết về thiết kế phần mềm Vinmec phục vụ hoạt động đăng ký lịch khám trực tuyến của bệnh nhân.

3.2.1.2. Phạm vi tài liệu

Tài liệu này là cơ sở giao tiếp của các thành viên trong đội phát triển dự án phần mềm Vinmec và tài liệu này cũng là căn cứ trong hoạt động kiểm thử, vận hành và bảo trì phần mềm.

3.2.1.3. Tài liệu tham khảo

- Giáo trình Nhập môn công nghệ phần mềm ĐHCNHN(2016) - Hoàng Quang Huy, Phùng Đức Hoà, Trịnh Bá Quý
- [https://www.studocu.com/vn/document/truong-dai-hoc-cong-nghep-ha-](https://www.studocu.com/vn/document/truong-dai-hoc-cong-nghep-ha-noi/)

noi/nhap-mon-cong-nghe-phan-mem/bai-tap-lon-nhap-mon-cong-nghe-phan-mem/66630123

- <https://tailieuhust.com/nhap-mon-cong-nghe-phan-mem-hust/>
- <https://tailieuhuongdan.com/bai-tap-lon-mon-nhap-mon-cong-nghe-phan-mem-614/>

3.2.2. Thiết kế kiến trúc phần mềm

3.2.2.1. Mô hình kiến trúc

Kiến trúc phần mềm chỉ một cấu trúc phần mềm, qua đó cung cấp một sự tích hợp chặt về mặt khái niệm của hệ thống, có vai trò quan trọng trong phát triển phần mềm, gồm 3 vai trò:

- Công cụ giao tiếp giữa những người liên quan
- Để phân tích hệ thống
- Sử dụng lại ở quy mô lớn

Kiến trúc không phải là thành phần hoạt động nhưng nó có tác động sâu rộng đến quá trình phát triển phần mềm, nó là 1 mô tả phần mềm cho phép các kỹ sư thực hiện công việc:

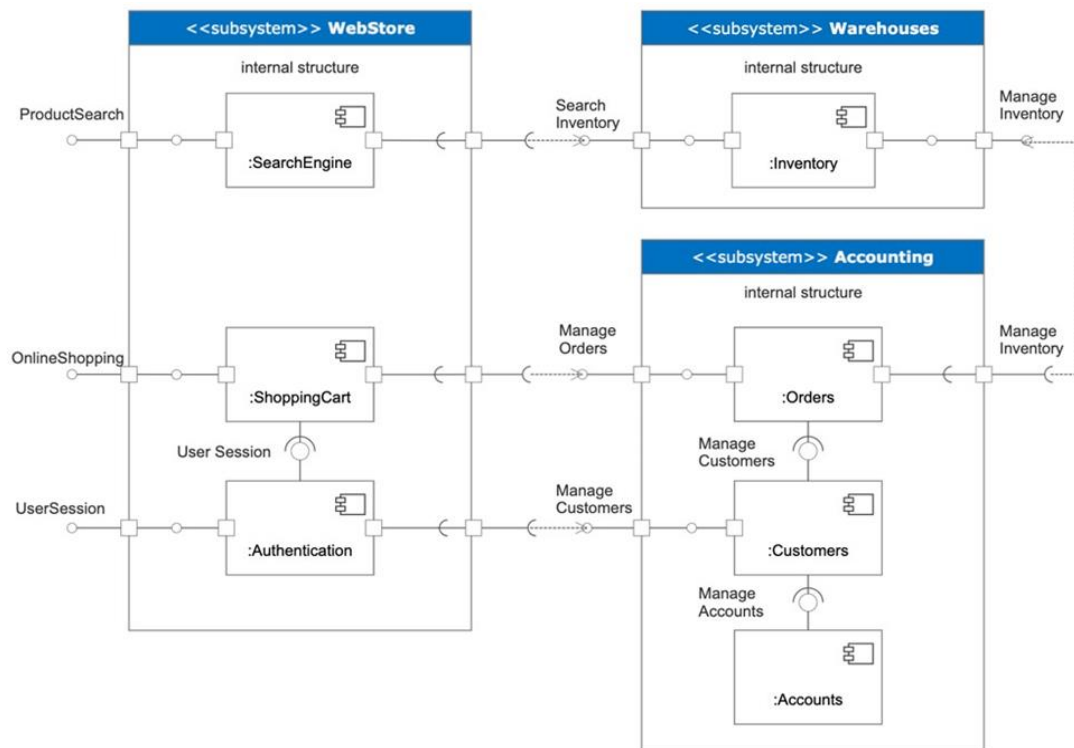
- Tăng cường hiểu biết về hệ thống cần xây dựng
- Phân tích hiệu quả
- Xem xét, sửa đổi kiến trúc từ sớm, giảm rủi ro

Một số mô hình kiến trúc phổ dụng:

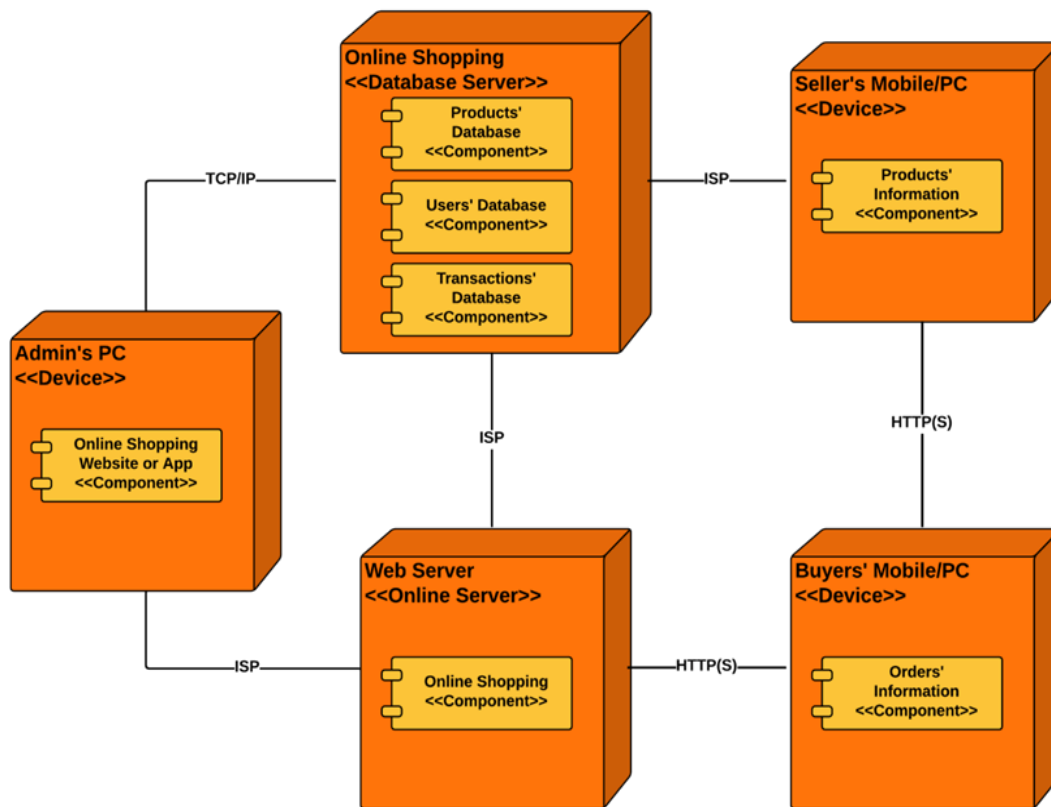
- Kiến trúc đơn thể: Là một kiểu kiến trúc phần mềm cổ điển, trong đó toàn bộ ứng dụng được phát triển và triển khai như một đơn vị duy nhất. Trong kiến trúc này, tất cả các thành phần và chức năng của ứng dụng được tích hợp và triển khai trong một đơn vị lớn. Mặc dù kiến trúc đơn thể đơn giản và dễ triển khai ban đầu, nhưng nó có thể gặp khó khăn trong việc mở rộng và bảo trì khi ứng dụng phát triển lớn.
- Kiến trúc các thành phần: Tách ứng dụng thành các thành phần độc lập,

mỗi thành phần đảm nhận một chức năng cụ thể. Các thành phần có thể được phát triển, triển khai và bảo trì độc lập với nhau, giúp dễ dàng tái sử dụng và mở rộng. Mỗi thành phần thường có giao diện rõ ràng để giao tiếp với các thành phần khác, thông qua đó tạo ra một hệ thống phức tạp từ các thành phần đơn giản.

- Kiến trúc hướng đối tượng: Ứng dụng được xây dựng xung quanh các đối tượng, mỗi đối tượng biểu diễn một thực thể hoặc một phần của hệ thống. Các đối tượng có thể bao gồm dữ liệu và các phương thức để thao tác dữ liệu đó, và chúng tương tác với nhau thông qua giao diện. Kiến trúc này thường được sử dụng để xây dựng các hệ thống phức tạp, giúp dễ dàng quản lý, bảo trì và mở rộng hệ thống.



Hình 3.1: Component diagram



Hình 3.2: Deployment diagram

3.3.2.2. Mô tả kiến trúc

Mô tả chi tiết kiến trúc hướng đối tượng:

- Kiến trúc hướng đối tượng: Là một phương pháp thiết kế phần mềm dựa trên các đối tượng. Mỗi đối tượng đại diện cho một thực thể trong thế giới thực và có các đặc điểm (thuộc tính) và hành vi (phương thức) riêng. Các đối tượng tương tác với nhau để hoàn thành các nhiệm vụ.
- Đặc điểm chính của OOA:
 - o Tập trung vào đối tượng: Hệ thống được mô hình hóa như một tập hợp các đối tượng tương tác với nhau.
 - o Trừu tượng hóa: Sử dụng các lớp để mô tả các đặc điểm chung của các đối tượng, giúp cho mã dễ tái sử dụng và bảo trì.
 - o Kế thừa: Cho phép các lớp kế thừa các thuộc tính và phương thức từ các lớp khác, tạo ra sự phân cấp các lớp.

- o Đa hình: Cho phép các đối tượng thuộc các lớp khác nhau phản hồi cùng một tin nhắn theo cách khác nhau.
- o Liên kết: Cho phép các đối tượng tham chiếu lẫn nhau, tương tác với nhau.
- Ưu điểm:
 - o Dễ bảo trì: Mã hướng đối tượng dễ hiểu, dễ sửa đổi và mở rộng hơn so với mã thủ tục.
 - o Tái sử dụng: Các lớp và thành phần có thể được tái sử dụng trong nhiều dự án khác nhau.
 - o Tính linh hoạt: Kiến trúc hướng đối tượng có thể dễ dàng thích ứng với những thay đổi về yêu cầu.
 - o Bảo trì: Hệ thống hướng đối tượng dễ bảo trì hơn do tính mô đun và tính trừu tượng hóa cao.
- Thành phần chính của OOA:
 - o Lớp: Lớp là bản mẫu cho các đối tượng, mô tả các thuộc tính và phương thức chung của các đối tượng thuộc lớp đó.
 - o Đối tượng: Đối tượng là một thể hiện cụ thể của một lớp, có trạng thái và hành vi riêng.
 - o Thuộc tính: Thuộc tính mô tả trạng thái của một đối tượng.
 - o Phương thức: Phương thức mô tả hành vi của một đối tượng.
 - o Mối quan hệ: Mối quan hệ mô tả mối liên hệ giữa các đối tượng.

Phân rã từng thành phần:

- Thành phần Quản lý tài khoản: Chịu trách nhiệm quản lý các tài khoản được khách hàng đăng ký trên ứng dụng. Gồm các lớp hoặc đối tượng như: User, Password, Email để đại diện cho người dùng, người quản trị. Cung cấp các phương thức để đăng nhập, thay đổi thông tin tài khoản.
- Thành phần Quản lý đơn khám: Thành phần này quản lý quá trình đặt lịch khám và xử lý lịch khám trên ứng dụng. Bao gồm các lớp hoặc đối tượng

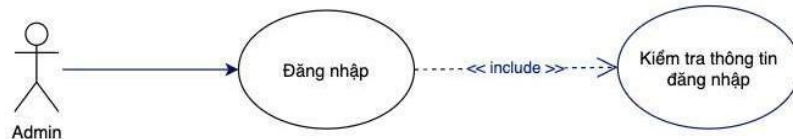
như Mã đơn, ngày khám, phòng khám, thời gian khám. Cung cấp các phương thức để thêm, hủy, thay đổi thời gian, thông tin trên phiếu khám.

- Thành phần Quản lý Giao diện Người dùng: Bao gồm các trang, menu, và giao diện người dùng khác. Gồm các lớp hoặc đối tượng như Page, Menu, Layout để đại diện cho thông tin trang, menu và bố cục giao diện. Cung cấp các phương thức để tạo và quản lý các trang, menu, và cập nhật giao diện người dùng.
- Thành phần Quản lý Tính năng: Quản lý các tính năng đặc biệt trên ứng dụng, như tìm kiếm sản phẩm, xem thông tin khuyến mãi, đánh giá và bình luận. Bao gồm các lớp hoặc đối tượng như Search Engine, Recommendation Engine, Rating, Comment để đại diện cho các tính năng này. Cung cấp các phương thức để thực hiện các tính năng này và tích hợp chúng vào giao diện người dùng.

3.2.3. Thiết kế cấu trúc phần mềm

3.2.3.1. Yêu cầu chức năng đăng nhập tài khoản

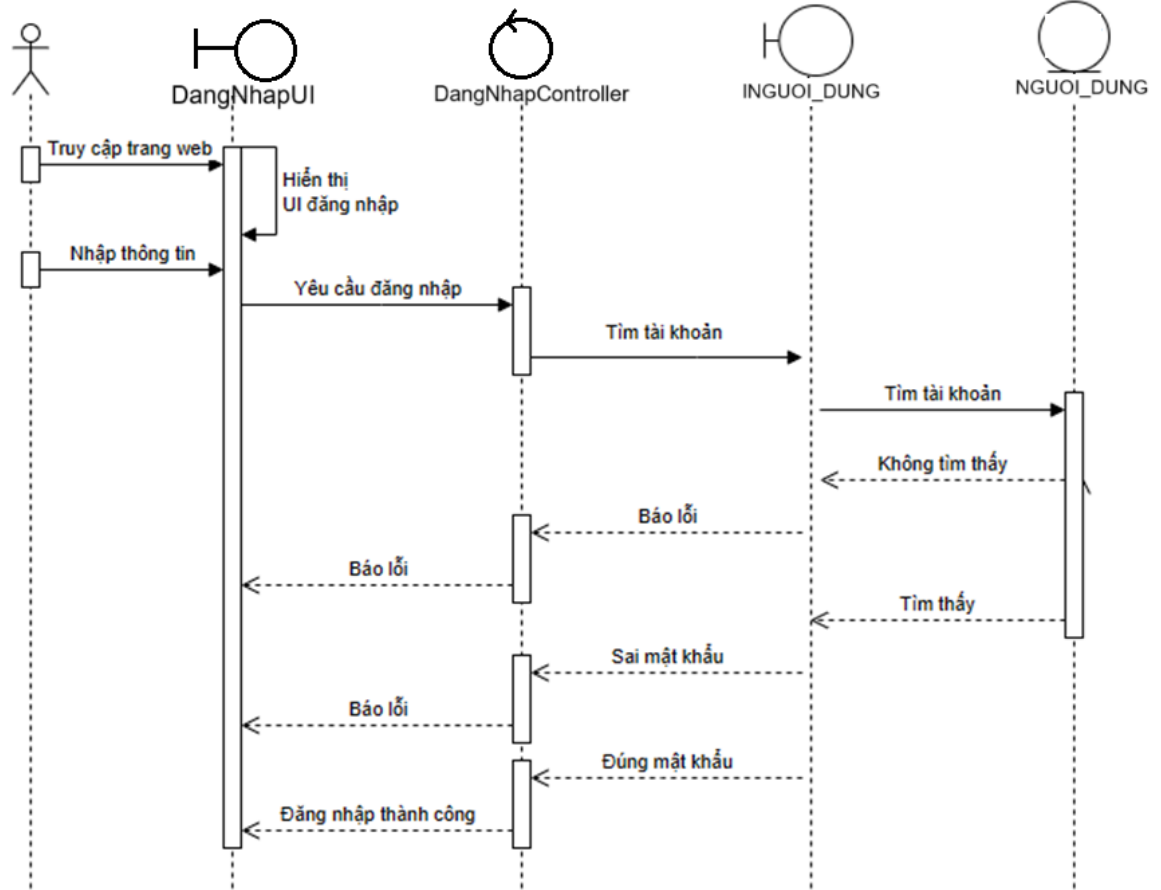
(1) Biểu đồ Use Case minh họa



Hình 3.3: Biểu đồ minh họa Use Case đăng nhập

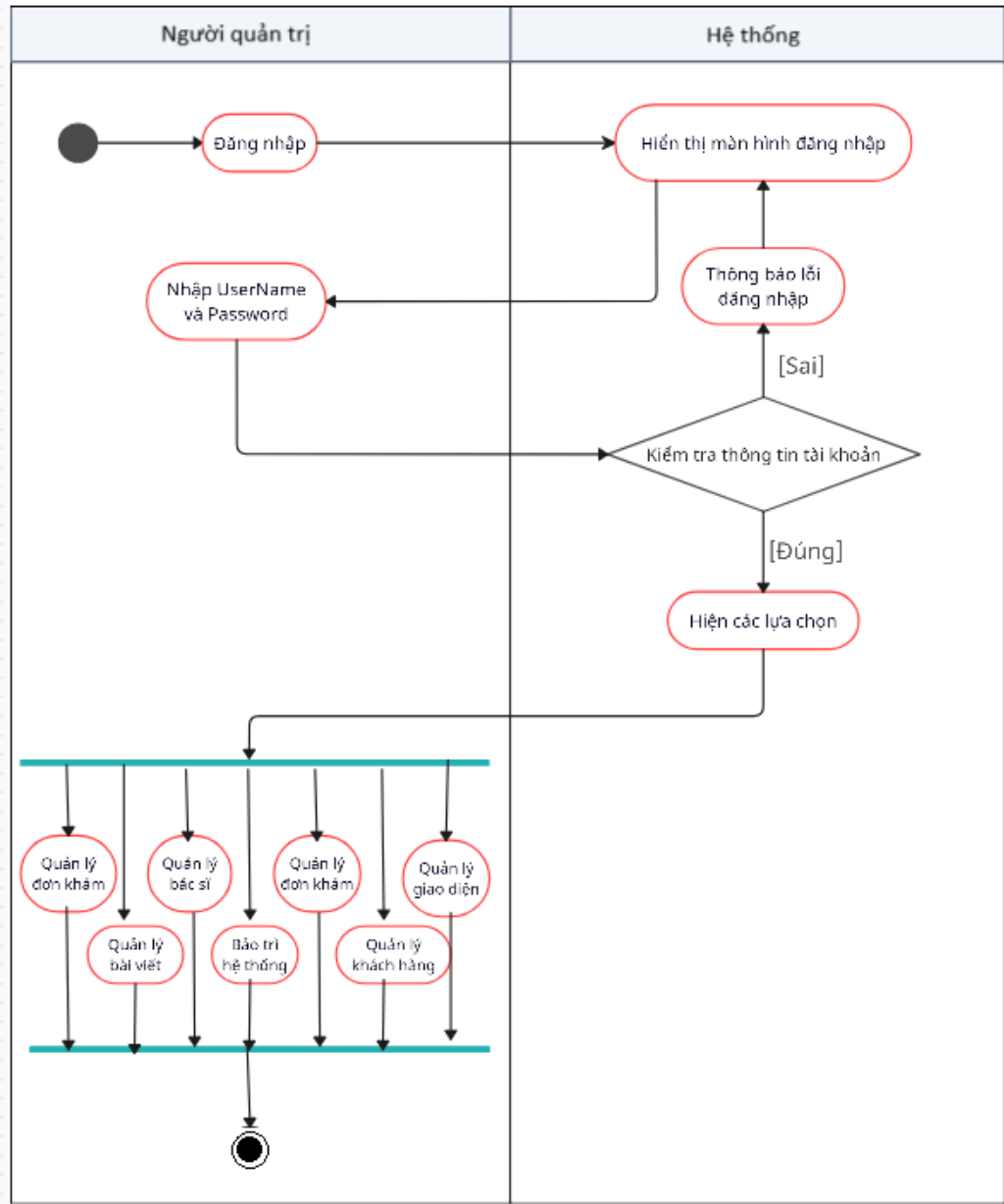
(2) Biểu đồ trình tự

Bảng 3.1: Biểu đồ trình tự chức năng đăng nhập



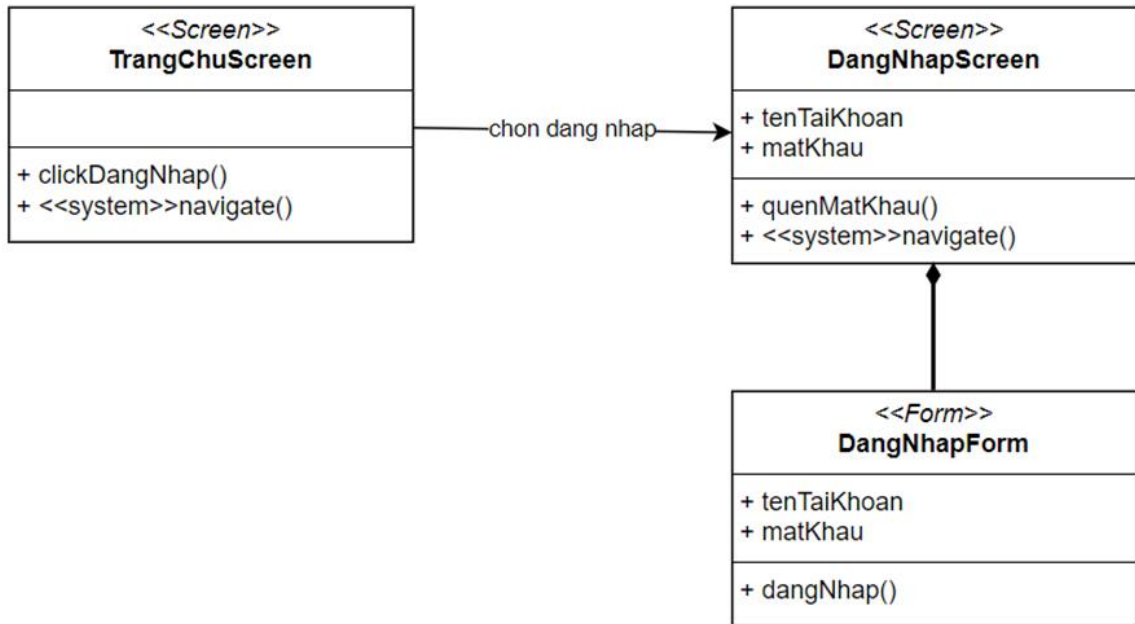
(3) Biểu đồ hoạt động

Bảng 3.2: Biểu đồ minh họa quy trình đăng nhập



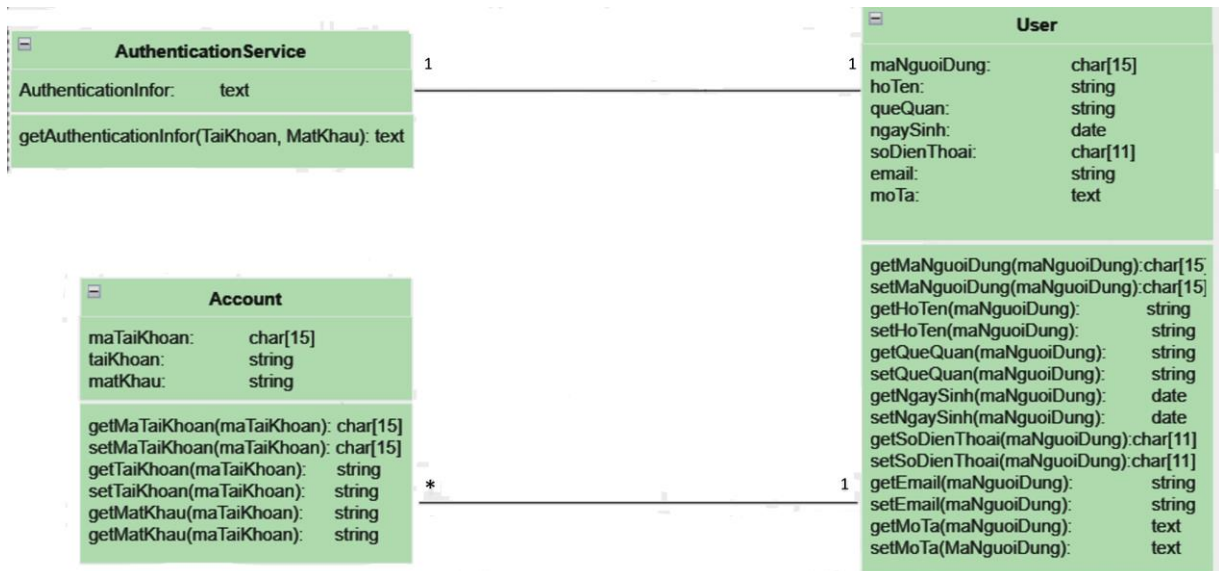
(4) Biểu đồ lớp tham gia ca sử dụng

Bảng 3.3: Biểu đồ lớp chức năng Đăng nhập



(5) Các lớp thực thể

Bảng 3.4: Biểu đồ các lớp thực thể chức năng Đăng Nhập



(6) Thuật toán

```

def DangNhap(TaiKhoan, MatKhai):
    getDangNhap = Lấy thông tin tài khoản, mật khẩu từ CSDL
    if getDangNhap is NULL:
  
```

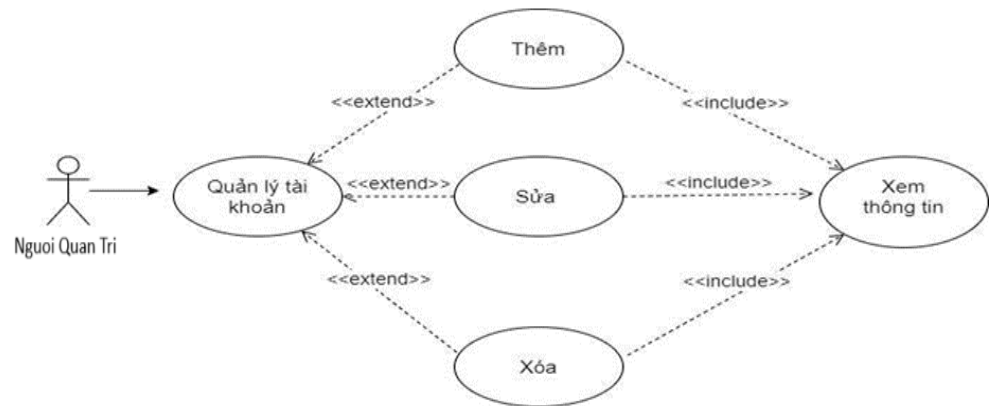
```

return "Tài khoản không tồn tại!!!"
elif TaiKhoan != getDangNhap.TaiKhoan
    or MatKhau != getDangNhap.MatKhau
    return "Tài khoản hoặc mật khẩu không chính xác!"
else:
    return "Đăng nhập thành công "

```

3.2.3.2. Yêu cầu chức năng quản lý tài khoản

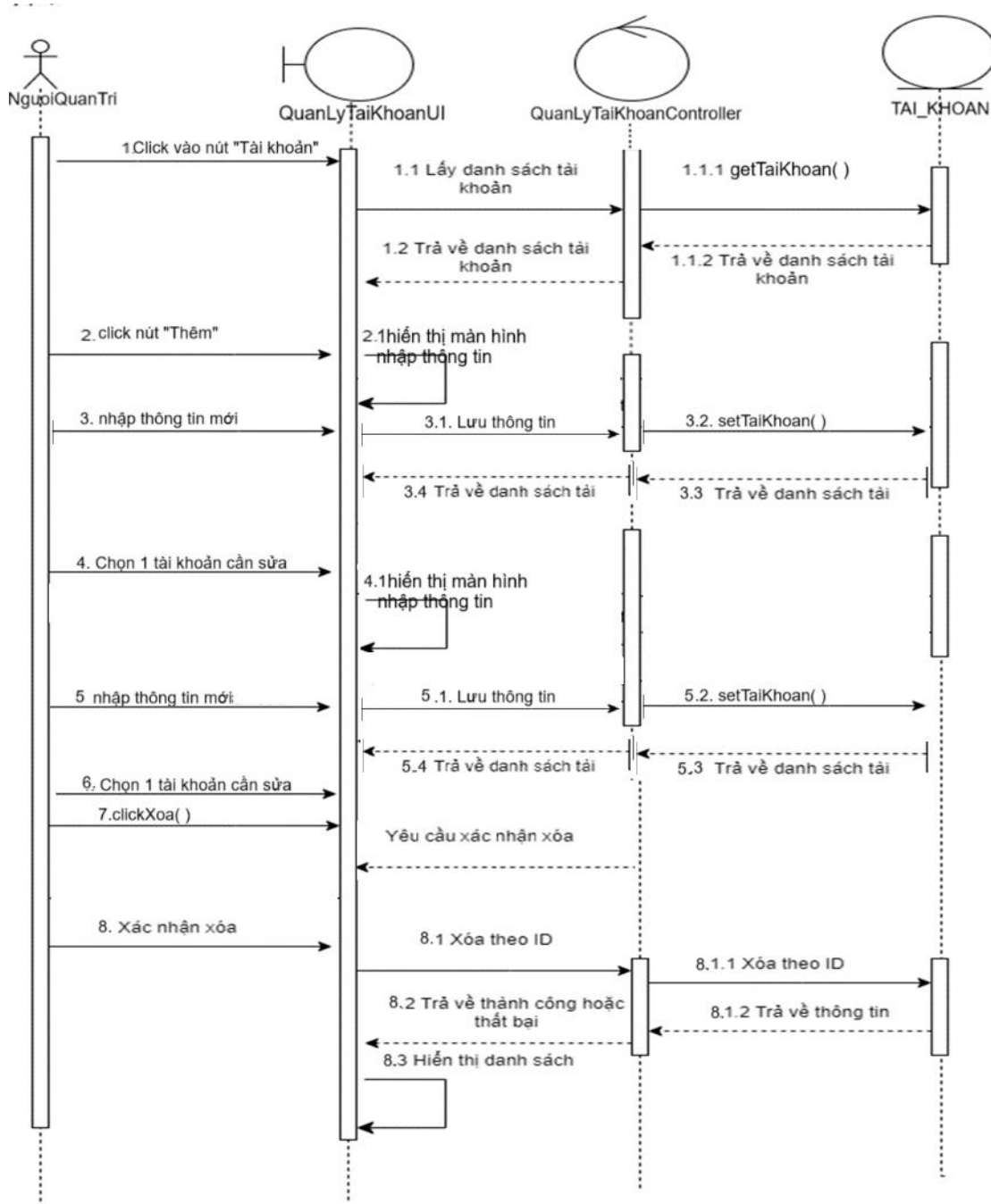
(1) Biểu đồ Use Case minh họa



Hình 3.4: Biểu đồ minh họa Use Case quản lý tài khoản

(2) Biểu đồ trình tự

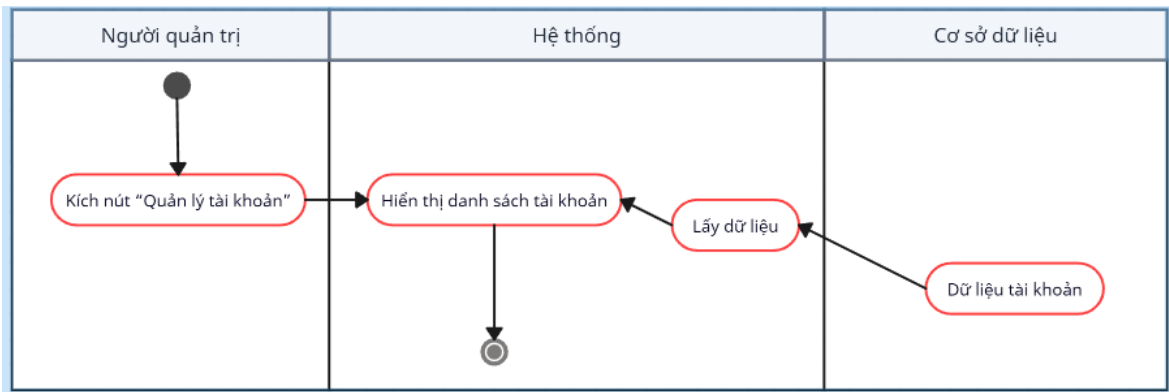
Bảng 3.5: Biểu đồ trình tự chức năng quản lý tài khoản



(3) Biểu đồ hoạt động

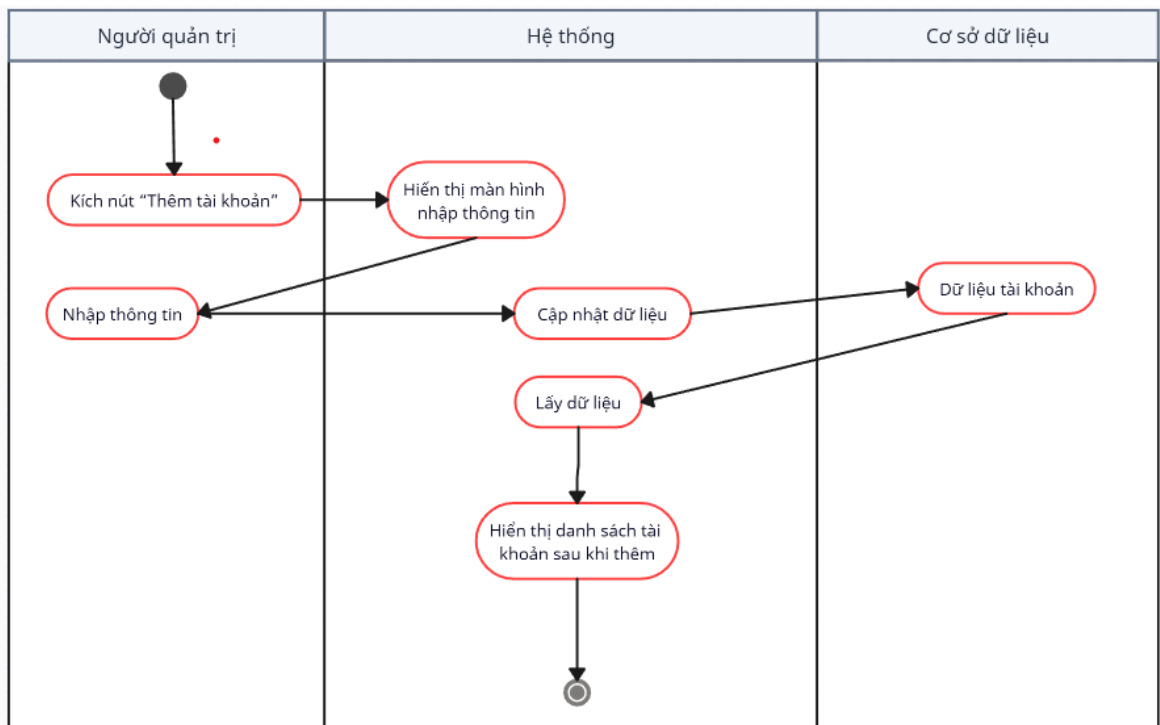
Xem danh sách tài khoản

Bảng 3.6: Biểu đồ minh họa quy trình xem danh sách tài khoản



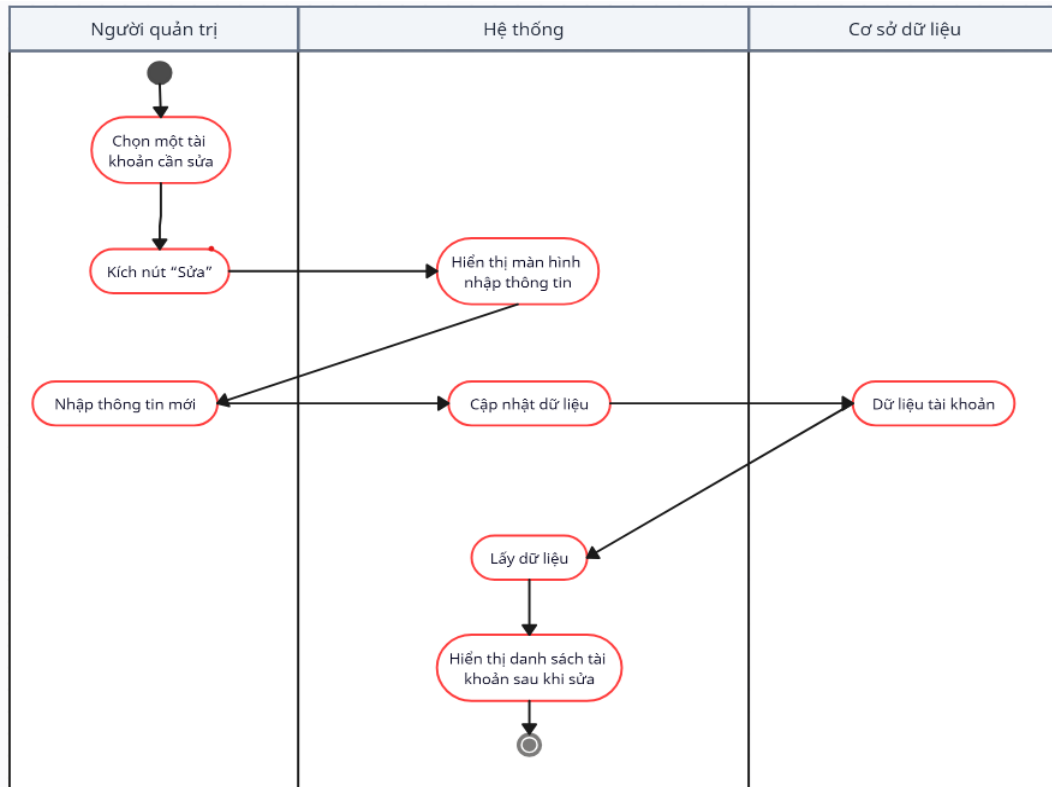
Thêm tài khoản khách hàng

Bảng 3.7: Biểu đồ minh họa quy trình thêm tài khoản



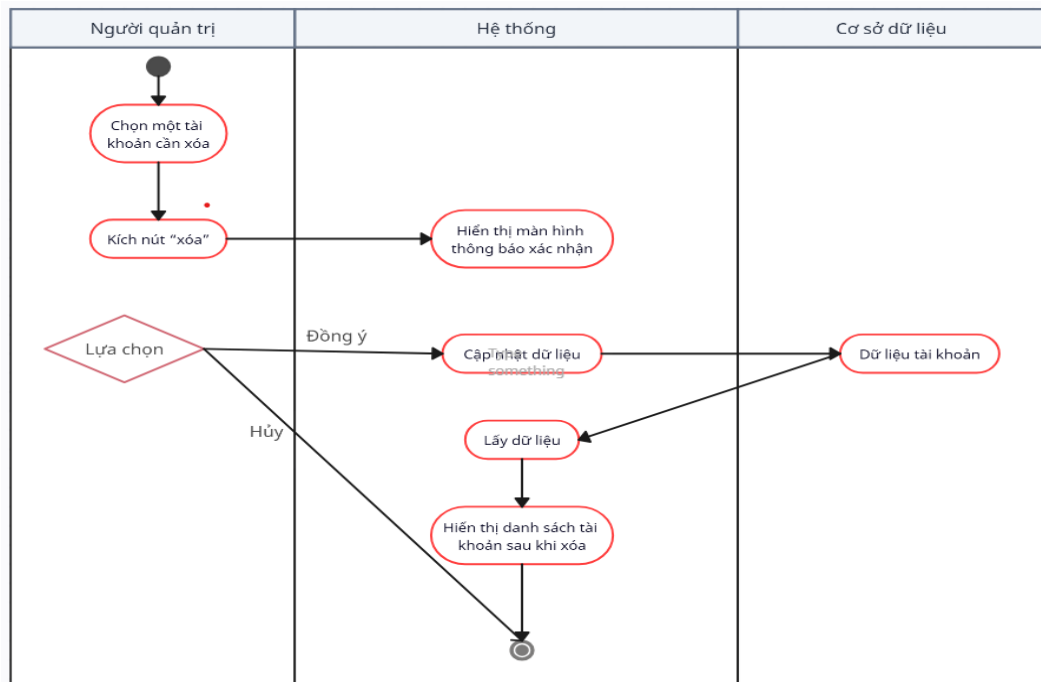
Sửa tài khoản khách hàng

Bảng 3.8: Biểu đồ minh họa quy trình sửa tài khoản



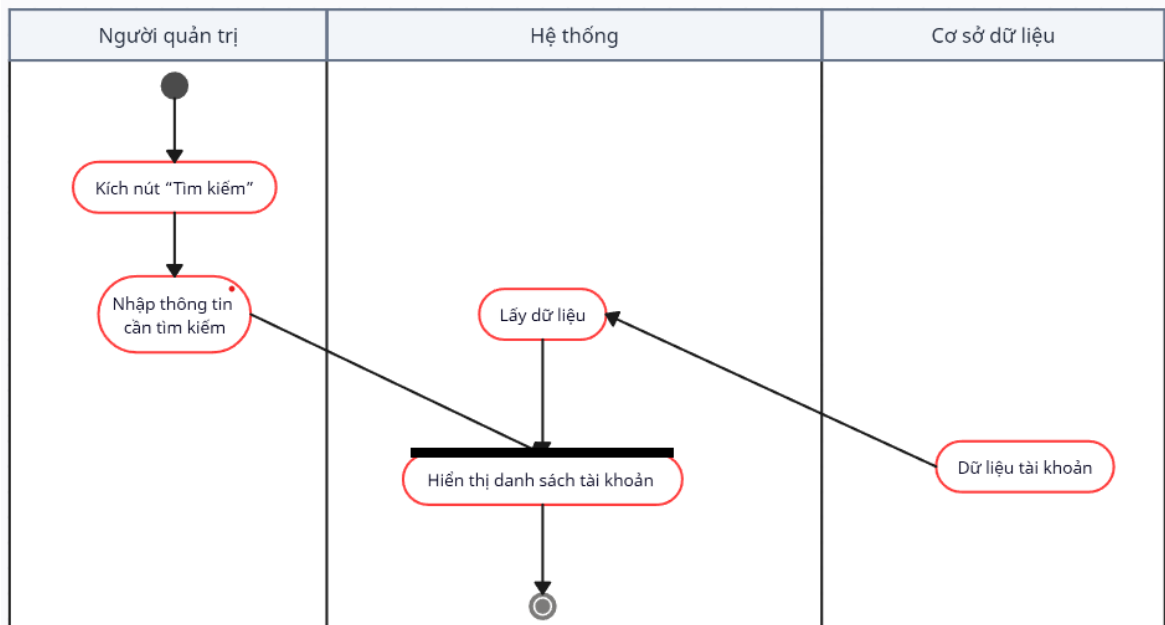
Xóa tài khoản khách hàng

Bảng 3.9: Biểu đồ minh họa quy trình xóa tài khoản



Tìm kiếm tài khoản

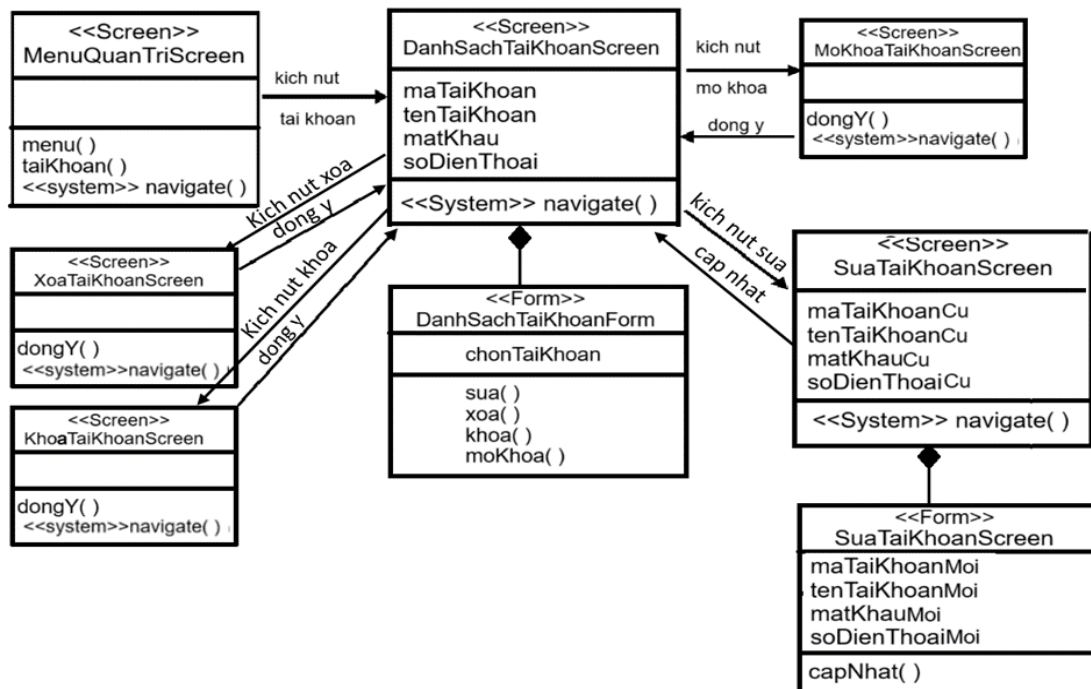
Bảng 3.10: Biểu đồ minh họa quy trình tìm kiếm tài khoản



*Chú ý: Các quy trình Thêm, sửa, xóa, tìm kiếm chỉ có thể được sử dụng khi người quản trị đã kích vào nút “Quản lý tài khoản”.

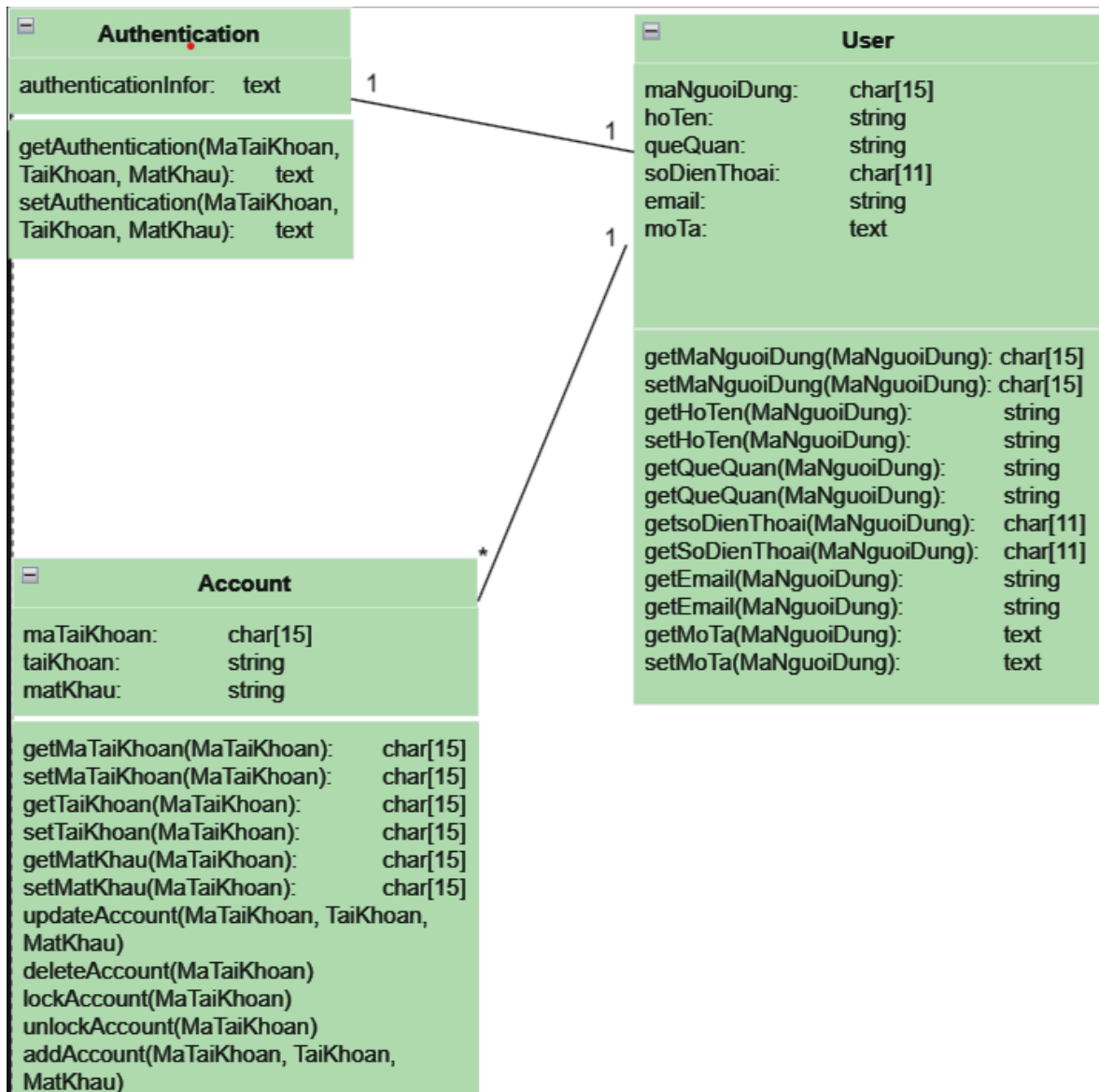
(4) Biểu đồ lớp tham gia ca sử dụng

Bảng 3.11: Biểu đồ lớp chức năng quản lý tài khoản



(5) Các lớp thực thể

Bảng 3.12: Biểu đồ các lớp thực thể chức năng quản lý tài khoản



(6) Thuật toán

```

class QuanLyTaiKhoan:
    def CSDL():
        Lưu trữ tài khoản = [ ]

    def ThemTaiKhoan(MaTaiKhoan, TaiKhoan, MatKhau):
        Tài khoản mới = { 'MaTaiKhoan':MaTaiKhoan, 'TaiKhoan':TaiKhoan,
        'MatKhau':MatKhau}
  
```



```

Thêm tài khoản vào CSDL
return "Thêm tài khoản khách hàng thành công"

def SuaTaiKhoan(MaTaiKhoan):
    for x in CSDL:
        if Mã tài khoản tồn tại:
            Nhập thông tin mới cho tài khoản
            return "Sửa tài khoản khách hàng thành công"
    return "Không tìm thấy tài khoản khách hàng"

def XoaTaiKhoan(MaTaiKhoan):
    for x in CSDL:
        if Mã tài khoản tồn tại:
            Xóa tài khoản khách hàng
            return "Xóa tài khoản khách hàng thành công"
    return "Không tìm thấy tài khoản khách hàng"

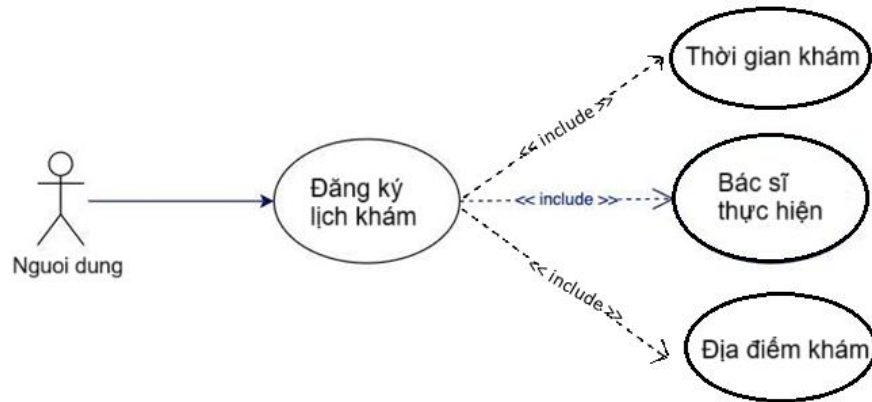
def HienThiTaiKhoan(MaTaiKhoan):
    for x in CSDL:
        if Mã tài khoản tồn tại:
            Hiển thị thông tin chi tiết tài khoản
            return "Hiển thị thành công"
        else:
            return "Không tìm thấy tài khoản khách hàng"

```

3.2.3.3. Yêu cầu chức năng đăng ký lịch khám

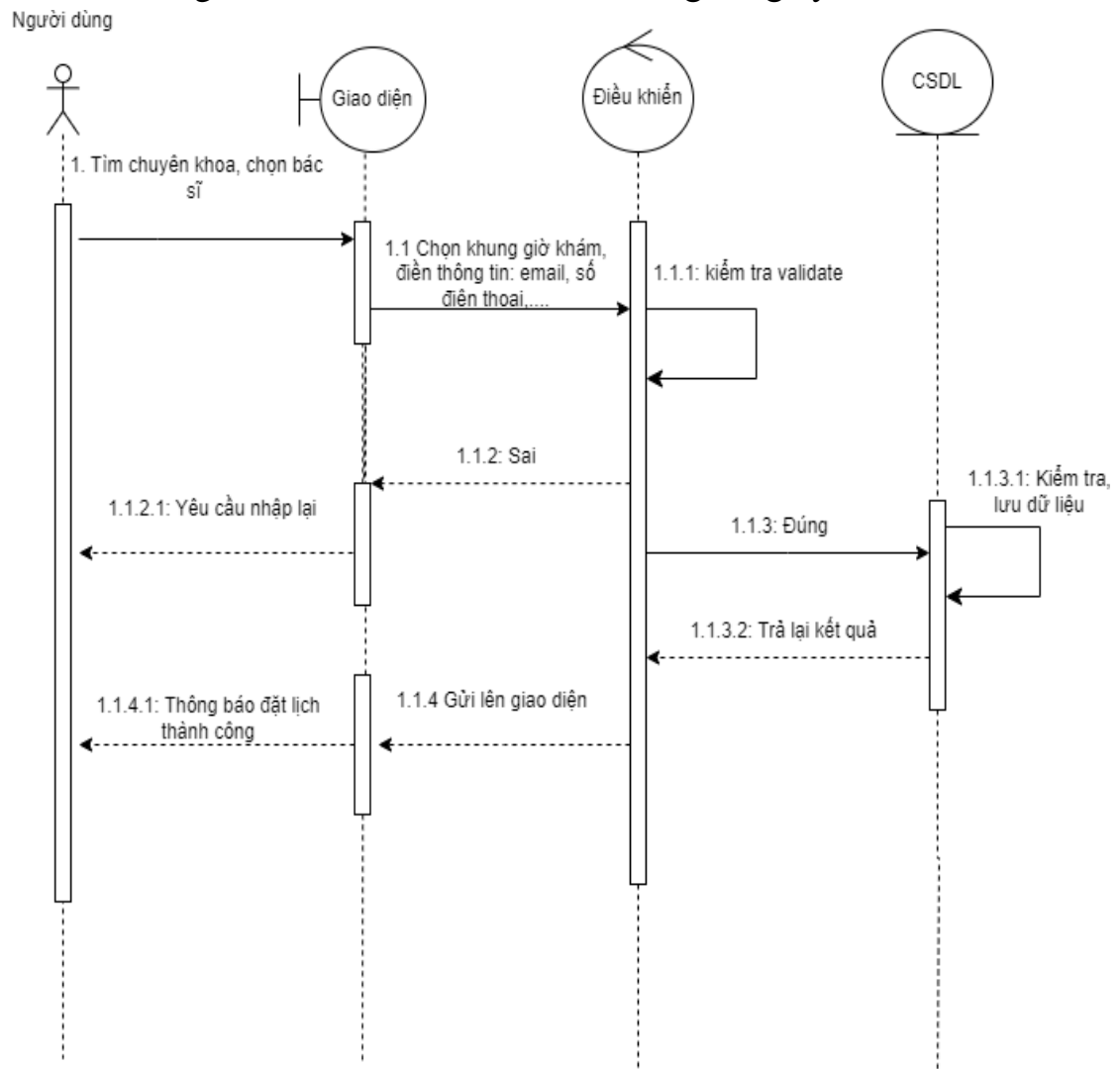
(1) Biểu đồ Use Case minh họa

Hình 3.5: Biểu đồ minh họa Use Case đăng ký lịch khám



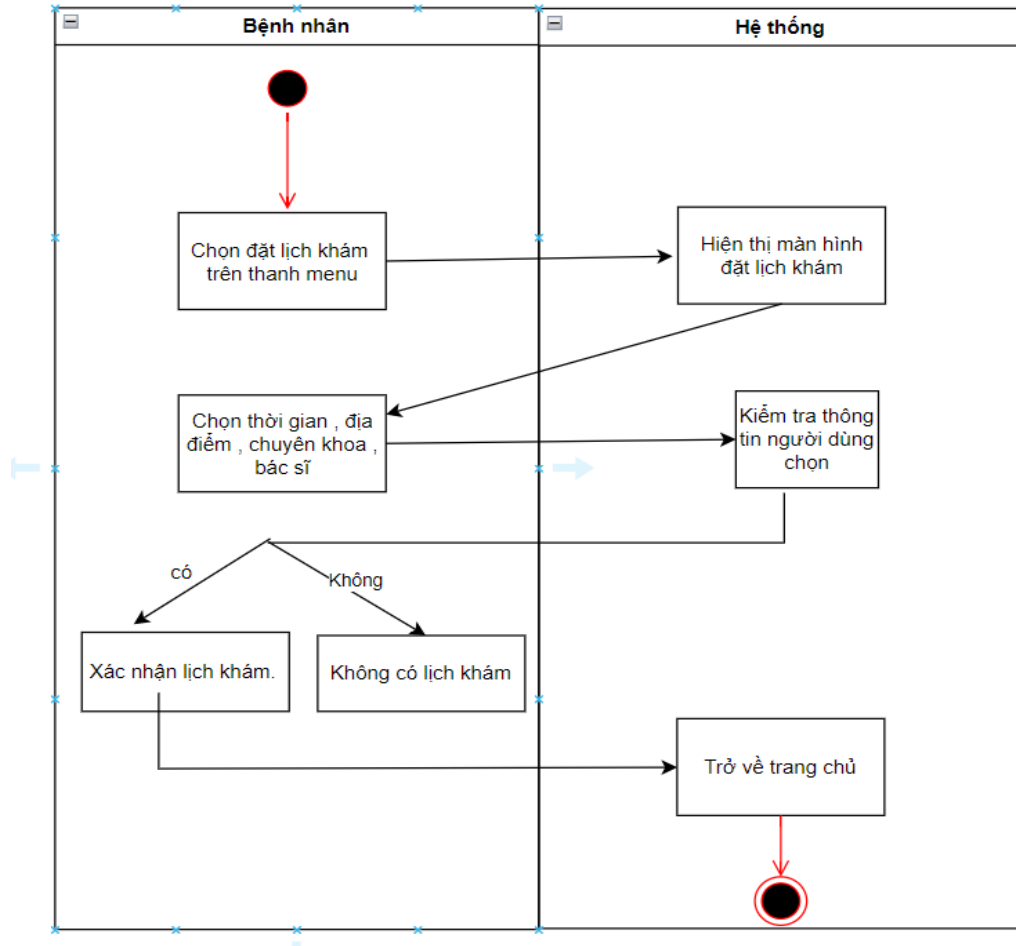
(2) Biểu đồ trình tự

Bảng 3.13: Biểu đồ trình tự chức năng đăng ký lịch khám



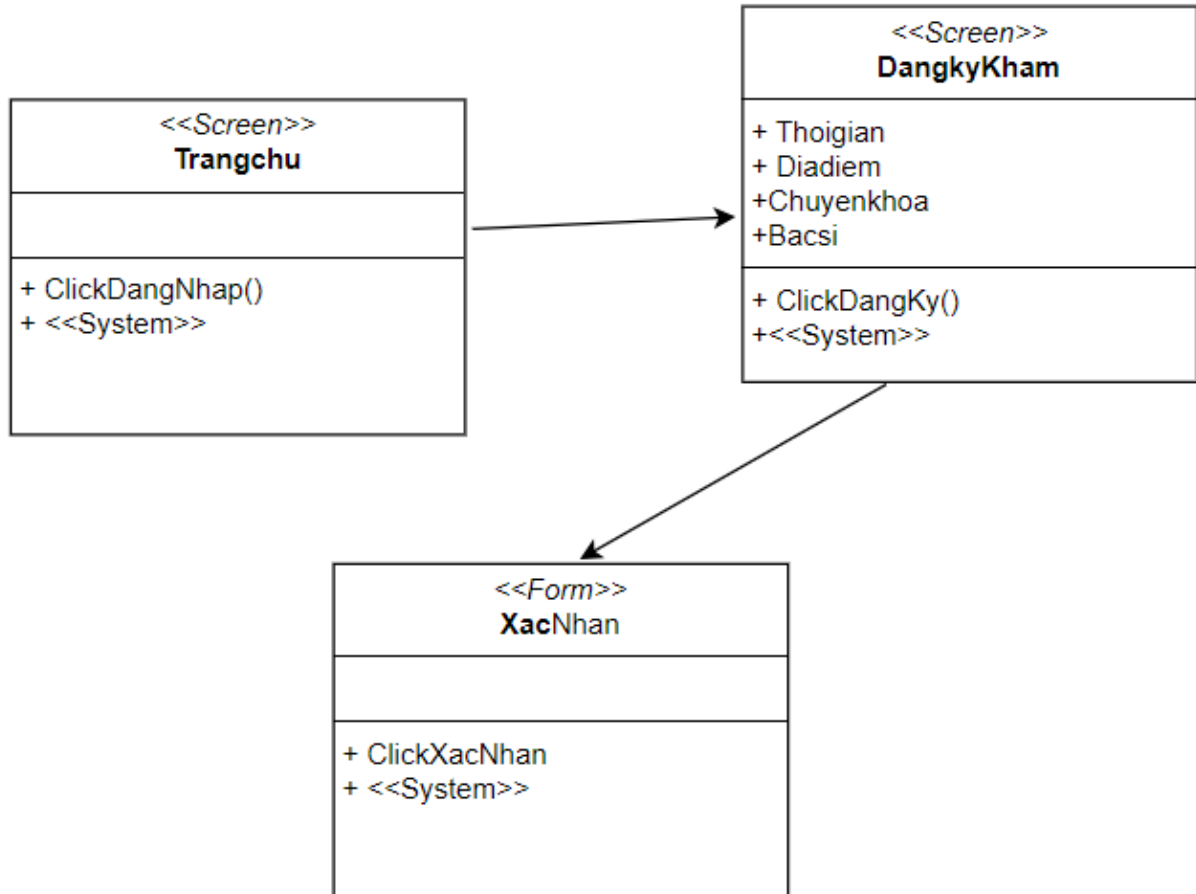
(3) Biểu đồ hoạt động

Bảng 3.14: Biểu đồ minh họa quy trình đăng ký lịch khám



(4) Biểu đồ lớp

Bảng 3.15: Biểu đồ lớp chức năng đăng ký lịch khám



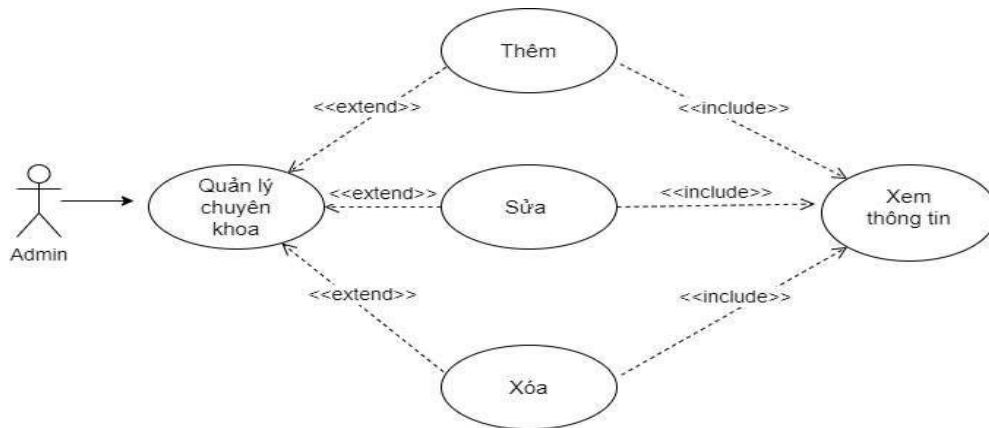
(5) Thuật toán

```

class Dangkylichkham:
    def Datlichkham(thoigian,diadiem,bacsi,chuyenkhoa):
        if (Datlichkham(thoigian,diadiem,bacsi,chuyenkhoa) )
            return “ xác nhân lịch khám “
        else:
            return “ đặt lại lịch khám “
  
```

3.2.3.4. Yêu cầu chức năng quản lý Chuyên khoa

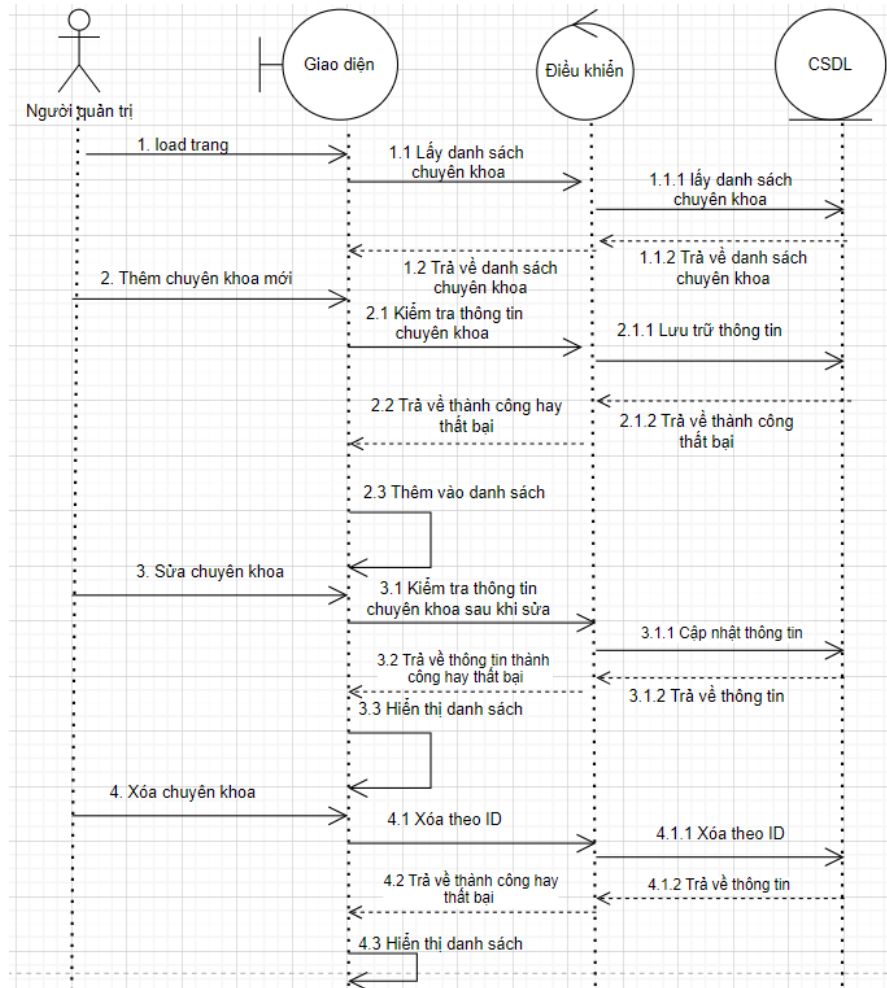
(1) Biểu đồ Use Case minh họa



Hình 3.6: Biểu đồ minh họa Use Case quản lý chuyên khoa

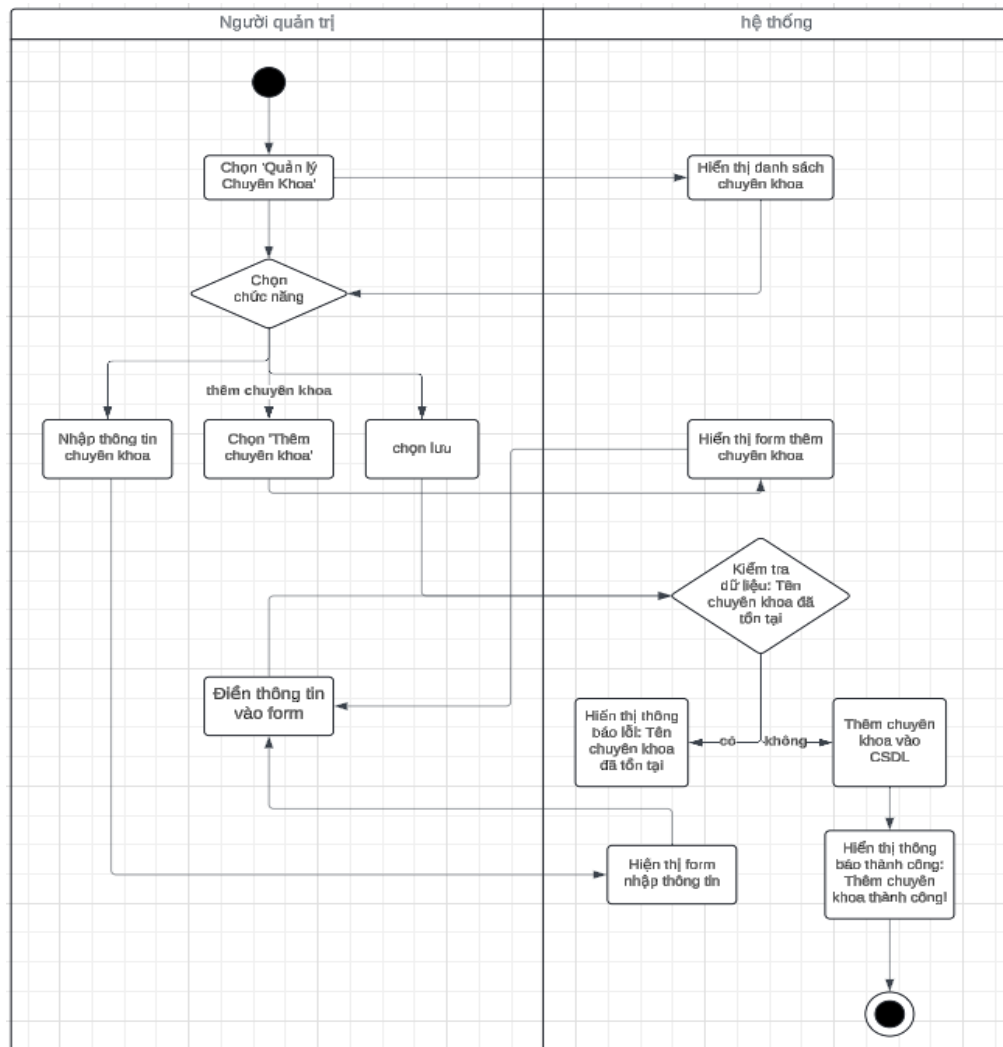
(2) Biểu đồ trình tự

Bảng 3.16: Biểu đồ trình tự chức năng quản lý chuyên khoa



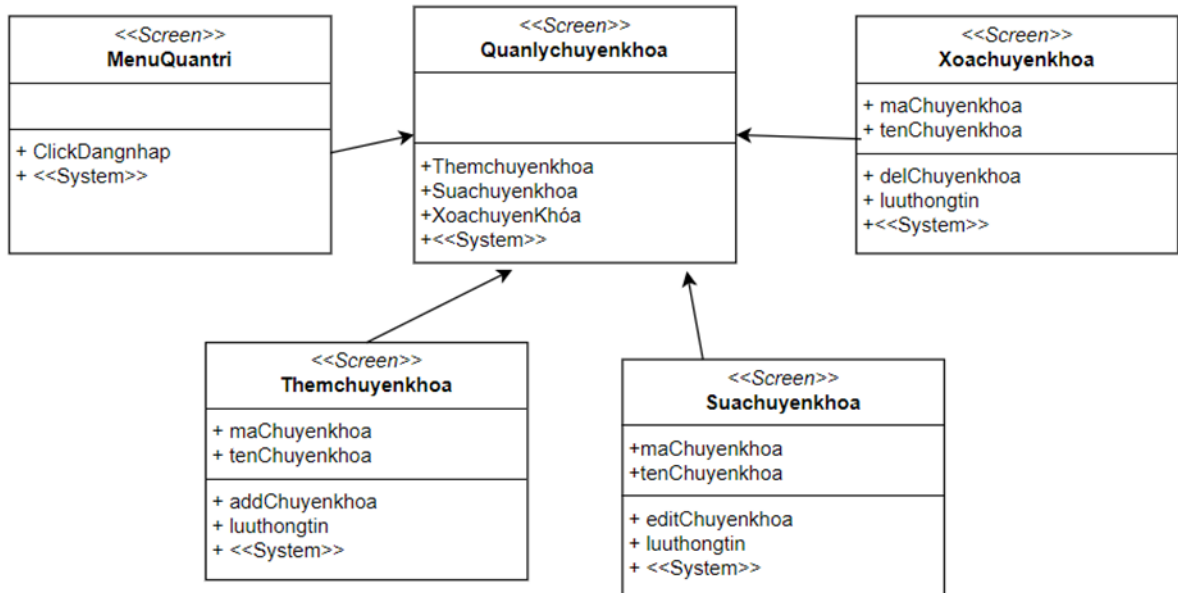
(3) Biểu đồ hoạt động

Bảng 3.17: Biểu đồ minh họa quy trình quản lý chuyên khoa



(4) Biểu đồ lớp

Bảng 3.18: Biểu đồ lớp chức năng quản lý chuyên khoa



(5) Thuật toán

```

class Quanlychuyenkhoa:
    def themChuyenKhoa(maChuyenkhoa,tenChuyenkhoa):
        if (themChuyenkhoa(maChuyenkhoa,tenChuyenkhoa)):
            return “chuyên khoa đã tồn tại!”
        else:
            return “chuyên khoa không tồn tại . có thể thêm “
    def suaChuyenKhoa(maChuyenkhoa,tenChuyenkhoa):
        if (suaChuyenkhoa(maChuyenkhoa,tenChuyenkhoa)):
            return “chuyên khoa tồn tại . có thể sửa “
        else:
            return “chuyên khoa không tồn tại “
    def xoaChuyenkhoa(maChuyenkhoa,tenChuyenkhoa):
        if (xoaChuyenkhoa(maChuyenkhoa,tenChuyenkhoa)):
            return “chuyên khoa tồn tại . có thể xóa “
        else:
            return “chuyên khoa không tồn tại “
  
```

3.2.4. Thiết kế dữ liệu

3.2.3.1 Thực thể dữ liệu người dùng

- Các thuộc tính: MaNguoiDung, HoTen, NgaySinh, QueQuan, DiaChi, SoDienThoai, Email, MaTaiKhoan.
- Các phương thức:
 - setNguoiDung(MaNguoiDung, HoTen, NgaySinh, QueQuan, DiaChi, SoDienThoai, Email): Thay đổi các thuộc tính của một người dùng.
 - deleteNguoiDung(MaNguoiDung): Xóa thông tin của người dùng
 - getNguoiDung(MaNguoiDung): Lấy thông tin chi tiết của người dùng dựa trên mã người dùng.

NGUOI_DUNG	
maNguoiDung:	char[15]
hoTen:	string
queQuan:	string
ngaySinh:	date
diaChi:	string
soDienThoai:	char[10]
email:	string
maTaiKhoan:	char[15]
setNguoiDung() getNguoiDung() deleteNguoiDung()	

Hình 3.7: Lớp thực thể người dùng

3.2.3.2 Thực thể dữ liệu chuyên khoa

- Thuộc tính: MaKhoa, TenKhoa, DiaChi, Email, MoTa
- Phương thức:
 - setChuyenKhoa(MaKhoa, TenKhoa, DiaChi, Email, MoTa): Cho phép thay đổi một số thông tin của chuyên khoa.
 - getChuyenKhoa(MaKhoa): Lấy thông tin chi tiết của một chuyên khoa theo mã khoa.

- deleteChuyenKhoa(MaKhoa): Xóa thông tin của khoa khỏi cơ sở dữ liệu theo mã khoa.

CHUYEN_KHOA	
maKhoa:	char[15]
tenKhoa:	string
diaChi:	string
email:	string
moTa:	text
setChuyenKhoa() getChuyenKhoa() deleteChuyenKhoa()	

Hình 3.8: Lớp thực thể chuyên khoa

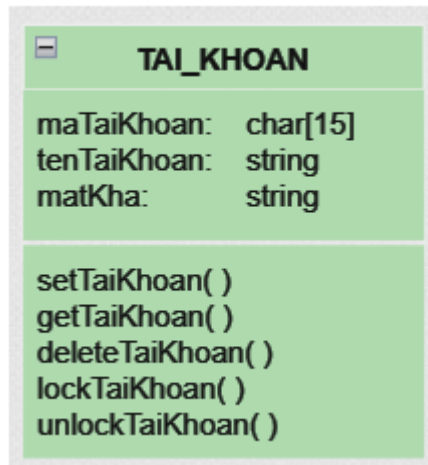
3.2.3.3 Thực thể dữ liệu tài khoản

Mô tả: Đại diện cho thông tin tài khoản của khách hàng khi họ đã đăng ký tài khoản.

Các thuộc tính: MaTaiKhoan, TenTaiKhoan, MatKhau

Các phương thức:

- setTaiKhoan(MaTaiKhoan, TenTaiKhoan, MatKhau): Thay đổi các thuộc tính của một tài khoản như mã tài khoản, tên tài khoản, và mật khẩu.
- getTaiKhoan(MaTaiKhoan): Lấy thông tin chi tiết của một tài khoản dựa trên mã tài khoản của người dùng.
- deleteTaiKhoan(MaTaiKhoan): Xóa tài khoản của người dùng có mã tài khoản tương ứng.
- lockTaiKhoan(MaTaiKhoan): Khóa tài khoản của người dùng có mã tài khoản tương ứng. (Trong khi bị khóa, người dùng không thể thực hiện một số chức năng của ứng dụng như: đặt lịch khám, tư vấn, phản hồi, ...)
- unlockTaiKhoan(MaTaiKhoan): Mở khóa tài khoản của người dùng có mã tài khoản tương ứng.



Hình 3.9: Lớp thực thể tài khoản

3.2.3.4. Thực thể dữ liệu Quản lý nhân sự

Mô tả: Đại diện cho những thông tin, chức năng liên quan đến các bác sĩ trong bệnh viện

Các thuộc tính: MaBacSi, MaBenhVien, SoLuong, KinhNghiemLamViec, HoTen, DiaChi, SoDienThoai, Email, ChucVu, TenBenhVien, DiaChiBenhVien.

Các phương thức:

- addBacSi(MaBacSi, HoTen, DiaChi, SoDienThoai, Email, ChucVu): Thêm một bác sĩ mới vào hệ thống.
- getBacSi(MaBacSi): Lấy thông tin chi tiết của một bác sĩ dựa trên mã bác sĩ.
- searchBacSi(ChucVu): Tìm kiếm các bác sĩ dựa trên chức vụ của họ.
- deleteBacSi(MaBacSi): Xóa bác sĩ ra khỏi hệ thống tương ứng với mã bác sĩ.
- updateBacSi(MaBacSi): Thay đổi các thông tin như: địa chỉ, số điện thoại, email của bác sĩ.

QUAN_LY_NHAN_SU	
maBacSi:	char[15]
maBenhVien:	string
soLuong:	int
kinhNghiemLamViec:	int
hoTen:	string
diaChi:	string
email:	string
chucVu:	string
diaChiBenhVien:	string
addBacSi() getBacSi() searchBacSi() deleteBacSi() updateBacSi()	

Hình 3.10: Lớp thực thể quản lý nhân sự

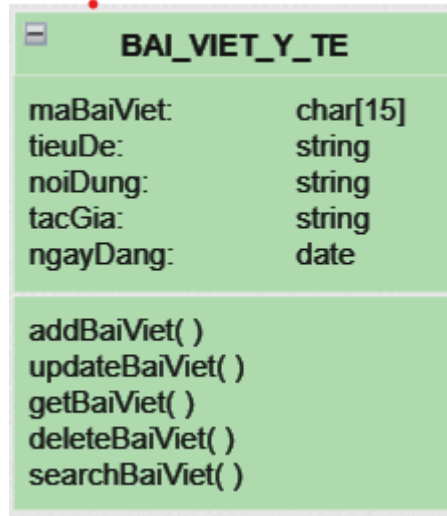
3.2.3.5. Thực thể dữ liệu bài viết

Mô tả: Đại diện cho thông tin của một bài viết y tế đã được đăng trong hệ thống.

Các thuộc tính: MaBaiViet, TieuDe, NoiDung, TacGia, NgayDang

Các phương thức:

- addBaiViet(MaBaiViet, TieuDe, NoiDung, TacGia, NgayDang): Thêm một bài viết mới vào hệ thống với các thông tin cần thiết.
- updateBaiViet(MaBaiViet): Thay đổi các thông tin: Tiêu đề, nội dung, tác giả, ngày viết của một bài viết.
- getBaiViet(MaBaiViet): Lấy thông tin chi tiết của một bài viết dựa trên mã bài viết.
- deleteBaiViet(MaBaiViet): Xóa bài viết có mã tương ứng khỏi hệ thống.
- searchBaiViet(NgayDang): Lấy ra các bài viết được đăng trong ngày được nhập



Hình 3.11: Lớp thực thể bài viết y tế

3.2.3.6. Thực thể dữ liệu Lịch Khám

Thuộc tính: MaLichKham, CoSoKham, ChuyenKhoaKham, ThoiGianKham, BacSiThucHien

Phương thức:

- setLichKham(MaLichKham, CoSoKham, ChuyenKhoaKham, ThoiGianKham, BacSiThucHien): Cho phép người dùng thay đổi cơ sở, chuyên khoa, thời gian hoặc bác sĩ thực hiện khám.
- getLichKham(MaLichKham): Lấy thông tin chi tiết về thời gian, địa điểm mà người dùng đã đặt trước đó.
- deleteLichKham(MaLichKham): Hủy lịch khám đã đặt trước đó
- addLichKham(MaLichKham, CoSoKham, ChuyenKhoaKham, ThoiGianKham, BacSiThucHien): Đặt thêm lịch khám

LICH_KHAM

maLichKham:

char[15]

coSoKham:

string

chuyenKhoaKham:

string

thoiGianKham:

date

bacSiThucHien:

string

setLichKham()
getLichKham()
deleteLichKham()
addLichKham()

Hình 3.12: Lớp thực thể lịch khám

3.2.5. Mô hình cơ sở dữ liệu mức vật lý

3.2.5.1. Thiết kế bảng

Bảng 3.19: Tác giả

Entity							
Entity Name			Table Name				
TAC GIA			TAC_GIA				
Attributes	Properties	Advanced	Indexes	Alternate Keys	Others	Relationship	Storage
Key	Name	Column Name	Datatype	Not null	Unique	Description	DDL
1	MaTacGia	MaTacGia	Int	Not null		Ma tac gia	
2	HoTen	HoTen	VarChar(50)	Not null		Ho ten tac gia	
3	NgaySinh	NgaySinh	Date	Not null		Ngay sinh tac gia	
4	QueQuan	QueQuan	VarChar(50)	Not null		Que tac gia	
5	DiaChi	DiaChi	VarChar(50)	Not null		Dia chi tac gia	
6	SoDT	SoDT	Int	Not null		So dien thoai tac gia	
7	Email	Email	VarChar(20)	Not null		Email tac gia	

Bảng 3.20: Bài viết y tế

Entity							
Entity Name			Table Name				
BAI VIET Y TE			BAI_VIET_Y_TE				
Attributes	Properties	Advanced	Indexes	Alternate Keys	Others	Relationship	Storage
Key	Name	Column Name	Datatype	Not null	Unique	Description	DDL
1	MaBaiViet	MaBaiViet	Integer	Not null		Ma bai viet	
2	TieuDe	TieuDe	VarChar(50)	Not null		Tieu de bai viet	
3	NoiDung	NoiDung	VarChar(100)	Not null		Noi dung chinh bai viet	
4	TacGia	TacGia	VarChar(20)	Not null		Tac gia bai viet	
5	NgayDang	NgayDang	Date	Not null		Ngay dang bai viet	

Bảng 3.21: Diễn đàn

Entity Name

DIEN DAN

Table Name

DIEN_DAN

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others

Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	MaTacGia	MaTacGia	Integer	Not null		
2	FK	MaBaiViet	MaBaiViet	Integer	Not null		

Bảng 3.22: Bác sĩ

Entity

Entity Name

BAC SI

Table Name

BAC_SI

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others

Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	Primary	MaBacSi	MaBacSi	Int	Not null		Ma bac si
2		HoTen	HoTen	VarChar(30)	Not null		Ho ten bac si
3		QueQuan	QueQuan	VarChar(60)	Not null		Que quan bac si
4		DiaChi	DiaChi	VarChar(60)	Not null		Dia chi bac si
5		SoDienThoai	SoDienThoai	Int	Not null		So dien thoai bac si
6		Email	Email	VarChar(50)	Not null		Email bac si
7	US	MaKhoa	MaKhoa	Int	Not null		

Bảng 3.23: Bệnh viện

Entity Name

BENH VIEN

Table Name

BENH_VIEN

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others


Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1		MaBenhVien	MaBenhVien	Integer	Not null		Ma benh vien
2		TenBenhVien	TenBenhVien	Varchar(50)	Not null		Ten benh vien
3		DiaChi	DiaChi	Varchar(60)	Not null		Dia chi benh vien
4		SoDienThoai	SoDienThoai	Integer	Not null		So dien thoai benh vien
5		Email	Email	Varchar(50)	Not null		Email benh vien

Bảng 3.24: Quản lý nhân sự

Entity Name

QUAN LY NHAN SU

Table Name

QUAN_LY_NHAN_SU

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others

Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	MaBacSi	MaBacSi	Integer	Not null		
2	UK	MaKhoa	MaKhoa	Integer	Not null		
3	UK	MaBenhVien	MaBenhVien	Integer	Not null		
4		KinhNghiemLamViec	KinhNghiemLamViec	Varchar(50)	Not null		Kinh nghiem lam viec
5		SoLuongNhanSu	SoLuongNhanSu	Integer	Not null		So luong nhan su

Bảng 3.25: Chuyên khoa

Entity Name

CHUYEN KHOA

>

Table Name

CHUYEN_KHOA

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others


Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1		MaKhoa	MaKhoa	Integer	Not null		Ma khoa
2		TenKhoa	TenKhoa	Varchar(50)	Not null		Ma khoa
3		DiaChi	DiaChi	Varchar(60)	Not null		Dia chi khoa
4		Email	Email	Varchar(30)	Not null		Dia chi khoa
5		MoTa	MoTa	Text	Not null		Mo ta

Bảng 3.26: Dịch vụ chăm sóc

Entity Name

DICH VU CHAM SOC

Table Name

DICH_VU_CHAM_SOC

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others


Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1		MaDichVu	MaDichVu	Integer	Not null		Ma dich vu
2		TenDichVu	TenDichVu	Varchar(60)	Not null		Ten dich vu
3		DichVu	DichVu	Varchar(50)	Not null		Dich vu

Bảng 3.27: Quản lý khoa

Entity Name

QUAN LY KHOA

Table Name

QUAN_LY_KHOA

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others



Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1		MaKhoa	MaKhoa	Integer	Not null		
2		MaBenhVien	MaBenhVien	Integer	Not null		

Bảng 3.28: Dịch vụ bệnh viện

Entity Name

DICH VU BENH VIEN

>

Table Name

DICH_VU_BENH_VIEN

Attributes

Properties

Advanced

Indexes

Alternate Keys

Others

Relationship

Storage

Comments

Notes

DDL

	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	MaBenhVien	MaBenhVien	Integer	Not null		
2		MaDichVu	MaDichVu	Integer	Not null		
3		SoLuongDichVu	SoLuongDichVu	Integer	Not null		So luong dich vu

Bảng 3.29: Lịch khám

Entity

Entity Name: LICH_KHAM Table Name: LICH_KHAM

Key	Name	Column Name	Datatype	Not null	Unique	Description
1	MaLichKham	MaLichKham	Int	Not null		Ma lịch khám
2	CoSoKham	CoSoKham	VarChar(30)	Not null		Thông tin cơ bản cơ sở khám
3	ChuyenKhoaKham	ChuyenKhoaKham	VarChar(30)	Not null		Chuyên khoa khám
4	ThoiGianKham	ThoiGianKham	Date	Not null		Thời gian khám
5	BacSiThucHienKh	BacSiThucHienKham	VarChar(50)	Not null		Thông tin cơ bản bác sĩ thực hiện
6	MaNguoiDung	MaNguoiDung	Int	Not null		
7	MaBenhVien	MaBenhVien	Int	Not null		

Bảng 3.3: Bảng người dùng

Entity

Entity Name: NGUOI_DUNG Table Name: NGUOI_DUNG

Key	Name	Column Name	Datatype	Not null	Unique	Description
1	MaNguoiDung	MaNguoiDung	Int	Not null		Ma người dùng
2	HoTen	HoTen	VarChar(30)	Not null		Họ tên người dùng
3	NgaySinh	NgaySinh	Date	Not null		Ngày sinh người dùng
4	QueQuan	QueQuan	VarChar(50)	Not null		Quê quán người dùng
5	DiaChi	DiaChi	VarChar(50)	Not null		Địa chỉ người dùng
6	SoDienThoai	SoDienThoai	Int	Not null		Số điện thoại người dùng
7	Email	Email	VarChar(30)	Not null		Email người dùng
8	MaTaiKhoan	MaTaiKhoan	Int	Not null		

Bảng 3.31: Bảng tài khoản

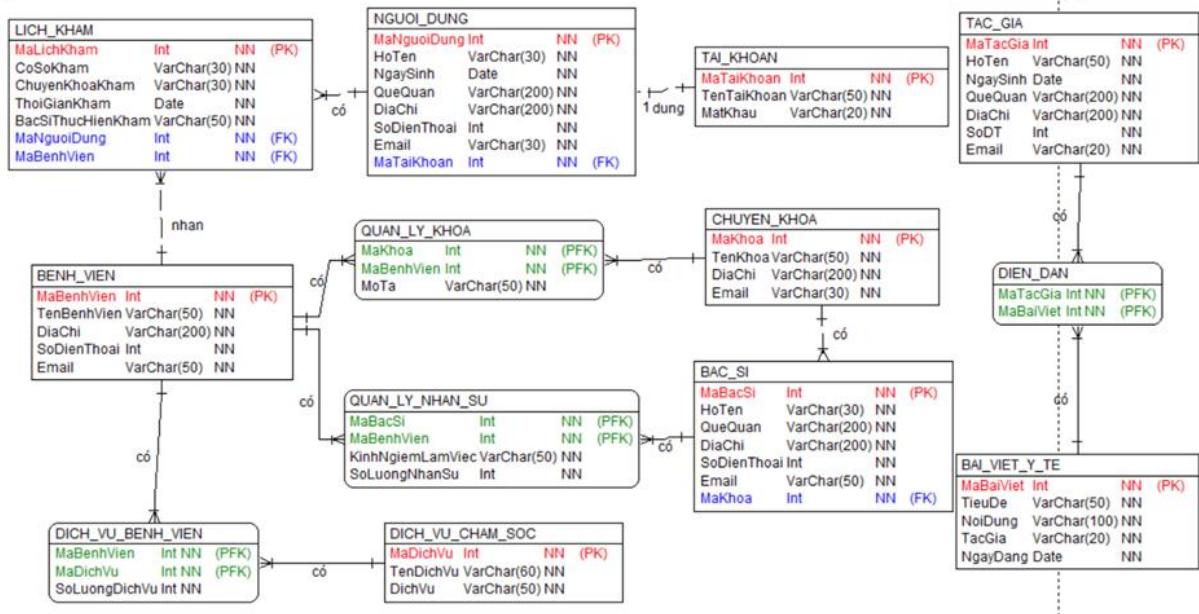
Entity

Entity Name: TAI KHOAN Table Name: TAI_KHOAN

Key	Name	Column Name	Datatype	Not null	Unique	Description
1	MaTaiKhoan	MaTaiKhoan	Int	Not null		Ma tài khoản
2	TenTaiKhoan	TenTaiKhoan	VarChar(50)	Not null		Tên tài khoản
3	MatKhau	MatKhau	VarChar(20)	Not null		Mật khẩu tài khoản

3.2.5.2. Biểu đồ thực thể liên kết mức vật lý.

Bảng 3.32: Biểu đồ thực thể liên kết mức vật lý



3.2.5. Thiết kế giao diện người sử dụng

3.2.5.1. Mô tả tổng quan

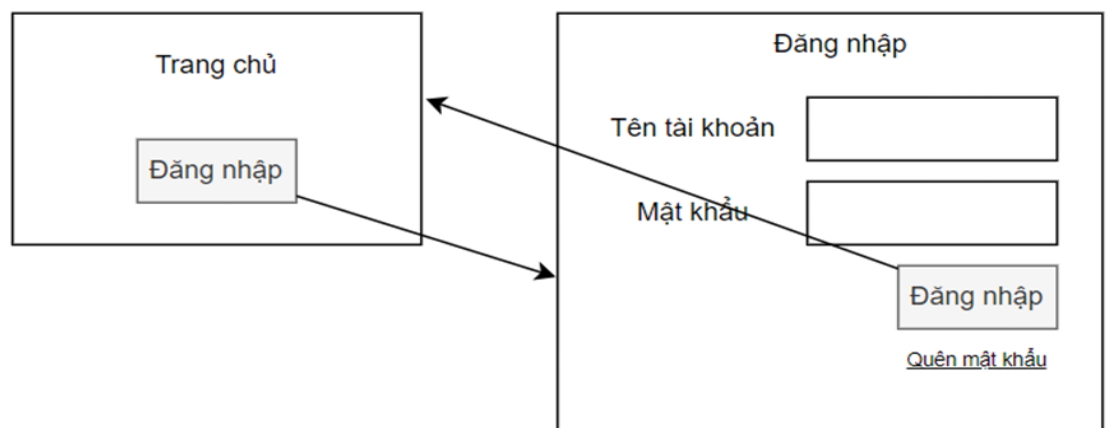
Bảng 3.33: Mô tả tổng quan giao diện người dùng

Chức năng	Người dùng	Hệ thống
Đăng nhập	<ul style="list-style-type: none"> Nhập email và mật khẩu đã đăng ký vào các ô tương ứng trên trang đăng nhập. Nhấn nút "Đăng nhập". 	<ul style="list-style-type: none"> Đăng nhập thành công: Hệ thống chuyển hướng người dùng đến trang chủ hoặc trang quản lý tài khoản cá nhân. Đăng nhập thất bại: Hiện thị thông báo lỗi "Email hoặc mật khẩu không chính xác. Vui lòng thử lại."
Quản lý tài khoản	<ul style="list-style-type: none"> Nhấn vào biểu tượng hoặc mục "Quản lý tài khoản" trong menu. Cập nhật các thông tin như tên, địa chỉ, số điện thoại, và mật khẩu. Ấn "Lưu" để thông tin được cập nhật 	<ul style="list-style-type: none"> Cập nhật thành công: Thông báo "Thông tin đã được cập nhật thành công." Cập nhật thất bại: Thông báo lỗi cụ thể tùy thuộc vào vấn đề gặp phải (ví dụ: "Số điện thoại không hợp lệ").
Xem lịch sử khám	<ul style="list-style-type: none"> Chọn mục "Thông tin khám" từ menu. 	Hiện thị thông tin chi tiết về cơ sở y tế bao gồm: địa chỉ, số điện thoại, các dịch vụ cung cấp, và bản đồ chỉ đường.

	- Danh sách các lịch sử khám hiện ra, người dùng chọn một lịch khám cụ thể để xem chi tiết.	
Xem chi tiết chuyên khoa	- Chọn mục "Chuyên khoa" từ menu chính. - Danh sách các chuyên khoa hiện ra, người dùng chọn một chuyên khoa để xem chi tiết.	Hiển thị thông tin chi tiết về chuyên khoa bao gồm: mô tả, các dịch vụ chuyên khoa cung cấp, và danh sách các bác sĩ thuộc chuyên khoa đó.
Bảo trì chuyên khoa	- Chọn mục "Bảo trì chuyên khoa" từ menu của người quản trị. - Xem danh sách các chuyên khoa hiện có. - Chọn chuyên khoa cụ thể để xem chi tiết hoặc chỉnh sửa thông tin. - Nhấn "Thêm chuyên khoa mới" để thêm chuyên khoa mới.	- Hiển thị danh sách các chuyên khoa gồm: Mã chuyên khoa, tên chuyên khoa, mô tả. - Cho phép người dùng chọn chuyên khoa để xem chi tiết hoặc chỉnh sửa. - Hiển thị nút "Thêm chuyên khoa mới".
Quản lý lịch khám	- Chọn mục "Lịch khám" từ menu của người quản trị. - Danh sách các lịch khám hiện ra, chọn một lịch khám để xem chi tiết.	Hiển thị thông tin chi tiết về lịch khám bao gồm: Mã lịch khám, thời gian khám, cơ sở khám, chuyên khoa khám, bác sĩ thực hiện.

3.2.5.2. Hình ảnh giao diện người dùng

(1) Chức năng đăng nhập



Hình 3.13: Giao diện màn hình đăng nhập

Tài khoản

 Mật khẩu

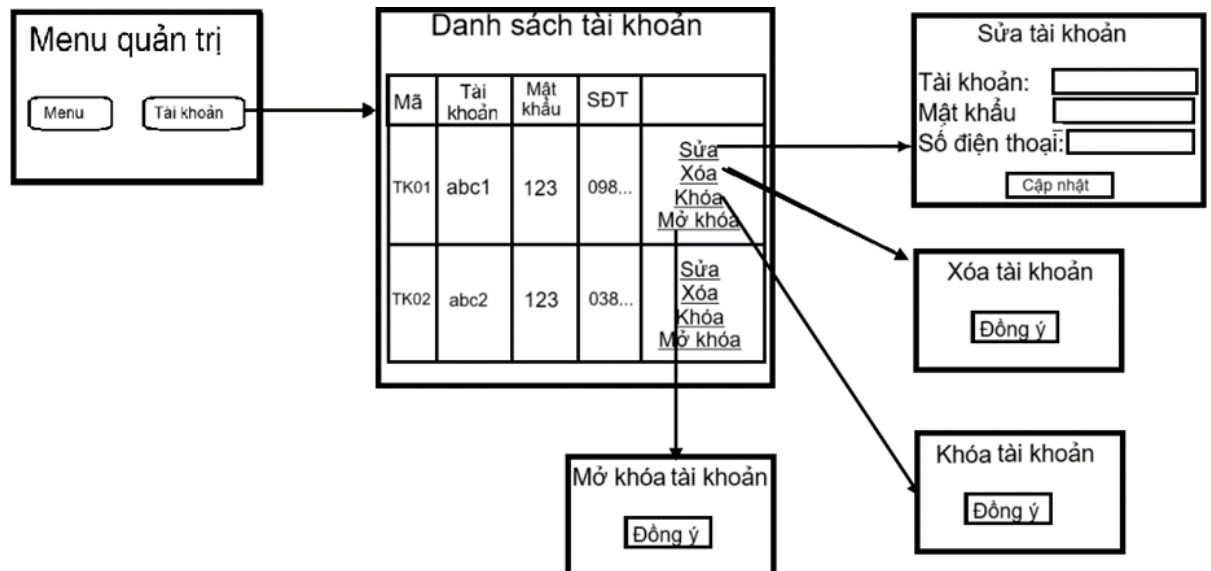
 Đăng nhập

LỖI
 Thông tin tài khoản hoặc
 mật khẩu của bạn không
 chính xác
 Đăng nhập

Đăng nhập thành công

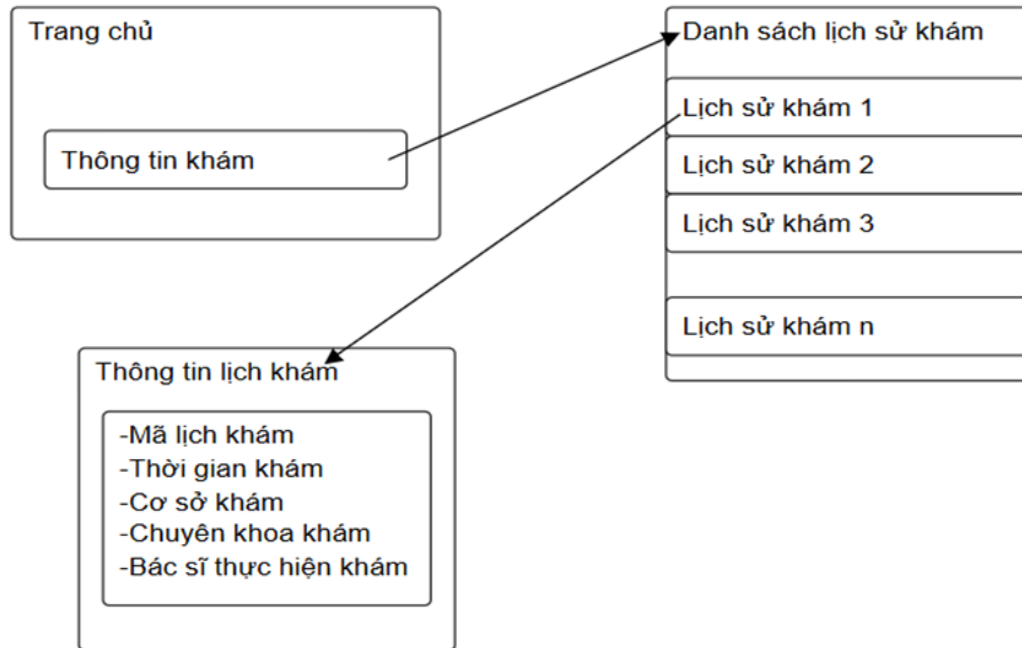
Hình 3.14: Giao diện màn hình đăng nhập (Rẽ nhánh)

(2) Quản lý tài khoản



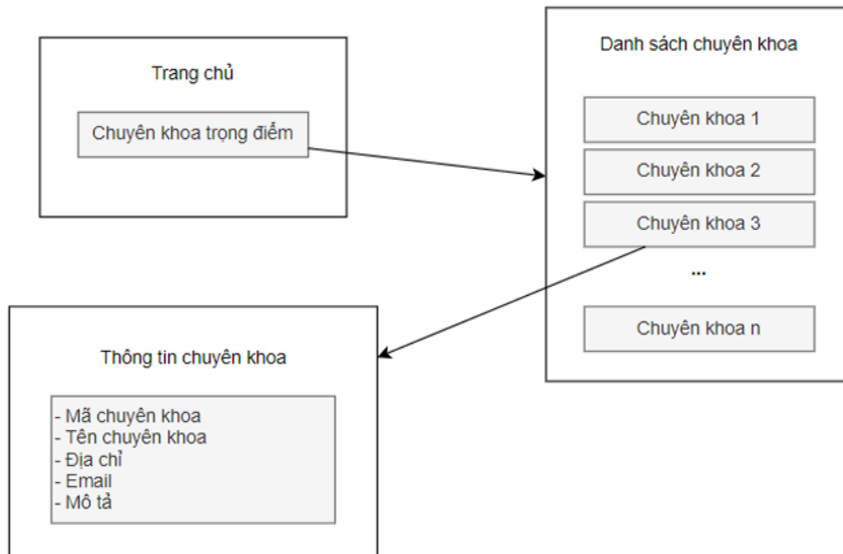
Hình 3.15: Giao diện màn hình quản lý tài khoản

(3) Xem lịch sử khám



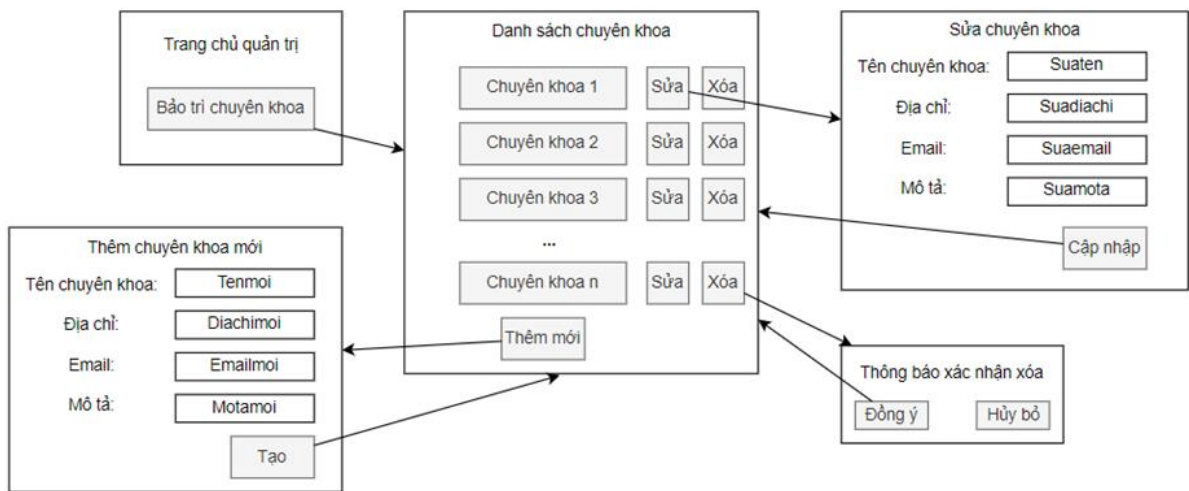
Hình 3.16: Giao diện màn hình xem lịch khám

(4) Xem chi tiết các chuyên khoa



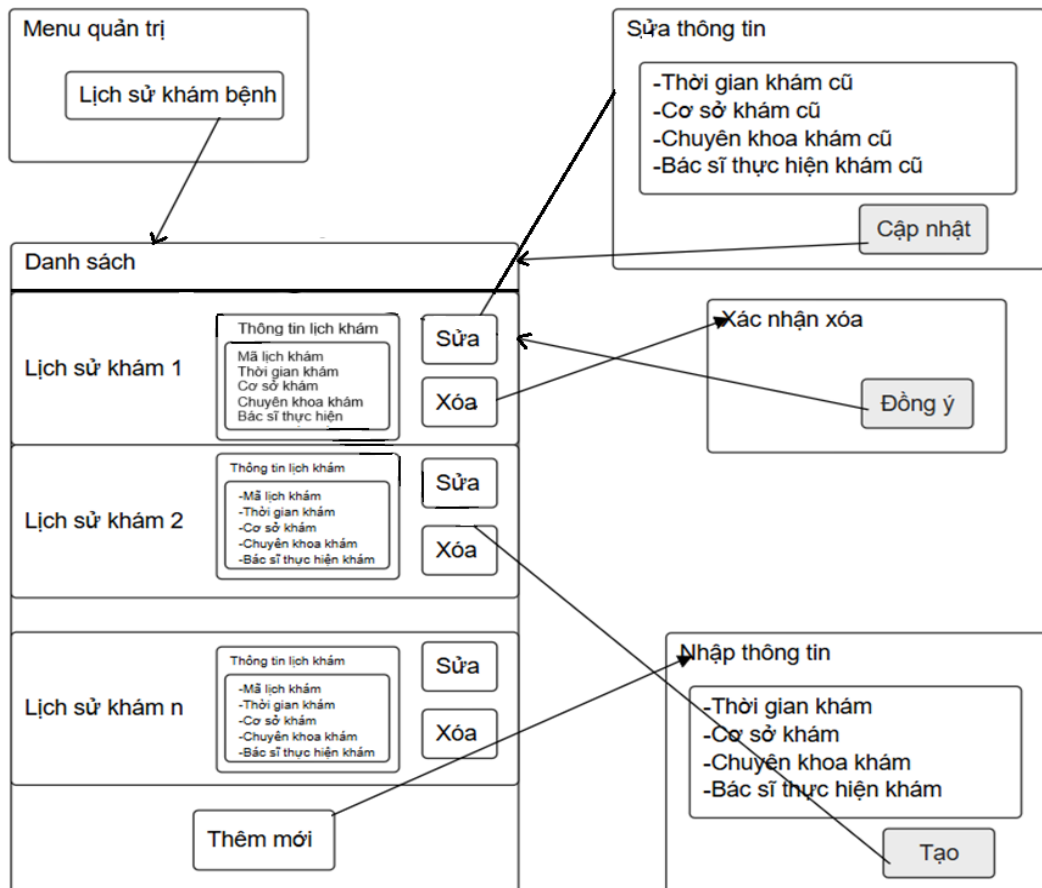
Hình 3.17: Giao diện màn hình xem các chuyên khoa

(5) Bảo trì chuyên khoa



Hình 3.15: Giao diện màn hình bảo trì chuyên khoa

(6) Quản lý lịch khám



Hình 3.16: Giao diện màn hình quản lý lịch khám

3.3. Phần kết luận chương

Trong chương III, chúng ta đã xem xét quá trình thiết kế phần mềm từ các góc độ khác nhau, bao gồm việc lựa chọn mô hình kiến trúc, phân tích yêu cầu, thiết kế giao diện người dùng và các phương pháp thiết kế phần mềm khác. Chúng ta đã thấy rằng thiết kế phần mềm không chỉ là bước đầu tiên và quan trọng trong quá trình phát triển phần mềm, mà còn đóng vai trò quyết định đối với chất lượng, hiệu suất và sự linh hoạt của hệ thống cuối cùng.

Qua việc áp dụng các phương pháp và kỹ thuật thiết kế phần mềm hiện đại, chúng ta có thể tạo ra các hệ thống phần mềm phản ánh chính xác nhu cầu của khách hàng và người dùng cuối. Bằng cách tập trung vào việc thiết kế phần mềm theo cách mà tối ưu hóa sự linh hoạt, dễ bảo trì và mở rộng, chúng ta có thể đảm bảo rằng sản phẩm phần mềm của chúng ta sẽ có tuổi thọ cao và sẽ có khả năng thích nghi với các yêu cầu thay đổi trong tương lai.

KẾT LUẬN

Trong suốt quá trình thực hiện đề tài, nhóm chúng em đã nỗ lực không ngừng để hoàn thành dự án phần mềm một cách hoàn hảo và chính chu nhất. Dự án của chúng em bao gồm việc giới thiệu tổng quan về dự án phần mềm, các công cụ, kỹ thuật và phương pháp phát triển phần mềm. Chúng em cũng đã giới thiệu chi tiết về đề tài, áp dụng mô hình thác nước và giải thích rõ ràng quy trình mô hình này.

Trong phần phân tích, chúng em đã trình bày về quy trình kỹ thuật yêu cầu và các tác nhân tham gia, từ đó phân tích đặc tả yêu cầu phần mềm cho ứng dụng thăm khám y tế Vinmec. Yêu cầu phần mềm này bao gồm yêu cầu về dữ liệu lưu trữ, yêu cầu chức năng và phi chức năng, nhằm đảm bảo rằng phần mềm đáp ứng đầy đủ các tiêu chí đặt ra.

Về hoạt động thiết kế phần mềm, chúng em đã nêu rõ vai trò và nhiệm vụ của từng thành viên trong nhóm. Đặc biệt, chúng em đã đặc tả thiết kế kiến trúc phần mềm, thiết kế cấu trúc phần mềm, thiết kế cơ sở dữ liệu và thiết kế giao diện người dùng cuối cho ứng dụng thăm khám y tế Vinmec. Qua đó, nhóm đã tích lũy được nhiều kỹ năng quan trọng như kỹ năng thành lập và quản lý nhóm, làm việc nhóm, quản lý tiến độ công việc, kỹ năng trình bày báo cáo, tìm kiếm và đọc hiểu tài liệu tham khảo. Bên cạnh đó, chúng em đã sử dụng thành thạo các công cụ hỗ trợ như Microsoft Word, Google Docs, Creately, MySQL.

Chúng em đã luôn cố gắng để hoàn thiện đề tài một cách tốt nhất, nhưng bởi vì sự giới hạn về thời gian và kiến thức nên nhiều mục còn sơ sài, chưa cụ thể và còn nhiều thiếu sót. Nhóm chúng em rất mong nhận được những đánh giá và góp ý từ thầy cô để giúp đề tài được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

- [1] Hoàng Quang Huy, Phùng Đức Hoà, Trịnh Bá Quý (2016), Giáo trình Nhập môn công nghệ phần mềm, Đại học Công nghiệp Hà Nội, Hà Nội.
- [2] Hoàng Văn Minh, Nguyễn Thị Thùy Dung (2018). "Thiết kế giao diện người dùng trong phát triển phần mềm" - Nhà xuất bản Đại học Quốc gia Hà Nội
- [3] Fowler, M. (2003), UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley Professional, Boston.
- [4] Fowler, M. (2003), UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley Professional, Boston.
- [5] Pressman, R. S. (2015), Software Engineering: A Practitioner's Approach, McGraw-Hill Education, New York.
- [6] Sommerville, I. (2016), Software Engineering, Pearson Education, London.
- [7]”<https://www.studocu.com/vn/document/truong-dai-hoc-cong-nghiep-ha-noi/nhap-mon-cong-nghe-phan-mem/bai-tap-lon-nhap-mon-cong-nghe-phan-mem/66630123> “ - (Truy cập lần cuối ngày 14/4/2024)
- [8]”<https://tailieuhust.com/nhap-mon-cong-nghe-phan-mem-hust/>” - Tài liệu HUST – (Truy cập lần cuối ngày 31/5/2024)
- [9]<https://www.studocu.com/vn/course/truong-dai-hoc-cong-nghiep-ha-noi/nhap-mon-cong-nghe-phan-mem/5274241> - (Truy cập lần cuối ngày 15/5/2024)