

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU
TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐÒ ÁN TỐT NGHIỆP
ĐỀ TÀI:
XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH
TẠI PHÒNG KHÁM BÌNH ĐỊNH

Giảng viên hướng dẫn : ThS. TRẦN THỊ DUNG
Sinh viên thực hiện : NGUYỄN CAO AN
Lớp : CÔNG NGHỆ THÔNG TIN
Khóa : 61

Thành Phố Hồ Chí Minh, năm 2024

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU
TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI:
XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH
TẠI PHÒNG KHÁM BÌNH ĐỊNH

Giảng viên hướng dẫn : ThS. TRẦN THỊ DUNG
Sinh viên thực hiện : NGUYỄN CAO AN
Lớp : CÔNG NGHỆ THÔNG TIN
Khóa : 61

Thành Phố Hồ Chí Minh, năm 2024

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

Mã sinh viên: 6151071032

Họ tên SV : Nguyễn Cao An

Khóa : 61

Lớp : Công nghệ thông tin

1. Tên đề tài:

XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH TẠI PHÒNG KHÁM BÌNH ĐỊNH

2. Mục đích, yêu cầu

a. Mục đích:

- Xây dựng Website đặt lịch khám bệnh tại phòng khám Bình Định giúp cho bệnh nhân và đội ngũ y bác sĩ thuận tiện hơn trong quá trình khám chữa bệnh.

b. Yêu cầu:

- Xây dựng trang web quản lý phòng khám
- Đáp ứng được các chức năng phù hợp với thực tiễn
- Giao diện thân thiện dễ sử dụng

3. Nội dung và phạm vi đề tài

a. Nội dung đề tài:

- Tổng quan đề tài
- Phân tích thiết kế hệ thống
- Xây dựng giao diện hệ thống
- Kiểm thử và kết luận

b. Phạm vi nghiên cứu

- Nghiên cứu Reactjs, Nodejs, MySQL và một số framework
- Xây dựng trang web quản lý

4. Công nghệ, công cụ và ngôn ngữ lập trình

a. Công nghệ sử dụng:

- ReactJS
- NodeJS
- ExpressJS
- RESTful API
- MySQL

b. Công cụ:

- Visual Studio Code
- MySQL Workbench
- Trình duyệt

c. Ngôn ngữ lập trình

- HTML
- CSS

- JavaScript

5. Giảng viên hướng dẫn:

c. Họ và tên : Trần Thị Dung

d. Đơn vị công tác : Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao
Thông Vận Tải phân hiệu TP.HCM

e. Điện thoại : 0388389579 - Email: ttdung@utc2.edu.vn

Ngày ... tháng 3 năm 2024
Trưởng BM Công Nghệ Thông Tin

Đã giao nhiệm vụ TKTN
Giảng viên hướng dẫn

ThS. Trần Phong Nhã

ThS. Trần Thị Dung

Đã nhận nhiệm vụ TKTN

Sinh viên: Nguyễn Cao AN

Điện thoại: 0915570872

Ký tên:

Email: 6151071032@st.utc2.edu.vn

LỜI CẢM ƠN

Kính gửi Ban giám hiệu Trường Đại học Giao thông Vận tải Phân hiệu tại Thành phố Hồ Chí Minh, cùng quý thầy cô Bộ môn Công nghệ Thông tin,

Lời đầu tiên, em muốn bày tỏ lòng biết ơn sâu sắc tới Ban giám hiệu cùng toàn thể quý thầy cô đã tạo điều kiện cho em được học tập và thực hiện đề tài tốt nghiệp "**Xây dựng Website đặt lịch khám bệnh tại phòng khám Bình Định.**" Trong quãng thời gian thực hiện đồ án, em đã có cơ hội khám phá và trải nghiệm nhiều điều mới mẻ, mà nếu không có sự hỗ trợ của quý thầy cô, em sẽ không thể hoàn thành được.

Về khoảng thời gian thực hiện đề tài, đó là những ngày tháng mà em không chỉ tập trung vào việc phát triển kỹ năng chuyên môn mà còn học cách làm việc, quản lý thời gian, và giải quyết vấn đề. Những thử thách không thể tránh khỏi đã trở thành những cơ hội để em rèn luyện bản thân. Nhưng điều tuyệt vời hơn cả là sự hướng dẫn tận tình của quý thầy cô đã luôn ở bên em, giúp em vượt qua những khó khăn đó. Quý thầy cô không chỉ là những người dạy dỗ, mà còn là những người truyền cảm hứng và động lực cho em.

Em muốn gửi lời cảm ơn đặc biệt đến cô ThS. Trần Thị Dung, người đã trực tiếp hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề tài. Cô Dung không chỉ là một người hướng dẫn tận tâm mà còn là một người truyền cảm hứng và luôn sẵn sàng lắng nghe mọi câu hỏi, thắc mắc của em. Cô đã dành rất nhiều thời gian để giúp em hiểu rõ các khái niệm, đưa ra những gợi ý quý báu, và hướng dẫn em qua từng giai đoạn của đề tài. Sự kiên nhẫn và lòng nhiệt huyết của cô là điều em luôn ghi nhớ và biết ơn.

Tuy nhiên, em hiểu rằng do thời gian có hạn, việc thực hiện đề tài không tránh khỏi những sai sót và hạn chế. Em hy vọng quý thầy cô sẽ thông cảm và dành thời gian gộp ý để em có thể hoàn thiện đề tài một cách tốt nhất. Những lời khuyên và sự hướng dẫn của quý thầy cô là vô cùng quý báu đối với em, không chỉ trong việc hoàn thiện đề tài mà còn là những bài học để em mang theo suốt đời.

Một lần nữa, em xin gửi lời cảm ơn chân thành nhất tới Ban giám hiệu, quý thầy cô Bộ môn Công nghệ Thông tin, và đặc biệt là cô ThS. Trần Thị Dung. Em chúc quý thầy cô luôn mạnh khỏe, hạnh phúc, và thành công trong sự nghiệp giáo dục. Mong rằng quý thầy cô sẽ tiếp tục truyền cảm hứng và dẫn dắt các thế hệ sinh viên tiếp theo, giống như những gì mà quý thầy cô đã làm cho em.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIẢNG VIÊN

TP. Hồ Chí Minh, ngày ... tháng 6 năm 2024
Giảng viên hướng dẫn

ThS. Trần Thị Dung

MỤC LỤC

MỞ ĐẦU	1
1. Tổng quan về đề tài:	1
2. Mục tiêu nghiên cứu:	1
3. Đối tượng:.....	2
4. Phạm vi nghiên cứu:	2
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	3
1.1. Tổng quan về HTML, SCSS.....	3
1.1.1. Khái niệm về HTML	3
1.1.2. Vai trò của HTML	3
1.1.3. Khái niệm về CSS	3
1.2. Tổng quan về ReactJS	4
1.2.1. Khái niệm về ReactJS	4
1.2.2. Components.....	4
1.2.3. Props và State	4
1.3. Tổng quan về NodeJS với Express FrameWork.....	4
1.3.1. Khái niệm về NodeJS	4
a. Các đặc tính của NodeJS:.....	5
b. Cách hoạt động NodeJS:	5
c. NPM: The Node Package Manager:.....	5
d. Ví dụ về NodeJS:.....	6
1.3.2. Khái niệm về ExpressJS	7
a. Cấu trúc của ExpressJS	8
b. Router trong Express	8
c. Các method all của router.....	9
d. Router.METHOD()	9
1.4. Tổng quan về RESTful API.....	10
1.4.1. RESTful API	10
1.4.2. Các thành phần	10
1.4.3. Cách thức hoạt động	11

1.4.4. Authentication và dữ liệu trả về	11
1.5. Tổng quan về MySQL	11
1.5.1. Khái niệm về MySQL	11
1.5.2. Các thành phần chính của SQL	11
1.5.3. Mô hình hoạt động của MySQL.....	12
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	13
2.1. Giới thiệu bài toán	13
2.2. Sơ đồ Use Case	13
2.2.1. Use Case tổng quát	13
2.2.2. Use Case bệnh nhân	14
2.2.3. Use Case bác sĩ.....	14
2.2.4. Use Case trưởng khoa	15
2.2.5. Use Case quản trị viên.....	15
2.3. Kịch bản Use Case chi tiết:.....	16
2.3.1. Use case tạo tài khoản	16
2.3.2. Use case thay đổi mật khẩu	16
2.3.3. Use case đặt lịch khám bệnh	17
2.3.4. Use Case quản lý tài khoản	17
2.3.5. Use Case quản lý chuyên khoa.....	18
2.3.6. Use Case quản lý thuốc	18
2.3.7. Use Case quản lý sự kiện	19
2.3.8. Use Case quản lý nội quy hệ thống	19
2.3.9. Use Case quản lý thông báo	20
2.3.10. Use Case phân lịch trực	20
2.4. Trạng thái chi tiết.....	21
2.4.1. Trạng thái tạo tài khoản.....	21
2.4.2. Trạng thái quên mật khẩu.....	22
2.4.3. Trạng thái đặt lịch khám bệnh.....	23
2.4.4. Trạng thái quản lý tài khoản	24
2.4.4. Trạng thái thông kê.....	25
2.4.5. Trạng thái quản lý chuyên khoa	26
2.4.6. Trạng thái quản lý thuốc.....	27
2.4.7. Trạng thái quản lý sự kiện.....	28
2.4.8. Trạng thái quản lý nội quy hệ thống	29

2.4.9. Trạng thái phân lịch trực	30
2.5. Sơ đồ phân rã chức năng.....	31
2.5.1. Sơ đồ phân rã chức năng của bệnh nhân	31
2.5.2. Sơ đồ phân rã chức năng của bác sĩ	32
2.5.3. Sơ đồ phân rã chức năng của quản trị viên	32
2.5.4. Sơ đồ phân rã chức năng của trưởng khoa	33
2.6. Sơ đồ cơ sở dữ liệu	33
2.7. Sơ đồ ERD	34
CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH.....	35
3.1. Phía người dùng.....	35
3.1.1. Giao diện trang chủ	35
3.1.2. Giao diện đăng nhập.....	35
3.1.3. Giao diện tạo tài khoản.....	36
3.1.4. Giao diện quên mật khẩu.....	36
3.1.5. Giao diện trang cá nhân.....	37
3.1.6. Giao diện thay đổi thông tin cá nhân.....	38
3.1.7. Giao diện đặt lịch khám bệnh.....	38
3.2. Về phía quản trị viên.....	39
3.2.1. Giao diện đăng nhập.....	39
3.2.2. Giao diện thống kê	40
3.2.3. Giao diện quản lý chuyên khoa	41
3.2.4. Giao diện quản lý khu vực	42
3.2.5. Giao diện quản lý dịch vụ	43
3.2.6. Giao diện quản lý bác sĩ	44
3.2.7. Giao diện quản lý thuốc	45
3.2.8. Giao diện quản lý sự kiện, tin tức	46
3.2.9. Giao diện quản lý nội quy hệ thống	47
3.2.10. Giao diện quản lý thông báo người dùng	48
3.3. Về phía bác sĩ.....	49
3.3.1. Giao diện đăng nhập.....	49
3.3.2. Giao diện quản lý khám bệnh.....	49
3.4. Về phía trưởng khoa	51
3.4.1. Giao diện phân lịch trực	51
KẾT LUẬN VÀ KIẾN NGHỊ.....	52

1.	Kết quả đạt được:.....	52
2.	Hạn chế:	52
3.	Hướng phát triển.....	52
	PHỤ LỤC	53
	Link Github.....	53
	Hướng dẫn cài đặt.....	53
	Tài liệu tham khảo	53

DANH MỤC BẢNG BIÊU

Bảng 2.1. Trạng thái tạo tài khoản	21
Bảng 2.2. Trạng thái quên mật khẩu.....	22
Bảng 2.3. Trạng thái đặt lịch khám	23
Bảng 2.4. Trạng thái quản lý tài khoản	24
Bảng 2.5. Trạng thái thống kê	25
Bảng 2.6. Trạng thái quản lý chuyên khoa.....	26
Bảng 2.7. Trạng thái quản lý thuốc	27
Bảng 2.8. Trạng thái quản lý sự kiện.....	28
Bảng 2.9. Trạng thái quản lý nội quy hệ thống	29
Bảng 2.10. Trạng thái phân lịch trực	30

DANH MỤC HÌNH ẢNH

Ảnh 1.1. Ví dụ SCSS	4
Ảnh 1.2. Ví dụ về NodeJS	7
Ảnh 1.3. Cấu trúc của Express	8
Ảnh 1.4. Thiết kế của một REST API	10
Ảnh 1.5. Quy trình và cách thức ứng dụng tương tác với khách hàng thông qua API	11
Ảnh 1.6. Mô hình được sử dụng trong MySQL	12
Ảnh 2.1. Use case tổng quát	13
Ảnh 2.2. Use Case bệnh nhân	14
Ảnh 2.3. Use Case bác sĩ	14
Ảnh 2.4. Use Case trưởng khoa	15
Ảnh 2.5. Use Case quản trị viên	15
Ảnh 2.6. Use case tạo tài khoản	16
Ảnh 2.7. Use case thay đổi mật khẩu	16
Ảnh 2.8. Use Case đặt lịch khám bệnh	17
Ảnh 2.9. Use Case quản lý tài khoản	17
Ảnh 2.10. Use Case quản lý chuyên khoa	18
Ảnh 2.11. Use Case quản lý thuốc	18
Ảnh 2.12. Use Case quản lý sự kiện	19
Ảnh 2.13. Use Case quản lý nội quy hệ thống	19
Ảnh 2.14. Use Case quản lý thông báo	20
Ảnh 2.15. Use Case phân lịch trực	20
Ảnh 2.16. Biểu đồ trạng thái tạo tài khoản	21
Ảnh 2.17. Biểu đồ trạng thái quên mật khẩu	22
Ảnh 2.18. Biểu đồ trạng thái đặt lịch khám bệnh	23
Ảnh 2.19. Biểu đồ trạng thái quản lý tài khoản	24
Ảnh 2.20. Biểu đồ trạng thái thông kê	25
Ảnh 2.21. Biểu đồ trạng thái quản lý chuyên khoa	26
Ảnh 2.22. Biểu đồ trạng thái quản lý thuốc	27
Ảnh 2.23. Biểu đồ trạng thái quản lý sự kiện	28
Ảnh 2.24. Biểu đồ trạng thái quản lý nội quy hệ thống	29
Ảnh 2.25 Biểu đồ trạng thái phân lịch trực	30
Ảnh 2.26. Sơ đồ phân rã chức năng của bệnh nhân	31
Ảnh 2.27. Sơ đồ phân rã chức năng của bác sĩ	32

Ảnh 2.28. Sơ đồ phân rã chức năng của quản trị viên.....	32
Ảnh 2.29. Sơ đồ phân rã chức năng của trưởng khoa	33
Ảnh 2.30. Sơ đồ cơ sở dữ liệu	33
Ảnh 2.31. Sơ đồ ERD	34
Ảnh 3.1. Giao diện trang chủ.....	35
Ảnh 3.2. Giao diện trang chủ.....	35
Ảnh 3.3. Giao diện đăng nhập phía người dùng.....	36
Ảnh 3.4. Giao diện tạo tài khoản	36
Ảnh 3.5. Giao diện quên mật khẩu	37
Ảnh 3.6. Giao diện trang cá nhân	37
Ảnh 3.7. Giao diện cập nhật thông tin cá nhân	38
Ảnh 3.8. Giao diện chi tiết bác sĩ	38
Ảnh 3.9. Giao diện đặt lịch khám bệnh	39
Ảnh 3.10. Giao diện đăng nhập phía quản trị viên	39
Ảnh 3.11. Giao diện thống kê.....	40
Ảnh 3.12. Giao diện quản lý tài khoản.....	40
Ảnh 3.13. Giao diện thêm tài khoản.....	41
Ảnh 3.14. Giao diện quản lý chuyên khoa	41
Ảnh 3.15. Giao diện tạo chuyên khoa mới	42
Ảnh 3.16. Giao diện quản lý khu vực.....	42
Ảnh 3.17. Giao diện thêm khu vực mới	43
Ảnh 3.18. Giao diện quản lý loại dịch vụ.....	43
Ảnh 3.19. Giao diện thêm mới loại dịch vụ mới	44
Ảnh 3.20. Giao diện quản lý dịch vụ.....	44
Ảnh 3.21. Giao diện quản lý bác sĩ	44
Ảnh 3.22. Giao diện thêm mới bác sĩ	45
Ảnh 3.23. Giao diện quản lý thuốc	45
Ảnh 3.24. Giao diện thêm mới thuốc	46
Ảnh 3.25. Giao diện quản lý sự kiện, tin tức	46
Ảnh 3.26. Giao diện thêm sự kiện, tin tức.....	47
Ảnh 3.27. Giao diện quản lý nội quy hệ thống.....	47
Ảnh 3.28. Giao diện thêm mới nội quy hệ thống	48
Ảnh 3.29. Giao diện quản lý thông báo gửi qua email.....	48
Ảnh 3.30. Giao diện gửi thông báo qua email người dùng	49

Ảnh 3.31 Giao diện đăng nhập phía quản trị viên.....	49
Ảnh 3.32. Giao diện quản lý bệnh nhân khám bệnh	50
Ảnh 3.33. Giao diện cập nhật lịch đặt khám	50
Ảnh 3.34. Hóa đơn khám bệnh.....	50
Ảnh 3.35. Giao diện phân lịch trực	51
Ảnh 3.36. Giao diện tạo lịch trực	51

DANH MỤC VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	HTML	Hypertext Markup Language	
2	CSS	Cascading Style Sheets	
3	JSX	JavaScript XML	
4	JS	JavaScript	
5	HTTP	Hyper Text Transfer Protocol	
6	NPM	Node package manager	
7	API	Application Programming Interface	
8	URL	Uniform Resource Locator	
9	ERD	Entity – Relationship Diagram	

MỞ ĐẦU

1. Tổng quan về đề tài:

Trong thời đại công nghệ ngày nay, sự tiện lợi và hiệu quả của việc sử dụng công nghệ thông tin trong quản lý và cung cấp dịch vụ y tế không thể phủ nhận. Đặc biệt, việc xây dựng một trang web quản lý không chỉ giúp tối ưu hóa quy trình làm việc mà còn nâng cao chất lượng phục vụ cho bệnh nhân.

Đề tài "Xây dựng Website đặt lịch khám bệnh tại phòng khám Bình Định" là một nỗ lực nhằm tận dụng sức mạnh của công nghệ để cải thiện hệ thống quản lý trong lĩnh vực y tế. Trang web này sẽ được thiết kế và phát triển để đáp ứng các nhu cầu cụ thể, từ việc quản lý hồ sơ bệnh nhân, lịch hẹn, tài liệu y tế, đến việc quản lý tài nguyên và nhân sự.

Mục tiêu của dự án là tạo ra một giao diện dễ sử dụng, linh hoạt và an toàn, giúp các nhân viên y tế và quản lý có thể tiếp cận thông tin một cách nhanh chóng và chính xác. Bên cạnh đó, trang web cũng sẽ mang lại lợi ích lớn cho bệnh nhân bằng cách cung cấp thông tin đầy đủ về các dịch vụ y tế, giúp họ dễ dàng đặt lịch hẹn, tra cứu hồ sơ y tế và tương tác trực tuyến với các bác sĩ.

Với sự kết hợp giữa kiến thức chuyên môn về y tế và kỹ năng lập trình và thiết kế web, đề tài này hứa hẹn mang lại những giải pháp hiệu quả và tiện ích cho cả phòng khám và cộng đồng người dùng. Đồng thời, nó cũng là một bước tiến quan trọng trong việc hiện đại hóa ngành y tế và cải thiện chất lượng dịch vụ y tế cho cộng đồng.

2. Mục tiêu nghiên cứu:

Phát triển và triển khai một hệ thống trang web chuyên biệt dành cho việc quản lý, nhằm cải thiện hiệu quả hoạt động và chất lượng dịch vụ y tế. Cụ thể, các mục tiêu nghiên cứu được đề ra như sau:

- a. Thiết kế giao diện trực quan và dễ sử dụng: Phát triển một giao diện người dùng thân thiện, dễ hiểu và dễ sử dụng cho cả nhân viên y tế và bệnh nhân, giúp họ dễ dàng tương tác với hệ thống.
- b. Quản lý thông tin bệnh nhân hiệu quả: Xây dựng các tính năng quản lý hồ sơ bệnh nhân, ghi chú y tế, lịch sử khám và điều trị, giúp nhân viên y tế có thể tra cứu và cập nhật thông tin một cách nhanh chóng và chính xác.
- c. Tối ưu hóa quy trình lịch hẹn và xếp lịch: Phát triển các tính năng đặt lịch hẹn trực tuyến, quản lý lịch làm việc của bác sĩ và nhân viên y tế, từ đó tối ưu hóa thời gian và nguồn lực.
- d. Đảm bảo bảo mật thông tin: Xây dựng các biện pháp bảo mật thông tin nhạy cảm của bệnh nhân, đảm bảo rằng dữ liệu y tế được lưu trữ và truy cập một cách an toàn và riêng tư.
- e. Tích hợp các tính năng tiện ích cho bệnh nhân: Phát triển các tính năng tra cứu thông tin y tế, đặt lịch hẹn, gửi yêu cầu và phản hồi trực tuyến, nhằm nâng cao trải nghiệm của người dùng cuối.

Bằng việc đạt được những mục tiêu nghiên cứu này, đề tài hy vọng đóng góp vào việc nâng cao chất lượng phục vụ y tế và tạo ra một hệ thống quản lý hiệu quả và tiện ích cho cả nhân viên y tế và bệnh nhân.

3. Đối tượng:

Bác sĩ và Y tá: Những người sử dụng chính của hệ thống, họ sẽ sử dụng website để quản lý thông tin bệnh nhân, lập lịch hẹn, xem kết quả xét nghiệm và chẩn đoán, ghi chú về điều trị và thăm khám.

Bệnh nhân: Đối tượng này sẽ sử dụng website để đặt lịch hẹn, tra cứu thông tin và các dịch vụ cung cấp, xem kết quả xét nghiệm và gửi phản hồi về chất lượng dịch vụ.

Quản lý: Bao gồm các quản lý cấp cao trong, họ sẽ sử dụng hệ thống để theo dõi hoạt động hàng ngày, quản lý tài nguyên nhân lực và vật lý, và phân tích dữ liệu để đưa ra quyết định chiến lược.

Nhân viên hỗ trợ kỹ thuật: Đối tượng này sẽ có trách nhiệm duy trì và hỗ trợ kỹ thuật cho hệ thống website, đảm bảo rằng nó hoạt động một cách liên tục và an toàn.

Quản trị viên hệ thống: Người quản trị hệ thống sẽ có trách nhiệm quản lý và bảo mật dữ liệu, cung cấp quyền truy cập cho người dùng và giải quyết các vấn đề kỹ thuật liên quan đến hệ thống.

4. Phạm vi nghiên cứu:

Phân tích nhu cầu và yêu cầu của các bên liên quan: Nghiên cứu sẽ tập trung vào việc thu thập thông tin từ các bác sĩ, y tá, bệnh nhân và quản lý để hiểu rõ về nhu cầu và yêu cầu cụ thể của họ đối với một hệ thống quản lý trực tuyến.

Thiết kế và phát triển hệ thống: Nghiên cứu sẽ tập trung vào quá trình thiết kế và phát triển website, bao gồm cả việc lựa chọn công nghệ phù hợp, giao diện người dùng, cơ sở dữ liệu, và tính bảo mật của hệ thống.

Tích hợp các chức năng quản lý: Phạm vi này sẽ tập trung vào việc tích hợp các chức năng quản lý như quản lý bệnh án điện tử, quản lý lịch hẹn, đặt, quản lý tài nguyên nhân lực, và quản lý thuốc, sắp xếp lịch trực.

Đảm bảo tính bảo mật và tuân thủ quy định: Nghiên cứu, xem xét các biện pháp để đảm bảo tính bảo mật của thông tin bệnh nhân

Đánh giá và kiểm tra: Sau khi triển khai, nghiên cứu sẽ tiến hành đánh giá hiệu suất, đánh giá tính hiệu quả và cải tiến của hệ thống.

Xem xét các ứng dụng tương lai: Nghiên cứu cũng có thể mở rộng để xem xét các ứng dụng tương lai của hệ thống.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về HTML, SCSS

1.1.1. Khái niệm về HTML

HTML hay HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ được sử dụng cho các tài liệu web.

Trên thực tế, đây không phải là một ngôn ngữ lập trình, HTML giống như một ngôn ngữ xác định đâu là ý nghĩa, mục đích và cấu trúc của một tài liệu. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các website. [1]

1.1.2. Vai trò của HTML

Một website thường chứa nhiều trang con. Mỗi trang con sẽ có một tập tin HTML riêng. Dù lập trình bằng bất cứ ngôn ngữ nào, trên bất cứ Framework nào, khi chạy trên nền website, chúng đều được biên dịch ra ngôn ngữ HTML. Đa số các trình soạn thảo văn bản trên website đều có 2 chế độ xem: HTML và văn bản thường.

Chức năng chính của HTML là xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. HTML thường được dùng để phân chia các đoạn văn, heading, links, blockquotes,... [2]

Với HTML, chúng ta có thể:

- Thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển.
- Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết.
- Xây dựng bảng, điều khiển một số kiểu mẫu

HTML là ngôn ngữ markup, dễ học, dễ hiểu, dễ áp dụng. Tuy nhiên, một website được viết bằng HTML rất đơn giản, nếu không muốn nói là nhảm chán. Để gây hứng thú với người truy cập, website cần có sự hỗ trợ của CSS và JavaScript.

Nếu HTML cung cấp cấu trúc cơ bản của các trang web, CSS và JavaScript sẽ cải tiến và sửa đổi website sinh động hơn.

1.1.3. Khái niệm về CSS

SCSS là một chương trình tiền xử lý CSS (CSS preprocessor). Nó giúp viết CSS theo cách của một ngôn ngữ lập trình, có cấu trúc rõ ràng, rành mạch, dễ phát triển và bảo trì code hơn. Ngoài ra nó có rất nhiều các thư viện hỗ trợ kèm theo giúp viết code CSS một cách dễ dàng vào đơn giản hơn. SCSS sử dụng cú pháp giống với Ruby (vì đơn giản nó được thiết kế bởi các lập trình viên Ruby). Có phần mở rộng là .scss , SCSS ra đời sau SASS và có cú pháp viết tương tự như cách viết CSS. Cú pháp này được tạo ra nhằm thu hẹp khoảng cách giữa SASS và CSS bằng cách mang lại một thứ gì đó thân thiện với CSS. Trong hình sau:

```

# styles.scss x
# styles.scss > .header > &__inner
1 $gray-f3: #f3f3f3;
2
3 .header {
4   color: $gray-f3;
5
6   &__inner {
7     border: 1px solid $gray-f3;
8     transform: translate(50%, -20%);
9   }
10 }
11

# styles.css x
# styles.css > ...
1 .header {
2   color: #f3f3f3;
3 }
4 .header__inner {
5   border: 1px solid #f3f3f3;
6   -webkit-transform: translate(50%, -20%);
7   |   |   |   transform: translate(50%, -20%);
8 }
9
10 /*# sourceMappingURL=styles.css.map */

```

Ảnh 1.1. Ví dụ SCSS

1.2. Tổng quan về ReactJS

1.2.1. Khái niệm về ReactJS

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa. [3]

1.2.2. Components

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

1.2.3. Props và State

Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. State: thể hiện trạng thái của ứng dụng, khi state thay đổi, component đồng thời render lại để cập nhật UI.

State: là một object có thể được sử dụng để chứa dữ liệu hoặc thông tin về components. State có thể được thay đổi bất cứ khi nào mong muốn. Khác với props bạn có thể truyền props sang các components khác nhau, state chỉ tồn tại trong phạm vi của components chứa nó, mỗi khi state thay đổi components đó sẽ được render lại.

1.3. Tổng quan về NodeJS với Express FrameWork

1.3.1. Khái niệm về NodeJS

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền

tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất. [10]

a. Các đặc tính của NodeJS:

- Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (non-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
- Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
- Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo ra hạn chế xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
- Không đệm: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
- Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.

b. Cách hoạt động NodeJS:

Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống, việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.

c. NPM: The Node Package Manager:

Khi thảo luận về Node js, một điều chắc chắn không nên bỏ qua là xây dựng package quản lý sử dụng các công cụ NPM mà mặc định với mọi cài đặt Node js. Ý tưởng của mô-đun NPM là khá tương tự như Ruby-Gems: một tập hợp các hàm có sẵn có thể sử dụng được, thành phần tái sử dụng, tập hợp các cài đặt dễ dàng thông qua kho lưu trữ trực tuyến với các phiên bản quản lý khác nhau.

Danh sách các mô-đun có thể tìm trên web NPM package hoặc có thể truy cập bằng cách sử dụng công cụ NPM CLI sẽ tự động cài đặt với Node js.

Một số các module NPM phổ biến nhất hiện nay là:

- expressjs.com/ - Express.js, một Sinatra-inspired web framework khá phát triển của Node.js, chứa rất nhiều các ứng dụng chuẩn của Node.js ngày nay.

- connect - Connect là một mở rộng của HTTP server framework cho Node.js, cung cấp một bộ sưu tập của hiệu suất cao "plugins" được biết đến như là trung gian; phục vụ như một nền tảng cơ sở cho Express
- socket.io and sockjs - Hai thành phần Server-side websockets components nổi tiếng nhất hiện nay.
- Jade - Một trong những engines mẫu, lấy cảm hứng từ HAML, một phần mặc định trong Express.js.
- mongo and mongojs - MongoDB hàm bao để cung cấp các API cho cơ sở dữ liệu đối tượng trong MongoDB Node.js
- redis - thư viện Redis client.
- coffee-script - CoffeeScript trình biên dịch cho phép developers viết các chương trình Node.js của họ dùng Coffee.
- underscore (lodash, lazy) - Thư viện tiện ích phổ biến nhất trong JavaScript, package được sử dụng với Node.js, cũng như hai đối tác của mình, hứa hẹn hiệu suất tốt hơn bằng cách lấy một cách tiếp cận thực hiện hơi khác nhau.
- forever - Có lẽ là tiện ích phổ biến nhất để đảm bảo rằng một kịch bản nút cho chạy liên tục. Giữ quá trình Node.js lên trong sản xuất đối mặt với bất kỳ thất bại không ngờ tới.

d. Ví dụ về NodeJS:

Chat thời gian thực, ứng dụng đa người dùng điển hình nhất, thông qua nhiều giao thức độc quyền và mở chạy trên các cổng không chuẩn, với khả năng thực hiện tất cả mọi thứ ngày hôm nay trong Node.js với WebSockets chạy qua cổng chuẩn 80.

Các ứng dụng chat thực sự là ví dụ điển hình nhất của Node.js: đó là một ứng dụng trao đổi thông tin nhanh chóng, lưu lượng truy cập cao, dữ liệu chuyên sâu (nhưng / xử lý tính toán thấp) ứng dụng chạy trên các thiết bị phân phối. Nó cũng là một trường hợp tuyệt vời cho học tập, vì nó đơn giản, nhưng nó bao gồm hầu hết các mô hình mà từng được sử dụng trong một ứng dụng Node.js điển hình.

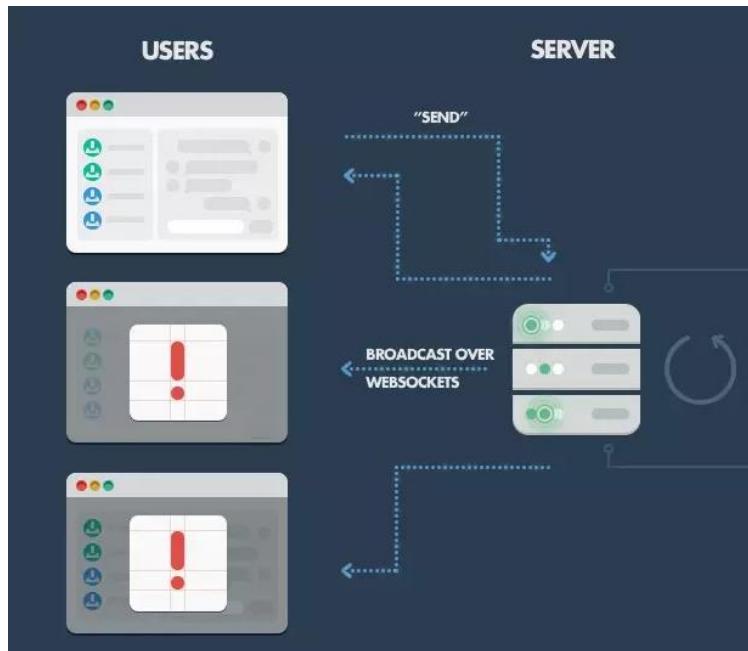
Trong ví dụ đơn giản nhất, có một phòng chat đơn trên trang web, nơi mọi người đến và có thể trao đổi một hoặc nhiều tin nhắn (trên thực tế tất cả). Ví dụ, có ba người trên tất cả các trang web kết nối với bảng tin.

Trên phía máy chủ, chúng ta có một ứng dụng đơn giản Express.js mà thực hiện hai điều:

- GET '/' xử lý yêu cầu phục vụ các trang web có chứa cả một bảng thông báo và một nút "Send" để khởi đầu vào tin nhắn mới.
- Một máy chủ WebSockets mà nghe cho thư mới được phát ra bởi các client WebSocket. Bên phía Client, chúng ta có một trang HTML với một vài xử lý thiết lập cho một sự kiện nhấn vào nút "Send", mà chọn lên các thông báo đầu vào và gửi nó xuống WebSocket, và một là lắng nghe cho message mới đến client WebSockets khác (tức là, các tin nhắn được gửi bởi người dùng khác, mà bên phía server muốn hiển thị trên toàn bộ các client).

Cùng phân tích quá trình khi một client gửi một message diễn ra các xử lý:

- Trình duyệt bắt sự kiện nhấn nút "Send" thông qua một trình xử lý JavaScript, chọn lên các giá trị từ các trường đầu vào (tức là, các tin nhắn văn bản), và phát ra một thông điệp WebSocket sử dụng máy khách kết nối với máy chủ WebSocket (khởi tạo trên web khởi tạo trang).
- Phản phía máy chủ của các kết nối WebSocket nhận message và chuyển tiếp nó tới tất cả các client kết nối khác sử dụng các phương thức broadcast.
- Tất cả các client nhận được message mới như tin nhắn push thông qua một thành phần client-side WebSockets chạy trong trang web. Sau đó, họ lấy nội dung tin nhắn và cập nhật các trang web tại chỗ bằng cách thêm các bài viết mới cho broadcast.



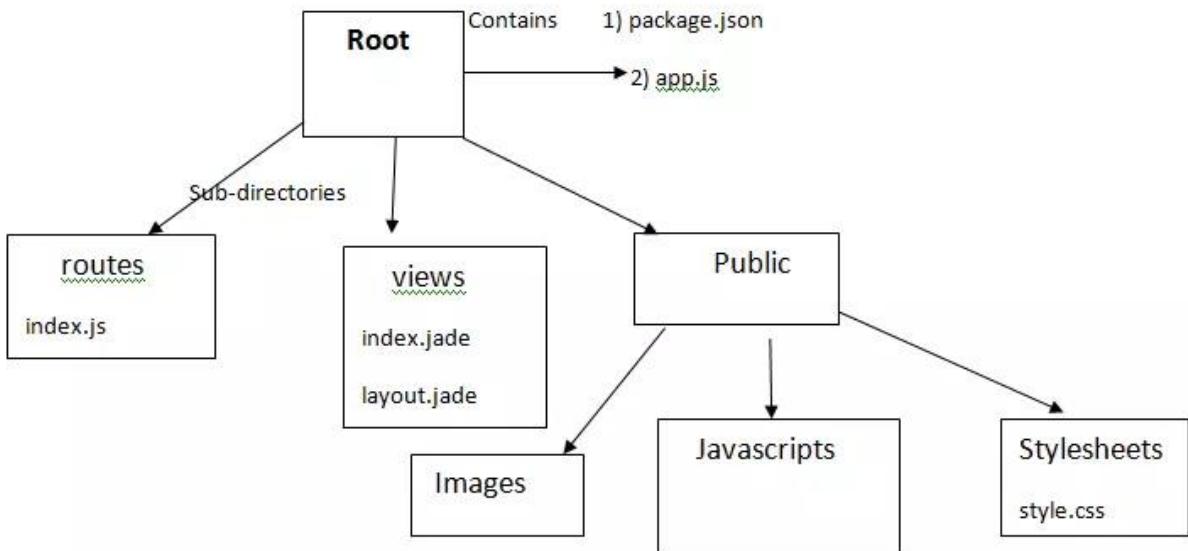
Ảnh 1.2. Ví dụ về NodeJS

Đây là ví dụ đơn giản nhất. Đối với một giải pháp mạnh mẽ hơn, có thể sử dụng một bộ nhớ cache đơn giản dựa trên các lưu trữ Redis. Hoặc trong một giải pháp tiên tiến hơn, một hàng đợi các message để xử lý việc định tuyến các message đến các client và một cơ chế phân phối mạnh mẽ hơn nên có thể trang trải cho các khoản lỗ kết nối tạm thời hoặc tin nhắn lưu trữ cho khách hàng đã đăng ký khi họ offline. Nhưng bất kể những cái thực hiện, Node.js sẽ vẫn được hoạt động theo các nguyên tắc cơ bản: phản ứng với các sự kiện, xử lý nhiều kết nối đồng thời, và duy trì tính lưu động trong trải nghiệm người dùng.

1.3.2. Khái niệm về ExpressJS

- Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile
- Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ
- Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.
- Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN,... [5]

a. Cấu trúc của ExpressJS



Ảnh 1.3. Cấu trúc của Express

Cấu trúc của express js

- Root
- app.js chứa các thông tin về cấu hình, khai báo, các định nghĩa,... để ứng dụng của chúng ta chạy ok.
- package.json chứa các package ứng dụng
- Folder routes: chứa các route có trong ứng dụng
- Folder view: chứa view/template cho ứng dụng
- Folder public chứa các file css, js, images,...cho ứng dụng

b. Router trong Express

Router là một Object, nó là một instance riêng của middleware và routes. Chính vì nó là một instance của middleware và route nên nó có các chức năng của cả hai. Có thể gọi nó là một mini-application

Các Application dùng ExpressJS làm core đều có phần Router được tích hợp sẵn trong đó.

Router hoạt động như một middleware nên có thể dùng nó như một arguments. Hoặc dùng nó như một arguments cho route khác. Ví dụ:

```
// invoked for any requests passed to this router
router.use(function(req, res, next) {
    // .. some logic here .. like any other middleware
    next();
});

// will handle any request that ends in /events
```

```
// depends on where the router is "use()'d"
router.get('/events', function(req, res, next) {
  // ..
});
```

Chúng ta cũng có thể sử dụng Router để chia route. Chẳng hạn:

```
app.use('/calendar', router);
```

c. Các method all của router

- router.all(). Method này phù hợp với việc định nghĩa mang tính chất toàn cục cho các prefix
- Ví dụ:

```
router.all('*', requireAuthentication, loadUser);
```

Nếu đặt route này trên cùng (top) nó yêu cầu tất cả các route bên dưới phải được requireAuthentication. Có nghĩa là xác thực trước khi thực hiện một hành động hay một task nào đó tiếp theo

d. Router.METHOD()

- Router.METHOD() cung cấp cho chức năng Routing trong ExpressJS. Cụ thể METHOD() ở đây là các HTTP method mà chúng ta thường xuyên sử dụng. Chẳng hạn GET, POST, PUT,...
- Lưu ý tên method phải được viết thường
- Ví dụ:

```
router.get('/user/profile', function(req, res, next) {
  res.send('user profile');
});

router.post('/update/user/:id', function (res, req, next) {
  res.send('Update user');
});

router.put('/update/posts/:id', function (req, res, next) {
  res.send('Update post');
});
```

Nếu muốn bảo mật hơn, có thể sử dụng Regex để bắt các Endpoint. Ví dụ:

```
router.get(/^\/commits\/(\w+)(?:\.\.(\w+))?$/, function(req, res){
  var from = req.params[0];
```

```

var to = req.params[1] || 'HEAD';

res.send('commit range ' + from + '..' + to);

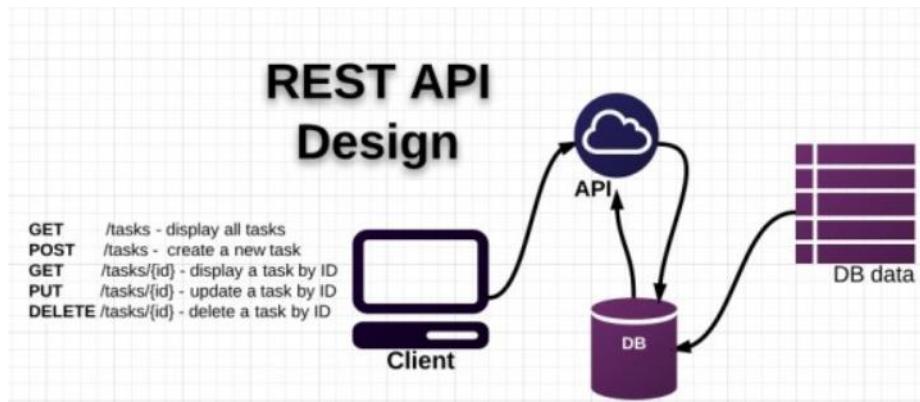
});

```

1.4. Tổng quan về RESTful API.

1.4.1. RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Ảnh 1.4. Thiết kế của một REST API

1.4.2. Các thành phần

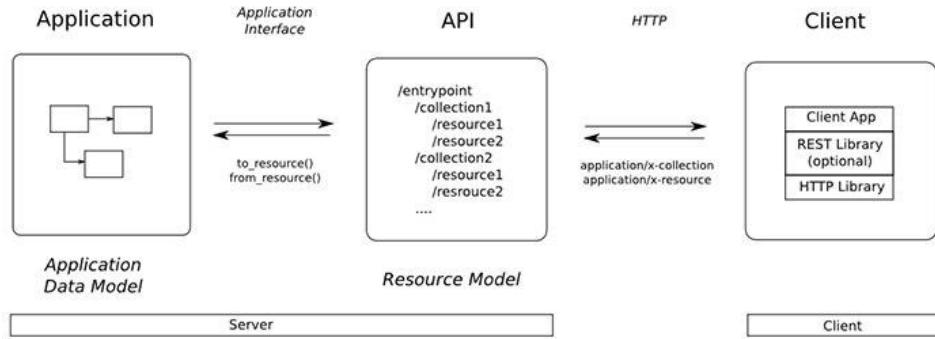
API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu cần cho ứng dụng ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản lý các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

1.4.3. Cách thức hoạt động



Ảnh 1.5. Quy trình và cách thức ứng dụng tương tác với khách hàng thông qua API

- REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.
- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

1.4.4. Authentication và dữ liệu trả về

RESTful API không sử dụng session và cookie, nó sử dụng một access_token với mỗi request. Dữ liệu trả về thường có cấu trúc như sau:

```
{  
  "data": {  
    "id": "1",  
    "name": "Benhvien"  
  }  
}
```

1.5. Tổng quan về MySQL.

1.5.1. Khái niệm về MySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (viết tắt RDBMS) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Hoạt động theo mô hình Client – Server. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu và mỗi cơ sở dữ liệu có nhiều bảng quan hệ chứa dữ liệu. [4]

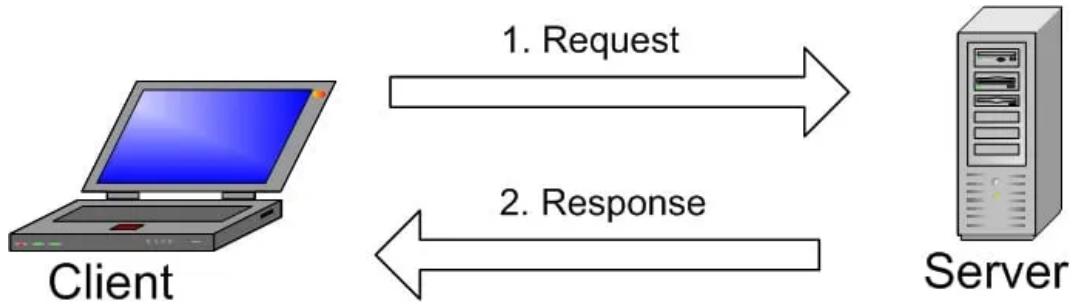
1.5.2. Các thành phần chính của SQL

- Định nghĩa dữ liệu bao gồm các câu lệnh giúp xác định cơ sở dữ liệu và các đối tượng của nó, ví dụ: table, view, trigger, stored procedures, v.v.
- Thao tác dữ liệu chứa các câu lệnh cho phép cập nhật và truy vấn dữ liệu.

- Điều khiển dữ liệu cho phép cấp quyền cho người dùng truy cập dữ liệu cụ thể trong cơ sở dữ liệu.

1.5.3. Mô hình hoạt động của MySQL

MySQL hoạt động theo mô hình Client – Server



Ảnh 1.6. Mô hình được sử dụng trong MySQL

Một hoặc nhiều thiết bị (máy khách) kết nối với máy chủ thông qua một mạng cụ thể. Mọi máy khách có thể đưa ra yêu cầu từ giao diện người dùng trên màn hình của họ và máy chủ sẽ tạo ra kết quả mong muốn, miễn là cả hai đều hiểu hướng dẫn. Các quy trình chính diễn ra trong môi trường MySQL đều giống nhau, đó là:

- MySQL tạo cơ sở dữ liệu để lưu trữ và thao tác dữ liệu, xác định mối quan hệ của từng bảng.
- Client có thể đưa ra yêu cầu bằng cách nhập các câu lệnh SQL cụ thể trên MySQL.
- Server application sẽ phản hồi với thông tin được yêu cầu và nó sẽ xuất hiện ở phía máy khách.

Ở phía Client, sẽ để ý đến MySQL GUI (Giao diện đồ họa người dùng) sử dụng. GUI càng nhẹ và thân thiện với người dùng, các hoạt động quản lý dữ liệu của họ sẽ nhanh hơn và dễ dàng hơn.

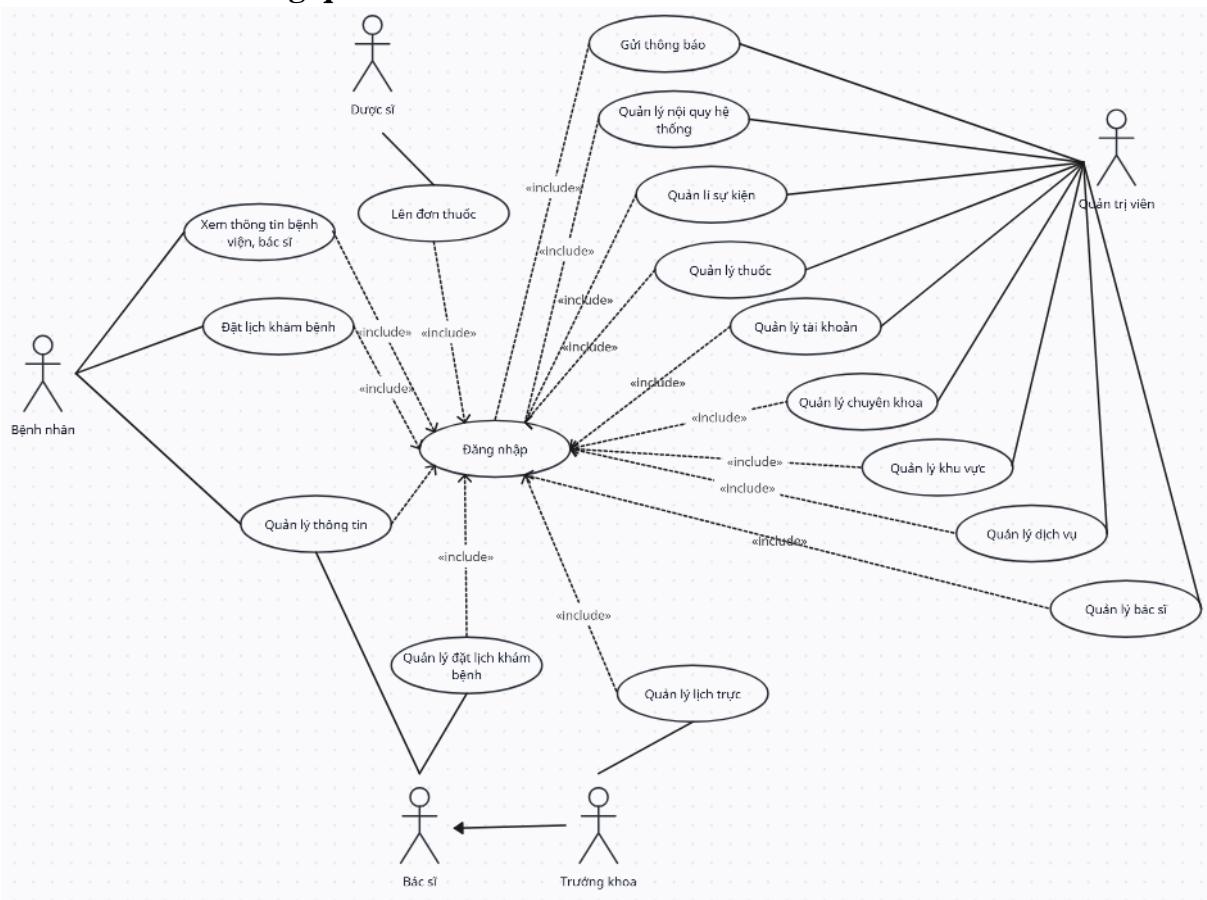
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Giới thiệu bài toán

Trong bối cảnh hiện đại, công nghệ thông tin đang ngày càng phát triển và ứng dụng rộng rãi vào nhiều lĩnh vực khác nhau, bao gồm cả y tế. Quản lý phòng khám là một nhiệm vụ phức tạp đòi hỏi sự phối hợp chặt chẽ giữa nhiều bộ phận và các hệ thống khác nhau. Một hệ thống quản lý phòng khám hiệu quả không chỉ giúp nâng cao chất lượng dịch vụ y tế mà còn giảm thiểu sai sót và tối ưu hóa quy trình làm việc. Bài toán đặt ra là xây dựng một hệ thống quản lý trực tuyến toàn diện, giúp cải thiện quy trình quản lý thông tin bệnh nhân, hồ sơ y tế, lịch hẹn, quản lý nhân viên y tế, và các hoạt động hành chính khác của phòng khám.

2.2. Sơ đồ Use Case

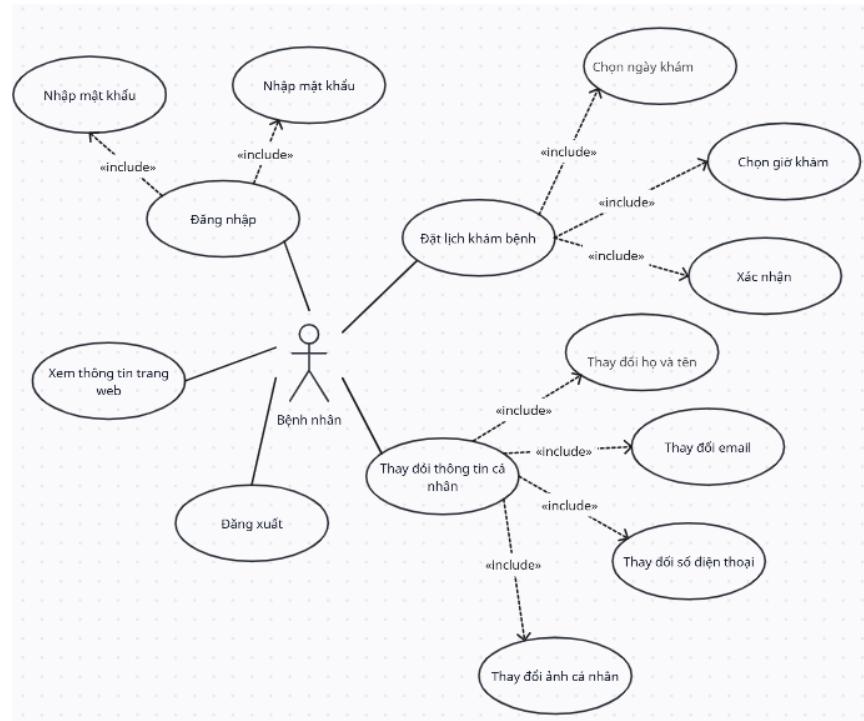
2.2.1. Use Case tổng quát



Ảnh 2.1. Use case tổng quát

Sơ đồ Use Case tổng quát này minh họa sự tương tác của bệnh nhân, bác sĩ, quản trị viên

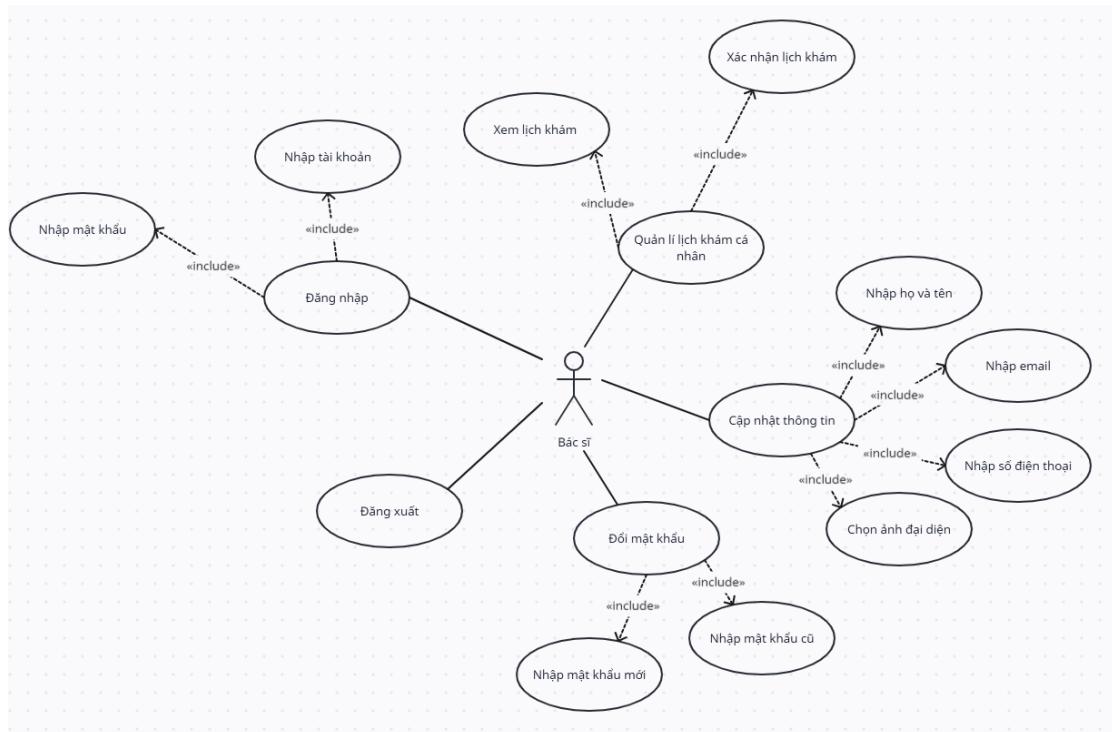
2.2.2. Use Case bệnh nhân



Ảnh 2.2. Use Case bệnh nhân

Bệnh nhân sau khi đăng nhập có thể xem thông tin của trang web, thay đổi thông tin cá nhân, đặt lịch khám bệnh.

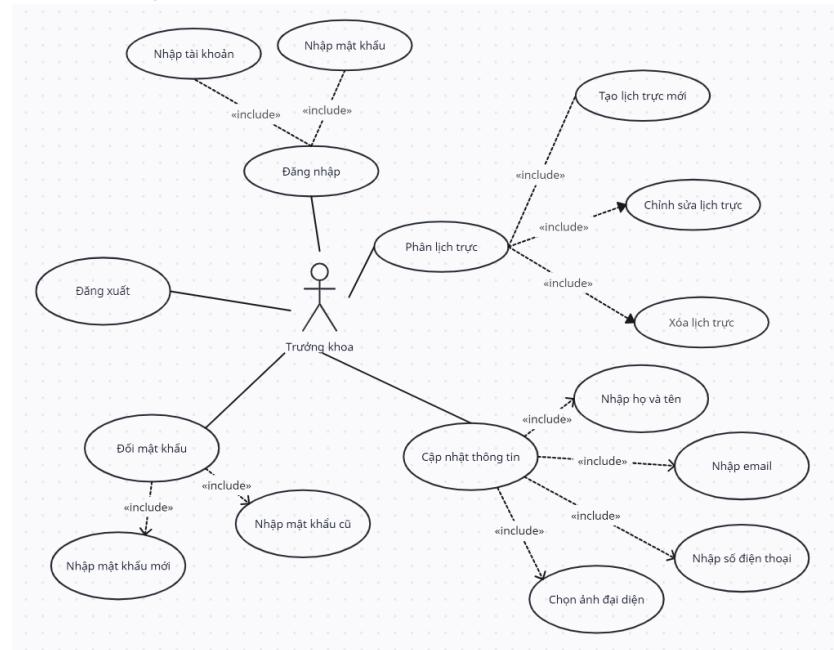
2.2.3. Use Case bác sĩ



Ảnh 2.3. Use Case bác sĩ

Bác sĩ sau khi đăng nhập vào hệ thống sẽ quản lý được lịch khám bệnh của mình, thay đổi các thông tin cá nhân, đổi mật khẩu.

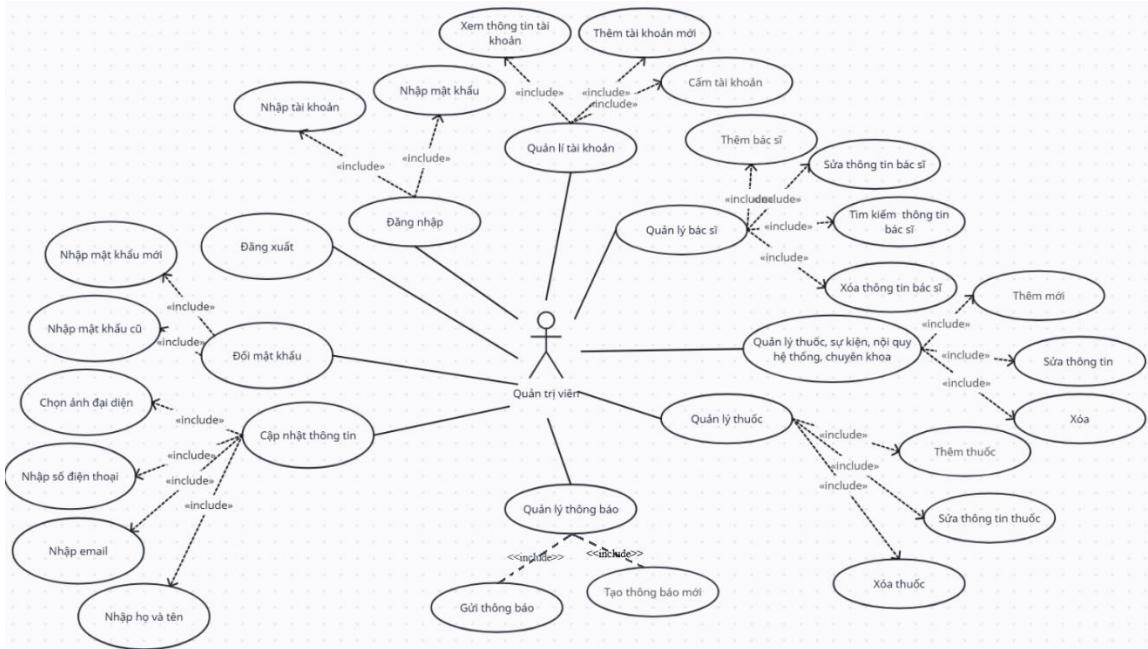
2.2.4. Use Case trưởng khoa



Ảnh 2.4. Use Case trưởng khoa

Trưởng khoa sau khi đăng nhập vào hệ thống sẽ quản lý, phân lịch trực cho bác sĩ, cập nhật thông tin cá nhân, đổi mật khẩu.

2.2.5. Use Case quản trị viên

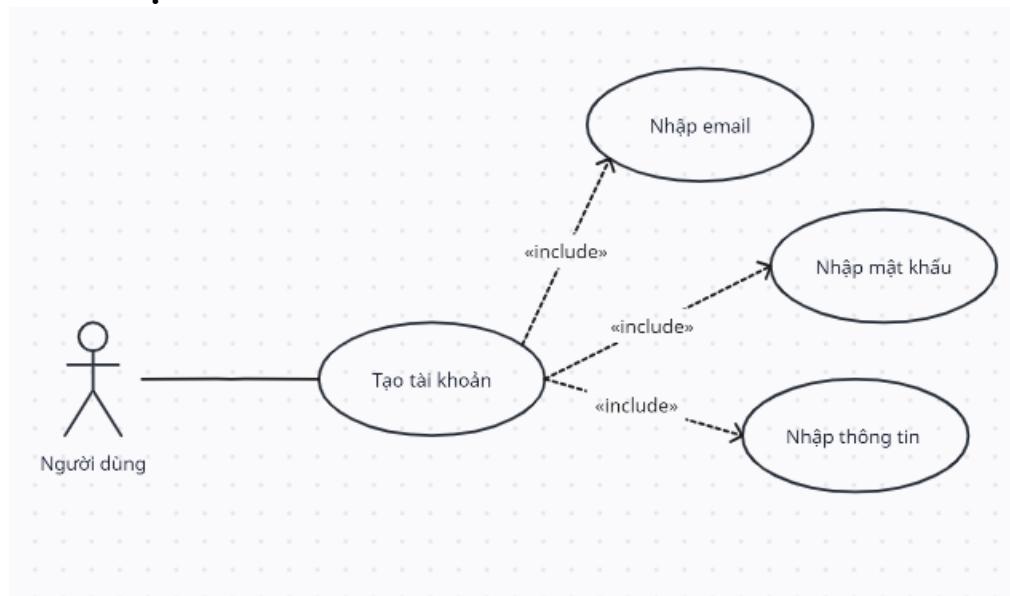


Ảnh 2.5. Use Case quản trị viên

Quản trị viên sau khi đăng nhập có thể thay đổi các thông tin cá nhân, cập nhật mật khẩu, quản lý các thành phần của hệ thống như: quản lý tài khoản, quản lý bác sĩ, quản lý thuốc, quản lý thông báo

2.3. Kịch bản Use Case chi tiết:

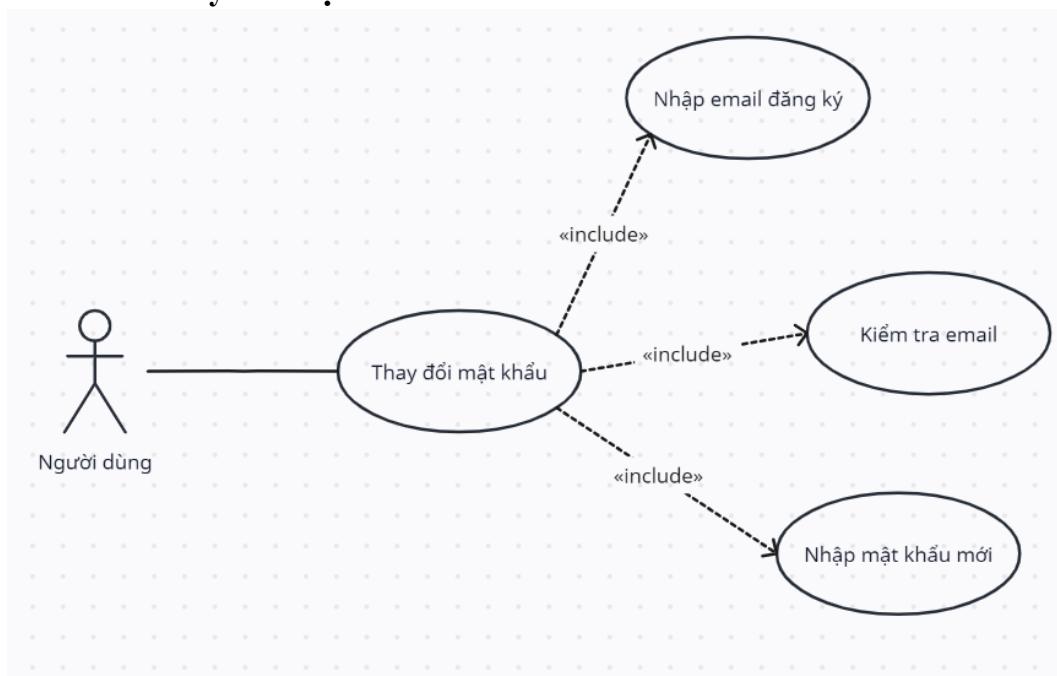
2.3.1. Use case tạo tài khoản



Ảnh 2.6. Use case tạo tài khoản

Người dùng khi chưa có tài khoản có thể tạo tài khoản. Khi tạo tài khoản cần nhập đầy đủ các thông tin như email, mật khẩu, tên hiển thị, số điện thoại

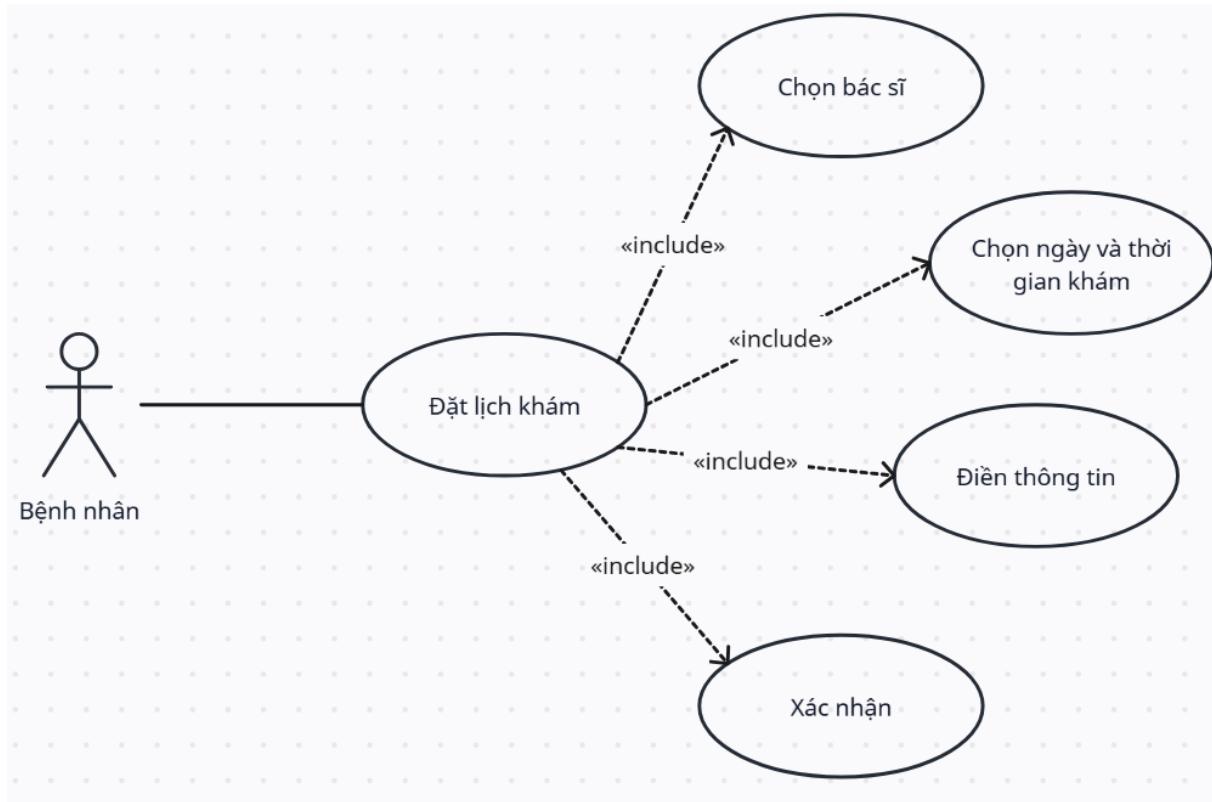
2.3.2. Use case thay đổi mật khẩu



Ảnh 2.7. Use case thay đổi mật khẩu

Người dùng khi không nhớ mật khẩu của mình có thể sử dụng chức năng thay đổi mật khẩu. Thay đổi bằng cách nhập email của tài khoản, kiểm tra email để lấy đường dẫn xác thực, sau đó khởi tạo mật khẩu mới

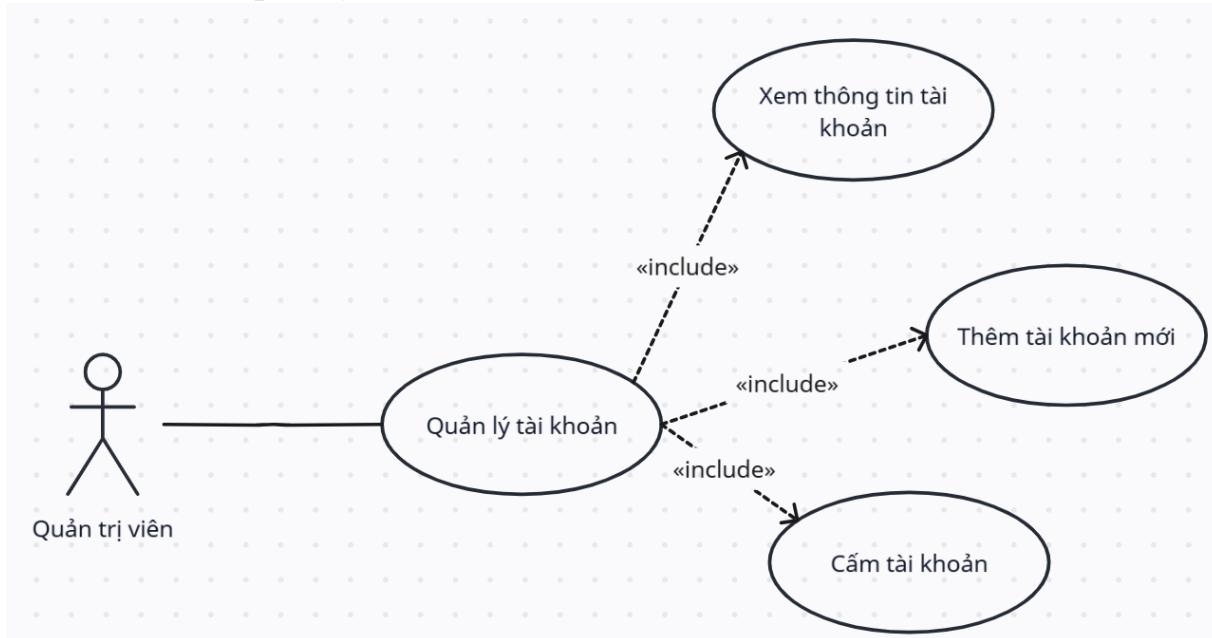
2.3.3. Use case đặt lịch khám bệnh



Ảnh 2.8. Use Case đặt lịch khám bệnh

Bệnh nhân sau khi đăng nhập có thể đặt lịch khám bệnh. Bệnh nhân sẽ chọn bác sĩ muôn đặt lịch khám bệnh, chọn ngày và giờ muôn đặt lịch khám, điền thông tin, xác nhận

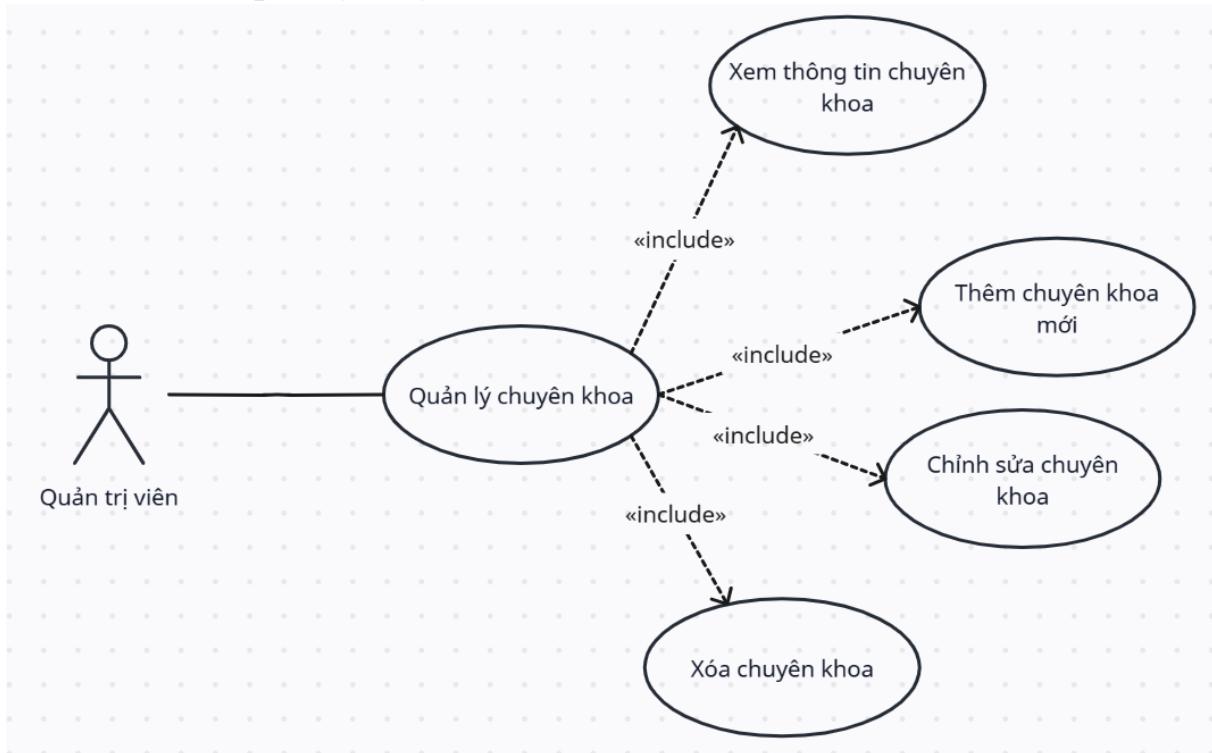
2.3.4. Use Case quản lý tài khoản



Ảnh 2.9. Use Case quản lý tài khoản

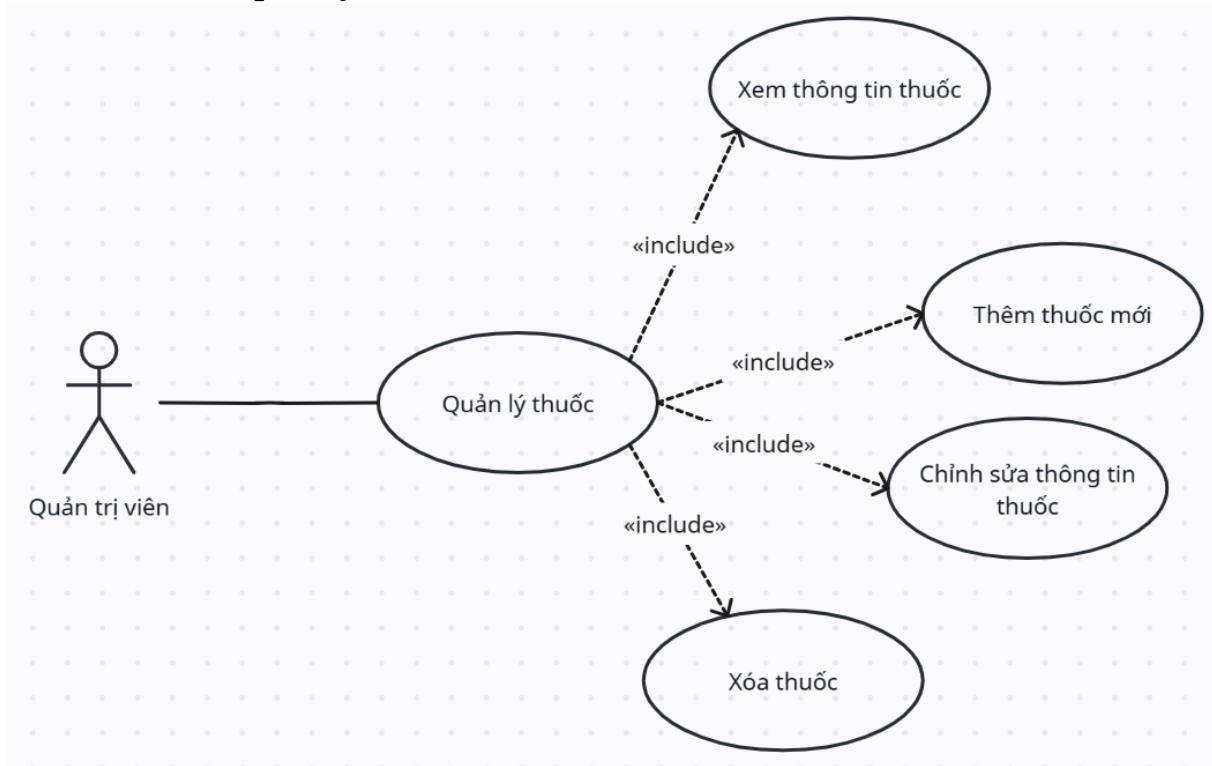
Quản trị viên sau khi đăng nhập có thể quản lý tài khoản như xem thông tin các tài khoản có trong hệ thống, thêm tài khoản mới, cấm tài khoản khỏi hệ thống

2.3.5. Use Case quản lý chuyên khoa



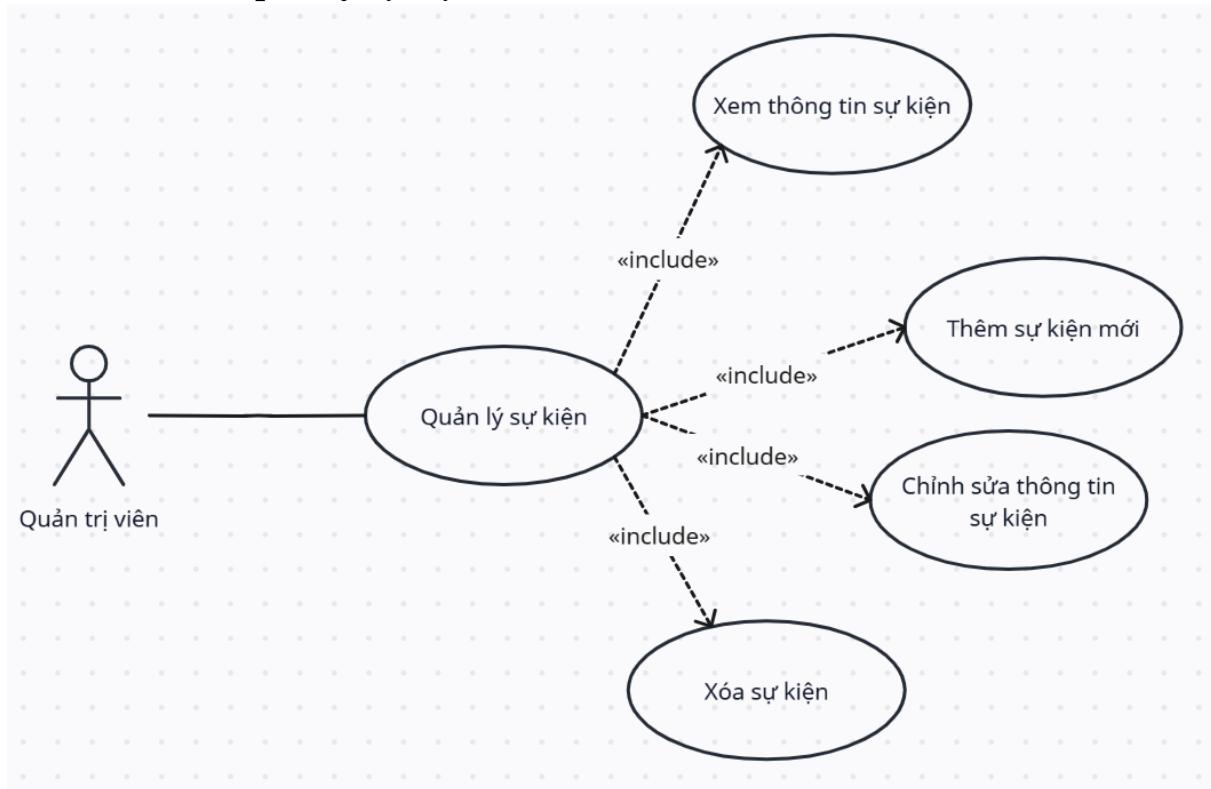
Ảnh 2.10. Use Case quản lý chuyên khoa

2.3.6. Use Case quản lý thuốc



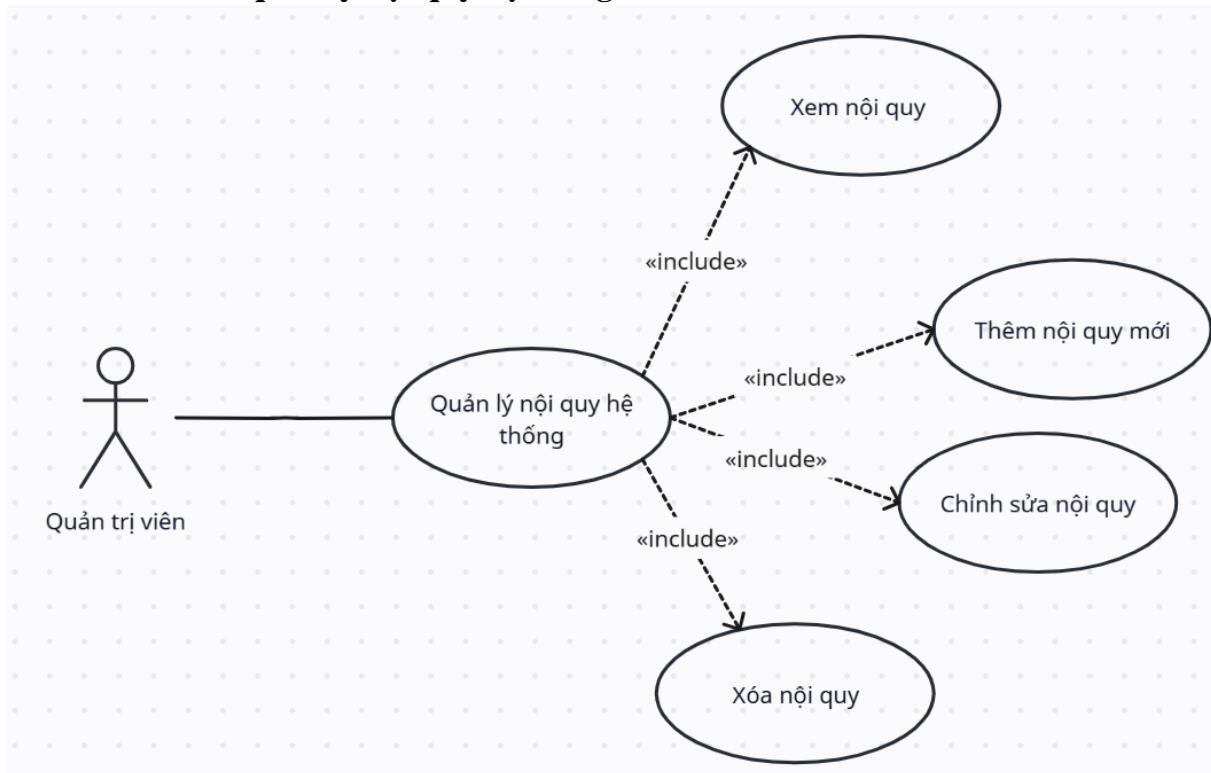
Ảnh 2.11. Use Case quản lý thuốc

2.3.7. Use Case quản lý sự kiện



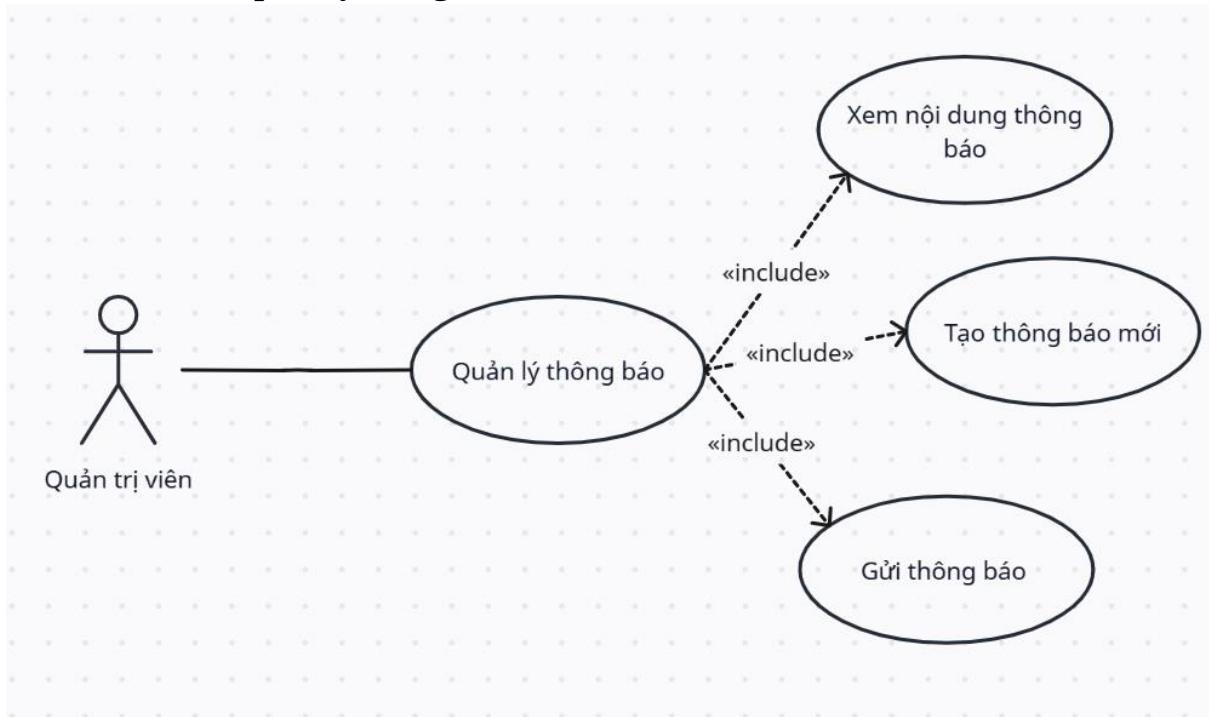
Ảnh 2.12. Use Case quản lý sự kiện

2.3.8. Use Case quản lý nội quy hệ thống



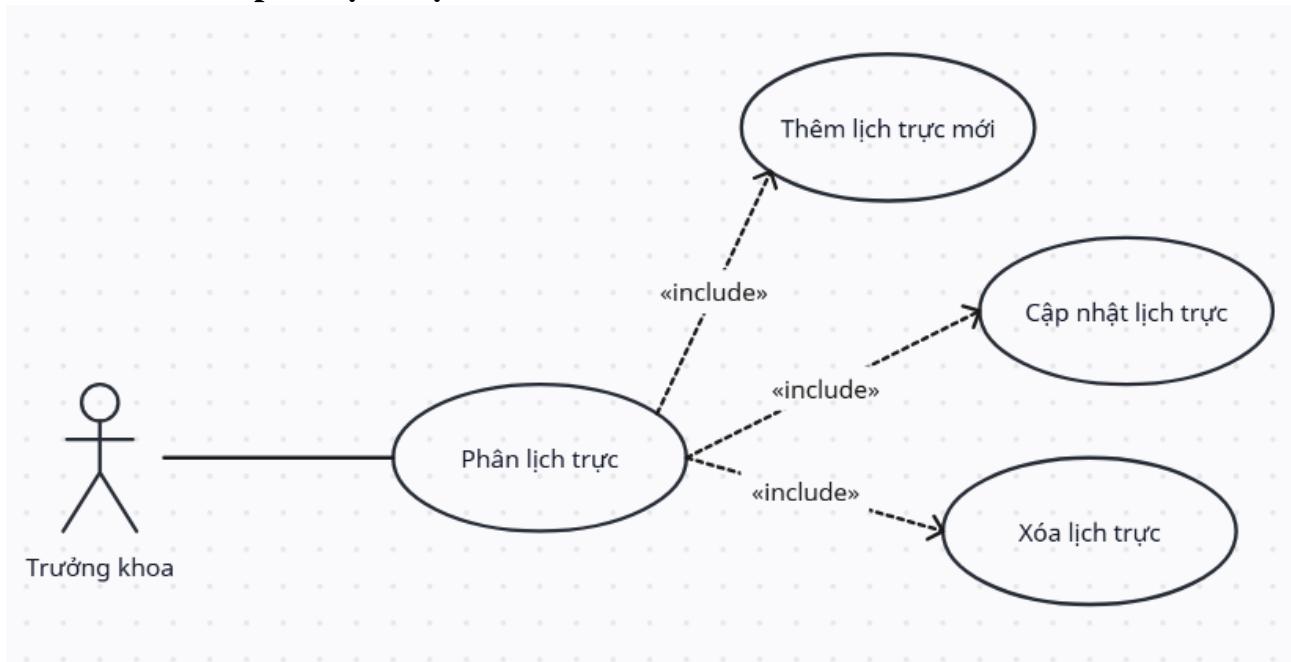
Ảnh 2.13. Use Case quản lý nội quy hệ thống

2.3.9. Use Case quản lý thông báo



Ảnh 2.14. Use Case quản lý thông báo

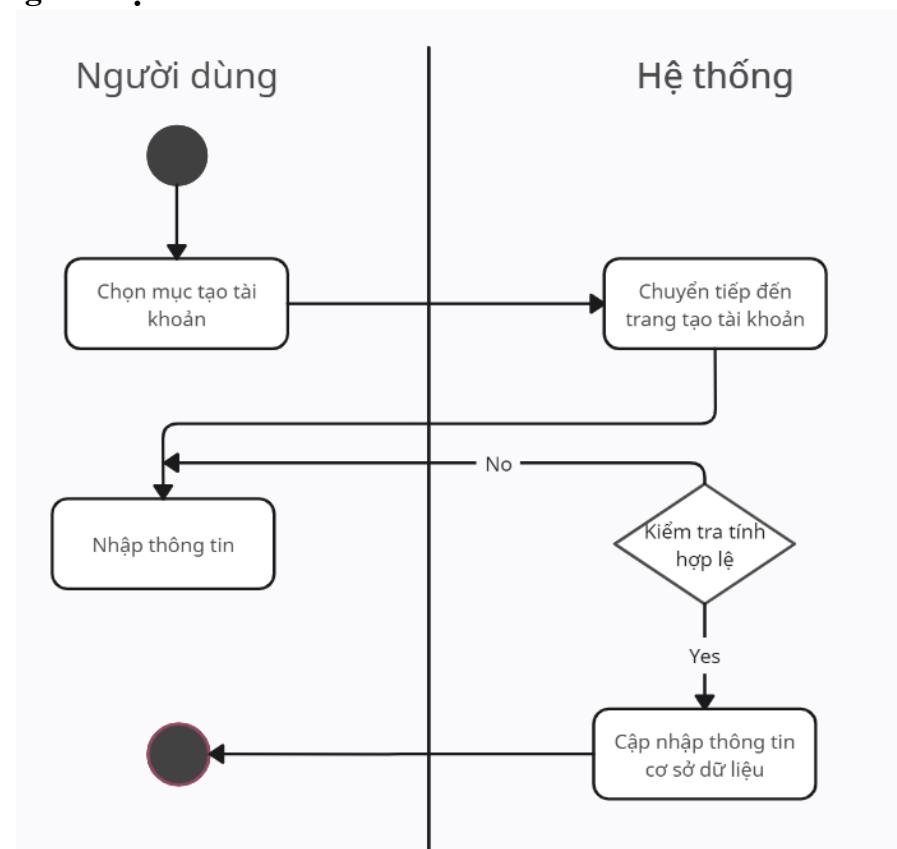
2.3.10. Use Case phân lịch trực



Ảnh 2.15. Use Case phân lịch trực

2.4. Trạng thái chi tiết

2.4.1. Trạng thái tạo tài khoản

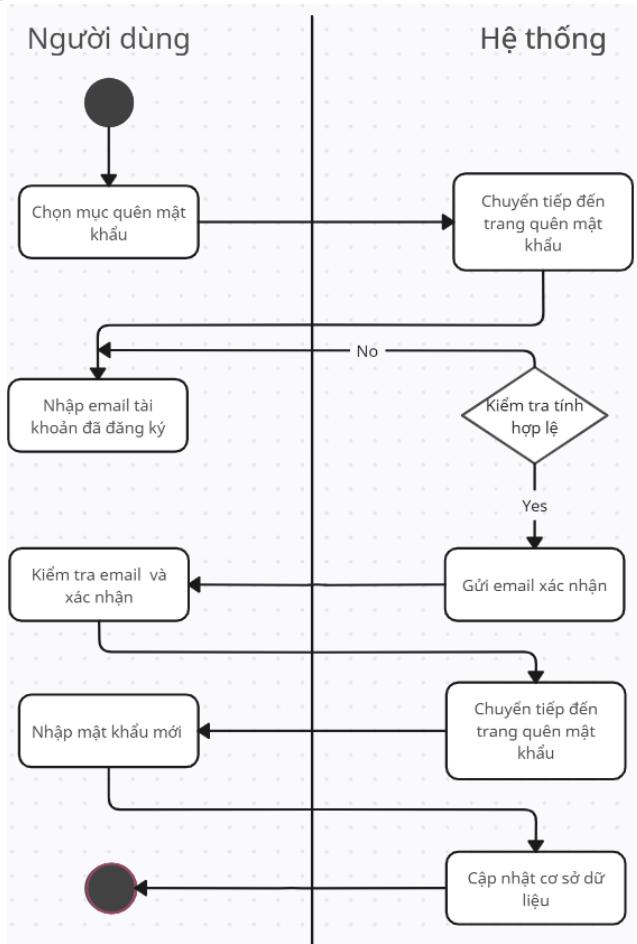


Ảnh 2.16. Biểu đồ trạng thái tạo tài khoản

Người dùng	Hệ thống
Truy cập vào trang web, chọn tạo tài khoản	Chuyển tiếp đến trang tạo tài khoản mới
Điền thông tin (email, mật khẩu, họ tên, ...)	Kiểm tra tính hợp lệ của thông tin mà người dùng đã nhập
Nhấn nút tạo tài khoản	Thêm tài khoản vào cơ sở dữ liệu

Bảng 2.1. Trạng thái tạo tài khoản

2.4.2. Trạng thái quên mật khẩu

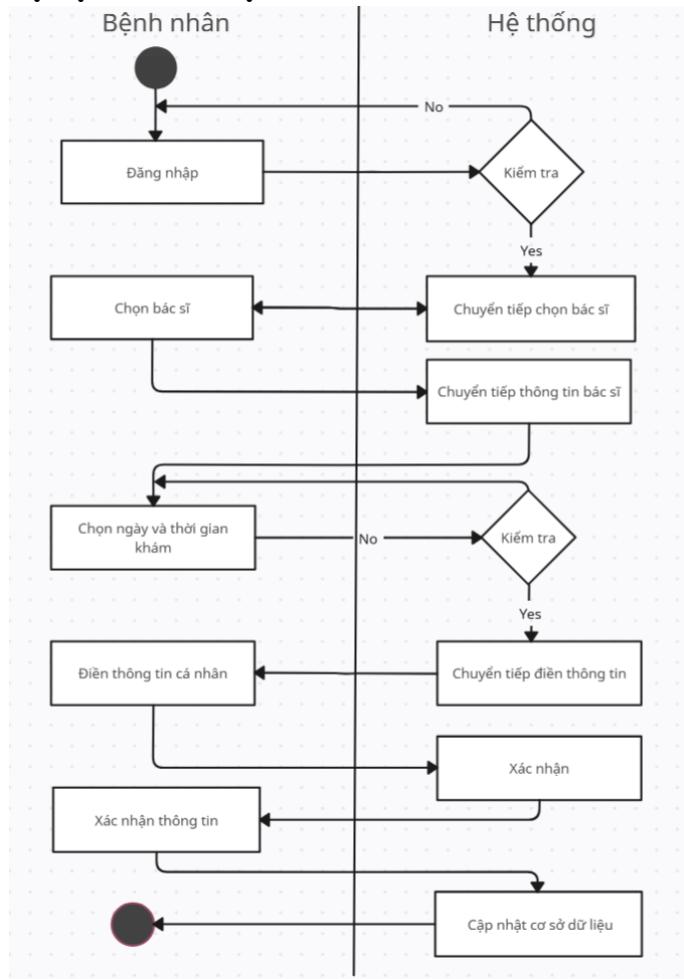


Ảnh 2.17. Biểu đồ trạng thái quên mật khẩu

Người dùng	Hệ thống
Truy cập vào trang web và chọn quên mật khẩu	Chuyển tiếp đến trang quên mật khẩu
Nhập email của tài khoản quên mật khẩu	Kiểm tra email, nếu tồn tại email trong hệ thống sẽ gửi email xác nhận, nếu không tồn tại sẽ quay trở lại nhập email
Kiểm tra email và xác nhận	Chuyển hướng đến trang nhập mật khẩu mới
Nhập mật khẩu mới	Cập nhật thông tin cơ sở dữ liệu

Bảng 2.2. Trạng thái quên mật khẩu

2.4.3. Trạng thái đặt lịch khám bệnh

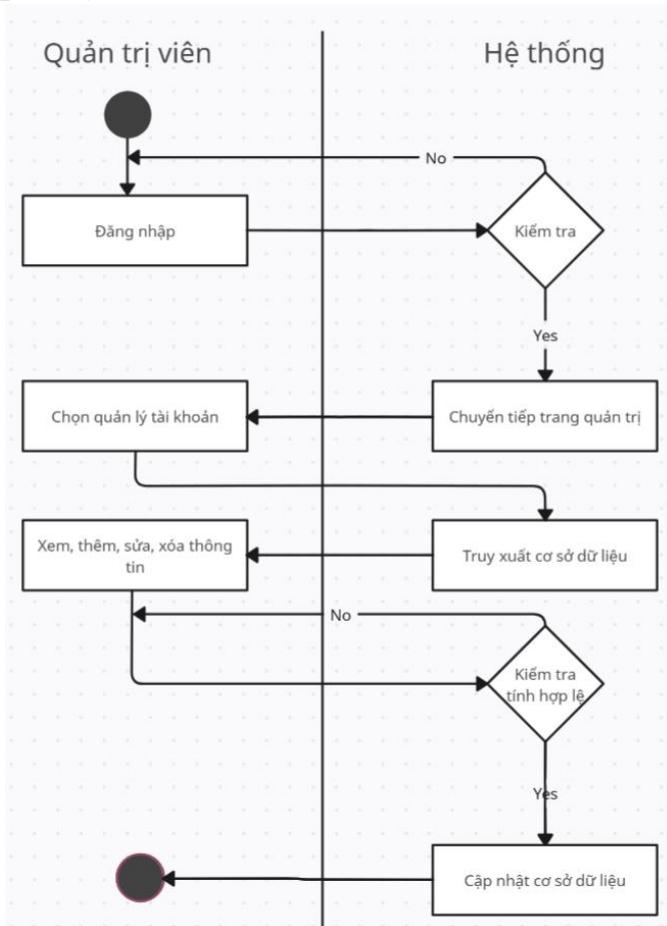


Ảnh 2.18. Biểu đồ trạng thái đặt lịch khám bệnh

Bệnh nhân	Hệ thống
Bệnh nhân đăng nhập vào trang web bằng tài khoản của bệnh nhân	
	Kiểm tra thông tin đăng nhập. Nếu đúng chuyển tiếp đến trang chủ, nếu sai quay lại trang đăng nhập
Chọn bác sĩ muốn đặt lịch khám	
	Chuyển tiếp đến trang thông tin của bác sĩ
Chọn ngày đặt lịch và khung thời gian khám	
	Chuyển tiếp đến trang điền thông tin
Điền thông tin của bệnh nhân	
	Kiểm tra xác nhận
	Cập nhật thông tin cơ sở dữ liệu

Bảng 2.3. Trạng thái đặt lịch khám

2.4.4. Trạng thái quản lý tài khoản

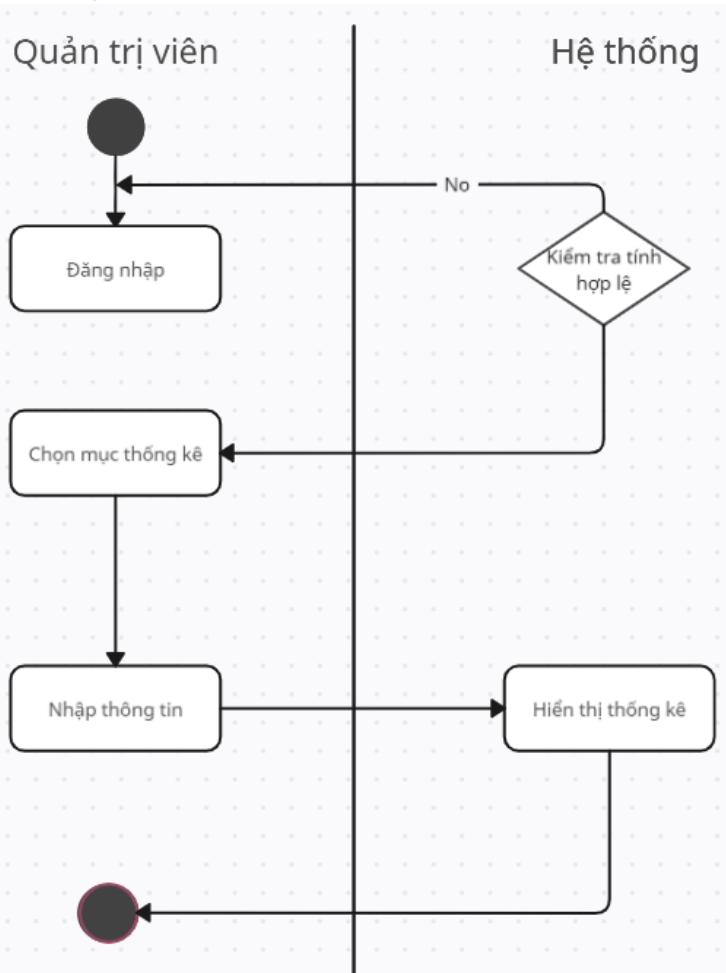


Ảnh 2.19. Biểu đồ trạng thái quản lý tài khoản

Quản trị viên	Hệ thống
Quản trị viên đăng nhập vào trang web bằng tài khoản quản trị viên	Kiểm tra thông tin đăng nhập. Nếu đúng chuyển tiếp đến trang quản trị, nếu sai quay lại trang đăng nhập
Chọn quản lý tài khoản	Truy xuất cơ sở dữ liệu, chuyển tiếp đến trang quản lý tài khoản
Xem, thêm, sửa, xóa thông tin tài khoản	Kiểm tra tính hợp lệ. Nếu đúng, cập nhật thông tin tài khoản vào cơ sở dữ liệu, nếu sai quay lại

Bảng 2.4. Trạng thái quản lý tài khoản

2.4.4. Trạng thái thống kê

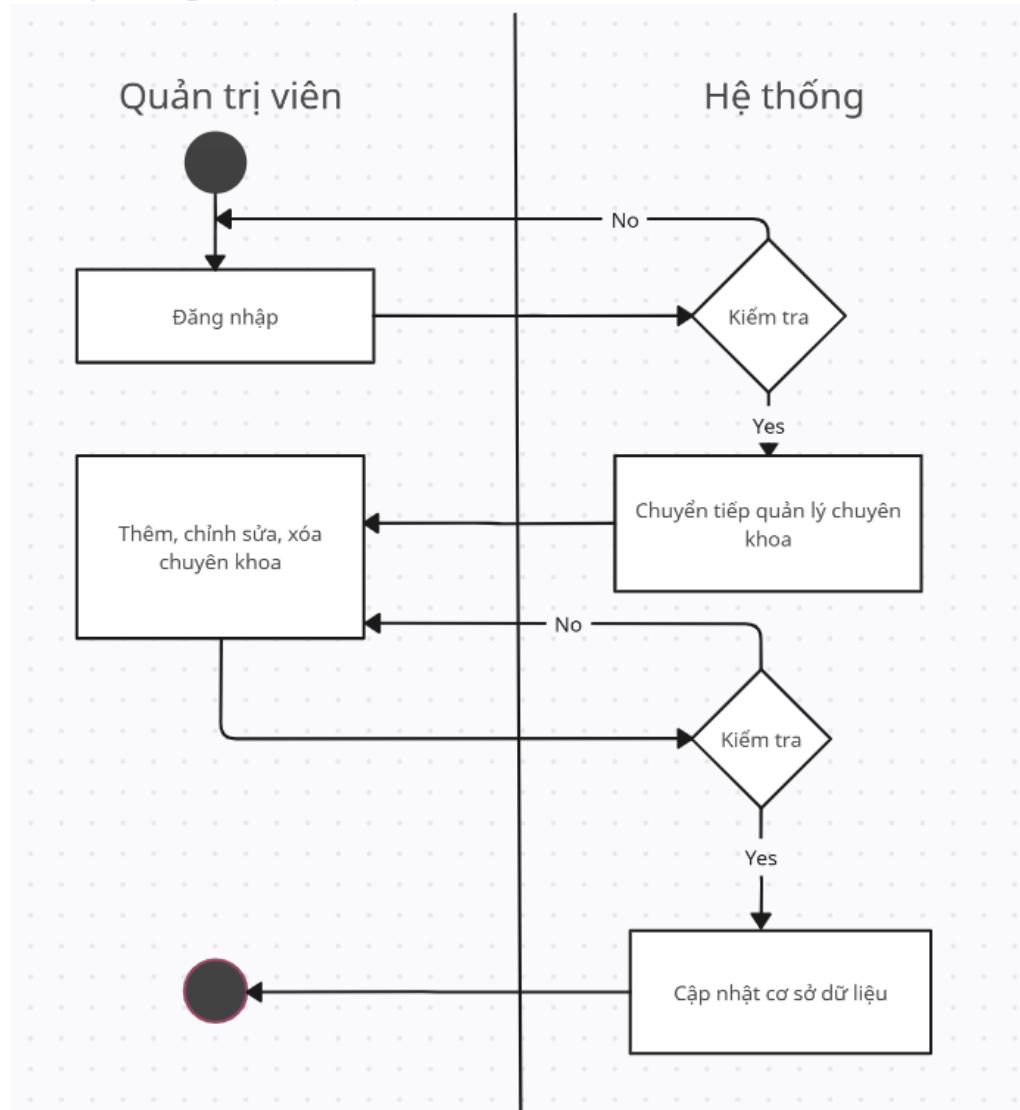


Ảnh 2.20. Biểu đồ trạng thái thống kê

Quản trị viên	Hệ thống
Quản trị viên đăng nhập vào trang web bằng tài khoản quản trị viên	
	Kiểm tra thông tin đăng nhập, nếu đúng chuyển tiếp đến trang quản trị, nếu sai quay trở lại đăng nhập
Chọn thống kê	
Chọn thông tin thống kê	Chuyển tiếp đến trang thống kê
	Hiển thị thông tin thống kê

Bảng 2.5. Trạng thái thống kê

2.4.5. Trạng thái quản lý chuyên khoa

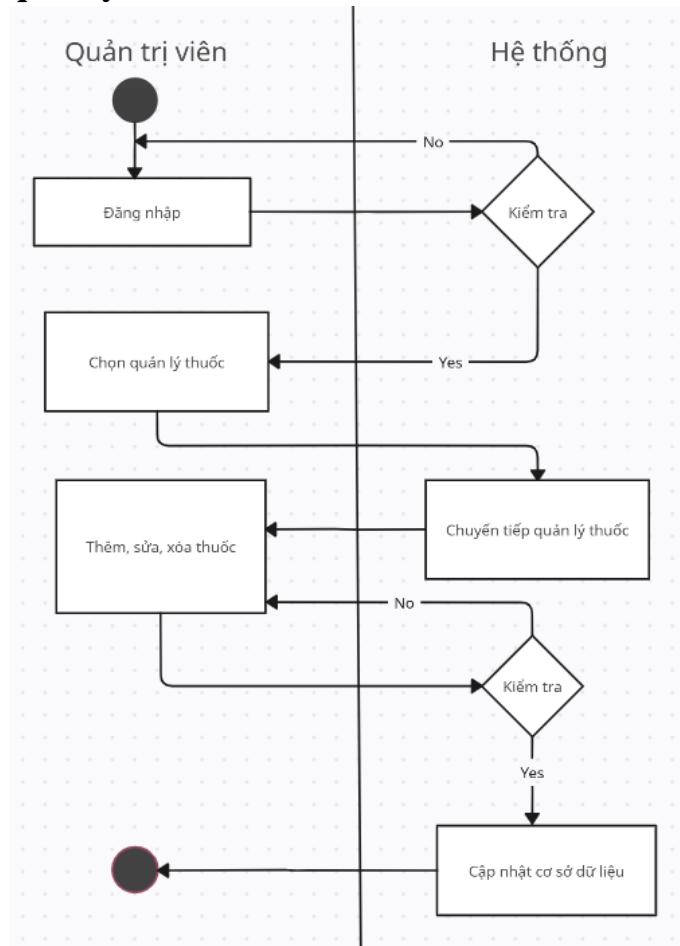


Ảnh 2.21. Biểu đồ trạng thái quản lý chuyên khoa

Quản trị viên	Hệ thống
Quản trị viên đăng nhập vào trang web bằng tài khoản quản trị viên	
	Kiểm tra thông tin đăng nhập, nếu đúng chuyển tiếp đến trang quản trị, nếu sai quay trở lại đăng nhập
Chọn quản lý chuyên khoa	Chuyển tiếp đến trang quản lý chuyên khoa
Thực hiện thêm mới, chỉnh sửa, xóa chuyên khoa	
	Kiểm tra xác nhận
	Cập nhật cơ sở dữ liệu

Bảng 2.6. Trạng thái quản lý chuyên khoa

2.4.6. Trạng thái quản lý thuốc

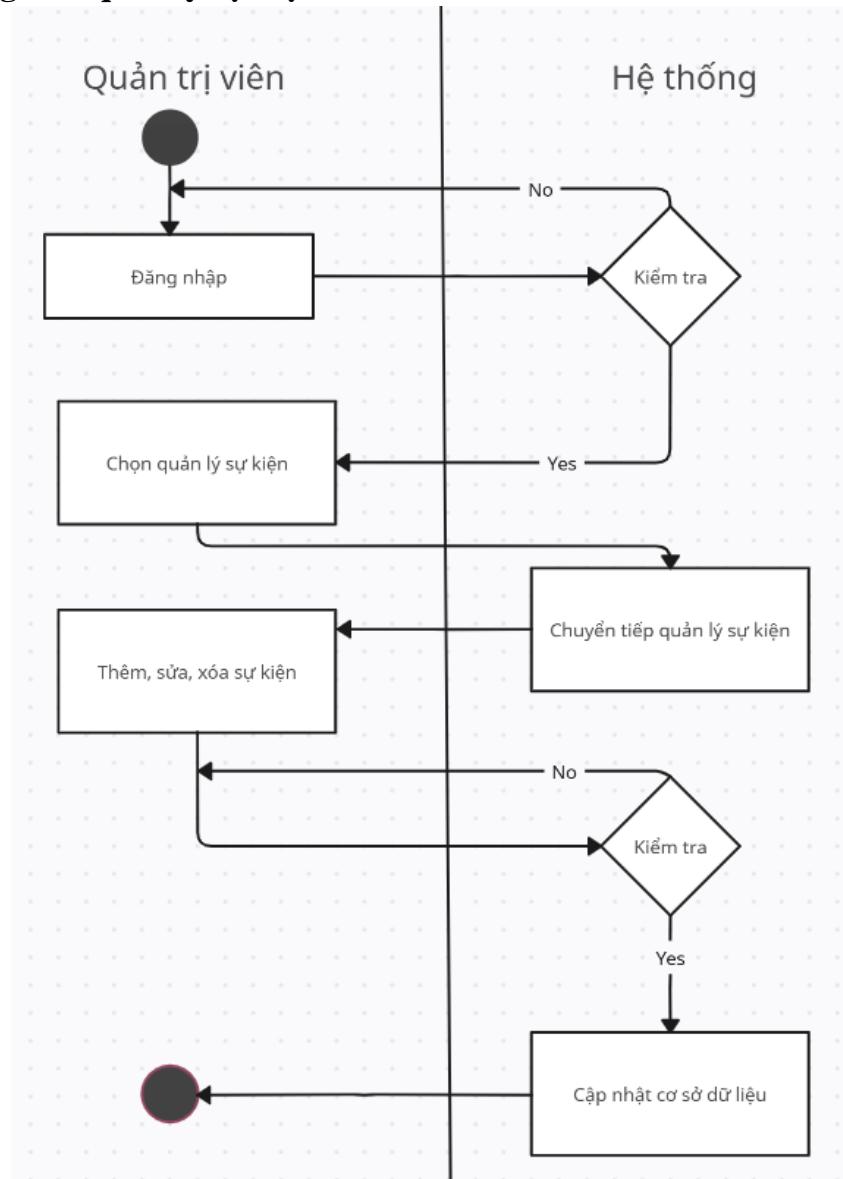


Ảnh 2.22. Biểu đồ trạng thái quản lý thuốc

Quản trị viên	Hệ thống
Quản trị viên đăng nhập vào trang web bằng tài khoản quản trị viên	
	Kiểm tra thông tin đăng nhập, nếu đúng chuyển tiếp đến trang quản trị, nếu sai quay trở lại đăng nhập
Chọn quản lý thuốc	Chuyển tiếp đến trang quản lý thuốc
Thực hiện thêm mới, chỉnh sửa, xóa thuốc	
	Kiểm tra xác nhận
	Cập nhật cơ sở dữ liệu

Bảng 2.7. Trạng thái quản lý thuốc

2.4.7. Trạng thái quản lý sự kiện

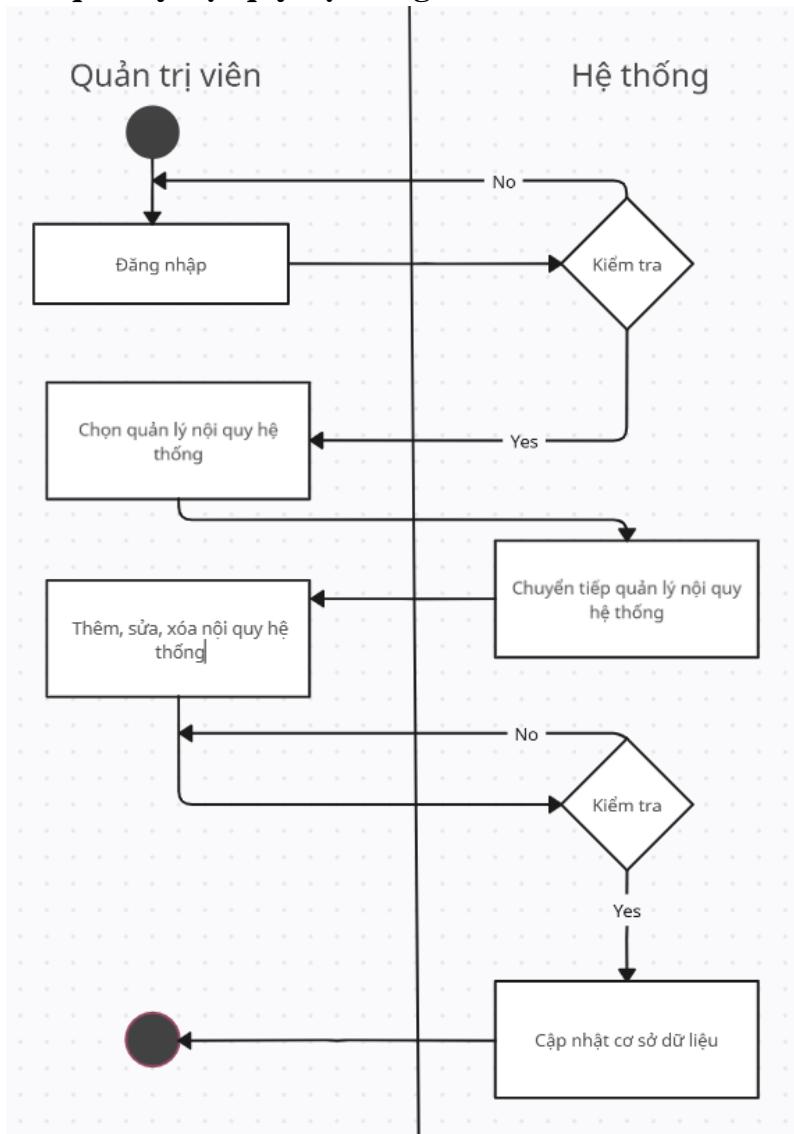


Ảnh 2.23. Biểu đồ trạng thái quản lý sự kiện

Quản trị viên	Hệ thống
Quản trị viên đăng nhập vào trang web bằng tài khoản quản trị viên	Kiểm tra thông tin đăng nhập, nếu đúng chuyển tiếp đến trang quản trị, nếu sai quay trở lại đăng nhập
Chọn quản sự kiện	Chuyển tiếp đến trang quản sự kiện
Thực hiện thêm mới, chỉnh sửa, xóa thuôc	Kiểm tra xác nhận
	Cập nhật cơ sở dữ liệu

Bảng 2.8. Trạng thái quản lý sự kiện

2.4.8. Trạng thái quản lý nội quy hệ thống

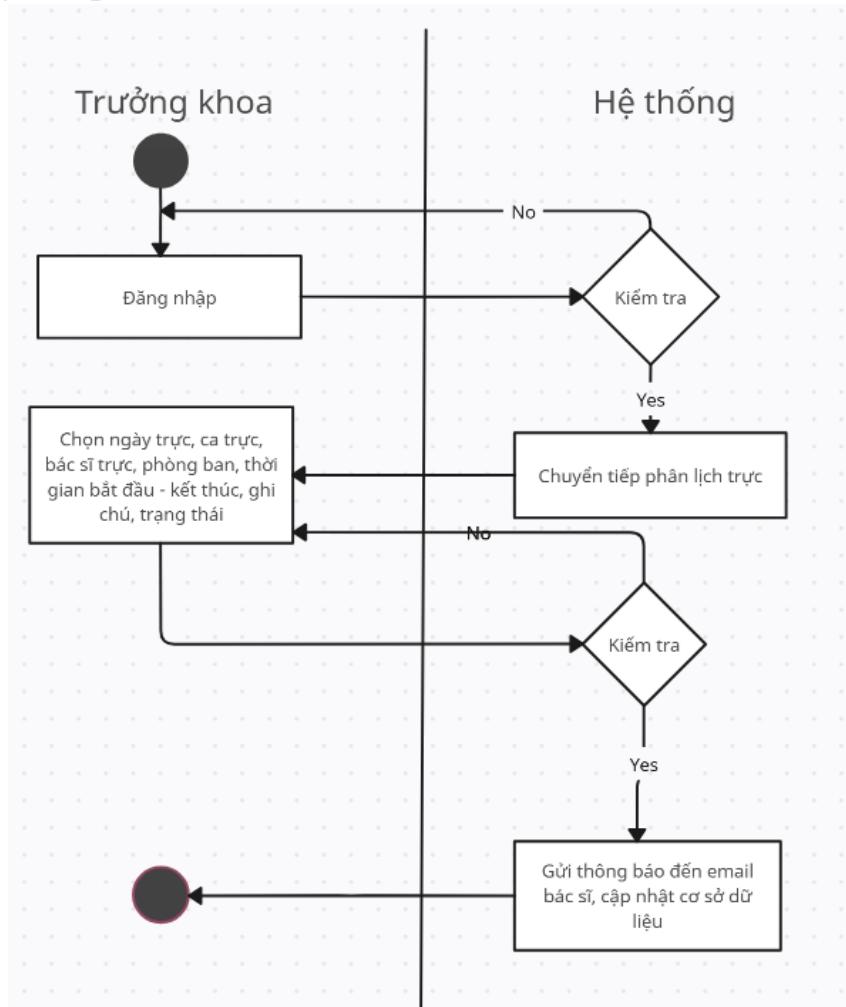


Ảnh 2.24. Biểu đồ trạng thái quản lý nội quy hệ thống

Quản trị viên	Hệ thống
Quản trị viên đăng nhập vào trang web bằng tài khoản quản trị viên	Kiểm tra thông tin đăng nhập, nếu đúng chuyển tiếp đến trang quản trị, nếu sai quay trở lại đăng nhập
Chọn quản lý nội quy hệ thống	Chuyển tiếp đến trang quản lý nội quy hệ thống
Thực hiện thêm mới, chỉnh sửa, xóa nội quy hệ thống	Kiểm tra xác nhận
	Cập nhật cơ sở dữ liệu

Bảng 2.9. Trạng thái quản lý nội quy hệ thống

2.4.9. Trạng thái phân lịch trực



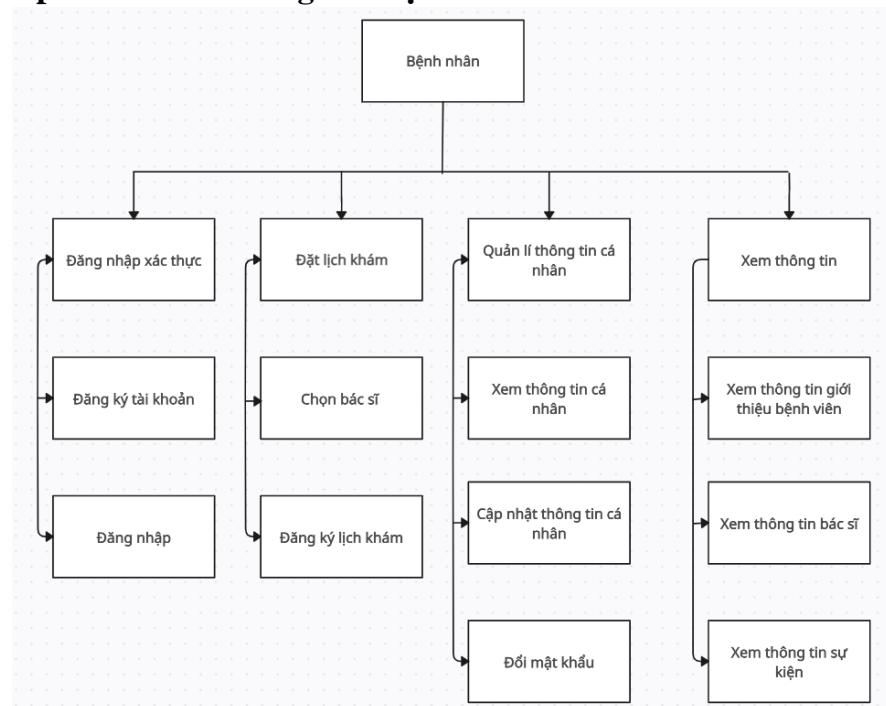
Ảnh 2.25 Biểu đồ trạng thái phân lịch trực

Trường khoa	Hệ thống
Truy cập vào trang web cho bác sĩ, chọn quản lý lịch trực	
	Chuyển tiếp đến trang tạo lịch trực
Chọn phân công ngày trực, ca trực, bác sĩ trực, phòng ban, thời gian bắt đầu – kết thúc, ghi chú, trạng thái	
	Kiểm tra xác nhận
	Gửi email cho bác sĩ được phân trực
	Cập nhật cơ sở dữ liệu

Bảng 2.10. Trạng thái phân lịch trực

2.5. Sơ đồ phân rã chức năng

2.5.1. Sơ đồ phân rã chức năng của bệnh nhân



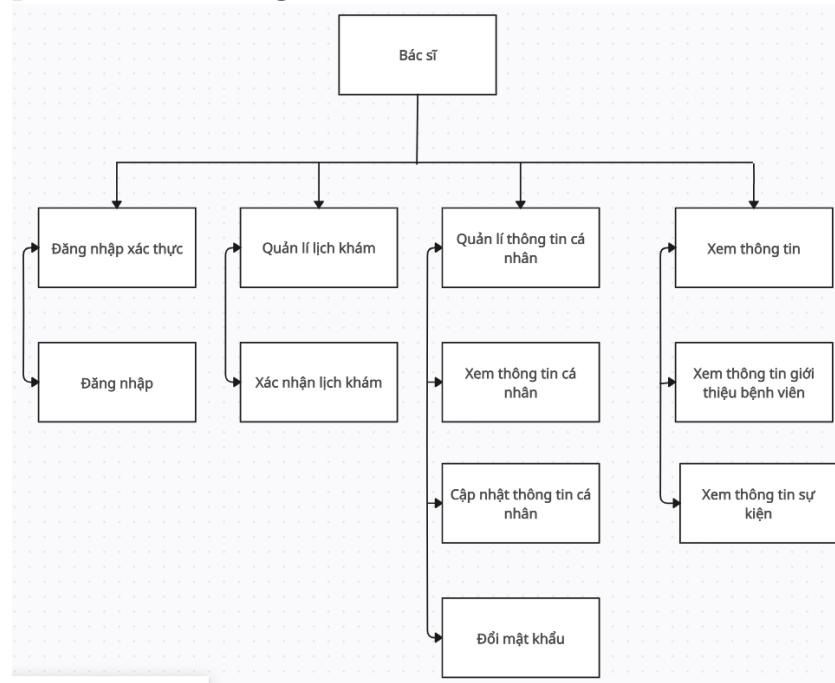
Ảnh 2.26. Sơ đồ phân rã chức năng của bệnh nhân

Chức năng đăng nhập xác thực kiểm tra tài khoản của bệnh nhân có hợp lệ hay không, có thể đăng ký tài khoản mới nếu chưa có tài khoản, có thể lấy lại mật khẩu nếu quên mật khẩu.

Sau khi đăng nhập, bệnh nhân có thể cập nhật các thông tin cá nhân của mình hoặc đổi mật khẩu.

Bệnh nhân chọn bác sĩ muốn đặt lịch khám sau đó đăng ký đặt lịch khám.

2.5.2. Sơ đồ phân rã chức năng của bác sĩ



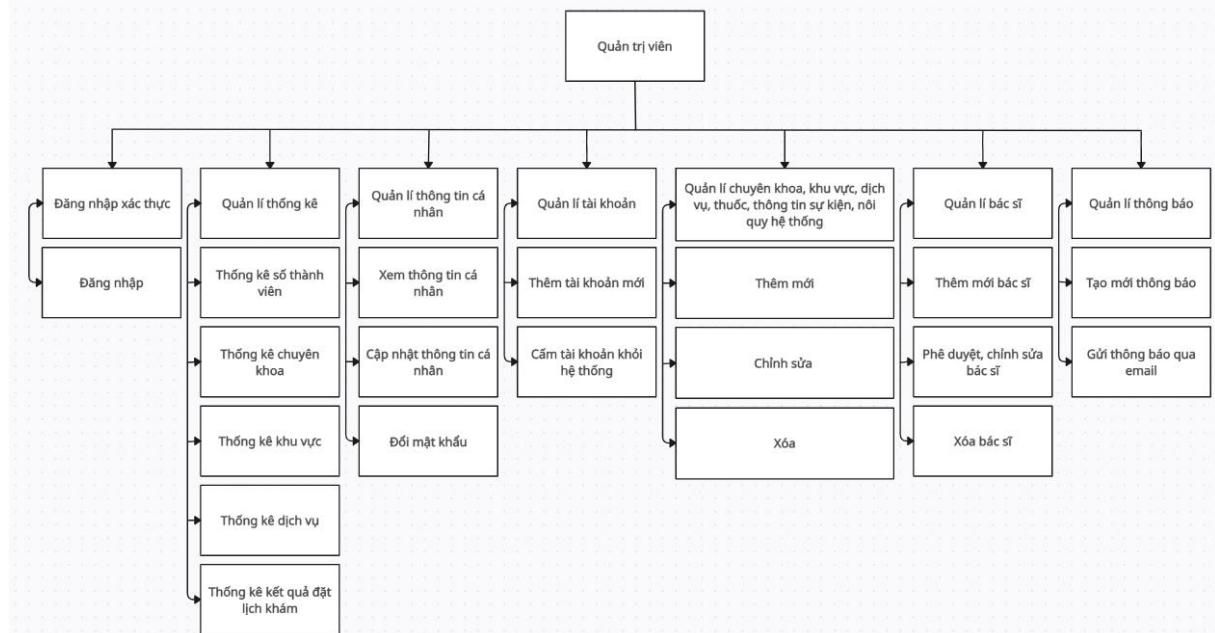
Ảnh 2.27. Sơ đồ phân rã chức năng của bác sĩ

Bác sĩ thực hiện đăng nhập vào hệ thống, có thể lấy lại mật khẩu

Sau khi đăng nhập vào bác sĩ có thể xem, thay đổi các thông tin cá nhân, đổi mật khẩu.

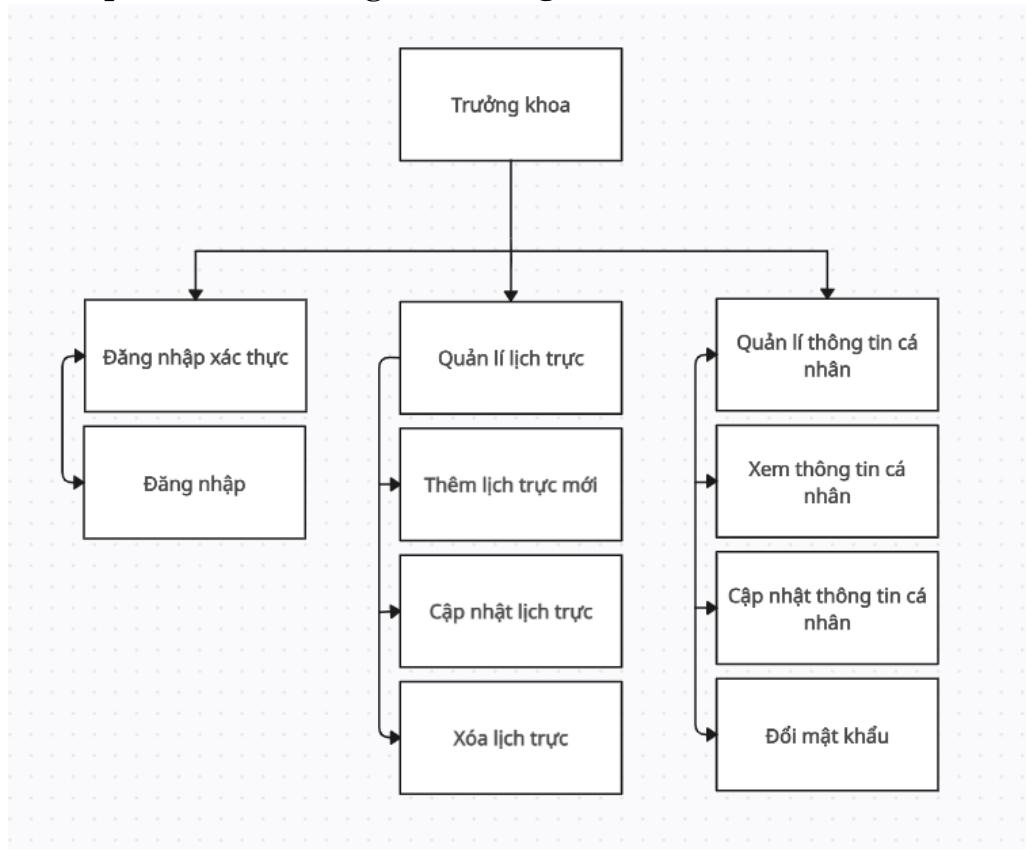
Quản lý lịch khám bệnh của mình, thay đổi các trạng thái của lịch khám như đang xem xét, đã xác nhận, hủy.

2.5.3. Sơ đồ phân rã chức năng của quản trị viên



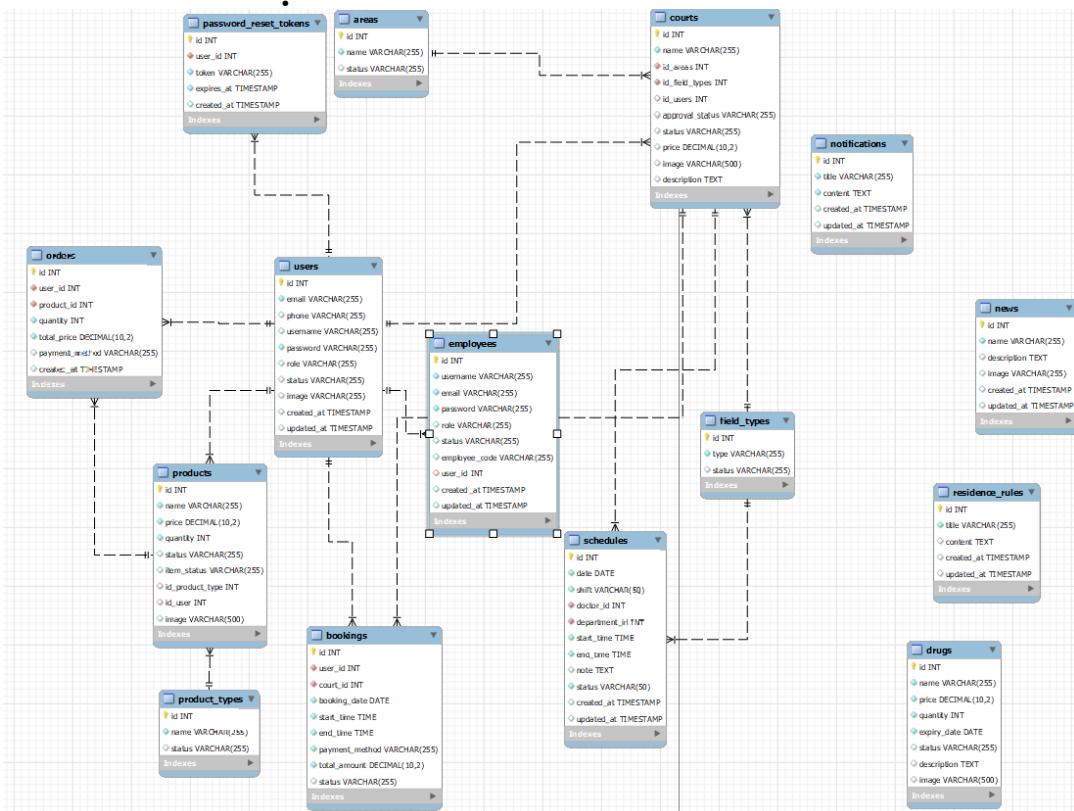
Ảnh 2.28. Sơ đồ phân rã chức năng của quản trị viên

2.5.4. Sơ đồ phân rã chức năng của trưởng khoa



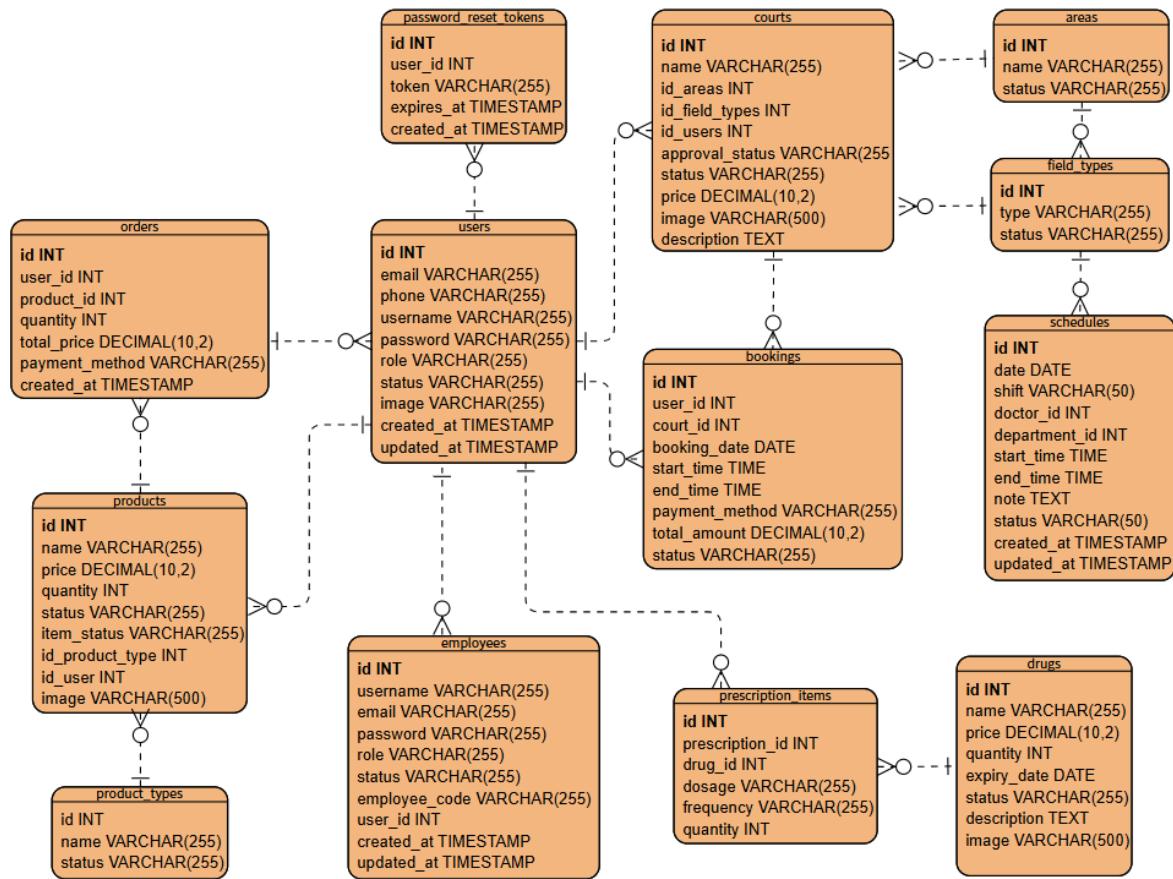
Ảnh 2.29. Sơ đồ phân rã chức năng của trưởng khoa

2.6. Sơ đồ cơ sở dữ liệu



Ảnh 2.30. Sơ đồ cơ sở dữ liệu

2.7. Sơ đồ ERD



Ảnh 2.31. Sơ đồ ERD

CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH

3.1. Phía người dùng

3.1.1. Giao diện trang chủ



Ảnh 3.1. Giao diện trang chủ

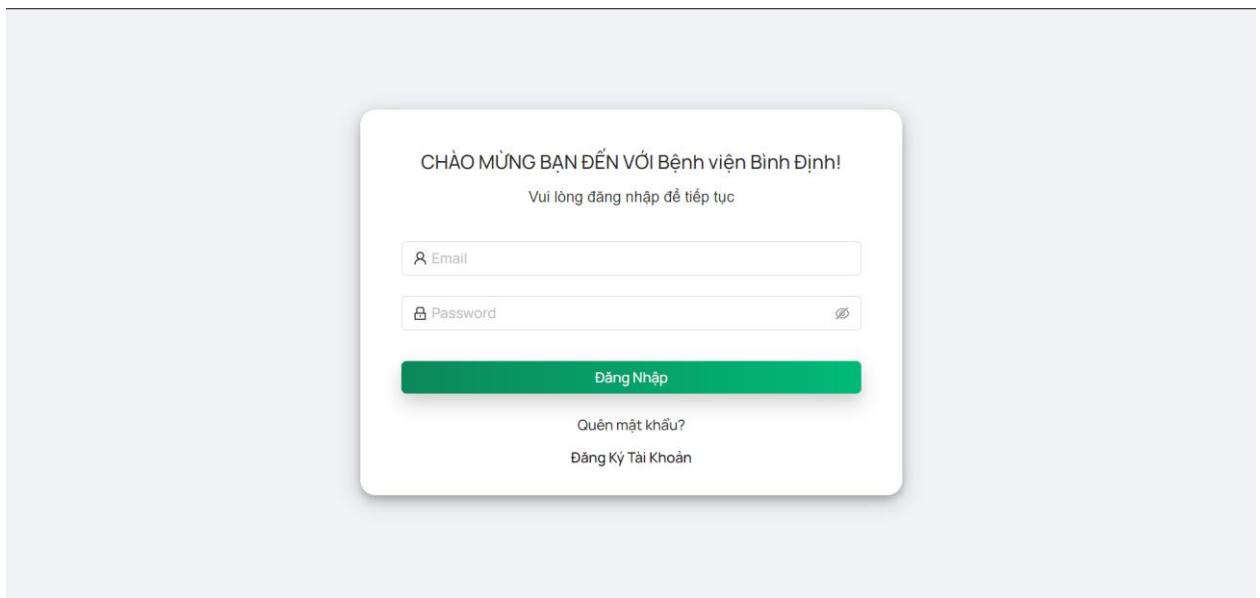
Đặt Lịch **Ngay**



Ảnh 3.2. Giao diện trang chủ

3.1.2. Giao diện đăng nhập

Để sử dụng các chức năng của trang web, người dùng cần phải đăng nhập tài khoản của mình.



Ảnh 3.3. Giao diện đăng nhập phía người dùng

3.1.3. Giao diện tạo tài khoản

Nếu người dùng chưa có tài khoản, người dùng sẽ phải đăng ký tài khoản để có thể sử dụng các chức năng của trang web.

Khi người dùng chọn đăng ký tài khoản, trang web sẽ chuyển hướng người dùng sang trang đăng ký tài khoản khách hàng.

Ảnh 3.4. Giao diện tạo tài khoản

Tại đây người dùng sẽ điền đầy đủ các thông tin như: tên hiển thị, mật khẩu, email, số điện thoại, sau đó nhấp vào đăng ký.

3.1.4. Giao diện quên mật khẩu

Nếu không nhớ mật khẩu mà mình đã tạo, người dùng có thể sử dụng chức năng quên mật khẩu. Sau khi nhấp vào quên mật khẩu, trang web sẽ chuyển người dùng sang trang quên mật khẩu.

Quên mật khẩu

Hủy Gửi đường dẫn đổi mật khẩu

Ảnh 3.5. Giao diện quên mật khẩu

Tại đây người dùng sẽ phải nhập đúng mật khẩu ứng với tài khoản mà mình đã đăng ký trước đó, trang web sẽ gửi một đường dẫn xác thực thay đổi mật khẩu về email của người dùng. Người dùng sẽ kiểm tra email, nhấp vào link xác nhận đổi mật khẩu để có thể đặt lại mật khẩu mới cho tài khoản của mình.

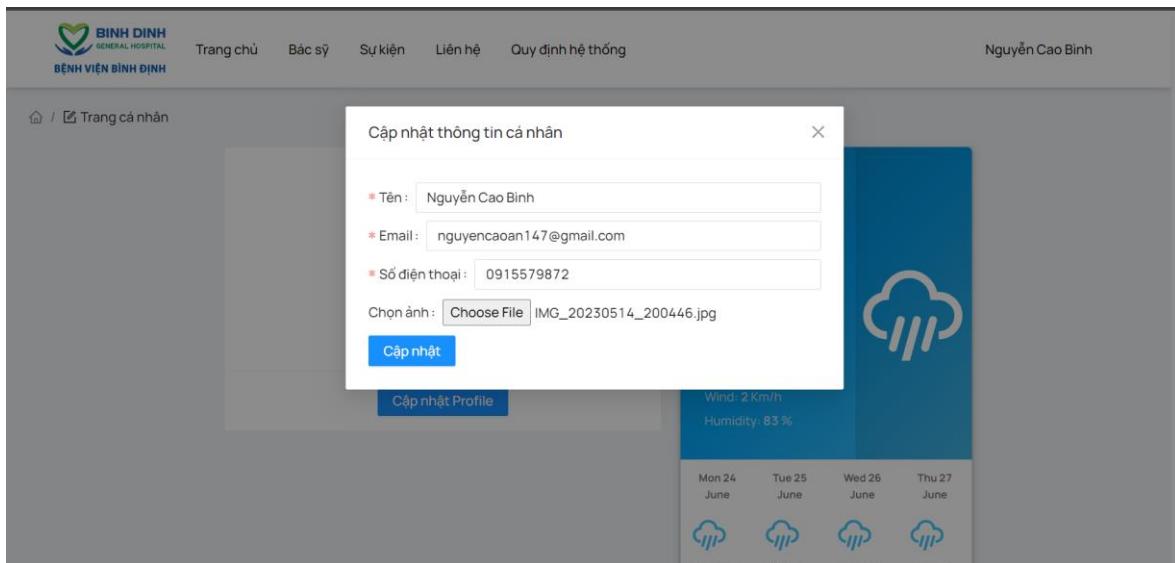
3.1.5. Giao diện trang cá nhân

Ở giao diện trang cá nhân, người dùng có thể xem một số thông tin cơ bản của tài khoản như tên hiển thị, email, số điện thoại

Ảnh 3.6. Giao diện trang cá nhân

3.1.6. Giao diện thay đổi thông tin cá nhân

Nếu muốn cập nhật thông tin cá nhân, người dùng sẽ chọn chức năng cập nhật Profile

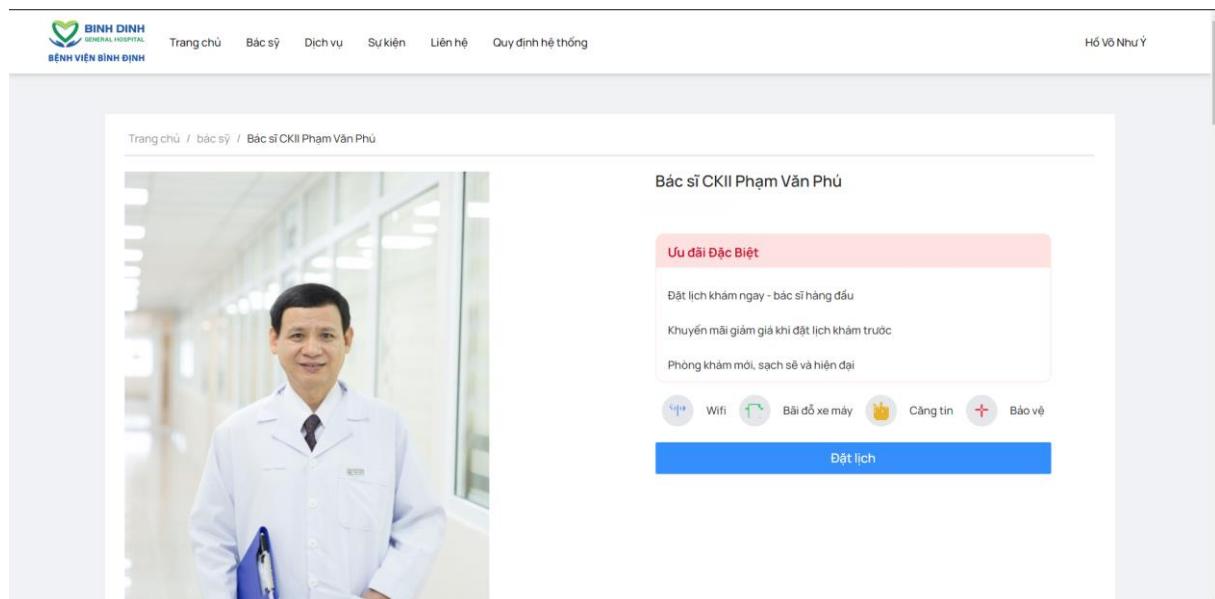


Ảnh 3.7. Giao diện cập nhật thông tin cá nhân

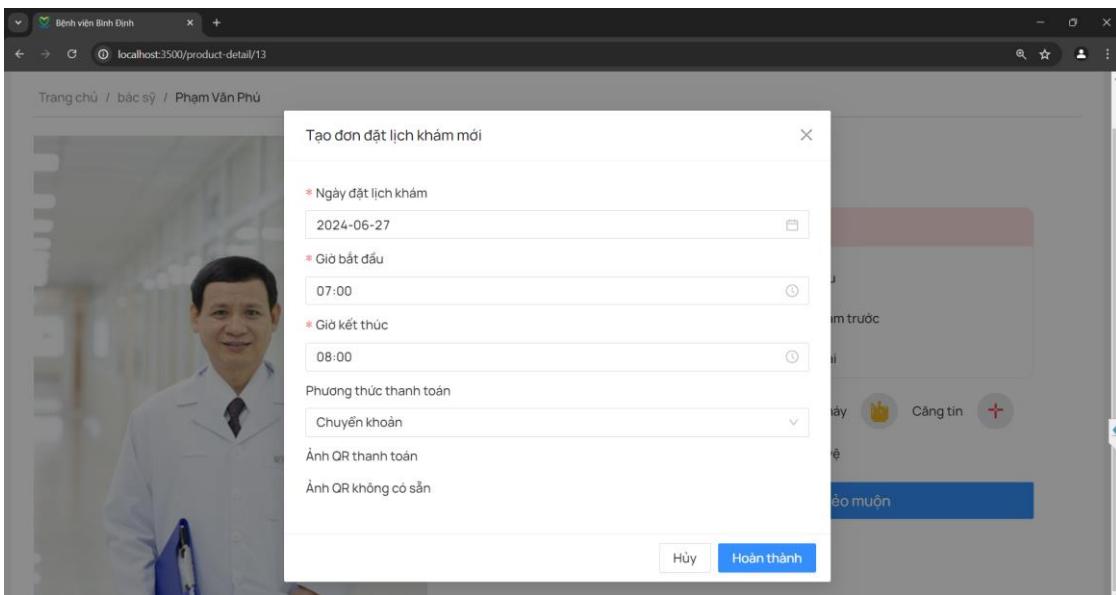
Tại đây người dùng có thể chỉnh sửa các thông tin của mình như: tên hiển thị, email, số điện thoại, ảnh đại diện

3.1.7. Giao diện đặt lịch khám bệnh

Khi người dùng muốn đặt lịch khám bệnh, người dùng sẽ vào trang web sau đó chọn bác sĩ mà mình muốn đặt lịch khám bệnh bằng cách nhấp vào “Đặt ngay”.



Ảnh 3.8. Giao diện chi tiết bác sĩ



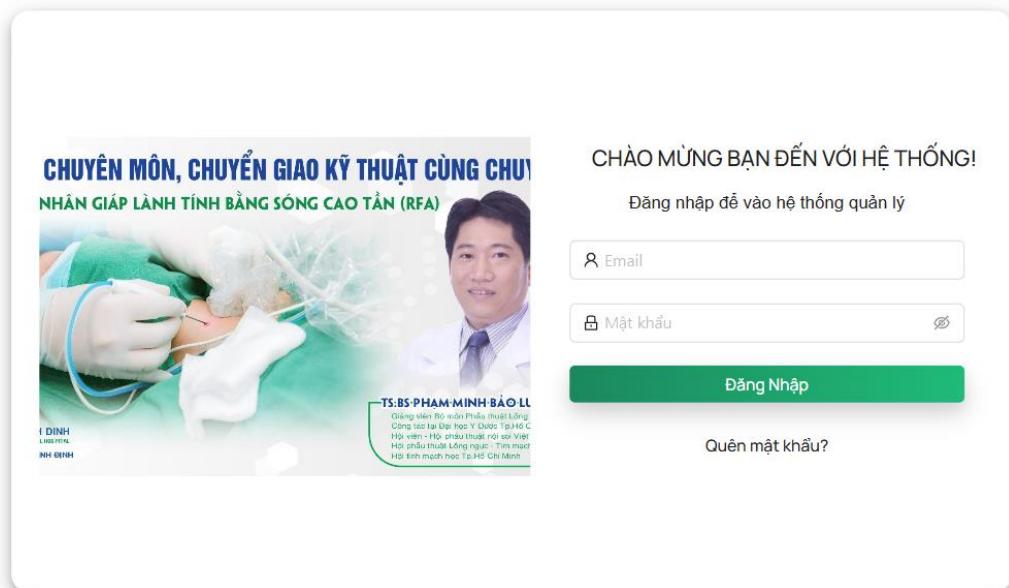
Ảnh 3.9. Giao diện đặt lịch khám bệnh

Người dùng sẽ nhập đầy đủ các thông tin cần thiết để có thể đặt được lịch khám.

3.2. Về phía quản trị viên

3.2.1. Giao diện đăng nhập

Để sử dụng các chức năng quản lý của trang web, quản trị viên cần đăng nhập vào hệ thống bằng tài khoản của mình.



Ảnh 3.10. Giao diện đăng nhập phía quản trị viên

3.2.2. Giao diện thống kê

Để thống kê lại các thông tin của trang web, admin sẽ sử dụng chức năng dashboards

Tên bác sĩ	Số Lượt Đặt	Tổng Doanh Thu
Cao Thị Thu Hà	3	1.475.000 đ
Phạm Văn Phú	1	400.000 đ
Trương Thị Quỳnh Hoa	1	593.333 đ
Hà Thị Diệp	1	491.667 đ

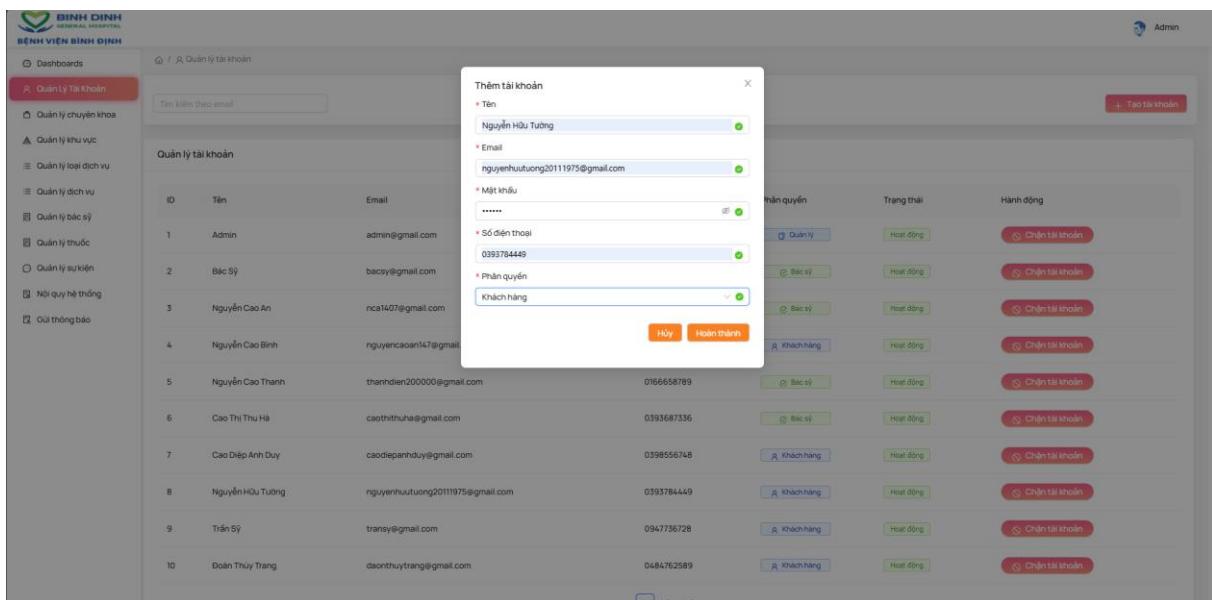
Ảnh 3.11. Giao diện thống kê

Tại đây admin có thể xem được các thông tin của trang web.

ID	Tên	Email	Số điện thoại	Phân quyền	Trạng thái	Hành động
1	Admin	admin@gmail.com	0938283923	Quản lý	Hoạt động	Chặn tài khoản
2	Bác Sỹ	bacsy@gmail.com	0938283923	Bác sỹ	Hoạt động	Chặn tài khoản
3	Nguyễn Cao An	nca1407@gmail.com	0915579872	Bác sỹ	Hoạt động	Chặn tài khoản
4	Nguyễn Cao Bình	nguyencacaan147@gmail.com	0915579872	Khách hàng	Hoạt động	Chặn tài khoản
5	Nguyễn Cao Thành	thanhndien200000@gmail.com	0166658789	Bác sỹ	Hoạt động	Chặn tài khoản
6	Cao Thị Thu Hà	caothithuha@gmail.com	0393687336	Bác sỹ	Hoạt động	Chặn tài khoản
7	Cao Diệp Anh Duy	caodiapanhduy@gmail.com	0398556748	Khách hàng	Hoạt động	Chặn tài khoản
8	Nguyễn Hữu Tường	nguyenhuutuong20111975@gmail.com	0393784449	Khách hàng	Hoạt động	Chặn tài khoản
9	Trần Sỹ	transy@gmail.com	0947736728	Khách hàng	Hoạt động	Chặn tài khoản
10	Đoàn Thúy Trang	daonthuytrang@gmail.com	0484762589	Khách hàng	Hoạt động	Chặn tài khoản

Ảnh 3.12. Giao diện quản lý tài khoản

Để xem thông tin các tài khoản, admin sẽ sử dụng chức năng quản lý tài khoản. Tại đây admin có thể chặn tài khoản ra khỏi hệ thống của trang web. Admin có thể thêm tài khoản mới bằng cách sử dụng chức năng tạo tài khoản.



Ảnh 3.13. Giao diện thêm tài khoản

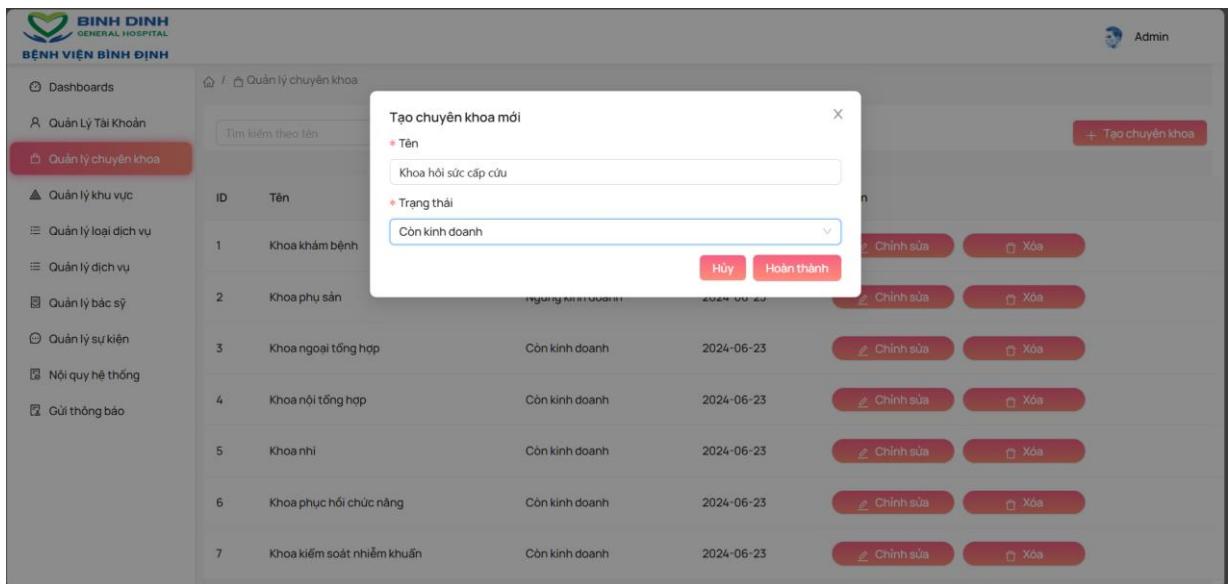
Ở chức năng thêm tài khoản mới, admin sẽ điền đầy đủ các thông tin của tài khoản muốn tạo mới như: tên, email, mật khẩu, số điện thoại và phân quyền của tài khoản.

3.2.3. Giao diện quản lý chuyên khoa

ID	Tên	Trạng thái	Ngày tạo	Action
1	Khoa khám bệnh	Còn kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>
2	Khoa phụ sản	Ngừng kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>
3	Khoa ngoại tổng hợp	Còn kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>
4	Khoa nội tổng hợp	Còn kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>
5	Khoa nhi	Còn kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>
6	Khoa phục hồi chức năng	Còn kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>
7	Khoa kiểm soát nhiễm khuẩn	Còn kinh doanh	2024-06-23	<button>Chỉnh sửa</button> <button>Xóa</button>

Ảnh 3.14. Giao diện quản lý chuyên khoa

Để quản lý các chuyên khoa của trang web, admin sẽ sử dụng chức năng quản lý chuyên khoa. Tại đây admin có thể thêm chuyên khoa mới, chỉnh sửa chuyên khoa đã có, xóa chuyên khoa.



Ảnh 3.15. Giao diện tạo chuyên khoa mới

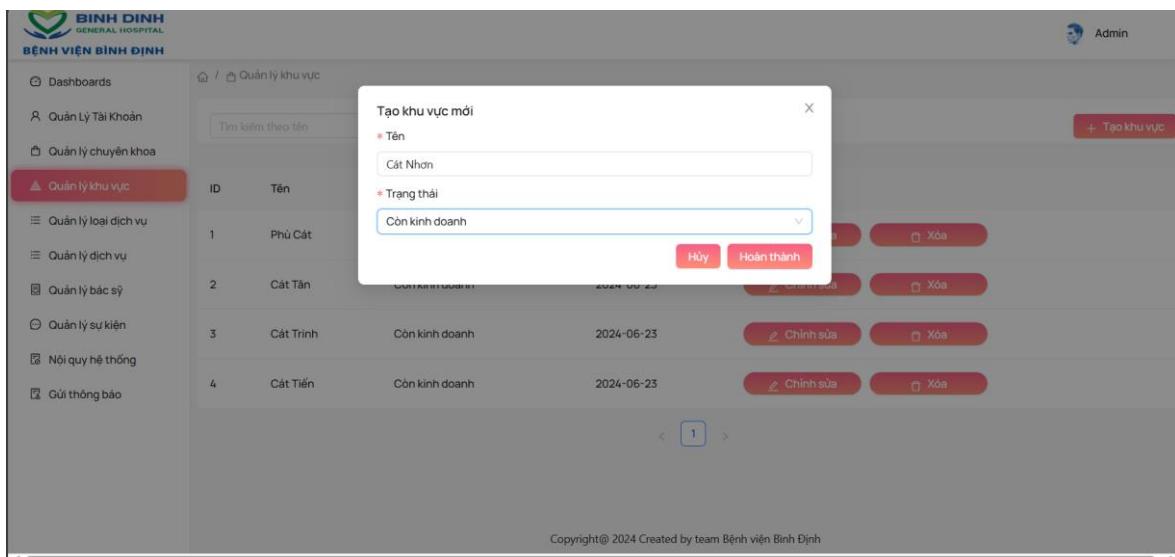
Khi chọn chức năng thêm chuyên khoa mới, admin sẽ điền đầy đủ các thông tin của chuyên khoa như: tên chuyên khoa, trạng thái hoạt động của chuyên khoa.

3.2.4. Giao diện quản lý khu vực

ID	Tên	Trạng thái	Ngày tạo	Action
1	Phù Cát	Ngừng kinh doanh	2024-06-23	
2	Cát Tân	Còn kinh doanh	2024-06-23	
3	Cát Trinh	Còn kinh doanh	2024-06-23	
4	Cát Tiến	Còn kinh doanh	2024-06-23	

Ảnh 3.16. Giao diện quản lý khu vực

Quản lý khu vực sẽ là nơi để admin thêm thông tin về khu vực làm việc của các bác sĩ có trong hệ thống.



Ảnh 3.17. Giao diện thêm khu vực mới

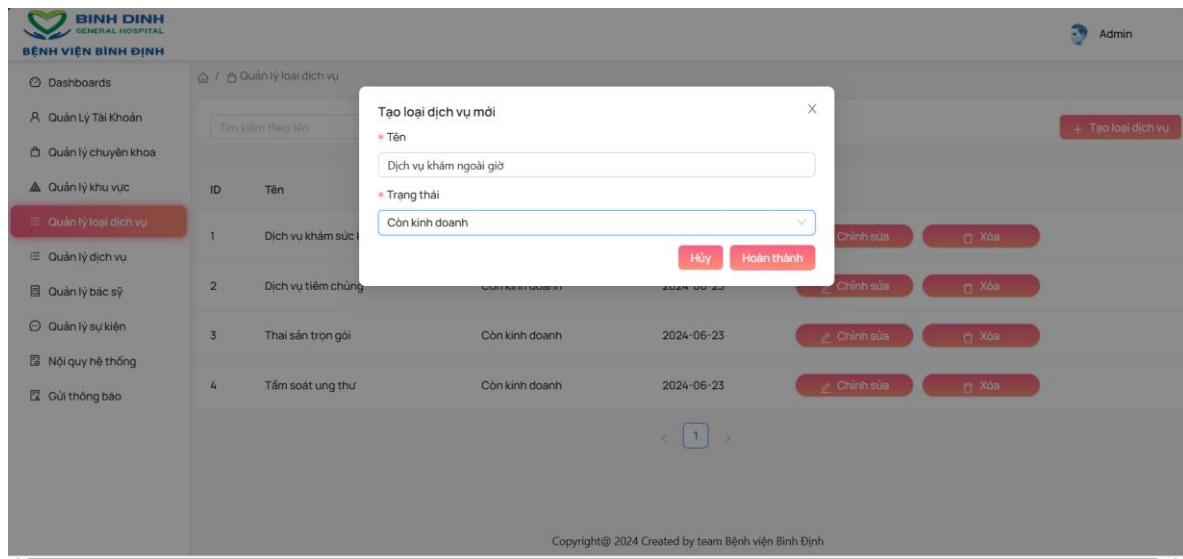
Admin có thể thêm khu vực làm việc mới của bác sĩ. Thông tin của khu vực mới sẽ bao gồm tên khu vực, trạng thái hoạt động của khu vực.

3.2.5. Giao diện quản lý dịch vụ

ID	Tên	Trạng thái	Ngày tạo	Action
1	Dịch vụ khám sức khỏe	Ngừng kinh doanh	2024-06-23	Chỉnh sửa Xóa
2	Dịch vụ tiêm chủng	Còn kinh doanh	2024-06-23	Chỉnh sửa Xóa
3	Thai sản trọn gói	Còn kinh doanh	2024-06-23	Chỉnh sửa Xóa
4	Tầm soát ung thư	Còn kinh doanh	2024-06-23	Chỉnh sửa Xóa

Ảnh 3.18. Giao diện quản lý loại dịch vụ

Admin sẽ quản lý được các loại hình dịch vụ đã và đang cung cấp cho khách hàng.



Ảnh 3.19. Giao diện thêm mới loại dịch vụ mới

ID	Ảnh	Tên	Số lượng	Loại dịch vụ	Trạng thái	Giá trị	Ngày tạo	Action
1		Khám, tầm soát sức khỏe ngay cùng Chuyên Gia Đầu Ngành	24	Thái sản trọn gói	active	12.000.000 đ	2024-06-24	
2		Gói tầm soát tim mạch	20	Gói khám sức khỏe chuyên sâu	active	5.000.000 đ	2024-06-24	
3		Gói tầm soát ung thư đường tiêu hóa	20	Tầm soát ung thư	active	10.000.000 đ	2024-06-24	

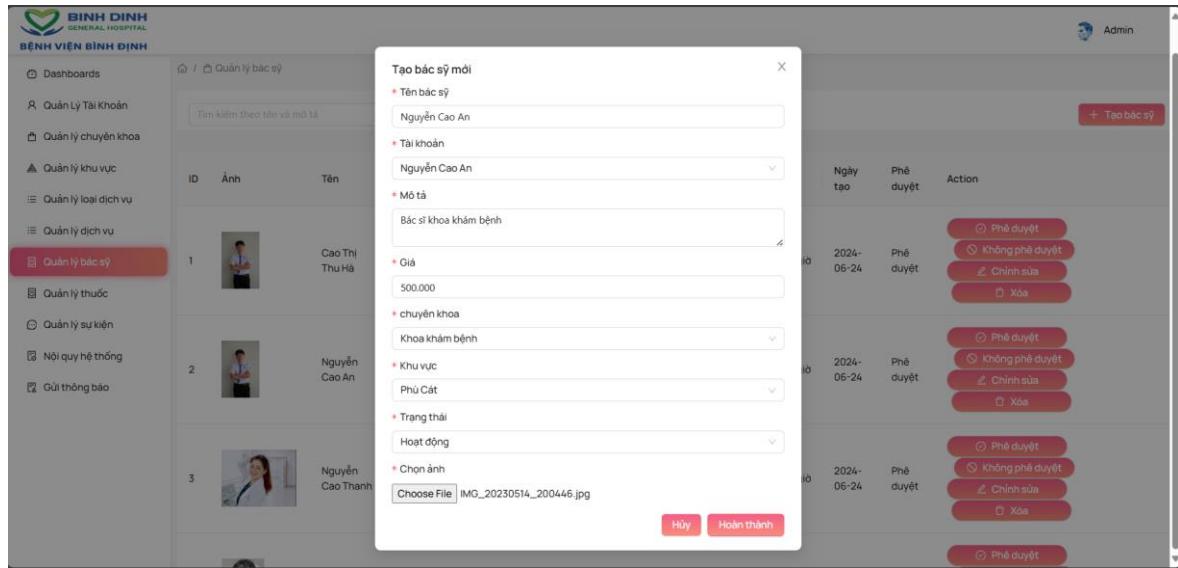
Ảnh 3.20. Giao diện quản lý dịch vụ

3.2.6. Giao diện quản lý bác sĩ

ID	Ảnh	Tên	Mô tả	Khu vực	Chuyên khoa	Bác sĩ	Trạng thái	Giá trị	Ngày tạo	Phê duyệt	Action
1		Cao Thị Thu Hà	Bác sĩ CKII	Quy Nhơn	Khoa phục hồi chức năng	Cao Thị Thu Hà	Hoạt động	500.000 đ/giờ	2024-06-24	<input checked="" type="radio"/> Phê duyệt <input type="radio"/> Không phê duyệt 	
2		Nguyễn Cao An	Bác sĩ CKI	An Lão	Khoa khám bệnh	Nguyễn Cao An	Hoạt động	300.000 đ/giờ	2024-06-24	<input checked="" type="radio"/> Phê duyệt <input type="radio"/> Không phê duyệt 	
3		Nguyễn Cao Thành	Bác sĩ CKII	Hoài Ân	Khoa xét nghiệm	Nguyễn Cao Thành	Hoạt động	500.000 đ/giờ	2024-06-24	<input checked="" type="radio"/> Phê duyệt <input type="radio"/> Không phê duyệt 	
4		Trương Thị Quỳnh Hoa	Bác sĩ CKII	Văn Canh	Khoa hồi sức cấp cứu	Trương Thị Quỳnh Hoa	Hoạt động	400.000 đ/giờ	2024-06-24	<input checked="" type="radio"/> Phê duyệt <input type="radio"/> Không phê duyệt 	

Ảnh 3.21. Giao diện quản lý bác sĩ

Ở chức năng quản lý bác sĩ, admin có thể quản lý được toàn bộ bác sĩ đang có trong hệ thống. Các thao tác ở chức năng này sẽ là phê duyệt hoặc từ chối phê duyệt bác sĩ đã được thêm, chỉnh sửa thông tin của bác sĩ, xóa bác sĩ.



Ảnh 3.22. Giao diện thêm mới bác sĩ

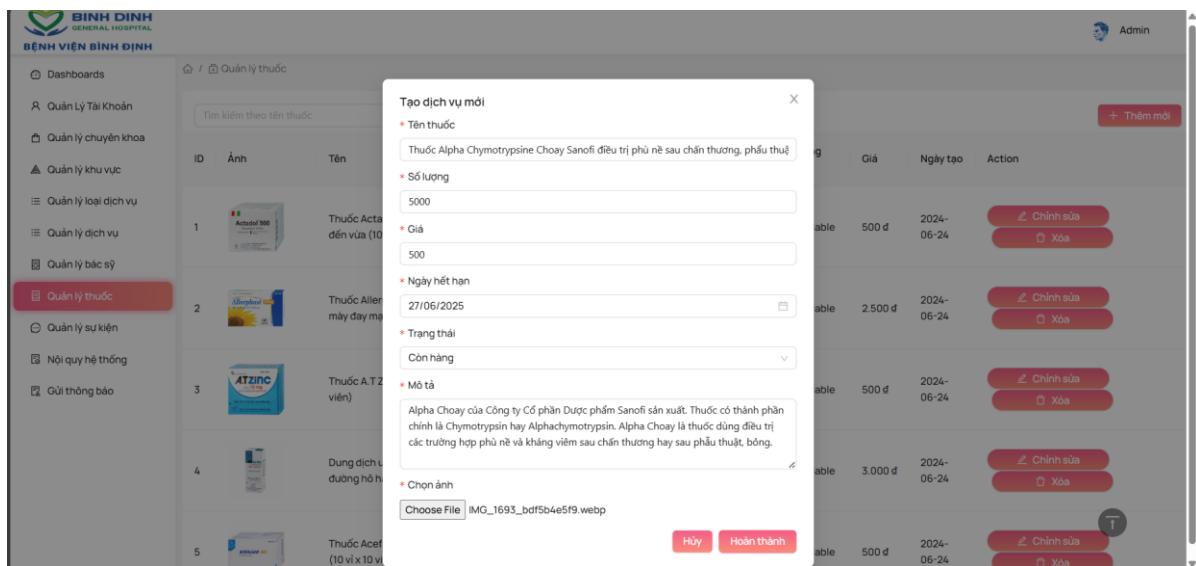
Ở chức năng này, admin sẽ thêm bác sĩ mới vào trong hệ thống, điều kiện ở đây là bác sĩ đã có tài khoản và phân quyền của tài khoản đó phải là bác sĩ.

3.2.7. Giao diện quản lý thuốc

ID	Ảnh	Tên	Số lượng	Loại thuốc	Trạng thái	Giá	Ngày tạo	Action
1		Thuốc Actadol 500 Medipharco điều trị các chứng đau và sốt từ nhẹ đến vừa (10 vỉ x 10 viên)	500		available	500 đ	2024-06-24	
2		Thuốc Allerphast 180mg Mebiphar điều trị viêm mũi dị ứng theo mùa, mày đay mạn tính và cảm (1 vỉ x 10 viên)	500		available	2.500 đ	2024-06-24	
3		Thuốc A.T Zinc 10mg An Thiên điều trị tiêu chảy kéo dài (10 vỉ x 10 viên)	1000		available	500 đ	2024-06-24	
4		Dung dịch uống Atilude 250mg/5ml An Thiên tiêu nhầy, rối loạn đường hô hấp (6 vỉ x 5 ống x 5ml)	1000		available	3.000 đ	2024-06-24	
5		Thuốc Acetofalgan 500mg Euvipharm giảm đau nhẹ đến vừa, hạ sốt (10 vỉ x 10 viên)	600		available	500 đ	2024-06-24	

Ảnh 3.23. Giao diện quản lý thuốc

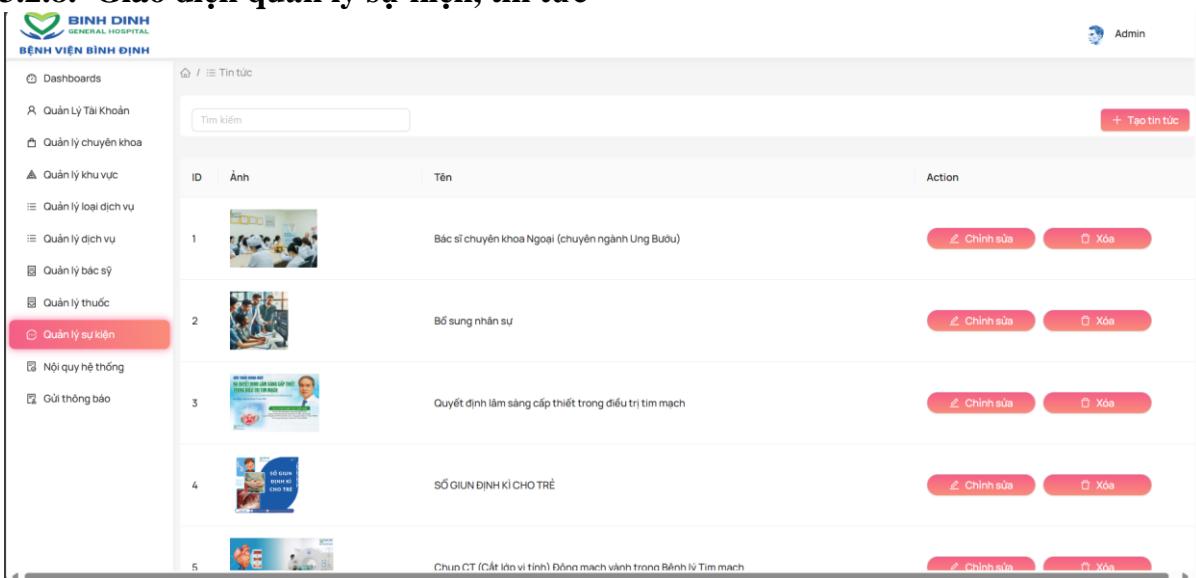
Admin sẽ thêm, chỉnh sửa hoặc xóa thuốc trong hệ thống. Các thông tin của thuốc cơ bản sẽ có những thông tin như: ảnh, tên thuốc, số lượng, trạng thái, giá.



Ảnh 3.24. Giao diện thêm mới thuốc

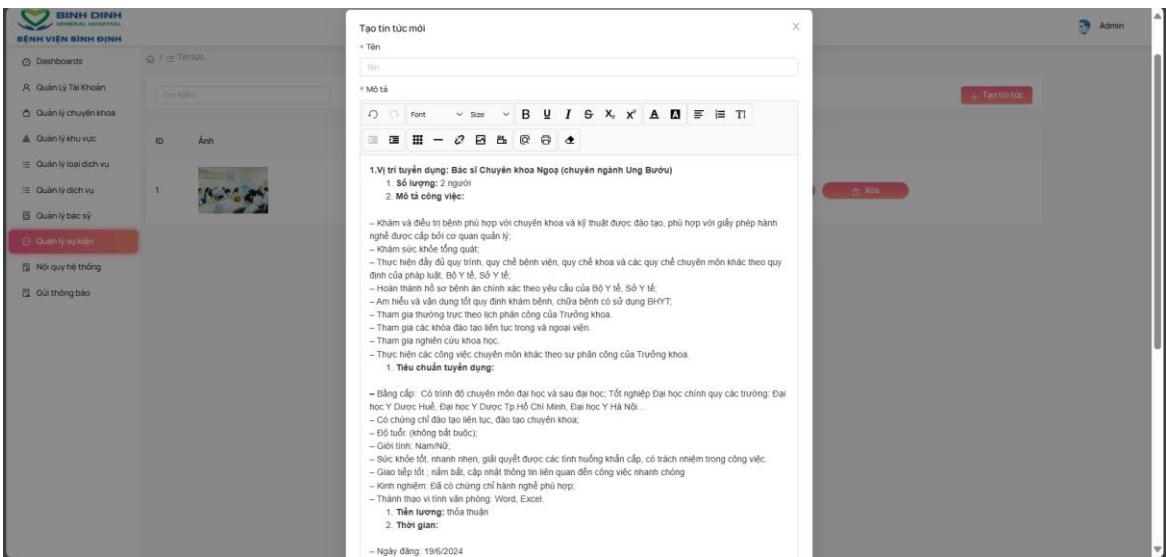
Admin sẽ thêm mới thuốc vào trong hệ thống. Các thông tin cơ bản của thuốc sẽ cần được điền đầy đủ như: tên thuốc, số lượng đang có, giá, ngày hết hạn của thuốc, trạng thái (đang kinh doanh, ngừng kinh doanh), mô tả ngắn về chức năng của thuốc, ảnh của thuốc.

3.2.8. Giao diện quản lý sự kiện, tin tức



Ảnh 3.25. Giao diện quản lý sự kiện, tin tức

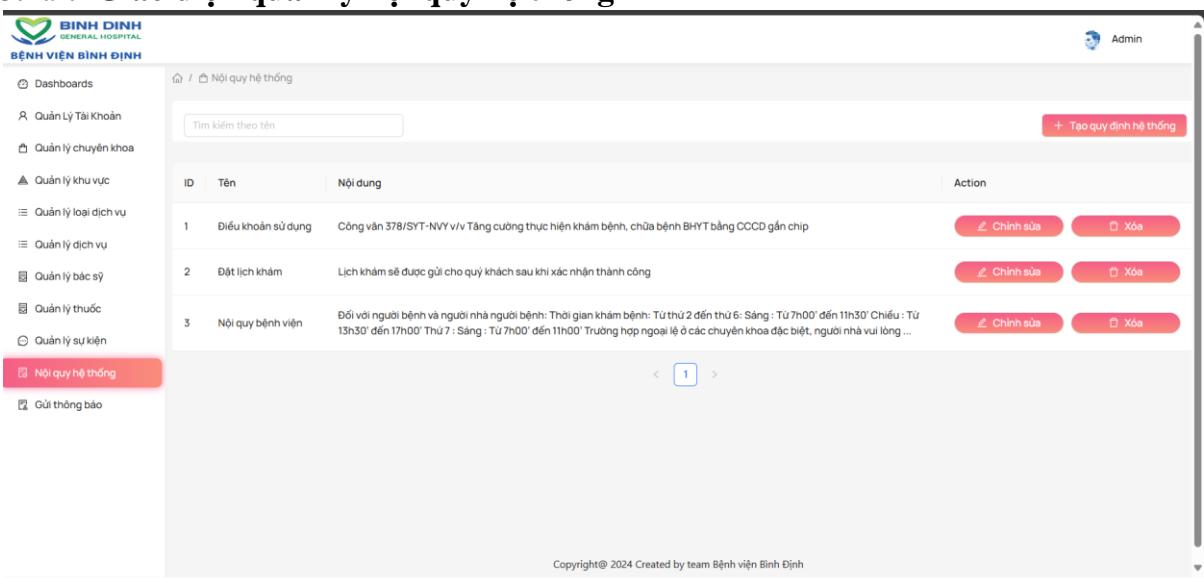
Admin sẽ thêm các sự kiện, tin tức để giúp người dùng nắm được các thông tin từ hệ thống. Các sự kiện tin tức này sẽ hiển thị ở trang chủ khi người dùng, bệnh nhân truy cập vào trang web.



Ảnh 3.26. Giao diện thêm sự kiện, tin tức

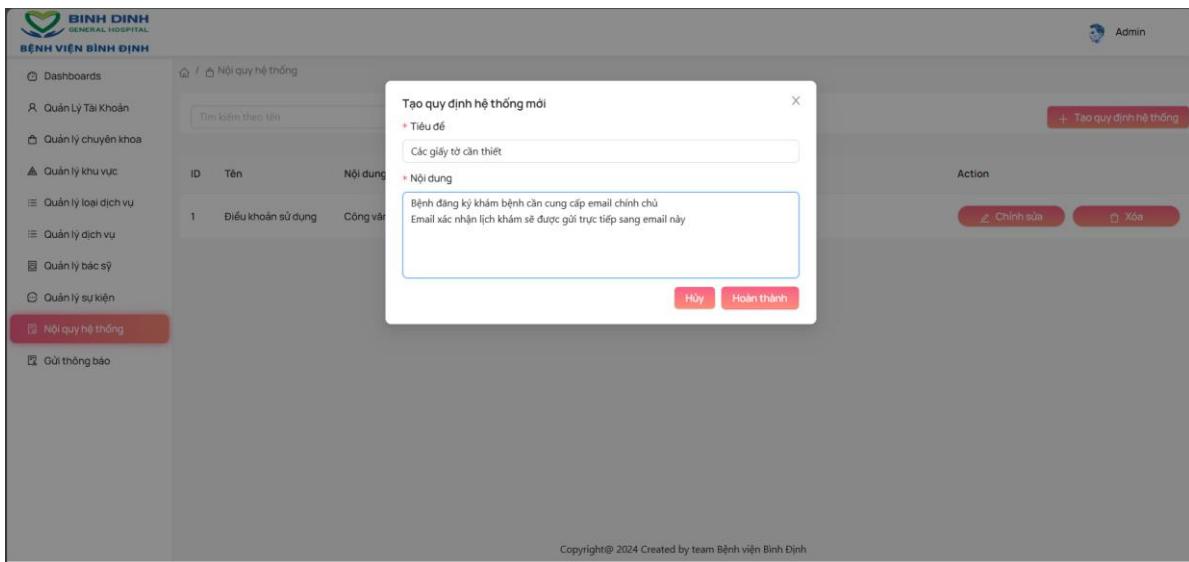
Admin thêm mới tin tức, sự kiện với các thông tin sau: tên sự kiện, tin tức, nội dung của sự kiện tin tức.

3.2.9. Giao diện quản lý nội quy hệ thống



Ảnh 3.27. Giao diện quản lý nội quy hệ thống

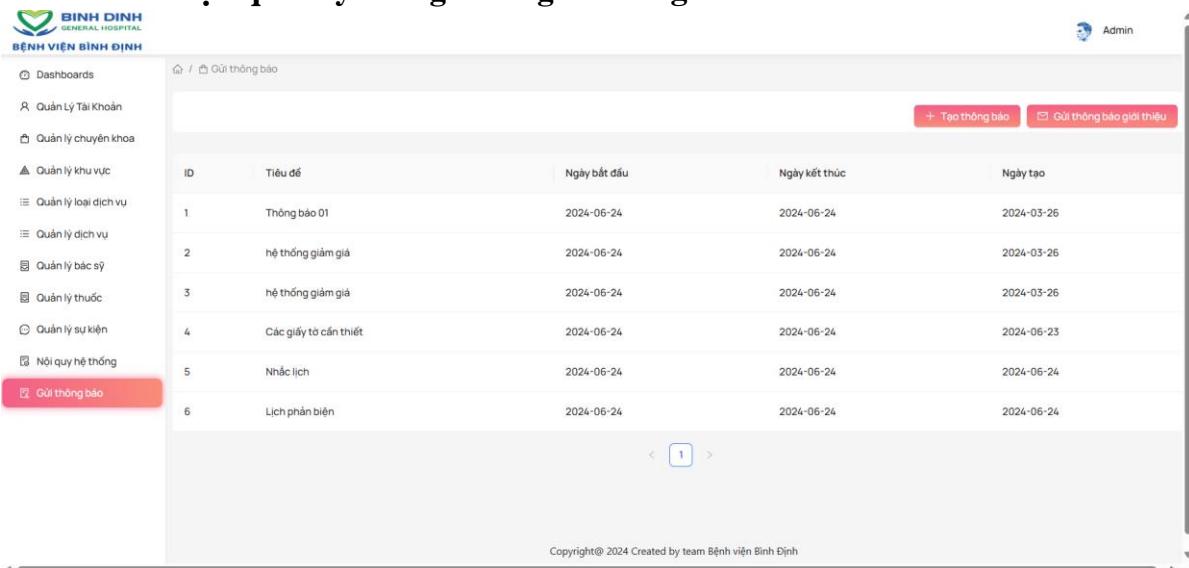
Ở chức năng này, admin có thể quản lý toàn bộ nội quy của hệ thống. Admin có thể thêm, chỉnh sửa, xóa các nội quy của hệ thống.



Ảnh 3.28. Giao diện thêm mới nội quy hệ thống

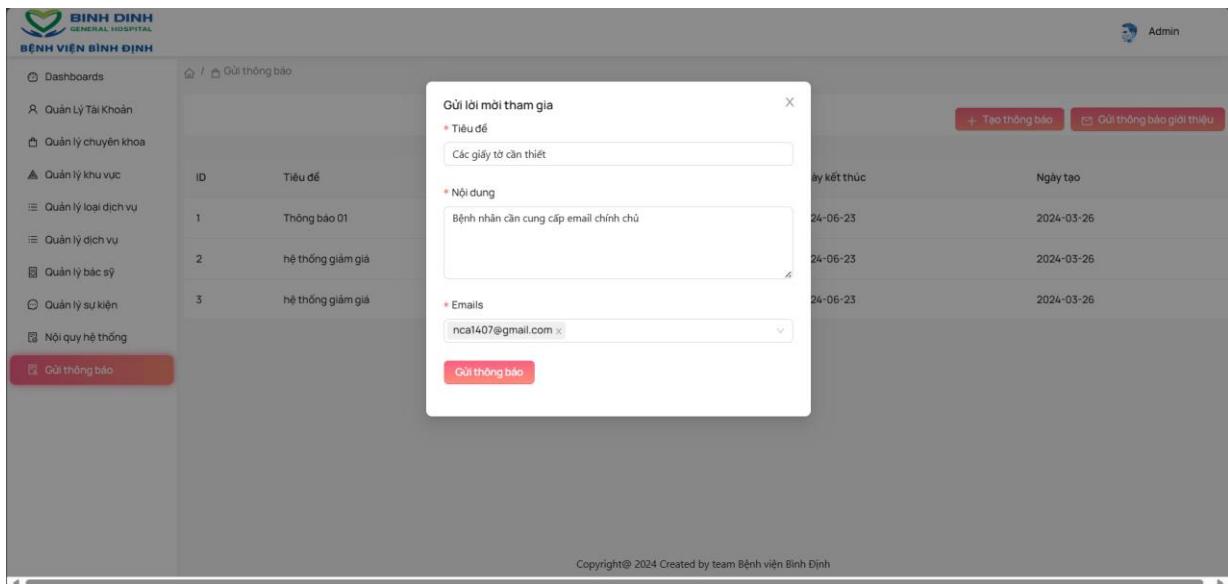
Admin sẽ thêm nội quy hệ thống mới với các thông tin: tiêu đề, nội dung. Nội quy hệ thống sẽ được hiển thị ở mục nội quy ở trang chủ.

3.2.10. Giao diện quản lý thông báo người dùng



Ảnh 3.29. Giao diện quản lý thông báo gửi qua email

Ở chức năng này, admin có thể gửi các thông báo trực tiếp đến email của người dùng khi có một thông báo nào đó từ hệ thống.



Ảnh 3.30. Giao diện gửi thông báo qua email người dùng

Chức năng gửi thông báo yêu cầu nhập đầy đủ các thông tin như tiêu đề, nội dung thông báo, email của người nhận thông báo

3.3. Về phía bác sĩ

3.3.1. Giao diện đăng nhập

Để sử dụng các chức năng quản lý của trang web, bác sĩ cần phải đăng nhập vào hệ thống bằng tài khoản của mình



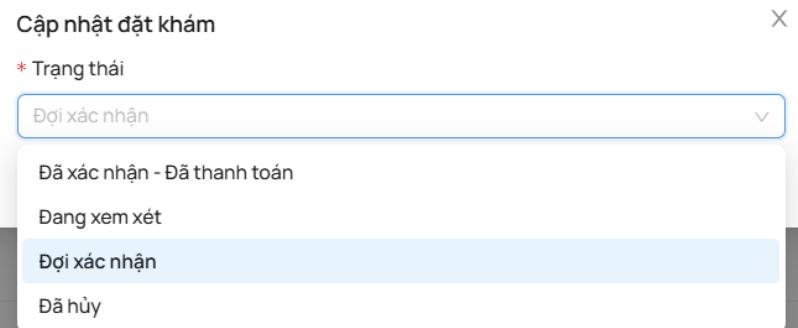
Ảnh 3.31 Giao diện đăng nhập phía quản trị viên

3.3.2. Giao diện quản lý khám bệnh

Khi người dùng đặt lịch khám thành công, thông tin đặt lịch khám của bệnh nhân sẽ hiển thị trong giao diện quản lý đặt lịch khám với trạng thái đợi xác nhận

Copyright@ 2024 Created by team Bệnh viện Bình Định

Ảnh 3.32. Giao diện quản lý bệnh nhân khám bệnh



Ảnh 3.33. Giao diện cập nhật lịch đặt khám

Bác sĩ sẽ lựa chọn trạng thái của lịch đặt bằng cách nhấp vào chỉnh sửa

Hóa đơn khám bệnh		Ngày: 6/24/2024
Tên:	Cao Thị Thu Hà	
Ngày đặt:	25/06/2024 00:00	
Giờ bắt đầu:	07:00:00	
Giờ kết thúc:	08:00:00	
Tổng tiền:	491666.67	

Ảnh 3.34. Hóa đơn khám bệnh

Sau bác sĩ đã xác nhận lịch đặt, bác sĩ và bệnh nhân có thể xuất được hóa đơn đặt lịch khám bệnh

3.4. Về phía trưởng khoa

3.4.1. Giao diện phân lịch trực

ID	Ngày	Ca làm việc	Bác sĩ	Phòng ban	Thời gian bắt đầu	Thời gian kết thúc	Ghi chú	Trạng thái	Hành động
4	2024-06-25	morning	Cao Thị Thu Hà	Khoa khám bệnh	07:00	11:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
5	2024-06-25	afternoon	Nguyễn Cao An	Khoa khám bệnh	13:00	18:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
6	2024-06-25	evening	Nguyễn Cao Thành	Khoa khám bệnh	13:00	05:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
7	2024-06-26	morning	Trương Thị Quỳnh Hoa	Khoa phụ sản	07:00	11:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
8	2024-06-26	afternoon	Hà Thị Diệp	Khoa phụ sản	07:00	11:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
9	2024-06-26	evening	Phạm Văn Phú	Khoa phụ sản	18:00	05:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
17	2024-06-26	morning	Nguyễn Cao An	Khoa khám bệnh	05:00	07:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>
18	2024-06-28	afternoon	Nguyễn Cao Thành	Khoa ngoại tổng hợp	13:00	17:00	Thực hiện công việc đã thông báo ở giao ban	active	<button>Chỉnh sửa</button> <button>Xóa</button>

Copyright© 2024 Created by team Bệnh viện Bình Định

Ảnh 3.35. Giao diện phân lịch trực

Trưởng khoa có thể xem được lịch trực của cán bộ y bác sĩ với các thông tin cơ bản như: thời gian trực, ca trực, bác sĩ được phân công trực, phòng ban, thời gian bắt đầu – kết thúc ca trực, ghi chú cho ca trực, trạng thái ca trực. Trưởng khoa có thể chỉnh sửa hoặc xóa ca trực

Tạo mới lịch trực

* Ngày: 2024-06-30

* Ca làm việc: Ca sáng

* Bác sĩ: Nguyễn Cao An

* Phòng ban: Khoa khám bệnh

* Thời gian bắt đầu: 07:00

* Thời gian kết thúc: 11:00

* Ghi chú: Thực hiện công việc đã thông báo ở giao ban

* Trạng thái: Hoạt động

Cancel OK

Copyright© 2024 Created by team Bệnh viện Bình Định

Ảnh 3.36. Giao diện tạo lịch trực

Trưởng khoa sẽ phân lịch trực cho đội ngũ y bác sĩ. Nội dung của phân lịch trực sẽ bao gồm chọn ngày trực, ca làm việc, bác sĩ đảm nhiệm ca trực, phòng ban trực, thời gian bắt đầu vào ca trực, thời gian kết thúc ca trực, ghi chú của ca trực, trạng thái của ca trực.

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết quả đạt được:

Sau khoảng thời gian học tập và rèn luyện, em đã cố gắng hoàn thành đồ án tốt nghiệp với đề tài “Xây dựng Website đặt lịch khám bệnh tại phòng khám Bình Định”. Qua đề tài này, bản thân em đã đạt được một số kết quả như sau: Tìm hiểu và ứng dụng React, Node, MySQL,.. để xây dựng nền trang web, cải thiện được kỹ năng làm báo cáo cho sản phẩm, rèn luyện khả năng tìm hiểu và học hỏi từ các nguồn tài liệu, hoàn thiện được trang web với những chức năng cơ bản, bám sát vào thực tiễn với những nhu cầu của người dùng.

Về mặt cơ bản trang web đã được được những kết quả như sau

- Hệ thống quản lý đã giúp tự động hóa nhiều quy trình công việc, giảm bớt công việc giấy tờ và tăng cường hiệu quả quản lý.
- Quản lý thông tin bệnh nhân, lịch hẹn và dịch vụ y tế một cách hệ thống và chính xác.
- Trang web giúp bệnh nhân dễ dàng đăng ký khám bệnh, tra cứu thông tin y tế, và theo dõi lịch hẹn trực tuyến.
- Nhân viên y tế có thể dễ dàng quản lý và theo dõi các nhiệm vụ của mình, từ việc điều phối lịch hẹn đến quản lý kho dược phẩm.
- Giảm bớt tình trạng chờ đợi và thời gian xử lý thủ tục, giúp nhân viên tập trung hơn vào chăm sóc bệnh nhân.
- Đảm bảo tuân thủ các quy định về bảo mật thông tin và quyền riêng tư của bệnh nhân.

2. Hạn chế:

Tuy đề tài hiện tại đã đến giai đoạn hoàn thành nhưng sơ bộ vẫn còn nhiều mặt hạn chế.

Về bản thân:

- Khả năng tìm hiểu là có, tuy còn hạn chế
- Chưa phân bổ thời gian hợp lý để đảm bảo tiến độ thực hiện đồ án.

Về sản phẩm:

- Sản phẩm chưa đáp ứng hoàn toàn được các nhu cầu cần có trong thực tiễn, cần được phát triển thêm
- Hiệu suất của trang web vẫn còn chưa ổn định
- Các thành phần chi tiết chưa được trau truốt chỉnh chu.

3. Hướng phát triển

Tiếp tục khảo sát thực tế phát triển các chức năng sao cho phù hợp với nhu cầu người dùng hệ thống, tìm hiểu cập nhật thêm công nghệ để ứng dụng vào sản phẩm, cải thiện hiệu suất sản phẩm, sửa lỗi các chức năng chưa được hoàn thiện.

PHỤ LỤC

Link Github

https://github.com/caoank61/Website_QuanLyBenhVienBinhDinh_New.git

Hướng dẫn cài đặt

- Cài đặt node
- Cài đặt CSDL: file CSDL Database/database.sql
- Thay đổi đường dẫn CSDL các file theo đường dẫn HealthBooker-Backend/app/config
- Chạy trang web bằng lệnh “npm start”
- Sử dụng tài khoản admin: Tài khoản: admin@gmail.com, mật khẩu: 123456
- Sử dụng tài khoản trưởng khoa: Tài khoản: truongkhoa@gmail.com, mật khẩu: 123456

Tài liệu tham khảo

- [1] "**Tự học HTML cơ bản và nâng cao**" - Trần Thị Hải, 2019, Nhà Xuất Bản Bách Khoa Hà Nội.
- [2] "**Học lập trình CSS từ cơ bản đến nâng cao**" - Nguyễn Văn Minh, 2018, Nhà Xuất Bản Thông Kê.
- [3] "**ReactJS thật đơn giản**" - Thùy Phương, 2023,
- [4] "**MySQL cho người mới bắt đầu**" - Nguyễn Minh Đức, 2020, Nhà Xuất Bản Đại Học Quốc Gia Hà Nội.
- [5] "**Xây dựng ứng dụng web với NodeJS + ExpressJS**" - Sơn Dương, 2023
- [6] "**Node.js Design Patterns – Second Edition**" - Mario Casciaro, Luciano Mammino, 2016, O'Reilly Media (bản dịch tiếng Việt)
- [7] "**Node.js in Action**" - Mike Cantelon, Marc Harter, Nathan Rajlich, T.J. Holowaychuk, 2017, Manning Publications (bản dịch tiếng Việt)
- [8] "**Phát triển RESTful API với Node.js**" - Sơn Dương, 2023, - <https://vntalking.com/xay-dung-ung-dung-web-voi-nodejs-expressjs.html> - Truy cập 5/2024
- [9] "**Hướng dẫn MySQL cho người mới bắt đầu**" - <https://www.hostinger.vn/huong-dan/mysql-la-gi/> - Truy cập 5/2024
- [10] "**Tài liệu học NodeJS bài bản, chuyên sâu**" - FreeC Asia, 2022 - <https://blog.freec.asia/tai-lieu-hoc-nodejs/> - Truy cập 05/2024
- [11] Tham khảo trang Web "**Bệnh viện Bình Định**" - <https://benhvienbinhdinh.com.vn/> - Truy cập 4/2024