

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG HỆ THỐNG ĐẶT LỊCH HẸN BÁC SĨ

Người hướng dẫn: **TS. LÊ VĂN MINH**
Sinh viên thực hiện: **NGUYỄN THẮNG LỢI**
Mã số sinh viên: **2151220245**
Lớp: **21CT5**

Đà Nẵng, Tháng 12 / 2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG HỆ THỐNG ĐẶT LỊCH HẸN BÁC SĨ

Người hướng dẫn: **TS. LÊ VĂN MINH**
Sinh viên thực hiện: **NGUYỄN THẮNG LỢI**
Mã số sinh viên: **2151220245**
Lớp: **21CT5**

Đà Nẵng, Tháng 12 / 2025

{Trang trắng này dùng để dán bản Nhận xét của người hướng dẫn, hoặc thay trang này bằng Nhận xét của người hướng dẫn}

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện}

TÓM TẮT

Tên đề tài: Xây dựng hệ thống đặt lịch hẹn bác sĩ.

Sinh viên thực hiện: Nguyễn Thắng Lợi

MSSV: 2151220245 Lớp 21CT5

Đề tài “Hệ Thống Đặt Lịch Hẹn Bác Sĩ sử dụng ReactJS, NodeJS và MongoDB” xây dựng một ứng dụng web hỗ trợ bệnh nhân dễ dàng tra cứu thông tin bác sĩ, chuyên khoa và đặt lịch khám trực tuyến một cách nhanh chóng, chính xác. Hệ thống được phát triển dựa trên kiến trúc MERN, trong đó ReactJS đảm nhiệm giao diện người dùng thân thiện và trực quan, NodeJS/ExpressJS xử lý các chức năng backend như xác thực tài khoản, quản lý lịch hẹn và kiểm tra trùng lịch, còn MongoDB lưu trữ và quản lý dữ liệu bác sĩ, bệnh nhân và lịch khám. Hệ thống cho phép bác sĩ theo dõi và quản lý lịch làm việc của mình, đồng thời trao quyền cho quản trị viên trong việc phê duyệt bác sĩ, quản lý người dùng và xem thống kê. Bên cạnh các chức năng chính như đặt lịch, hủy lịch, xác nhận lịch và gửi email thông báo, hệ thống còn hướng tới khả năng mở rộng như tích hợp thanh toán trực tuyến, nhắc lịch tự động và đánh giá chất lượng dịch vụ. Kết quả đề tài mang đến một giải pháp hiệu quả giúp giảm tải cho cơ sở y tế, tối ưu thời gian chờ đợi và nâng cao trải nghiệm khám chữa bệnh cho người dùng.

LỜI CẢM ƠN

Trải qua thời gian học tập và nghiên cứu các học phần chuyên ngành Công nghệ thông tin tại Trường Đại Học Kiến Trúc Đà Nẵng, nhờ có những tài liệu quý giá, sự quan tâm và chỉ bảo tận tình của các giảng viên bộ môn đã giúp chúng em hoàn thành tốt bài báo cáo tốt nghiệp.

Qua đây chúng em xin gửi lời cảm ơn chân thành và bày tỏ lòng biết ơn đến sự chỉ bảo tận tình của các thầy cô bộ môn thuộc Công nghệ thông tin. Đặc biệt là Thầy **Nguyễn Văn Minh** người đã trực tiếp hướng dẫn, chỉ bảo và giúp đỡ chúng em trong thời gian thực hiện đồ án này.

Sau cùng, em xin cảm ơn tất cả bạn bè, những người đã giúp đỡ tôi về mặt tinh thần để hoàn thành tốt chuyên đề khóa luận tốt nghiệp lần này.

Đề tài đồ án **Xây dựng hệ thống đặt lịch hẹn bác sĩ** tuy đã hoàn thành nhưng chắc hẳn còn có nhiều sai sót, rất mong nhận được ý kiến đóng góp và giúp đỡ của thầy cô và bạn bè để giúp đề tài được hoàn thành tốt nhất cho báo cáo này. Rất mong nhận được những đóng góp từ quý thầy cô, bạn bè và những người quan tâm để luận văn có thể hoàn thiện hơn.

Do điều kiện thời gian có hạn và kinh nghiệm làm báo cáo chưa có nhiều nên em không thể tránh khỏi những thiếu sót trong lúc trình bày cũng như làm báo cáo, kính mong nhận được những ý kiến đóng góp từ quý thầy cô.

Chúng em xin gửi đến Thầy và Anh/Chị lời chúc sức khỏe và lời cảm ơn chân thành!

Đà Nẵng, ngày tháng năm 2025
Sinh viên thực hiện
(ký tên và ghi rõ họ tên)

LỜI CAM ĐOAN

Tôi xin cam đoan rằng báo cáo tốt nghiệp là công trình nghiên cứu của bản thân mình. Những phần có sử dụng tài liệu tham khảo có trong báo cáo đã được liệt kê và nêu rõ ra tại phần tài liệu tham khảo. Đồng thời những số liệu hay kết quả trình bày trong báo cáo đều mang tính chất trung thực, không sao chép, đạo nhái.

Nếu như sai tôi xin chịu hoàn toàn trách nhiệm và chịu tất cả các kỷ luật của bộ môn cũng như nhà trường đề ra.

Đà Nẵng, ngày tháng.....năm 2025

*Người thực hiện
(ký tên và ghi rõ họ tên)*

MỤC LỤC

TÓM TẮT	
LỜI CẢM ƠN.....	i
LỜI CAM ĐOAN	ii
DANH MỤC HÌNH ẢNH	vi
DANH MỤC BẢNG.....	viii
DANH MỤC VIẾT TẮT	viiix
PHẦN MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI.....	2
1.1. Tên đề tài.....	2
1.2. Lý do chọn đề tài	2
1.3. Mục đích nghiên cứu.....	3
1.4. Đối tượng và phạm vi nghiên cứu	3
1.4.1. Đối tượng.....	3
1.4.2. Phạm vi nghiên cứu	4
1.5. Phương pháp nghiên cứu	4
1.6. Ý nghĩa khoa học và thực tiễn	5
1.6.1. Ý nghĩa khoa học	5
1.6.2. Ý nghĩa thực tiễn	5
1.7. Chức năng đề tài.....	6
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	8
2.1. Tìm hiểu ReactJs	8
2.1.1. React là gì ?	8
2.1.2. Các tính năng nổi bật	9
2.1.3. Cách sử dụng ReactJS trong phát triển web	10
2.2. Tìm hiểu JavaScript.....	11
2.2.1. Khái niệm cơ bản về ngôn ngữ JavaScript.....	11
2.2.2. Một số điểm nổi bật của JavaScript	12
2.3. Tìm hiểu các hook cơ bản:.....	13

2.3.1. useState	13
2.3.2. useEffect	14
2.4. Gọi API cho FrontEnd.....	15
2.4.1. Fetch API.....	16
2.4.2. Axios.....	16
2.5. Giới thiệu NodeJS	17
2.5.1. Khái niệm cơ bản về NodeJS.....	17
2.5.2. Đặc điểm NodeJS	17
2.6. Tìm hiểu về Figma	18
2.6.1. Giới thiệu về Figma.....	18
2.6.2. Ưu điểm của Figma.	19
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	23
3.1. Phân tích hệ thống.....	23
3.1.1. Các chức năng hệ thống.....	23
3.1.2. Phân tích tác nhân hệ thống.....	24
3.2. Phân tích hệ thống.....	24
3.2.1. Sơ đồ chức năng	24
3.2.2. Biểu đồ usecase tổng quan	25
3.2.3. Lược đồ usecase	25
3.2.4. Đặc tả usecase.....	32
3.2.5. Sơ đồ tuần tự	38
3.2.6. Thiết kế giao diện hệ thống	42
CHƯƠNG 4. XÂY DỰNG HỆ THỐNG ĐẶT LỊCH HẸN BÁC SĨ.....	44
4.1. Giao diện người dùng	44
4.1.1. Giao diện phần trang chủ.....	44
4.1.2. Giao diện đặt lịch khám bệnh	46
4.1.3. Giao diện nhập thông tin đặt lịch khám.....	47
4.1.4. Giao diện lịch hẹn.....	48
4.1.5. Giao diện thanh toán lịch khám online.....	49
4.1.6. Giao diện đăng ký trở thành bác sĩ.....	50
4.1.7. Giao diện thông báo xác nhận lịch hẹn qua gmail	51

4.1.8. Giao diện đăng nhập	52
4.1.9. Giao diện đăng ký.....	53
4.1.10. Giao diện hồ sơ bệnh nhân	53
4.1.11. Giao diện thay đổi thông tin bệnh nhân	54
4.1.12. Giao diện danh sách bác sĩ	55
4.2. Giao diện bác sĩ	56
4.2.1. Giao diện tổng quan bác sĩ	56
4.2.2. Giao diện lịch hẹn của bác sĩ.....	57
4.2.3. Giao diện lịch trình bác sĩ	57
4.2.4. Giao diện hồ sơ bác sĩ	58
4.2.5. Giao diện đăng nhập bác sĩ	59
4.2.6. Giao diện thay đổi mật khẩu bác sĩ	59
4.2.7. Giao diện thông tin bệnh nền gửi cho bác sĩ	60
4.3. Giao diện admin	60
4.3.1. Giao diện đăng nhập Admin.....	60
4.3.2. Giao diện tổng quan Admin	61
4.3.3. Giao diện lịch hẹn Admin	61
4.3.4. Giao diện thêm mới bác sĩ.....	62
4.3.5. Giao diện quản lý bác sĩ	62
4.3.6. Giao diện quản lý thông tin bệnh nhân	63
4.3.7. Giao diện quản lý từ chối đăng ký trở thành bác sĩ.....	63
4.3.8. Giao diện chấp nhận đăng ký trở thành bác sĩ và gửi qua gmail	64
4.3.9. Giao diện thông báo từ chối đăng ký làm bác sĩ	64
4.3.10. Giao diện admin chỉnh sửa thông tin bác sĩ	65
4.3.11. Giao diện admin chỉnh sửa thông tin bệnh nhân	66
4.3.12. Giao diện hồ sơ đăng ký bác sĩ.....	66
KẾT LUẬN.....	67
TÀI LIỆU THAM KHẢO	69

DANH MỤC HÌNH ẢNH

Hình 2.1: ReactJS	10
Hình 2.2: React component	11
Hình 2.3: Axios.....	16
Hình 2.4: Giới thiệu về Figma.....	19
Hình 2.5: Figma hỗ trợ cả máy tính bàn và thiết bị di động.....	20
Hình 2.6: Chia sẻ tệp nhanh chóng trên Figma	21
Hình 2.7: Figma giúp dễ dàng cộng tác với những người khác	22
Hình 3.1: Sơ đồ tổng quát.....	25
Hình 3.2: Biểu đồ usecase tổng quan	25
Hình 3.3: Lược đồ usecase khách hàng.....	26
Hình 3.4: Lược đồ usecase bệnh nhân.....	27
Hình 3.5: Usecase xem thông tin khám bệnh	27
Hình 3.6: Usecase đặt lịch hẹn khám	28
Hình 3.7: Usecase quản lý tài khoản cá nhân.....	28
Hình 3.8: Lược đồ usecase bác sĩ.....	29
Hình 3.9: Usecase quản lý lịch hẹn khám bệnh	30
Hình 3.10: Usecase quản lý thông tin khám bệnh cho bệnh nhân.....	30
Hình 3.11: Lược đồ usecase Admin	31
Hình 3.12: Usecase quản lý tài khoản hệ thống	31
Hình 3.13: Biểu đồ tuần tự đăng ký	39
Hình 3.14: Biểu đồ tuần tự đăng nhập.....	39
Hình 3.15: Biểu đồ tuần tự đặt lịch khám bệnh.....	40
Hình 3.16: Biểu đồ tuần tự thêm bác sĩ.....	40
Hình 3.17: Biểu đồ xóa bác sĩ	41

Hình 3.18: Biểu đồ tuần tự sửa bác sĩ.....	42
Hình 3.19: Biểu đồ tuần tự thêm lịch khám	42
Hình 3.20: Hình ảnh vẽ giao diện trên Figma	43
Hình 4.1: Giao diện trang chủ DoctorGo	46
Hình 4.2: Giao diện đặt lịch khám bệnh	47
Hình 4.3: Giao diện nhập thông tin lịch khám.....	48
Hình 4.4: Giao diện lịch hẹn.....	49
Hình 4.5: Giao diện thanh toán lịch hẹn.....	50
Hình 4.6: Giao diện đăng ký trở thành bác sĩ	51
Hình 4.7: Giao diện thông báo xác nhận lịch hẹn qua gmail.....	52
Hình 4.8: Giao diện đăng nhập	52
Hình 4.9: Giao diện đăng ký.....	53
Hình 4.10: Giao diện hồ sơ bệnh nhân	54
Hình 4.11: Giao diện thay đổi thông tin bệnh nhân	55
Hình 4.12: Giao diện danh sách bác sĩ	56
Hình 4.13: Giao diện tổng quan bác sĩ	57
Hình 4.14: Giao diện lịch hẹn của bác sĩ.....	57
Hình 4.15: Giao diện lịch trình bác sĩ.....	58
Hình 4.16: Giao diện hồ sơ bác sĩ.....	59
Hình 4.17: Giao diện đăng nhập bác sĩ.....	59
Hình 4.18: Giao diện thay đổi mật khẩu bác sĩ	60
Hình 4.19: Giao diện thông tin bệnh nền gửi cho bác sĩ	60
Hình 4.20: Giao diện đăng nhập Admin.....	61
Hình 4.21: Giao diện tổng quan Admin	61
Hình 4.22: Giao diện lịch hẹn Admin	62
Hình 4.23: Giao diện thêm mới bác sĩ.....	62

Hình 4.24: Giao diện quản lý bác sĩ	63
Hình 4.25: Giao diện quản lý thông tin bệnh nhân.....	63
Hình 4.26: Giao diện quản lý từ chối đăng ký trở thành bác sĩ.....	64
Hình 4.27: Giao diện chấp nhận đăng ký trở thành bác sĩ và gửi qua gmail	64
Hình 4.28: Giao diện thông báo từ chối đăng ký làm bác sĩ	65
Hình 4.29: Giao diện admin chỉnh sửa thông tin bác sĩ	65
Hình 4.30: Giao diện admin chỉnh sửa thông tin bệnh nhân	66
Hình 4.31: Giao diện hồ sơ đăng ký bác sĩ.....	66

DANH MỤC BẢNG

Bảng 3.1: Usecase lọc.....	32
Bảng 3.2: Usecase xem chi tiết bác sĩ	33
Bảng 3.3: Use case đặt lịch khám.....	33
Bảng 3.4: Usecase xác nhận lịch khám bệnh thông báo gmail	34
Bảng 3.5: Usecase đăng nhập	34
Bảng 3.6: Usecase đăng ký.....	35
Bảng 3.7: Usecase đăng nhập admin.....	35
Bảng 3.8:Usecase đăng ký tài khoản bác sĩ.....	36
Bảng 3.9: Usecase sửa thông tin bác sĩ	36
Bảng 3.10: Usecase xóa tài khoản bác sĩ.....	37
Bảng 3.11: Usecase thêm bác sĩ	37
Bảng 3.12: Usecase sửa thông tin chi tiết bệnh nhân	38
Bảng 3.13: Usecase thêm lịch khám.....	38

DANH MỤC VIẾT TẮT

STT	Kí hiệu viết tắt	Chữ viết tắt đầy đủ
1	YOLO	You Only Look Once
2	DOM	Document Object Model
3	SPA	Single Page Application
4	CRUD	Create, Read, Update, Delete
5	API	Application Programming Interface
6	NPM	Node Package Manager
7	CLI	Command Line Interface
8	ESLint	ECMAScript Linting Utility
9	JSX	JavaScript Syntax Extension
10	UI	User Interface

PHẦN MỞ ĐẦU

Trong thời đại ngày nay, việc chuyển đổi từ mô hình truyền thống sang mô hình số hóa là một xu hướng không thể phủ nhận trong xã hội. Đối mặt với thách thức này, việc áp dụng hệ thống đặt lịch khám trực tuyến không chỉ là sự tận dụng hiệu quả xu hướng này mà còn là một bước tiến quan trọng, mang lại không ít những ưu điểm và sự thuận tiện cho người dùng.

Đầu tiên, việc sử dụng ứng dụng đặt lịch khám chữa bệnh trực tuyến không chỉ giúp giảm thiểu thời gian chờ đợi đối với bệnh nhân mà còn giảm bớt gánh nặng công đoạn giấy tờ tại các cơ sở y tế. Những tiện ích này không chỉ tạo điều kiện thuận lợi cho bệnh nhân mà còn hỗ trợ tối ưu hóa nguồn lực quý báu trong hệ thống y tế ngày nay.

Thứ hai, bệnh nhân có khả năng đặt lịch khám bất cứ lúc nào, ở bất kỳ đâu thông qua trang web của ứng dụng. Sự thuận tiện này không chỉ giúp họ tiết kiệm thời gian và công sức so với việc phải di chuyển đến phòng khám trực tiếp hoặc thậm chí là cuộc gọi điện thoại, mà còn mang lại sự linh hoạt cho lịch trình hàng ngày của họ.

Bên cạnh đó, khả năng tra cứu thông tin về bác sĩ, các dịch vụ khám, và lịch sử khám bệnh của bệnh nhân trở nên dễ dàng hơn bao giờ hết. Việc này không chỉ giúp bệnh nhân tự quản lý sức khỏe của mình một cách hiệu quả mà còn tạo ra một môi trường thông tin tổng quan về bản thân.

Cuối cùng, việc đặt lịch trực tuyến mở ra khả năng cho bệnh nhân lựa chọn bác sĩ, chuyên khoa, và thời gian khám theo mong muốn cá nhân mà không cần phải đến trực tiếp phòng khám. Điều này tạo ra sự thoải mái và quyền lực trong quyết định chăm sóc sức khỏe cá nhân.

Bên cạnh đó dưới sự hướng dẫn tận tình thầy Nguyễn Văn Minh em đã hoàn thành bài báo cáo đồ án này. Dù đã cố gắng hết sức trong việc nghiên cứu, phân tích, và triển khai hệ thống, em nhận thức rằng không tránh khỏi những thiếu sót.

Mong rằng, sự thông cảm và góp ý từ quý anh chị cùng Thầy cô sẽ giúp em hoàn thiện hơn. Em chân thành cảm ơn sự quan tâm và hỗ trợ

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

Chương 1 giới thiệu chung về các vấn đề liên quan đến đề tài, lý do chọn đề tài, mục tiêu và phương pháp nghiên cứu, xác định các vấn đề chính cần thực hiện để hoàn thành đề án

1.1. Tên đề tài

Đề tài: “XÂY DỰNG HỆ THỐNG ĐẶT LỊCH HẸN BÁC SĨ”

1.2. Lý do chọn đề tài

Trong hành trình đưa người thân đến các phòng khám, em không thể không chú ý đến những thách thức và phiền toái mà họ phải đối mặt. Việc phải xếp hàng và bốc số thứ tự, không chỉ là một thách thức về thời gian mà còn tạo ra những trải nghiệm không mong muốn khi phải đối mặt với bác sĩ mà họ không chọn lựa được. Thêm vào đó, tình trạng phòng khám đóng cửa không theo đúng giờ là một vấn đề khác tạo ra sự phiền toái không đáng có.

Những hiểu biết này đã làm cho em quyết định đưa ra giải pháp hiện đại và tận dụng sức mạnh của công nghệ thông tin để giải quyết những thách thức này. "Đặt Lịch Khám Chữa Bệnh " không chỉ là một ứng dụng website, mà là một cầu nối mang đến trải nghiệm khám chữa bệnh tận tâm.

Qua việc xây dựng ứng dụng này, mục tiêu của em không chỉ là đơn giản là giúp bệnh nhân dễ dàng đặt lịch hẹn trực tuyến mọi lúc, mọi nơi. Mà còn là tạo ra một không gian y tế đồng bộ, nơi mà sự thoải mái và sự quan tâm đến sức khỏe của người bệnh được đặt lên hàng đầu.

Bằng cách này, em hy vọng ứng dụng sẽ không chỉ giảm bớt gánh nặng thời gian và công sức cho những người đang cần chăm sóc y tế mà còn mang lại sự yên tâm và tin tưởng khi họ biết rằng họ có thể dễ dàng đặt lịch và nhận được sự chăm sóc tận tâm từ các chuyên gia y tế. Đây không chỉ là một ứng dụng, mà là một cơ hội để tạo ra

những thay đổi tích cực và tạo nên sự an bình trong việc quản lý sức khỏe.

1.3. Mục đích nghiên cứu

Thuận lợi và tiết kiệm thời gian: Việc đặt lịch trực tuyến giúp bệnh nhân tiết kiệm thời gian và năng lượng so với việc phải đến trực tiếp tại phòng khám. Họ có thể dễ dàng chọn thời gian phù hợp mà không cần phải xếp hàng chờ đợi.

Quản lý lịch hẹn hiệu quả: Hệ thống đặt lịch trực tuyến giúp tối ưu hóa lịch trình của các bác sĩ và phòng mạch, giảm nguy cơ chồng chéo lịch và tăng cường hiệu suất làm việc.

Tính bảo mật và riêng tư: Việc tích hợp với hệ thống quản lý bệnh án điện tử đảm bảo an toàn và bảo mật thông tin y tế cá nhân. Điều này giúp bảo vệ quyền riêng tư của bệnh nhân.

Tối ưu hóa trải nghiệm người dùng: Tạo ra một giao diện người dùng thân thiện và dễ sử dụng, giúp người bệnh dễ dàng đặt lịch, tra cứu thông tin và tương tác với hệ thống một cách thuận tiện nhất.

Tăng cường hiệu suất lịch hẹn: Nghiên cứu và triển khai giải pháp để tối ưu hóa quy trình đặt lịch, giảm thời gian chờ đợi và đảm bảo tính chính xác và hiệu quả của lịch hẹn.

Mục tiêu nghiên cứu này nhằm đảm bảo rằng ứng dụng sẽ không chỉ đáp ứng nhu cầu thực tế của cộng đồng y tế mà còn mang lại lợi ích to lớn về hiệu suất, tiện ích, và chất lượng dịch vụ.

Tóm lại, cơ sở lý thuyết cho đề tài này không chỉ bám sát vào sự thuận tiện trong quá trình đặt lịch khám mà còn chú trọng đến khía cạnh chăm sóc sức khỏe toàn diện và hiệu quả sử dụng công nghệ trong lĩnh vực y tế.

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng

Người Bệnh: Những người có nhu cầu sử dụng dịch vụ y tế và đặt lịch khám trực tuyến.

Người Thân: Người thân đưa người bệnh đến các cơ sở y tế.

Bác sĩ: Các chuyên gia y tế tham gia quá trình khám và điều trị bệnh nhân. Nhân viên y tế: Nhân viên quản lý lịch hẹn và hồ sơ bệnh nhân.

Phòng mạch và bệnh viện: Những phòng mạch hoặc bệnh viện có quan tâm đến cải thiện hiệu suất và trải nghiệm của bệnh nhân.

1.4.2. Phạm vi nghiên cứu

Vị trí: Nghiên cứu có thể tập trung vào một khu vực cụ thể hoặc có thể mở rộng phạm vi để bao gồm nhiều địa điểm y tế.

Loại hình y tế: Các phòng mạch tư nhân, bệnh viện công lập, hoặc cả hai để đảm bảo tính đại diện và áp dụng

Dịch vụ y tế: Nghiên cứu có thể tập trung vào các chuyên khoa y tế cụ thể như nội khoa, nhi khoa, hoặc nhiều chuyên ngành y tế khác nhau.

Quy mô cơ sở y tế: Phạm vi có thể bao gồm cả các phòng mạch y tế nhỏ, cũng như các bệnh viện lớn với nhiều chuyên khoa. Nền tảng công nghệ: ứng dụng có thể phát triển cho nền tảng web, di động hoặc cả hai để đáp ứng đối tượng sử dụng rộng rãi.

Phân khúc người dùng: Các đối tượng sử dụng có thể bao gồm cả người dùng thông thường và nhóm đặc biệt như người cao tuổi, người khuyết tật, hoặc nhóm đa dạng văn hóa.

Thời gian nghiên cứu: Nghiên cứu có thể thực hiện trong khoảng thời gian cố định để đánh giá hiệu suất và thay đổi theo thời gian.

Mục tiêu của việc xác định đối tượng và phạm vi nghiên cứu là đảm bảo rằng kết quả thu được có thể áp dụng rộng rãi và mang lại lợi ích cho nhiều bên liên quan trong lĩnh vực y tế.

1.5. Phương pháp nghiên cứu

Nghiên cứu theo tình hình hiện tại: Tiến hành phân tích tình hình hiện tại của quy trình đặt lịch khám và chăm sóc sức khỏe ở các cơ sở y tế có sẵn. Đánh giá những thách thức và vấn đề mà người bệnh và nhân viên y tế đang phải đối mặt.

Phân tích nhu cầu người dùng: Thực hiện khảo sát và phỏng vấn người bệnh, người thân, bác sĩ, và nhân viên y tế để hiểu rõ nhu cầu và mong muốn của họ đối với một ứng dụng đặt lịch trực tuyến.

Phát triển ứng dụng: Sử dụng các công nghệ phù hợp để phát triển ứng dụng, bao gồm cả việc kết nối với hệ thống quản lý bệnh án điện tử và tích hợp trí tuệ nhân tạo nếu cần.

Kiểm thử và tối ưu hóa: Thực hiện các bước kiểm thử chất lượng để đảm bảo ứng dụng hoạt động đúng đắn và đáp ứng tốt với các yêu cầu đã đề ra. Tiến hành các phiên tối ưu hóa dựa trên phản hồi từ người dùng và nhóm người dùng thử nghiệm.

Triển khai và đánh giá: Triển khai ứng dụng cho một số nhỏ người dùng để theo dõi hiệu suất thực tế và thu thập ý kiến phản hồi. Đánh giá hiệu suất dựa trên các chỉ số như thời gian đặt lịch, tần suất sử dụng, và độ hài lòng của người dùng.

Đánh giá hiệu quả: Đánh giá hiệu suất dựa trên các chỉ số như thời gian đặt lịch, tần suất sử dụng, và độ hài lòng của người dùng.

Phân tích dữ liệu và báo cáo: Phân tích dữ liệu thu thập được để đưa ra các kết luận về hiệu suất và đề xuất cải tiến. Soạn thảo báo cáo nghiên cứu và trình bày các kết quả và phát hiện.

Tóm lại: Phương pháp nghiên cứu này cần tích hợp sự tương tác chặt chẽ với người sử dụng và các bên liên quan, đồng thời linh hoạt để điều chỉnh dự án dựa trên các phản hồi thực tế và thay đổi trong quy trình nghiên cứu.

1.6. Ý nghĩa khoa học và thực tiễn

1.6.1. Ý nghĩa khoa học

Nâng cao kiến thức y tế: Nghiên cứu có thể mang lại cái nhìn sâu sắc về cách ứng dụng công nghệ có thể cải thiện quy trình chăm sóc sức khỏe và trải nghiệm người dùng.

Phát triển công nghệ y tế: Tạo cơ hội để phát triển và thử nghiệm các công nghệ mới như trí tuệ nhân tạo và hệ thống đặt lịch thông minh trong lĩnh vực y tế.

Xây dựng cơ sở lý thuyết: Góp phần vào việc xây dựng cơ sở lý thuyết về việc tích hợp công nghệ trong quy trình chăm sóc sức khỏe và quản lý lịch hẹn.

1.6.2. Ý nghĩa thực tiễn

Tăng cường trải nghiệm người dùng: Cải thiện trải nghiệm người dùng khi đặt lịch và sử dụng dịch vụ y tế, giúp họ tiết kiệm thời gian và tăng tính thuận tiện.

Giảm áp lực cho hệ thống y tế: Giảm áp lực cho cơ sở y tế bằng cách tự động hóa quy trình đặt lịch và giảm tình trạng bỏ lỡ.

Tăng hiệu quả quản lý lịch hẹn: Cải thiện quản lý lịch hẹn, giảm thời gian chờ đợi và tối ưu hóa sử dụng nguồn lực y tế.

Nâng cao chất lượng dịch vụ y tế: Tăng cường khả năng theo dõi và quản lý thông tin y tế, giúp cải thiện chất lượng dịch vụ và tăng sự chăm sóc đồng bộ.

Thúc đẩy phát triển cộng đồng y tế: Tạo cơ hội cho các cơ sở y tế tham gia vào sự đổi mới và phát triển trong lĩnh vực y tế.

Tạo lập mô hình cho các dự án tương tự: Cung cấp một mô hình hoạt động cho các dự án phát triển ứng dụng y tế tương tự, có thể mở rộng và tùy chỉnh cho các cơ sở y tế khác nhau.

Đánh bại thách thức y tế xã hội: Hỗ trợ giải quyết các vấn đề như thời gian chờ đợi, quản lý lịch hẹn, và tính tiện lợi trong quá trình sử dụng dịch vụ y tế.

Tóm lại, đề tài này không chỉ mang lại giá trị khoa học mà còn tạo ra những cải tiến thực tế và hữu ích trong lĩnh vực y tế, đồng thời thúc đẩy sự phát triển của cộng đồng y tế và công nghệ thông tin.

1.7. Chức năng đề tài

Chức năng của đề tài "Hệ Thống Đặt Lịch Hẹn Bác Sĩ" là tạo ra một hệ thống ứng dụng web để giải quyết những thách thức và nhu cầu cụ thể trong quá trình đặt lịch khám và chăm sóc sức khỏe. Dưới đây là một số chức năng quan trọng của đề tài:

Đặt lịch trực tuyến: Cho phép người bệnh đặt lịch khám chữa bệnh trực tuyến thông qua giao diện dễ sử dụng.

Tra cứu thông tin bác sĩ: Cung cấp thông tin chi tiết về các bác sĩ, bao gồm chuyên môn, kinh nghiệm, và lịch làm việc.

Thanh toán trực tuyến: Hỗ trợ thanh toán trực tuyến thông qua cổng thanh toán phổ biến, giúp rút ngắn quy trình tiếp nhận và nâng cao tính tiện lợi cho người bệnh.

Đăng ký trở thành bác sĩ: Cho phép các bác sĩ gửi hồ sơ đăng ký tham gia hệ thống giúp hệ thống đảm bảo chất lượng bác sĩ tham gia và vận hành an toàn.

Quản lý lịch hẹn: Tích hợp hệ thống quản lý lịch hẹn để giảm thiểu thời gian chờ đợi và tăng cường hiệu suất.

Gửi thông báo qua email: Gửi thông báo và nhắc nhở đến người bệnh, chức

năng của đề tài này nhằm mục đích cải thiện trải nghiệm người dùng, tối ưu hóa quy trình làm việc của cơ sở y tế và đóng góp vào sự hiện đại hóa trong lĩnh vực y tế thông qua ứng dụng công nghệ.

Các chức năng trên nhằm cải thiện trải nghiệm người bệnh, tối ưu hóa quy trình làm việc của bác sĩ, giảm tải thủ tục truyền thống, tăng tính hiện đại và tự động hóa trong lĩnh vực y tế và tạo môi trường tương tác hiệu quả giữa bệnh nhân và bác sĩ.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này trình bày nội dung lý thuyết về lập trình ReactJs, Nodejs. Các thành phần cơ bản của một dự án React component.

2.1. Tìm hiểu ReactJs

2.1.1. React là gì ?

React là một thư viện JavaScript declarative, hiệu quả và linh hoạt cho việc xây dựng giao diện người dùng. React cho phép bạn tạo những giao diện (UI) phức tạp từ những đoạn code nhỏ và độc lập. Những đoạn code này được gọi là “components”.

Lịch sử hình thành

ReactJS là một thư viện JavaScript được phát triển bởi Facebook và được giới thiệu công khai lần đầu tiên vào năm 2013. Dưới đây là một tóm tắt về lịch sử hình thành của ReactJS:

Nhu Cầu Tăng Cao (2007-2011): Trước khi React xuất hiện, Facebook đã đối mặt với một số vấn đề về hiệu suất trên giao diện người dùng của mình. Sự gia tăng đột ngột về lượng người sử dụng và các tính năng phức tạp đặt ra thách thức cho việc quản lý và duy trì mã nguồn.

Tạo Ra React (2011): Jordan Walke, một nhà phát triển tại Facebook, đã bắt đầu phát triển một thư viện JavaScript mới để giải quyết những vấn đề hiệu suất mà Facebook đang gặp phải. Walke tạo ra React với mục tiêu chính là tối ưu hóa quá trình render UI trên trình duyệt.

Giới Thiệu React (2013): React được giới thiệu công khai tại một hội nghị Facebook F8 vào tháng 5 năm 2013. Nó không chỉ giải quyết vấn đề hiệu suất mà còn đưa ra mô hình lập trình mới và linh hoạt

Độ Mở và Sự Thụ Động (2013-2014): React được công bố với giấy phép mã nguồn mở (Open Source) và ngay lập tức thu hút sự quan tâm của cộng đồng lập trình. Cộng đồng React nhanh chóng phát triển và đóng góp vào sự phát triển của thư viện.

React Native (2015): React Native, một framework cho việc phát triển ứng

dụng di động sử dụng React, được giới thiệu. React Native cho phép việc chia sẻ mã nguồn giữa ứng dụng di động iOS và Android.

Phổ Biến Trong Cộng Đồng (2016-2018): React trở nên ngày càng phổ biến trong cộng đồng phát triển web. Công ty và tổ chức nhiều ngành công nghiệp bắt đầu sử dụng React để xây dựng ứng dụng web chất lượng cao.

Hooks và Phiên Bản React 16 (2018): React 16 được giới thiệu với nhiều cải tiến, trong đó có Hooks - một cách mới để sử dụng state và lifecycle trong functional components. Hooks mang lại sự linh hoạt và tái sử dụng mã nguồn cao hơn.

Cộng Đồng Phát Triển Mạnh Mẽ (2019-nay): React tiếp tục được phát triển và cập nhật thường xuyên với sự đóng góp của cộng đồng. Các tính năng và cải tiến mới được thêm vào để duy trì và nâng cao chất lượng phát triển ứng dụng web.

2.1.2. Các tính năng nổi bật

ReactJS có nhiều tính năng hữu ích cho việc phát triển ứng dụng web, bao gồm:

Components: ReactJS cho phép phát triển ứng dụng web theo mô hình component. Các component là các phần tử UI độc lập có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.

Virtual DOM: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM được lưu trữ trong bộ nhớ và được cập nhật một cách nhanh chóng khi có thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.

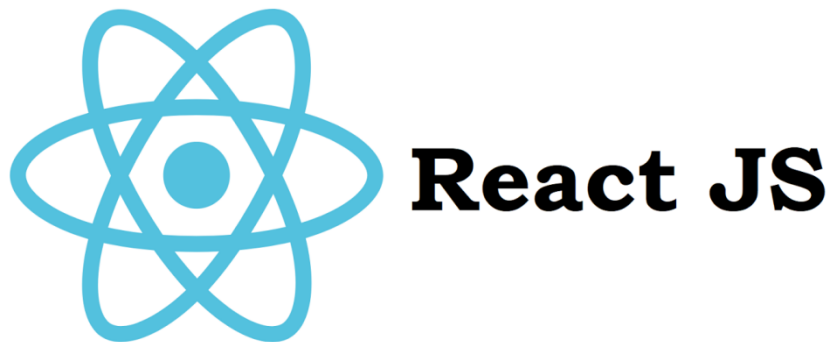
JSX: JSX là một ngôn ngữ lập trình phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI. JSX kết hợp HTML và JavaScript, giúp cho việc viết mã dễ hiểu và dễ bảo trì hơn.

State và Props: ReactJS cho phép quản lý trạng thái của các thành phần UI thông qua State và Props. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.

Hỗ trợ tốt cho SEO: ReactJS hỗ trợ tốt cho việc tối ưu hóa SEO. Với các thư viện như React Helmet, các nhà phát triển có thể quản lý các phần tử meta và title cho từng trang web, giúp tăng khả năng tìm kiếm và tăng cường trải nghiệm người dùng.

Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

Redux: Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.



Hình 2.1: ReactJS

2.1.3. Cách sử dụng ReactJS trong phát triển web

Để sử dụng ReactJS trong phát triển web, bạn có thể làm theo các bước sau:

Bước 1 - Cài đặt Node.js và npm: ReactJS được xây dựng trên nền tảng Node.js, do đó bạn cần cài đặt Node.js và npm để phát triển ứng dụng ReactJS.

Bước 2 - Tạo một ứng dụng React: Bạn có thể tạo một ứng dụng React bằng cách sử dụng lệnh "create-react-app" trong Command Prompt hoặc Terminal.

Bước 3 - Tạo các component: Tạo các component để xây dựng giao diện người dùng cho ứng dụng của bạn. Bạn có thể tạo component bằng cách sử dụng class hoặc hàm.

Bước 4 - Xây dựng giao diện người dùng: Sử dụng JSX để xây dựng giao diện người dùng cho ứng dụng của bạn. JSX là một ngôn ngữ phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI.

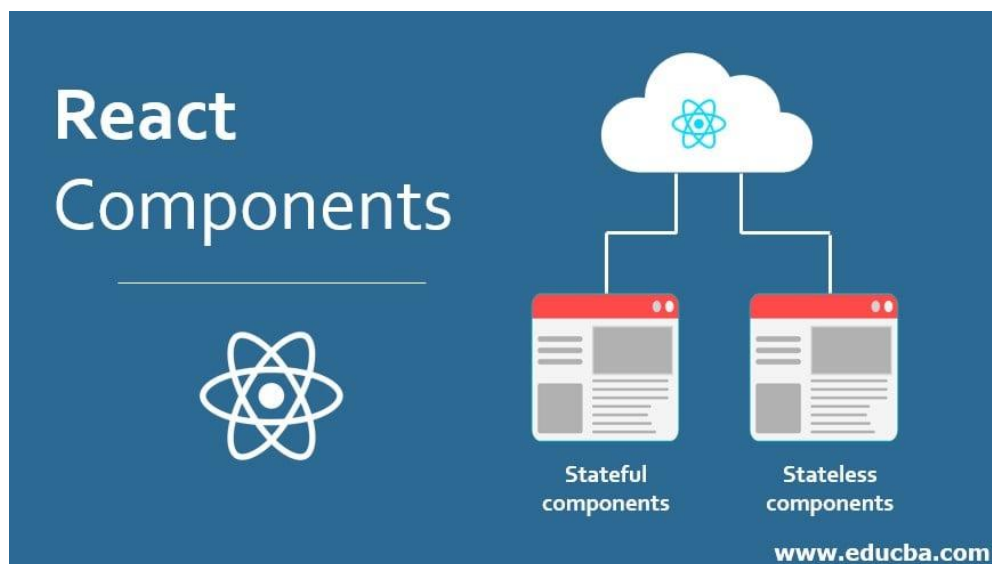
Bước 5 - Quản lý trạng thái: Sử dụng State và Props để quản lý trạng thái của các thành phần UI. State là trạng thái của một thành phần được quản lý bởi nó chính,

trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.

Bước 6 - Kết nối với API: Sử dụng thư viện như Axios để kết nối với API và lấy dữ liệu từ server.

Bước 7 - Build và triển khai ứng dụng: Sử dụng lệnh "npm run build" để build ứng dụng của bạn và triển khai nó trên môi trường sản phẩm.

Tóm lại, để sử dụng ReactJS trong phát triển web, bạn cần cài đặt Node.js và npm, tạo các component, xây dựng giao diện người dùng, quản lý trạng thái, kết nối với API, build và triển khai ứng dụng.



Hình 2.2: React component

2.2. Tìm hiểu JavaScript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

2.2.1. Khái niệm cơ bản về ngôn ngữ JavaScript

Trước đây, các trang web thường có định dạng tĩnh, tương tự như các trang trong

một cuốn sách. Các trang tĩnh thường chỉ hiển thị thông tin theo một bố cục. JavaScript đã xuất hiện như một công nghệ phía trình duyệt, giúp làm cho các ứng dụng web trở nên linh hoạt hơn. Bằng cách sử dụng JavaScript, trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web.

Với sự phát triển và hoàn thiện của ngôn ngữ này, các nhà phát triển JavaScript đã xây dựng các thư viện, frameworks và phong cách lập trình mới. Điều này đã mở ra khả năng sử dụng JavaScript không chỉ trong phạm vi trình duyệt web mà còn trên nhiều nền tảng khác, kể cả phía máy chủ.

Ngày nay, JavaScript không chỉ được sử dụng để tạo ra các ứng dụng web mà còn mở rộng hoạt động phát triển ở cả phía máy khách (client-side) và máy chủ (server-side). Dưới đây là một số trường hợp sử dụng phổ biến của ngôn ngữ này:

Tương Tác Người Dùng: JavaScript giúp tạo ra trải nghiệm người dùng tương tác trực tiếp và đáp ứng trên trình duyệt.

Thay Đổi Động Trên Trang Web: Sử dụng JavaScript, trang web có thể thay đổi động bố cục, nội dung mà không cần phải tải lại trang.

Phát Triển Ứng Dụng Web: JavaScript là cơ sở cho nhiều frameworks và thư viện mạnh mẽ như React, Angular, và Vue.js để phát triển ứng dụng web hiện đại.

Phát Triển Máy Chủ: Với sự xuất hiện của Node.js, JavaScript có thể được sử dụng để phát triển ứng dụng máy chủ, mở rộng sức mạnh của ngôn ngữ này ngoài phạm vi trình duyệt.

JavaScript, với sự linh hoạt và tính tương tác, đóng vai trò quan trọng trong cả ngành phát triển web và ứng dụng di động hiện đại.

2.2.2. Một số điểm nổi bật của JavaScript

JavaScript là một ngôn ngữ lập trình rất phổ biến với nhiều điểm nổi bật đặc sắc. Dưới đây là một số điểm nổi bật của JavaScript:

Ngôn Ngữ Client-Side: JavaScript thường được thực thi trực tiếp trên trình duyệt của người dùng, giúp tạo ra trải nghiệm tương tác và động lực trên phía client.

Dễ Học và Sử Dụng: Cú pháp của JavaScript rất giống với các ngôn ngữ lập trình khác như Java và C, điều này giúp lập trình viên dễ học và làm việc với JavaScript.

Đa Mô Hình và Lỏng Lẻo: JavaScript là một ngôn ngữ đa mô hình, hỗ trợ cả lập trình hướng đối tượng (OOP) và lập trình hàm. Loại ngôn ngữ lỏng lẻo (loosely typed) giúp linh hoạt khi làm việc với kiểu dữ liệu.

Sự Phổ Biến và Cộng Đồng Lớn: JavaScript được sử dụng rộng rãi và có một cộng đồng lập trình viên lớn. Điều này có nghĩa là có nhiều tài nguyên, thư viện và frameworks hỗ trợ JavaScript.

EcmaScript và Tiêu Chuẩn Hóa: JavaScript dựa trên tiêu chuẩn EcmaScript, giúp đảm bảo tính tương thích và khả năng chuyển đổi giữa các môi trường thực thi.

Async Programming và Promise: JavaScript hỗ trợ lập trình không đồng bộ thông qua cơ chế callback và Promise, giúp xử lý công việc không đồng bộ một cách hiệu quả.

Frameworks và Thư Viện Phong Phú: Có nhiều frameworks và thư viện mạnh mẽ được xây dựng trên nền JavaScript như React, Angular, Vue.js, Express, Node.js, giúp tăng cường khả năng phát triển ứng dụng.

Phát Triển Ứng Dụng Web và Mobile: JavaScript không chỉ được sử dụng trong phát triển web mà còn trong phát triển ứng dụng di động thông qua frameworks như React Native.

Dự Án Open Source và Phiên Bản Mới: JavaScript có nhiều dự án mã nguồn mở và nhận được sự phát triển liên tục, với các phiên bản mới của ECMAScript thường xuyên ra mắt với các tính năng và cải tiến.

Khả Năng Tương Tác Với DOM: JavaScript có thể tương tác với Document Object Model (DOM) để thay đổi cấu trúc và nội dung của trang web, làm cho nó trở thành một công cụ mạnh mẽ cho phát triển web tương tác.

Những điểm nổi bật này giúp JavaScript trở thành một trong những ngôn ngữ lập trình quan trọng và phổ biến nhất trong ngành phát triển web và ứng dụng di động.

2.3. Tìm hiểu các hook cơ bản:

2.3.1. *useState*

Hook `useState` là một trong những hook cơ bản nhất trong React và cho phép bạn thêm trạng thái vào các component chức năng (functional component). Nó cung cấp cách đơn giản để lưu trữ và cập nhật các giá trị trạng thái trong component.

Để sử dụng `useState`, bạn cần import nó từ thư viện React như sau:

```
import React, { useState } from 'react';
```

Sau đó, bạn có thể sử dụng `useState` trong component của mình:

```
const MyComponent = () => {  
  // Sử dụng useState để khởi tạo một biến trạng thái  
  const [count, setCount] = useState(0);  
  // ...  
};
```

Trạng thái (state) trong `useState` là một hook trong React cho phép bạn thêm trạng thái vào các functional component. Nó cung cấp một biến trạng thái và một hàm để cập nhật giá trị trạng thái đó. Khi giá trị trạng thái thay đổi, component sẽ render lại để phản ánh các thay đổi đó.

2.3.2. *useEffect*

Được sử dụng để thực hiện các side effect trong functional component. Side effect có thể là các tác động phụ như gọi API, đăng ký sự kiện, hoặc cập nhật DOM. `useEffect` sẽ được gọi sau mỗi lần component được render hoặc khi các dependencies thay đổi.

Để tránh việc hàm `useEffect` luôn chạy mỗi khi có thay đổi State thì ta sẽ truyền những giá trị vào tham số thứ hai (mảng array), khi những giá trị này thay đổi thì hàm `useEffect` mới được thực thi. Hoặc truyền vào một array rỗng thì nó chỉ chạy một lần đầu tiên khi render.

Để sử dụng `useEffect`, bạn cần import nó từ thư viện React như sau:

```
import React, { useEffect } from 'react';
```

Sau đó, bạn có thể sử dụng `useEffect` trong component của mình:

```
const MyComponent = () => {  
  useEffect(() => {  
    // Các tác vụ hiệu lực sẽ được thực thi ở đây  
    console.log('Component đã được render');  
  
    // Trả về một hàm để dọn dẹp (cleanup) khi component bị hủy  
    return () => {  
      console.log('Component bị hủy');  
    };  
  }, []);  
  // ...  
};
```

Trong ví dụ trên, chúng ta đã sử dụng `useEffect` để thực hiện hai tác vụ. Đầu tiên, trong hàm callback của `useEffect`, chúng ta đã in ra một thông báo cho biết component đã được render. Đồng thời, chúng ta đã trả về một hàm để dọn dẹp (cleanup) khi component bị hủy. Trong trường hợp này, khi component bị hủy, thông báo "Component bị hủy" sẽ được in ra.

Tham số thứ hai của `useEffect` là một mảng dependencies, xác định các giá trị mà `useEffect` sẽ theo dõi. Nếu dependencies không được cung cấp hoặc được truyền vào một mảng rỗng [], `useEffect` chỉ được gọi một lần sau khi component được render lần đầu tiên. Nếu dependencies được cung cấp, `useEffect` sẽ được gọi lại mỗi khi giá trị trong dependencies thay đổi.

`useEffect` giúp bạn thực hiện các tác vụ hiệu lực trong React. Bằng cách sử dụng `useEffect`, bạn có thể tương tác với DOM, gửi yêu cầu HTTP, đăng ký sự kiện, và thực hiện các tác vụ khác không trực tiếp liên quan đến việc render giao diện.

2.4. Gọi API cho FrontEnd

2.4.1. Fetch API

Cho phép gửi và nhận dữ liệu qua mạng. Cung cấp một giao diện đơn giản để thực hiện yêu cầu HTTP như GET, POST, PUT, DELETE và nhiều loại yêu cầu khác từ phía client.

Cách sử dụng: gọi hàm ‘`fetch()`’ và truyền url của tài nguyên muốn truy xuất.

Hàm này trả về một Promise, cho phép xử lý dữ liệu nhận về từ server.

2.4.2. Axios

➤ Khái niệm

Axios là một thư viện HTTP client được sử dụng trong JavaScript để gửi yêu cầu HTTP từ một trình duyệt hoặc một môi trường Node.js. Nó cung cấp một cách dễ dàng và linh hoạt để tương tác với các API HTTP và xử lý các phản hồi từ máy chủ.



Hình 2.3: Axios

Dưới đây là một số khái niệm chính liên quan đến Axios:

Gửi yêu cầu HTTP: Axios cho phép bạn gửi các yêu cầu HTTP như GET, POST, PUT, DELETE, v.v. Bạn có thể chỉ định URL, dữ liệu gửi kèm, tiêu đề, thời gian chờ, v.v. trong yêu cầu.

Promise-based: Axios sử dụng Promise để xử lý yêu cầu và phản hồi. Điều này

cho phép bạn sử dụng cú pháp như `.then()` và `.catch()` để xử lý kết quả và lỗi của yêu cầu.

Tương thích với JSON: Axios hỗ trợ gửi và nhận dữ liệu JSON một cách dễ dàng. Bạn có thể gửi đối tượng JavaScript và Axios sẽ tự động chuyển đổi chúng thành chuỗi JSON. Ngược lại, Axios cũng tự động phân tích các phản hồi JSON thành đối tượng JavaScript.

Xử lý phản hồi: Axios cung cấp các phương thức để truy cập vào dữ liệu phản hồi, tiêu đề, mã trạng thái và thông tin khác từ phản hồi HTTP. Bạn có thể xử lý phản hồi thành công hoặc xử lý lỗi nếu có.

Cấu hình toàn cục: Bạn có thể cấu hình Axios một cách toàn cục để thiết lập các thông số chung cho tất cả các yêu cầu, chẳng hạn như URL cơ sở, tiêu đề, thời gian chờ, v.v.

2.5. Giới thiệu NodeJS

Node.js là một môi trường thực thi mã nguồn mở xây dựng dựa trên JavaScript và được sử dụng để chạy mã nguồn JavaScript ở phía máy chủ. Nó được xây dựng trên Chrome V8 JavaScript runtime engine, tạo ra một môi trường thực thi hiệu quả cho việc phát triển ứng dụng web và các dịch vụ máy chủ

2.5.1. Khái niệm cơ bản về NodeJS

Node.js là một môi trường thực thi mã nguồn mở xây dựng trên JavaScript và được sử dụng để chạy mã nguồn JavaScript ở phía máy chủ. Nó giúp xây dựng các ứng dụng máy chủ hiệu quả, có khả năng xử lý đồng thời cao và phản hồi nhanh chóng. Node.js sử dụng mô hình không đồng bộ và sự kiện để tối ưu hóa hiệu suất.

2.5.2. Đặc điểm NodeJS

JavaScript Everywhere: Node.js sử dụng JavaScript, cho phép lập trình viên sử dụng cùng một ngôn ngữ (JavaScript) cả ở phía máy chủ và máy khách.

Asynchronous và Non-Blocking: Node.js sử dụng mô hình không đồng bộ, giúp xử lý nhiều yêu cầu một cách hiệu quả mà không làm chậm quá trình thực thi. Sự kiện (event) và callback chơi vai trò quan trọng trong lập trình Node.js.

V8 JavaScript Engine: Sử dụng trình duyệt Chrome V8 JavaScript runtime engine, giúp tối ưu hóa hiệu suất thực thi mã nguồn JavaScript.

NPM (Node Package Manager): NPM là một trình quản lý gói mạnh mẽ, giúp cài đặt, quản lý và chia sẻ các thư viện, frameworks và công cụ phát triển.

Server-Side Development: Node.js được thiết kế chủ yếu để xây dựng ứng dụng máy chủ, đặc biệt là cho các ứng dụng web yêu cầu khả năng xử lý đồng thời.

Cross-Platform: Node.js có sẵn trên nhiều nền tảng, bao gồm Windows, macOS và Linux, cho phép việc phát triển đa nền tảng một cách dễ dàng.

Community và Ecosystem: Node.js có một cộng đồng lớn, tích cực và sáng tạo, với nhiều thư viện và frameworks như Express.js, Socket.io và hơn thế nữa.

Real-time Applications: Node.js được sử dụng rộng rãi cho các ứng dụng yêu cầu tương tác thời gian thực như chat applications và game servers.

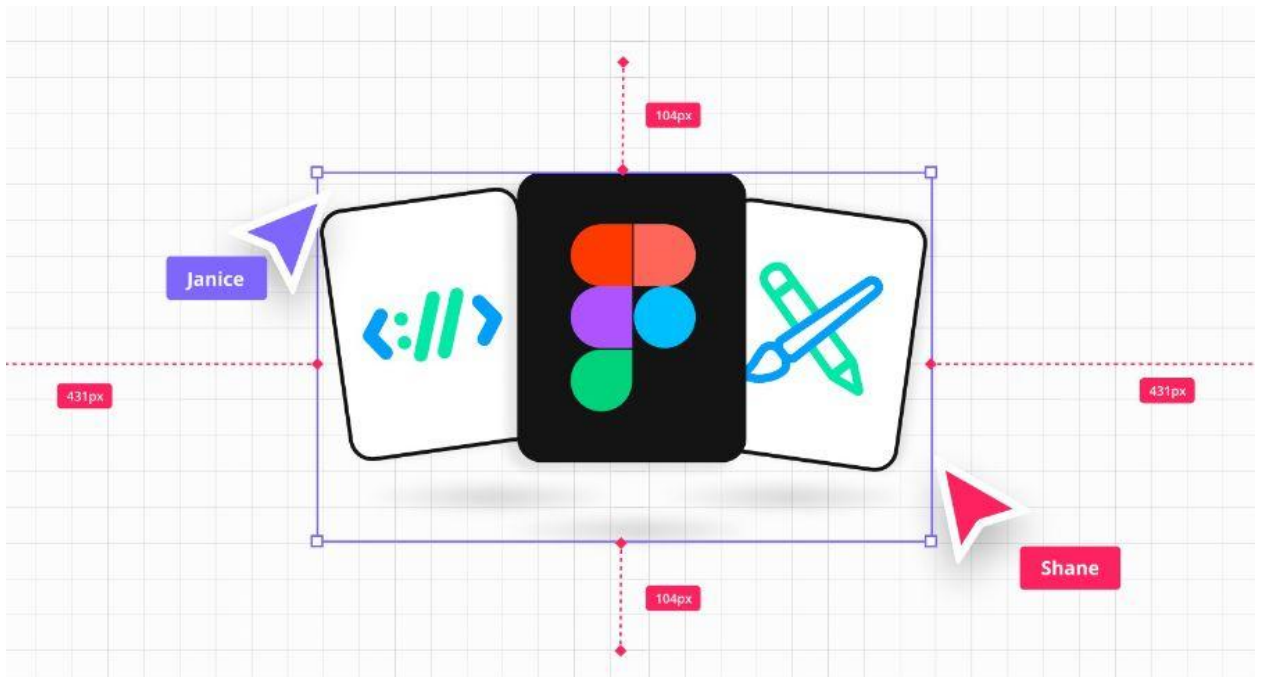
2.6. Tìm hiểu về Figma

2.6.1. Giới thiệu về Figma

- Figma là một ứng dụng hoạt động dựa trên nền tảng web, được thiết kế dành riêng cho việc thiết kế giao diện và trải nghiệm người dùng (thiết kế UI UX). Figma là một công cụ tuyệt vời cho các nhà thiết kế giao diện, tuy nhiên Figma là ứng dụng được sử dụng rộng rãi bởi cộng đồng designers, developers, marketers và chủ doanh nghiệp.

- Figma dùng trong product design: Bạn có thể sử dụng Figma để tạo wireframes, mockup, prototypes và thiết kế tương tác. Với Figma, bạn có thể xây dựng toàn bộ trang web hoặc ứng dụng của mình ngay bên trong ứng dụng. Bạn thậm chí có thể nhập trực tiếp các tệp Photoshop PSD vào Figma.

- Figma được tạo ra vào năm 2015 bởi một nhóm nhỏ các nhà phát triển từ Google, khi họ muốn có một cách làm việc hiệu quả hơn cùng nhau trong các dự án thiết kế giao diện người dùng UI UX. Đối với họ, các công cụ sẵn có như Photoshop và Sketch không đủ linh hoạt. Vì vậy, họ bắt đầu tạo ra Figma - một ứng dụng mới để giúp mọi người hoàn thành công việc nhanh hơn.



Hình 2.4: Giới thiệu về Figma

- Figma được ra mắt công khai vào năm 2016. Ngay sau đó, Figma đã được tải xuống hàng triệu lần và được sử dụng bởi hàng nghìn nhà thiết kế chuyên nghiệp. Mức độ phổ biến của Figma đã tăng lên nhanh chóng vì nó cung cấp cho người dùng sự kết hợp độc đáo của các tính năng mà phần mềm khác không có, cho phép bạn kéo và thả hình ảnh trực tiếp vào tài liệu của mình. Đối với làm việc nhóm, Figma cũng cung cấp khả năng cộng tác dễ dàng. Bạn có thể chia sẻ tài liệu với người khác và nhận xét về những thay đổi do người khác thực hiện.

2.6.2. Ưu điểm của Figma.

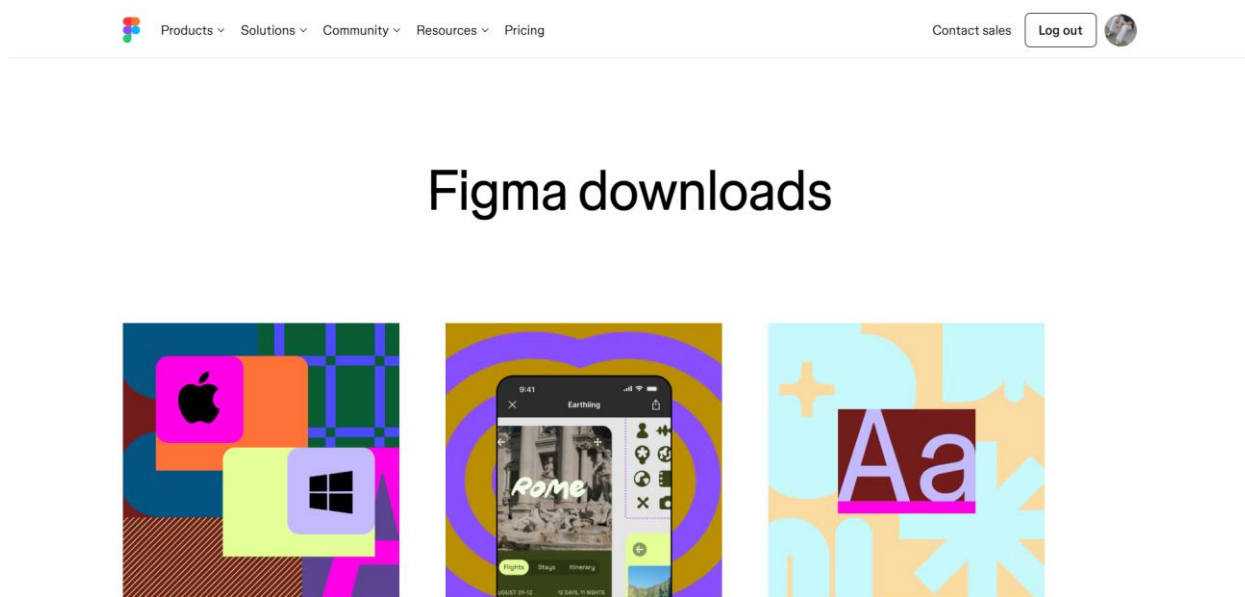
- Figma là một phần mềm thiết kế thân thiện với người dùng, Figma cho phép người dùng tạo đồ họa và hình minh họa tuyệt đẹp một cách dễ dàng. Bạn có muốn tạo ra các thiết kế đẹp, chuyên nghiệp nhanh hơn và dễ dàng hơn bao giờ hết không? Nếu vậy, Figma có thể là ứng dụng hoàn hảo dành cho bạn dưới đây là một số ưu điểm của Figma:

- **Figma giúp dễ dàng tạo Website Mockup** - Ưu điểm rõ ràng nhất của Figma là bạn không cần biết cách viết mã code. Bạn chỉ cần kéo và thả các đối tượng vào canvas để xây dựng prototype của mình.

- **Giao diện Figma đơn giản, dễ sử dụng** - Nếu bạn đã từng thử sử dụng Photoshop trước đây, có thể bạn sẽ thấy khó điều hướng qua giao diện. Tuy nhiên,

Figma dễ sử dụng hơn nhiều vì chỉ cần một cú nhấp chuột để thêm đối tượng.

- **Figma hỗ trợ cả máy tính để bàn và thiết bị di động** - Bạn không cần phải lo lắng về các vấn đề tương thích nếu chọn Figma. Vì Figma hoạt động tốt trên mọi thiết bị bao gồm máy tính để bàn, máy tính xách tay, máy tính bảng, điện thoại thông minh, nên bạn có thể dễ dàng truy cập các file thiết kế của mình từ mọi nơi.



Hình 2.5: Figma hỗ trợ cả máy tính bàn và thiết bị di động

- **Với Figma, chương trình tự động sẽ điều chỉnh bố cục của bạn theo kích thước của màn hình.** Khi bạn đang tạo một trang web, bạn sẽ thường cần thay đổi kích thước các thành phần trên trang của mình. Nếu bạn không cẩn thận, những thành phần này có thể trông lộn xộn khi xem trên màn hình nhỏ hơn. Nhưng với Figma, bạn không cần lo lắng.

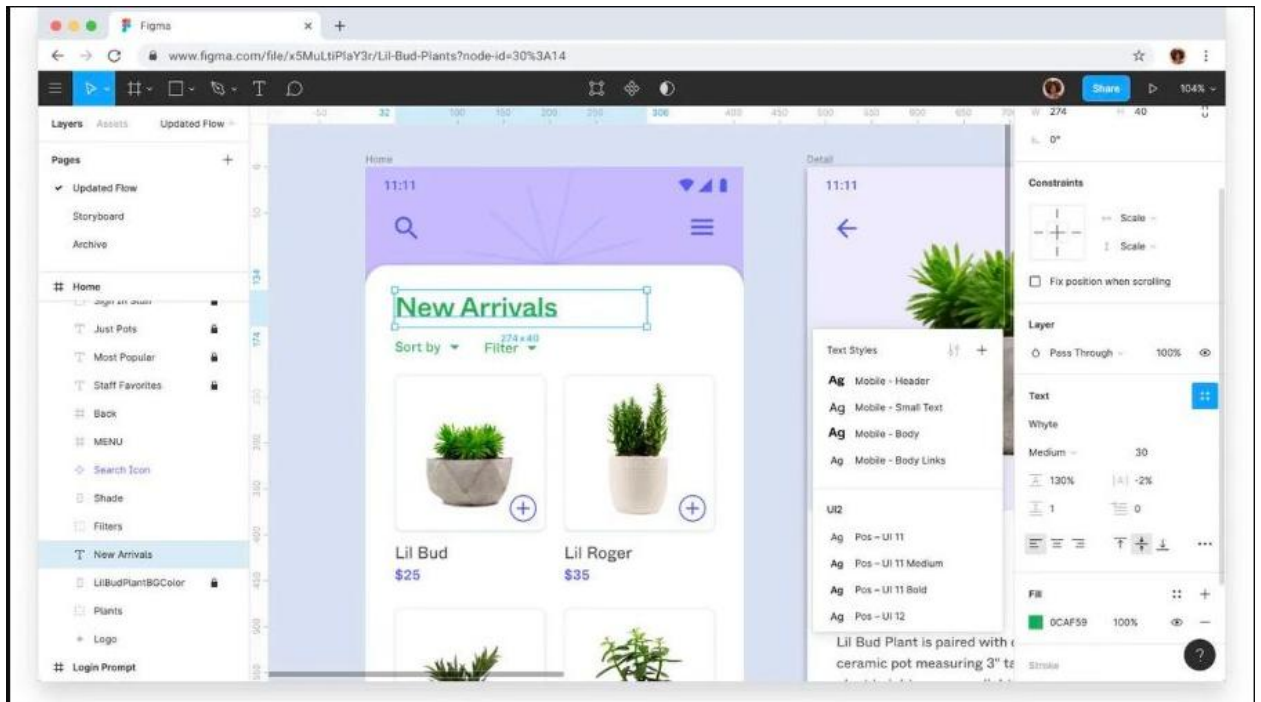
- **Figma hỗ trợ trên nhiều nền tảng** - Figma tương thích với nhiều nền tảng bao gồm Windows, Mac OS X, iOS, Android và Linux. Vì vậy, bạn không cần phải lo lắng về vấn đề tương thích.

- **Chia sẻ tệp nhanh chóng trên Figma** - Sau khi thiết kế xong dự án, bạn có thể chia sẻ trực tiếp từ cửa sổ trình duyệt. Điều này giúp bạn dễ dàng gửi thiết kế của mình cho khách hàng mà không cần phải lưu chúng trước.



Hình 2.6: Chia sẻ tệp nhanh chóng trên Figma

- **Mẫu trên Figma có thể tùy chỉnh** - Với Figma, bạn có thể tùy chỉnh các mẫu có sẵn dựa trên nhu cầu của mình. Ví dụ: bạn có thể thay đổi màu sắc, phông chữ và hình nền để tạo phong cách độc đáo của riêng mình.
- **Figma hỗ trợ xuất dưới dạng PDF** - Figma cho phép bạn xuất các dự án của mình dưới dạng tài liệu PDF. Điều này có nghĩa là bạn có thể in các thiết kế của mình ra và giao cho khách hàng
- **Figma giúp dễ dàng cộng tác với những người khác** - Vì Figma dựa trên đám mây nên bạn có thể cộng tác với những người khác bằng cách chia sẻ tệp của mình trực tuyến.



Hình 2.7: Figma giúp dễ dàng cộng tác với những người khác

- **Figma dễ học** - Trong khi một số người thấy Photoshop đáng sợ, thì Figma lại rất dễ học. Ngay cả những người mới bắt đầu cũng có thể sử dụng nó trong vòng vài phút.
- **Figma sở hữu công cụ mạnh mẽ** - Một trong những tính năng tốt nhất của Figma là nó đi kèm với những công cụ mạnh mẽ như Pen tool, Shape tool, Text tool, and Magic Wand tool.
- **Figma có tốc độ tải nhanh** - Một lý do khác khiến Figma là lựa chọn phổ biến của các nhà thiết kế là nó tải nhanh. Chỉ mất chưa đầy hai giây để mở một tệp mới.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương 3 này tập trung vào phân tích thiết kế hệ thống của một ứng dụng quản lý thông tin y tế.

3.1. Phân tích hệ thống

3.1.1. Các chức năng hệ thống

- Xem trang chủ màn hình chính: gồm có các thông tin về tài khoản của người dùng, thông tin các bác sĩ, thông tin các chuyên khoa khám bệnh.
- Xem chi tiết thông tin bác sĩ
- Xem chi tiết thông tin chuyên khoa khám bệnh
- Xem lịch sử khám bệnh: bệnh nhân có thể xem lại lịch sử khám bệnh của mình
- Đăng nhập và phân quyền: mỗi người dùng có một quyền nhất định. Tài khoản quản trị viên sẽ có tất cả các quyền. Tài khoản của bác sĩ thì không thể thực hiện xóa, sửa, thêm tài khoản, được quyền thêm những ca chính bác sĩ ấy trông để chữa bệnh và quản lý bệnh nhân. Tài khoản bệnh nhân thì chỉ xem được các thông tin bác sĩ, bệnh viện, chuyên khoa khám bệnh, lịch sử khám bệnh và thông tin cá nhân của mình.
- Đăng ký tài khoản: mặc định đăng ký tài khoản thành công thì sẽ trao cho tài khoản đăng ký là bệnh nhân.
- Cập nhật thông tin bệnh nhân: bệnh nhân có thể cập nhật thông tin hồ sơ của mình
- Quản lý bệnh nhân: bác sĩ có thể xem thông tin các bệnh nhân đặt lịch, gửi qua email cho bệnh nhân.

- Quản lý thời gian trống để bệnh nhân đặt lịch: bác sĩ
- Chọn ngày trong hệ thống sau đó chọn ca mà bác sĩ rảnh có thể khám bệnh để bệnh nhân biết và đặt lịch khám bệnh cho chính mình
- Quản lý tài khoản: Quản trị viên có thể xem thông tin của tất cả tài khoản trong hệ thống(bác sĩ, bệnh nhân)
- Quản lý chuyên khoa: Quản trị viên có thể thêm, cập nhật, xóa chuyên khoa khám bệnh
- Thống kê doanh thu các chuyên khoa
- Thống kê tổng số bác sĩ
- Thống kê tổng số lịch hẹn khám bệnh đã hoàn thành trong hệ thống

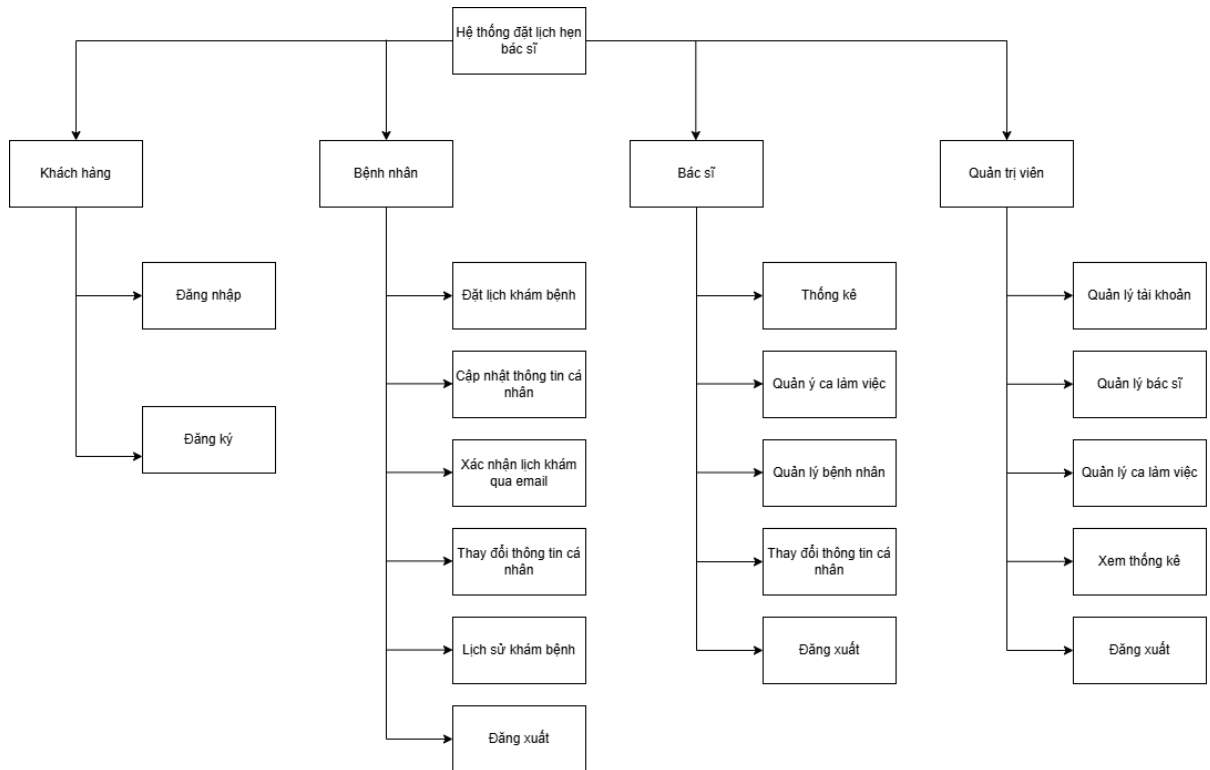
3.1.2. Phân tích tác nhân hệ thống

Tác nhân quản trị viên (Admin) có các chức năng: quản lý người dùng, quản lý bác sĩ, quản lý lịch khám, thống kê doanh thu.

Tác nhân người dùng (khách hàng) có các chức năng: đăng ký tài khoản, đăng nhập, xem thông tin chi tiết thông tin bác sĩ, tìm kiếm, đặt lịch khám.

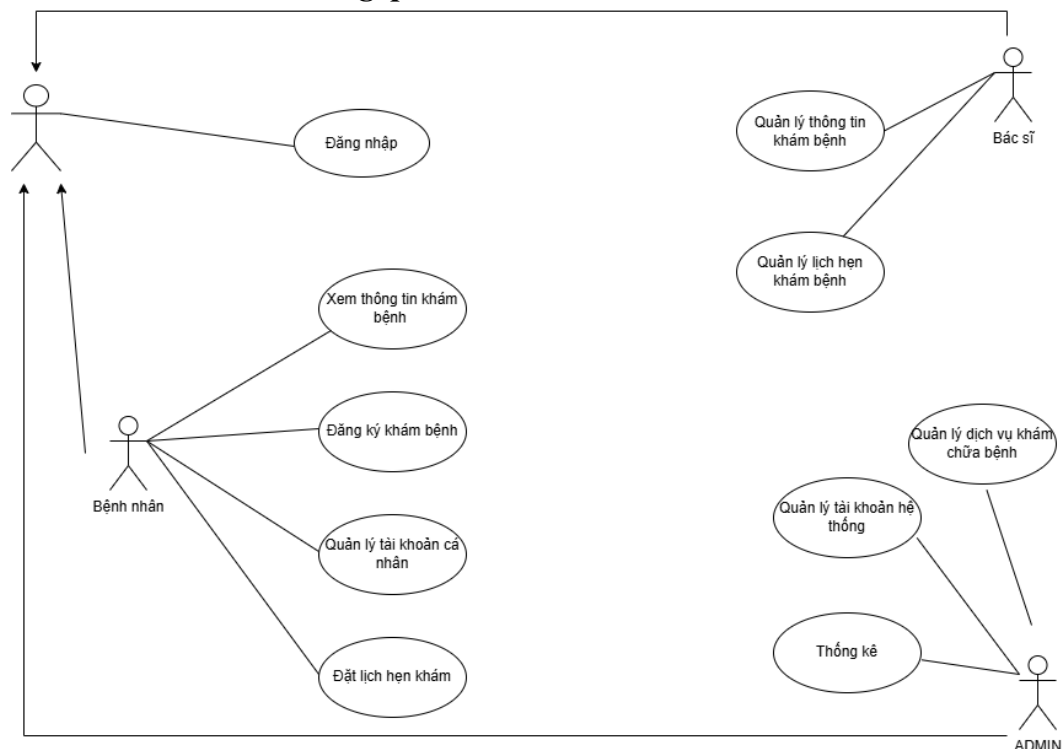
3.2. Phân tích hệ thống

3.2.1. Sơ đồ chức năng



Hình 3.1: Sơ đồ tổng quát

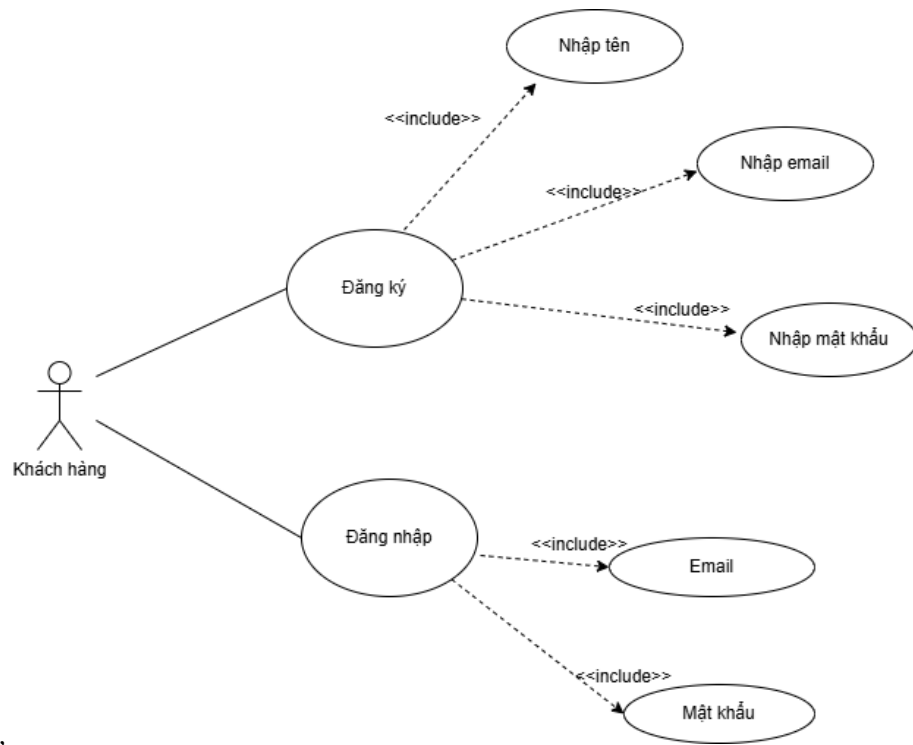
3.2.2. Biểu đồ usecase tổng quan



Hình 3.2: Biểu đồ usecase tổng quan

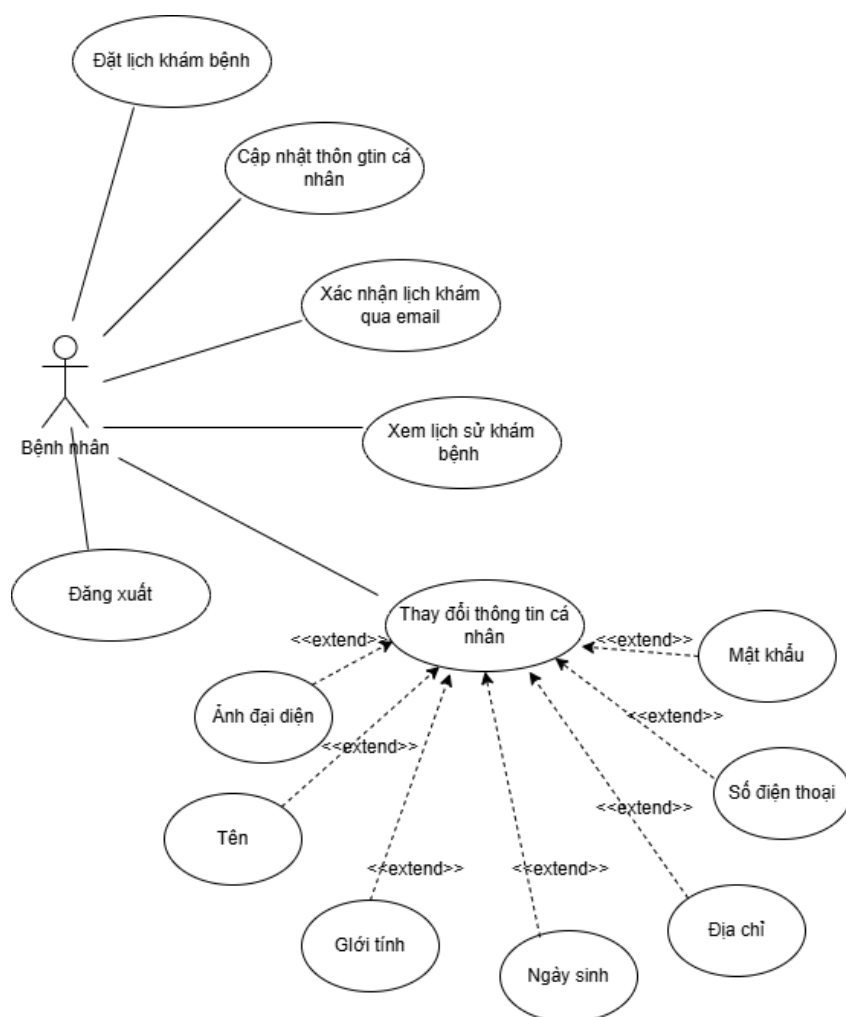
3.2.3. Lược đồ usecase

- ❖ Khách hàng
 - ✓ Lược đồ usecase khách hàng



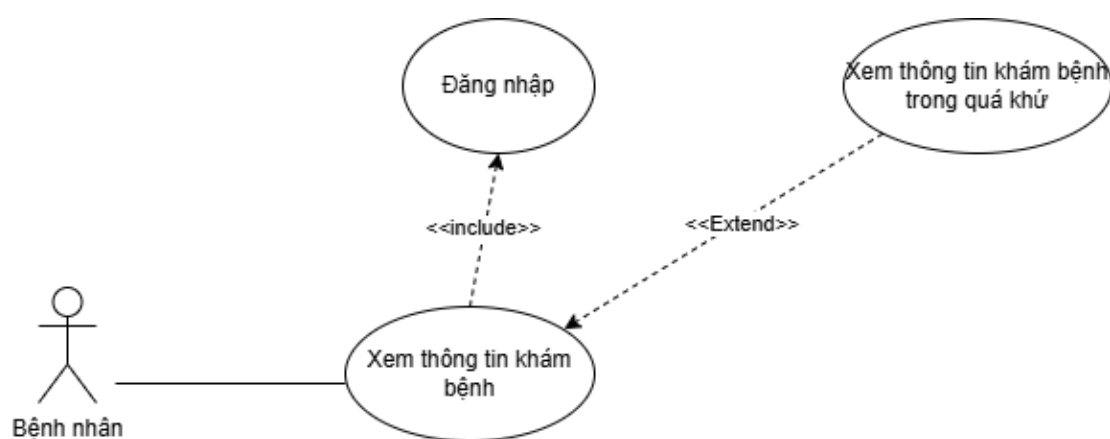
Hình 3.3: Lược đồ usecase khách hàng

❖ Bệnh nhân



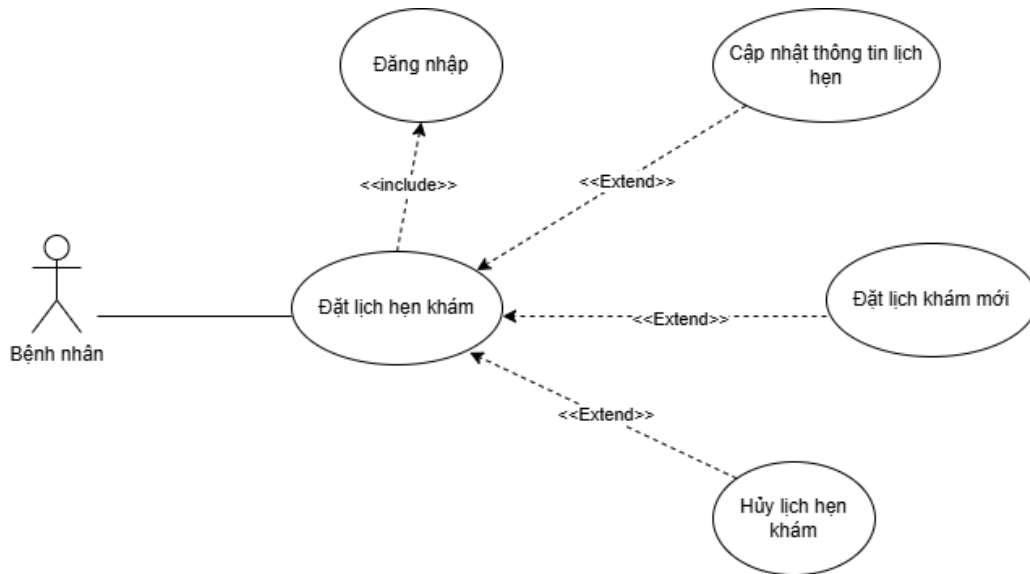
Hình 3.4: Lược đồ usecase bệnh nhân

✓ Usecase xem thông tin khám bệnh



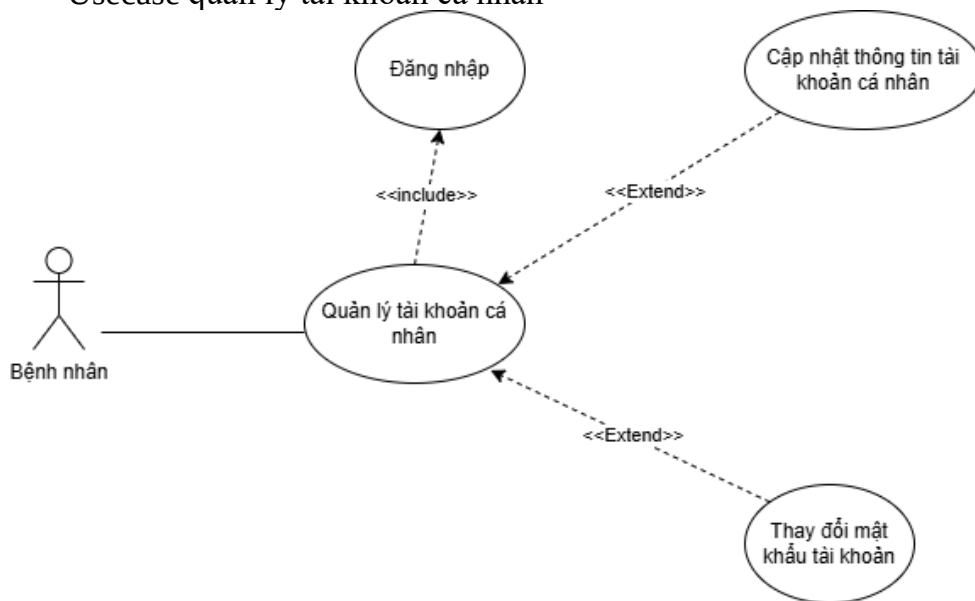
Hình 3.5: Usecase xem thông tin khám bệnh

✓ Usecase đặt lịch hẹn khám



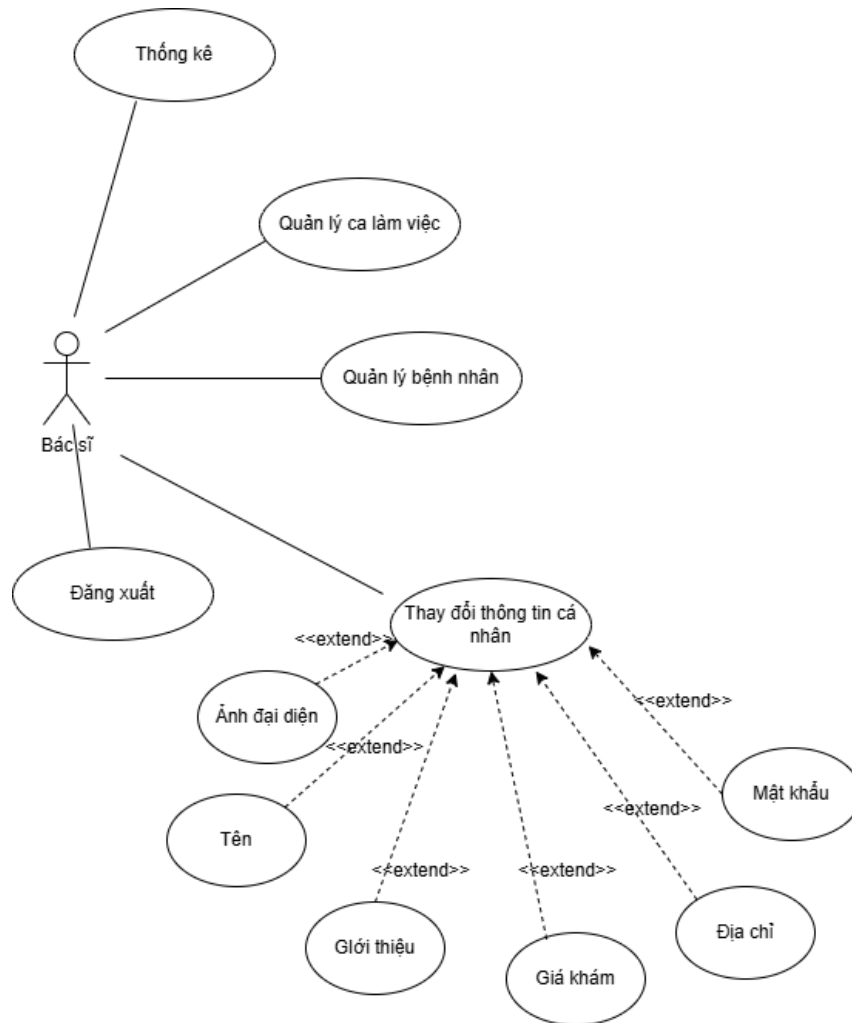
Hình 3.6: Usecase đặt lịch hẹn khám

✓ Usecase quản lý tài khoản cá nhân



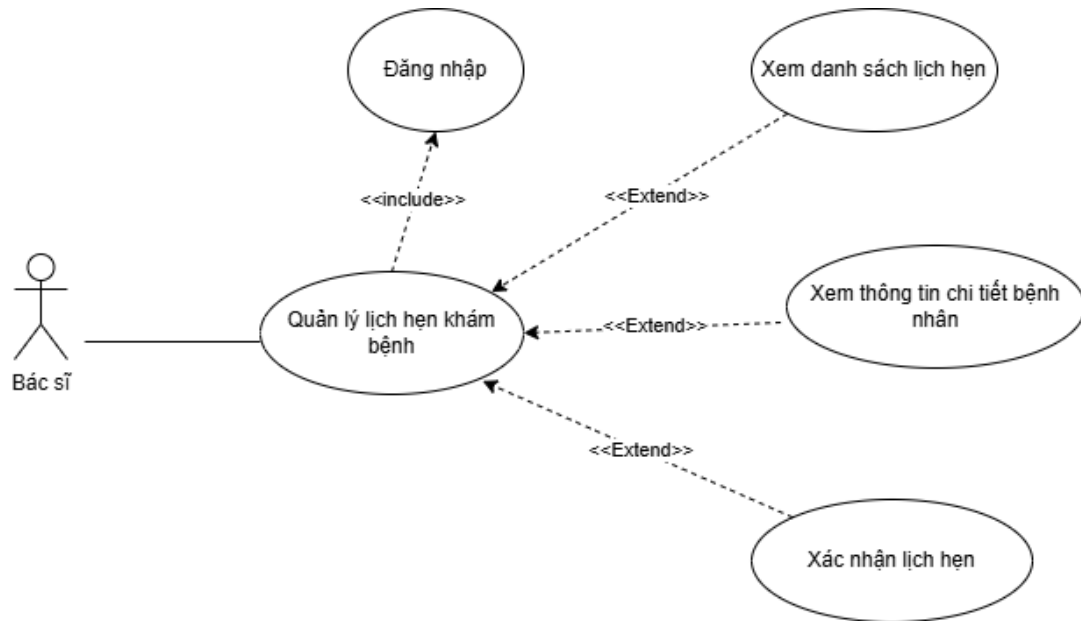
Hình 3.7: Usecase quản lý tài khoản cá nhân

❖ Bác sĩ



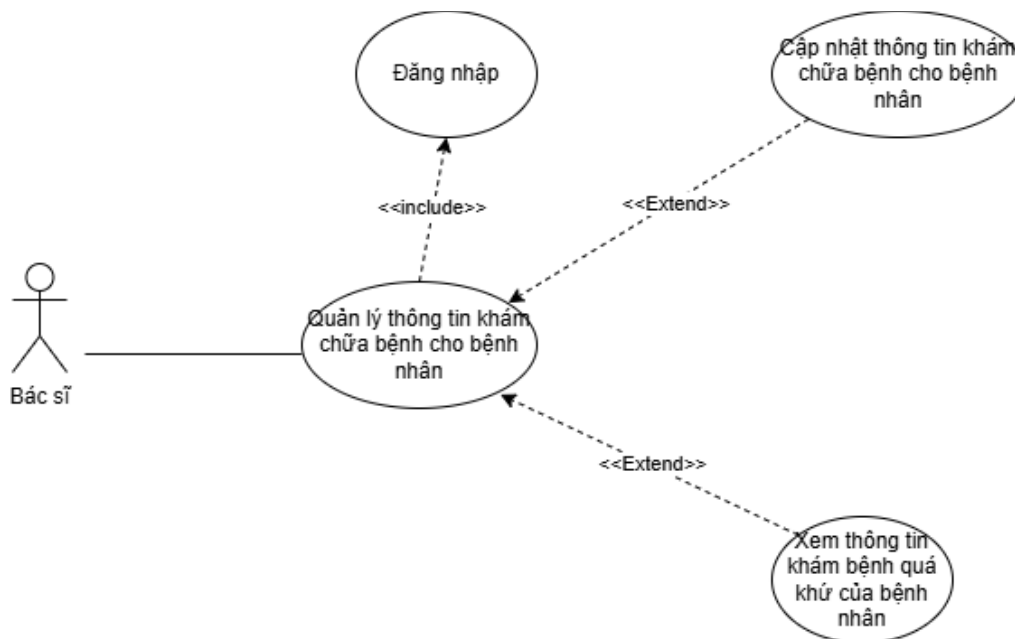
Hình 3.8: Lược đồ usecase bác sĩ

✓ Usecase quản lý lịch hẹn khám bệnh



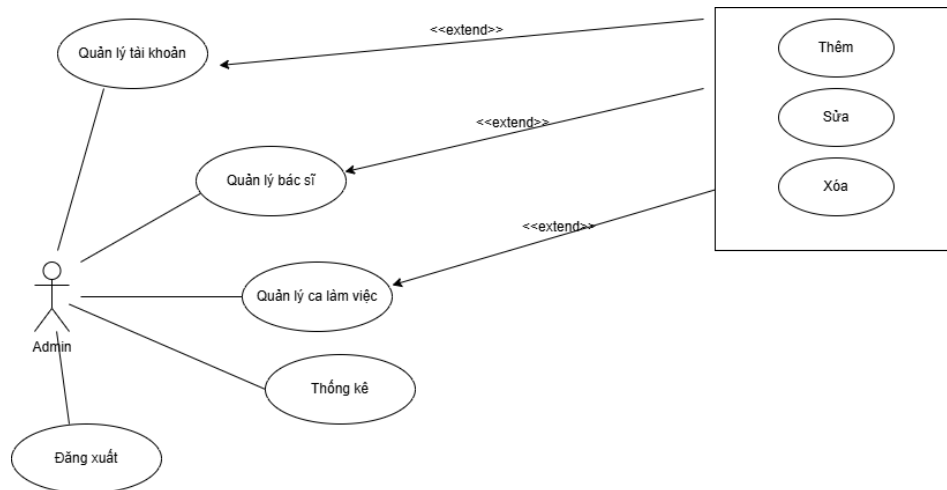
Hình 3.9: Usecase quản lý lịch hẹn khám bệnh

✓ Usecase quản lý thông tin khám bệnh cho bệnh nhân



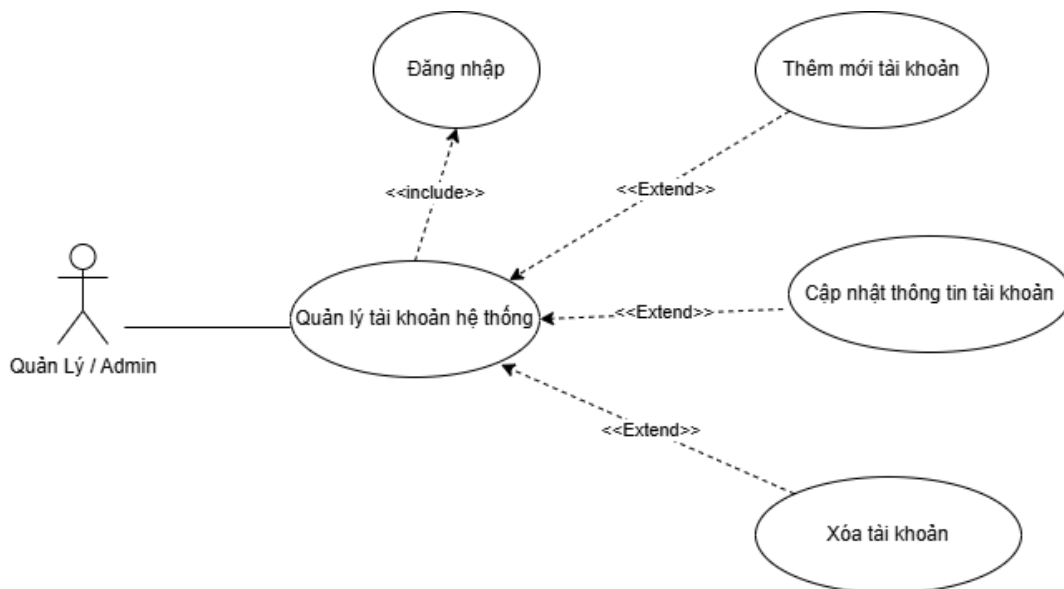
Hình 3.10: Usecase quản lý thông tin khám bệnh cho bệnh nhân

❖ Quản trị viên



Hình 3.11: Lược đồ usecase Admin

✓ Usecase quản lý tài khoản hệ thống



Hình 3.12: Usecase quản lý tài khoản hệ thống

3.2.4. Đặc tả usecase

✓ Usecase Lọc

Use Case – Lọc	
Tên	Lọc
Định nghĩa	Cho phép lọc các bác sĩ
Actors	Khách hàng
Điều kiện tiên quyết	Khách hàng đang ở giao diện website
Dòng sự kiện	
Dòng sự kiện chính	1. Khách hàng di chuyển chuột vào thanh lọc 2. Hệ thống sẽ kiểm tra thông tin lọc 3. Nếu bác sĩ tồn tại thì kết quả tìm kiếm sẽ được hiển thị
Dòng sự kiện thay thế	Không
Điều kiện sau	Hiển thị danh sách kết lọc

Bảng 3.1: Usecase lọc

✓ Usecase xem chi tiết bác sĩ

Use Case – Xem chi tiết bác sĩ	
Tên	Xem chi tiết bác sĩ
Định nghĩa	Cho phép khách hàng xem chi tiết về các thông tin bác sĩ có trong website
Actors	Khách hàng
Điều kiện tiên quyết	Khách hàng đang ở giao diện website
Dòng sự kiện	
Dòng sự kiện chính	1. Khách hàng chọn bác sĩ cần xem 2. Hệ thống hiển thị thông tin chi tiết về bác sĩ đã chọn
Dòng sự kiện thay thế	Không

Điều kiện sau	Hiển thị thông tin chi tiết bác sĩ đã chọn
---------------	--

Bảng 3.2: Usecase xem chi tiết bác sĩ

✓ Usecase đặt lịch khám bệnh

Use Case – Đặt lịch khám bệnh	
Tên	Đặt lịch khám bệnh
Định nghĩa	Cho phép khách hàng tùy chọn bác sĩ và giờ hẹn để đặt lịch khám có trong website
Actors	Khách hàng
Điều kiện tiên quyết	Khách hàng đang ở giao diện website
Dòng sự kiện	
Dòng sự kiện chính	1. Khách hàng chọn bác sĩ cần xem và muốn đặt lịch khám 2. Hệ thống hiển thị thông tin chi tiết về bác sĩ đã chọn 3. Khách hàng chọn khung giờ thích hợp để đặt lịch 4. Khách hàng nhập ghi chú 5. Bấm xác nhận lịch khám bệnh 6. Nếu mọi thông tin điều đúng và hợp lệ hệ thống sẽ xác nhận đặt lịch thành công
Dòng sự kiện thay thế	Không
Điều kiện sau	Hiển thị thông tin đặt lịch khám thành công

Bảng 3.3: Use case đặt lịch khám

✓ Usecase xác nhận đăng ký lịch khám bệnh

Use Case – Xác nhận đăng ký lịch khám bệnh và gửi gmail xác nhận	
Tên	Xác nhận đăng ký lịch khám bệnh
Định nghĩa	Khách hàng sẽ nhận được gmail khi xác nhận lịch hẹn thành công
Actors	Bác sĩ
Điều kiện tiên quyết	Khi xác nhận lịch hẹn
Dòng sự kiện	

Dòng sự kiện chính	Lịch hẹn sẽ xác nhận khi bác sĩ chấp nhận lịch hẹn sau đó hệ thống sẽ gửi gmail thông báo đã xác nhận thành công
Dòng sự kiện thay thế	Không
Điều kiện sau	Hiển thị thông tin đặt lịch khám thành công

Bảng 3.4: Usecase xác nhận lịch khám bệnh thông báo gmail

✓ Usecase đăng nhập

Use Case – Đăng nhập	
Tên	Đăng nhập
Định nghĩa	Đăng nhập vào hệ thống website
Actors	Người dùng, bác sĩ, admin
Điều kiện tiên quyết	Đã truy cập vào trang chủ
Dòng sự kiện	
Dòng sự kiện chính	1. Chọn mục đăng nhập 2. Nhập thông tin cần thiết 3. Bấm đăng nhập. 4. Hệ thống kiểm tra thông tin nếu không hợp lệ sẽ quay lại login 5. Trả về giao diện trang chủ khi đăng nhập thành công
Dòng sự kiện thay thế	Quay lại màn hình đăng nhập
Điều kiện sau	Hiển thị thông tin trang chủ

Bảng 3.5: Usecase đăng nhập

✓ Usecase đăng ký

Use Case – Đăng ký	
Tên	Đăng ký
Định nghĩa	Đăng ký tài khoản sử dụng chức năng đặt lịch
Actors	Khách hàng
Điều kiện tiên quyết	Đã truy cập vào trang chủ
Dòng sự kiện	

Dòng sự kiện chính	1. Chọn mục đăng ký 2. Nhập thông tin cần thiết 3. Bấm đăng ký. 4. Nếu hợp lệ hệ thống tự động đăng nhập vào hệ thống 5. Hệ thống di chuyển đến trang chủ
Dòng sự kiện thay thế	Quay lại màn hình đăng ký

Bảng 3.6: Usecase đăng ký

✓ Usecase đăng nhập admin

Use Case – Đăng nhập	
Tên	Đăng nhập
Định nghĩa	Người quản trị muốn vào hệ thống
Actors	Người quản trị
Điều kiện tiên quyết	Đã truy cập vào trang admin
Dòng sự kiện	
Dòng sự kiện chính	Hệ thống hiển thị giao diện đăng nhập, yêu cầu Người quản trị nhập tên tài khoản và mật khẩu. Người quản trị nhập tên tài khoản và mật khẩu, chọn đăng nhập. Hệ thống kiểm tra thông tin tài khoản và mật khẩu, nếu nhập sai tên tài khoản và mật khẩu thì không vào.
Dòng sự kiện thay thế	Không
Điều kiện sau	Người quản trị đăng nhập thành công và có thể sử dụng các chức năng mà hệ thống cung cấp.

Bảng 3.7: Usecase đăng nhập admin

✓ Usecase đăng ký tài khoản bác sĩ

Use Case – Đăng ký tài khoản bác sĩ	
Tên	Đăng ký tài khoản cho bác sĩ
Định nghĩa	Cho phép admin duyệt đăng ký tài khoản cho bác sĩ
Actors	Khách hàng
Điều kiện tiên quyết	Đã truy cập vào trang chủ
Dòng sự kiện	

Dòng sự kiện chính	1. Chọn chức năng hợp tác cho bác sĩ 2. Nhập thông tin cần hợp tác 3. Bấm gửi thông tin 4. Admin kiểm tra thông tin nếu không hợp lệ sẽ báo không được duyệt 5. Nếu thông tin hợp lệ gửi email xác nhận trở thành bác sĩ
Dòng sự kiện thay thế	Quay lại màn hình đăng ký thông tin cho bác sĩ
Điều kiện sau	Hiển thị thông tin đăng ký thành công

Bảng 3.8: Usecase đăng ký tài khoản bác sĩ

✓ Usecase sửa thông tin tài khoản bác sĩ

Use Case – Sửa thông tin tài khoản bác sĩ	
Tên	Sửa thông tin tài khoản cho bác sĩ
Định nghĩa	Cho phép admin sửa tài khoản cho bác sĩ
Actors	Người quản trị
Điều kiện tiên quyết	Đã truy cập vào trang admin
Dòng sự kiện	
Dòng sự kiện chính	1. Chọn danh sách thông tin của bác sĩ 2. Chọn bác sĩ cần sửa thông tin 3. Bấm chọn sửa và bấm lưu thông tin 4. Hệ thống kiểm tra thông tin nếu không hợp lệ sẽ quay lại màn hình
Dòng sự kiện thay thế	Quay lại màn hình thông tin của bác sĩ
Điều kiện sau	Hiển thị thông tin sửa thành công

Bảng 3.9: Usecase sửa thông tin bác sĩ

✓ Usecase xóa tài khoản bác sĩ

Use Case – Xóa tài khoản bác sĩ	
Tên	Xóa thông tin tài khoản cho bác sĩ
Định nghĩa	Cho phép admin xóa tài khoản cho bác sĩ
Actors	Người quản trị
Điều kiện tiên quyết	Đã truy cập vào trang admin
Dòng sự kiện	

Dòng sự kiện chính	1. Chọn danh sách thông tin của bác sĩ 2. Chọn bác sĩ cần xóa thông tin 3. Bấm chọn xóa 4. Hệ thống kiểm tra thông tin nếu không hợp lệ sẽ quay lại màn hình
Dòng sự kiện thay thế	Quay lại màn hình thông tin của bác sĩ
Điều kiện sau	Hiển thị thông tin xóa thành công

Bảng 3.10: Usecase xóa tài khoản bác sĩ

✓ Usecase thêm bác sĩ

Use Case – Thêm thông tin chi tiết bác sĩ	
Tên	Thêm bác sĩ
Định nghĩa	Cho phép admin thêm bác sĩ
Actors	Người quản trị
Điều kiện tiên quyết	Đã truy cập vào trang admin
Dòng sự kiện	
Dòng sự kiện chính	1. Chọn mục thêm mới bác sĩ 2. Nhập thông tin cần thiết 3. Bấm thêm bác sĩ 4. Hệ thống kiểm tra thông tin nếu không hợp lệ sẽ quay lại màn hình
Dòng sự kiện thay thế	Quay lại màn hình thêm bác sĩ
Điều kiện sau	Hiển thị thông tin thêm thành công

Bảng 3.11: Usecase thêm bác sĩ

✓ Usecase sửa thông tin chi tiết bệnh nhân

Use Case – Sửa thông tin chi tiết bệnh nhân	
Tên	Sửa thông tin cho bệnh nhân
Định nghĩa	Cho phép admin sửa thông tin cho bệnh nhân
Actors	Người quản trị
Điều kiện tiên quyết	Đã truy cập vào trang admin
Dòng sự kiện	

Dòng sự kiện chính	1. Chọn mục sửa thông tin của bệnh nhân 2. Chọn bệnh nhân cần sửa thông tin 3. Nhập thông tin cần sửa 4. Bấm lưu thông tin. 5. Hệ thống kiểm tra thông tin nếu không hợp lệ sẽ quay lại màn hình 6. Trả về giao diện nếu lưu thành công
Dòng sự kiện thay thế	Quay lại màn hình thông tin của bệnh nhân
Điều kiện sau	Hiển thị thông tin lưu thành công

Bảng 3.12: Usecase sửa thông tin chi tiết bệnh nhân

✓ Usecase thêm lịch khám cho bác sĩ

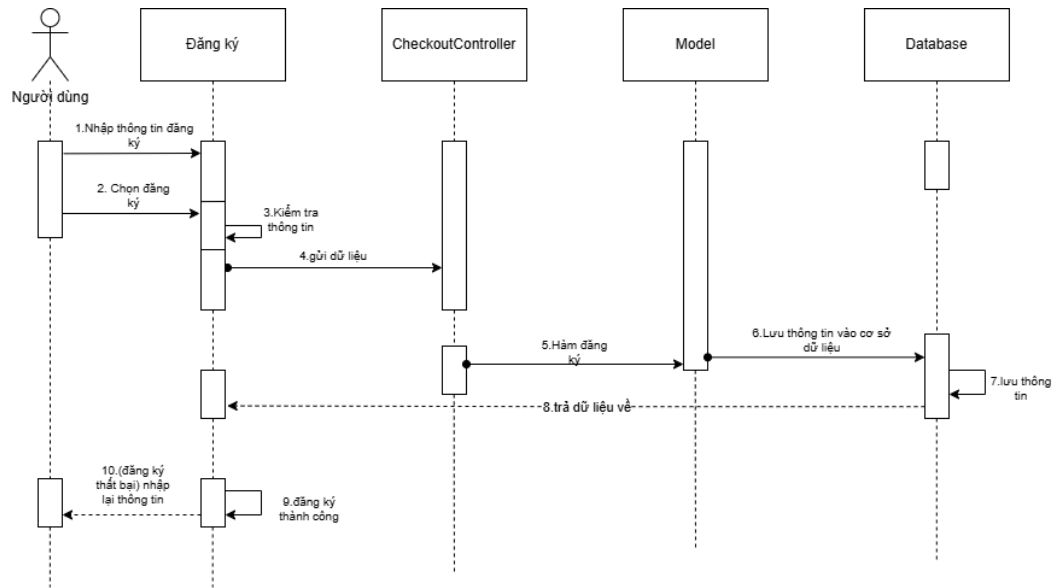
Use Case – Thêm lịch khám bác sĩ	
Tên	Thêm lịch khám cho bác sĩ
Định nghĩa	Cho phép thêm lịch khám cho bác sĩ
Actors	Bác sĩ
Điều kiện tiên quyết	Đã truy cập vào trang bác sĩ
Dòng sự kiện	
Dòng sự kiện chính	1. Chọn mục lịch trình 2. Chọn ngày và giờ để lên lịch khám cho bác sĩ 3. Bấm lưu thông tin. 4. Hệ thống kiểm tra thông tin nếu không hợp lệ sẽ quay lại màn hình Trả về giao diện nếu lưu thành công
Dòng sự kiện thay thế	Quay lại màn hình thêm lịch khám của bác sĩ
Điều kiện sau	Hiển thị thông tin lưu thành công

Bảng 3.13: Usecase thêm lịch khám

3.2.5. Sơ đồ tuần tự

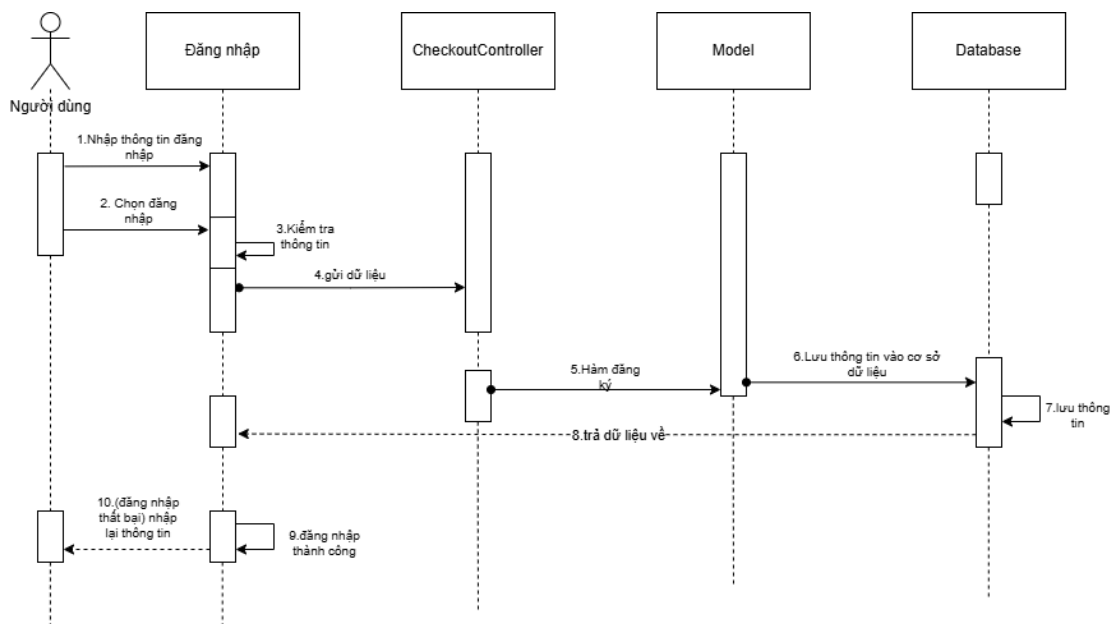
➤ Người dùng

✓ Biểu đồ tuần tự đăng ký



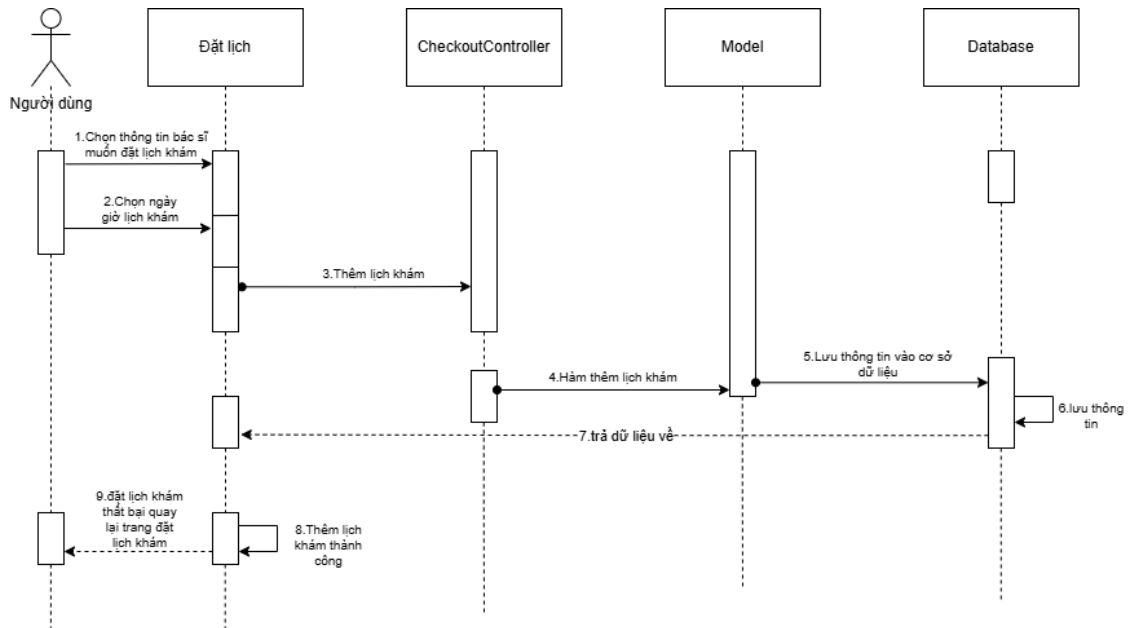
Hình 3.13: Biểu đồ tuần tự đăng ký

✓ Biểu đồ tuần tự đăng nhập



Hình 3.14: Biểu đồ tuần tự đăng nhập

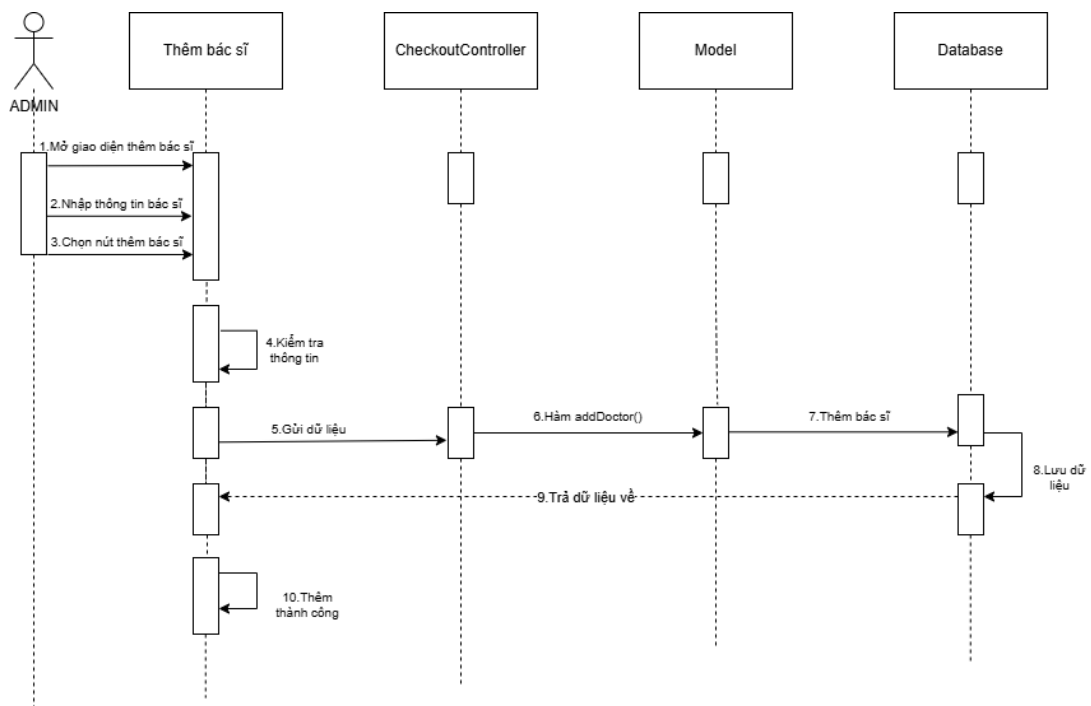
✓ Biểu đồ tuần tự đặt lịch khám bệnh



Hình 3.15: Biểu đồ tuần tự đặt lịch khám bệnh

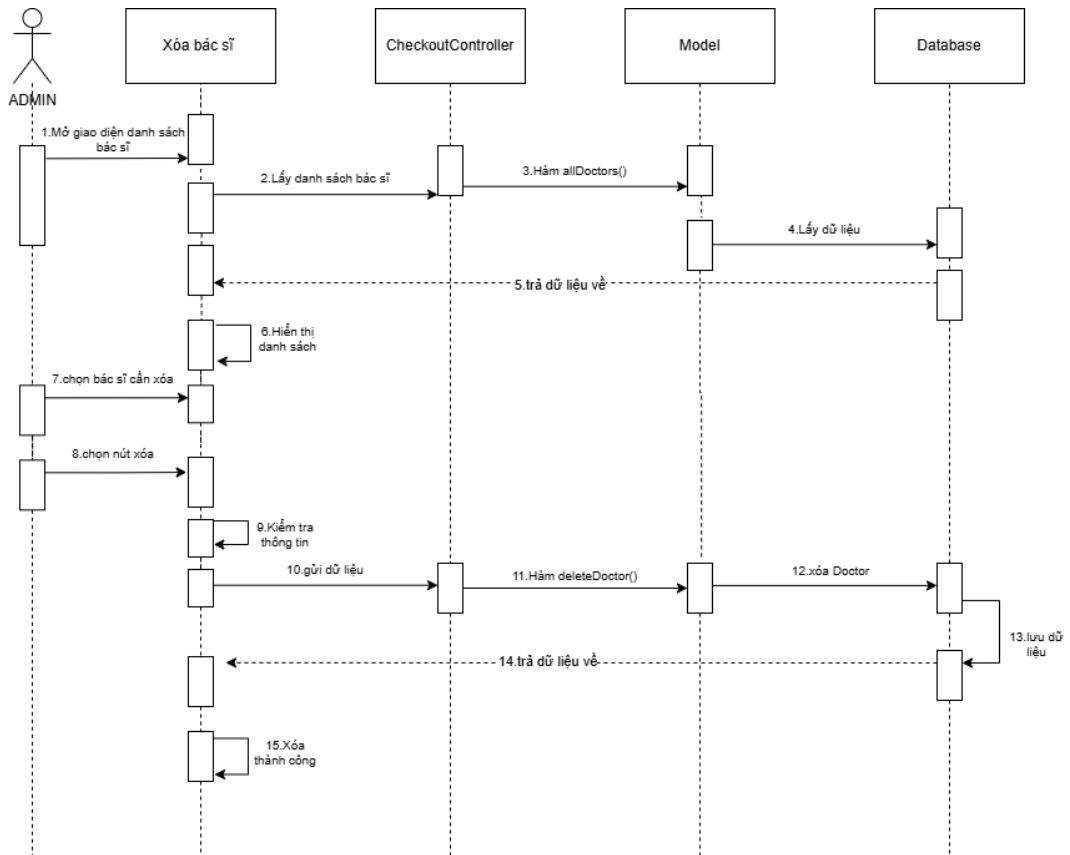
➤ Admin

✓ Biểu đồ tuần tự thêm bác sĩ



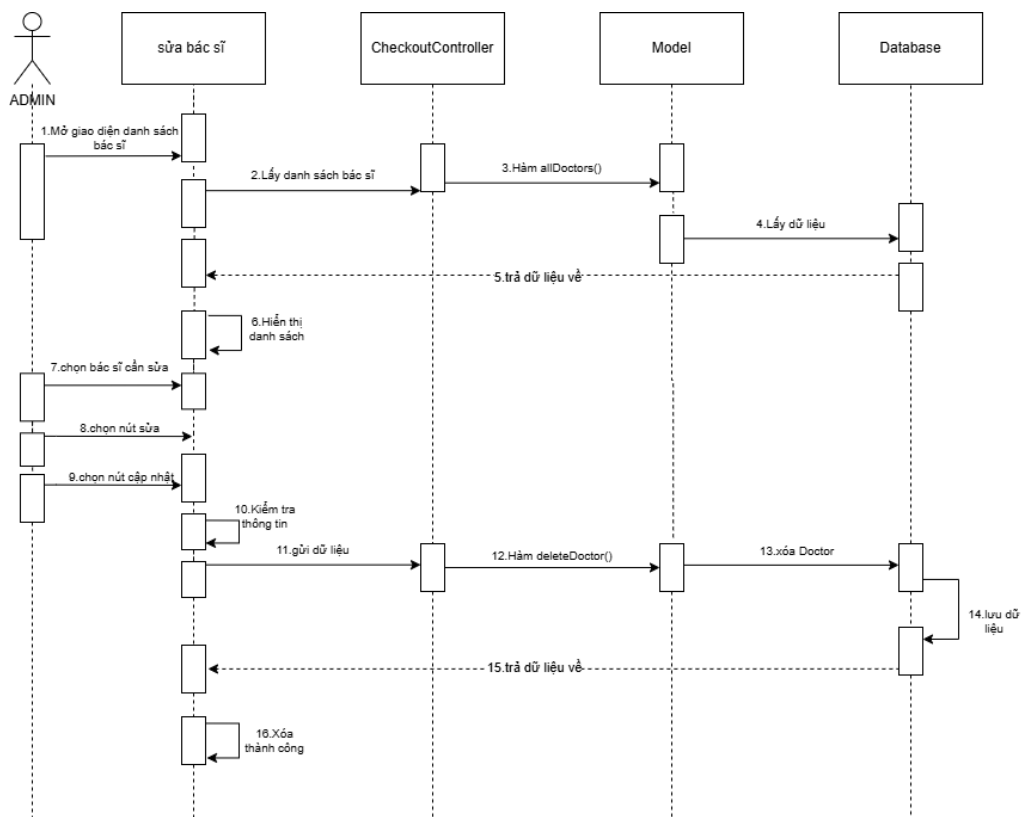
Hình 3.16: Biểu đồ tuần tự thêm bác sĩ

✓ Biểu đồ tuần tự xóa bác sĩ



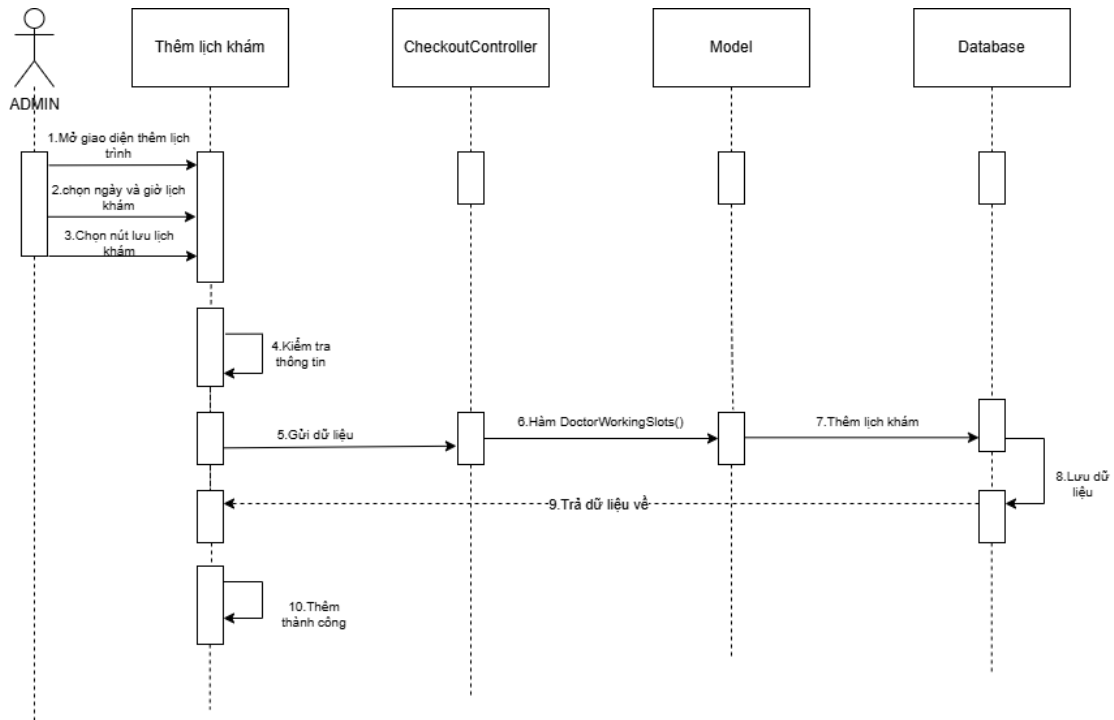
Hình 3.17: Biểu đồ xóa bác sĩ

✓ Biểu đồ tuần tự sửa bác sĩ



Hình 3.18: Biểu đồ tuần tự sửa bác sĩ

✓ Biểu đồ tuần tự thêm lịch khám

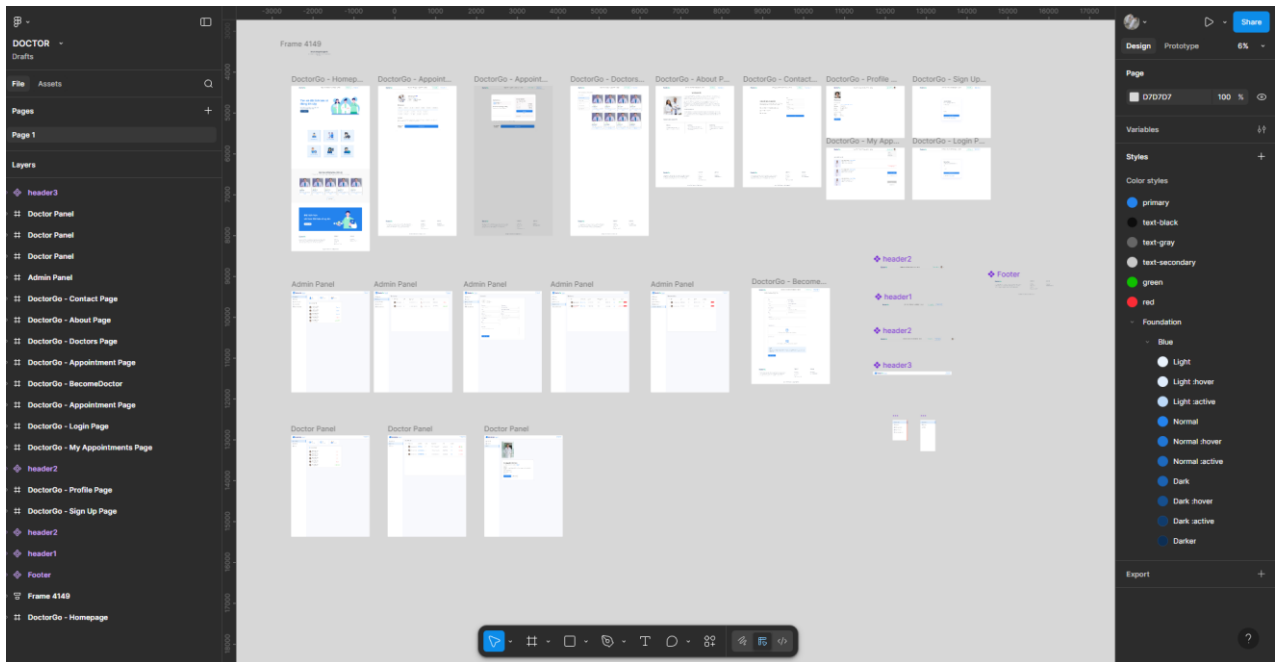


Hình 3.19: Biểu đồ tuần tự thêm lịch khám

3.2.6. Thiết kế giao diện hệ thống

✓ Mô tả:

Mục đích: Thiết kế giao diện cho hệ thống đặt lịch hẹn bác sĩ theo phong cách tối giản, phù hợp và thân thiện với người dùng giúp người dùng sử dụng 1 cách dễ dàng



Hình 3.20: Hình ảnh về giao diện trên Figma

CHƯƠNG 4. XÂY DỰNG HỆ THỐNG ĐẶT LỊCH HẸN BÁC SĨ

Chương 4 sẽ cho thấy cái nhìn tổng quát về giao diện của hệ thống đặt lịch khám chữa bệnh.

4.1. Giao diện người dùng

4.1.1. Giao diện phần trang chủ

Tại trang chủ, người dùng sẽ ngay lập tức nhận thức được sự đa dạng và toàn diện của dịch vụ y tế được cung cấp. Một phần quan trọng là thông tin về cơ sở y tế sẽ được hiển thị một cách rõ ràng.

Ngoài ra, trang chủ cũng sẽ giới thiệu về các chuyên khoa y tế mà phòng khám hỗ trợ. Điều này giúp bệnh nhân có cái nhìn tổng quan về sự chuyên sâu và chất lượng của dịch vụ, đồng thời cung cấp họ thông tin cần thiết để chọn lựa bác sĩ chuyên môn phù hợp với nhu cầu của mình.

Bên cạnh đó, thông tin về đội ngũ bác sĩ cũng được hiển thị một cách minh bạch, kèm theo hồ sơ chuyên môn và kinh nghiệm làm việc. Điều này tăng tính xác thực và uy tín của phòng khám trong tâm trí của bệnh nhân.

Trang chủ cũng là nơi tập trung các bác sĩ hàng đầu và danh mục chuyên khoa. Trang chủ của phòng khám y tế không chỉ là điểm xuất phát cho việc tìm kiếm thông tin, mà còn là bức tranh tổng quan về sự phong phú, chuyên nghiệp, và tận tâm trong việc cung cấp dịch vụ y tế. Điều này giúp tạo ra một trải nghiệm trực tuyến tích cực và tạo niềm tin từ phía người sử dụng.

Tìm và đặt lịch bác sĩ đáng tin cậy

Khám phá các bác sĩ đáng tin cậy gần bạn và đặt lịch hẹn chỉ bằng vài cú nhấp chuột.

Đặt lịch hẹn →



Chọn chuyên ngành

Duyệt qua các chuyên khoa để tìm bác sĩ phù hợp và đặt lịch hẹn dễ dàng.



Bác sĩ đa khoa



Bác sĩ phụ khoa



Bác sĩ da liễu



Bác sĩ nhi khoa



Bác sĩ thần kinh



Bác sĩ tiêu hóa

Các bác sĩ hàng đầu hiện có

Khám phá và kết nối với đội ngũ bác sĩ giỏi, lịch trình rõ ràng, hỗ trợ đặt hẹn nhanh chóng và hiệu quả.



TS. Nguyễn Minh An

Bác sĩ đa khoa

📍 123 Lê Lợi

Đặt Ngay



BS. Trần Thị Hồng

Bác sĩ phụ khoa

📍 78 Trần Hưng Đạo, Quận 5

Đặt Ngay



PGS. Lê Văn Bình

Bác sĩ da liễu

📍 78 Trần Hưng Đạo, Quận Gò Vấp

Đặt Ngay



BS. Phạm Thị Linh

Bác sĩ nhi khoa

📍 20 Phạm Văn Đồng

Đặt Ngay



TS. Huỳnh Thị Mai

Bác sĩ thần kinh

📍 12 Võ Thị Sáu

Đặt Ngay



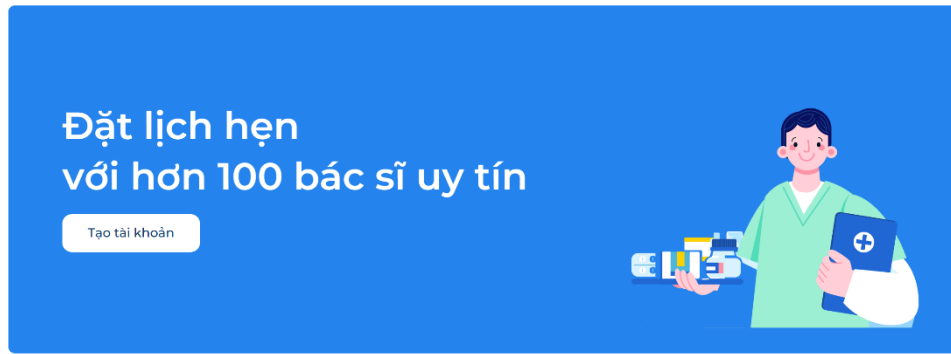
BS. Vũ Minh Khoa

Bác sĩ tiêu hóa

📍 94 Thanh Lương 12

Đặt Ngay

Xem thêm



DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ
Về chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ

0966 205 011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.1: Giao diện trang chủ DoctorGo

4.1.2. Giao diện đặt lịch khám bệnh

Giao diện giúp người dùng dễ dàng tra cứu thông tin về các bác sĩ, chi phí khám bệnh, và đặt lịch hẹn một cách thuận lợi. Giao diện chính của ứng dụng cung cấp một cái nhìn tổng quan về các bác sĩ và các khoa chuyên môn.

DoctorGo

TRANG CHỦTẤT CẢ BÁC SĨVỀ CHÚNG TÔI LIÊN HỆ

Hộp tác



Phó Giáo sư - Tiến sĩ TS. Nguyễn Minh An

4 năm kinh nghiệm

Chuyên khoaBác sĩ đa khoa

Nơi khám123 Lê Lợi - Đà Nẵng

Giá khám120.000 VNĐ

Đặt khám ngay

Thứ 4 - 26/11

Thứ 5 - 27/11

Thứ 6 - 28/11

Thứ 7 - 29/11

10:00 - 10:30

11:00 - 11:30

12:30 - 13:00

13:00 - 13:30

13:30 - 14:00

Giới thiệu

Hơn 4 năm kinh nghiệm khám và điều trị bệnh nội tổng quát, tận tâm và chuyên nghiệp.

Hỗ trợ đặt khám
1900-2805

ĐẶT KHÁM NGAY

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ
Về chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ

0966 205 011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.2: Giao diện đặt lịch khám bệnh

4.1.3. Giao diện nhập thông tin đặt lịch khám

Giao diện cho phép người dùng nhập thông tin các nhân để đặt lịch khám bệnh

Sinh viên thực hiện: Nguyễn Thắng Lợi Hướng dẫn: TS Nguyễn Văn Minh

47

DoctorGo

TRANG CHỦTẤT CẢ BÁC SĨVỀ CHÚNG TÔI LIÊN HỆ

Hợp tác

Hồ sơ bệnh nhân

Thăng LợiTối11/11/2002

Thông tin bổ sung (không bắt buộc)

Triệu chứng, thuốc đang dùng, tiền sử, ...

Thông tin đặt khám

TS. Nguyễn Minh An
123 Lê Lợi - Đà Nẵng

Ngày khám

29/11/2025

Khung giờ

10:00 - 10:30

Bệnh nhân

Thăng Lợi

Đặt lịch

Hủy

Bằng cách nhấn nút xác nhận, bạn đã đồng ý với các điều khoản và điều kiện đặt khám

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ
Về chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ

0966.205.011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.3: Giao diện nhập thông tin lịch khám

4.1.4. Giao diện lịch hẹn

Đây là giao diện lịch hẹn và đợi xác nhận bên bác sĩ

TRANG CHỦ

TẤT CẢ BÁC SĨ

VỀ CHÚNG TÔI

LIÊN HỆ

Hộp tác

Lịch khám của tôi

	TS. Nguyễn Minh An - Bác sĩ đa khoa Ngày và giờ: 27 Tháng 11 2025 - 10:00 Địa chỉ: 123 Lê Lợi, Đà Nẵng	Thanh toán Hủy cuộc hẹn
	TS. Nguyễn Minh An - Bác sĩ đa khoa Ngày và giờ: 29 Tháng 11 2025 - 10:30 Địa chỉ: 123 Lê Lợi, Quận 1, Đà Nẵng	Cuộc hẹn đã bị hủy
	TS. Nguyễn Minh Anh - Bác sĩ đa khoa Ngày và giờ: 18 Tháng 11 2025 - 11:00 Địa chỉ: 123 Lê Lợi, Quận 1, Đà Nẵng	Cuộc hẹn đã bị hủy
	TS. Nguyễn Minh Anh - Bác sĩ đa khoa Ngày và giờ: 17 Tháng 11 2025 - 10:30 Địa chỉ: 123 Lê Lợi, Quận 1, Đà Nẵng	Cuộc hẹn đã bị hủy
	TS. Nguyễn Minh An - Bác sĩ đa khoa Ngày và giờ: 15 Tháng 11 2025 - 11:00 Địa chỉ: 123 Lê Lợi, Quận 1, TP Hồ Chí Minh	Đã hoàn thành
	TS. Nguyễn Minh An - Bác sĩ đa khoa Ngày và giờ: 15 Tháng 11 2025 - 10:30 Địa chỉ: 123 Lê Lợi, Quận 1, TP Hồ Chí Minh	Cuộc hẹn đã bị hủy
	BS. Trần Thị Hồng - Bác sĩ phụ khoa Ngày và giờ: 13 Tháng 11 2025 - 11:00 Địa chỉ: 78 Trần Hưng Đạo, Quận 5, TP Hồ Chí Minh	Đã thanh toán Hủy cuộc hẹn
	BS. Trần Thị Hồng - Bác sĩ phụ khoa Ngày và giờ: 13 Tháng 11 2025 - 10:30 Địa chỉ: 78 Trần Hưng Đạo, Quận 5, TP Hồ Chí Minh	Thanh toán Hủy cuộc hẹn
	BS. Trần Thị Hồng - Bác sĩ phụ khoa Ngày và giờ: 10 Tháng 11 2025 - 11:00 Địa chỉ: 78 Trần Hưng Đạo, Quận 5, TP Hồ Chí Minh	Thanh toán Hủy cuộc hẹn

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ
Về chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ

0966 205 011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.4: Giao diện lịch hẹn

4.1.5. Giao diện thanh toán lịch khám online

Đây là giao diện thanh toán lịch khám online tích hợp cổng thanh toán Stripe

The image shows a payment interface for a doctor appointment. It is divided into two main sections: a confirmation screen on the left and a payment form on the right.

Left Panel (Confirmation):

- Header: "New business sandbox" with a "Sandbox" button.
- Section: "Thanh toán" (Payment).
- Amount: "₫450,000".
- Text: "Lịch hẹn với bác sĩ BS. Trần Thị Hồng" (Appointment with doctor Dr. Trần Thị Hồng).

Right Panel (Payment Form):

- Header: "Pay with link" button.
- Separator: "OR".
- Email field: "email@example.com".
- Section: "Payment method".
- Card information section:
 - Card number: "1234 1234 1234 1234".
 - Expiry: "MM / YY".
 - CVC: "CVC".
 - Cardholder name: "Full name on card".
 - Country or region: "Vietnam".
- Checkbox: "Save my information for faster checkout".
- Text: "Pay securely at New business sandbox and everywhere Link is accepted."
- Pay button: "Pay".
- Footer: "Powered by stripe | Terms Privacy".

Hình 4.5: Giao diện thanh toán lịch hẹn

4.1.6. Giao diện đăng ký trở thành bác sĩ

Đây là giao diện đăng ký trở thành bác sĩ, người dùng cần điền những thông tin cần thiết để trở thành bác sĩ cho admin xét duyệt

THÔNG TIN LIÊN HỆ HỢP TÁC

Tên	Chuyên khoa
Nhập tên	Bác sĩ đa khoa
Email	Trình độ chuyên môn
Email	Bác sĩ
Kinh nghiệm	Địa chỉ
1 năm	Nhập số nhà, Phường
Giá	Nhập hoặc chọn tỉnh / thành
Giá VNĐ	
Giới thiệu	
Nhập giới thiệu ngắn về bác sĩ	
Tài liệu xác minh	
 Tải lên giấy tờ hỗ trợ xác minh thông tin của bạn	
Ảnh chứng chỉ	
 Tải lên chứng chỉ hành nghề hoặc bằng cấp liên quan	
LƯU Ý Kết quả sẽ trả về khi được người quản trị xét duyệt và thông báo kết quả trong mục Hồ sơ của người dùng.	
Cử ý cầu	

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ
Vé chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ

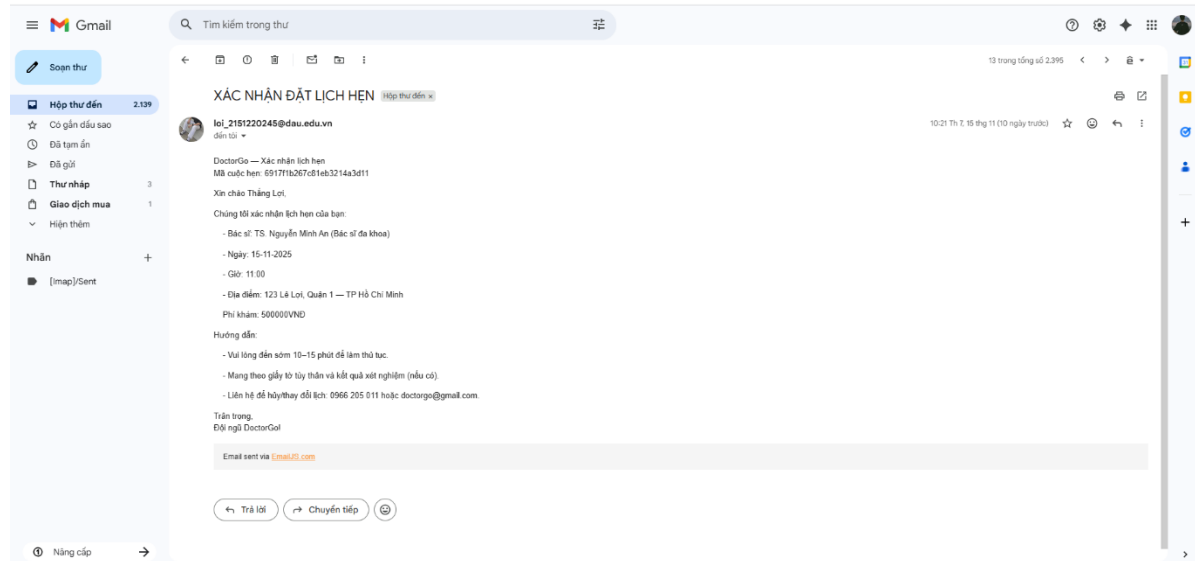
0966 205 011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.6: Giao diện đăng ký trở thành bác sĩ

4.1.7. Giao diện thông báo xác nhận lịch hẹn qua gmail

Đây là giao diện nhận thông tin qua gmail nó sẽ hiển thị bao gồm tên bác sĩ, giờ bạn đặt và địa chỉ giá khám cụ thể.



Hình 4.7: Giao diện thông báo xác nhận lịch hẹn qua gmail

4.1.8. Giao diện đăng nhập

DoctorGo

TRANG CHỦTẤT CẢ BÁC SĨVỀ CHÚNG TỒILIÊN HỆ

Hợp tácĐăng nhập

Đăng nhập

Vui lòng đăng nhập để đặt lịch hẹn.

Email

Mật khẩu

Đăng nhập

Tạo tài khoản mới? [Bấm vào đây](#)

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ

Về chúng tôi

Liên hệ

Chính sách bảo mật

LIÊN HỆ

0966 205 011

doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.8: Giao diện đăng nhập

4.1.9. Giao diện đăng ký

DoctorGo

[TRANG CHỦ](#) [TẤT CẢ BÁC SĨ](#) [VỀ CHÚNG TÔI](#) [LIÊN HỆ](#)

Hợp tác

Đăng nhập

Tạo tài khoản
Vui lòng đăng ký để đặt lịch hẹn.

Tên đăng nhập

Email

Mật khẩu

Tạo tài khoản

Bạn đã có tài khoản? [Đăng nhập ngay](#)

DoctorGo
Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY
Trang chủ
Về chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ
0966 205 011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.9: Giao diện đăng ký

4.1.10. Giao diện hồ sơ bệnh nhân

DoctorGo

TRANG CHỦTẤT CẢ BÁC SĨVỀ CHÚNG TÔI LIÊN HỆ

Hợp tác

Thăng Lợi

THÔNG TIN LIÊN HỆ

Email

Điện thoại

Địa chỉ

Mật khẩu

huynhvanhungnp123@gmail.com

0966205011

94 Thanh Lương 16
Hải Phòng

THÔNG TIN CƠ BẢN

Giới tính

Ngày sinh

Nữ

2002-11-11

Chỉnh sửa

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ
Về chúng tôi
Liên hệ
Chính sách bảo mật

LIÊN HỆ

0966 205 011
doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.10: Giao diện hồ sơ bệnh nhân

4.1.11. Giao diện thay đổi thông tin bệnh nhân

DoctorGo

TRANG CHỦTẤT CẢ BÁC SĨVÉ CHÚNG TÔI LIÊN HỆ

Hộp tác



Thăng Lợi

THÔNG TIN LIÊN HỆ

Email

huynhvanhungnp123@gmail.com

Điện thoại

0966205011

Địa chỉ

94 Thanh Lương 16

Hải Phòng

Mật khẩu

Mật khẩu cũ

Nhập mật khẩu hiện tại

Mật khẩu mới

Nhập mật khẩu mới

THÔNG TIN CƠ BẢN

Giới tính

Nữ

Ngày sinh

11/11/2002

Lưu thay đổi

DoctorGo

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ

Vé chúng tôi

Liên hệ

Chính sách bảo mật

LIÊN HỆ

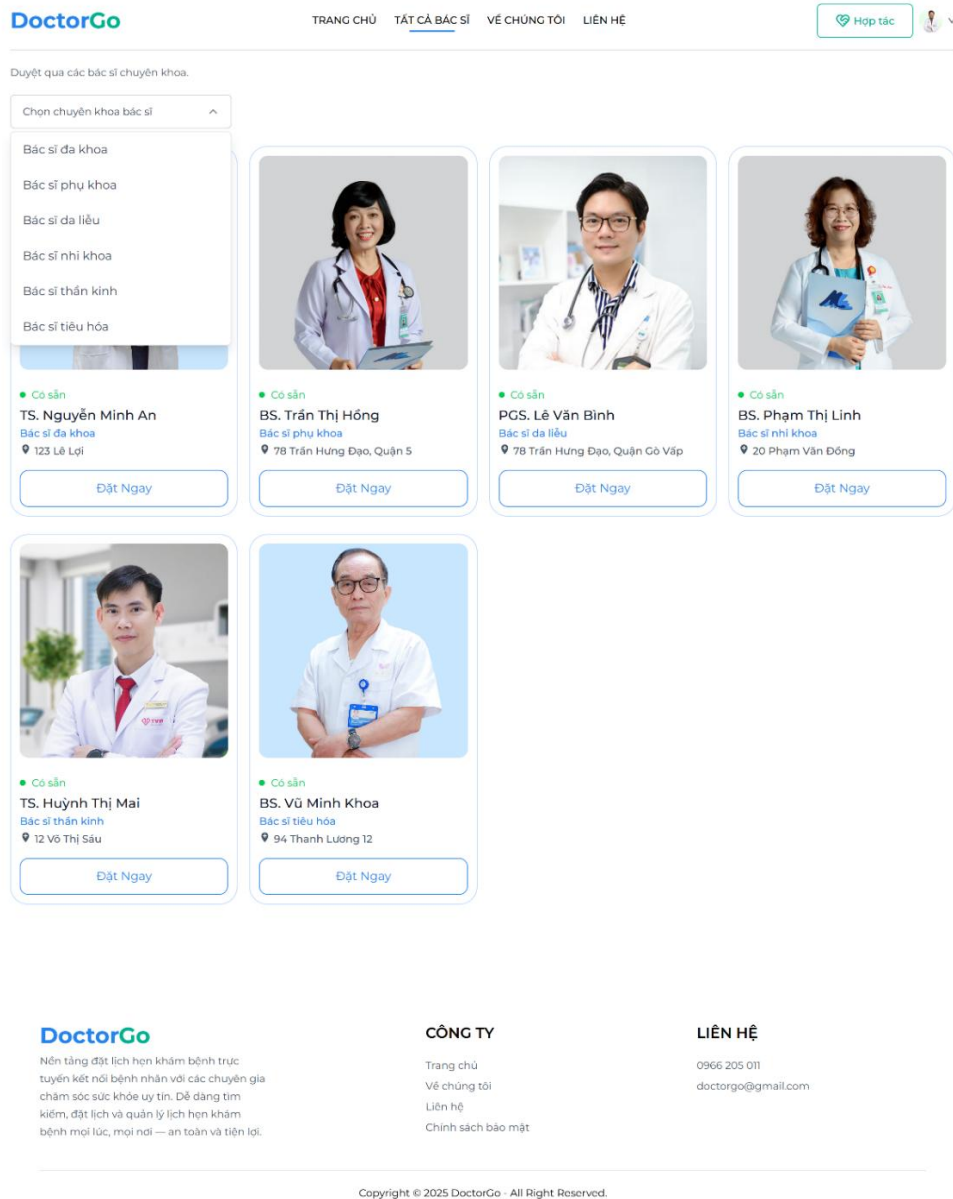
0966 205 011

doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.11: Giao diện thay đổi thông tin bệnh nhân

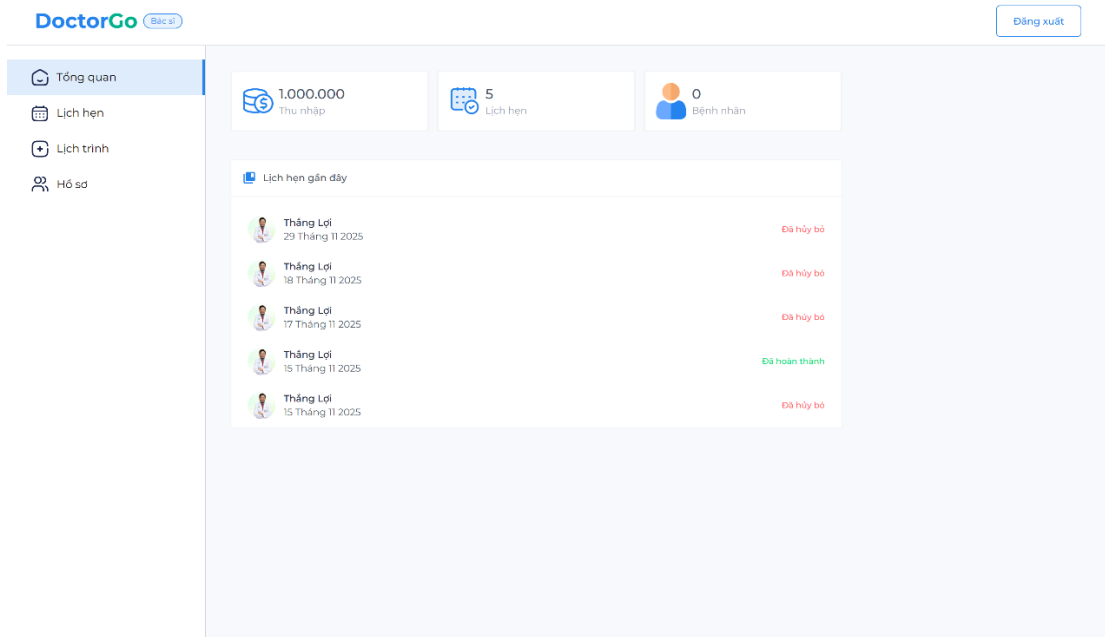
4.1.12. Giao diện danh sách bác sĩ



Hình 4.12: Giao diện danh sách bác sĩ

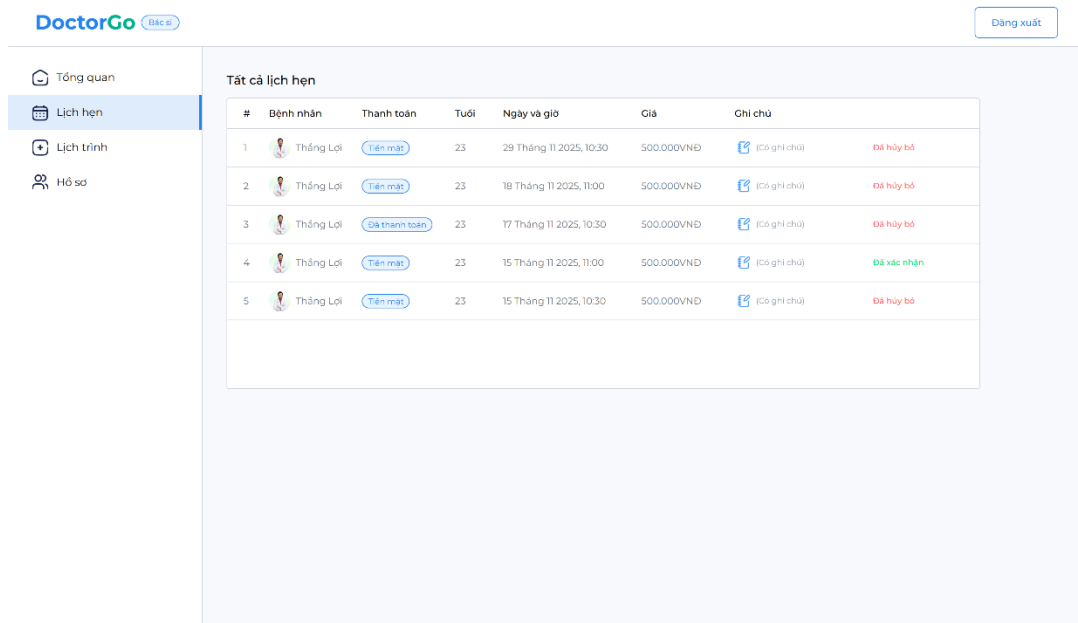
4.2. Giao diện bác sĩ

4.2.1. Giao diện tổng quan bác sĩ



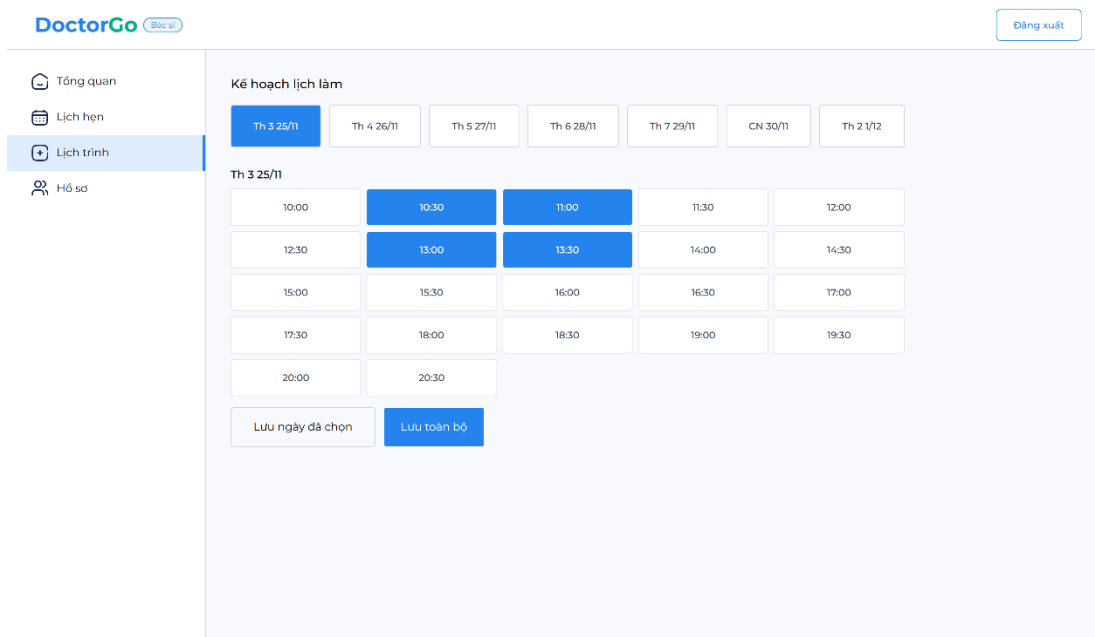
Hình 4.13: Giao diện tổng quan bác sĩ

4.2.2. Giao diện lịch hẹn của bác sĩ



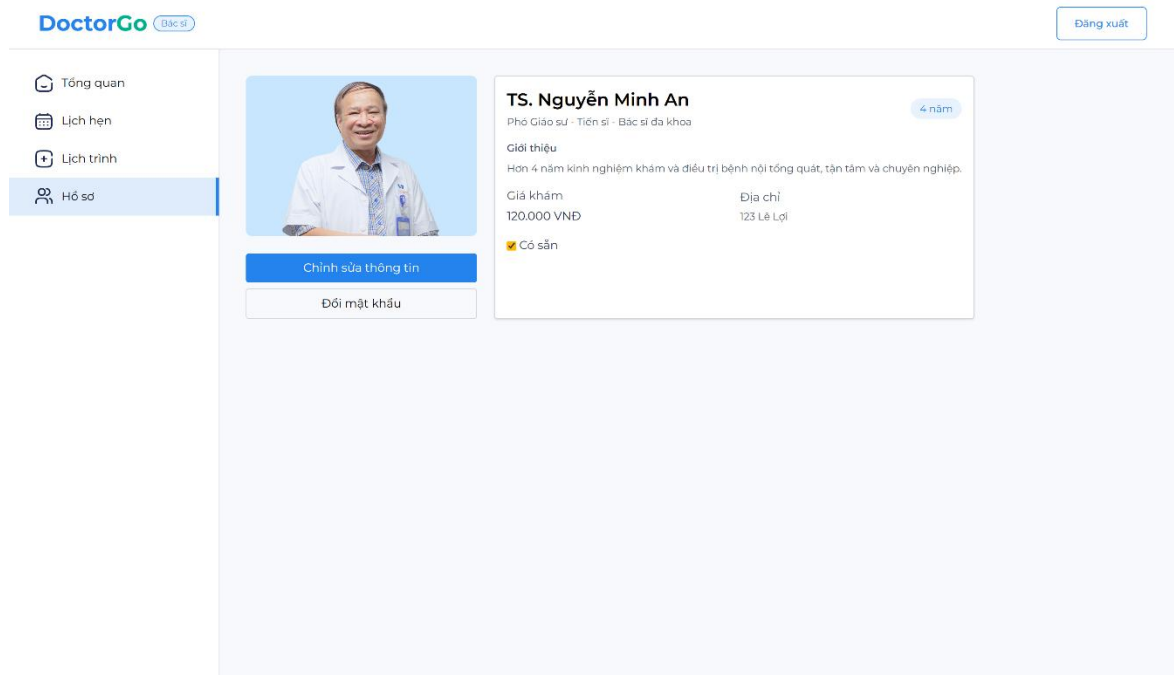
Hình 4.14: Giao diện lịch hẹn của bác sĩ

4.2.3. Giao diện lịch trình bác sĩ



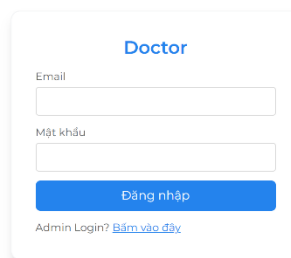
Hình 4.15: Giao diện lịch trình bác sĩ

4.2.4. Giao diện hồ sơ bác sĩ



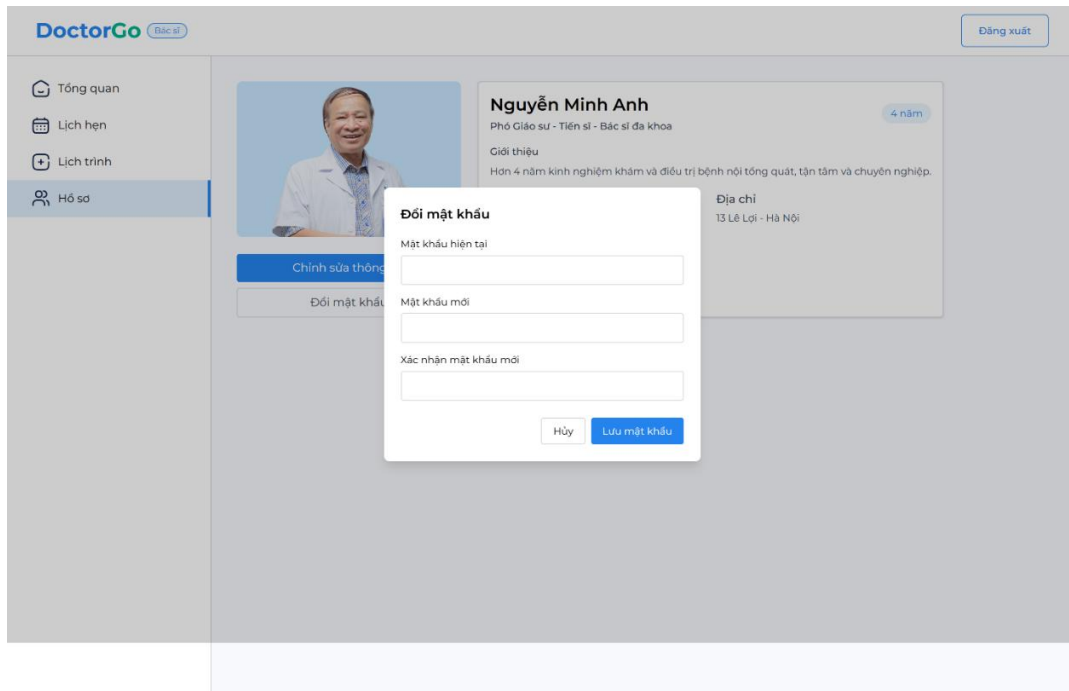
Hình 4.16: Giao diện hồ sơ bác sĩ

4.2.5. Giao diện đăng nhập bác sĩ



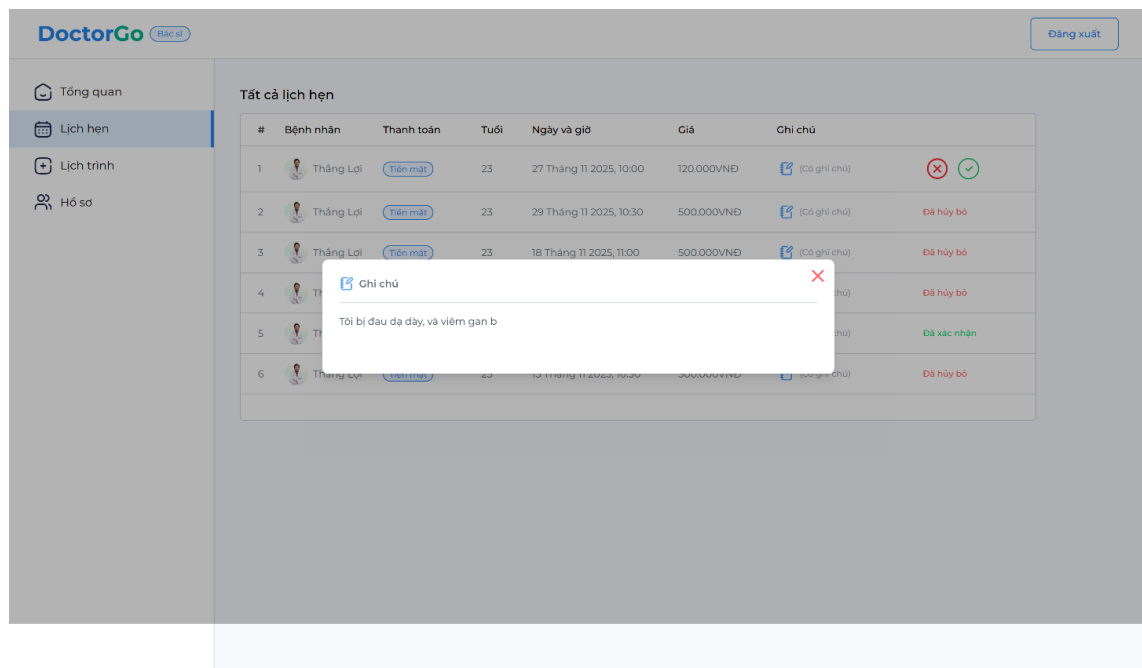
Hình 4.17: Giao diện đăng nhập bác sĩ

4.2.6. Giao diện thay đổi mật khẩu bác sĩ



Hình 4.18: Giao diện thay đổi mật khẩu bác sĩ

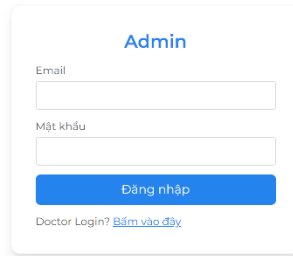
4.2.7. Giao diện thông tin bệnh nền gửi cho bác sĩ



Hình 4.19: Giao diện thông tin bệnh nền gửi cho bác sĩ

4.3. Giao diện admin

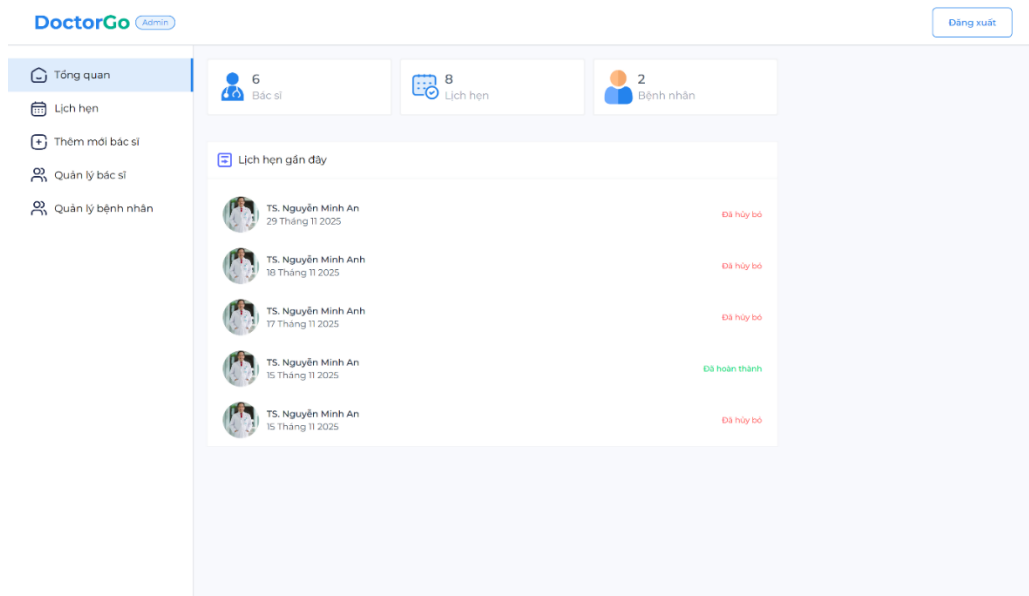
4.3.1. Giao diện đăng nhập Admin



The image shows a login form for an Admin user. It has a title 'Admin' in blue. Below it are two input fields: 'Email' and 'Mật khẩu' (Password). A blue button labeled 'Đăng nhập' (Login) is positioned below the password field. At the bottom, there is a link 'Doctor Login? [Bấm vào đây](#)'.

Hình 4.20: Giao diện đăng nhập Admin

4.3.2. Giao diện tổng quan Admin

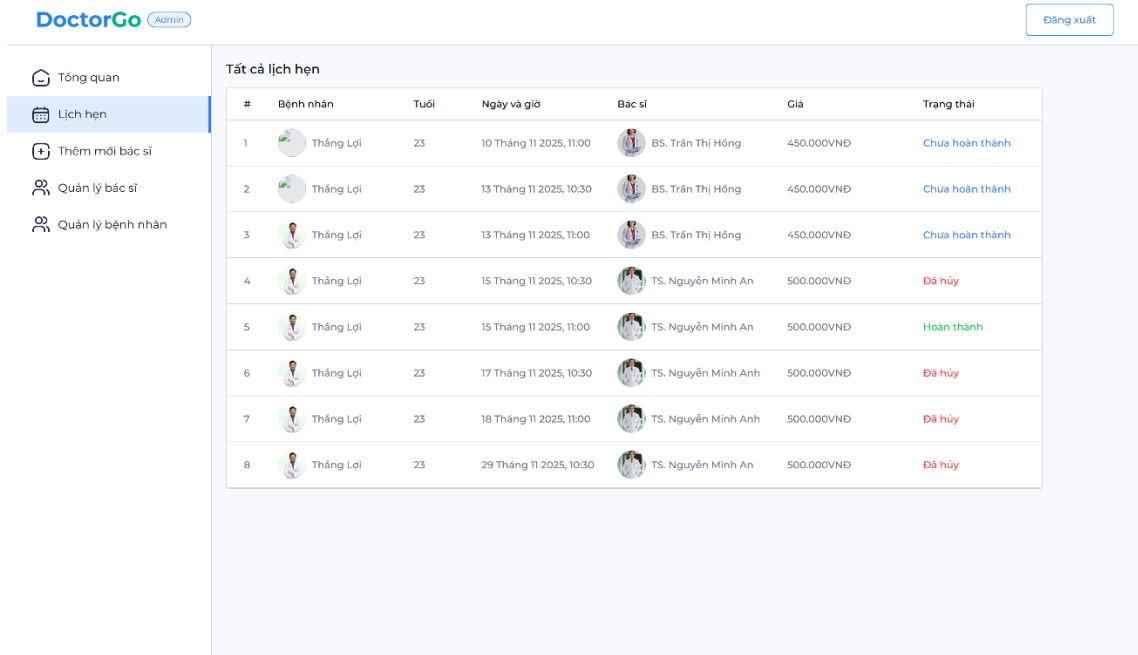


The image displays the Admin dashboard overview. At the top left is the 'DoctorGo Admin' logo, and at the top right is a 'Đăng xuất' (Logout) button. The dashboard features three summary cards: '6 Bác sĩ' (Doctors), '8 Lịch hẹn' (Appointments), and '2 Bệnh nhân' (Patients). Below these is a section titled 'Lịch hẹn gần đây' (Recent appointments) containing a list of five appointments. Each entry includes a doctor's profile picture, name, and appointment date, followed by a status indicator.

Doctor	Date	Status
TS. Nguyễn Minh An	29 Tháng 11 2025	Đã hủy bỏ
TS. Nguyễn Minh An	18 Tháng 11 2025	Đã hủy bỏ
TS. Nguyễn Minh An	17 Tháng 11 2025	Đã hủy bỏ
TS. Nguyễn Minh An	15 Tháng 11 2025	Đã hoàn thành
TS. Nguyễn Minh An	15 Tháng 11 2025	Đã hủy bỏ

Hình 4.21: Giao diện tổng quan Admin

4.3.3. Giao diện lịch hẹn Admin



Hình 4.22: Giao diện lịch hẹn Admin

4.3.4. Giao diện thêm mới bác sĩ

Thêm bác sĩ

Tải lên bác sĩ hình ảnh

Tên bác sĩ:

Chuyên khoa:

Email:

Trình độ chuyên môn:

Mật khẩu:

Địa chỉ:

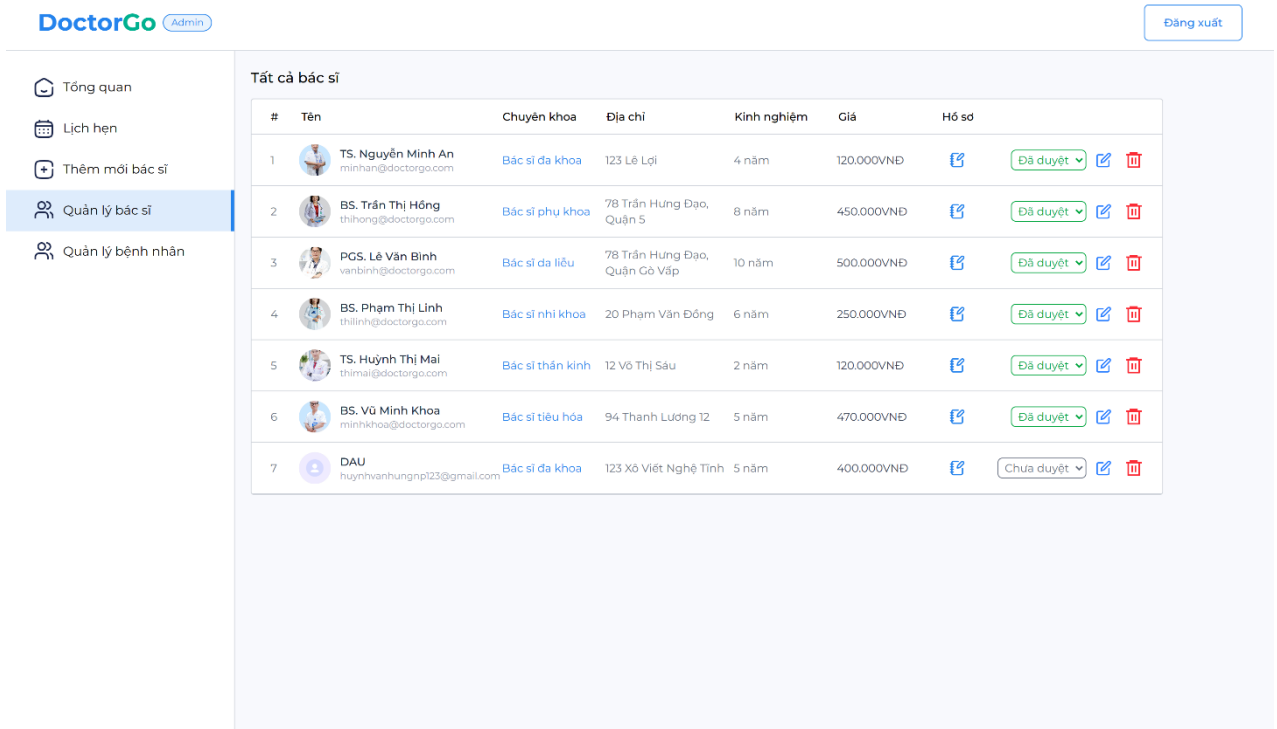
Kinh nghiệm:

Giá:

Giới thiệu:

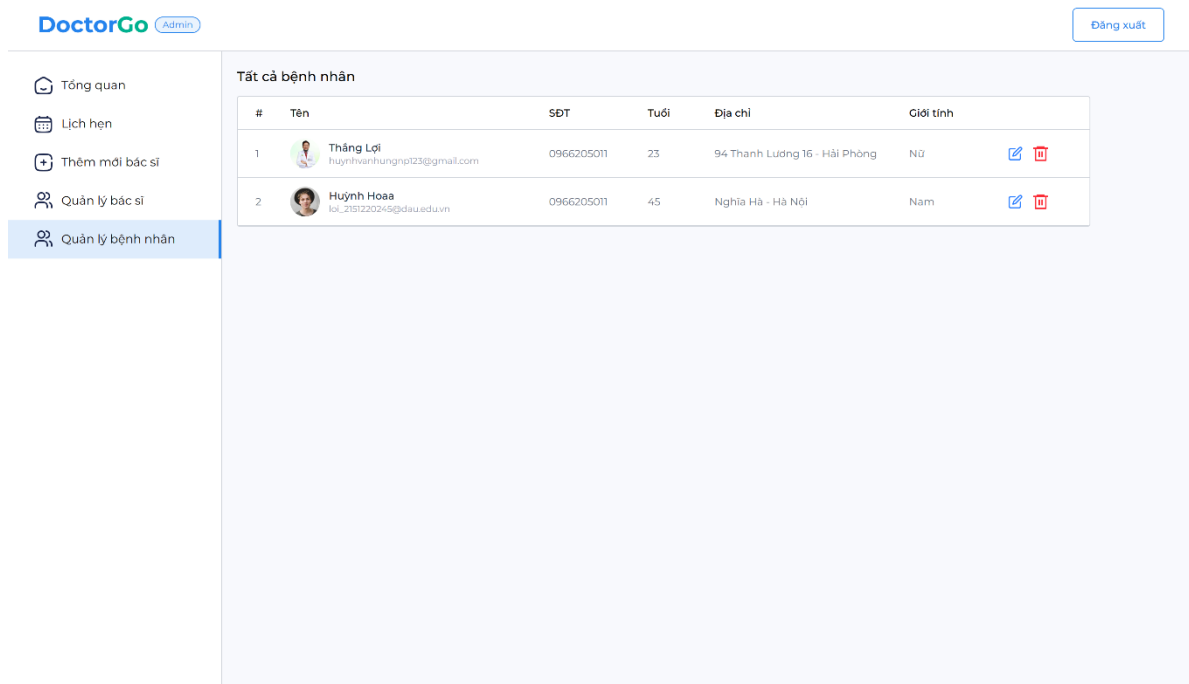
Hình 4.23: Giao diện thêm mới bác sĩ

4.3.5. Giao diện quản lý bác sĩ



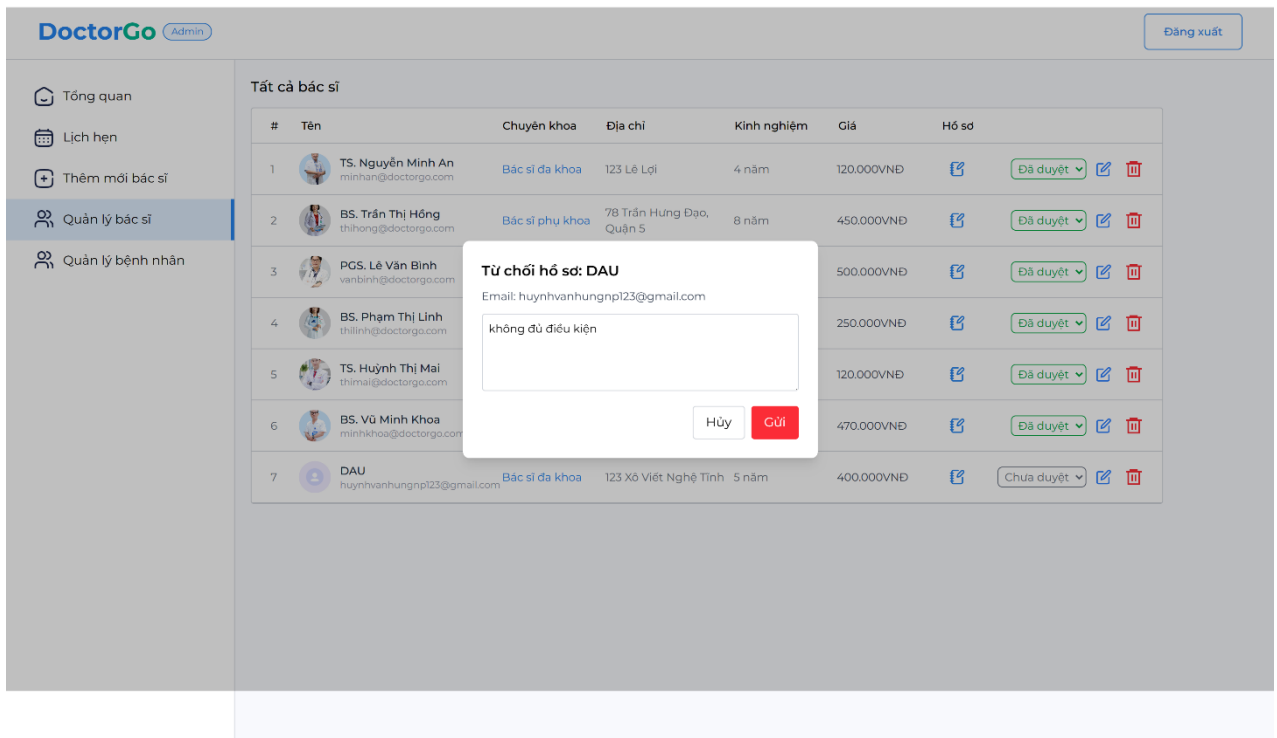
Hình 4.24: Giao diện quản lý bác sĩ

4.3.6. Giao diện quản lý thông tin bệnh nhân



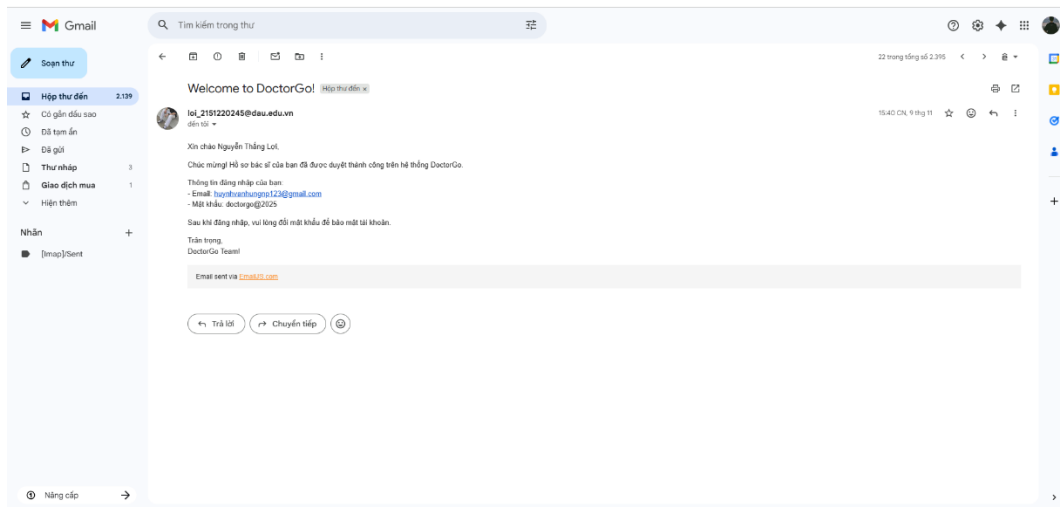
Hình 4.25: Giao diện quản lý thông tin bệnh nhân

4.3.7. Giao diện quản lý từ chối đăng ký trở thành bác sĩ



Hình 4.26: Giao diện quản lý từ chối đăng ký trở thành bác sĩ

4.3.8. Giao diện chấp nhận đăng ký trở thành bác sĩ và gửi qua gmail



Hình 4.27: Giao diện chấp nhận đăng ký trở thành bác sĩ và gửi qua gmail

4.3.9. Giao diện thông báo từ chối đăng ký làm bác sĩ

TRANG CHỦ

TẤT CẢ BÁC SĨ

VỀ CHÚNG TÔI

LIÊN HỆ

Hợp tác

Trạng thái hồ sơ: Đã bị từ chối

Lý do: không đủ điều kiện

Thăng Lợi

THÔNG TIN LIÊN HỆ

Email

huynhvanhungnp123@gmail.com

Điện thoại

0966205011

Địa chỉ

94 Thanh Lương 16
Hải Phòng

Mật khẩu

THÔNG TIN CƠ BẢN

Giới tính

Nữ

Ngày sinh

2002-11-11

Chỉnh sửa

Nền tảng đặt lịch hẹn khám bệnh trực tuyến kết nối bệnh nhân với các chuyên gia chăm sóc sức khỏe uy tín. Dễ dàng tìm kiếm, đặt lịch và quản lý lịch hẹn khám bệnh mọi lúc, mọi nơi — an toàn và tiện lợi.

CÔNG TY

Trang chủ

Về chúng tôi

Liên hệ

Chính sách bảo mật

LIÊN HỆ

0966 205 011

doctorgo@gmail.com

Copyright © 2025 DoctorGo - All Right Reserved.

Hình 4.28: Giao diện thông báo từ chối đăng ký làm bác sĩ

4.3.10. Giao diện admin chỉnh sửa thông tin bác sĩ

Admin

Đăng xuất

Tổng quan

Lịch hẹn

Thêm mới bác sĩ

Quản lý bác sĩ

Quản lý bệnh nhân

Tất cả bác sĩ

Chỉnh sửa bác sĩ

Họ tên

BS. Vũ Minh Khoa

Email

minhkhoea@doctorgo.com

Chuyên khoa

Bác sĩ tiêu hóa

Học vị

Bác sĩ

Kinh nghiệm

5 năm

Giá khám (VNĐ)

470.000

Giới thiệu

Chuyên về bệnh dạ dày, đại tràng và chăm sóc dinh dưỡng tiêu hóa.

Địa chỉ

94 Thanh Lương 12

Tỉnh / Thành

Đà Nẵng

Hình 4.29: Giao diện admin chỉnh sửa thông tin bác sĩ

Sinh viên thực hiện: Nguyễn Thăng Lợi Hướng dẫn: TS Nguyễn Văn Minh

65

4.3.11. Giao diện admin chỉnh sửa thông tin bệnh nhân

Chỉnh sửa thông tin bệnh nhân

Tên:

Số điện thoại:

Ngày sinh:

Giới tính:

Địa chỉ:

Hải Phòng

Hình 4.30: Giao diện admin chỉnh sửa thông tin bệnh nhân

4.3.12. Giao diện hồ sơ đăng ký bác sĩ

Hồ sơ: DAU

#	Tên	Chuyên khoa	Địa chỉ	Kinh nghiệm	Giá	Hồ sơ
1	TS. Nguyễn Minh An minhanh@doctorgo.com	Bác sĩ đa khoa	123 Lê Lợi	4 năm	120.000VNĐ	Đã duyệt Chỉnh sửa Xóa
2	BS. Trần Thị Hồng thihong@doctorgo.com	Bác sĩ phụ khoa	78 Trần Hưng Đạo, Quận 5	8 năm	450.000VNĐ	Đã duyệt Chỉnh sửa Xóa
3						Đã duyệt Chỉnh sửa Xóa
4						Đã duyệt Chỉnh sửa Xóa
5						Đã duyệt Chỉnh sửa Xóa
6						Đã duyệt Chỉnh sửa Xóa
7	DAU huynhvanhungnpl23@gmail.com	Bác sĩ đa khoa	123 Xã Viết Nghệ Tĩnh	5 năm	400.000VNĐ	Chưa duyệt Chỉnh sửa Xóa

Hình 4.31: Giao diện hồ sơ đăng ký bác sĩ

KẾT LUẬN

1. Kết quả đạt được

Thành công trong quy trình đặt lịch: Hệ thống đặt lịch trực tuyến giúp bệnh nhân dễ dàng và nhanh chóng đặt lịch khám mọi lúc, mọi nơi, giảm đáng kể thời gian và công sức so với việc đến trực tiếp tại phòng khám.

Tối ưu hóa lịch trình: Hệ thống quản lý lịch hẹn tích hợp giúp tối ưu hóa lịch trình của các bác sĩ và cơ sở y tế. Điều này giảm nguy cơ chồng chéo lịch và tăng cường hiệu suất làm việc của nhân viên y tế.

Giao diện thân thiện và dễ sử dụng: Ứng dụng được phát triển với giao diện người dùng thân thiện và dễ sử dụng, giúp người bệnh dễ dàng đặt lịch, tra cứu thông tin và tương tác thuận tiện.

Thuận lợi và tiết kiệm thời gian cho bệnh nhân: Người bệnh có thể dễ dàng đặt lịch trực tuyến mọi lúc, mọi nơi, giảm gánh nặng thời gian và công sức.

Quản lý lịch hẹn hiệu quả cho cơ sở y tế: Cơ sở y tế có thể tối ưu hóa lịch trình làm việc và giảm thời gian chờ đợi, tăng cường hiệu suất làm việc của nhân viên y tế.

Định hình cơ sở lý thuyết: Góp phần vào việc xây dựng cơ sở lý thuyết về việc tích hợp công nghệ trong quy trình chăm sóc sức khỏe và quản lý lịch hẹn.

2. Hướng phát triển

Tối ưu hóa mô hình chức năng: Tổ chức và mô tả chức năng hệ thống cần được làm rõ và đồng nhất hơn để tăng khả năng hiểu và quản lý dự án.

Kiểm thử và tối ưu hóa hiệu suất: Tăng cường quá trình kiểm thử để đảm bảo tính ổn định và độ tin cậy của hệ thống, cũng như tối ưu hóa hiệu suất để đáp ứng nhu cầu người dùng.

Phát triển giao diện người dùng (UI/UX): Tăng cường trải nghiệm người dùng bằng cách phát triển giao diện thân thiện, dễ sử dụng, và tương tác.

Hỗ trợ ngôn ngữ và quốc gia: Mở rộng khả năng hỗ trợ đa ngôn ngữ và quốc gia để mở rộng sự tiếp cận và sử dụng hệ thống trên quy mô toàn cầu.

Kế hoạch triển khai và duy trì: Xây dựng một kế hoạch triển khai chi tiết và quy

trình duy trì để đảm bảo sự ổn định và tiếp tục cải tiến sau khi hệ thống đã đi vào hoạt động.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Anh

- [1] HTML To React: The Ultimate Guide - <https://www.scribd.com/document/552527408/HTML-to-React-the-Ultimate-Guide-PDF-1>
- [2] React Design Partterns and best practice - https://omm.org.mx/wp-content/uploads/2020/07/9781786464538-REACT_DESIGN_PATTERNS_AND_BEST_PRACTICES.pdf
- [3] NodeJS Web Development - <https://nodejs.org/en>
- [4] JavaScript Tutorial - <https://www.w3schools.com/js/>
- [5] Tailwind document - <https://v2.tailwindcss.com/docs>
- [6] EmailJS - <https://www.emailjs.com/docs/>
- [7] ExpressJS team. Routing - <https://expressjs.com/en/guide/routing.html>

Tài liệu Tiếng Việt

- [8] Tài liệu học ReactJS bằng tiếng việt - <https://vntalking.com/tai-lieu-hoc-reactjs-tieng-viet>
- [9] Hỏi dân IT <https://www.youtube.com/@hoidanit>.
- [10] Sách lập trình nodejs thật đơn giản - <https://vntalking.com/wp-content/uploads/2020/04/sach-lap-trinh-nodejs-that-don-gian-sample.pdf>
- [11] Giới thiệu về ExpressJS - <https://hiepsiit.com/detail/nodejs/node-js/express-framework>
- [12] 12+ App khám bệnh online phổ biến nhất hiện nay - <https://www.mediplus.vn/cong-nghe-cho-y-te/ung-dung-app-kham-benh-online.html>