

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 ỨNG DỤNG QUẢN LÝ HỒ SƠ BỆNH
NHÂN VÀ DỊCH VỤ Y TẾ TẠI BỆNH VIỆN
PHÚC AN**

Người hướng dẫn	: ThS. PHẠM THỊ DUNG
Sinh viên thực hiện	: NGUYỄN NGỌC ANH TÚ
Mã số sinh viên	: 2151220086
Lớp	: 21CT2

Đà Nẵng, 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 ỨNG DỤNG QUẢN LÝ HỒ SƠ BỆNH
NHÂN VÀ DỊCH VỤ Y TẾ TẠI BỆNH VIỆN
PHÚC AN**

Người hướng dẫn	: ThS. PHẠM THỊ DUNG
Sinh viên thực hiện	: NGUYỄN NGỌC ANH TÚ
Mã số sinh viên	: 2151220086
Lớp	: 21CT2

Đà Nẵng, 12/2025

TÓM TẮT

Tên đề tài: Xây dựng ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại bệnh viện PHÚC AN.

Sinh viên thực hiện: Nguyễn Ngọc Anh Tú.

MSSV: 2151220086 - Lớp: 21CT2

Trong bối cảnh chuyển đổi số ngành y tế, việc quản lý hồ sơ bệnh nhân và dịch vụ y tế bằng phương pháp thủ công hoặc các hệ thống rời rạc gây ra nhiều hạn chế như trùng lặp dữ liệu, khó tra cứu thông tin, khó theo dõi lịch sử khám chữa bệnh và khó tối ưu quy trình làm việc. Đề tài “*Xây dựng ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại bệnh viện PHÚC AN*” được thực hiện nhằm xây dựng một hệ thống phần mềm hỗ trợ bệnh viện quản lý tập trung, chính xác và hiệu quả các thông tin liên quan đến bệnh nhân, bác sĩ, khoa phòng, dịch vụ y tế và lịch hẹn khám chữa bệnh.

Ứng dụng được thiết kế với các phân hệ chính: quản lý hồ sơ bệnh nhân (thông tin cá nhân, tiền sử bệnh, lịch sử khám chữa bệnh), quản lý dịch vụ y tế (danh mục dịch vụ, giá, khoa phụ trách), quản lý lịch hẹn và quá trình khám (đặt lịch, xác nhận, nhắc lịch qua email), quản lý đội ngũ bác sĩ và phân công theo chuyên khoa. Bên cạnh chức năng dành cho cán bộ y tế, hệ thống còn hỗ trợ người bệnh tra cứu thông tin dịch vụ, đặt lịch khám trực tuyến và nhận thông báo nhắc lịch, góp phần nâng cao trải nghiệm người dùng.

Về mặt kỹ thuật, đề tài áp dụng mô hình ứng dụng web với kiến trúc hiện đại, sử dụng framework PHP Laravel kết hợp cơ sở dữ liệu quan hệ để lưu trữ và truy vấn dữ liệu một cách an toàn, toàn vẹn và nhất quán. Hệ thống được xây dựng theo hướng phân quyền người dùng (quản trị viên, bác sĩ, bệnh nhân), đảm bảo chỉ những đối tượng có thẩm quyền mới được truy cập và thao tác trên các nhóm dữ liệu tương ứng. Đồng thời, các chức năng ghi log, gửi email nhắc lịch và xử lý tự động được tích hợp để hỗ trợ công tác vận hành.

Kết quả triển khai cho thấy ứng dụng giúp giảm thiểu đáng kể thời gian tìm kiếm hồ sơ, hạn chế sai sót trong nhập liệu, hỗ trợ bệnh viện quản lý dịch vụ y tế một cách rõ ràng, minh bạch và dễ mở rộng. Đối với bệnh nhân, hệ thống mang lại sự thuận tiện trong việc đặt lịch, theo dõi thông tin khám chữa bệnh và tương tác với bệnh viện. Đề tài góp phần xây dựng nền tảng ban đầu cho việc phát triển hệ thống quản lý bệnh viện số tại PHÚC AN.

LỜI NÓI ĐẦU

Để hoàn thành được Đồ án tốt nghiệp này em xin chân thành cảm ơn quý thầy cô trong khoa Công Nghệ Thông Tin - Trường Đại Học Kiến Trúc Đà Nẵng, đặc biệt là cô ThS. Phạm Thị Dung đã tận tình truyền đạt kiến thức trong những năm học vừa qua. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình làm đồ án tốt nghiệp mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Em cũng xin gửi lời cảm ơn đến Khoa và các thầy cô trong khoa đã luôn đồng hành và hỗ trợ em trong suốt quá trình học tập. Những kiến thức và kinh nghiệm mà các thầy cô truyền đạt đã là nền tảng vững chắc giúp em tự tin hơn khi bước vào môi trường thực tế.

Em không thể không nhắc đến sự ủng hộ và động viên từ gia đình và bạn bè. Chính nhờ sự hiện diện và sự khích lệ không ngừng của các bạn mà em đã vượt qua được những thử thách và khó khăn trong thời gian thực tập. Em vô cùng biết ơn sự đồng hành và sự động viên này.

Trong quá trình làm bài đồ án tốt nghiệp, do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp của quý báu của các thầy cô để em có thể học hỏi thêm được nhiều kinh nghiệm, kiến thức.

Em xin trân trọng cảm ơn!

CAM ĐOAN

Em xin cam đoan kết quả đạt được trong đồ án tốt nghiệp là sản phẩm của riêng cá nhân dưới sự hướng dẫn của ThS. Phạm Thị Dung, không sao chép lại của người khác. Trong toàn bộ nội dung của đồ án, những điều được trình bày hoặc là của bản thân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp. Em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Sinh viên thực hiện

Nguyễn Ngọc Anh Tú

MỤC LỤC

TÓM TẮT	i
LỜI NÓI ĐẦU	ii
CAM ĐOAN.....	iii
MỤC LỤC	iv
DANH SÁCH CÁC BẢNG	vi
DANH SÁCH HÌNH VẼ	vii
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	x
MỞ ĐẦU	1
Chương 1. CƠ SỞ LÝ THUYẾT	3
1.1. Giới thiệu công nghệ và công cụ sử dụng	3
1.1.1. <i>PHP</i>	3
1.1.2. <i>HTML</i>	4
1.1.3. <i>CSS</i>	5
1.1.4. <i>Javascript</i>	6
1.1.6. <i>Giới thiệu MySQL</i>	7
1.1.6. <i>Giới thiệu và cài đặt Xampp</i>	9
1.1.7. <i>Giới thiệu và cài đặt Visual studio code</i>	11
1.2. Framework Laravel.....	12
1.3. Đặc điểm và yêu cầu chung của hệ thống	16
1.4. Kết luận chương 1.....	17
Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	18
2.1. Phân tích yêu cầu	18
2.1.1. <i>Đặc tả yêu cầu chức năng của hệ thống</i>	18
2.1.2. <i>Yêu cầu phi chức năng</i>	19
2.2.1. <i>Kiến trúc hệ thống</i>	20
2.2.2. <i>Sơ đồ use-case</i>	22
2.2.3. <i>Sơ đồ tuần tự</i>	29
2.2.4. <i>Sơ đồ hoạt động</i>	40
2.2.5. <i>Thiết kế lược đồ dữ liệu MySQL</i>	50
2.3 Kết luận chương 2.....	59
Chương 3. CÀI ĐẶT VÀ THỰC NGHIỆM.....	60

3.1. Giao diện chức năng	60
3.1.1. Giao diện chức năng Admin.....	60
3.1.2. Giao diện chức năng bệnh nhân	66
3.1.3. Giao diện chức năng Bác sĩ.....	75
3.2. Cài đặt chương trình	78
3.2.1. Cài đặt phát triển mã nguồn admin	78
3.2.2. Cài đặt phát triển mã nguồn bệnh nhân.....	82
3.2.3. Cài đặt phát triển mã nguồn bác sĩ	84
3.4. Kết luận chương 3.....	86
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	87
TÀI LIỆU THAM KHẢO.....	88

DANH SÁCH CÁC BẢNG

Bảng 2.1. Mô tả các bảng cơ sở dữ liệu	51
Bảng 2.2. Mô tả bảng roles.....	51
Bảng 2.3. Mô tả bảng users	52
Bảng 2.4. Mô tả bảng patients	52
Bảng 2.5. Mô tả bảng doctor	53
Bảng 2.6. Mô tả bảng departments.....	54
Bảng 2.7. Mô tả bảng services.....	54
Bảng 2.8. Mô tả bảng service_symptoms.....	54
Bảng 2.9. Mô tả bảng appointments.....	55
Bảng 2.10. Mô tả bảng diseases	56
Bảng 2.11. Mô tả bảng diseases_symptoms.....	56
Bảng 2.12. Mô tả bảng medical_records.....	56
Bảng 2.13. Mô tả bảng lab_tests	57
Bảng 2.14. Mô tả bảng messages	58
Bảng 2.15. Mô tả bảng test_types	58

DANH SÁCH HÌNH VẼ

Hình 1.1. PHP là một dạng mã lệnh	4
Hình 1.2. Ngôn ngữ HTML.....	4
Hình 1.3. Ngôn ngữ CSS	5
Hình 1.4. Ngôn ngữ JavaScript	7
Hình 1.5. Cơ sở dữ liệu MySql.....	8
Hình 1.6. Cơ sở dữ liệu Xampp.....	9
Hình 1.7. Cài đặt môi trường Xampp	10
Hình 1.8. Cài đặt Visual Studio Code	12
Hình 2.1. Kiến trúc controller.....	20
Hình 2.2. Kiến trúc giao diện	21
Hình 2.3. Kiến model	21
Hình 2.4. Kiến trúc database	21
Hình 2.5. Use-case tổng quát.....	22
Hình 2.6. Use-case chức năng đăng nhập.....	23
Hình 2.7. Use-case chức năng đặt lịch và thanh toán.....	23
Hình 2.8. Use-case chức năng thiết lập thông tin của bệnh nhân.....	24
Hình 2.9. Use-case chức năng thiết lập thông tin Bác sĩ.....	24
Hình 2.10. Use-case chức năng thiết lập chuyên khoa.....	25
Hình 2.11. Use-case chức năng thiết lập dịch vụ	25
Hình 2.12. Use-case chức năng thiết lập bệnh	26
Hình 2.13. Use-case chức năng thiết lập loại xét nghiệm	26
Hình 2.14. Use-case chức năng chat box trực tiếp với Admin.....	27
Hình 2.15. Use-case chức năng xem lịch khám	27
Hình 2.16. Use-case chức năng tư vấn theo triệu chứng.....	28
Hình 2.17. Use-case chức năng gửi mail xác nhận lịch hẹn.....	28
Hình 2.18. Sơ đồ tuần tự chức năng đăng nhập	29
Hình 2.19. Sơ đồ tuần tự chức năng đặt lịch và thanh toán.....	30
Hình 2.20. Sơ đồ tuần tự chức năng thiết lập thông tin của bệnh nhân	31
Hình 2.21. Sơ đồ tuần tự chức năng thiết lập thông tin Bác sĩ.....	32
Hình 2.22. Sơ đồ tuần tự chức năng thiết lập chuyên khoa.....	33
Hình 2.23. Sơ đồ tuần tự chức năng thiết lập dịch vụ	34
Hình 2.24. Sơ đồ tuần tự chức năng thiết lập nhật bệnh	35

Hình 2.25. Sơ đồ tuần tự chức năng thiết lập loại xét nghiệm	36
Hình 2.26. Sơ đồ tuần tự chức năng chatbox trực tiếp với Admin.....	37
Hình 2.27. Sơ đồ tuần tự chức năng xem lịch khám	38
Hình 2.28. Sơ đồ tuần tự chức năng tư vấn theo triệu chứng.....	38
Hình 2.29. Sơ đồ tuần tự chức năng gửi mail xác nhận lịch hẹn	39
Hình 2.30. Sơ đồ hoạt động chức năng đăng nhập.....	40
Hình 2.31. Sơ đồ hoạt động chức năng thiết lập thông tin của bệnh nhân.....	40
Hình 2.32. Sơ đồ hoạt động chức năng đặt lịch và thanh toán.....	41
Hình 2.33. Sơ đồ hoạt động chức năng thiết lập thông tin của bệnh nhân.....	42
Hình 2.34. Sơ đồ hoạt động chức năng thiết lập thông tin Bác sĩ	43
Hình 2.35. Sơ đồ hoạt động chức năng thiết lập chuyên khoa	44
Hình 2.36. Sơ đồ hoạt động chức năng thiết lập dịch vụ	44
Hình 2.37. Sơ đồ hoạt động chức năng thiết lập bệnh.....	45
Hình 2.38. Sơ đồ hoạt động chức năng thiết lập loại xét nghiệm	45
Hình 2.39. Sơ đồ hoạt động chức năng chatbox trực tiếp với Admin.....	46
Hình 2.40. Sơ đồ hoạt động chức năng xem lịch khám.....	47
Hình 2.41. Sơ đồ hoạt động chức năng tư vấn theo triệu chứng	48
Hình 2.42. Sơ đồ hoạt động chức năng gửi mail xác nhận lịch hẹn.....	49
Hình 2.43. Thiết kế lược đồ dữ liệu MySQL	50
Hình 3.1. Giao diện đăng nhập admin.....	61
Hình 3.2. Giao diện dashboard admin	61
Hình 3.3. Giao diện quản lý danh sách lịch hẹn.....	62
Hình 3.4. Giao diện quản lý danh sách Bác sĩ.....	63
Hình 3.5. Giao diện quản lý danh sách khoa.....	63
Hình 3.6. Giao diện quản lý danh sách dịch vụ.....	64
Hình 3.7. Giao diện quản lý danh sách các loại bệnh	64
Hình 3.8. Giao diện quản lý danh sách loại xét nghiệm.....	65
Hình 3.9. Giao diện danh sách được yêu cầu xét nghiệm	65
Hình 3.10. Giao diện quản lý danh sách tài khoản.....	66
Hình 3.11. Mail lịch hẹn được xác nhận	66
Hình 3.12. Giao diện form đặt lịch khám.....	67
Hình 3.13. Giao diện trang chủ	68
Hình 3.14. Giao diện mô tả các khoa cho bệnh nhân	69
Hình 3.15. Giao diện thông tin bác sĩ cho bệnh nhân	70
Hình 3.16. Giao diện tư vấn theo triệu chứng cho bệnh nhân.....	70

Hình 3.17. Giao diện quản lý thông tin cá nhân của bệnh nhân.....	71
Hình 3.18. Giao diện quản lý danh sách lịch hẹn của bệnh nhân.....	72
Hình 3.19. Giao diện chi tiết lịch hẹn của bệnh nhân	72
Hình 3.20. Giao diện hồ sơ khám bệnh của bệnh nhân	73
Hình 3.21. Xuất hồ sơ khám bệnh của bệnh nhân.....	74
Hình 3.22. Giao diện lịch sử khám của Bác sĩ	75
Hình 3.23. Giao diện chính của Bác sĩ	75
Hình 3.24. Giao diện yêu cầu xét nghiệm của Bác sĩ.....	76
Hình 3.25. Giao diện quản lý hồ sơ cá nhân của bác sĩ.....	77
Hình 3.26. Mã nguồn quản lý lịch khám	78
Hình 3.27. Mã nguồn phát triển chức năng quản lý bác sĩ.....	79
Hình 3.28. Mã nguồn phát triển chức năng quản lý người dùng.....	80
Hình 3.29. Mã nguồn phát triển chức năng thông báo mail	81
Hình 3.30. Mã nguồn phát triển chức năng đặt lịch khám	82
Hình 3.31. Mã nguồn phát triển chức năng quản lý thông tin cá nhân bệnh nhân.....	83
Hình 3.32. Mã nguồn phát triển chức năng danh sách lịch khám	84
Hình 3.33. Mã nguồn phát triển chức năng hồ sơ khám	85

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Từ viết tắt	Tên đầy đủ
1	PHP	Hypertext Preprocessor
2	HTML	HyperText Markup Language
3	CSS	Cascading Style Sheets
4	MVC	Model – View – Controller
5	ORM	Object–Relational Mapping
6	CLI	Command Line Interface
7	RDBMS	Relational Database Management System
8	XAMPP	Cross-platform, Apache, MariaDB/MySQL, PHP, Perl
9	DB	Database
10	URL	Uniform Resource Locator
11	SQL	Structured Query Language
12	VSCode	Visual Studio Code
13	FK	Foreign Key
14	PK	Primary Key
15	UI/UX	User Interface / User Experience

MỞ ĐẦU

1. Tính cấp thiết và lý do chọn đề tài

Nhu cầu chăm sóc sức khỏe tăng cao buộc các bệnh viện, phòng khám phải nâng cấp chất lượng quản lý và phục vụ; với những đơn vị tư nhân hoặc quy mô nhỏ như Bệnh viện Phúc An, việc quản lý hồ sơ bệnh nhân, dịch vụ y tế, lịch khám, lịch hẹn bằng phương thức thủ công hoặc nhiều phần mềm rời rạc gây tốn thời gian nhập liệu, tra cứu, đồng bộ; khó thống kê, báo cáo; dễ sai sót do trùng lặp hoặc thiếu nhất quán; và đặc biệt khó đảm bảo bảo mật, toàn vẹn dữ liệu. Do đó, một hệ thống quản lý hồ sơ bệnh nhân và dịch vụ y tế tập trung, chuẩn hóa, tự động là rất cần thiết để giảm tải thao tác thủ công, nâng cao hiệu quả, độ chính xác, tăng cường bảo mật và tra cứu thuận tiện. Với vai trò sinh viên năm cuối ngành công nghệ thông tin, em xem đây là cơ hội vận dụng kiến thức lập trình web, thiết kế hệ thống, cơ sở dữ liệu để xây dựng sản phẩm thực tiễn, góp phần hiện đại hóa quy trình quản lý y tế và hỗ trợ Bệnh viện Phúc An chuyển đổi số, nâng cao chất lượng phục vụ bệnh nhân.

2. Tổng quan thực trạng và Lịch sử nghiên cứu

Trong nước, hiện vẫn rất hiếm các ứng dụng web chuyên sâu hỗ trợ quản lý hồ sơ bệnh nhân và dịch vụ y tế theo mô hình tập trung. Nhiều bệnh viện hoặc phòng khám chỉ sử dụng các phần mềm nhỏ lẻ hoặc website đơn giản cho chức năng đặt lịch - còn việc quản lý bệnh án, dịch vụ, lịch sử khám chữa, thống kê thường được thực hiện thủ công hoặc bán thủ công. Một vài sản phẩm như Vindentist, Nha khoa Aca, FPT.eHospital có cung cấp chức năng đặt lịch khám, tuy nhiên thường thiếu các tính năng quản lý tổng thể: hồ sơ bệnh nhân, lịch sử khám, quản lý dịch vụ - bác sĩ - bệnh nhân.

Ở nước ngoài, có nhiều nền tảng y tế hiện đại và hệ thống quản lý bệnh viện sử dụng công nghệ thông tin - nhưng phần lớn phục vụ các mô hình y tế lớn, cơ sở hạ tầng công nghệ cao; đồng thời có thể yêu cầu chuẩn dữ liệu y tế phức tạp, tích hợp phần mềm chuyên dụng - điều ít phù hợp với điều kiện tại các bệnh viện nhỏ hoạt động trung bình ở Việt Nam.

Như vậy, tồn tại một khoảng trống rõ rệt: thiếu một giải pháp quản lý hồ sơ và dịch vụ y tế đơn giản, dễ triển khai, phù hợp với bệnh viện quy mô nhỏ - vừa đáp ứng nhu cầu quản lý, vừa dễ áp dụng với nguồn lực hạn chế. Đề tài của em nhằm lấp đầy khoảng trống này, xây dựng hệ thống thực tế, dễ dùng, hiệu quả và phù hợp với bối cảnh Việt Nam nói chung và Bệnh viện Phúc An nói riêng.

3. Mục tiêu của đề tài

Mục tiêu chính của đề tài là phát triển một ứng dụng web dựa trên Laravel và MySQL để quản lý tập trung toàn bộ nghiệp vụ bệnh viện, bao gồm hồ sơ bệnh nhân, hồ sơ bác sĩ, danh mục dịch vụ y tế, đặt lịch khám trực tuyến, quản lý lịch hẹn và dịch vụ, đồng thời hỗ trợ kênh tương tác giữa bệnh nhân và bệnh viện. Hệ thống cần có giao diện thân thiện, dễ dùng cho cả bệnh nhân lẫn nhân viên y tế, dễ mở rộng và bảo trì trong tương lai. Cụ thể, đề tài hướng tới việc thiết kế trải nghiệm người dùng rõ ràng, thuận tiện; cung cấp đầy đủ chức năng quản lý thông tin bệnh nhân (thêm, sửa, tra cứu, xem lịch sử khám); quản lý bác sĩ, dịch vụ y tế, lịch khám; cho phép bệnh nhân đặt lịch trực tuyến, quản lý và sắp xếp lịch hẹn hợp lý; tích hợp hệ thống phân quyền người dùng (bệnh nhân, bác sĩ, nhân viên) với các cơ chế bảo mật dữ liệu cơ bản; và đảm bảo kiến trúc mở để có thể bổ sung thêm các chức năng mới khi cần.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: công nghệ và công cụ sử dụng (PHP, Laravel, MySQL, HTML, CSS, JavaScript, công cụ phát triển - XAMPP, VSCode, Composer, Git), cùng hệ thống web quản lý hồ sơ - dịch vụ y tế. Người dùng hệ thống là bệnh nhân, bác sĩ, nhân viên y tế tại Bệnh viện Phúc An; bệnh nhân chủ yếu ở khu vực Đà Nẵng.

Phạm vi nghiên cứu: xây dựng phiên bản prototype của hệ thống - gồm các chức năng cơ bản: quản lý bệnh nhân, bác sĩ, dịch vụ, đặt lịch khám, quản lý lịch hẹn; chưa nghiên cứu các yếu tố phức tạp như tích hợp hệ thống y tế chuẩn, mã hóa y tế, hay kết nối máy thiết bị y tế.

Chương 1. CƠ SỞ LÝ THUYẾT

Đồ án “*xây dựng ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại bệnh viện Phúc An*” là một hệ thống quản lý bệnh viện được xây dựng trên nền tảng Laravel, tập trung vào việc số hóa quy trình đặt lịch khám và quản lý hồ sơ khám chữa bệnh. Hệ thống hỗ trợ ba nhóm người dùng chính là quản trị viên (admin), bác sĩ (doctor) và bệnh nhân (patient), mỗi nhóm được phân quyền truy cập và chức năng riêng biệt. Đối với bệnh nhân, ứng dụng cho phép đăng ký tài khoản, quản lý hồ sơ cá nhân, đặt lịch khám theo bác sĩ, khoa và dịch vụ, theo dõi trạng thái lịch hẹn, xem kết quả khám và hồ sơ bệnh án, cũng như thực hiện và tra cứu thông tin thanh toán. Đối với bác sĩ, hệ thống cung cấp trang dashboard riêng để xem lịch khám theo ngày, thống kê số lượng lịch hẹn theo trạng thái, truy cập danh sách bệnh nhân, ghi nhận hồ sơ bệnh án và yêu cầu các xét nghiệm cần thiết. Trong khi đó, quản trị viên có thể quản lý toàn bộ dữ liệu cốt lõi của bệnh viện như tài khoản người dùng, danh sách bác sĩ, bệnh nhân, phòng ban, dịch vụ, bệnh, lịch hẹn, loại xét nghiệm và kết quả xét nghiệm, đồng thời theo dõi các chỉ số tổng quan và doanh thu thông qua trang dashboard thống kê chi tiết theo ngày, tháng và năm. Nhờ cấu trúc phân quyền rõ ràng và các chức năng được tổ chức hợp lý, hệ thống “*bệnh viện Phúc An*” góp phần hỗ trợ bệnh viện vận hành hiệu quả hơn, giảm bớt thao tác thủ công và nâng cao trải nghiệm cho cả nhân viên y tế lẫn người bệnh.

1.1. Giới thiệu công nghệ và công cụ sử dụng

1.1.1. PHP

PHP là từ viết tắt của thuật ngữ Personal Home Page và hiện nay đang có tên Hypertext Preprocessor. Đây là một dạng mã lệnh hoặc một chuỗi ngôn ngữ kịch bản. Trong đó, ngôn ngữ PHP chủ yếu được phát triển để dành cho những ứng dụng nằm trên máy chủ.

Mỗi khi các lập trình viên PHP viết các chương trình thì các chuỗi lệnh sẽ được chạy ở trên server, từ đó sinh ra mã HTML. Nhờ vậy mà những ứng dụng trên các website có thể chạy được một cách dễ dàng.



Hình 1.1. PHP là một dạng mã lệnh

Ngôn ngữ PHP thông thường sẽ phù hợp với việc lập trình website bởi nó có thể dễ dàng kết nối với các website khác có sử dụng HTML để chạy trên các trình duyệt web. Vì vậy, đây là ngôn ngữ lập trình được người dùng đánh giá là khá dễ đọc. Ngôn ngữ PHP cũng trở thành ngôn ngữ web dev phổ biến mà các PHP Developer phải học trước khi bắt đầu vào nghề.

1.1.2. HTML

HTML (viết tắt của từ Hypertext Markup Language, hay là “Ngôn ngữ Đánh dấu Siêu văn bản”) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như CSS và các ngôn ngữ kịch bản giống như JavaScript.



Hình 1.2. Ngôn ngữ HTML

Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện. HTML mô tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.

Các phần tử HTML là các khối xây dựng của các trang HTML. Với cấu trúc HTML, hình ảnh và các đối tượng khác như biểu mẫu tương tác có thể được nhúng vào trang được hiển thị. HTML cung cấp một phương tiện để tạo tài liệu có cấu trúc bằng cách biểu thị ngữ nghĩa cấu trúc cho văn bản như headings, paragraphs, lists, links, quotes và các mục khác. Các phần tử HTML được phân định bằng các tags, được viết bằng dấu ngoặc nhọn. Các tags như và giới thiệu trực tiếp nội dung vào trang. Các tags khác như ao quanh và cung cấp thông tin về văn bản tài liệu và có thể bao gồm các thẻ khác làm phần tử phụ. Các trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để diễn giải nội dung của trang.

1.1.3. CSS

CSS là viết tắt của "Cascading Style Sheets", một ngôn ngữ lập trình được sử dụng để định dạng và trang trí các trang web. CSS cho phép bạn điều chỉnh giao diện của trang web, bao gồm màu sắc, font chữ, khoảng cách và bố cục của nội dung. Nền tảng cũng cho phép người dùng tạo các hiệu ứng và chuyển động trên trang web.



Hình 1.3. Ngôn ngữ CSS

Mô hình hoạt động cơ bản chính là liên kết các tập tin CSS với trang HTML. Trong tập tin CSS, bạn xác định các quy tắc kiểu mẫu (style rules) cho các phần tử HTML cụ thể. Khi trình duyệt web tải trang, nó sẽ áp dụng các kiểu mẫu từ tập tin CSS vào nội dung HTML. Từ đó hiển thị trang web theo cách mà bạn đã thiết lập.

CSS cũng hỗ trợ các đặc tính kế thừa (inheritance) và độ ưu tiên (specificity) nhằm tổ chức kiểu mẫu một cách hiệu quả trên trang web. Nền tảng có tác dụng hạn chế việc lặp lại mã code và nâng cao tính duy trì (maintainability) của trang web.

Bộ cục của một đoạn CSS là gì? Cascading Style Sheets có các phần chính sau:

Selector: Đây là phần của đoạn CSS xác định các phần tử HTML mà bạn muốn áp dụng kiểu mẫu. Selector có thể là tên của một thẻ HTML cụ thể (ví dụ: "p" cho đoạn văn), một lớp CSS (ví dụ: ".container" cho class "container") hoặc một ID (ví dụ: "#header" cho ID "header").

Property: Đây là thuộc tính mà bạn muốn định dạng cho các phần tử được chọn thông qua selector. Ví dụ, các thuộc tính bao gồm màu sắc (color), font chữ (font-family), kích thước (size), độ dày (font-weight) và bề mặt tiếp xúc (padding, margin).

Value: Đây là giá trị của thuộc tính mà bạn muốn áp dụng cho các phần tử được chọn. Ví dụ, nếu bạn muốn thiết lập màu chữ cho đoạn văn là màu xanh dương, giá trị của thuộc tính "color" sẽ là "blue".

1.1.4. Javascript

JavaScript là ngôn ngữ kịch bản cho phép tạo ra trang web động - cập nhật nội dung theo ngữ cảnh, điều khiển đa phương tiện, hoạt cảnh các hình ảnh và nhiều thứ hay ho khác. (Dĩ nhiên không phải mọi thứ, nhưng chỉ với một vài dòng code, JavaScript có thể làm được nhiều điều đáng kinh ngạc.)



Hình 1.4. Ngôn ngữ JavaScript

JavaScript cho phép bạn làm cho các trang HTML của bạn tương tác với thao tác của người dùng hơn. Ví dụ khi bạn đăng nhập vào một website thành công, bạn nhận được thông báo đăng nhập thành công. Thì điều đó được tạo nên từ javascript đây

Sau khi học xong 3 ngôn ngữ chính (Html, CSS, Javascript) thì lúc này bạn đã đủ khả năng để học các framework, bạn hãy chọn cho mình 1 framework để bắt đầu, đừng quá lo lắng khi chọn framework vì khi bạn đã thuần thục Html, CSS và Javascript thì bạn sẽ đủ tư duy để bắt đầu 1 framework. Khi thuần thục 1 framework thì framework sau cũng sẽ trở nên dễ dàng với bạn.

1.1.6. Giới thiệu MySQL

MySQL là hệ thống quản trị cơ sở dữ liệu (RDBMS) mã nguồn mở và hoạt động theo mô hình client-server. MySQL đã trở thành một phần mềm quan trọng trong việc lưu trữ, quản lý và truy xuất dữ liệu. Được phát hành chính thức từ những năm 1990, MySQL đã phát triển và trở nên mạnh mẽ và linh hoạt, được tích hợp với cả Apache và PHP.



Hình 1.5. Cơ sở dữ liệu MySql

MySQL cho phép quản lý dữ liệu thông qua các cơ sở dữ liệu, trong đó, mỗi cơ sở dữ liệu có thể chứa nhiều bảng quan hệ với dữ liệu tương ứng. Công cụ này cũng hỗ trợ truy xuất và thao tác dữ liệu bằng ngôn ngữ truy vấn SQL (Structured Query Language). SQL giúp người dùng tạo, cập nhật và xóa dữ liệu trong cơ sở dữ liệu MySQL một cách dễ dàng và hiệu quả.

MySQL dần trở nên phổ biến và được ưa chuộng trong nhiều lĩnh vực, từ các ứng dụng web cá nhân cho đến hệ thống thông tin doanh nghiệp lớn. Điều này là nhờ vào hiệu suất và tính ổn định, khả năng mở rộng linh hoạt và tính bảo mật cao của công cụ này. Cộng đồng người dùng MySQL cũng đóng vai trò quan trọng trong việc phát triển và cải thiện hệ thống, cung cấp hỗ trợ và chia sẻ kiến thức cho những người mới làm quen với ngành phần mềm.

MySQL có hai phiên bản chính là MySQL Community Server (miễn phí) và MySQL Enterprise Server (có phí). Phiên bản miễn phí được sử dụng rộng rãi trong cộng đồng phát triển phần mềm và các dự án cá nhân, trong khi phiên bản Enterprise Server cung cấp thêm các tính năng và hỗ trợ chuyên nghiệp dành cho doanh nghiệp.

Cơ chế hoạt động của MySQL là gì? Ở môi trường như hiện tại, MySQL có cách vận hành như sau:

Tạo và quản lý bảng: MySQL cho phép tạo ra các bảng để lưu trữ dữ liệu. Mỗi bảng chứa các cột và hàng, trong đó mỗi cột đại diện cho một thuộc tính và mỗi hàng đại diện cho một bản ghi trong cơ sở dữ liệu. Người dùng có thể định nghĩa các ràng buộc, quan hệ và chỉ mục trong bảng để xác định sự liên quan giữa các bảng.

Gửi yêu cầu SQL: Người dùng hoặc Client có thể gửi các yêu cầu SQL đặc biệt tới MySQL. Yêu cầu này có thể bao gồm các câu truy vấn SELECT, INSERT, UPDATE, DELETE và nhiều loại câu lệnh khác.

Xử lý yêu cầu: MySQL Server nhận và xử lý các yêu cầu SQL từ Client. Công cụ sẽ kiểm tra cú pháp và tính hợp lệ của yêu cầu, sau đó thực hiện các thao tác tương ứng trên cơ sở dữ liệu. MySQL sử dụng bộ trình diễn truy vấn (query optimizer) để tối ưu hóa hiệu suất truy vấn.

Trả kết quả: Sau khi xử lý yêu cầu, MySQL Server sẽ trả về kết quả cho Client. Kết quả có thể là dữ liệu truy vấn từ cơ sở dữ liệu hoặc thông báo về việc cập nhật, thêm, xóa dữ liệu.

1.1.6. Giới thiệu và cài đặt Xampp

XAMPP là một phần mềm để thiết kế và phát triển website theo ngôn ngữ PHP. Phần mềm này hoạt động dựa trên sự tích hợp của 5 yếu tố chính là Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) và Perl (P). XAMPP cho phép giả lập môi trường server hosting ngay trên máy tính của bạn, để bạn chạy demo website mà không cần phải mua hosting hay VPS.



Hình 1.6. Cơ sở dữ liệu Xampp

Theo phần đa người dùng đã thử nghiệm và đánh giá, XAMPP có khả năng tương thích cực cao, dễ dàng hoạt động trên nhiều hệ điều hành và sự tích hợp nhiều công cụ, tính năng. Bên cạnh đó, ngoài việc được dùng để xây dựng và phát triển website theo ngôn ngữ PHP. XAMPP còn thông qua localhost của máy tính cá nhân, phát triển và nghiên cứu website. Nhờ vậy mà có thể biến máy tính cá nhân thành máy chủ, lưu trữ cho máy chủ trang web bằng ổ cứng máy tính. Tuy nhiên, có 1 lưu ý nhỏ cho người

dùng là đối với các website kinh doanh, cần vận hành liên tục thì không nên sử dụng phần mềm XAMPP bởi không khả thi. Nguyên nhân là bởi localhost dùng máy tính để làm máy chủ dẫn tới việc duy trì máy chủ không đảm bảo tốc độ và gây khó khăn trong hoạt động.

Ưu nhược điểm của phần mềm XAMPP

Xét về ưu điểm, Xampp được ưa chuộng và sử dụng rộng rãi bởi những điểm cộng tuyệt vời như: Có thể chạy được trên tất cả các hệ điều hành như Windows, MacOS, Linux; cấu hình cực kỳ đơn giản, dễ sử dụng cho người mới bắt đầu; Nhiều tính năng hữu ích như giả lập Server, giả lập Mail Server; tích hợp nhiều thành phần với các chức năng như Apache, PHP, MYSQL; mã nguồn mở, giao diện quản lý vô cùng tiện lợi.

Về nhược điểm, Xampp còn tồn tại một số nhược điểm nhất định. Cụ thể: XAMPP không hỗ trợ Module; không được tích hợp Version MySQL, do đó, đôi khi sẽ mang đến sự bất tiện cho người dùng; đặc biệt nhất dung lượng của XAMPP khá nặng, khoảng 141MB cho file cài đặt.

Cách cài đặt XAMPP

Cài đặt tập tin

Để cài đặt XAMPP, trước hết bạn cần phải tải tập tin cài đặt trên trang chủ hệ thống.

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

Version	Checksum	Size
7.4.27 / PHP 7.4.27	What's Included? md5 sha1 Download (64 bit)	160 Mb
8.0.15 / PHP 8.0.15	What's Included? md5 sha1 Download (64 bit)	161 Mb
8.1.2 / PHP 8.1.2	What's Included? md5 sha1 Download (64 bit)	164 Mb

[Requirements](#) [Add-ons](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Hình 1.7. Cài đặt môi trường Xampp

Cài đặt vào máy

Sau khi tải xong các phần mềm cài đặt, bạn thực hiện cài đặt XAMPP vào máy.

Bước 1: Khởi chạy file. Quá trình cài đặt có thể bị chặn bởi các chương trình antivirus. Bạn cứ chọn “Yes” để tiếp tục các bước tiếp theo.

Bước 2: Chọn các gói thành phần. Bạn chọn các gói thành phần của XAMPP mà bạn muốn sử dụng. Apache, PHP và MySQL là 3 phần bạn sẽ sử dụng nhiều nhất.

Bước 3: Sau khi XAMPP giải nén xong, bạn chọn “Finish” để kết thúc quá trình cài đặt.

1.1.7. Giới thiệu và cài đặt Visual studio code

Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code. Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux. Hơn thế nữa, VS Code còn cho khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng.

Visual Studio Code hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting. Đặc biệt là tự hoàn thành mã thông minh, Snippets, và khả năng cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép các lập trình viên thay đổi Theme, phím tắt, và đa dạng các tùy chọn khác. Mặc dù trình soạn thảo Code này tương đối nhẹ, nhưng lại bao gồm các tính năng mạnh mẽ.

Dù mới được phát hành nhưng VSCode là một trong những Code Editor mạnh mẽ và phổ biến nhất dành cho lập trình viên. Nhờ hỗ trợ nhiều ngôn ngữ lập trình phổ biến, tích hợp đầy đủ các tính năng và khả năng mở rộng, nên VSCode trở nên cực kỳ thân thuộc với bất kỳ lập trình viên nào.

Visual Studio Code là gì được rất nhiều người tìm hiểu. Đây cũng là một trong các ứng dụng được dân IT “săn đón” và tải về và sử dụng rất nhiều. Visual Studio Code cũng luôn có những cải tiến và tạo ra đa dạng các tiện ích đi kèm từ đó giúp cho các lập trình viên sử dụng dễ dàng hơn. Trong đó có thể kể đến những ưu điểm sau:

Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,...

Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.

Các tiện ích mở rộng rất đa dạng và phong phú.

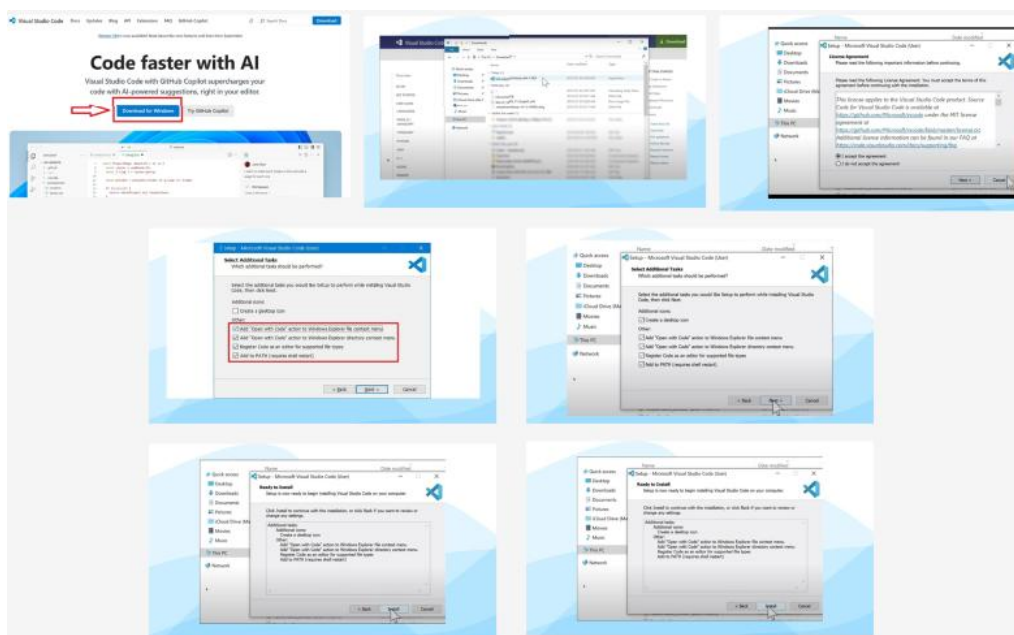
Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug),...

Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.

Visual Studio Code là một trong những trình biên tập Code rất phổ biến nhất hiện nay. Ứng dụng này cũng ngày càng chứng tỏ ưu thế vượt trội của mình khi so sánh với những phần mềm khác. Tuy bản miễn phí không có nhiều các tính năng nâng cao nhưng Visual Studio Code thực sự có thể đáp ứng được hầu hết nhu cầu cơ bản của lập trình viên.

Visual Studio Code là phần mềm miễn phí. Chính vì vậy ứng dụng chuyên biên tập, soạn thảo Code này trở nên phổ biến nhất hiện nay. Với việc không ngừng cải tiến và áp dụng rất nhiều các công nghệ mới, Visual Studio Code đã được các lập trình viên chứng minh hiệu quả. Ngoài các Developers, thì Testers hay Data Administrators và tất cả những người dùng đang quan tâm đến lập trình đều có thể sử dụng Visual Studio.

Cài đặt Visual Studio Code



Hình 1.8. Cài đặt Visual Studio Code

1.2. Framework Laravel

Laravel là một framework web mã nguồn mở được viết bằng ngôn ngữ PHP. Nó được phát triển bởi Taylor Otwell và được phát hành lần đầu vào năm 2011. Laravel cung cấp một kiến trúc MVC (Model-View-Controller) cho phép phát triển ứng dụng web dễ dàng hơn và có tính bảo mật cao hơn.

Laravel cũng cung cấp nhiều tính năng mạnh mẽ, bao gồm hệ thống định tuyến (routing), tạo truy vấn cơ sở dữ liệu (query builder), hỗ trợ giao diện dòng lệnh (command line interface) và cách tiếp cận đối tượng (object-oriented approach). Laravel cũng có thư viện mở rộng (package) khá phong phú và được cộng đồng hỗ trợ nhiệt tình.

Laravel được sử dụng rộng rãi trong phát triển ứng dụng web, từ các ứng dụng đơn giản đến các hệ thống phức tạp và lớn. Nó là một trong những framework PHP phổ biến nhất hiện nay và được đánh giá là có hiệu suất tốt và dễ dàng sử dụng.

Các tính năng chính của Laravel bao gồm:

Kiến trúc MVC: Laravel cung cấp một kiến trúc MVC (Model-View-Controller) cho phép phân tách code logic, dữ liệu và giao diện người dùng. Điều này giúp dễ dàng bảo trì và phát triển ứng dụng web.

Hiệu suất cao: Laravel được thiết kế để tối ưu hóa hiệu suất và tốc độ, cho phép xử lý nhanh hơn và giảm thời gian phản hồi của ứng dụng web.

Tính bảo mật cao: Laravel có tính bảo mật cao và cung cấp các tính năng bảo vệ như xác thực, mã hóa mật khẩu và chống tấn công CSRF.

Hệ thống định tuyến mạnh mẽ: Laravel cung cấp hệ thống định tuyến mạnh mẽ và linh hoạt, cho phép xử lý các yêu cầu đến từ nhiều phương thức HTTP khác nhau.

Truy vấn cơ sở dữ liệu: Laravel cung cấp một cách tiếp cận đơn giản và hiệu quả để truy vấn cơ sở dữ liệu, cho phép truy vấn thông qua cú pháp đơn giản hoặc sử dụng một bộ lọc mạnh mẽ.

Hỗ trợ tốt cho giao diện dòng lệnh: Laravel có thể được sử dụng để tạo và quản lý các tác vụ liên quan đến dòng lệnh như tạo migration, seed database hoặc tạo test.

Cộng đồng phát triển mạnh mẽ: Laravel có một cộng đồng phát triển rộng lớn và nhiệt tình, với nhiều tài liệu hướng dẫn, ví dụ mã nguồn và các gói mở rộng được chia sẻ miễn phí.

Những lý do trên chỉ là một phần nhỏ trong những lợi ích của Laravel. Nó là một trong những framework web PHP mạnh mẽ nhất và được đánh giá cao bởi cộng đồng phát triển và người dùng.

Ưu và nhược điểm của Laravel

Ưu điểm của Laravel:

Tính mô-đun cao: Laravel là một framework modul được phát triển để cho phép các nhà phát triển xây dựng ứng dụng web một cách linh hoạt và dễ dàng hơn.

Định tuyến mạnh mẽ: Laravel cung cấp hệ thống định tuyến mạnh mẽ và dễ sử dụng, cho phép xác định các URL và điều hướng người dùng đến các trang web cụ thể.

Tính năng bảo mật cao: Laravel cung cấp các tính năng bảo mật mạnh mẽ như mã hóa, xác thực và phân quyền.

Tính năng Artisan: Laravel có tính năng Artisan cho phép tạo các lệnh tùy chỉnh và tự động hóa các tác vụ thường xuyên trong quá trình phát triển.

Tính năng Blade Template: Laravel cung cấp một hệ thống mẫu gọi là Blade, giúp phát triển các giao diện người dùng dễ dàng hơn.

Tính năng ORM (Object-Relational Mapping): Laravel cung cấp một ORM mạnh mẽ, cho phép tương tác với CSDL một cách dễ dàng.

Hỗ trợ cộng đồng: Laravel có một cộng đồng lớn, với nhiều tài liệu, ví dụ và gói phần mềm có sẵn để giúp các nhà phát triển.

Nhược điểm của Laravel:

Đòi hỏi kiến thức cao: Laravel có nhiều tính năng và cấu hình phức tạp, vì vậy nó đòi hỏi kiến thức lập trình PHP cao hơn so với các framework khác.

Tốc độ tải trang chậm hơn: Vì Laravel cung cấp nhiều tính năng và phải xử lý nhiều thư viện, nó có thể dẫn đến tốc độ tải trang chậm hơn so với các framework khác.

Cấu hình phức tạp: Cấu hình Laravel có thể phức tạp, đặc biệt là đối với các nhà phát triển mới, khi phải làm quen với các tính năng mới và cấu trúc của framework.

Độ phức tạp của quá trình triển khai: Vì Laravel cung cấp nhiều tính năng và phụ thuộc vào nhiều thư viện khác nhau, quá trình triển khai có thể phức tạp hơn so với các framework khác.

Khả năng mở rộng có hạn: Mặc dù Laravel cung cấp nhiều tính năng và gói mở rộng, nhưng nó vẫn có những giới hạn về khả năng mở rộng khi cần tích hợp các tính năng mới hoặc các công nghệ khác.

Yêu cầu cấu hình máy chủ tốt: Vì Laravel là một framework PHP mạnh mẽ và có nhiều tính năng, vì vậy để chạy một ứng dụng Laravel cần có cấu hình máy chủ tốt hơn so với các framework PHP khác.

Tuy nhiên, những nhược điểm này có thể được khắc phục bằng cách nâng cao kiến thức và kinh nghiệm phát triển web của các nhà phát triển, cũng như tối ưu hóa mã nguồn và cấu hình máy chủ để tăng tốc độ và hiệu suất của ứng dụng. Với những ưu

điểm mạnh mẽ của mình, Laravel vẫn là một trong những lựa chọn phổ biến cho việc phát triển các ứng dụng web PHP.

Các tính năng chính của Laravel

Laravel là một framework phát triển ứng dụng web PHP mạnh mẽ và linh hoạt, cung cấp nhiều tính năng và công cụ hữu ích để giúp các nhà phát triển xây dựng ứng dụng web hiệu quả và dễ bảo trì. Dưới đây là một số tính năng chính của Laravel:

Routing: Laravel cung cấp một hệ thống routing mạnh mẽ và linh hoạt, cho phép các nhà phát triển định nghĩa các tuyến đường URL và xử lý các yêu cầu HTTP dễ dàng.

Middleware: Laravel hỗ trợ middleware, cho phép các nhà phát triển xử lý các yêu cầu HTTP trước khi chúng được xử lý bởi ứng dụng.

Blade Template Engine: Blade là một engine mẫu dựa trên PHP cung cấp các tính năng như kế thừa mẫu, phân mảnh mẫu và cú pháp ngắn gọn, giúp các nhà phát triển tạo ra các giao diện người dùng đẹp và dễ bảo trì.

Eloquent ORM: Eloquent là một ORM (Object Relational Mapping) mạnh mẽ, giúp các nhà phát triển tương tác với cơ sở dữ liệu dễ dàng hơn bằng cách sử dụng các đối tượng PHP thay vì viết câu truy vấn SQL.

Migration: Laravel cung cấp một hệ thống migration cho phép các nhà phát triển quản lý cơ sở dữ liệu của ứng dụng dễ dàng và an toàn.

Artisan CLI: Artisan là một công cụ dòng lệnh mạnh mẽ cho phép các nhà phát triển tạo và quản lý các tệp mẫu, thực hiện các tác vụ cần thiết cho ứng dụng, và nhiều tính năng khác.

Authentication: Laravel cung cấp các tính năng xác thực và phân quyền mạnh mẽ, giúp các nhà phát triển xây dựng các hệ thống đăng nhập và phân quyền dễ dàng và an toàn.

Queueing: Laravel hỗ trợ queueing, cho phép các nhà phát triển xử lý các tác vụ nền một cách hiệu quả, giúp tăng tốc độ và hiệu suất của ứng dụng.

Cài đặt ứng dụng

Cài đặt XAMPP để có Apache, PHP, MySQL (và phpMyAdmin) - XAMPP là giải pháp phổ biến để chạy Laravel trên localhost.

Cài Composer - công cụ quản lý thư viện PHP - để cài Laravel và các package.

Mở XAMPP, khởi động Apache và MySQL. Vào <http://localhost> trong trình duyệt để kiểm tra xem XAMPP chạy đúng chưa.

Mở command prompt (CMD) hoặc Git Bash, chuyển đến thư mục htdocs của XAMPP:

```
cd C:\xampp\htdocs
```

Chạy lệnh tạo project Laravel mới:

```
composer create-project --prefer-dist laravel/laravel hospital
```

Vào thư mục project:

```
cd hospital
```

Mở file .env, cấu hình thông số kết nối database MySQL: tên database, user, password. Ví dụ:

```
DB_CONNECTION=mysql
```

```
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
```

```
DB_DATABASE=ten_database_cua_ban
```

```
DB_USERNAME=root
```

```
DB_PASSWORD=
```

Chạy migration để Laravel tạo các bảng theo thiết kế:

```
php artisan migrate
```

(Nếu có) Chạy seeder để thêm dữ liệu mẫu: bác sĩ, dịch vụ, chuyên khoa, ... tùy theo bạn đã chuẩn bị. Ví dụ:

```
php artisan db:seed
```

Sau khi cấu hình xong, chạy lệnh:

```
php artisan serve
```

Mở trình duyệt và truy cập <http://localhost:8000> - nếu thấy trang mặc định của Laravel (Welcome page), nghĩa là ứng dụng đã chạy thành công.

1.3. Đặc điểm và yêu cầu chung của hệ thống

Hệ thống y tế đòi hỏi phải quản lý đồng thời nhiều loại đối tượng khác nhau như bệnh nhân, bác sĩ, phòng ban, dịch vụ khám chữa bệnh, lịch hẹn, xét nghiệm và hồ sơ

bệnh án. Do đó, hệ thống cần được tổ chức theo mô hình phân lớp rõ ràng, có các mô hình dữ liệu (model) cho từng thực thể như 'User', 'Patient', 'Doctor', 'Appointment', 'Service',... đi kèm quan hệ chặt chẽ giữa chúng. Mỗi người dùng được gán một vai trò (admin, doctor, patient) và phải trải qua bước xác thực, phân quyền nghiêm ngặt để đảm bảo chỉ truy cập được đúng phạm vi chức năng và dữ liệu cho phép. Đây là yêu cầu rất quan trọng đối với hệ thống y tế, bởi dữ liệu sức khỏe là thông tin nhạy cảm, cần được bảo mật, kiểm soát truy cập theo nguyên tắc “ít quyền nhất”.

Bên cạnh đó, hệ thống phải hỗ trợ quy trình nghiệp vụ y tế phức tạp nhưng vận hành mạch lạc. Ví dụ, việc đặt lịch khám không chỉ đơn thuần là lưu thông tin lịch hẹn, mà còn phải kiểm soát số lượng ca khám tối đa theo buổi cho mỗi bác sĩ, tránh trùng lịch, phân biệt trạng thái (chờ duyệt, đã xác nhận, đã hoàn thành, đã hủy), gắn kết với thông tin bệnh nhân, bác sĩ, dịch vụ và cả thanh toán. Hệ thống cũng cần khả năng thống kê, báo cáo theo nhiều tiêu chí (số lượng lịch hẹn, tình trạng lịch, doanh thu theo ngày-tháng-năm, số lượng bác sĩ, bệnh nhân, dịch vụ...) để hỗ trợ công tác quản lý và ra quyết định của ban lãnh đạo bệnh viện. Cuối cùng, một hệ thống y tế hiện đại còn phải nâng cao trải nghiệm người dùng thông qua các tính năng hỗ trợ như chat giữa bệnh nhân và admin, quản lý hồ sơ bệnh án điện tử, tra cứu kết quả xét nghiệm và theo dõi lịch sử khám chữa bệnh, từ đó góp phần số hóa toàn diện quy trình hoạt động của bệnh viện.

1.4. Kết luận chương 1

Chương 1 đã trình bày những cơ sở lý thuyết nền tảng phục vụ cho việc xây dựng ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại bệnh viện PHÚC AN. Đề tài đã giới thiệu tổng quan về dự án, mục tiêu số hóa quy trình đặt lịch khám và quản lý hồ sơ khám chữa bệnh, cũng như mô hình phân quyền ba vai trò chính: quản trị viên, bác sĩ và bệnh nhân. Chương đã làm rõ các công nghệ và công cụ chủ đạo được sử dụng như ngôn ngữ PHP, bộ ba HTML, CSS, JavaScript cho phần giao diện, framework Laravel cho phía backend, hệ quản trị cơ sở dữ liệu MySQL. Cuối cùng, chương đã phân tích các đặc điểm và yêu cầu chung của một hệ thống thông tin y tế, nhấn mạnh đến tính bảo mật, quản lý dữ liệu đa thực thể, hỗ trợ quy trình nghiệp vụ phức tạp và khả năng thống kê, báo cáo phục vụ quản lý. Những nội dung này là nền tảng quan trọng để sang các chương tiếp theo, nơi sẽ tập trung vào phân tích, thiết kế và hiện thực hóa hệ thống theo đúng mục tiêu đề ra.

Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu

Hệ thống Quản lý Bệnh viện là một giải pháp toàn diện được phát triển trên nền tảng Laravel, cung cấp các tính năng quản lý hiện đại cho các cơ sở y tế. Ứng dụng cho phép quản lý thông tin bệnh nhân, đặt lịch hẹn khám trực tuyến, quản lý hồ sơ bác sĩ và hỗ trợ đa người dùng với các vai trò khác nhau. Hệ thống còn tích hợp các chức năng xét nghiệm y khoa, quản lý khoa phòng, dịch vụ y tế và thanh toán viện phí. Với giao diện thân thiện, dễ sử dụng, ứng dụng giúp tối ưu hóa quy trình làm việc, nâng cao hiệu quả quản lý và chất lượng phục vụ bệnh nhân, đồng thời cung cấp các công cụ báo cáo thống kê đa dạng phục vụ công tác quản lý và ra quyết định.

2.1.1. Đặc tả yêu cầu chức năng của hệ thống

Tiến hành nghiên cứu chi tiết các quy trình nghiệp vụ và luồng hoạt động của hệ thống ứng dụng quản lý hồ sơ bệnh nhân tại Bệnh viện Phúc An, nhằm đảm bảo nắm rõ cấu trúc, chức năng và cách thức vận hành của toàn bộ hệ thống.

Xây dựng và phát triển các chức năng của hệ thống dựa trên những yêu cầu từ người dùng nhằm tạo nên một ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế hoàn chỉnh, hiệu quả và dễ sử dụng tại Bệnh viện Phúc An.

Xây dựng các chức năng từ những yêu cầu của người dùng:

Chia sẻ thông tin về Bác sĩ: Cung cấp thông tin của Bác sĩ (Họ tên đầy đủ: Tên Bác sĩ giúp bệnh nhân dễ dàng nhận diện và chọn lựa, Chuyên môn: Liệt kê lĩnh vực chuyên môn của Bác sĩ. Kinh nghiệm: Số năm kinh nghiệm làm việc và những vị trí công tác nổi bật trước đây. Bằng cấp và chứng chỉ: Bác sĩ cần liệt kê các bằng cấp từ các trường đại học, các chứng chỉ về chuyên môn hoặc đào tạo nâng cao.)

Chia sẻ thông tin về chuyên khoa trong hệ thống ứng dụng: Ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại Bệnh viện Phúc An cần cho phép cung cấp đầy đủ thông tin về các chuyên khoa, bao gồm cơ sở vật chất và dịch vụ y tế chất lượng cao nhằm đáp ứng nhu cầu chăm sóc sức khỏe của bệnh nhân. Mỗi chuyên khoa phải được mô tả rõ ràng về chức năng, dịch vụ khám - chẩn đoán - điều trị. Hệ thống cần hỗ trợ bệnh viện giới thiệu cơ sở hạ tầng của từng chuyên khoa, như phòng khám hiện đại, khu vực điều trị vô trùng, thiết bị chẩn đoán tiên tiến (ví dụ: máy X-quang kỹ thuật số, thiết bị nội soi, hệ thống xét nghiệm). Bên cạnh đó, ứng dụng phải cho phép hiển thị thông tin về đội ngũ y bác sĩ, bao gồm trình độ chuyên môn, chứng chỉ hành nghề, kinh nghiệm

điều trị từ các bệnh lý thông thường đến các kỹ thuật chuyên sâu. Ngoài bác sĩ, hệ thống cũng cần thể hiện thông tin về đội ngũ y tá, kỹ thuật viên và nhân viên hỗ trợ liên quan, giúp bệnh nhân hiểu rõ về chất lượng phục vụ của từng chuyên khoa.

Đặt lịch khám trực tuyến: Tính năng đặt lịch khám trực tuyến đóng vai trò quan trọng, giúp bệnh nhân chủ động lựa chọn chuyên khoa, bác sĩ và thời gian khám phù hợp, đồng thời hỗ trợ bệnh viện quản lý lịch hẹn hiệu quả. Khi đặt lịch khám thành công admin duyệt thì hệ thống sẽ tự động gửi mail xác nhận lịch hẹn qua email để khách hàng xác nhận chắc chắn lịch hẹn, giúp web và khách hàng yên tâm và tránh được tình trạng quá tải hoặc chông chéo lịch khám.

Tư vấn theo triệu chứng: Tính năng tư vấn theo triệu chứng khi bệnh nhân không biết các triệu chứng của mình bệnh gì hay nên khám khoa nào thì có thể nhập triệu chứng vào ô tư vấn và hệ thống sẽ gợi ý ra các bệnh triệu chứng đó bệnh gì và gợi ý khoa, bác sĩ và các dịch vụ của khoa đó cho bệnh nhân.

2.1.2. Yêu cầu phi chức năng

Bên cạnh các yêu cầu về chức năng, hệ thống ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại Bệnh viện Phúc An còn phải đáp ứng tốt các yêu cầu phi chức năng để đảm bảo chất lượng sử dụng trong thực tế. Trước hết, bảo mật là yếu tố then chốt, do hệ thống xử lý dữ liệu sức khỏe và thông tin cá nhân nhạy cảm của bệnh nhân. Ứng dụng cần áp dụng cơ chế xác thực và phân quyền chặt chẽ theo vai trò (admin, bác sĩ, bệnh nhân), mã hóa mật khẩu, kiểm soát phiên đăng nhập, hạn chế truy cập trái phép và có chính sách bảo vệ dữ liệu phù hợp. Về hiệu suất, hệ thống phải đảm bảo thời gian phản hồi nhanh, xử lý ổn định ngay cả khi có nhiều người dùng truy cập đồng thời, đặc biệt trong các khung giờ cao điểm đặt lịch và tra cứu thông tin. Đồng thời, kiến trúc hệ thống cần được thiết kế theo hướng khả năng mở rộng, cho phép dễ dàng bổ sung chức năng mới (ví dụ thêm chuyên khoa, dịch vụ, module xét nghiệm, bảo hiểm y tế...) hoặc mở rộng quy mô người dùng mà không phải thay đổi quá nhiều cấu trúc lõi.

Yêu cầu quan trọng khác là tính dễ sử dụng và thân thiện với người dùng. Giao diện cần rõ ràng, trực quan, bố cục hợp lý cho từng nhóm đối tượng: bệnh nhân có thể dễ dàng đặt lịch, xem lịch sử khám; bác sĩ thao tác nhanh với danh sách lịch hẹn và hồ sơ bệnh án; admin thuận tiện trong việc quản lý danh mục và báo cáo. Ngôn ngữ, biểu tượng, thông báo lỗi và hướng dẫn trên giao diện phải đơn giản, dễ hiểu đối với người dùng phổ thông, đồng thời hỗ trợ tốt trên các trình duyệt phổ biến. Về giao diện và trải nghiệm người dùng (UI/UX), hệ thống nên có thiết kế đồng nhất, màu sắc hài hòa, hỗ trợ responsive để sử dụng được trên nhiều kích thước màn hình (máy tính, laptop). Cuối

cùng, hệ thống cũng cần đảm bảo tính tin cậy và ổn định, hạn chế lỗi phát sinh, có cơ chế sao lưu và phục hồi dữ liệu, nhật ký hoạt động (log) để hỗ trợ giám sát và xử lý sự cố khi cần thiết. Những yêu cầu phi chức năng này là nền tảng giúp hệ thống vận hành an toàn, bền vững và mang lại trải nghiệm tốt cho người dùng.

2.2.1. Kiến trúc hệ thống

Hệ thống quản lý hồ sơ bệnh nhân và dịch vụ y tế tại Bệnh viện Phúc An được xây dựng theo kiến trúc web 3 lớp trên nền tảng MVC của Laravel, bao gồm: lớp giao diện (View), lớp xử lý nghiệp vụ (Controller/Service) và lớp dữ liệu (Model/Database).

Xử lý nghiệp vụ (Controller/Service).

└─ app/Http/	# LỚP XỬ LÝ NGHIỆP VỤ (CONTROLLER / MIDDLEWARE)
└─ Controllers/	
└─ AppointmentController.php	# Đặt lịch, hủy lịch, xem chi tiết, hồ sơ khám
└─ HomeController.php	# Trang chủ, dịch vụ, chuyên khoa, tư vấn...
└─ PaymentController.php	# Xử lý thanh toán lịch khám
└─ ChatController.php	# Xử lý luồng chat giữa người dùng
└─ ProfileController.php	# Cập nhật hồ sơ cá nhân bệnh nhân
└─ PatientController.php	# Dashboard bệnh nhân, danh sách lịch hẹn
└─ ServiceController.php	# Hiển thị dịch vụ, chuyên khoa cho patient
└─ AIChatController.php	# Tư vấn/trò chuyện AI (nếu có)
└─ Admin/	# Nhóm controller dành cho ADMIN
└─ DashboardController.php	# Thống kê, báo cáo, doanh thu
└─ AppointmentController.php	# Quản lý toàn bộ lịch hẹn
└─ DoctorController.php	# Quản lý hồ sơ bác sĩ
└─ DepartmentController.php	# Quản lý chuyên khoa
└─ DiseaseController.php	# Quản lý bệnh / triệu chứng
└─ LabTestController.php	# Quản lý xét nghiệm (admin)
└─ PatientController.php	# Quản lý danh sách bệnh nhân
└─ TestTypeController.php	# Quản lý loại xét nghiệm
└─ Doctor/	# Nhóm controller dành cho BÁC SĨ
└─ DoctorDashboardController.php	# Dashboard lịch khám theo ngày
└─ MedicalRecordController.php	# Tạo / xem hồ sơ bệnh án
└─ LabTestController.php	# Bác sĩ yêu cầu xét nghiệm
└─ PatientController.php	# Danh sách bệnh nhân của bác sĩ
└─ Auth/	# Đăng nhập, reset mật khẩu
└─ Middleware/	# Kiểm tra đăng nhập, role: admin/doctor/patient
└─ Kernel.php	# Đăng ký middleware, nhóm route

Hình 2.1. Kiến trúc controller

Giao diện (View)

└ resources/	# LỚP GIAO DIỆN (VIEW / PRESENTATION)
└ views/	# Giao diện người dùng (Blade templates)
└ layouts/	# • Khung layout chung (header, footer, sidebar)
└ home/	# • Trang giới thiệu, dịch vụ, chuyên khoa
└ appointments/	# • Đặt lịch khám, danh sách lịch hẹn, hồ sơ khám
└ admin/	# • Giao diện quản trị: dashboard, quản lý danh mục
└ doctor/	# • Dashboard bác sĩ, lịch hẹn, bệnh án
└ patients/ (trong profile/)	# • Trang bệnh nhân, hồ sơ cá nhân
└ auth/	# • Đăng nhập, đăng ký, quên mật khẩu
└ chat/	# • Giao diện chat bệnh nhân - bác sĩ/admin
└ emails/	# • Mẫu email (xác nhận lịch hẹn, thông báo...)
└ welcome.blade.php	# • Trang chào mừng mặc định

Hình 2.2. Kiến trúc giao diện

Dữ liệu (Model/Database).

└ app/Models/	# LỚP DỮ LIỆU (MODEL / DOMAIN)
└ User.php	# Người dùng hệ thống (liên kết role, patient, doctor)
└ Role.php	# Vai trò: admin, doctor, patient
└ Patient.php	# Hồ sơ bệnh nhân
└ Doctor.php	# Hồ sơ bác sĩ
└ Appointment.php	# Lịch hẹn khám bệnh
└ MedicalRecord.php	# Hồ sơ bệnh án sau khám
└ Department.php	# Khoa / chuyên khoa
└ Service.php	# Dịch vụ y tế
└ Disease.php	# Bệnh / chẩn đoán
└ DiseaseSymptom.php	# Liên kết bệnh - triệu chứng
└ ServiceSymptom.php	# Liên kết dịch vụ - triệu chứng
└ LabTest.php	# Phiếu xét nghiệm
└ TestType.php	# Loại xét nghiệm
└ Message.php	# Tin nhắn chat

Hình 2.3. Kiến model

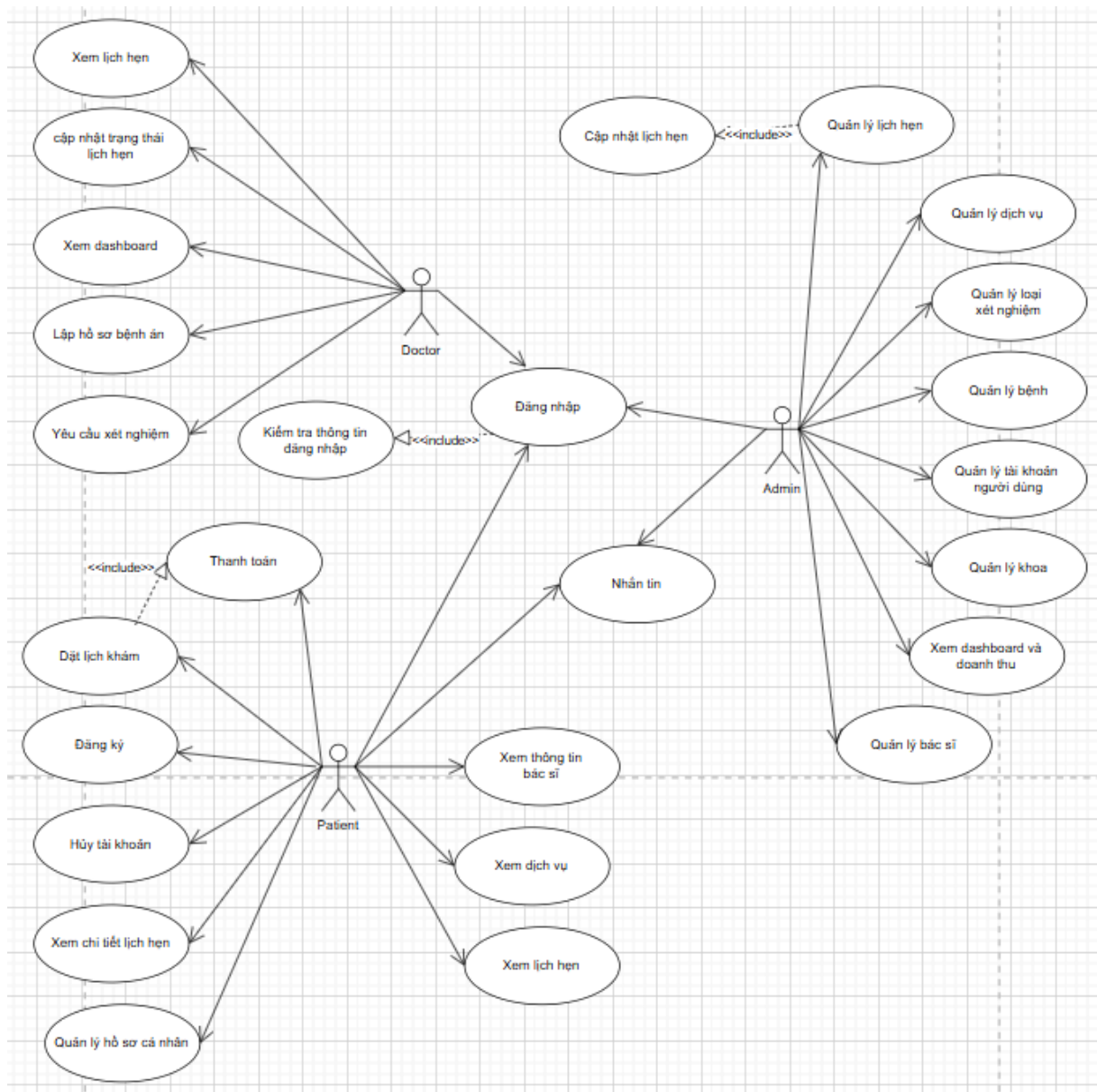
└ database/	# CẤU TRÚC & DỮ LIỆU DB (DATABASE)
└ migrations/	# File migration tạo bảng: users, patients, doctors,
	# appointments, services, departments, lab_tests,...
└ seeders/	# Dữ liệu mẫu: roles, admin, chuyên khoa, dịch vụ, bệnh
└ factories/	# Sinh nhanh dữ liệu test

Hình 2.4. Kiến trúc database

2.2.2. Sơ đồ use-case

Hệ thống Quản lý Bệnh viện Phúc An là một giải pháp toàn diện được phát triển trên nền tảng Laravel, giúp quản lý hiệu quả các hoạt động khám chữa bệnh và chăm sóc sức khỏe. Hệ thống cung cấp đa dạng các chức năng như đặt lịch hẹn khám trực tuyến, quản lý thông tin bệnh nhân, quản lý bác sĩ và dịch vụ y tế, cùng với hệ thống phân quyền người dùng linh hoạt. Với giao diện thân thiện, dễ sử dụng, hệ thống giúp tối ưu hóa quy trình làm việc, giảm thiểu thời gian chờ đợi và thủ tục giấy tờ, đồng thời đảm bảo an toàn và bảo mật thông tin. Hỗ trợ đa nền tảng, hệ thống cho phép người dùng dễ dàng truy cập và sử dụng mọi lúc mọi nơi, góp phần nâng cao chất lượng dịch vụ y tế và mang lại trải nghiệm tốt nhất cho cả bệnh nhân và đội ngũ y bác sĩ.

Use-case tổng quát.



Hình 2.5. Use-case tổng quát

Use-case chức năng đăng nhập.

Bệnh nhân khi đăng nhập vào website, sẽ hiển thị lên trang chủ, form đặt lịch, dịch vụ nổi bật, các chuyên khoa và thông tin Bác sĩ.

Bác sĩ khi đăng nhập vào website, sẽ hiển thị lên tất cả các danh mục quản lý của Bác sĩ và lịch khám của mình.

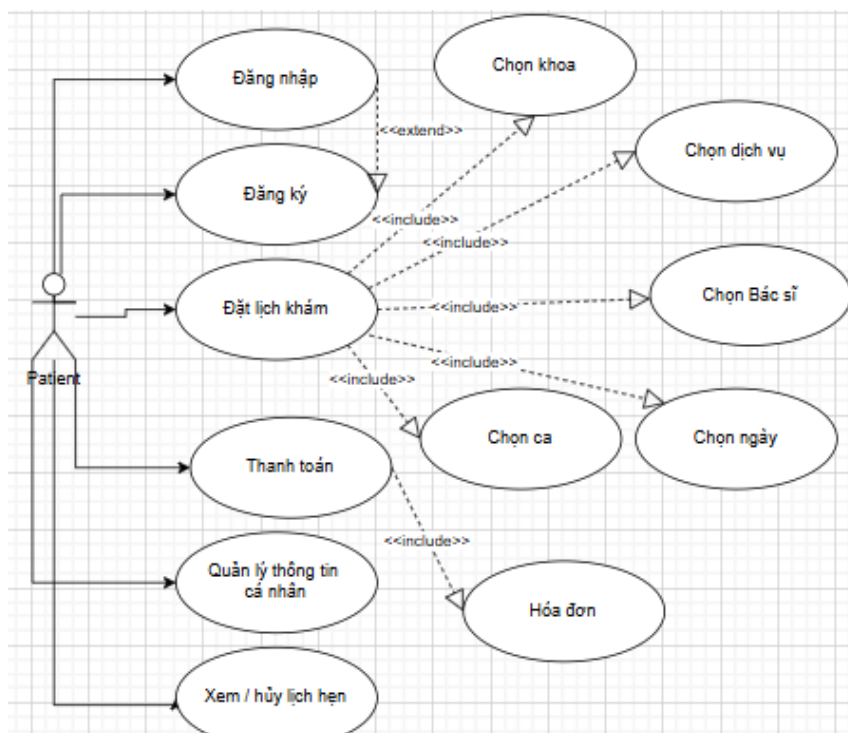
Admin khi đăng nhập vào website, sẽ hiển thị những chức năng của Admin, có quyền thêm sửa xóa và cập nhật thông tin của Bác sĩ, bệnh nhân, dịch vụ, khoa và loại xét nghiệm. Và quản lý lịch khám của Bác sĩ.



Hình 2.6. Use-case chức năng đăng nhập

Use-case chức năng đặt lịch và thanh toán.

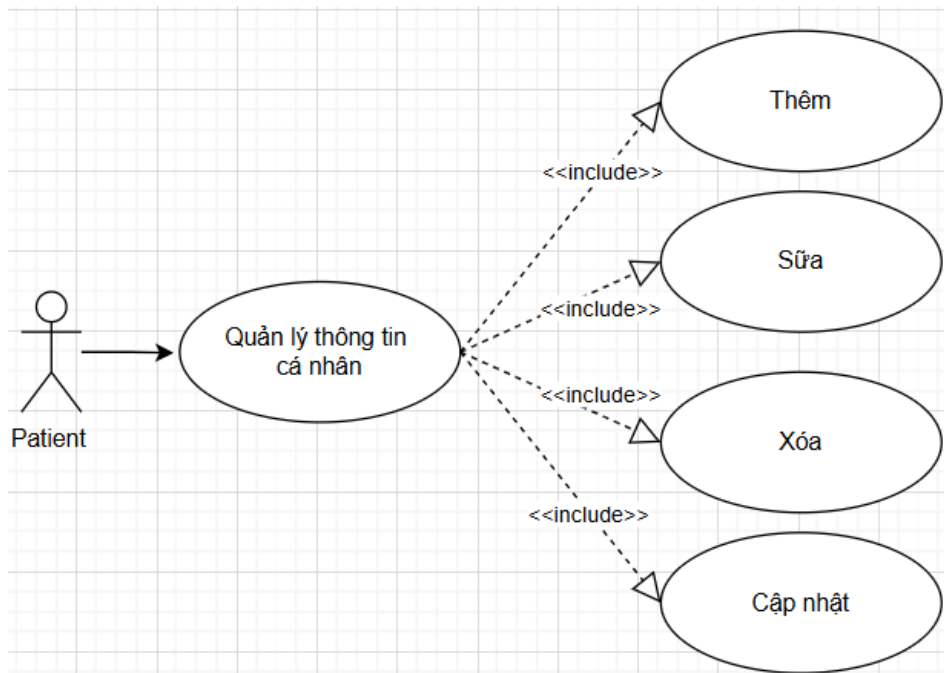
Người dùng truy cập vào Website, chọn dịch vụ bác sĩ và ngày giờ muốn khám, điền thông tin vào form rồi sẽ thanh toán với 80% của tổng số tiền của dịch vụ đó còn 20% thanh toán trực tiếp tại bệnh viện, nếu bệnh nhân sinh vào tháng mà khám bệnh thì được giảm 10% từ tổng 100% của dịch vụ là thanh toán 70% và thanh toán 20% tại bệnh viện. Sau khi đặt lịch thì Admin sẽ duyệt, khi Admin duyệt thì sẽ gửi mail cho bệnh nhân đến thông báo thời gian khám mà bệnh nhân đã chọn.



Hình 2.7. Use-case chức năng đặt lịch và thanh toán

Use-case chức năng thiết lập thông tin của bệnh nhân.

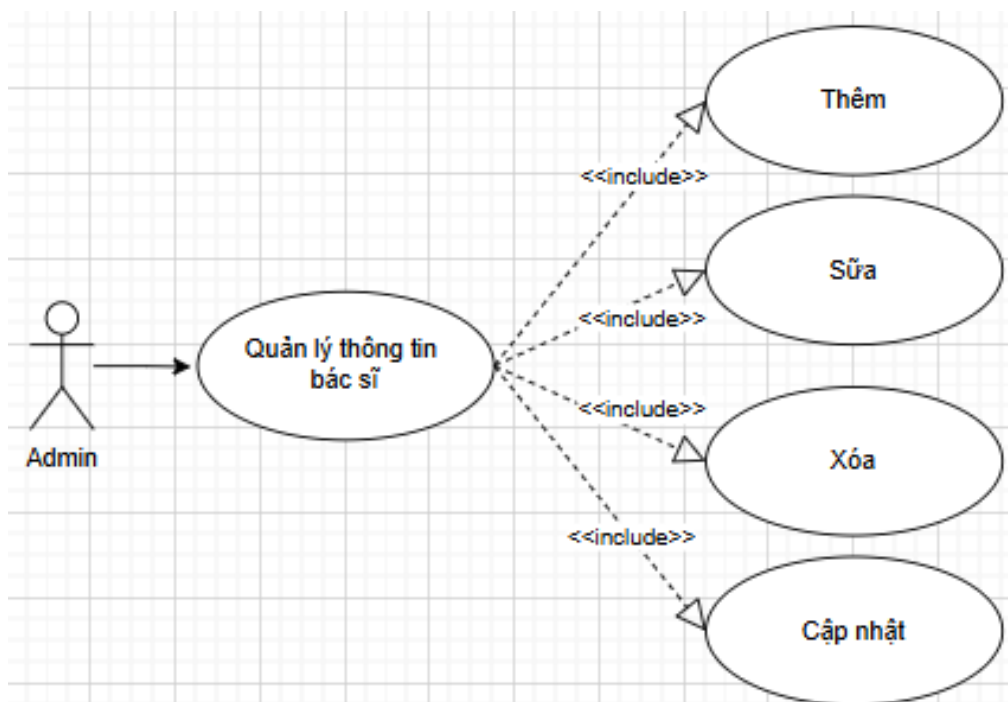
Chức năng này cho phép bệnh nhân tự quản lý thông tin cá nhân của mình. Nếu bệnh nhân không cập nhật thông tin cá nhân thì không thể đặt lịch.



Hình 2.8. Use-case chức năng thiết lập thông tin của bệnh nhân

Use-case chức năng thiết lập thông tin Bác sĩ.

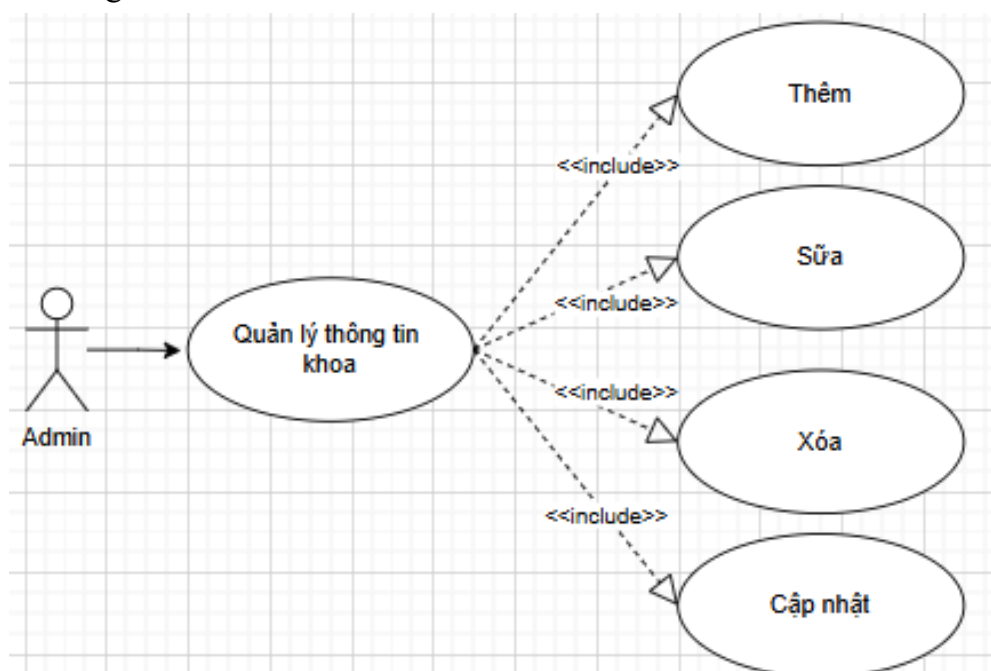
Chức năng này cho phép quản trị viên hoặc Bác sĩ thêm mới hoặc sửa thông tin Bác sĩ vào hệ thống. Thông tin của Bác sĩ sẽ được hiển thị lên trang chính để bệnh nhân có thể xem trước khi đặt lịch.



Hình 2.9. Use-case chức thiết lập thông tin Bác sĩ

Use-case chức năng thiết lập chuyên khoa.

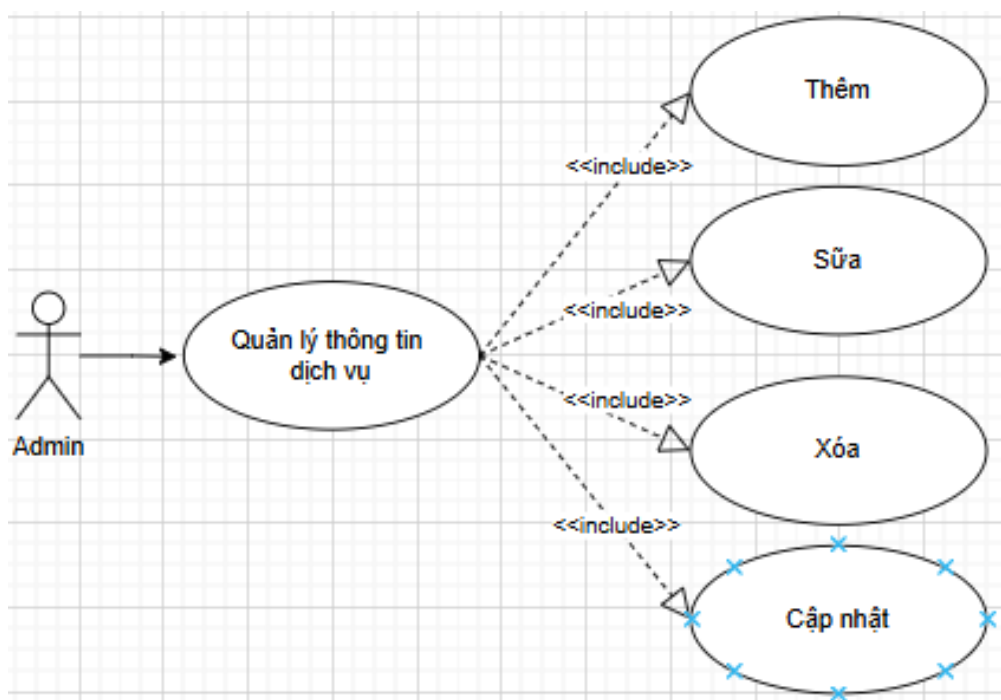
Chức năng này cho phép quản trị viên thêm mới các chuyên khoa và mô tả chi tiết khoa đó khám gì để bệnh nhân có thể xem khi lựa chọn dịch vụ.



Hình 2.10. Use-case chức năng thiết lập chuyên khoa

Use-case chức năng thiết lập dịch vụ.

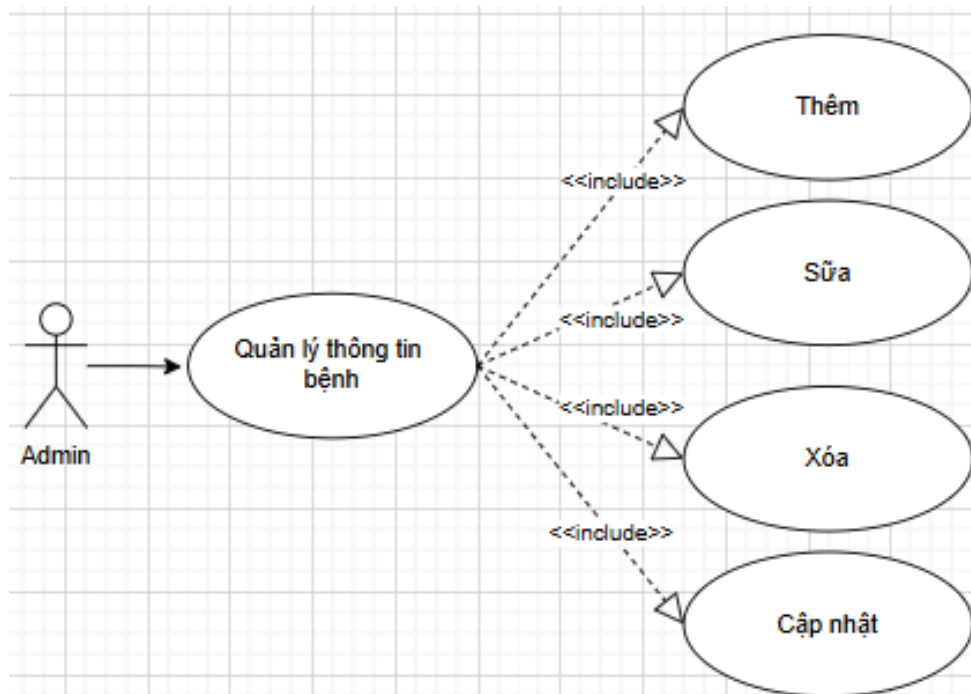
Chức năng này cho phép quản trị viên thêm mới các dịch vụ và khi muốn xem thì bệnh nhân có thể chọn dịch vụ khi đặt lịch.



Hình 2.11. Use-case chức năng thiết lập dịch vụ

Use-case chức năng thiết lập bệnh.

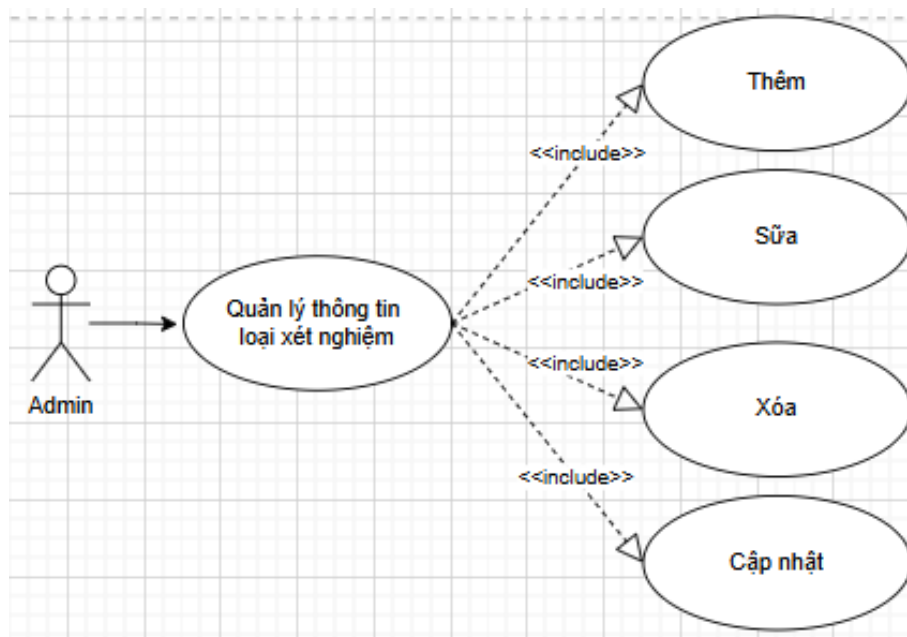
Chức năng này cho phép quản trị viên thêm mới các bệnh khi bệnh nhân mô tả các triệu chứng có trong bệnh này thì có thể tìm hiểu triệu chứng của mình có phải bệnh này không.



Hình 2.12. Use-case chức năng thiết lập bệnh

Use-case chức năng thiết lập loại xét nghiệm.

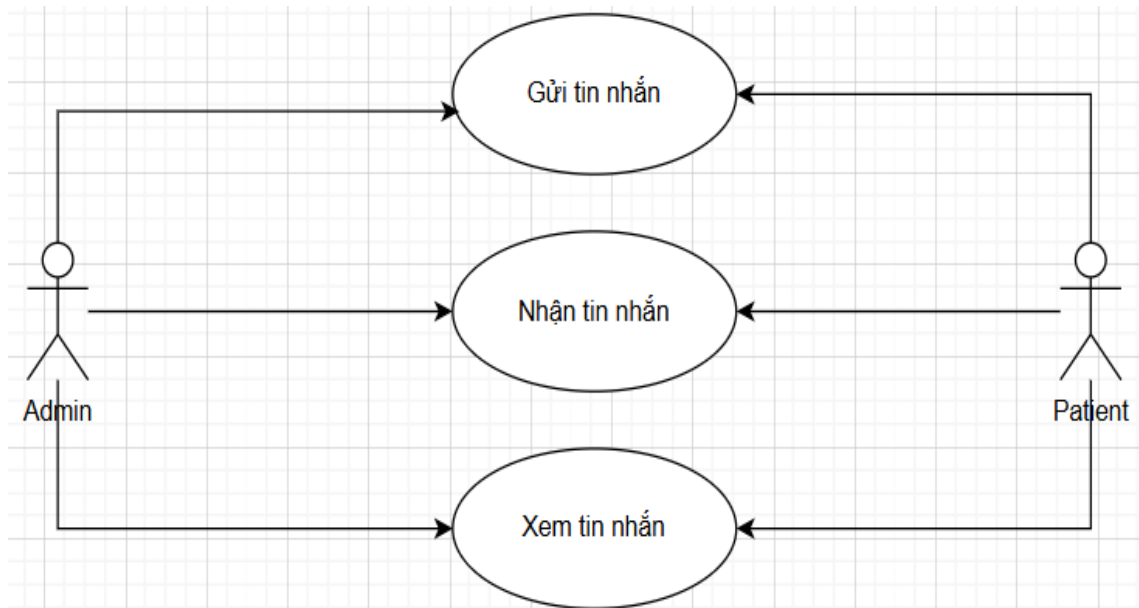
Chức năng này cho phép quản trị viên thêm mới các loại xét nghiệm và khi bệnh nhân yêu cầu loại xét nghiệm thì bác sĩ gửi yêu cầu cho admin khi bệnh nhân xét nghiệm xong admin sẽ gửi lại kết quả cho bác sĩ.



Hình 2.13. Use-case chức năng thiết lập loại xét nghiệm

Use-case chức năng chat box trực tiếp với Admin.

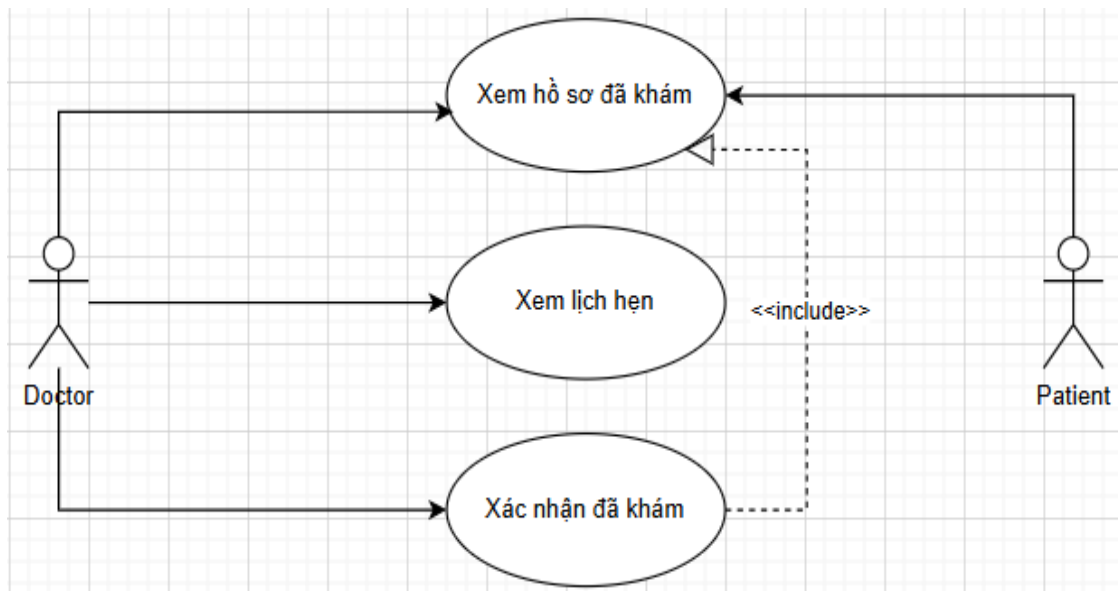
Khi người dùng thắc mắc về ứng dụng thì có chat box với admin sẽ hỗ trợ cho người dùng mọi thông tin liên quan đến ứng dụng.



Hình 2.14. Use-case chức năng chat box trực tiếp với Admin

Use-case chức năng xem lịch khám.

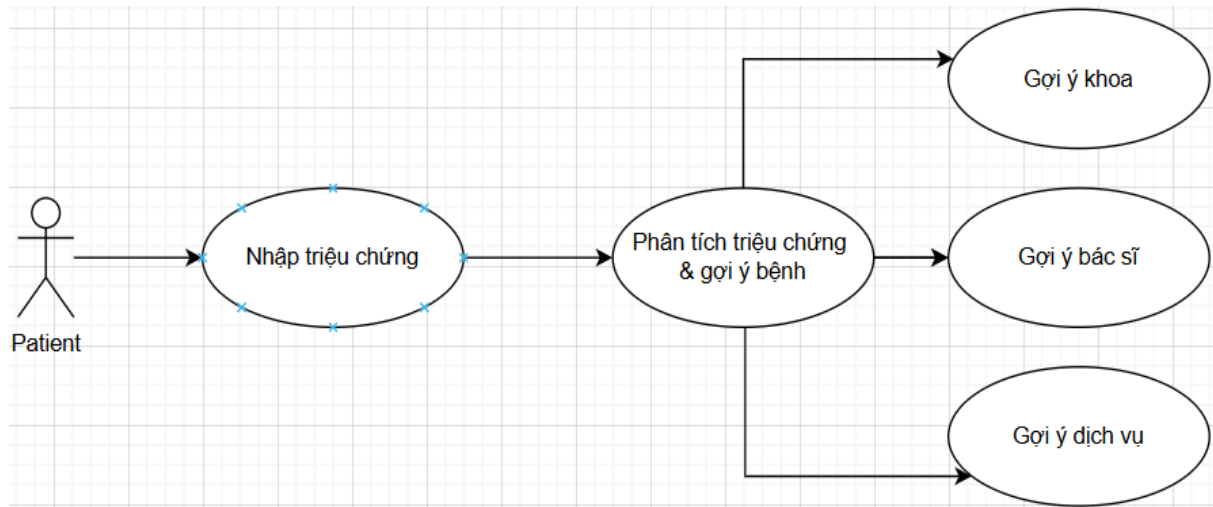
Bác sĩ có thể xem lịch khám và thực hiện khám trên đó và xác nhận đã khám thì bệnh nhân có thể xem được hồ sơ đã khám của mình.



Hình 2.15. Use-case chức năng xem lịch khám

Use-case chức năng tư vấn theo triệu chứng.

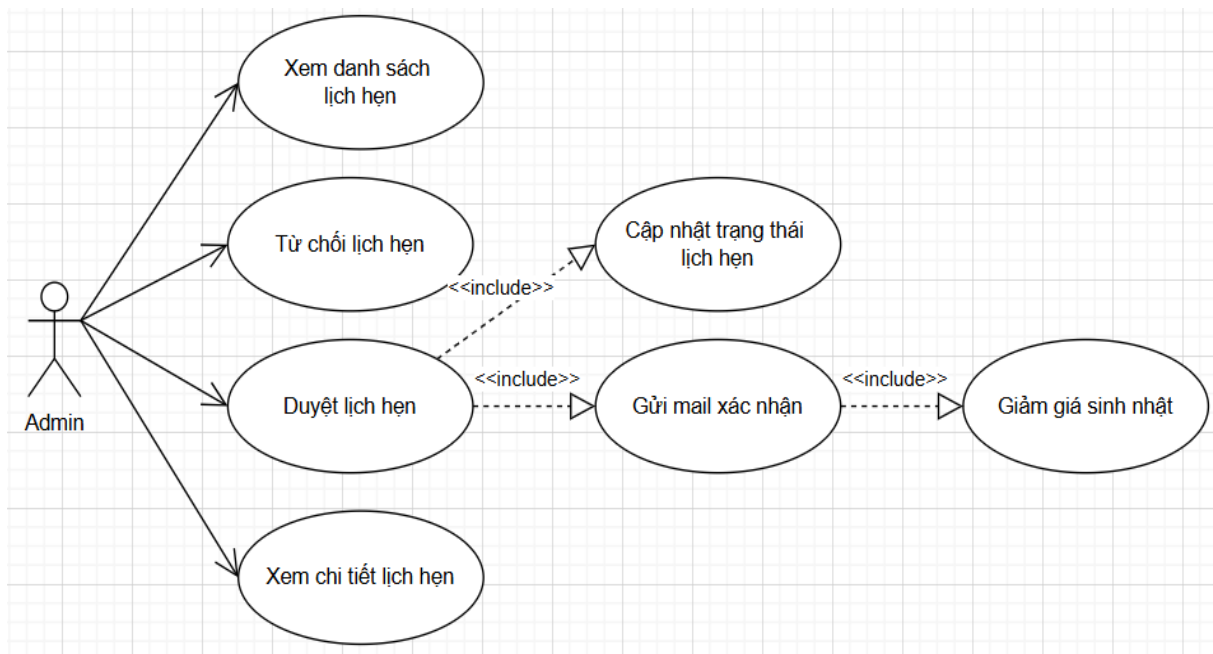
Bệnh nhân nhập triệu chứng mình mắc phải và hệ thống gợi ý bệnh, khoa, bác sĩ, dịch vụ và bệnh nhân có muốn đặt dịch vụ đó không.



Hình 2.16. Use-case chức năng tư vấn theo triệu chứng

Use-case chức năng gửi mail xác nhận lịch hẹn.

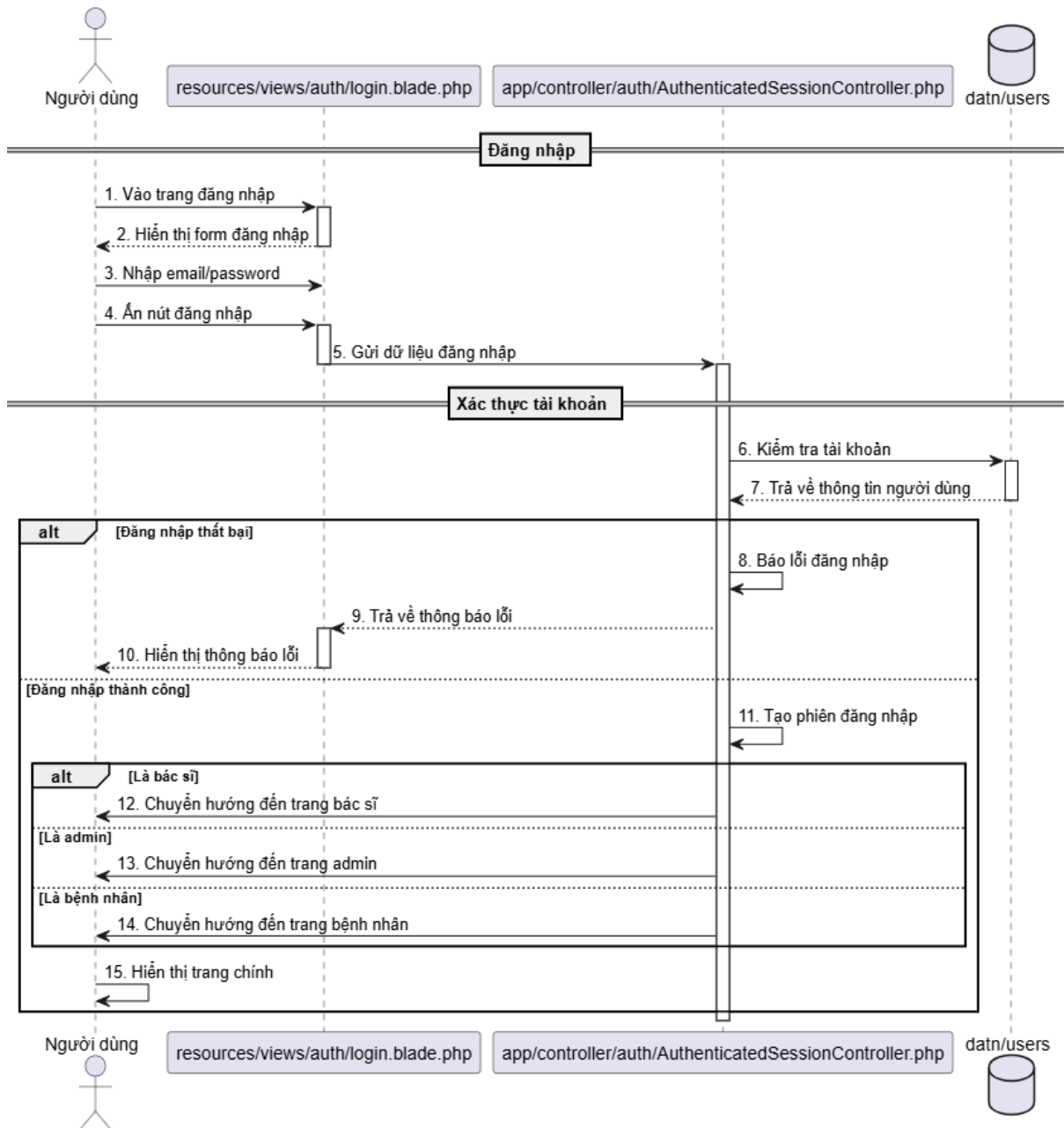
Admin nhận thông báo vào kiểm tra và duyệt khi admin duyệt thì hệ thống sẽ gửi mail thông báo cho bệnh nhân còn khi admin hủy lịch hẹn mà bệnh nhân đã thanh toán thì tiền sẽ được hoàn lại.



Hình 2.17. Use-case chức năng gửi mail xác nhận lịch hẹn

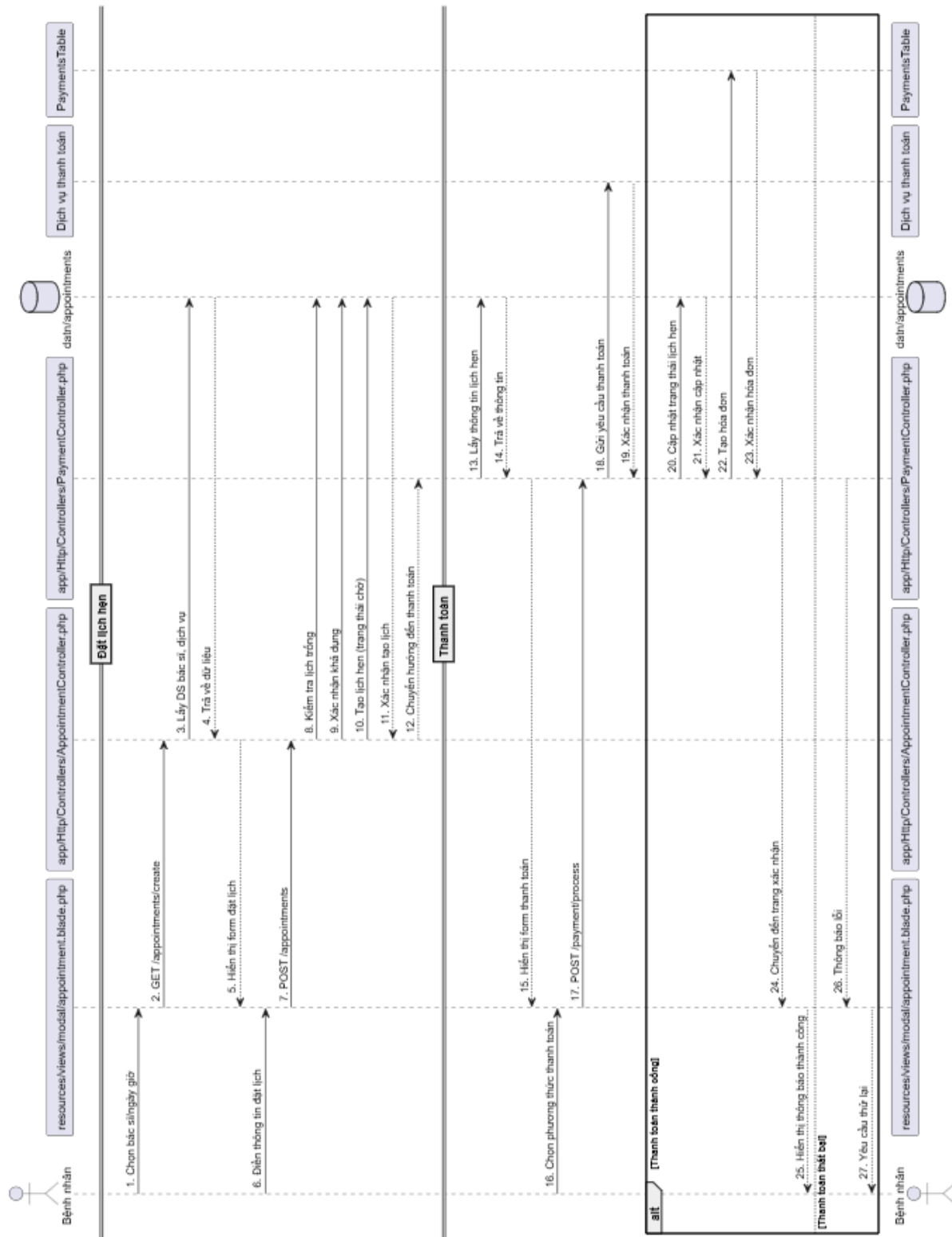
2.2.3. Sơ đồ tuần tự

Sơ đồ tuần tự chức năng đăng nhập.



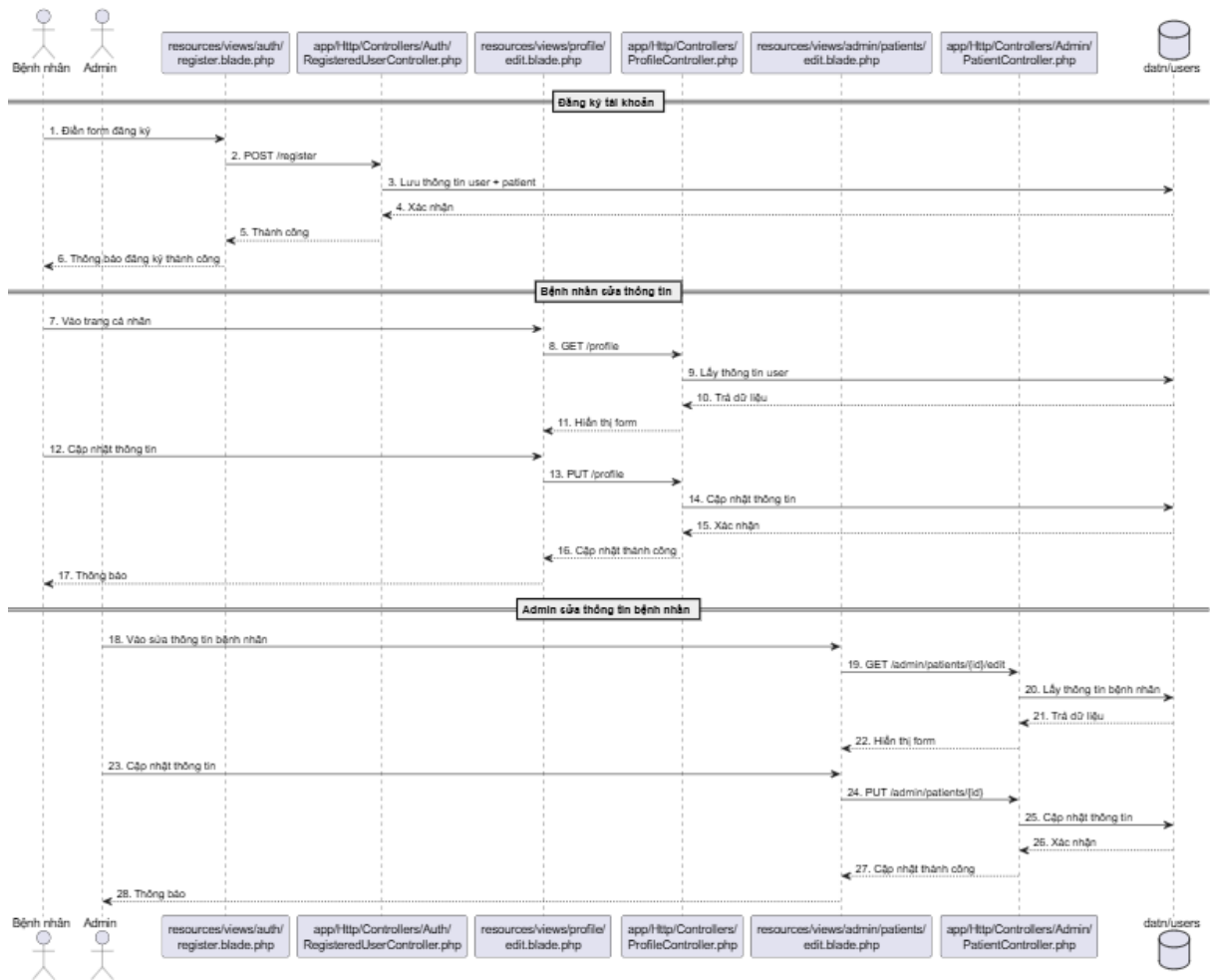
Hình 2.18. Sơ đồ tuần tự chức năng đăng nhập

Sơ đồ tuần tự chức năng đặt lịch và thanh toán.



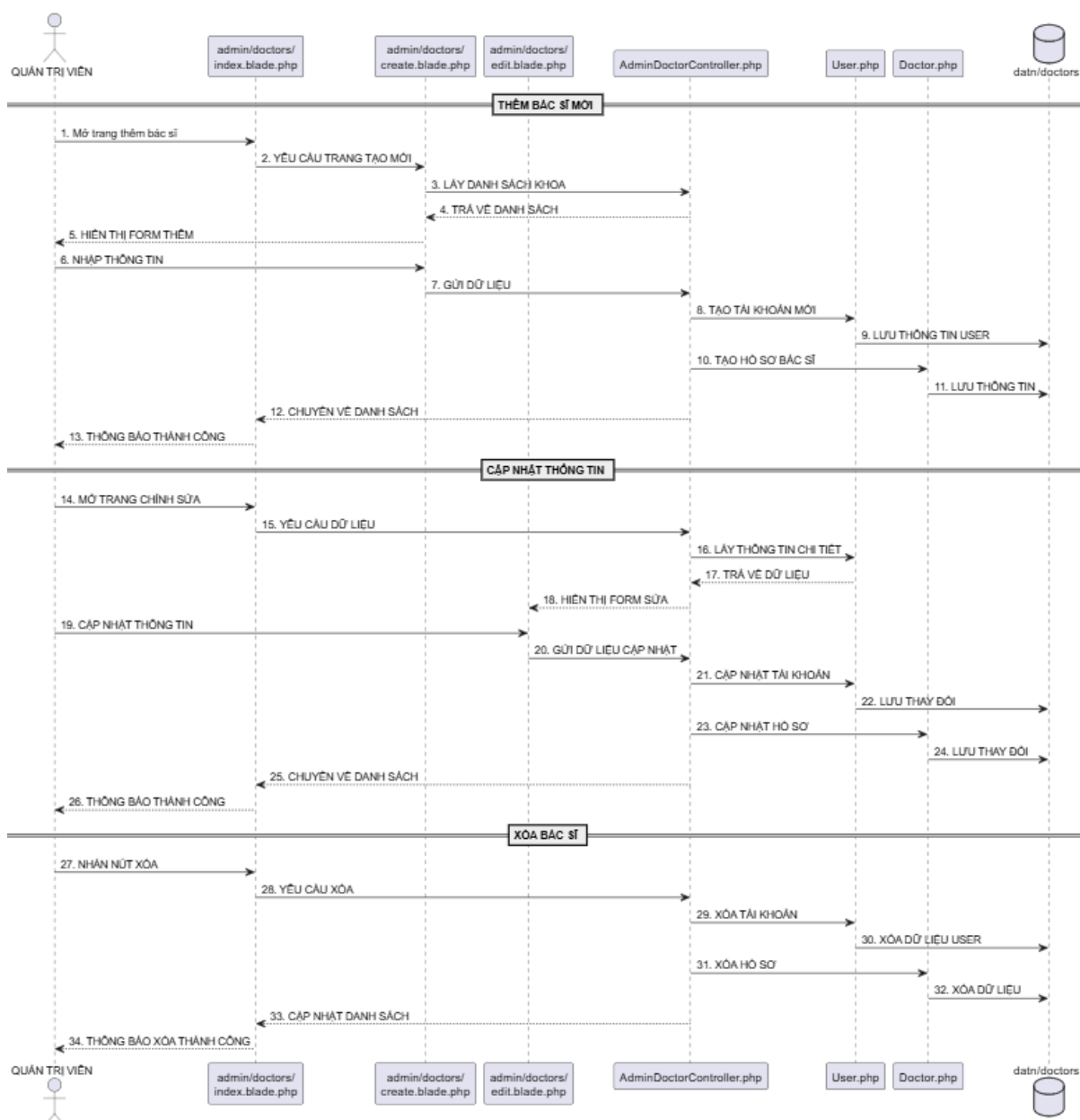
Hình 2.19. Sơ đồ tuần tự chức năng đặt lịch và thanh toán

Sơ đồ tuần tự chức năng thiết lập thông tin của bệnh nhân.



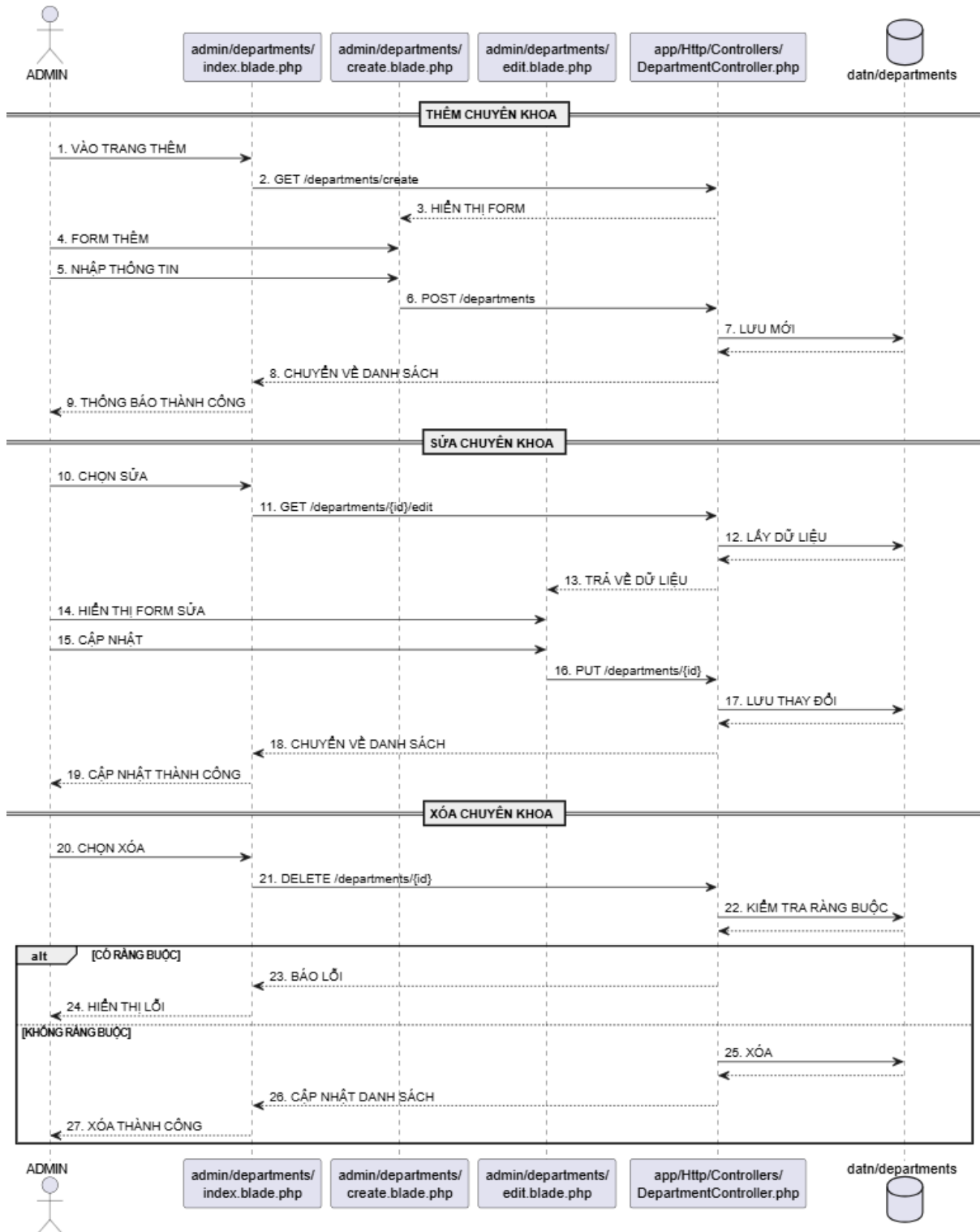
Hình 2.20. Sơ đồ tuần tự chức năng thiết lập thông tin của bệnh nhân

Sơ đồ tuần tự chức năng thiết lập thông tin Bác sĩ.



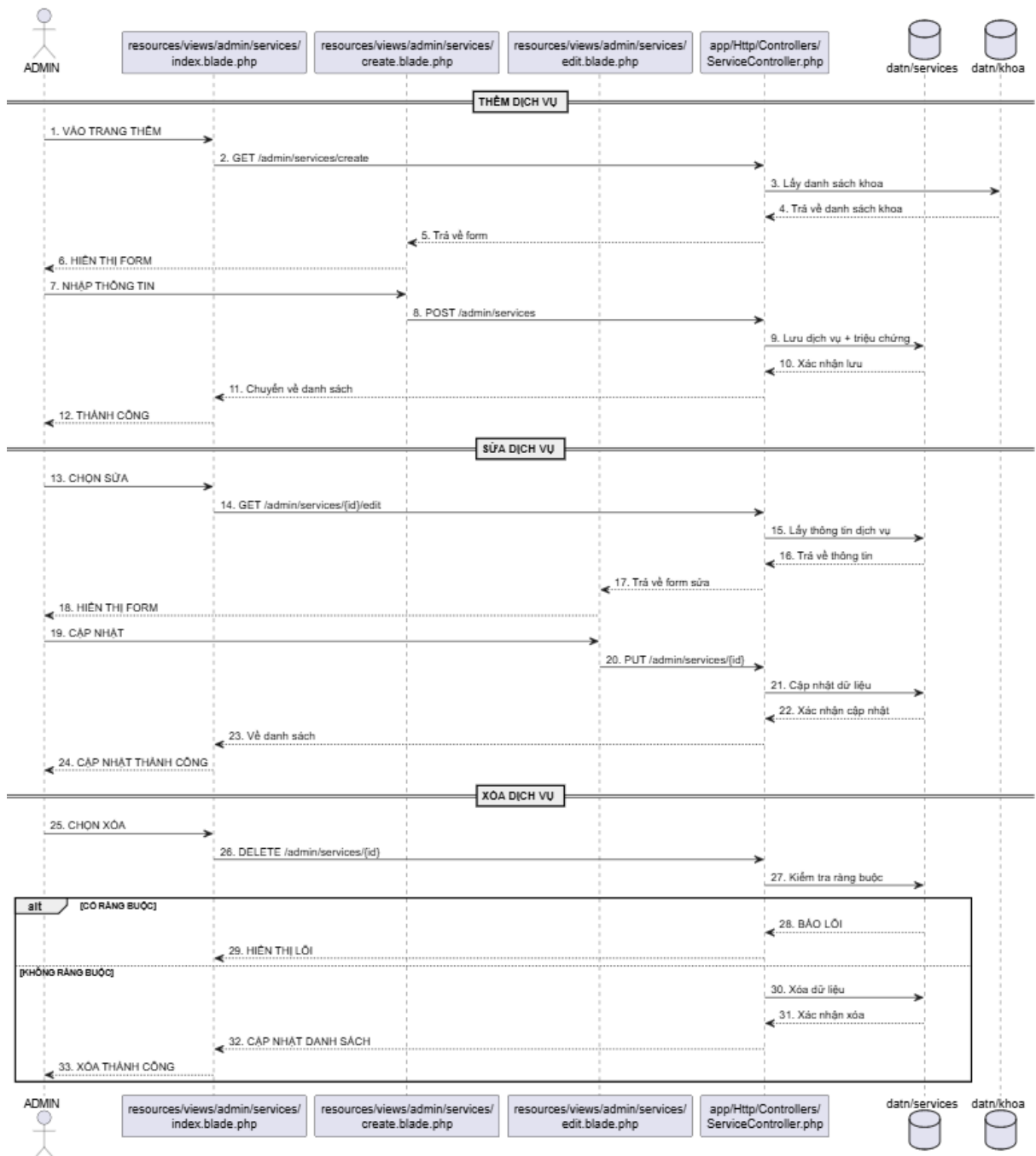
Hình 2.21. Sơ đồ tuần tự chức năng thiết lập thông tin Bác sĩ

Sơ đồ tuần tự chức năng thiết lập nhật chuyên khoa.



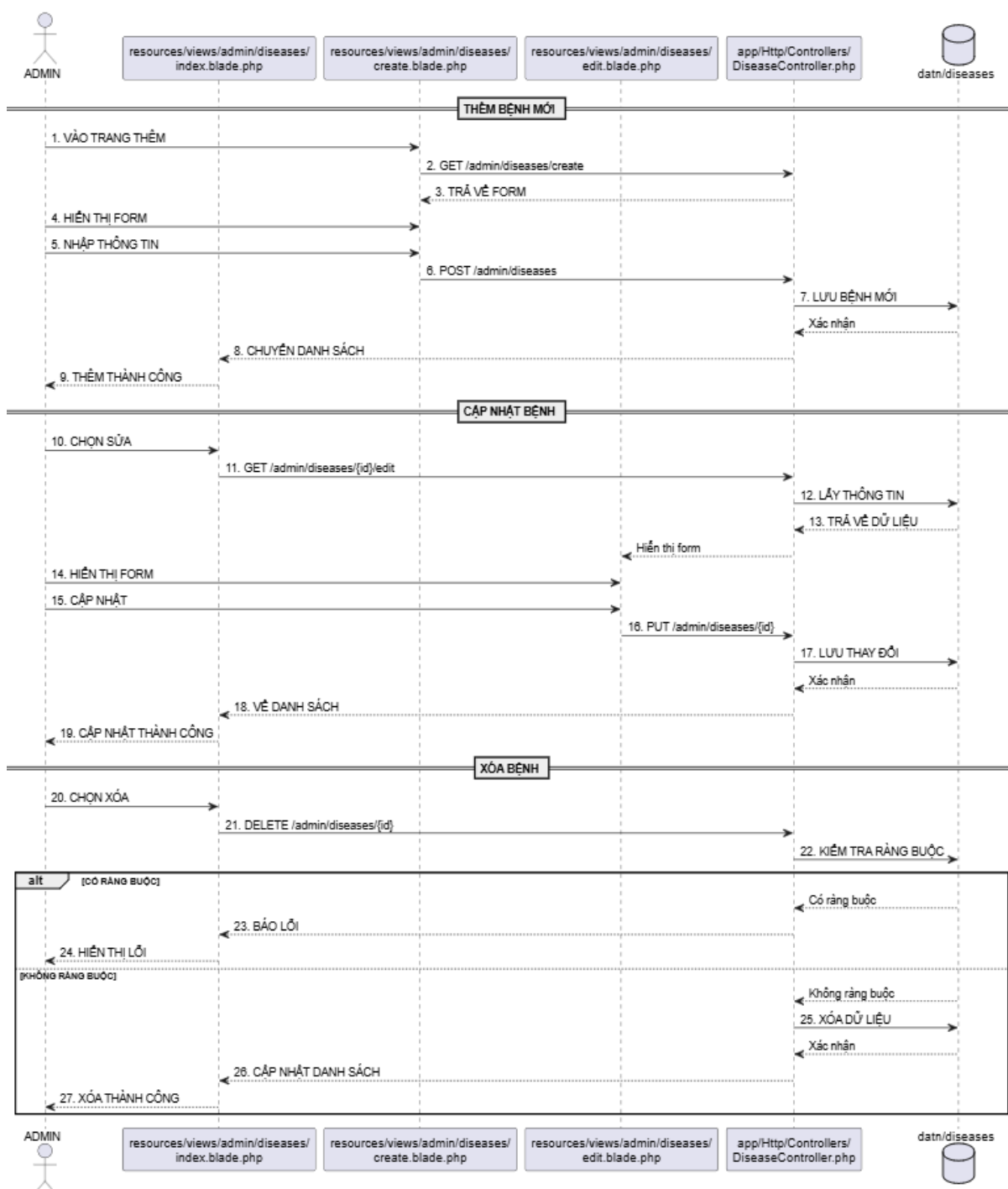
Hình 2.22. Sơ đồ tuần tự chức năng thiết lập chuyên khoa

Sơ đồ tuần tự chức năng thiết lập dịch vụ.



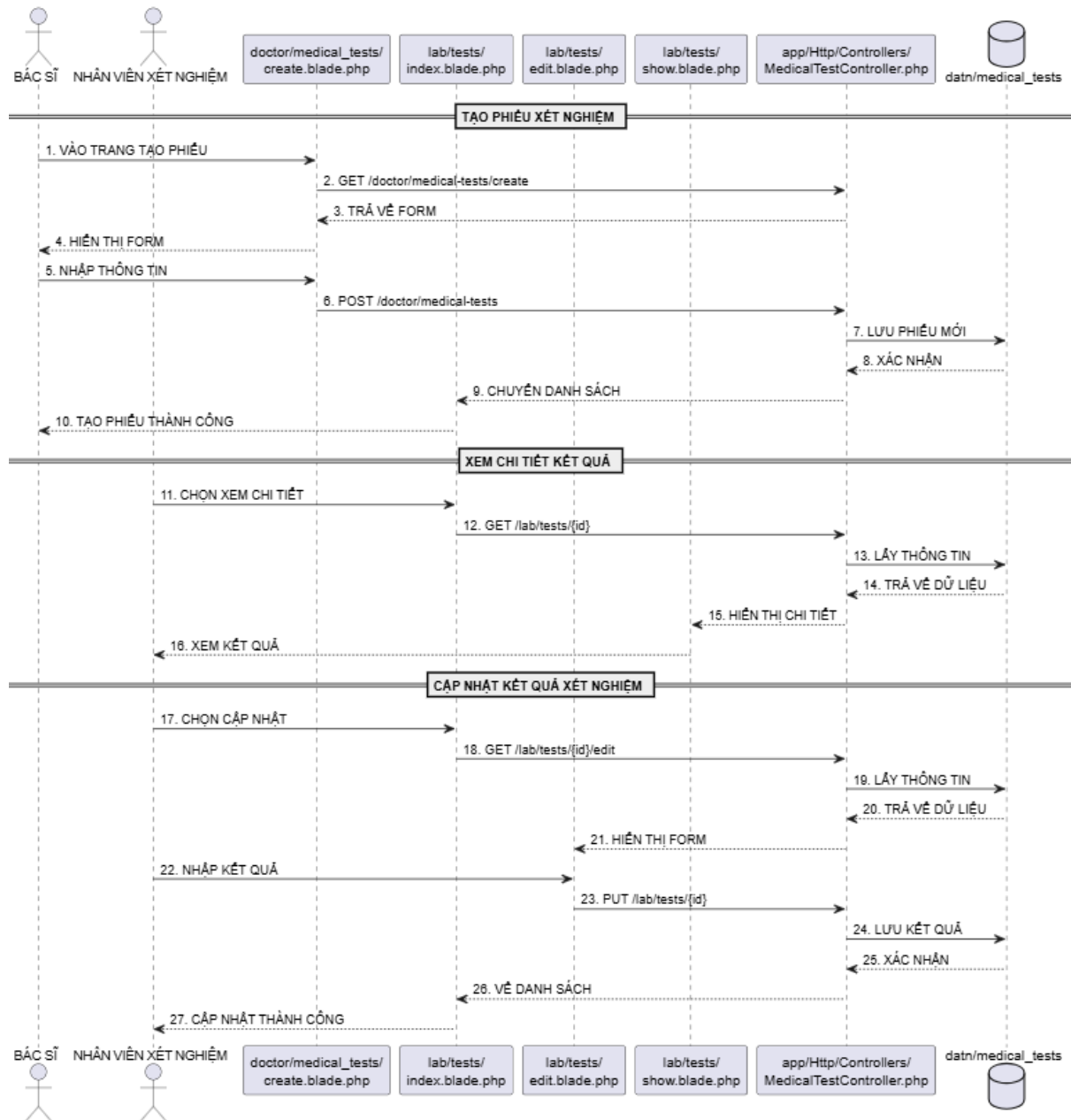
Hình 2.23. Sơ đồ tuần tự chức năng thiết lập dịch vụ

Sơ đồ tuần tự chức năng thiết lập bệnh.



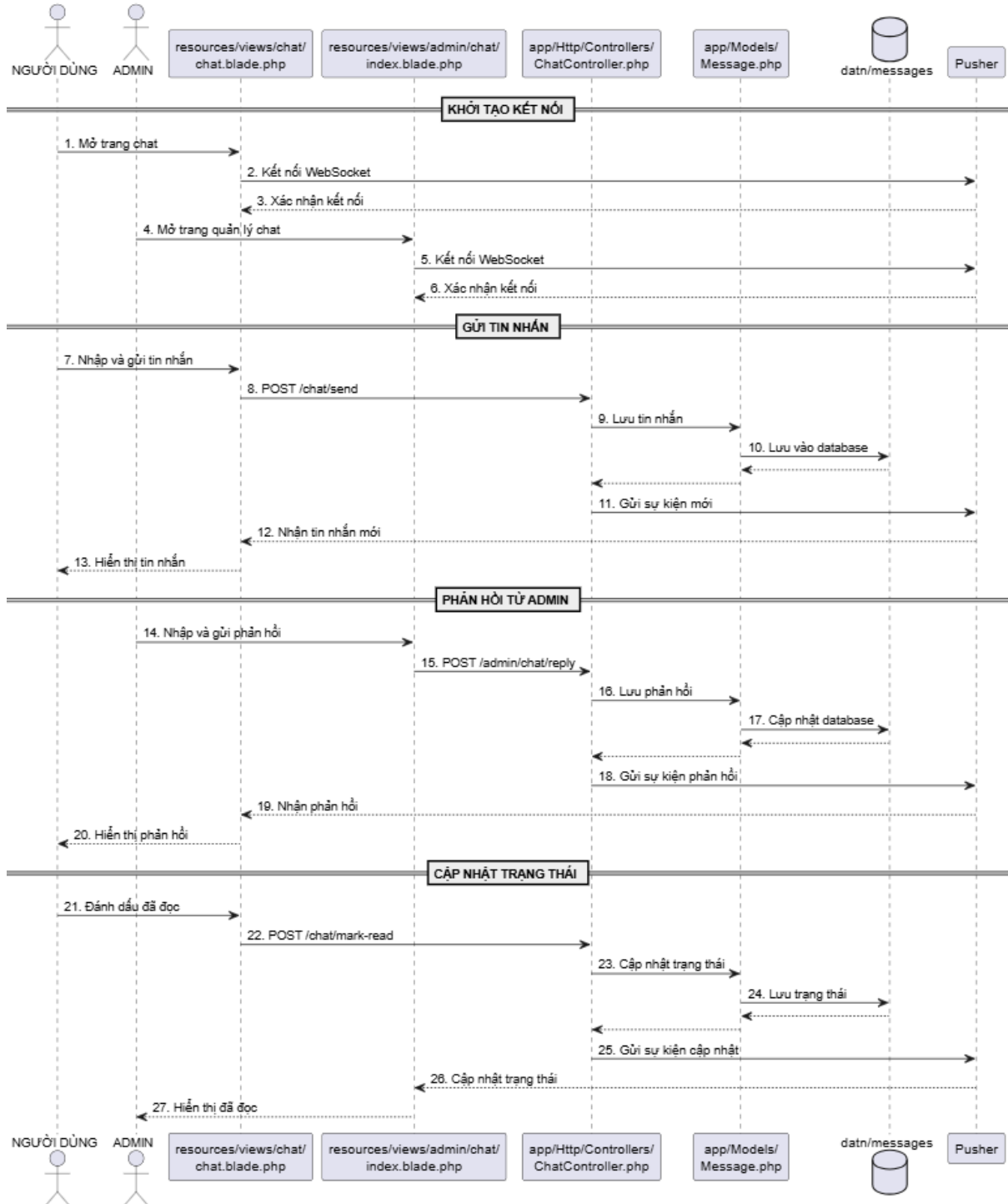
Hình 2.24. Sơ đồ tuần tự chức năng thiết lập nhật bệnh

Sơ đồ tuần tự chức năng thiết lập loại xét nghiệm.



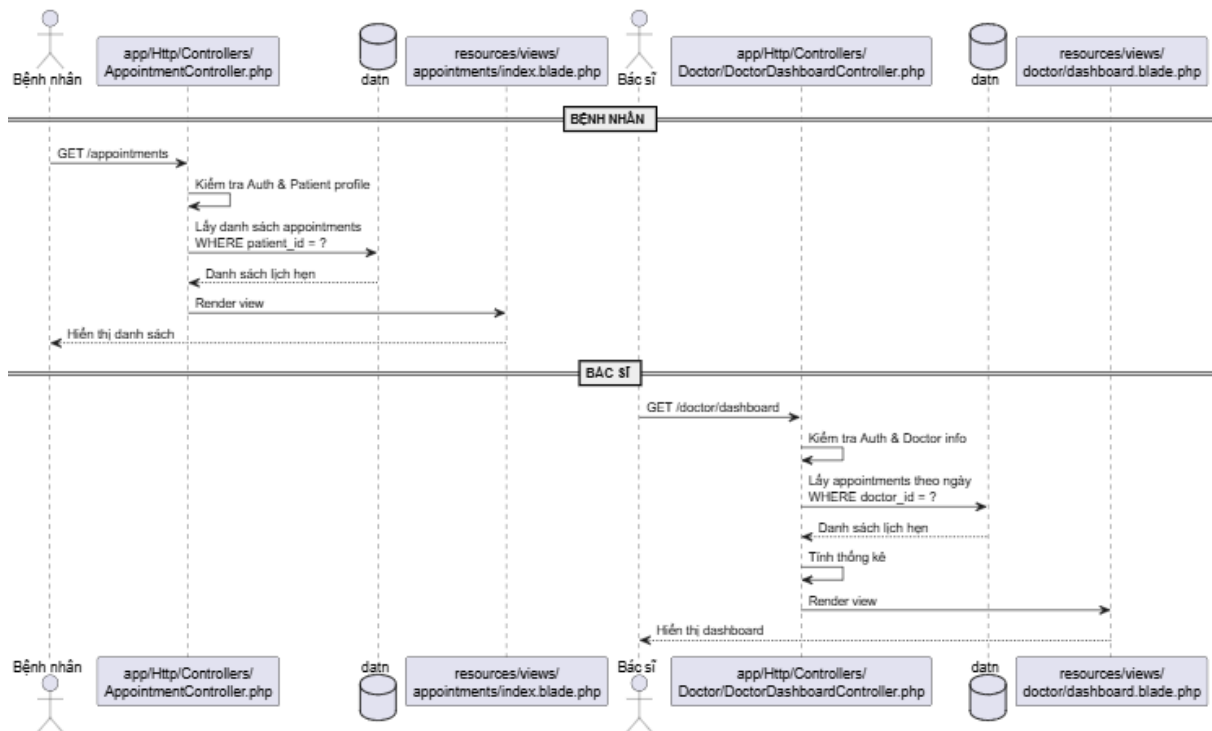
Hình 2.25. Sơ đồ tuần tự chức năng thiết lập loại xét nghiệm

Sơ đồ tuần tự chức năng chatbox trực tiếp với Admin.



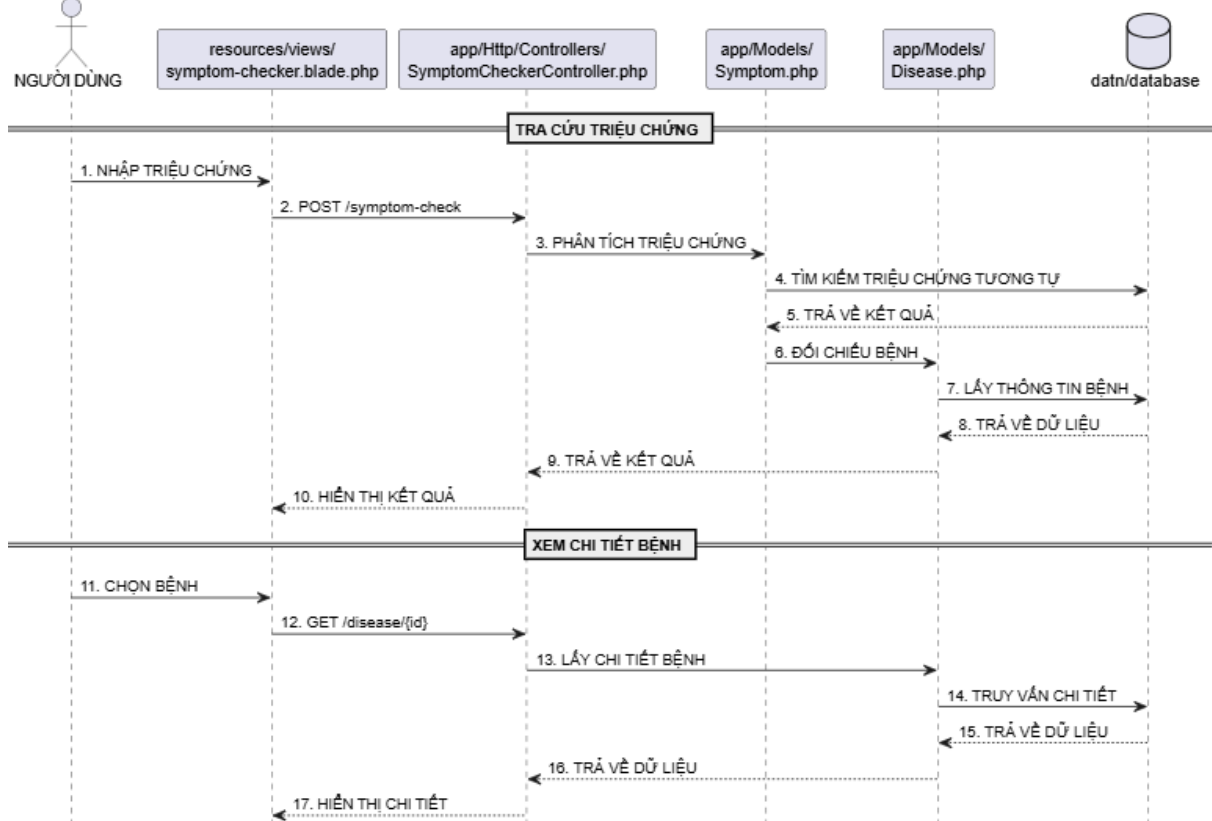
Hình 2.26. Sơ đồ tuần tự chức năng chatbox trực tiếp với Admin

Sơ đồ tuần tự chức năng xem hồ sơ khám.



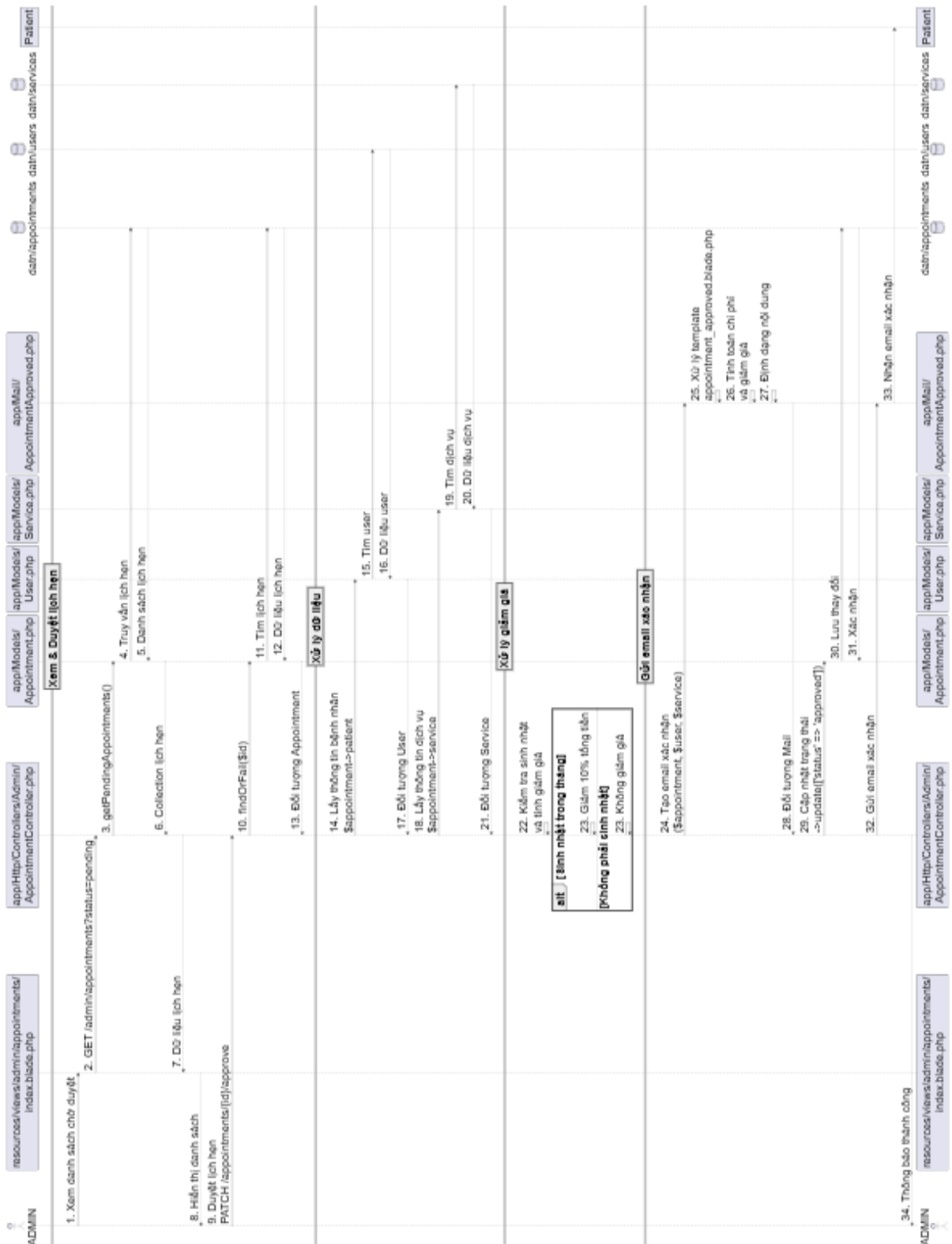
Hình 2.27. Sơ đồ tuần tự chức năng xem lịch khám

Sơ đồ tuần tự chức năng tư vấn theo triệu chứng.



Hình 2.28. Sơ đồ tuần tự chức năng tư vấn theo triệu chứng

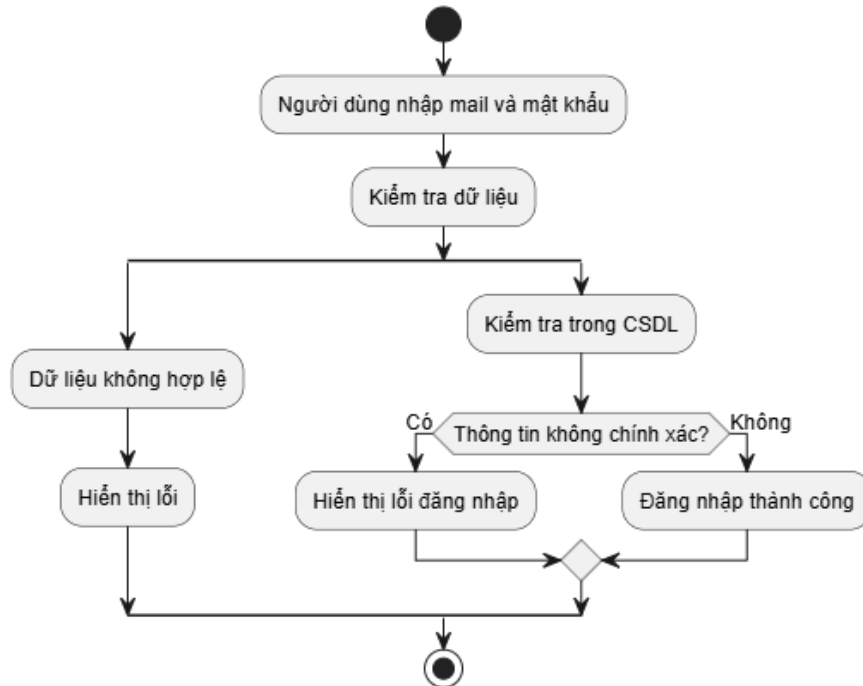
Sơ đồ tuần tự chức năng gửi mail xác nhận lịch hẹn.



Hình 2.29. Sơ đồ tuần tự chức năng gửi mail xác nhận lịch hẹn

2.2.4. Sơ đồ hoạt động

Sơ đồ hoạt động chức năng đăng nhập.



Hình 2.30. Sơ đồ hoạt động chức năng đăng nhập

Sơ đồ hoạt động chức năng thiết lập thông tin của bệnh nhân.



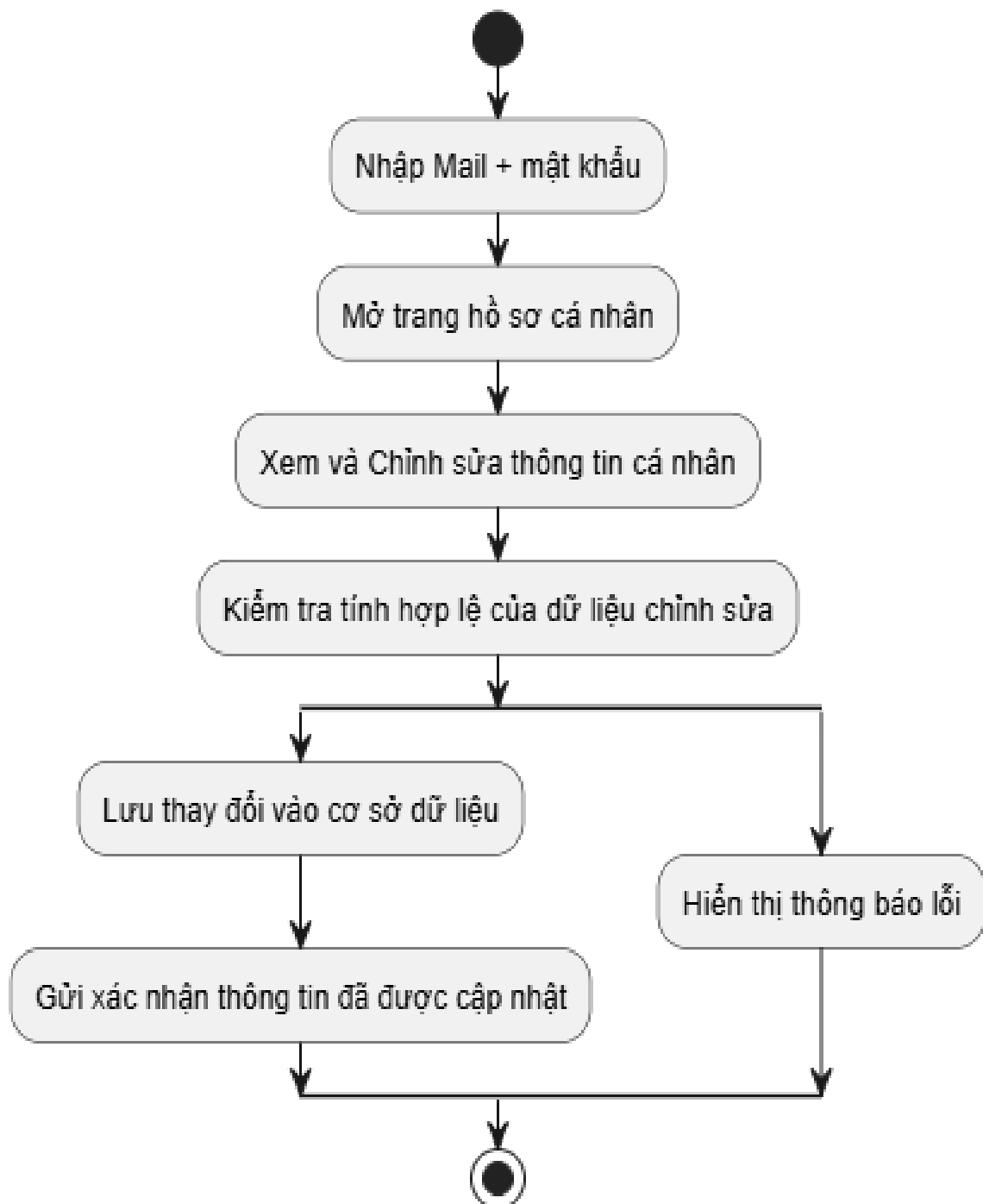
Hình 2.31. Sơ đồ hoạt động chức năng thiết lập thông tin của bệnh nhân

Sơ đồ hoạt động chức năng đặt lịch và thanh toán.



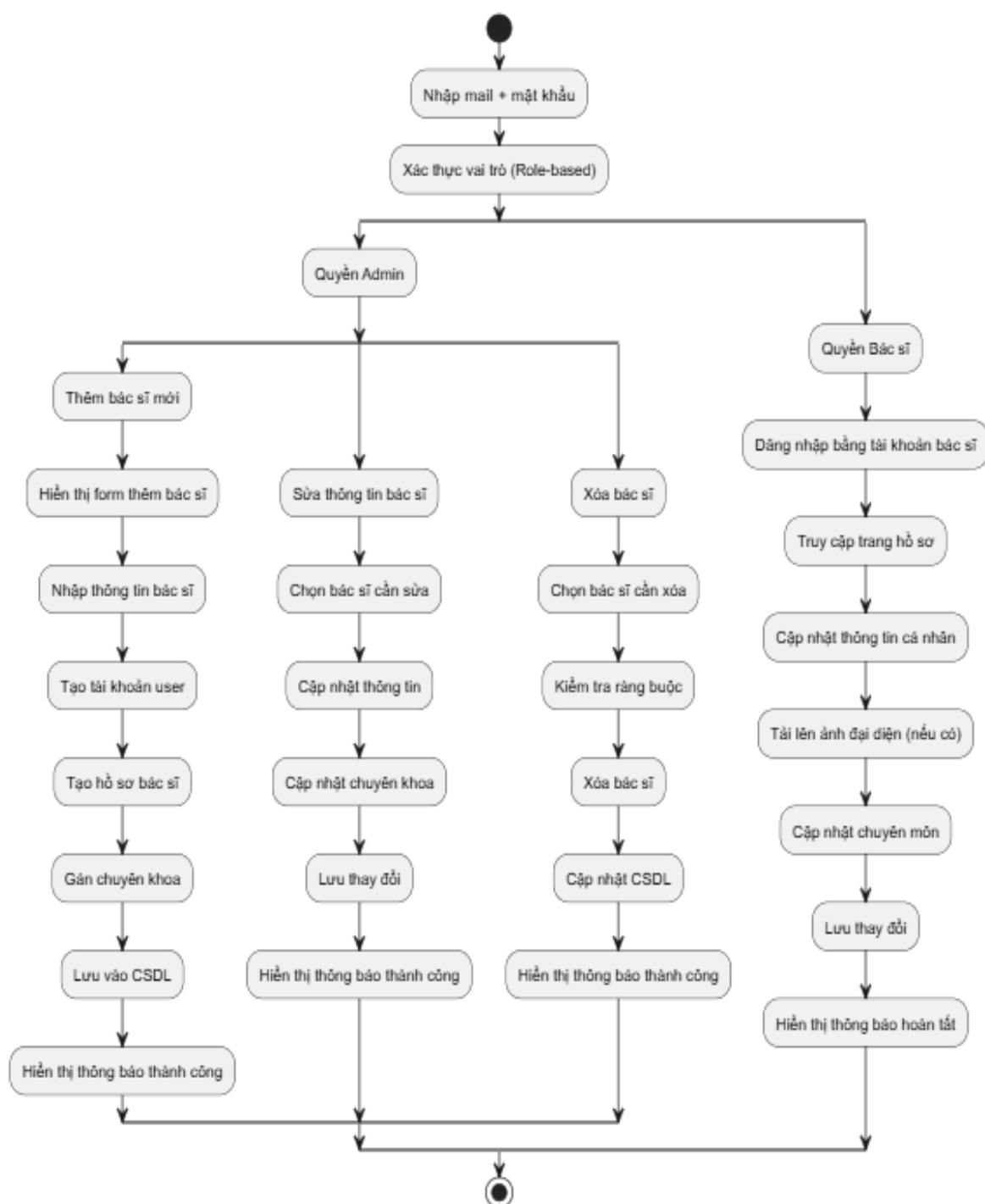
Hình 2.32. Sơ đồ hoạt động chức năng đặt lịch và thanh toán

Sơ đồ hoạt động chức năng thiết lập thông tin của bệnh nhân.



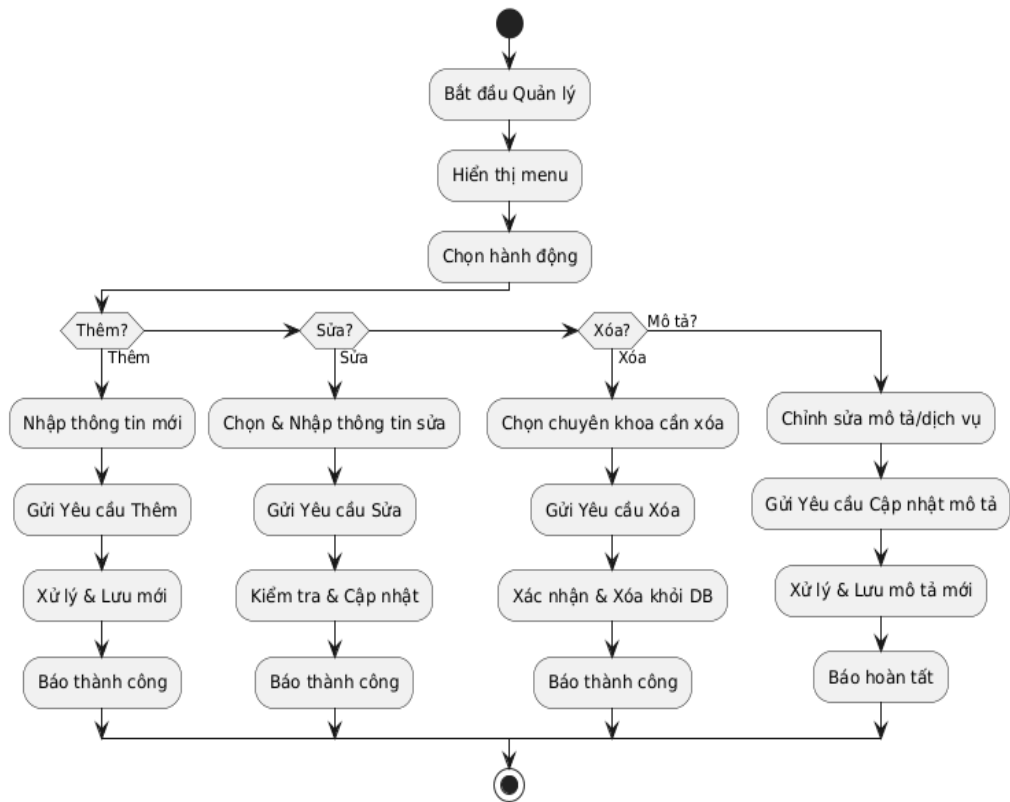
Hình 2.33. Sơ đồ hoạt động chức năng thiết lập thông tin của bệnh nhân

Sơ đồ hoạt động chức năng thiết lập thông tin Bác sĩ.



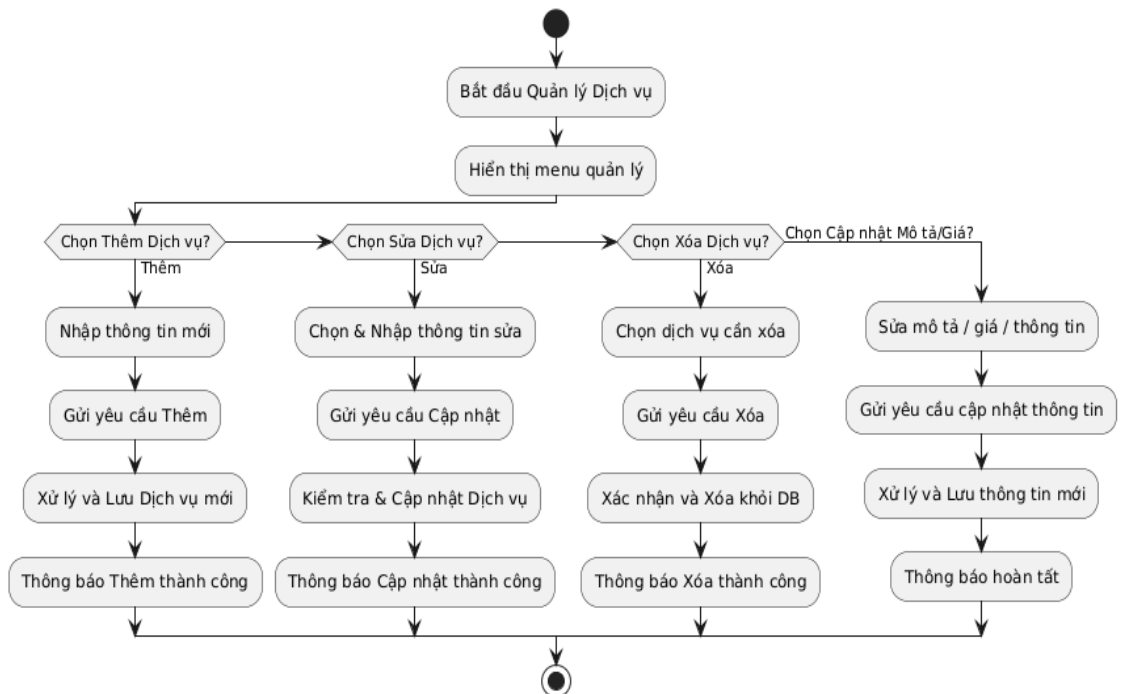
Hình 2.34. Sơ đồ hoạt động chức năng thiết lập thông tin Bác sĩ

Sơ đồ hoạt động chức năng thiết lập chuyên khoa.



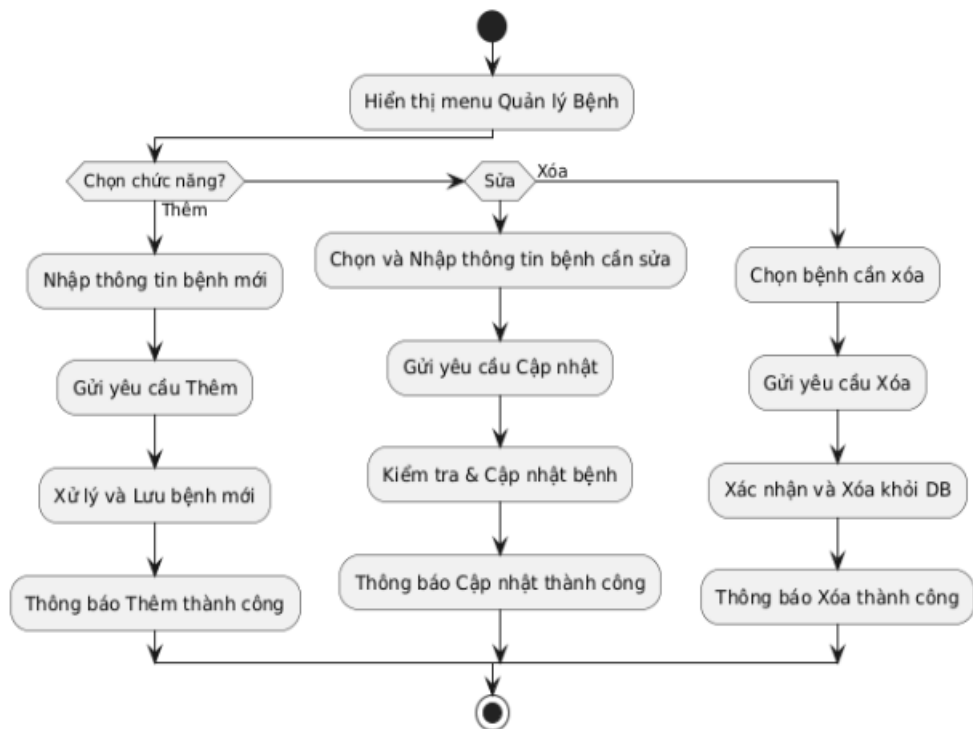
Hình 2.35. Sơ đồ hoạt động chức năng thiết lập chuyên khoa

Sơ đồ hoạt động chức năng thiết lập dịch vụ.



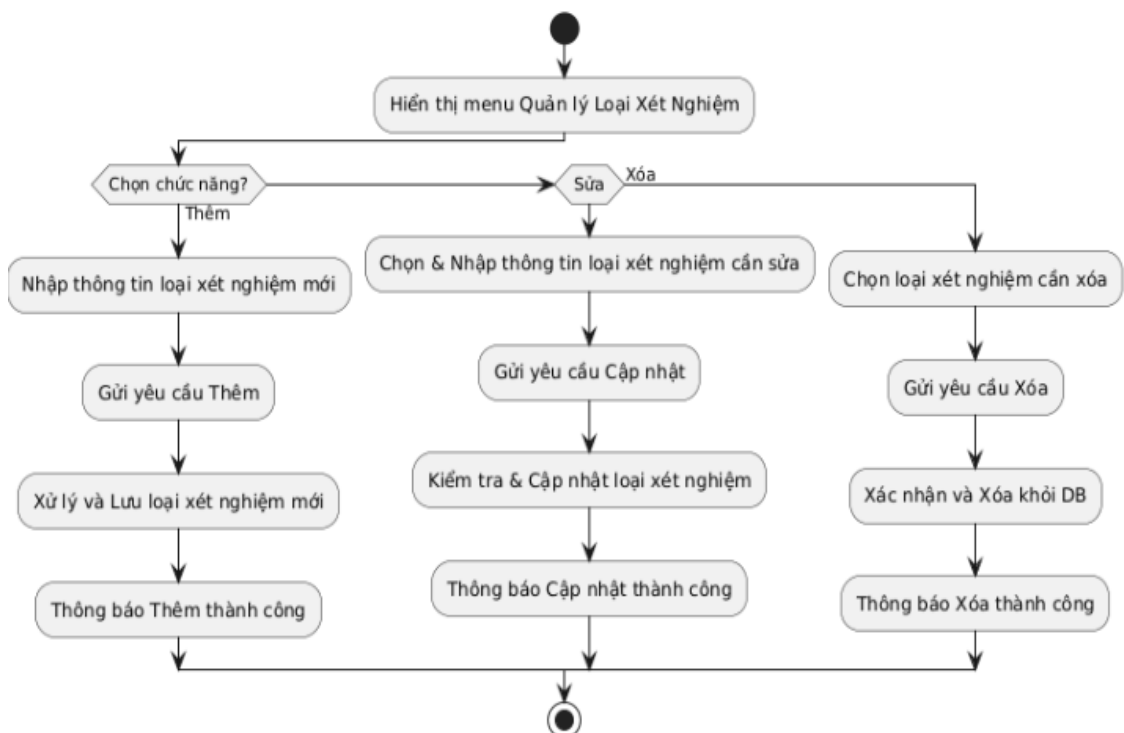
Hình 2.36. Sơ đồ hoạt động chức năng thiết lập dịch vụ

Sơ đồ hoạt động chức năng thiết lập bệnh.



Hình 2.37. Sơ đồ hoạt động chức năng thiết lập bệnh

Sơ đồ hoạt động chức năng thiết lập loại xét nghiệm.



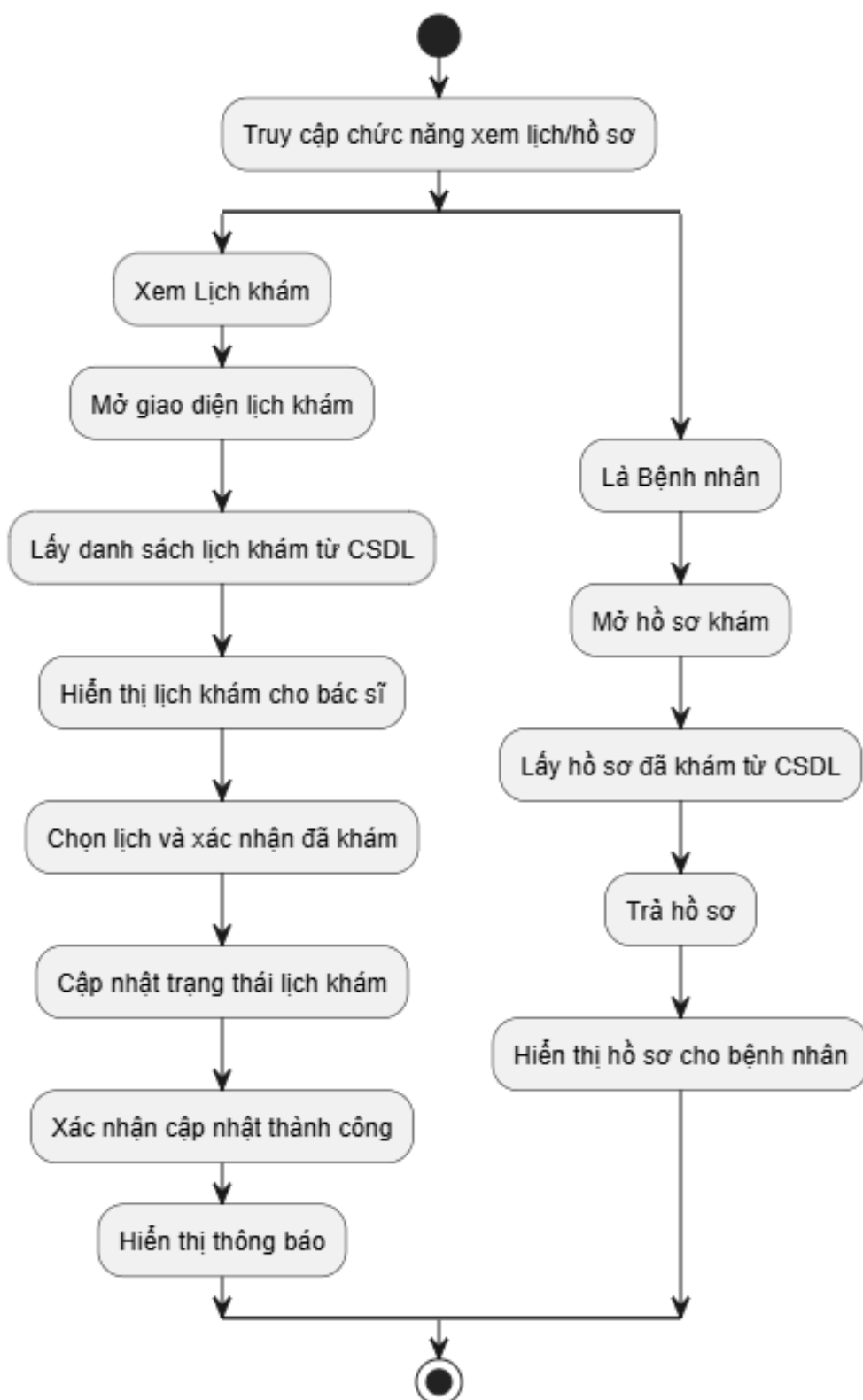
Hình 2.38. Sơ đồ hoạt động chức năng thiết lập loại xét nghiệm

Sơ đồ hoạt động chức năng chatbox trực tiếp với Admin.



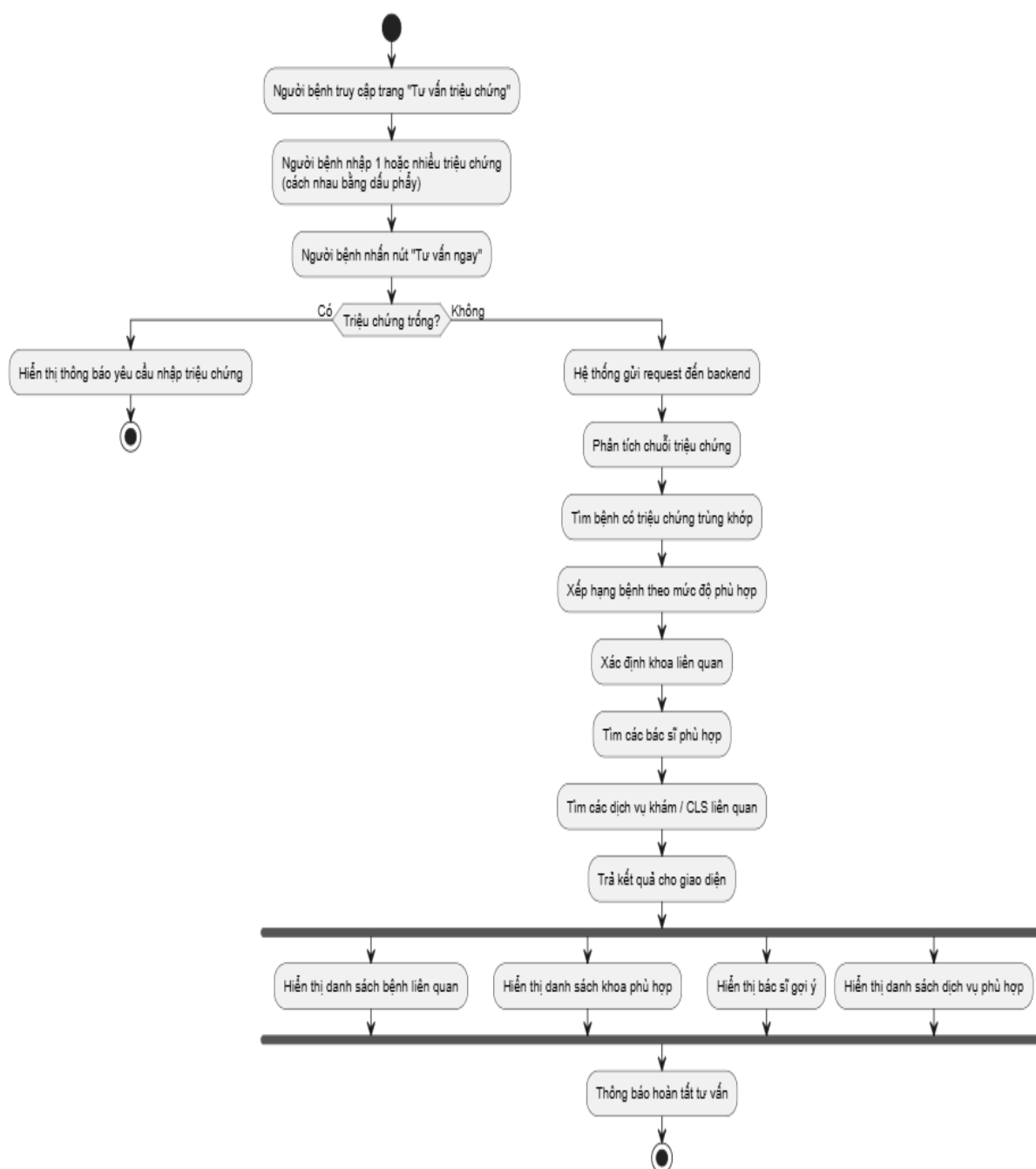
Hình 2.39. Sơ đồ hoạt động chức năng chatbox trực tiếp với Admin

Sơ đồ hoạt động chức năng xem lịch khám.



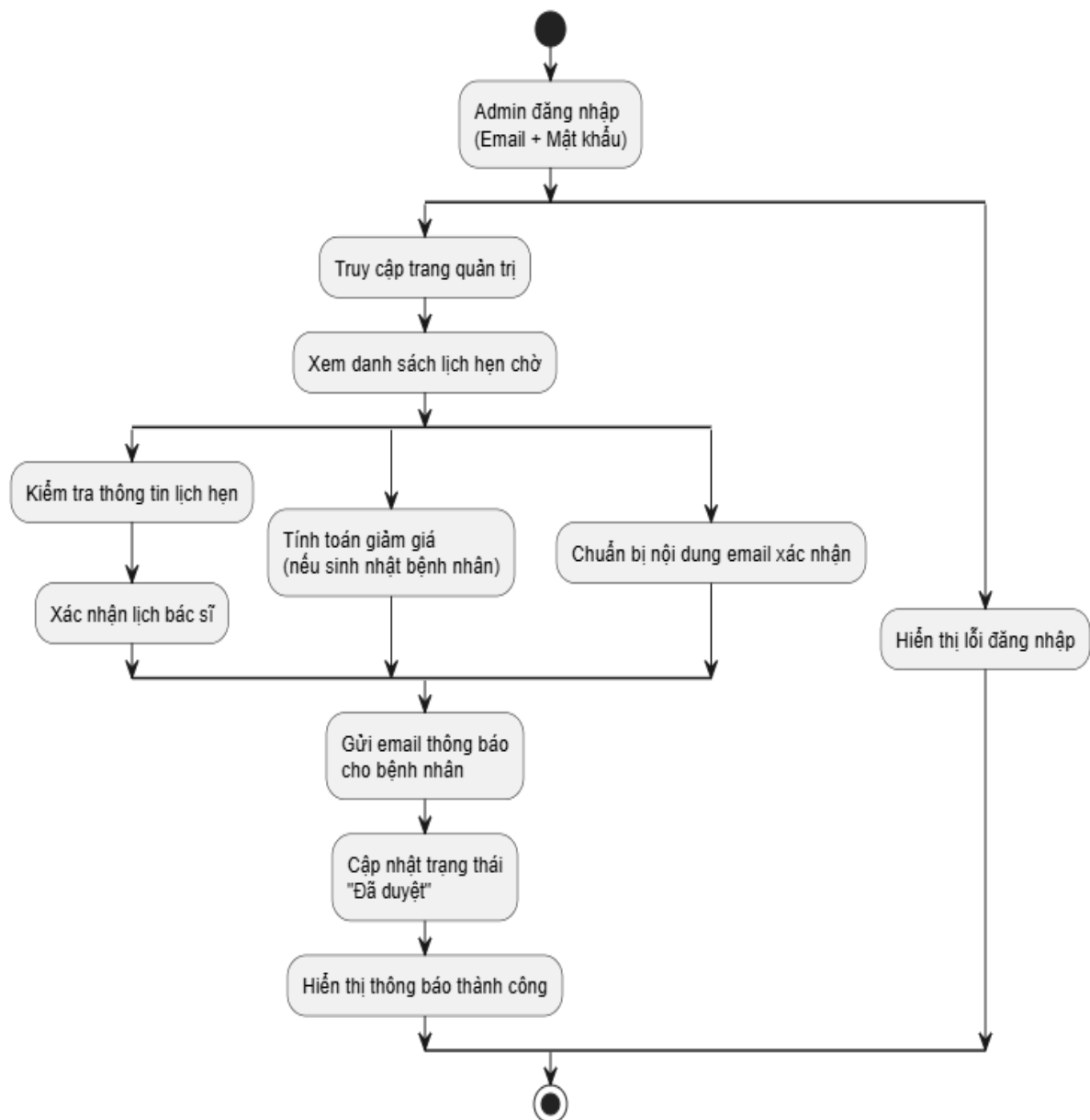
Hình 2.40. Sơ đồ hoạt động chức năng xem lịch khám

Sơ đồ hoạt động chức năng tư vấn theo triệu chứng.



Hình 2.41. Sơ đồ hoạt động chức năng tư vấn theo triệu chứng

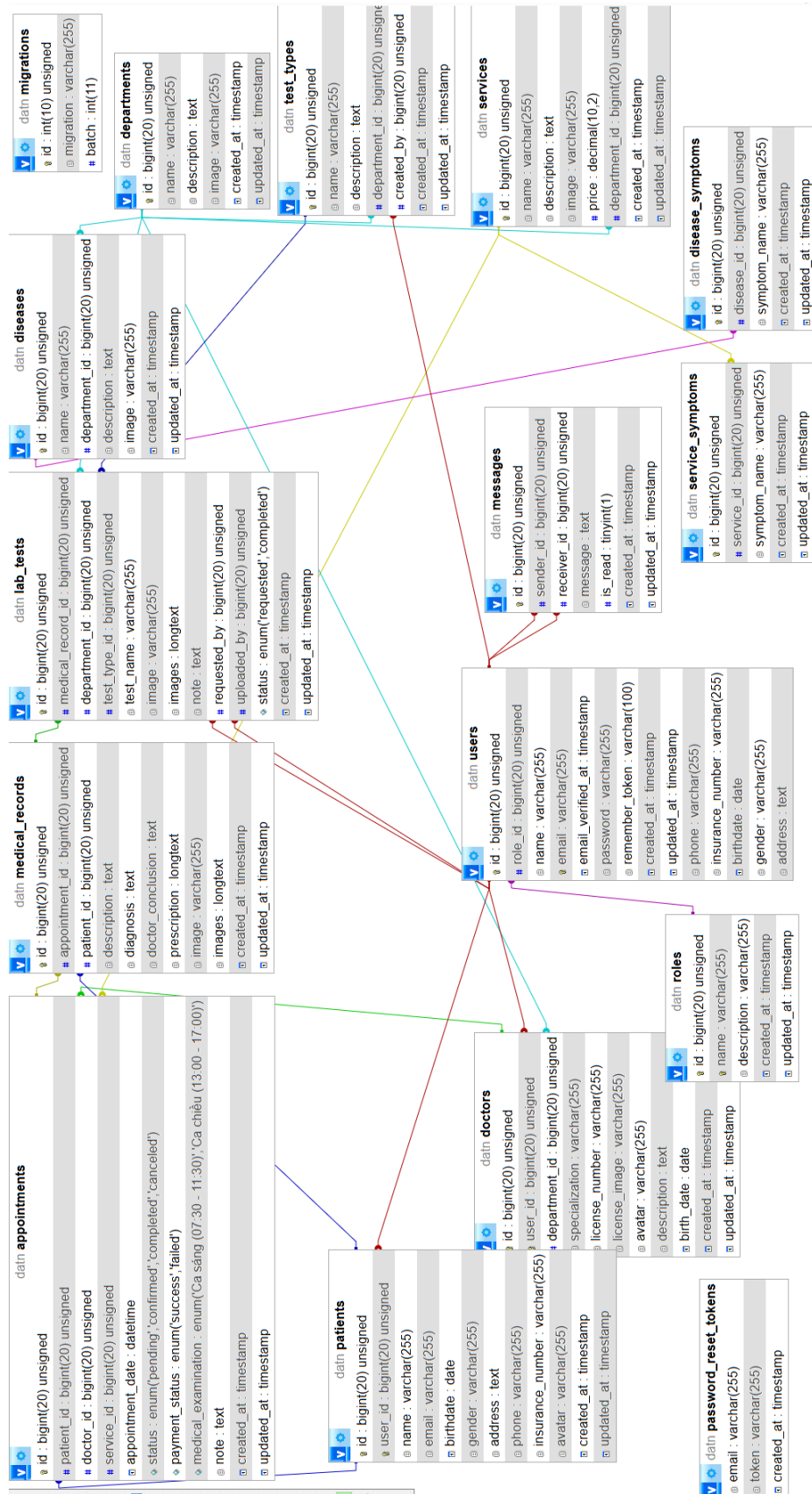
Sơ đồ hoạt động chức năng gửi mail xác nhận lịch hẹn.



Hình 2.42. Sơ đồ hoạt động chức năng gửi mail xác nhận lịch hẹn

2.2.5. Thiết kế lược đồ dữ liệu MySQL

Thiết kế lược đồ dữ liệu MySQL.



Hình 2.43. Thiết kế lược đồ dữ liệu MySQL

Các bảng trong cơ sở dữ liệu.

Bảng 2.1. Mô tả các bảng cơ sở dữ liệu

STT	Tên bảng	Mô tả
1	appointments	Lưu danh sách lịch hẹn (với mọi trạng thái như: đặt và đã thanh toán, đặt chưa thanh toán, chờ duyệt, đã duyệt, đã khám và đã hủy)
2	departments	lưu danh sách các khoa
3	doctors	Lưu danh sách Bác sĩ
4	lab_tests	Lưu ảnh và thông tin đã xét nghiệm
5	medical_records	Lưu hồ sơ bệnh án của bệnh nhân
6	messages	Lưu lịch sử tin nhắn bệnh nhân với admin
7	patients	Lưu danh sách bệnh nhân
8	roles	Phân quyền
9	services	Lưu danh sách dịch vụ
10	test_types	Lưu danh sách xét nghiệm
11	users	Lưu tất cả tài khoản Admin, Bác sĩ, Bệnh nhân

Bảng roles.

Bảng roles chức năng là lưu trữ thông tin về các vai trò trong hệ thống như admin, bác sĩ hay bệnh nhân, dùng để phân quyền truy cập. Bảng này liên kết đến bảng users qua trường role_id, đảm bảo mỗi tài khoản được gán một vai trò cụ thể.

Bảng 2.2. Mô tả bảng roles

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint UNSIGNED	Không	PK	Mã vai trò
2	name	varchar(255)	Không	Unique	Tên vai trò
3	description	varchar(255)	Có		Mô tả vai trò
4	created_at	timestamp	Có		Ngày tạo
5	updated_at	timestamp	Có		Ngày cập nhật

Bảng users.

Bảng USERS quản lý thông tin đăng nhập và hồ sơ cơ bản của tất cả tài khoản trong hệ thống. USERS liên kết với nhiều bảng khác như patients, doctors, messages, test_types và lab_tests.

Bảng 2.3. Mô tả bảng users

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã user
2	role_id	bigint	Không	FK	Vai trò user
3	name	varchar	Không		Tên hiển thị
4	email	varchar	Không	Unique	Email đăng nhập
5	email_verified_at	timestamp	Có		Ngày xác minh email
6	password	varchar	Không		Mật khẩu
7	remember_token	varchar	Có		Token ghi nhớ
8	phone	varchar	Có		Số điện thoại
9	insurance_number	varchar	Có		Mã bảo hiểm
10	birthdate	date	Có		Ngày sinh
11	gender	varchar	Có		Giới tính
12	address	text	Có		Địa chỉ
13	created_at	timestamp	Có		Ngày tạo
14	updated_at	timestamp	Có		Ngày cập nhật

Bảng patients.

Bảng patients lưu hồ sơ chi tiết của bệnh nhân và liên kết với users thông qua user_id. Bệnh nhân có thể có nhiều lịch hẹn trong bảng appointments và nhiều hồ sơ khám trong bảng medical_records.

Bảng 2.4. Mô tả bảng patients

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã bệnh nhân
2	user_id	bigint	Không	FK	Liên kết user
3	name	varchar	Có		Tên bệnh nhân
4	email	varchar	Có		Email liên hệ

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
5	birthdate	date	Có		Ngày sinh
6	gender	varchar	Có		Giới tính
7	address	text	Có		Địa chỉ
8	phone	varchar	Có		Số điện thoại
9	insurance_number	varchar	Có		Mã bảo hiểm
10	avatar	varchar	Có		Ảnh đại diện
11	created_at	timestamp	Có		Ngày tạo
12	updated_at	timestamp	Có		Ngày cập nhật

Bảng doctors.

Bảng doctors quản lý thông tin và hồ sơ của bác sĩ, liên kết với users và departments. Bác sĩ có nhiều lịch hẹn trong appointments và có thể yêu cầu xét nghiệm trong lab_tests.

Bảng 2.5. Mô tả bảng doctor

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã bác sĩ
2	user_id	bigint	Không	FK	Liên kết user
3	department_id	bigint	Không	FK	Khoa công tác
4	specialization	varchar	Có		Chuyên môn
5	license_number	varchar	Có		Mã chứng chỉ
6	license_image	varchar	Có		Ảnh chứng chỉ
7	avatar	varchar	Có		Ảnh đại diện
8	description	text	Có		Mô tả bác sĩ
9	birth_date	date	Có		Ngày sinh
10	created_at	timestamp	Có		Ngày tạo
11	updated_at	timestamp	Có		Ngày cập nhật

Bảng departments.

Bảng departments lưu trữ danh sách các khoa của bệnh viện. Bảng này là gốc của nhiều bảng khác như doctors, services, lab_tests và test_types, đảm bảo phân loại các thành phần theo khoa chuyên môn.

Bảng 2.6. Mô tả bảng departments

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã khoa
2	name	varchar	Không		Tên khoa
3	description	text	Có		Mô tả khoa
4	created_at	timestamp	Có		Ngày tạo
5	updated_at	timestamp	Có		Ngày cập nhật

Bảng services.

Bảng services lưu thông tin các dịch vụ khám chữa bệnh và liên kết với departments. Mỗi dịch vụ có thể được bệnh nhân chọn khi đặt lịch qua bảng appointments.

Bảng 2.7. Mô tả bảng services

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint UNSIGNED	Không	PK	Mã dịch vụ
2	name	varchar(255)	Không		Tên dịch vụ
3	description	text	Có		Mô tả
4	image	varchar(255)	Có		Ảnh
5	price	decimal(10,2)	Không		Giá
6	department_id	bigint UNSIGNED	Không	FK	Khoa
7	created_at	timestamp	Có		Ngày tạo
8	updated_at	timestamp	Có		Ngày cập nhật

Bảng service_symptoms.

Bảng service_symptoms mô tả các triệu chứng gợi ý cho từng dịch vụ, liên kết với services qua service_id. Đây là bảng hỗ trợ gợi ý dịch vụ phù hợp theo triệu chứng bệnh nhân.

Bảng 2.8. Mô tả bảng service_symptoms

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint UNSIGNED	Không	PK	Mã triệu chứng

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
2	service_id	bigint UNSIGNED	Không	FK	Dịch vụ
3	symptom_name	varchar(255)	Không		Triệu chứng
4	created_at	timestamp	Có		Ngày tạo
5	updated_at	timestamp	Có		Ngày cập nhật

Bảng appointments.

Bảng appointments quản lý toàn bộ lịch hẹn giữa bác sĩ và bệnh nhân, liên kết với patients, doctors và services. Mỗi lịch hẹn có thể dẫn tới một hồ sơ khám trong bảng medical_records.

Bảng 2.9. Mô tả bảng appointments

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã lịch hẹn
2	patient_id	bigint	Không	FK	Bệnh nhân
3	doctor_id	bigint	Không	FK	Bác sĩ khám
4	service_id	bigint	Không	FK	Dịch vụ khám
5	appointment_date	datetime	Không		Ngày giờ khám
6	status	enum	Có		Trạng thái lịch
7	payment_status	enum	Không		Trạng thái thanh toán
8	medical_examination	enum	Có		Ca khám
9	note	text	Có		Ghi chú
10	created_at	timestamp	Có		Ngày tạo
11	updated_at	timestamp	Có		Ngày cập nhật

Bảng diseases.

Bảng diseases lưu danh sách bệnh và các mô tả liên quan, được liên kết với departments để xác định bệnh thuộc khoa nào.

Bảng 2.10. Mô tả bảng diseases

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint UNSIGNED	Không	PK	Mã bệnh
2	name	varchar(255)	Không		Tên bệnh
3	department_id	bigint UNSIGNED	Không	FK	Thuộc khoa
4	description	text	Có		Mô tả
5	image	varchar(255)	Có		Ảnh
6	created_at	timestamp	Có		Ngày tạo
7	updated_at	timestamp	Có		Ngày cập nhật

Bảng disease_symptoms.

Bảng disease_symptoms mô tả triệu chứng của từng bệnh và liên kết với diseases. Bảng này hỗ trợ hệ thống phân tích triệu chứng theo từng loại bệnh.

Bảng 2.11. Mô tả bảng diseases_symptoms

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint UNSIGNED	Không	PK	Mã triệu chứng bệnh
2	disease_id	bigint UNSIGNED	Không	FK	Bệnh
3	symptom_name	varchar(255)	Không		Triệu chứng
4	created_at	timestamp	Có		Ngày tạo
5	updated_at	timestamp	Có		Ngày cập nhật

Bảng medical_records.

Bảng medical_records lưu hồ sơ bệnh án chi tiết, bao gồm mô tả, chẩn đoán, kết luận bác sĩ và đơn thuốc. Bảng này liên kết với appointments và patients, và có thể chứa nhiều xét nghiệm trong lab_tests.

Bảng 2.12. Mô tả bảng medical_records

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã hồ sơ khám
2	appointment_id	bigint	Không	FK	Lịch hẹn

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
3	patient_id	bigint	Không	FK	Bệnh nhân
4	description	text	Có		Mô tả khám
5	diagnosis	text	Có		Chuẩn đoán
6	doctor_conclusion	text	Có		Kết luận bác sĩ
7	prescription	longtext	Có		Đơn thuốc
8	image	varchar	Có		Ảnh
9	images	longtext	Có		Nhiều ảnh
10	created_at	timestamp	Có		Ngày tạo
11	updated_at	timestamp	Có		Ngày cập nhật

Bảng lab_tests.

Bảng lab_tests quản lý xét nghiệm, liên kết với medical_records, departments, test_types, requested_by và uploaded_by. điều này giúp tracking đầy đủ quá trình tạo và xử lý xét nghiệm.

Bảng 2.13. Mô tả bảng lab_tests

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã xét nghiệm
2	medical_record_id	bigint	Không	FK	Hồ sơ khám
3	department_id	bigint	Không	FK	Khoa thực hiện
4	test_type_id	bigint	Có	FK	Loại xét nghiệm
5	test_name	varchar	Không		Tên xét nghiệm
6	image	varchar	Có		Ảnh chính
7	images	longtext	Có		Ảnh chi tiết
8	note	text	Có		Ghi chú
9	requested_by	bigint	Không	FK	Người yêu cầu
10	uploaded_by	bigint	Có	FK	Người tải kết quả
11	status	enum	Không		Trạng thái
12	created_at	timestamp	Có		Ngày tạo
13	updated_at	timestamp	Có		Ngày cập nhật

Bảng messages.

Bảng messages quản lý tin nhắn giữa các tài khoản, liên kết với users qua sender_id và receiver_id, tạo hệ thống trao đổi thông tin tiện lợi.

Bảng 2.14. Mô tả bảng messages

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã tin nhắn
2	sender_id	bigint	Không	FK	Người gửi
3	receiver_id	bigint	Không	FK	Người nhận
4	message	text	Không		Nội dung
5	is_read	tinyint	Không		Đã xem chưa
6	created_at	timestamp	Có		Ngày tạo
7	updated_at	timestamp	Có		Ngày cập nhật

Bảng test_types.

Bảng test_types lưu loại xét nghiệm và liên kết với departments và users. Các loại xét nghiệm này được sử dụng trong lab_tests.

Bảng 2.15. Mô tả bảng test_types

STT	Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
1	id	bigint	Không	PK	Mã loại xét nghiệm
2	name	varchar	Không		Tên xét nghiệm
3	description	text	Có		Mô tả
4	department_id	bigint	Có	FK	Khoa liên quan
5	created_by	bigint	Có	FK	Người tạo
6	created_at	timestamp	Có		Ngày tạo
7	updated_at	timestamp	Có		Ngày cập nhật

2.3 Kết luận chương 2

Chương 2 đã trình bày chi tiết quá trình phân tích yêu cầu, mô hình hóa nghiệp vụ và thiết kế kiến trúc hệ thống thông qua các sơ đồ Use Case, tuần tự, hoạt động và mô hình cơ sở dữ liệu.

Các kết quả phân tích này là cơ sở quan trọng để đảm bảo hệ thống được xây dựng đúng nhu cầu, đúng chức năng và có khả năng mở rộng. Ở chương 3, hệ thống sẽ được hiện thực hóa dựa trên các mô hình và thiết kế đã được đề xuất.

Chương 3. CÀI ĐẶT VÀ THỰC NGHIỆM

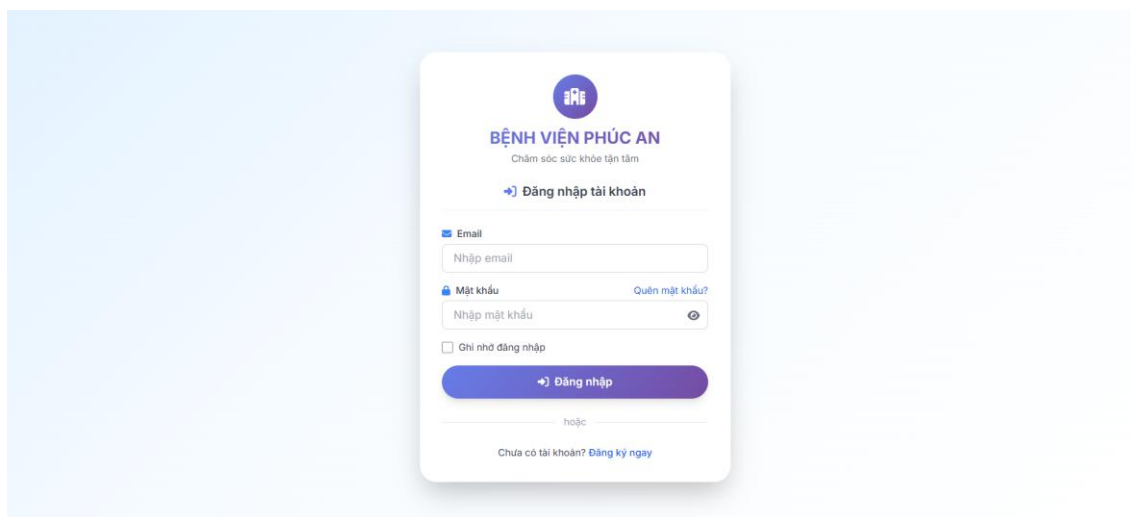
Hệ thống Quản lý Bệnh viện được phát triển trên nền tảng Laravel với cơ sở dữ liệu MySQL, đảm bảo tính bảo mật và hiệu suất cao. Quá trình cài đặt được thực hiện đơn giản thông qua việc sao chép mã nguồn từ kho lưu trữ, cài đặt các gói phụ thuộc bằng Composer và NPM, sau đó cấu hình cơ sở dữ liệu trong file .env. Hệ thống yêu cầu môi trường chạy XAMPP với PHP 8.0+ và MySQL 5.7+. Sau khi cài đặt, dữ liệu mẫu được tạo tự động thông qua hệ thống seeding của Laravel, bao gồm các tài khoản người dùng mẫu cho các vai trò quản trị viên, bác sĩ và bệnh nhân. Quá trình thực nghiệm đã được tiến hành trên nhiều module chính như quản lý người dùng, đặt lịch hẹn, quản lý bệnh án và xét nghiệm. Các chức năng được kiểm thử kỹ lưỡng thông qua kiểm thử đơn vị và kiểm thử tích hợp, đảm bảo hệ thống hoạt động ổn định, giao diện thân thiện và đáp ứng đầy đủ các yêu cầu nghiệp vụ của bệnh viện.

3.1. Giao diện chức năng

Giao diện chức năng tổng quan của hệ thống được thiết kế hiện đại, thân thiện và dễ sử dụng, đáp ứng nhu cầu của cả ba đối tượng người dùng chính: bệnh nhân, bác sĩ và admin. Giao diện chính được tổ chức khoa học với thanh điều hướng rõ ràng, hiển thị đầy đủ các chức năng theo từng vai trò. Hệ thống sử dụng bảng điều khiển (dashboard) trực quan với các biểu đồ thống kê, giúp người dùng nắm bắt nhanh chóng thông tin quan trọng. Giao diện đặt lịch khám được tối ưu hóa với lịch trực quan, cho phép bệnh nhân dễ dàng chọn khung giờ phù hợp. Hệ thống còn tích hợp tính năng thông báo thời gian thực, giúp người dùng luôn cập nhật được tình trạng lịch hẹn.

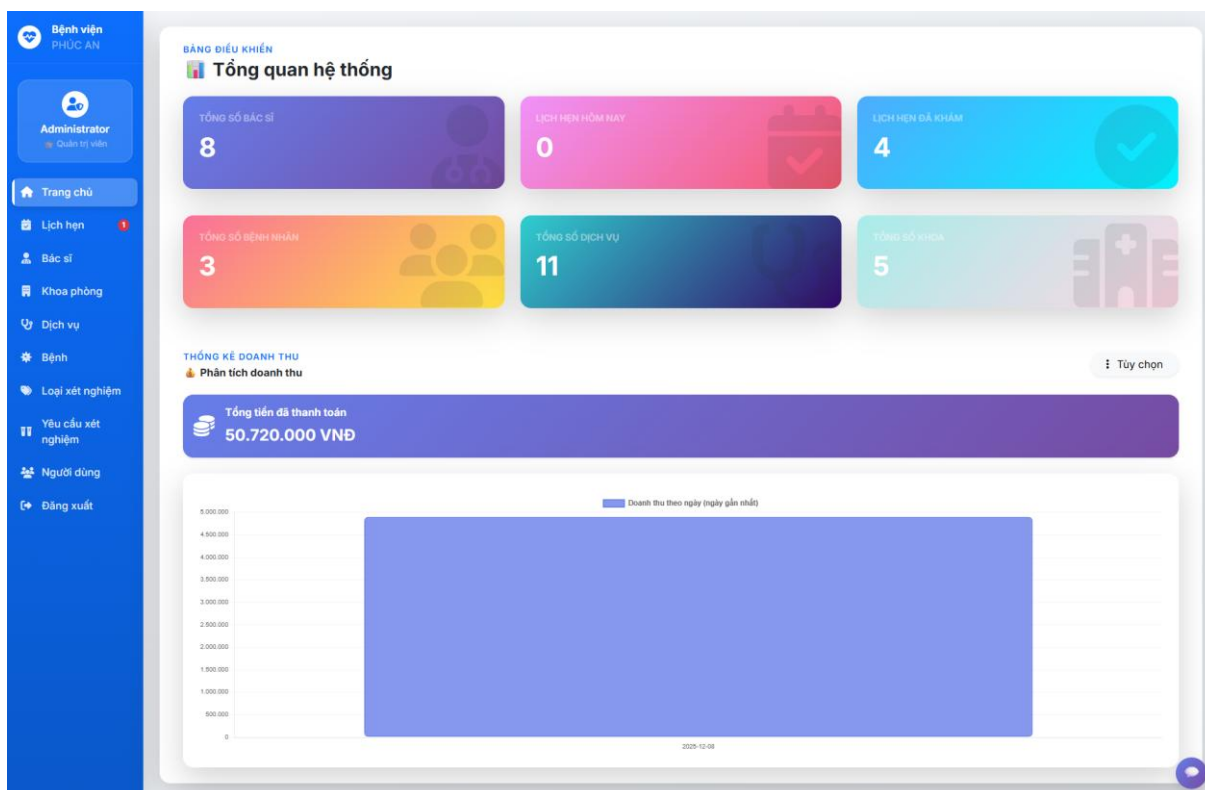
3.1.1. Giao diện chức năng Admin

Giao diện màn hình đăng nhập dành cho quản trị viên gồm các trường nhập email và mật khẩu, kèm nút đăng nhập hệ thống. Thiết kế đơn giản, rõ ràng, tập trung vào chức năng xác thực tài khoản, giúp đảm bảo chỉ những người dùng có quyền quản trị mới có thể truy cập khu vực quản lý.



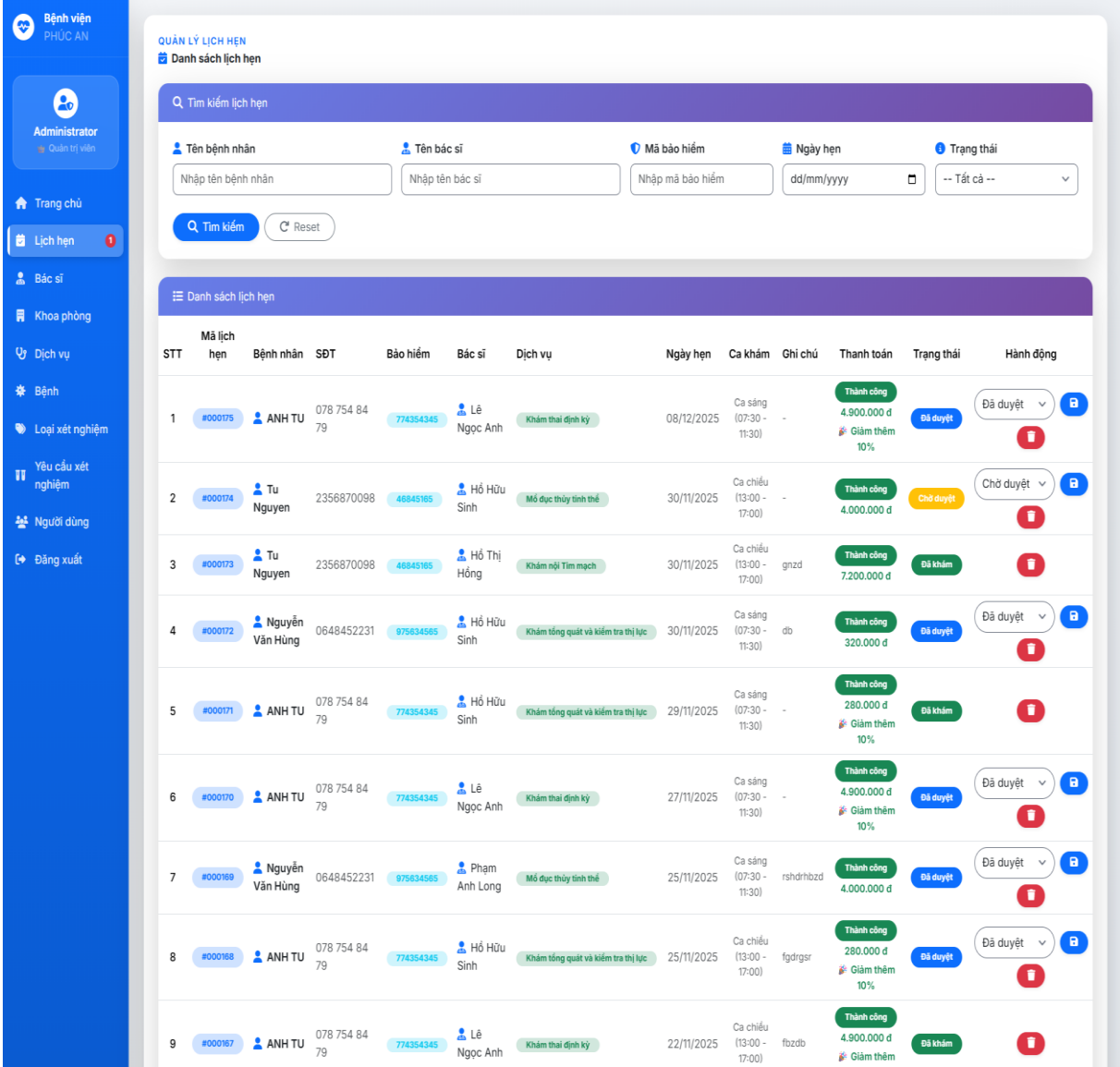
Hình 3.1. Giao diện đăng nhập admin

Giao diện bảng điều khiển (dashboard) dành cho admin sau khi đăng nhập thành công. Màn hình cung cấp các thống kê tổng quan như tổng số bác sĩ, số bệnh nhân, số lịch hẹn, số dịch vụ, số khoa và các chỉ số liên quan đến lịch hẹn trong ngày. Bên cạnh đó, dashboard còn hiển thị biểu đồ doanh thu theo ngày, tháng, năm, giúp quản trị viên nắm bắt nhanh tình hình hoạt động của hệ thống.



Hình 3.2. Giao diện dashboard admin

Giao diện quản lý danh sách lịch hẹn từ phía admin. Dữ liệu được hiển thị dưới dạng bảng, bao gồm các thông tin như bệnh nhân, bác sĩ, dịch vụ, ngày hẹn, ca khám, trạng thái và các thao tác như xem chi tiết, xác nhận, cập nhật trạng thái hoặc xóa lịch hẹn. Ngoài ra, giao diện có các ô tìm kiếm và bộ lọc theo tên bệnh nhân, bác sĩ, ngày hẹn, trạng thái... nhằm hỗ trợ admin tra cứu nhanh chóng.

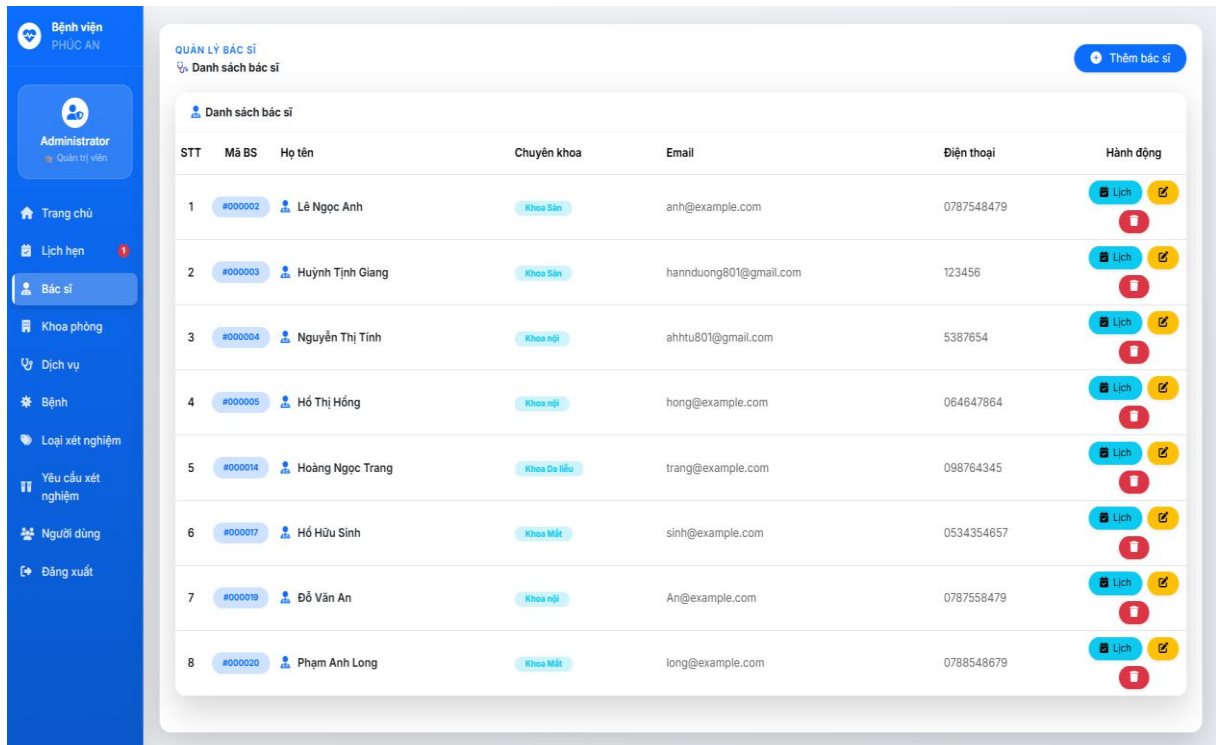


The screenshot displays a web application for managing appointments. On the left is a blue sidebar with navigation links: Trang chủ, Lịch hẹn (active), Bác sĩ, Khoa phòng, Dịch vụ, Bệnh, Loại xét nghiệm, Yêu cầu xét nghiệm, Người dùng, and Đăng xuất. The main content area is titled 'QUẢN LÝ LỊCH HẸN' and 'Danh sách lịch hẹn'. It features a search bar and filters for patient name, doctor name, insurance code, appointment date, and status. Below is a table with 9 appointment records.

STT	Mã lịch hẹn	Bệnh nhân	SĐT	Bảo hiểm	Bác sĩ	Dịch vụ	Ngày hẹn	Ca khám	Ghi chú	Thanh toán	Trạng thái	Hành động
1	#000175	ANH TU	078 754 84 79	774354345	Lê Ngọc Anh	Khám thai định kỳ	08/12/2025	Ca sáng (07:30 - 11:30)	-	Thành công 4.900.000 đ Giảm thêm 10%	Đã duyệt	Chi tiết Xóa
2	#000174	Tu Nguyen	2356870098	46845165	Hồ Hữu Sinh	Mô dục thủy tinh thể	30/11/2025	Ca chiều (13:00 - 17:00)	-	Thành công 4.000.000 đ	Chờ duyệt	Chi tiết Xóa
3	#000173	Tu Nguyen	2356870098	46845165	Hồ Thị Hồng	Khám nội Tim mạch	30/11/2025	Ca chiều (13:00 - 17:00)	gnzd	Thành công 7.200.000 đ	Đã khám	Chi tiết Xóa
4	#000172	Nguyễn Văn Hùng	0648452231	975634565	Hồ Hữu Sinh	Khám tổng quát và kiểm tra thị lực	30/11/2025	Ca sáng (07:30 - 11:30)	do	Thành công 320.000 đ	Đã duyệt	Chi tiết Xóa
5	#000171	ANH TU	078 754 84 79	774354345	Hồ Hữu Sinh	Khám tổng quát và kiểm tra thị lực	29/11/2025	Ca sáng (07:30 - 11:30)	-	Thành công 280.000 đ Giảm thêm 10%	Đã khám	Chi tiết Xóa
6	#000170	ANH TU	078 754 84 79	774354345	Lê Ngọc Anh	Khám thai định kỳ	27/11/2025	Ca sáng (07:30 - 11:30)	-	Thành công 4.900.000 đ Giảm thêm 10%	Đã duyệt	Chi tiết Xóa
7	#000169	Nguyễn Văn Hùng	0648452231	975634565	Phạm Anh Long	Mô dục thủy tinh thể	25/11/2025	Ca sáng (07:30 - 11:30)	rshdrhbdz	Thành công 4.000.000 đ	Đã duyệt	Chi tiết Xóa
8	#000168	ANH TU	078 754 84 79	774354345	Hồ Hữu Sinh	Khám tổng quát và kiểm tra thị lực	25/11/2025	Ca chiều (13:00 - 17:00)	fgdrgrs	Thành công 280.000 đ Giảm thêm 10%	Đã duyệt	Chi tiết Xóa
9	#000167	ANH TU	078 754 84 79	774354345	Lê Ngọc Anh	Khám thai định kỳ	22/11/2025	Ca chiều (13:00 - 17:00)	fbzdo	Thành công 4.900.000 đ Giảm thêm 10%	Đã khám	Chi tiết Xóa

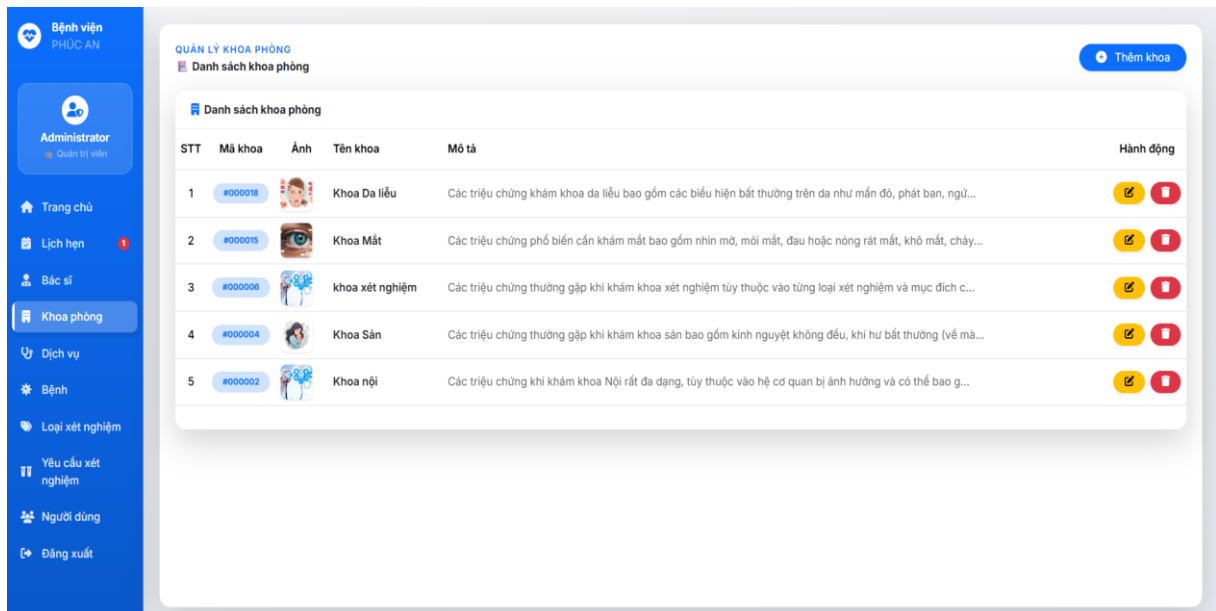
Hình 3.3. Giao diện quản lý danh sách lịch hẹn

Giao diện danh sách bác sĩ trong khu vực quản trị. Màn hình hiển thị các thông tin cơ bản về bác sĩ như họ tên, khoa làm việc, số điện thoại, email và các thao tác chỉnh sửa hoặc xóa. Admin có thể thêm mới bác sĩ, cập nhật thông tin và quản lý tài khoản liên kết với từng hồ sơ bác sĩ.



Hình 3.4. Giao diện quản lý danh sách Bác sĩ

Giao diện quản lý danh sách khoa trong bệnh viện. Danh sách thể hiện tên khoa, mô tả chức năng, hình ảnh minh họa (nếu có) và các nút thao tác thêm, sửa, xóa khoa. Giao diện này giúp admin tổ chức và cập nhật thông tin về các chuyên khoa một cách hệ thống.



Hình 3.5. Giao diện quản lý danh sách khoa

Giao diện danh sách các dịch vụ khám chữa bệnh mà hệ thống hỗ trợ. Bảng hiển thị tên dịch vụ, khoa phụ trách, mô tả, giá dịch vụ và các thao tác quản lý. Admin có thể thêm mới dịch vụ, chỉnh sửa thông tin hoặc ngừng cung cấp một dịch vụ khi cần thiết.

STT	Mã DV	Ảnh	Tên dịch vụ	Mô tả	Triệu chứng	Giá	Khoa	Hành động
1	#000043		Khám viêm phổi	Khám lâm sàng vùng phổi – ngực Nghe phổi bằng ống nghe để phát hiện ran ẩm/ran...	Ho - sốt - khó thở +1	500.000 đ	Khoa nội	
2	#000041		Khám phụ khoa	Các bước thực hiện: Hỏi bệnh sử: khi hư, chu kỳ kinh, đau bụng... Khám vùng...	Ngứa vùng kín - Chu kỳ không đều	600.000 đ	Khoa Sản	
3	#000040		Soi da (Dermatoscopy)	Các bước thực hiện: Làm sạch vùng da Dùng kính dermatoscope để phóng đại b...	Mụn nhiều - Tồn thương da kéo dài - Mảng sắc tố lạ	800.000 đ	Khoa Da liễu	
4	#000039		Xét nghiệm máu tổng quát	Các bước thực hiện: Lấy mẫu máu (1-3 ml) Phân tích các chỉ số: công thức m...	Mệt mỏi - Da xanh xao - Ngủ thiếu ngủ +2	500.000 đ	Khoa nội	
5	#000038		khám da liễu	Quy trình khám và chẩn đoán tại khoa Da liễu Khi đến khám, bác sĩ da liễu sẽ th...	Thay đổi màu sắc da - Ngứa da kéo dài - Nổi mụn, phát ban, mề đay +3	3.000.000 đ	Khoa Da liễu	
6	#000027		Khám da liễu đường	Các phương pháp chẩn đoán và kiểm tra khi đi khám Khi có các triệu chứng trên...	Ưng nhọt - Tiểu nhiều - Ăn nhiều +6	3.000.000 đ	Khoa nội	
7	#000036		Khám nội Tim mạch	Đánh giá triệu chứng: Khám các triệu chứng như đau ngực, khó thở, hồi hộp, chóng...	Đau ngực - Khó thở - Hồi hộp, đánh trống ngực +2	9.000.000 đ	Khoa nội	
8	#000035		Chăm sóc sau sinh	Theo dõi sức khỏe của mẹ và bé. Hướng dẫn chăm sóc trẻ sơ sinh. Tư vấn nuôi co...	Chăm sóc sau sinh - Chăm sóc rốn - Vệ sinh +8	1.000.000 đ	Khoa Sản	
9	#000034		Khám thai định kỳ	Theo dõi sự phát triển của thai nhi và sức khỏe của mẹ bầu Siêu âm Xét nghiệm...	Siêu âm thai	7.000.000 đ	Khoa Sản	
10	#000030		Mổ đục thủy tinh thể	Bước 1: Kiểm tra thị lực, làm xét nghiệm Bước 2: Tiến hành mổ đục thủy tinh thể...	Giảm thị lực rõ rệt - Nhìn mờ, lóa, chói - Khó khăn khi thực hiện sinh hoạt +1	5.000.000 đ	Khoa Mắt	

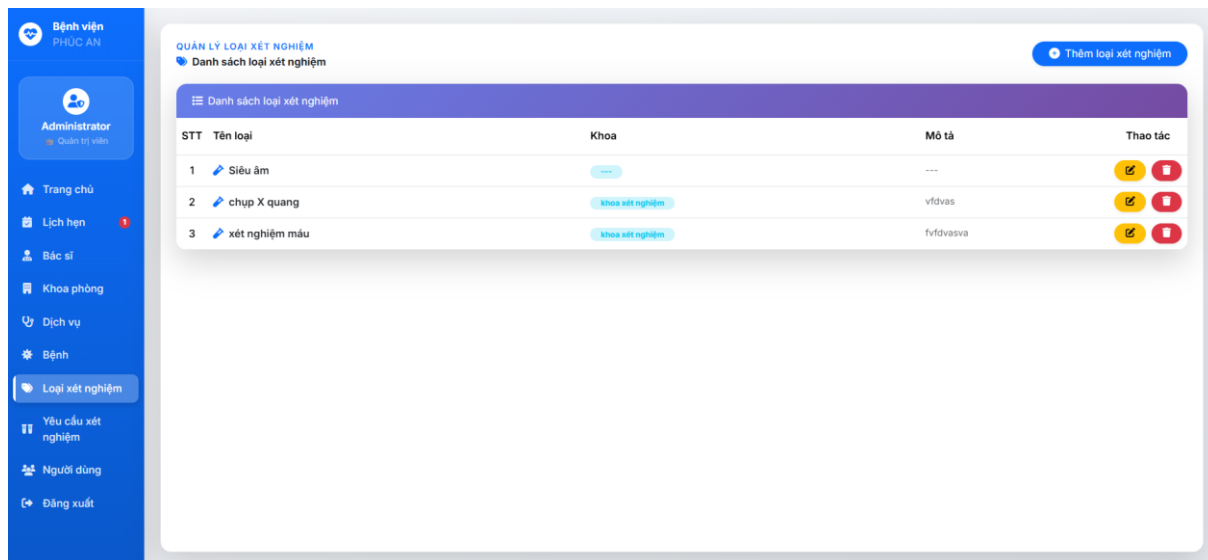
Hình 3.6. Giao diện quản lý danh sách dịch vụ

Giao diện quản lý danh mục các loại bệnh. Mỗi bản ghi bệnh bao gồm tên bệnh, khoa phụ trách, mô tả và các triệu chứng liên quan. Thông qua giao diện này, admin có thể cập nhật thông tin bệnh, hỗ trợ các chức năng tư vấn theo triệu chứng và gợi ý dịch vụ phù hợp.

STT	Tên bệnh	Khoa	Triệu chứng	Thao tác
1	Mụn trứng cá mức độ nhẹ - nặng	Khoa Da liễu	1 triệu chứng	
2	Viêm da tiếp xúc, eczema, zona	Khoa Da liễu	3 triệu chứng	
3	Viêm da dị ứng, mề đay, nám da	Khoa Da liễu	1 triệu chứng	
4	Viêm phụ khoa, u nang, rối loạn kinh nguyệt	Khoa Sản	1 triệu chứng	
5	Viêm âm đạo, nấm, viêm lộ tuyến	Khoa Sản	2 triệu chứng	
6	Mang thai, rối loạn nội tiết	Khoa Sản	2 triệu chứng	
7	Viêm kết mạc, dị ứng mắt, khô mắt	Khoa Mắt	2 triệu chứng	
8	Cận thị, loạn thị, đục thủy tinh thể, tăng nhãn áp	Khoa Mắt	2 triệu chứng	
9	Viêm phế quản, hen, viêm họng	Khoa nội	1 triệu chứng	
10	Viêm dạ dày, trào ngược, loét dạ dày - tá tràng	Khoa nội	1 triệu chứng	

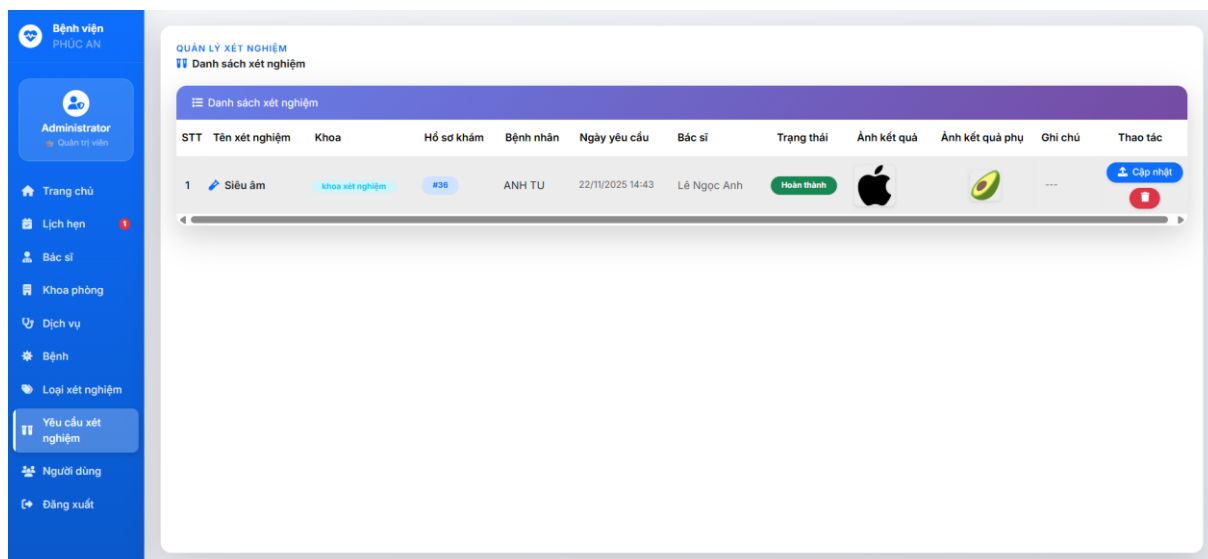
Hình 3.7. Giao diện quản lý danh sách các loại bệnh

Giao diện quản lý danh sách các loại xét nghiệm trong hệ thống. Danh sách thể hiện tên loại xét nghiệm, mô tả ngắn gọn và các thao tác thêm mới, chỉnh sửa, xóa. Giao diện này giúp chuẩn hóa các loại xét nghiệm được sử dụng khi bác sĩ yêu cầu thực hiện xét nghiệm cho bệnh nhân.



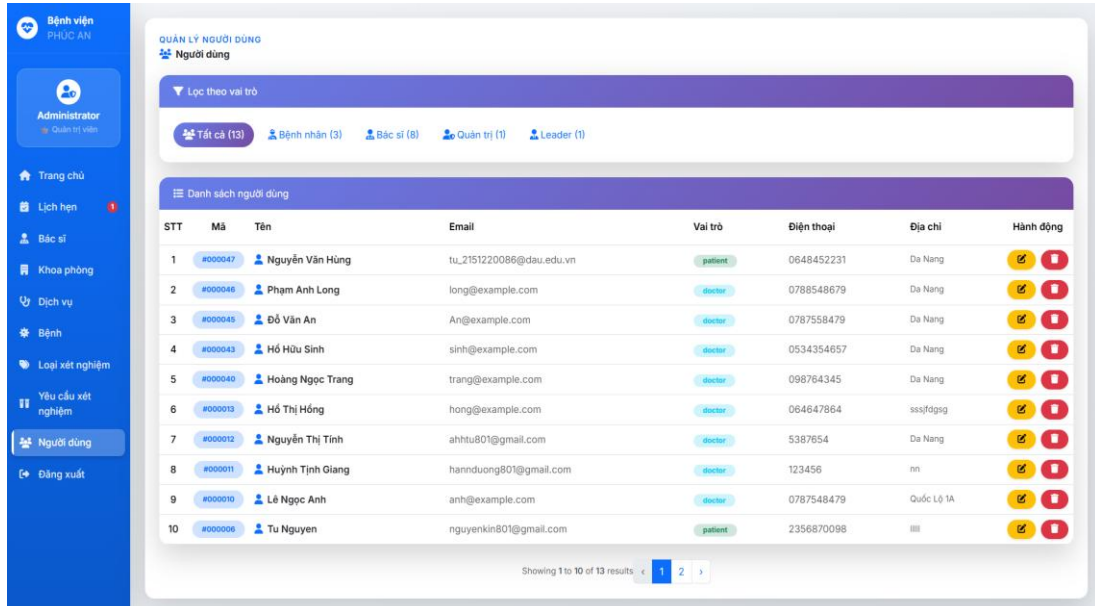
Hình 3.8. Giao diện quản lý danh sách loại xét nghiệm

Giao diện danh sách các yêu cầu xét nghiệm đã được tạo trong hệ thống. Bảng dữ liệu hiển thị thông tin bệnh nhân, bác sĩ chỉ định, loại xét nghiệm, ngày yêu cầu, trạng thái kết quả và các thao tác xử lý như xem chi tiết, tải kết quả hoặc xóa yêu cầu. Giao diện hỗ trợ bộ phận xét nghiệm theo dõi và cập nhật tiến độ thực hiện.



Hình 3.9. Giao diện danh sách được yêu cầu xét nghiệm

Giao diện quản lý tài khoản người dùng trong hệ thống, bao gồm tài khoản bệnh nhân, bác sĩ và admin. Danh sách thể hiện thông tin cơ bản như tên, email, vai trò, trạng thái tài khoản, đồng thời cung cấp các chức năng chỉnh sửa, phân quyền hoặc xóa tài khoản khi cần.



Hình 3.10. Giao diện quản lý danh sách tài khoản

3.1.2. Giao diện chức năng bệnh nhân

Gửi email thông báo lịch hẹn đã được xác nhận gửi đến bệnh nhân. Email trình bày rõ thời gian khám, bác sĩ phụ trách, dịch vụ, địa điểm và các lưu ý cần thiết trước khi đến khám. Tự động gửi mail xác nhận giúp tăng tính chuyên nghiệp và hạn chế nhầm lẫn về thông tin lịch hẹn.



Bệnh Viện Phúc An <kinnguyen801@gmail.com>
đến tôi ▾

01:5

Xin chào ANH TU

Lịch hẹn của bạn đã được **xác nhận**.

- **Dịch vụ:** Khám thai định kỳ
- **Bác sĩ:** Lê Ngọc Anh
- **Thời gian:** 08/12/2025
- **Ca khám:** Ca sáng (07:30 - 11:30)
- **Giá gốc:** 7.000.000 đ
- **Giảm giá:** 10% (700.000 đ)



Sinh nhật trong tháng! Bạn chỉ cần thanh toán 70% giá gốc (đã bao gồm ưu đãi sinh nhật).

- Thanh toán trước: **4.900.000 đ**
- Còn lại: 1.400.000 đ Thanh toán tại bệnh viện
- Ghi chú: Không có

Cảm ơn bạn đã đặt lịch tại hệ thống!

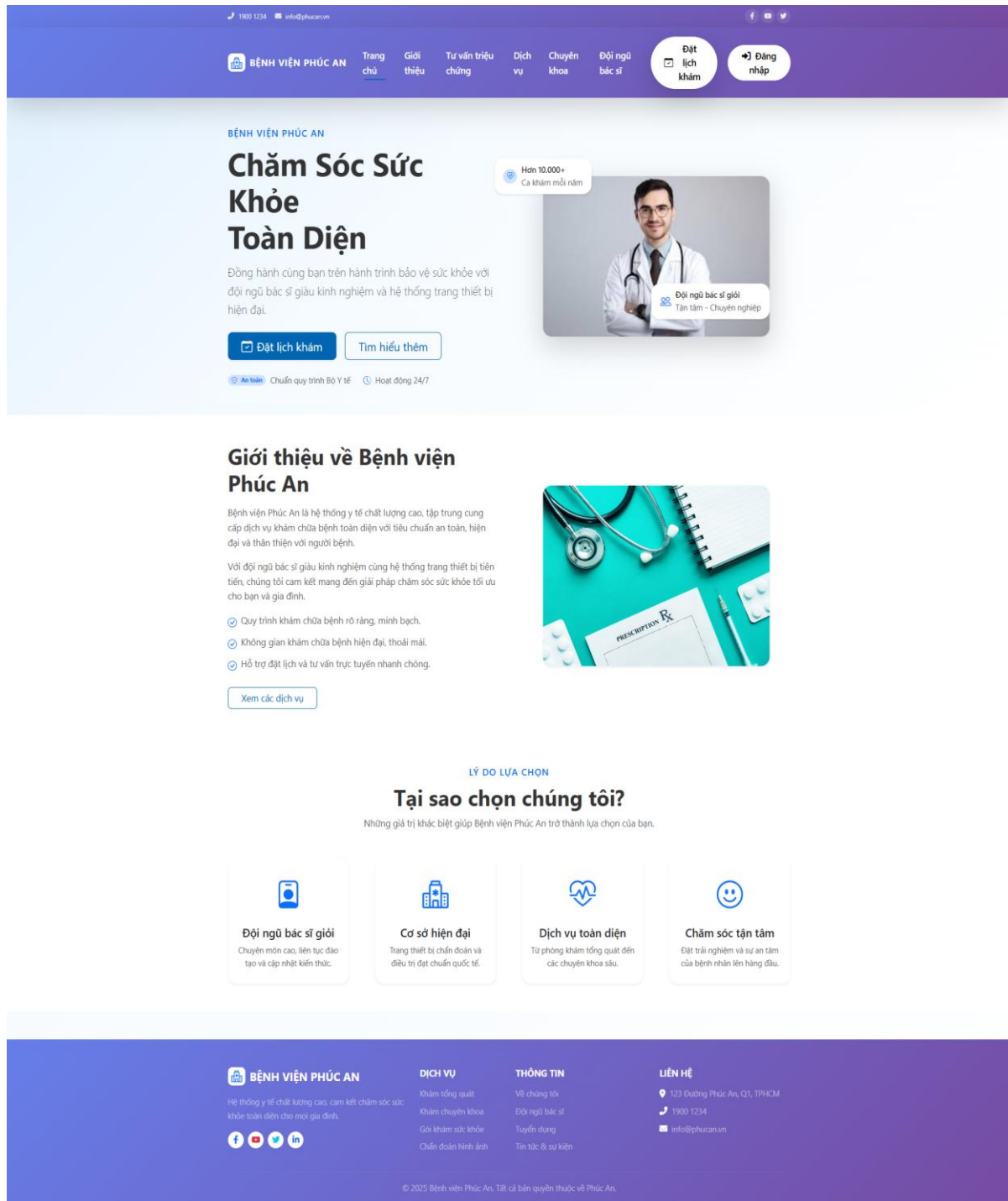
Hình 3.11. Mail lịch hẹn được xác nhận

Giao diện form đặt lịch khám bệnh cho bệnh nhân. Tại đây, người dùng có thể lựa chọn bác sĩ, dịch vụ, khoa, ngày khám, ca khám (sáng/chiều) và nhập ghi chú triệu chứng. Các trường bắt buộc được kiểm tra hợp lệ trước khi gửi, giúp quá trình đặt lịch diễn ra chính xác, hạn chế trùng lặp hoặc sai sót.

The screenshot displays a web application interface for booking medical appointments. The header is purple with white text and icons, including the hospital name 'BỆNH VIỆN PHÚC AN', contact information '1900 1234' and 'info@phucan.vn', and a navigation menu with links like 'Trang chủ', 'Giới thiệu', 'Tư vấn triệu chứng', 'Dịch vụ', 'Chuyên khoa', 'Đội ngũ bác sĩ', and a 'Đặt lịch khám' button. The main content area is a white card with a purple header 'ĐẶT LỊCH KHÁM' and a sub-header 'Đặt lịch khám bệnh'. Below this, a message says 'Vui lòng điền đầy đủ thông tin để đặt lịch khám'. The form is divided into two columns. The left column, titled 'Thông tin dịch vụ', contains a large purple box with a white icon and text: 'Chưa chọn dịch vụ. Vui lòng chọn dịch vụ ở form bên cạnh'. The right column, titled 'Thông tin đặt lịch', contains several input fields: a green message box 'Xin chào, ANH TU! Bạn đã đăng nhập và sẵn sàng đặt lịch.', a 'Khoa' dropdown menu, a 'Dịch vụ trong khoa' dropdown menu, a 'Bác sĩ' dropdown menu, a 'Ngày khám' date picker set to 'dd/mm/2025', a 'Ca khám' dropdown menu, and a 'Ghi chú' text area with the placeholder 'Nhập ghi chú (nếu có)...'. At the bottom of the right column is a purple button labeled 'Thanh toán VNPAY'. The footer is purple with white text, containing the hospital name, a brief description of the medical system, social media icons, a list of services (Khám tổng quát, Khám chuyên khoa, Gói khám sức khỏe, Chẩn đoán hình ảnh), a list of team members (Vũ Chung Tới, Đội ngũ bác sĩ, Tuyển dụng, Tin tức & sự kiện), contact information, and a copyright notice '© 2025 Bệnh viện Phúc An. Tất cả bản quyền thuộc về Phúc An.'

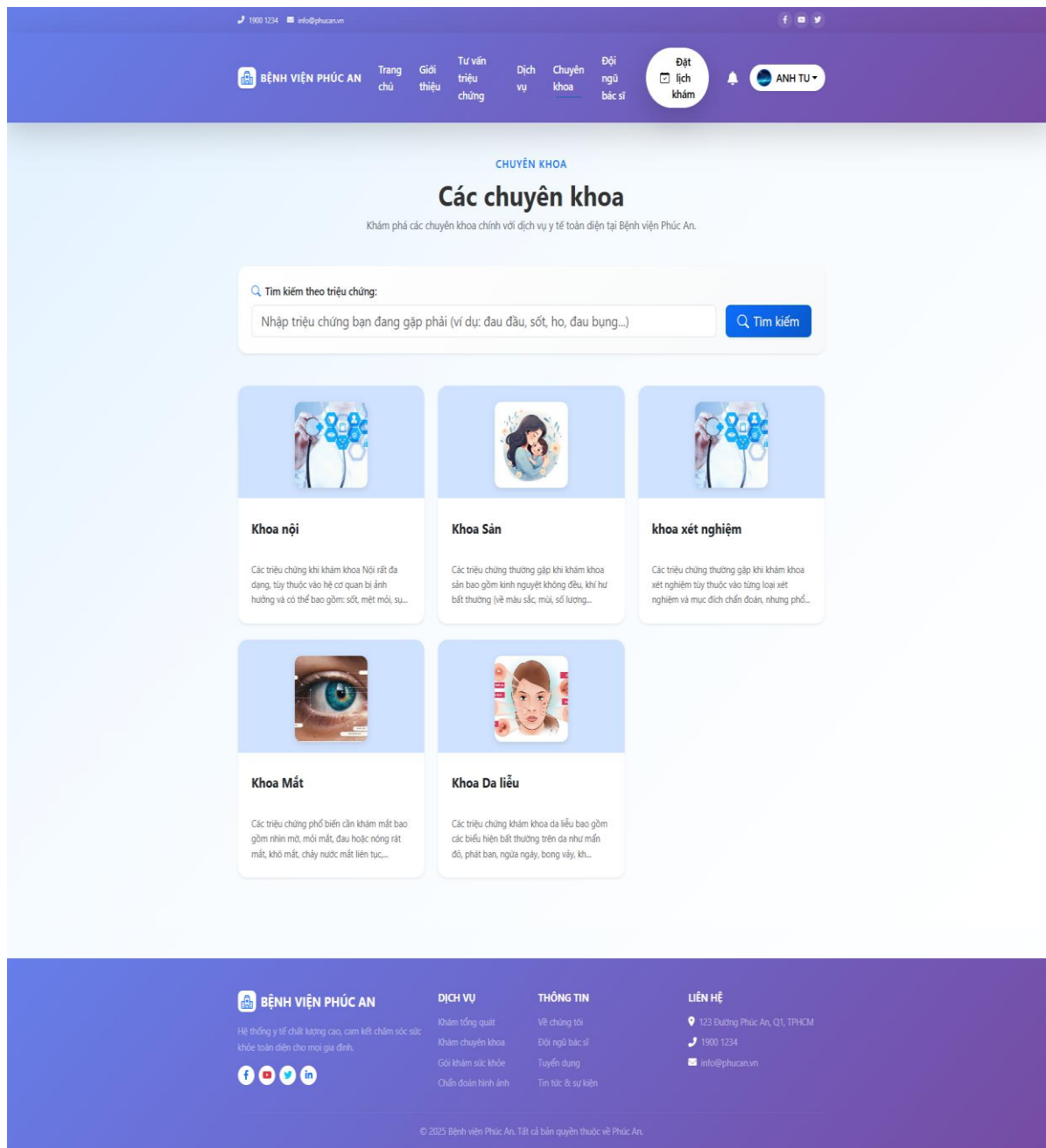
Hình 3.12. Giao diện form đặt lịch khám

Giao diện trang chủ của hệ thống. Trang chủ hiển thị tổng quan về bệnh viện, các chuyên khoa nổi bật, đội ngũ bác sĩ, dịch vụ phổ biến và các nút truy cập nhanh như đặt lịch khám, tư vấn theo triệu chứng, xem danh sách khoa hoặc bác sĩ. Thiết kế hướng đến sự trực quan, thân thiện với người dùng.



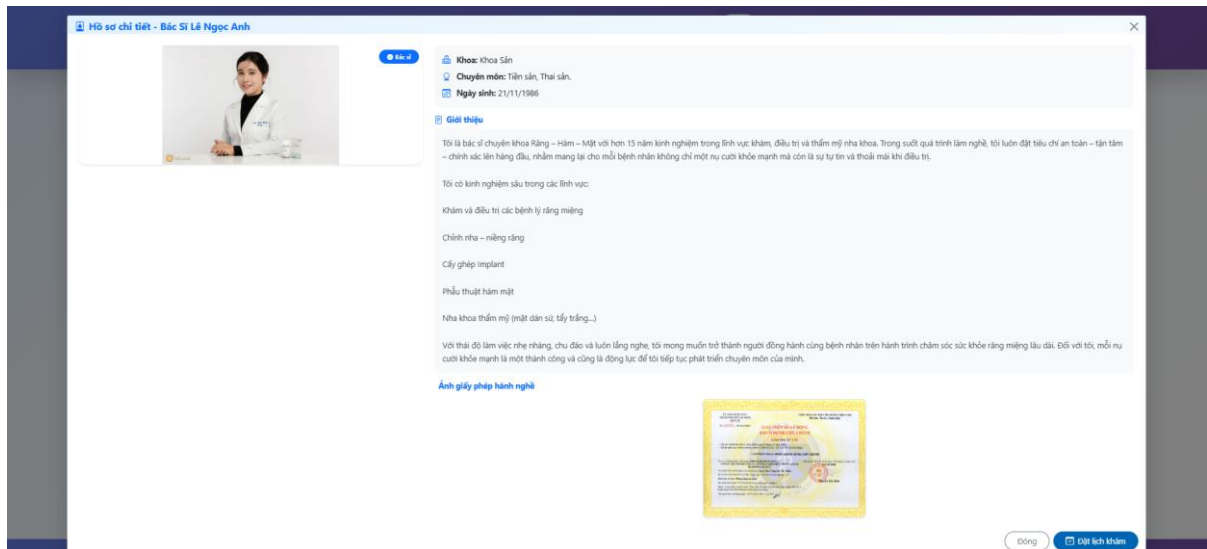
Hình 3.13. Giao diện trang chủ

Giao diện danh sách và mô tả chi tiết các khoa cho bệnh nhân. Mỗi khoa được hiển thị với tên, mô tả chức năng và có thể kèm hình ảnh minh họa. Bệnh nhân có thể xem thông tin, tìm hiểu chuyên môn của từng khoa, từ đó lựa chọn khoa phù hợp với triệu chứng của mình.



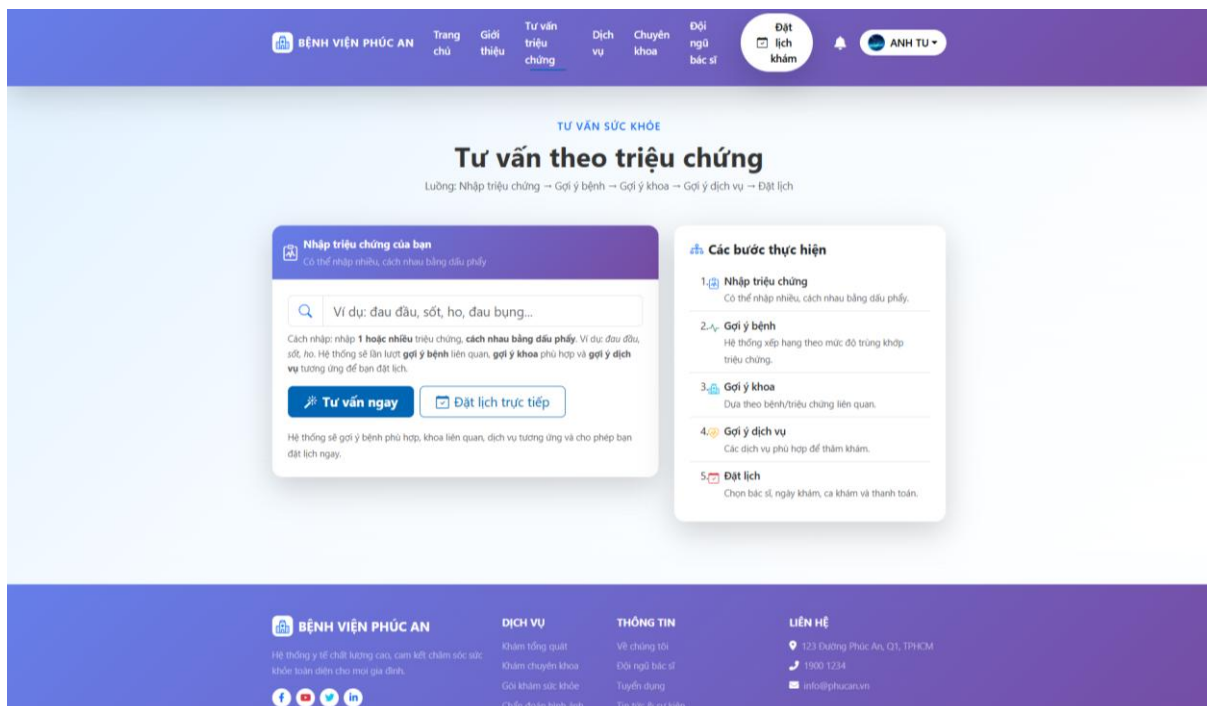
Hình 3.14. Giao diện mô tả các khoa cho bệnh nhân

Giao diện danh sách bác sĩ dành cho bệnh nhân. Giao diện hiển thị ảnh đại diện, họ tên, khoa làm việc, chuyên môn và thông tin liên hệ cơ bản. Bệnh nhân có thể tìm kiếm theo tên hoặc bộ lọc, giúp chọn được bác sĩ phù hợp trước khi tiến hành đặt lịch.



Hình 3.15. Giao diện thông tin bác sĩ cho bệnh nhân

Giao diện chức năng tư vấn theo triệu chứng. Bệnh nhân nhập các triệu chứng đang gặp phải, hệ thống sẽ phân tích và gợi ý các dịch vụ, bệnh lý và khoa phù hợp, đồng thời đề xuất một số bác sĩ liên quan. Chức năng này hỗ trợ bệnh nhân định hướng khám bệnh ban đầu một cách thuận tiện.



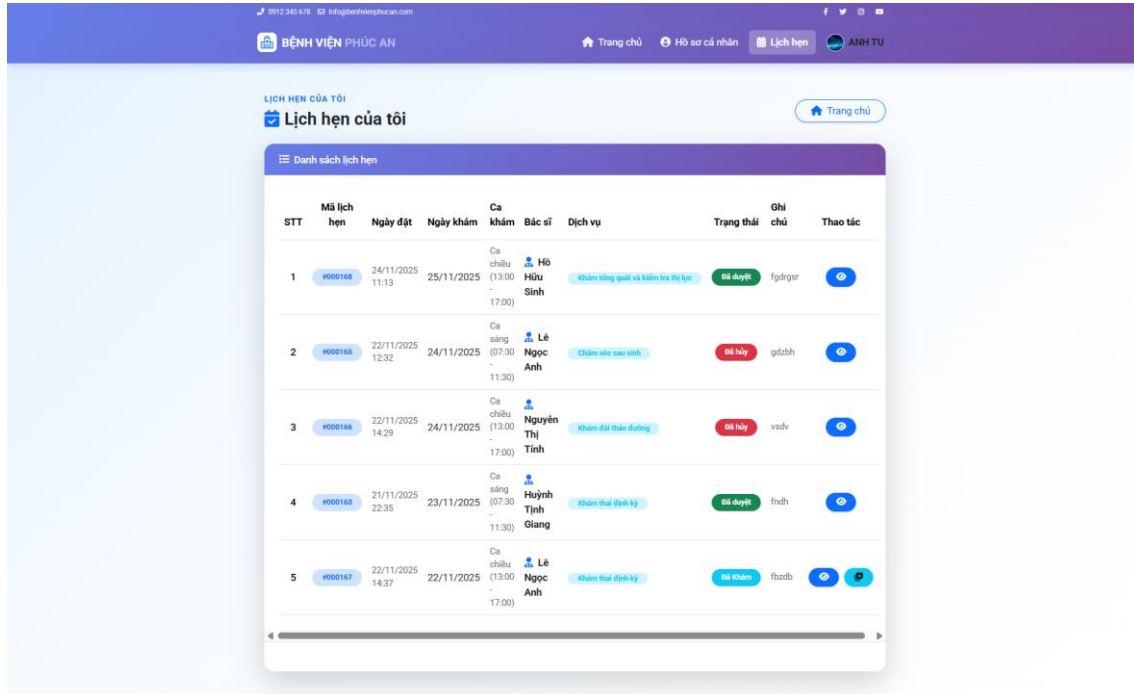
Hình 3.16. Giao diện tư vấn theo triệu chứng cho bệnh nhân

Giao diện cho phép bệnh nhân xem và chỉnh sửa thông tin cá nhân của mình như họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, thông tin bảo hiểm... Giao diện giúp bệnh nhân chủ động cập nhật dữ liệu, đảm bảo hồ sơ luôn chính xác phục vụ quá trình khám chữa bệnh.

The screenshot displays the 'Cập nhật thông tin cá nhân' (Update personal information) page of the Phúc An Hospital website. The page features a purple header with the hospital's name and navigation links. The main content area is divided into three sections: 'Thông tin cá nhân' (Personal information), 'Thay đổi mật khẩu' (Change password), and 'Xóa tài khoản' (Delete account). The 'Thông tin cá nhân' section includes a profile picture upload area, a 'Chọn tệp' (Choose file) button, and a 'Không có tệp nào được chọn' (No files selected) message. Below this, there are input fields for 'Họ và tên' (Last name and first name) with the value 'ANH TU', 'Email' with 'kinnguyen801@gmail.com', 'Số điện thoại' (Phone number) with '078 754 84 79', 'Ngày sinh' (Date of birth) with '6/12/2000', 'Giới tính' (Gender) with 'Nam' (Male), 'Số thẻ BHYT' (Health Insurance Card Number) with '774354345', and 'Địa chỉ' (Address) with 'Da Nang'. The 'Thay đổi mật khẩu' section includes fields for 'Mật khẩu hiện tại' (Current password), 'Mật khẩu mới' (New password), and 'Xác nhận mật khẩu' (Confirm password), along with a 'Nhập lại mật khẩu mới' (Re-enter new password) field. The 'Xóa tài khoản' section has a 'Xóa tài khoản' button. The page also includes a 'Đặt lại' (Reset) button and a 'Lưu thay đổi' (Save changes) button.

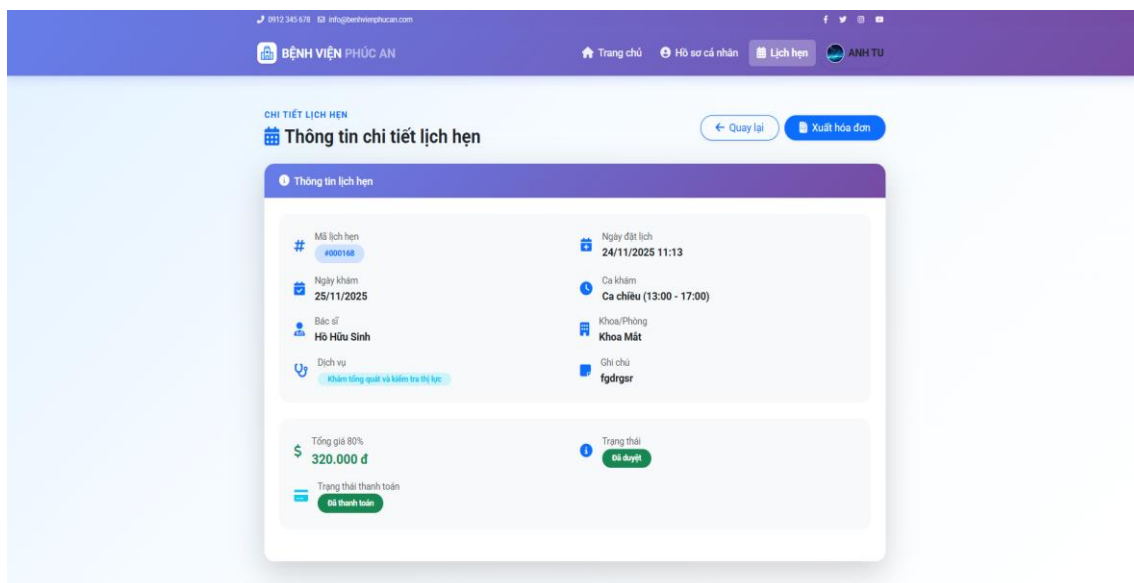
Hình 3.17. Giao diện quản lý thông tin cá nhân của bệnh nhân

Giao diện danh sách lịch hẹn ở phía bệnh nhân. Màn hình liệt kê các lịch hẹn mà bệnh nhân đã đặt, bao gồm thông tin bác sĩ, dịch vụ, ngày và ca khám, trạng thái lịch hẹn. Bệnh nhân có thể theo dõi, hủy hoặc xem chi tiết từng lịch hẹn ngay trên giao diện này.



Hình 3.18. Giao diện quản lý danh sách lịch hẹn của bệnh nhân

Giao diện hiển thị chi tiết một lịch hẹn cụ thể. Các thông tin như thời gian, địa điểm, bác sĩ phụ trách, dịch vụ, ghi chú triệu chứng, trạng thái lịch hẹn được trình bày đầy đủ. Giao diện giúp bệnh nhân nắm rõ nội dung cuộc hẹn trước khi đến khám.



Hình 3.19. Giao diện chi tiết lịch hẹn của bệnh nhân

Giao diện hồ sơ khám bệnh của bệnh nhân trên hệ thống. Hồ sơ bao gồm thông tin các lần khám, chẩn đoán, kết quả xét nghiệm, phương án điều trị và lịch sử đơn thuốc (nếu có). Việc số hóa hồ sơ giúp bệnh nhân và bác sĩ dễ dàng tra cứu lại thông tin các lần khám trước.

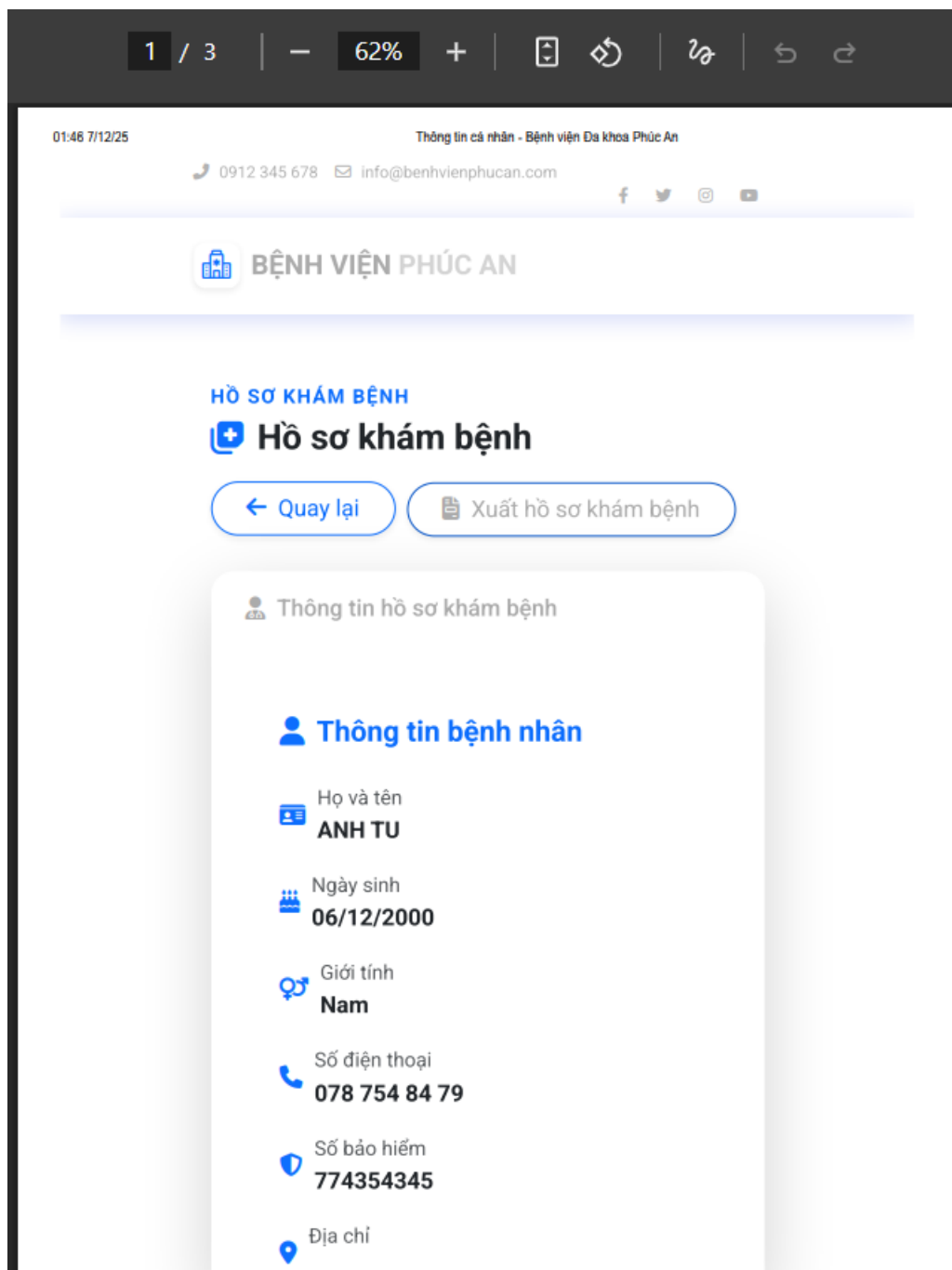
The screenshot displays a web application for managing patient medical records. The interface is titled "HỒ SƠ KHÁM BỆNH" (Medical Record) and includes a navigation bar with links to "Trang chủ" (Home), "Hồ sơ cá nhân" (Personal Record), "Lịch hẹn" (Appointment), and a user profile for "ANH TU".

The main content area is divided into several sections:

- Thông tin hồ sơ khám bệnh** (Medical Record Information): This section contains a sub-section "Thông tin bệnh nhân" (Patient Information) with the following details:
 - Họ và tên: ANH TU
 - Ngày sinh: 06/12/2000
 - Giới tính: Nam
 - Số điện thoại: 078 754 84 79
 - Số bảo hiểm: 774354345
 - Địa chỉ: Da Nang Da Nang
- Thông tin khám bệnh** (Medical Examination Information): This section contains a sub-section "Thông tin khám bệnh" (Medical Examination Information) with the following details:
 - Bác sĩ phụ trách: Lê Ngọc Anh
 - Dịch vụ khám: Khám thai định kỳ
 - Ngày khám: 22/11/2025
 - Cà khám: Cà chiều (13:00 - 17:00)
 - Chỉ chủ: fbzdb
- Gói dịch vụ bệnh nhân đã chọn** (Selected Patient Service Package): This section contains a sub-section "Khám thai định kỳ" (Regular Prenatal Examination) with the following details:
 - Theo dõi sự phát triển của thai nhi và sức khỏe của mẹ bầu
 - Siêu âm
 - Xét nghiệm máu và nước tiểu
 - Tư vấn dinh dưỡng
- Ảnh xét nghiệm nếu có** (If any test results): This section contains a sub-section "Siêu âm - Ảnh chính:" (Ultrasound - Main Image) with a placeholder image of an Apple logo.
- Chẩn đoán & Kết luận** (Diagnosis & Conclusion): This section contains a sub-section "Chẩn đoán ban đầu:" (Initial Diagnosis) with a text input field labeled "Nhập chẩn đoán...".
- Kết luận bác sĩ:** (Physician's Conclusion): This section contains a sub-section "Kết luận bác sĩ:" (Physician's Conclusion) with a text input field labeled "Nhập kết luận bác sĩ...".
- Toa thuốc:** (Prescription): This section contains a sub-section "Toa thuốc:" (Prescription) with a text input field labeled "Vi dụ: Paracetamol 500mg - Uống 2 lần/ngày Vitamin C 1000mg - Sáng 1 viên".

Hình 3.20. Giao diện hồ sơ khám bệnh của bệnh nhân

Giao diện chức năng xuất hồ sơ khám bệnh của bệnh nhân, thường dưới dạng file PDF hoặc tài liệu in. Giao diện cho phép lựa chọn bệnh nhân, khoảng thời gian hoặc lần khám cụ thể, sau đó hệ thống tổng hợp thông tin để xuất thành hồ sơ hoàn chỉnh, hỗ trợ công tác lưu trữ, chuyển tuyến hoặc tư vấn thêm.



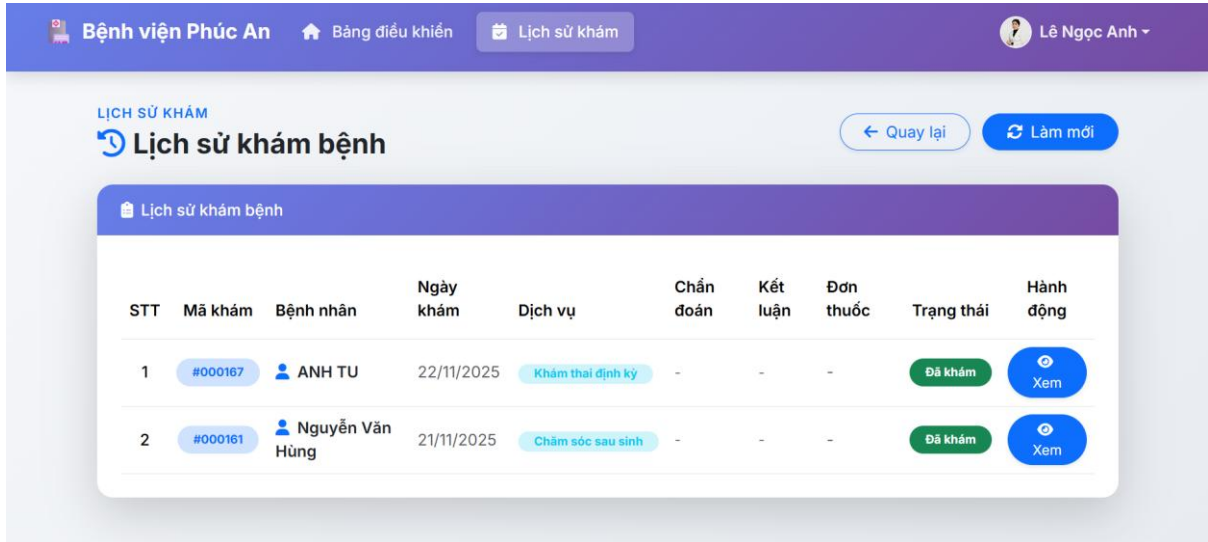
The screenshot displays a mobile application interface for exporting medical records. At the top, there's a header with the time '01:46 7/12/25' and the hospital name 'Thông tin cá nhân - Bệnh viện Đa khoa Phúc An'. Below this, contact information '0912 345 678' and 'info@benhvienphucan.com' is shown, along with social media icons. The main section is titled 'HỒ SƠ KHÁM BỆNH' (Medical Record) and features a large blue button with a plus icon and the text 'Hồ sơ khám bệnh'. Below this are two buttons: '← Quay lại' (Back) and 'Xuất hồ sơ khám bệnh' (Export medical record). A card titled 'Thông tin hồ sơ khám bệnh' (Medical record information) contains the following details:

- Thông tin bệnh nhân** (Patient information)
- Họ và tên (Name): **ANH TU**
- Ngày sinh (Date of birth): **06/12/2000**
- Giới tính (Gender): **Nam** (Male)
- Số điện thoại (Phone number): **078 754 84 79**
- Số bảo hiểm (Insurance number): **774354345**
- Địa chỉ (Address): (indicated by a location pin icon)

Hình 3.21. Xuất hồ sơ khám bệnh của bệnh nhân

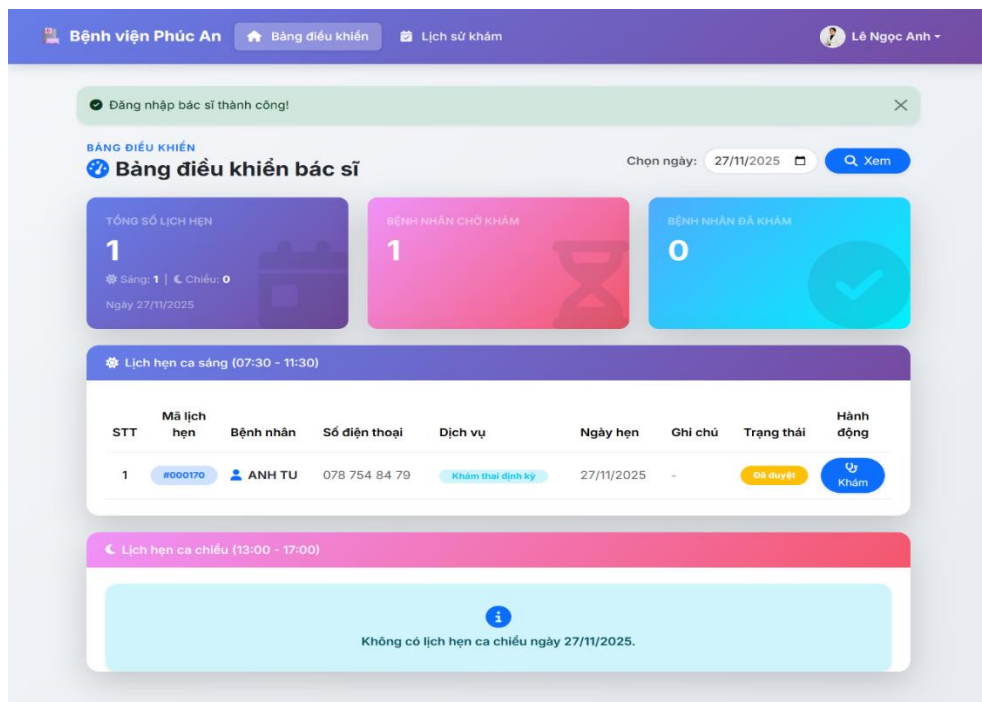
3.1.3. Giao diện chức năng Bác sĩ

Giao diện lịch sử khám bệnh từ góc nhìn của bác sĩ. Màn hình cho phép bác sĩ xem lại danh sách bệnh nhân đã khám, thông tin các buổi hẹn, chẩn đoán và điều trị trước đó. Điều này hỗ trợ bác sĩ có cái nhìn tổng quan, thuận tiện cho việc theo dõi và quản lý quá trình điều trị lâu dài.



Hình 3.22. Giao diện lịch sử khám của Bác sĩ

Giao diện chính (dashboard) dành cho bác sĩ sau khi đăng nhập. Màn hình cho bác sĩ cái nhìn tổng quan về lịch hẹn trong ngày, danh sách bệnh nhân sắp khám, các thông báo hoặc yêu cầu xét nghiệm mới. Giao diện được tối ưu để bác sĩ dễ dàng truy cập nhanh vào các chức năng cần thiết trong công việc hằng ngày.



Hình 3.23. Giao diện chính của Bác sĩ

Giao diện cho phép bác sĩ tạo và quản lý yêu cầu xét nghiệm cho bệnh nhân. Bác sĩ có thể chọn loại xét nghiệm, ghi chú lâm sàng, xem trạng thái xét nghiệm và kết quả trả về. Giao diện này kết nối chặt chẽ giữa bác sĩ lâm sàng và bộ phận xét nghiệm, giúp quy trình diễn ra đồng bộ và minh bạch.

Bệnh viện Phúc An | Bảng điều khiển | Lịch sử khám | Lê Ngọc Anh

HỒ SƠ KHÁM BỆNH | Quay lại danh sách

Thông tin hồ sơ khám bệnh

Thông tin bệnh nhân

Họ và tên: ANH TU | Ngày sinh: 06/12/2000 | Giới tính: Nam | Số điện thoại: 078 754 84 79

Số bảo hiểm: 774354345 | Địa chỉ: Da Nang Da Nang

Thông tin khám bệnh

Bác sĩ phụ trách: Lê Ngọc Anh | Dịch vụ khám: Khám thai định kỳ | Ngày khám: 27/11/2025

Cả khám: Ca sáng (07:30 - 11:30) | Ghi chú: Không có ghi chú

Gói dịch vụ bệnh nhân đã chọn

Khám thai định kỳ

- Theo dõi sự phát triển của thai nhi và sức khỏe của mẹ bầu
- Siêu âm
- Xét nghiệm máu và nước tiểu
- Tư vấn dinh dưỡng.

Khoa: Khoa Sản

Chẩn đoán & Kết luận

Chuẩn đoán ban đầu:

Nhập chuẩn đoán...

Kết luận bác sĩ:

Nhập kết luận bác sĩ...

Trạng thái lịch hẹn:

Đã duyệt

Toa thuốc (mỗi dòng 1 loại thuốc):

Ví dụ: Paracetamol 500mg - Uống 2 lần/ngày
Vitamin C 1000mg - Sáng 1 viên

Lưu thông tin

Yêu cầu xét nghiệm

Chọn loại xét nghiệm: -- Chọn loại -- | Tên xét nghiệm: Nhập tên xét nghiệm

Chọn để tự động điền Tên xét nghiệm và Khoa.

Khoa phụ trách: -- Chọn khoa --

Ghi chú: Ghi chú thêm (nếu có)

Gửi yêu cầu xét nghiệm

Hình 3.24. Giao diện yêu cầu xét nghiệm của Bác sĩ

Giao diện quản lý hồ sơ cá nhân dành cho bác sĩ. Tại đây, bác sĩ có thể xem và cập nhật thông tin như họ tên, khoa làm việc, chuyên môn, số điện thoại, địa chỉ liên hệ... giao diện giúp đảm bảo thông tin bác sĩ luôn đầy đủ và chính xác, phục vụ cho việc hiển thị cho bệnh nhân và các chức năng nội bộ khác.

Bệnh viện Phúc An | Bảng điều khiển | Lịch sử khám | **Lê Ngọc Anh**

HỒ SƠ BÁC SĨ | [← Quay lại](#)

Thông tin cá nhân & chuyên môn

Ảnh đại diện

Thông tin cá nhân

Họ và tên: Lê Ngọc Anh

Email: anh@example.com | Số điện thoại: 0787548479

Ngày sinh: 21 Tháng 11 1986 | Giới tính: Nữ

Địa chỉ: Quốc Lộ 1A

Thông tin chuyên môn

Khoa: Khoa Sản | Chuyên môn: Tiền sản, Thai sản.

Số giấy phép hành nghề: 1234567

Ảnh giấy phép hành nghề:

Mô tả bản thân:

Tôi là bác sĩ chuyên khoa Răng – Hàm – Mặt với hơn 15 năm kinh nghiệm trong lĩnh vực khám, điều trị và thẩm mỹ nha khoa. Trong suốt quá trình làm nghề, tôi luôn đặt tiêu chí an toàn – tận tâm – chính xác lên hàng đầu, nhằm mang lại cho mỗi bệnh nhân không chỉ một nụ cười khỏe mạnh mà còn là sự tự tin và thoải mái khi điều trị.

Tôi có kinh nghiệm sâu trong các lĩnh vực:

Lưu thông tin

Thay đổi mật khẩu

Mật khẩu hiện tại: | Mật khẩu mới: | Xác nhận mật khẩu mới:

Cập nhật mật khẩu

Hình 3.25. Giao diện quản lý hồ sơ cá nhân của bác sĩ

3.2. Cài đặt chương trình

Hệ thống bệnh viên Phúc An cung cấp các chức năng như quản lý thông tin bệnh nhân, bác sĩ, đặt lịch khám, lưu trữ hồ sơ khám, quản lý dịch vụ y tế và các khoa. Ngoài ra hệ thống còn có chatbox cho admin và gửi mail thông báo cho bệnh nhân đã đặt lịch.

3.2.1. Cài đặt phát triển mã nguồn admin

Mã nguồn quản lý lịch khám.

Mã nguồn hiện thực chức năng quản lý lịch khám trong hệ thống. Mã nguồn xây dựng các phương thức cho phép admin xem danh sách lịch hẹn, lọc theo nhiều tiêu chí (tên bệnh nhân, bác sĩ, ngày khám, trạng thái), xác nhận hoặc cập nhật trạng thái lịch hẹn và xóa lịch hẹn khi cần thiết. Việc sử dụng eloquent orm kết hợp với cơ chế phân trang giúp mã nguồn ngắn gọn, dễ bảo trì, đồng thời vẫn đảm bảo hiệu năng khi truy vấn dữ liệu lịch hẹn số lượng lớn.

```

1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Appointment;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Mail;
9  use App\Mail\AppointmentApprovedMail;
10
11 class AppointmentController extends Controller
12 {
13     /**
14      * Hiển thị danh sách lịch hẹn kèm bộ lọc tìm kiếm.
15      *
16      * 📌 Mục đích:
17      * - Hiển thị danh sách tất cả lịch hẹn trong hệ thống (dành cho admin).
18      * - Cho phép admin lọc dữ liệu theo nhiều tiêu chí khác nhau (tên bệnh nhân, bác sĩ, bảo hiểm, ngày hẹn, trạng thái).
19      * - Kết quả được phân trang và hiển thị trên view 'admin.appointments.index'.
20      */
21     public function index(Request $request)
22     {
23         // Khởi tạo query gốc với các quan hệ liên quan để tránh N+1 query:
24         // - patient.user → thông tin người dùng của bệnh nhân
25         // - doctor.user → thông tin người dùng của bác sĩ
26         // - service → dịch vụ khám bệnh
27         $query = Appointment::with(['patient.user', 'doctor.user', 'service']);
28
29         // 🔍 Lọc theo tên bệnh nhân nếu có nhập từ form tìm kiếm
30         if ($request->filled('patient_name')) {
31             $query->whereHas('patient.user', function ($q) use ($request) {
32                 $q->where('name', 'like', '%' . $request->patient_name . '%');
33             });
34         }
35
36         // 🔍 Lọc theo tên bác sĩ
37         if ($request->filled('doctor_name')) {
38             $query->whereHas('doctor.user', function ($q) use ($request) {
39                 $q->where('name', 'like', '%' . $request->doctor_name . '%');
40             });
41         }
42
43         // 🔍 Lọc theo mã bảo hiểm của bệnh nhân
44         if ($request->filled('insurance_number')) {
45             $query->whereHas('patient.user', function ($q) use ($request) {

```

Hình 3.26. Mã nguồn quản lý lịch khám

Mã nguồn phát triển chức năng quản lý bác sĩ.

Mã nguồn của module quản lý bác sĩ. Đoạn mã cài đặt các chức năng cơ bản như: liệt kê danh sách bác sĩ, thêm mới, chỉnh sửa thông tin và xóa bác sĩ. Khi thêm hoặc sửa, hệ thống đồng thời quản lý cả tài khoản đăng nhập (bảng users) lẫn hồ sơ bác sĩ (bảng doctors), bảo đảm ràng buộc giữa thông tin chuyên môn và tài khoản người dùng. Mã nguồn cũng kiểm tra ràng buộc khóa ngoại với bảng lịch hẹn, không cho phép xóa bác sĩ khi vẫn còn lịch khám liên quan, qua đó đảm bảo toàn vẹn dữ liệu.

```
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Doctor;
7  use App\Models\User;
8  use App\Models\Department;
9  use App\Models\Appointment;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\DB;
12 use Carbon\Carbon;
13
14 class DoctorController extends Controller
15 {
16     /**
17      * Hiển thị danh sách bác sĩ (admin).
18      */
19     public function index()
20     {
21         $doctors = Doctor::with(['user', 'department'])->paginate(10);
22         return view('admin.doctors.index', compact('doctors'));
23     }
24
25     /**
26      * Hiển thị form thêm mới bác sĩ.
27      */
28     public function create()
29     {
30         $departments = Department::all();
31         return view('admin.doctors.create', compact('departments'));
32     }
33
34     /**
35      * Lưu bác sĩ mới vào CSDL.
36      */
37     public function store(Request $request)
38     {
39         $request->validate([
40             'name' => 'required|string|max:255',
41             'email' => 'required|email|unique:users,email',
42             'password' => 'required|min:6',
43             'phone' => 'nullable|string|max:20',
44             'gender' => 'nullable|in:male,female,other',
45             'address' => 'nullable|string|max:255',
```

Hình 3.27. Mã nguồn phát triển chức năng quản lý bác sĩ

Mã nguồn phát triển chức năng quản lý người dùng.

Mã nguồn xử lý chức năng quản lý người dùng trong hệ thống. Tại đây, admin có thể xem danh sách tài khoản, chỉnh sửa thông tin cơ bản (họ tên, email, vai trò, trạng thái) và xóa tài khoản khi cần thiết. Logic xử lý được tổ chức trong controller chuyên biệt, kết hợp với eloquent giúp thao tác với bảng users một cách rõ ràng, tách bạch giữa tầng giao diện và tầng truy cập dữ liệu, thuận tiện cho việc mở rộng thêm các quyền hay phân hệ trong tương lai.

```
1  <?php
2  namespace App\Http\Controllers\Admin;
3
4  use App\Http\Controllers\Controller;
5  use App\Models\Patient;
6  use App\Models\User;
7  use Illuminate\Http\Request;
8
9  class PatientController extends Controller
10 {
11     /**
12      * Hiển thị danh sách tất cả bệnh nhân.
13      *
14      * 📌 Mục đích:
15      * - Lấy danh sách bệnh nhân cùng thông tin user liên kết.
16      * - Hiển thị trên trang quản lý bệnh nhân cho admin.
17      */
18     public function index()
19     {
20         // Lấy danh sách bệnh nhân kèm thông tin người dùng (quan hệ user)
21         // latest() → sắp xếp theo thời gian tạo mới nhất
22         // paginate(10) → hiển thị 10 bản ghi mỗi trang
23         $patients = Patient::with('user')->latest()->paginate(10);
24
25         // Trả về view 'admin.patients.index' với biến $patients
26         return view('admin.patients.index', compact('patients'));
27     }
28
29     /**
30      * Hiển thị form thêm mới bệnh nhân.
31      *
32      * 📌 Dùng để admin nhập thông tin bệnh nhân mới (họ tên, email, sdt,...)
33      */
34     public function create()
35     {
36         // Trả về giao diện form thêm bệnh nhân
37         return view('admin.patients.create');
38     }
39
40     /**
41      * Lưu thông tin bệnh nhân mới vào cơ sở dữ liệu.
42      *
43      * 📌 Luồng xử lý:
44      * 1 Validate dữ liệu đầu vào.
45      * 2 Tạo user tương ứng (vì bệnh nhân cũng là một user trong hệ thống).
```

Hình 3.28. Mã nguồn phát triển chức năng quản lý người dùng

Mã nguồn phát triển chức năng thông báo mail.

Mã nguồn hiện thực chức năng gửi email thông báo lịch hẹn. Khi admin xác nhận một lịch khám, controller cập nhật trạng thái lịch hẹn sang “đã xác nhận” và khởi tạo một lớp mail (mailable) để gửi thông tin chi tiết lịch hẹn đến địa chỉ email của bệnh nhân. Mã nguồn sử dụng tiện ích gửi mail sẵn có của framework, tách riêng phần nội dung email ở file view, giúp việc thay đổi mẫu email (giao diện appointment_approved) trở nên linh hoạt mà không ảnh hưởng đến logic xử lý nghiệp vụ.

```

1  <?php
2
3  namespace App\Mail;
4
5  use Illuminate\Bus\Queueable;
6  use Illuminate\Mail\Mailable;
7  use Illuminate\Queue\SerializesModels;
8  use App\Models\Appointment;
9
10 class AppointmentApprovedMail extends Mailable
11 {
12     use Queueable, SerializesModels;
13
14     public Appointment $appointment;
15
16     public function __construct(Appointment $appointment)
17     {
18         $this->appointment = $appointment->loadMissing(['patient', 'doctor.user', 'service']);
19     }
20
21     public function build()
22     {
23         return $this->subject('Lịch hẹn của bạn đã được xác nhận')
24             ->view('emails.appointment_approved');
25     }
26 }
27
28 |<h2>Xin chào {{ $appointment->patient->name ?? 'Quý khách' }}</h2>
29
30 |<p>Lịch hẹn của bạn đã được <strong>xác nhận</strong>.</p>
31
32 |<ul>
33 |<li><strong>Dịch vụ:</strong> {{ $appointment->service->name ?? 'N/A' }}</li>
34 |<li><strong>Bác sĩ:</strong> {{ $appointment->doctor->user->name ?? 'N/A' }}</li>
35 |<li><strong>Thời gian:</strong> {{ \Carbon\Carbon::parse($appointment->appointment_date)->format('d/m/Y') }}</li>
36 |<li><strong>Ca khám:</strong> {{ $appointment->medical_examination ?? '---' }}</li>
37 |</ul>
38
39 |@php
40 |use Carbon\Carbon;
41
42 |$price = $appointment->total ?? ($appointment->service->price ?? 0);
43 |$birthdate = $appointment->patient->birthdate ?? null;
44
45 |// Mặc định thanh toán 80%
46 |$prepayPercent = 80;
47 |$discountPercent = 0;
48
49 |// Nếu sinh trong tháng → tổng thanh toán thực tế = 70%
50 |if ($birthdate && Carbon::parse($birthdate)->format('m') == now()->format('m')) {
51 |    $discountPercent = 10; // Giảm 10%
52 |    $prepayPercent = 70; // Tổng thanh toán thực tế chỉ 70% giá gốc
53 |}
54
55 |// Tính toán
56 |$discountAmount = $price * ($discountPercent / 100);
57 |$finalPrice = $price - $discountAmount; // giá sau giảm (chỉ để hiển thị)
58 |$prepayAmount = $price * ($prepayPercent / 100); // thanh toán theo tỉ lệ thực tế
59 |$remainAmount = max(0, $finalPrice - $prepayAmount);
60
61 |@endphp

```

Hình 3.29. Mã nguồn phát triển chức năng thông báo mail

3.2.2. Cài đặt phát triển mã nguồn bệnh nhân

Mã nguồn phát triển chức năng đặt lịch khám.

Mã nguồn xử lý chức năng đặt lịch khám phía bệnh nhân. Đoạn mã thực hiện các bước: kiểm tra tài khoản đã gắn với hồ sơ bệnh nhân hay chưa, kiểm tra tính hợp lệ của dữ liệu đầu vào (bác sĩ, dịch vụ, ngày khám, ca khám), giới hạn số ca khám tối đa cho mỗi bác sĩ trong một buổi và kiểm tra trùng lịch trước khi lưu. Nếu tất cả điều kiện đều thỏa mãn, hệ thống tạo bản ghi lịch hẹn mới với trạng thái “đang chờ xác nhận”. Nhờ đó, quy trình đặt lịch bảo đảm tính chính xác và tránh xung đột lịch khám.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Schema;
7  use Illuminate\Support\Facades\Log;
8  use Illuminate\Support\Facades\DB;
9  use App\Models\{Appointment, Service};
10 use Carbon\Carbon;
11
12 class PaymentController extends Controller
13 {
14     public function vnpay_payment(Request $request)
15     {
16         try {
17             Log::info('VNPay Payment Request:', $request->all());
18
19             $validated = $request->validate([
20                 'patient_id'      => 'required|exists:patients,id',
21                 'doctor_id'       => 'required|exists:doctors,id',
22                 'service_id'      => 'required|exists:services,id',
23                 'appointment_date' => 'required|date',
24                 'appointment_time' => 'required',
25                 'medical_examination' => 'required|in:Ca sáng (07:30 - 11:30),Ca chiều (13:00 - 17:00)',
26                 'note'            => 'nullable|string',
27             ]);
28
29             $service = Service::findOrFail($validated['service_id']);
30
31             DB::beginTransaction();
32
33             $appointmentData = [
34                 'patient_id'      => $validated['patient_id'],
35                 'doctor_id'       => $validated['doctor_id'],
36                 'service_id'      => $validated['service_id'],
37                 'appointment_date' => $validated['appointment_date'],
38                 'medical_examination' => $validated['medical_examination'],
39                 'payment_status'  => 'failed',
40                 'note'            => $validated['note'] ?? null,
41             ];
42
43             if (Schema::hasColumn('appointments', 'appointment_time')) {
44                 $appointmentData['appointment_time'] = $validated['appointment_time'];
45             }

```

Hình 3.30. Mã nguồn phát triển chức năng đặt lịch khám

Mã nguồn phát triển chức năng quản lý thông tin cá nhân bệnh nhân.

Mã nguồn phục vụ chức năng quản lý thông tin cá nhân của bệnh nhân. Thông qua các phương thức tương ứng, hệ thống cho phép bệnh nhân lấy ra thông tin hồ sơ hiện tại và cập nhật các trường như họ tên, ngày sinh, địa chỉ, số điện thoại, thông tin bảo hiểm... Mã nguồn có sử dụng cơ chế xác thực và ràng buộc quyền truy cập, đảm bảo mỗi bệnh nhân chỉ được phép xem và chỉnh sửa hồ sơ của chính mình, từ đó bảo vệ tính riêng tư và an toàn dữ liệu cá nhân.

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Http\Requests\ProfileUpdateRequest;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Redirect;
10 use Illuminate\Support\Facades\Storage;
11 use Illuminate\View\View;
12
13 class ProfileController extends Controller
14 {
15     /**
16      * Hiển thị trang chính của hồ sơ cá nhân.
17      * Ví dụ: /profile
18      */
19     public function index()
20     {
21         // Trả về giao diện trang hồ sơ
22         return view('profile.index');
23     }
24
25     /**
26      * Hiển thị trang chỉnh sửa hồ sơ người dùng.
27      * Truyền dữ liệu user hiện tại vào view.
28      */
29     public function edit(Request $request): View
30     {
31         return view('profile.edit', [
32             'user' => $request->user(), // lấy thông tin người dùng đang đăng nhập
33         ]);
34     }
35
36     /**
37      * Cập nhật thông tin hồ sơ người dùng,
38      * đồng thời đồng bộ với bảng "patients".
39      */
40     public function update(ProfileUpdateRequest $request)
41     {
42         $user = $request->user(); // Lấy người dùng hiện tại
43         $validated = $request->validated(); // Lấy dữ liệu đã được xác thực từ request
44
45         // Gán dữ liệu hợp lệ vào model user
```

Hình 3.31. Mã nguồn phát triển chức năng quản lý thông tin cá nhân bệnh nhân

3.2.3. Cài đặt phát triển mã nguồn bác sĩ

Mã nguồn phát triển chức năng danh sách lịch khám.

Mã nguồn hiện thực chức năng hiển thị danh sách lịch khám ở phía bệnh nhân. Mã nguồn truy vấn bảng appointments theo patient_id tương ứng với bệnh nhân đang đăng nhập, đồng thời nạp kèm thông tin liên quan như bác sĩ, dịch vụ và trạng thái lịch hẹn. Kết quả được phân trang và truyền sang view để hiển thị dưới dạng bảng. Cách tổ chức này giúp bệnh nhân dễ dàng theo dõi toàn bộ lịch sử đặt lịch và các cuộc hẹn sắp tới.

```

1  <?php
2
3  namespace App\Http\Controllers\Doctor;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\User;
7  use App\Models\Doctor;
8  use App\Models\Appointment;
9  use Illuminate\Http\Request;
10 use Illuminate\Support\Facades\Auth;
11 use Illuminate\Support\Facades\Hash;
12
13 class DoctorDashboardController extends Controller
14 {
15     /**
16      * Hiển thị trang dashboard của bác sĩ
17      */
18     public function index(Request $request)
19     {
20         $doctor = Auth::user()->doctor ?? null;
21
22         if (!$doctor) {
23             return view('doctor.dashboard', [
24                 'error' => 'Không tìm thấy thông tin bác sĩ!',
25                 'appointments' => collect(),
26                 'pending' => 0,
27                 'confirmed' => 0,
28                 'completed' => 0,
29                 'total' => 0,
30                 'selectedDate' => now()->format('Y-m-d'),
31             ]);
32         }
33
34         // Ngày được chọn (mặc định hôm nay)
35         $selectedDate = $request->query('date', now()->format('Y-m-d'));
36
37         // Lấy danh sách lịch hẹn trong ngày được chọn
38         $appointments = Appointment::with(['patient', 'service', 'doctor.user'])
39             ->where('doctor_id', $doctor->id)
40             ->whereDate('appointment_date', $selectedDate)
41             ->orderBy('appointment_date')
42             ->get();
43
44         // Thống kê theo trạng thái
45         $pending = $appointments->where('status', 'pending')->count();

```

Hình 3.32. Mã nguồn phát triển chức năng danh sách lịch khám

Mã nguồn phát triển chức năng hồ sơ khám.

Mã nguồn chức năng quản lý hồ sơ khám bệnh. Đoạn mã chịu trách nhiệm tạo mới hoặc lấy hồ sơ khám gắn với từng lịch hẹn, cho phép bác sĩ ghi nhận chẩn đoán, kết luận, toa thuốc, danh sách các dịch vụ đã thực hiện, kèm tệp đính kèm như hình ảnh kết quả cận lâm sàng. Mã nguồn cũng cung cấp phương thức cập nhật hồ sơ và đồng bộ trạng thái lịch hẹn (ví dụ chuyển sang “đã hoàn thành” sau khi khám xong). Việc tách riêng medicalrecord và liên kết với appointment giúp hệ thống lưu trữ hồ sơ khám một cách có cấu trúc, thuận lợi cho việc tra cứu và thống kê sau này.

```
1  <?php
2
3  namespace App\Http\Controllers\Doctor;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use App\Models\{Appointment, Patient, MedicalRecord, Service};
8  use Illuminate\Support\Facades\Log;
9  use Illuminate\Support\Facades\Storage;
10
11 class DoctorRecordController extends Controller
12 {
13     /**
14      * Hiển thị hồ sơ khám bệnh của bệnh nhân
15      */
16     public function showPatientRecord($appointmentId)
17     {
18         $appointment = Appointment::with(['patient', 'doctor.user', 'service'])->findOrFail($appointmentId);
19
20         $services = Service::with('department')->orderBy('name')->get();
21
22         $record = MedicalRecord::firstOrCreate(
23             ['appointment_id' => $appointmentId],
24             [
25                 'patient_id' => $appointment->patient_id,
26                 'description' => null,
27                 'diagnosis' => null,
28                 'doctor_conclusion' => null,
29                 'prescription' => [],
30             ]
31         );
32
33         return view('doctor.patient_record', compact('appointment', 'record', 'services'))
34             ->with('patient', $appointment->patient);
35     }
36
37     /**
38      * Cập nhật thông tin chẩn đoán, kết luận và toa thuốc.
39      */
40     public function update(Request $request, MedicalRecord $record)
41     {
42         $request->validate([
43             'diagnosis' => 'nullable|string|max:1000',
44             'doctor_conclusion' => 'nullable|string|max:1000',
45             'prescription' => 'nullable|string|max:2000',
```

Hình 3.33. Mã nguồn phát triển chức năng hồ sơ khám

3.4. Kết luận chương 3

Chương 3 đã xây dựng giao diện thiết kế và triển khai hệ thống quản lý của bệnh viện và xây dựng các chức năng chính. Qua quá trình phân tích và bố trí các thành phần của hệ thống, có thể thấy kiến trúc đã được xây dựng theo hướng rõ ràng tách biệt từng phần và dễ mở rộng.

Về giao diện cho quản trị viên, hệ thống đã đáp ứng đủ nhu cầu quản lý thiết yếu của bệnh viện nhưng quan lý lịch hẹn, bác sĩ, khoa, dịch vụ, loại xét nghiệm và quản lý tài khoản người dùng. Các chức năng xây dựng theo hướng trực quan, dễ thao tác, đảm bảo thống nhất giao diện để người quản trị thao tác một cách hiệu quả.

Đối với giao diện dành cho khách hàng, hệ thống đã cung cấp đầy đủ các trang và chức năng hỗ trợ quá trình tư vấn theo triệu chứng, xem thông tin đến đặt lịch khám. Từ trang chủ đến chi tiết lịch hẹn, đến các trang quản lý thông tin cá nhân, tất cả đều thống nhất thiết kế giao diện theo hướng thân thiện để sử dụng phù hợp với nhiều thiết bị khác nhau. Điều này mang đến trải nghiệm linh hoạt, thuận tiện đồng thời giúp bệnh nhân dễ tương tác với hệ thống.

Tổng thể chương 3 đã hoàn thiện phần thiết kế và triển khai tạo nên một hệ thống bệnh viện đầy đủ chức năng. Hệ thống hiện đã sẵn sàng để chuyển sang các giai đoạn tiếp theo như kiểm thử, đánh giá và triển khai thực tế, góp phần nâng cao hiệu quả quản lý và chất lượng phục vụ của bệnh viện.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trên đây là toàn bộ báo cáo đồ án tốt nghiệp. Như vậy, sau một thời gian tìm hiểu, nghiên cứu thì em đã hoàn thành đồ án tốt “Xây dựng ứng dụng quản lý hồ sơ bệnh nhân và dịch vụ y tế tại bệnh viện PHÚC AN”.

1. Kết quả đạt được

Trong quá trình thực hiện đồ án, em đã củng cố và nâng cao kiến thức về công nghệ web, đặc biệt là Laravel Framework. Em đã hiểu rõ hơn về quy trình phân tích yêu cầu, thiết kế hệ thống, xây dựng cơ sở dữ liệu, phát triển chức năng và triển khai vận hành ứng dụng. Thông qua quá trình này, em đã biết cách vận dụng linh hoạt các kiến thức đã học vào thực tế, từ đó hình thành tư duy lập trình có hệ thống và khả năng giải quyết vấn đề hiệu quả hơn.

Sản phẩm cuối cùng - ứng dụng web đặt lịch khám bệnh trực tuyến - đã hoàn thiện đầy đủ các chức năng theo mục tiêu đề ra ban đầu. Ứng dụng cho phép bệnh nhân dễ dàng đăng ký tài khoản, đặt lịch hẹn với bác sĩ, tra cứu thông tin và theo dõi lịch sử khám bệnh. Giao diện được thiết kế thân thiện, trực quan và tối ưu trải nghiệm người dùng, giúp thao tác trở nên nhanh chóng và thuận tiện. Hệ thống hoạt động ổn định, đảm bảo tính chính xác, bảo mật và hiệu năng tốt, đáp ứng hiệu quả nhu cầu thực tế của người dùng và mục tiêu học tập của đề tài.

2. Hạn chế

Mặc dù đã đạt được nhiều kết quả tích cực, em vẫn nhận thấy đồ án còn tồn tại một số hạn chế cần được khắc phục trong giai đoạn tiếp theo. Về hiệu suất, ứng dụng có thể chưa ổn định khi số lượng người dùng và dữ liệu tăng cao, do đó cần tối ưu hóa truy vấn cơ sở dữ liệu, phân trang hợp lý, áp dụng bộ nhớ đệm (cache) và hạn chế vấn đề N+1 query để đảm bảo hệ thống vận hành mượt mà, ổn định hơn. Về bảo mật, cần tăng cường các biện pháp bảo vệ dữ liệu nhạy cảm, kiểm soát truy cập chặt chẽ theo vai trò người dùng, bổ sung cơ chế xác thực đa yếu tố cho các tài khoản quan trọng, đồng thời thiết lập hệ thống giám sát và ghi log các truy cập bất thường để nâng cao an toàn thông tin. Ngoài ra, về mặt tính năng, ứng dụng hiện mới đáp ứng được các nhu cầu cơ bản; trong tương lai, cần mở rộng thêm các tiện ích như chức năng đánh giá dịch vụ và bác sĩ, hệ thống nhắc lịch tự động, gợi ý dịch vụ phù hợp, cũng như báo cáo và thống kê chi tiết hơn nhằm mang lại trải nghiệm đa dạng, thân thiện và toàn diện hơn cho người dùng.

3. Hướng phát triển

Dựa trên các hạn chế đã được nhận diện, hướng phát triển sắp tới của hệ thống sẽ tập trung vào bốn trụ cột chính. Thứ nhất, tối ưu hiệu suất hệ thống bằng cách rà soát và tinh chỉnh truy vấn cơ sở dữ liệu, bổ sung chỉ mục, hạn chế tình trạng N+1 query, áp dụng cache cho dữ liệu tĩnh hoặc cache từng phần (partial cache), đồng thời phân trang hợp lý và chuyển các tác vụ xử lý nặng sang hàng đợi (queue) để đảm bảo hệ thống vận hành mượt mà ngay cả khi lượng truy cập tăng cao. Thứ hai, nâng cấp bảo mật nhằm bảo vệ tốt hơn dữ liệu người dùng, bao gồm việc mã hóa thông tin nhạy cảm, bổ sung xác thực hai yếu tố (2FA) cho các tài khoản quan trọng, kiểm soát phân quyền chặt chẽ theo vai trò (bệnh nhân, bác sĩ, nhân viên, quản trị viên), ghi log và giám sát truy cập bất thường, đồng thời áp dụng giới hạn tần suất (rate limiting) đối với các điểm API nhạy cảm. Thứ ba, mở rộng tính năng để nâng cao trải nghiệm người dùng, chẳng hạn như cho phép đánh giá chất lượng dịch vụ hoặc bác sĩ, gửi nhắc lịch tự động qua email, SMS hoặc thông báo đẩy, tích hợp tính năng tư vấn và gợi ý dịch vụ y tế bằng AI dựa trên lịch sử khám và triệu chứng của bệnh nhân, cùng với việc bổ sung hệ thống báo cáo và thống kê chuyên sâu phục vụ công tác quản trị. Cuối cùng, phát triển đa nền tảng nhằm mở rộng phạm vi sử dụng, bao gồm triển khai phiên bản PWA hoặc xây dựng ứng dụng di động (Flutter/React Native) dùng chung API, tối ưu hóa giao diện và trải nghiệm người dùng (UI/UX) trên thiết bị di động, đồng thời hỗ trợ thông báo đẩy (push notification) để mang lại sự tiện lợi và tương tác cao hơn cho người dùng.

TÀI LIỆU THAM KHẢO

- [1] https://hocwebchuan.com/tutorial/laravel/install_composer.php
- [2] <https://viblo.asia/p/giai-ma-ve-lap-trinh-web-front-end-back-end-va-full-stack-07LKXmPJZV4>
- [3] <https://viblo.asia/p/task-scheduling-cung-nhau-doi-no-theo-mot-2019-part-2-djeZ14doKWz>
- [4] <https://viblo.asia/p/tich-hop-cong-thanh-toan-vnpay-voi-laravel-Az45bGD6KxY>
- [5] <https://viblo.asia/p/huong-dan-tinh-nang-gui-mail-trong-laravel-qua-mini-project-OeVKBvjJKkW>
- [6] <https://fptshop.com.vn/tin-tuc/danh-gia/tim-hieu-mysql-la-gi-172297>
- [7] <https://fptshop.com.vn/tin-tuc/danh-gia/visual-studio-code-la-gi-cac-tinh-nang-noi-bat-cua-visual-studio-code-146213>