

**ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP  
NGÀNH: CÔNG NGHỆ THÔNG TIN  
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

**XÂY DỰNG ỨNG DỤNG ĐẶT CHỖ  
KHÁM BỆNH**

Người hướng dẫn: **THS. NGUYỄN THỊ MINH HỶ**

Sinh viên thực hiện: **VÕ VĂN TRINH**

Số thẻ sinh viên: **102150207**

Lớp: **15T3**

Đà Nẵng, 12/2019

# **NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN**

*Dà Nẵng, ngày ... tháng ... năm 2019*  
Gảng viên hướng dẫn

Ths. Nguyễn Thị Minh Hỷ

# NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN

*Đà Nẵng, ngày ... tháng ... năm 2019*

# PHIẾU DUYỆT ĐỒ ÁN TỐT NGHIỆP

## I. Phần dành cho Sinh viên

1. Họ và tên: VÕ VĂN TRINH      2. Mã Sinh viên: 102150207      3. Lớp: 15T3

4. Tên đề tài: Xây dựng ứng dụng đặt chỗ khám bệnh

5. Số điện thoại: 0974076085      6. E-mail: vovantrinhhdhbkdn@gmail.com

7. Họ và tên GVHD: ThS. Nguyễn Thị Minh Hỷ

## II. Phần dành cho Hội đồng

STT	Nội dung đánh giá	Kết luận
1.	Trình bày báo cáo theo đúng mẫu qui định của Khoa	
2.	Không có sự sao chép nội dung báo cáo và chương trình đã có	
3.	Biên dịch mã nguồn và chạy được chương trình	
4.	Có kịch bản thực hiện với dữ liệu thử nghiệm	
5.	Kết quả thực hiện chương trình đúng theo báo cáo	
6.	Có sự đóng góp, phát triển của tác giả trong đồ án	

Ý kiến khác: .....

Kết luận:       Đạt yêu cầu       Phải sửa chữa lại       Không đạt yêu cầu

Đà Nẵng, ngày tháng 12 năm 2019

Chủ tịch Hội đồng

(Ký và ghi họ tên)

Cán bộ duyệt kiểm tra

(Ký và ghi họ tên)

TÓM TẮT

## Tên đề tài: Xây dựng ứng dụng đặt chỗ khám bệnh

Sinh viên thực hiện: Võ Văn Trinh

Số thẻ SV: 102150207 Lớp: 15T3

Ứng dụng đặt chỗ khám bệnh cung cấp cho các bác sĩ, phòng khám, và bệnh viện có thể đăng ký lịch khám một cách dễ dàng, đồng thời giúp các bác sĩ có thể chủ động đăng ký lịch khám phù hợp với thời gian của bản thân. Bên cạnh đó, cung cấp cho người dùng một ứng dụng thân thiện giúp việc đặt lịch khám chữa bệnh một cách nhanh chóng và không mất quá nhiều thời gian cho việc khám bệnh. Mọi người cũng có thể chọn lịch phù hợp với quỹ thời gian cá nhân để lựa chọn và đặt lịch trên ứng dụng.

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: VÕ VĂN TRINH

Số thẻ sinh viên: 102150207

Lớp: 15T3

Khoa: Công nghệ thông tin

Ngành: Công nghệ thông tin

1. *Tên đề tài đồ án:*

Xây dựng ứng dụng đặt chỗ khám bệnh

2. *Đề tài thuộc diện:*  Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. *Các số liệu và dữ liệu ban đầu:*

Không có.

4. *Nội dung các phần thuyết minh và tính toán:*

Nội dung của thuyết minh gồm:

**Mở đầu:** Phần mở đầu của luận văn, giới thiệu về nhu cầu thực tế và lý do thực hiện đề tài, đồng thời giới thiệu sơ lược về đề tài và mục tiêu phải đạt được.

**Chương 1.** Cơ sở lý thuyết: trình bày những lý thuyết học được và đã áp dụng vào hệ thống.

**Chương 2.** Phân tích và thiết kế: trình bày các hồ sơ phân tích và hồ sơ thiết kế trong xây dựng hệ thống và luồng hoạt động của hệ thống.

**Chương 3.** Triển khai và đánh giá kết quả: mô tả cách cài đặt, vận hành hệ thống và đánh giá kết quả đạt được.

5. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*

Không có.

6. *Họ tên người hướng dẫn:* THS. NGUYỄN THỊ MINH HỶ

7. *Ngày giao nhiệm vụ đồ án:* ...../...../2019

8. *Ngày hoàn thành đồ án:* ...../...../2019

Đà Nẵng, ngày tháng 12 năm 2019

Trưởng Bộ môn .....

Người hướng dẫn

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến các thầy cô trong Khoa Công nghệ thông tin, cũng như tất cả các thầy cô trong trường Đại học Bách khoa – Đại học Đà Nẵng đã dùi dắt, dạy dỗ và truyền đạt kiến thức, kinh nghiệm quý báu của mình trong suốt quá trình em học tập và nghiên cứu tại trường.

Em xin bày tỏ tình cảm và lòng biết ơn chân thành của em tới cô giáo Ths. Nguyễn Thị Minh Hỷ, người đã từng bước hướng dẫn, giúp đỡ em tận tình trong quá trình thực hiện đồ án tốt nghiệp của mình. Nhờ đó em có thể hoàn thành đồ án đúng tiến độ và tích lũy cho mình nhiều kiến thức quý báu.

Con xin gửi lời cảm ơn to lớn nhất đến cha, mẹ và gia đình. Cha, mẹ và gia đình đã luôn ở bên con, là nguồn động lực không mệt mỏi và là chỗ dựa tinh thần vững chắc giúp con vượt qua những khó khăn để hoàn thành đồ án này.

Mặc dù đã cố gắng hoàn thành đồ án tốt nhất nhưng thời gian và kiến thức còn có hạn nên sẽ không tránh khỏi những thiếu sót nhất định, rất mong được sự cảm thông và tận tình chỉ bảo, góp ý của quý thầy cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn!

Võ Văn Trinh

## **CAM ĐOAN**

Em xin cam đoan:

1. Nội dung trong đồ án này là do em thực hiện dưới sự hướng dẫn trực tiếp của ThS Nguyễn Thị Minh Hỷ.
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, em xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày tháng 12 năm 2019

Sinh viên thực hiện

**Võ Văn Trinh**

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	i
<b>CAM ĐOAN.....</b>	ii
<b>MỤC LỤC.....</b>	iii
<b>DANH SÁCH CÁC BẢNG .....</b>	v
<b>DANH SÁCH CÁC HÌNH VẼ.....</b>	vi
<b>DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....</b>	viii
<b>GIỚI THIỆU ĐỀ TÀI.....</b>	1
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG .....</b>	3
1.1. KIẾN THỨC CƠ BẢN .....	3
1.2. TỔNG QUAN VỀ JAVA VÀ SPRING FRAMEWORK .....	3
1.3. TỔNG QUAN RESTFULL API .....	10
1.4. TỔNG QUAN VỀ REACTJS .....	13
1.5. CÔNG CỤ SỬ DỤNG.....	18
<b>CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	20
2.1. KHẢO SÁT BÀI TOÁN THỰC TẾ .....	20
2.2. PHÂN TÍCH CHỨC NĂNG .....	21
2.2.1. Khách có các chức năng sau.....	21
2.2.2. Thành viên có các chức năng sau.....	22
2.2.3. Bác sĩ có các chức năng sau .....	23
2.2.4. Quản trị viên có các chức năng sau .....	23
2.3. SƠ ĐỒ CA SỬ DỤNG .....	24
2.3.1. Các tác nhân trong hệ thống .....	24
2.3.2. Sơ đồ ca sử dụng tổng quát của hệ thống .....	25
2.3.3. Sơ đồ ca sử dụng đăng ký tài khoản của khách.....	26
2.3.4. Sơ đồ ca sử dụng của thành viên .....	26
2.3.5. Sơ đồ ca sử dụng của bác sĩ.....	28
2.3.6. Sơ đồ ca sử dụng của quản trị viên .....	29
2.4. ĐẶC TẢ CA SỬ DỤNG.....	30
2.4.1. Đặc tả ca sử dụng Đăng ký .....	30
2.4.2. Đặc tả ca sử dụng đăng nhập.....	31
2.4.3. Đặc tả ca sử dụng quản lý tài khoản .....	31
2.4.4. Đặc tả ca sử dụng đăng ký trở thành bác sĩ.....	32
2.4.5. Đặc tả ca sử dụng đặt lịch khám bệnh.....	32
2.4.6. Đặc tả ca sử dụng đánh giá bác sĩ .....	33

2.4.7. <i>Đặc tả ca sử dụng bình luận bác sĩ</i> .....	33
2.4.8. <i>Đặc tả ca sử dụng quản lý người dùng</i> .....	34
2.4.9. <i>Đặc tả ca sử dụng quản lý mã code để đăng ký thành bác sĩ</i> .....	34
2.4.10. <i>Đặc tả ca sử dụng quản lý phòng khám</i> .....	35
<b>2.5. SƠ ĐỒ HOẠT ĐỘNG</b> .....	36
2.5.1. <i>Sơ đồ hoạt động của thành viên</i> .....	36
2.5.2. <i>Sơ đồ hoạt động của quản lý</i> .....	37
2.5.3. <i>Sơ đồ hoạt động của quản trị viên</i> .....	38
<b>2.6. KIẾN TRÚC HỆ THỐNG</b> .....	39
<b>2.7. SƠ ĐỒ LỚP</b> .....	40
<b>2.8. THIẾT KẾ CƠ SỞ DỮ LIỆU</b> .....	41
2.8.1. <i>Mô hình thiết kế cơ sở dữ liệu</i> .....	41
2.8.2. <i>Chi tiết các bảng dữ liệu</i> .....	53
<b>CHƯƠNG 3: TRIỀN KHAI VÀ CHẠY THỬ NGHIỆM</b> .....	54
<b>3.1. MÔI TRƯỜNG TRIỀN KHAI</b> .....	54
3.1.1. <i>Quản lý mã nguồn</i> .....	54
3.1.2. <i>Môi trường cài đặt</i> .....	54
3.2. KẾT QUẢ DEMO VÀ HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH.....	56
<b>KẾT LUẬN</b> .....	Error! Bookmark not defined.
<b>TÀI LIỆU THAM KHẢO</b> .....	76

## DANH SÁCH CÁC BẢNG

Bảng 2.1: Đặc tả ca sử dụng đăng ký.....	30
Bảng 2.2: Đặc tả ca sử dụng đăng nhập .....	31
Bảng 2.3: Đặc tả ca sử dụng quản lý tài khoản .....	31
Bảng 2.4: Đặc tả ca sử dụng đăng ký thành bác sĩ .....	32
Bảng 2.5: Đặc tả ca sử dụng đặt lịch lịch khám bệnh.....	32
Bảng 2.6: Đặc tả ca sử dụng đánh giá bác sĩ .....	33
Bảng 2.7: Đặc tả ca sử dụng bình luận bác sĩ.....	33
Bảng 2.8: Đặc tả ca sử dụng quản lý người dùng .....	34
Bảng 2.9: Đặc tả ca sử dụng quản lý mã code.....	34
Bảng 2.10: Đặc tả ca sử dụng quản lý phòng khám.....	35
Bảng 2.11: Bảng dữ liệu user .....	41
Bảng 2.12: Bảng dữ liệu role .....	42
Bảng 2.13: Bảng dữ liệu user_role .....	42
Bảng 2.14: Bảng dữ liệu degree .....	43
Bảng 2.15: Bảng dữ liệu user_degree .....	43
Bảng 2.16: Bảng dữ liệu faculty .....	44
Bảng 2.17: Bảng dữ liệu user_faculty .....	44
Bảng 2.18: Bảng dữ liệu clinic .....	45
Bảng 2.19: Bảng dữ liệu user_clinic .....	45
Bảng 2.20: Bảng dữ liệu clinic_faculty.....	46
Bảng 2.21: Bảng dữ liệu price .....	46
Bảng 2.22: Bảng dữ liệu booking .....	47
Bảng 2.23: Bảng dữ liệu rate .....	48
Bảng 2.24: Bảng dữ liệu expert_code .....	48
Bảng 2.25: Bảng dữ liệu comment .....	49
Bảng 2.26: Bảng dữ liệu post_type .....	49
Bảng 2.27: Bảng dữ liệu post .....	50
Bảng 2.28: Bảng dữ liệu attachment_type .....	50
Bảng 2.29: Bảng dữ liệu attachment .....	51
Bảng 2.30: Bảng dữ liệu attachment_attachmenttype .....	52
Bảng 2.31: Bảng dữ liệu province .....	51
Bảng 2.32: Bảng dữ liệu distrist .....	52

## DANH SÁCH CÁC HÌNH VẼ

Hình 1.1: Mô hình MVC .....	4
Hình 1.2: Mô hình các thành phần trong spring freamwork .....	6
Hình 1.3: Mô hình cấu trúc json web token .....	9
Hình 1.4: Mô hình hoạt động spring secirity kết hợp jwt .....	9
Hình 1.5: Mô tả RESTful API .....	10
Hình 1.6: Các phương thức HTTP của RESTful API.....	11
Hình 1.7: Bảng tương quan giữa phương thức HTTP, CRUD và các lệnh SQL .....	11
Hình 1.8: Hoạt động của RESTful API.....	12
Hình 1.9: Jordan Walke, người sáng lập ra ReactJS.....	<b>Error! Bookmark not defined.</b>
Hình 1.10: Component trong ReactJS.....	13
Hình 1.11: Sử dụng state trong Component .....	14
Hình 1.12: Sơ đồ vòng đời của component trong ReactJS .....	15
Hình 1.13: Công cụ draw.io .....	18
Hình 1.14: Công cụ Editor VSCode .....	18
Hình 1.15: Công cụ Postman.....	19
Hình 1.16: Công cụ Editor Eclipse .....	19
Hình 2.1: Thống kê lượt khám bệnh 6 tháng đầu năm 2019 tại TP Hồ Chí Minh .....	20
Hình 2.2: Thống kê lượt khám bệnh 9 tháng đầu năm 2019 tại TP Hồ Chí Minh .....	21
Hình 2.3: Sơ đồ ca sử dụng tổng quát của hệ thống .....	25
Hình 2.4: Sơ đồ ca sử dụng đăng ký tài khoản của Khách.....	26
Hình 2.5: Sơ đồ ca sử dụng quản lý tài khoản trong hệ thống .....	26
Hình 2.6: Sơ đồ ca sử dụng bình luận với bác sĩ .....	26
Hình 2.8: Sơ đồ ca sử dụng đặt lịch khám bệnh. ....	27
Hình 2.9: Sơ đồ ca sử dụng đăng ký trở thành bác sĩ .....	27
Hình 2.10: Sơ đồ ca sử dụng nhận email khi đăng ký lịch khám bệnh.....	28
Hình 2.11: Sơ đồ ca sử dụng quản phòng khám.....	28
Hình 2.12: Sơ đồ ca sử dụng nhận email khi có thành viên đặt lịch .....	29
Hình 2.13: Sơ đồ ca sử dụng quản lý người dùng trong hệ thống .....	29
Hình 2.14: Sơ đồ ca sử dụng quản lý mã code để cấp phép tạo bác sĩ.....	29
Hình 2.15: Sơ đồ ca sử dụng quản lý phòng khám.....	30
Hình 2.16: Sơ đồ hoạt động các chức năng của thành viên. ....	36
Hình 2.17: Sơ đồ hoạt động các chức năng của bác sĩ.....	37
Hình 2.18: Sơ đồ hoạt động các chức năng của quản trị viên .....	38
Hình 2.19: Sơ đồ kiến trúc hệ thống .....	39
Hình 2.20: Sơ đồ lớp ứng dụng đặt lịch khám bệnh .....	40
Hình 2.21: Sơ đồ quan hệ cơ sở dữ liệu .....	<b>Error! Bookmark not defined.</b>

Hình 3.1: Lưu trữ mã nguồn trên sourcetree .....	54
Hình 3.2: Node.js .....	<b>Error! Bookmark not defined.</b>
Hình 3.3: Trình quản lý thư viện npm .....	55
Hình 3.4: Ngôn ngữ back-end sử dụng .....	55
Hình 3.5: Ngôn ngữ font-end sử dụng .....	55
Hình 3.6: Hệ quản trị cơ sở dữ liệu.....	56
Hình 3.7: Môi trường heroku deploy bản production.....	56
Hình 3.8 Màn hình chính của hệ thống.....	57
Hình 3.9: Màn hình đăng ký dành cho khách.....	57
Hình 3.10: Màn hình đăng nhập hệ thống.....	58
Hình 3.11: Màn hình chính danh sách các bác sĩ .....	58
Hình 3.12: Màn hình trước khi bình luận.....	<b>Error! Bookmark not defined.</b>
Hình 3.13: Màn hình kết quả sau khi bình luận.....	<b>Error! Bookmark not defined.</b>
Hình 3.14: Màn hình trước khi đánh giá .....	62
Hình 3.15: Màn hình kết quả sau khi đánh giá .....	62
Hình 3.16: Màn hình chính trước khi đặt lịch .....	63
Hình 3.17: Màn hình form đăng ký khi đặt lịch .....	63
Hình 3.18: Màn hình chính sau khi đã được đặc lịch .....	64
Hình 3.19: Màn hình thông báo đến bác sĩ khi có người đặt lịch.....	64
Hình 3.20: Màn hình thông báo đến người đã đặt lịch .....	64
Hình 3.21: Màn hình chính trước khi tạo lịch .....	67
Hình 3.22: Màn hình form khởi tạo lịch .....	67
Hình 3.23: Màn hình kết quả sau khi đặt lịch.....	68
Hình 3.24: Màn hình trước khi khởi tạo bài viết .....	68
Hình 3.25: Màn form để khởi tạo bài viết.....	69
Hình 3.26: Màn hình kết quả sau khi tạo thành công bài viết .....	69
Hình 3.27: Màn hình form tạo danh sách giá .....	70
Hình 3.28: Màn hình danh sách giá của phòng khám.....	70
Hình 3.29: Màn hình trước khi thêm bác sĩ vào phòng khám .....	71
Hình 3.30: Màn hình kết quả sau khi thêm bác sĩ vào phòng khám.....	71

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Tù	Viết tắt của	Điễn giải
API	Application Programming Interface	Giao diện lập trình ứng dụng
RESTful	REpresentational State Transfer	Một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
MVC	Model-View-Controller	Mô hình - Giao diện - Bộ điều khiển
DB	Database	Cơ sở dữ liệu
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
CSS	Cascading Style Sheets	Điều khiển cách hiển thị nội dung trang web
UI	User Interface	Giao diện người dùng
DOM	Document Object Model	Mô hình đối tượng tài liệu
JSX	JavaScript XML	Một loại cú pháp mở rộng dành cho ngôn ngữ JavaScript viết theo kiểu XML
SQL	Structured Query Language	Ngôn ngữ truy vấn cấu trúc

**Framework:** được định nghĩa là một bộ khung hay giàn giáo cung cấp các chức năng, giải pháp được cài đặt sẵn giúp tiết kiệm thời gian trong quá trình phát triển ứng dụng. Bạn cũng có thể hiểu Framework tương tự như khung của một ngôi nhà nó cung cấp một nền tảng chắc chắn và cơ bản để bạn có thể xây dựng được đa số các kiểu nhà được thiết kế trong bản vẽ. Mặc dù bạn có thể xây hoàn chỉnh một ngôi nhà từ đầu, nhưng bắt đầu với một dàn khung sẽ giúp bạn tiết kiệm được tối đa công sức cho việc bắt đầu. Framework trong lập trình ứng dụng cũng vậy, nó cung cấp đa số các kiểu mẫu thiết kế phù hợp với ứng dụng bạn sắp thực hiện, các thư viện, API, trình biên dịch và hơn thế nữa để hợp lý hóa việc tạo ra phần mềm mới. Biết được framework nào để sử dụng cho dự án nào là một kỹ năng quan trọng đối với bất kỳ nhà phát triển nào.

## GIỚI THIỆU ĐỀ TÀI

### 1. Lý do chọn đề tài

Ngày nay, với cuộc công nghiệp 4.0 như hiện nay. Con người đang tập trung vào công việc phát triển kinh tế và quỹ thời gian giành cho bản thân của họ có quá ít đặt biệt là những người lao động, công nhân, nhân viên công chức... Để giải quyết bài toán tiết kiệm thời gian cũng như chủ động thời gian cho việc khám chữa bệnh tại các bệnh viện hoặc các trung tâm khám bệnh dành cho phòng khám tư nhân. Chính vì vấn đề nêu trên cần có một ứng dụng giúp các bệnh nhân có thể đặt lịch phù hợp với thời gian cá nhân. Cũng như giúp các bác sĩ có thể tạo thêm thu nhập trong quỹ thời gian ngoài công sở. Ứng dụng giúp các bệnh nhân không phải mất quá nhiều thời gian cho việc chờ đợi để khám bệnh và chủ động về mặc thời gian cho bệnh nhân.

Trong thực tế, qua các buổi khám bệnh tại các bệnh viện và phòng khám tư nhân em quan sát và thấy vấn đề chờ đợi để được khám bệnh mất quá nhiều thời gian và công sức. Để được khám bệnh mọi người phải bỏ ra thời gian trung bình khoảng 3 đến 4 tiếng để được khám bệnh tại các bệnh viện và 1 đến 2 tiếng để khám bệnh tại các phòng khám tư nhân. Qua đó, nhu cầu giải quyết vấn đề trên là có khả thi và thực tế.

Cùng với đó, trong bối cảnh của sự bùng nổ công nghệ thông tin nói chung và mạng internet nói riêng. Con người đang cần những ứng dụng thực tế để giúp rút ngắn thời gian và thoải mái trong quá trình làm việc và sinh hoạt cá nhân. Ứng dụng đặt chỗ khám bệnh cũng là một trong những ứng dụng được ra đời để giải quyết nhu cầu trên.

Vì những lí do trên, em đã quyết định chọn đề tài “**Xây dựng ứng dụng đặt chỗ khám bệnh**” làm đề tài tốt nghiệp của mình.

### 2. Mục đích của đề tài

“**Xây dựng ứng dụng đặt chỗ khám bệnh**” với mục đích xây dựng một ứng dụng đơn giản, thân thiện, dễ sử dụng chạy trên nền tảng web, tạo ra một ứng dụng có thể hỗ trợ cho các bác sĩ có thể chủ động trong thời gian khám bệnh và đồng thời cũng giúp các bệnh nhân chủ động thời gian đặt lịch khám bệnh. Ứng dụng cho phép người dùng đăng ký làm thành viên, khi là thành viên bạn có thể đăng ký lịch khám bệnh, bình luận và cho đánh giá về bác sĩ mình thích hoặc đã đặt lịch. Ngoài ra thành viên cũng có thể đăng ký trở thành bác sỹ nếu có sự cho phép của người quản trị ứng dụng, qua đó bác sĩ có thể đăng ký phòng khám và tạo lịch khám.

### 3. Nội dung của đề tài

Những nội dung công việc cần thực hiện để hoàn thành đề tài bao gồm:

- Phân tích bài toán và thiết kế hệ thống thông tin.
- Tìm hiểu và phân tích UML.
- Thiết kế cơ sở dữ liệu cho hệ thống.
- Tìm hiểu và nghiên cứu, áp dụng các ngôn ngữ, công cụ lập trình JAVA và Javascript, framework Spring với hệ sinh thái : Spring boot và Spring security và ReactJS, cách thức xây dựng WebAPI theo cơ chế RESTful.
- Nghiên cứu và ứng dụng MySQL làm nền tảng cơ sở dữ liệu.
- Xây dựng ứng dụng với các chức năng chính.
- Viết báo cáo.

### 4. Phạm vi sử dụng đề tài

Ứng dụng được sử dụng tại các phòng khám tư nhân hoặc các bệnh viện với vai trò là bác sĩ và tất cả mọi thành viên có thể đăng ký thành viên để đặt lịch khám bệnh với các bác sĩ mà mình tin tưởng, yêu thích.

### 5. Bố cục của báo cáo

Đồ án được chia làm 3 chương chính:

Với chương thứ nhất đi vào tìm hiểu và nghiên cứu cơ sở lý thuyết để thực hiện áp dụng vào đề tài. Tìm hiểu công cụ và ngôn ngữ lập trình.

Trong chương thứ hai đi vào quá trình khảo sát hiện trạng thực tế, phân tích và thiết kế hệ thống thông tin, xác định các tác nhân chính của hệ thống, thiết kế và xây dựng cơ sở dữ liệu.

Tiếp nối chương thứ hai, chương thứ ba sẽ đi vào quá trình triển khai hệ thống, thực hiện xây dựng mã nguồn chương trình và chạy thử nghiệm.

Đề tài được đánh giá lại qua phần kết luận, đưa ra các đánh giá kết quả về những vấn đề đã giải quyết, những vấn đề chưa giải quyết được và các giải pháp cho đề cải tiến, khắc phục những vấn đề đó.

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

### 1.1. KIẾN THỨC CƠ BẢN

#### 1.1.1. *Html*

Là viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ đánh dấu siêu văn bản", được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

#### 1.1.2. *Css*

Là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,... thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,...rất nhiều.

#### 1.1.2. *Javascript*

Là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

### 1.2. TỔNG QUAN VỀ JAVA VÀ SPRING FRAMEWORK

JAVA là ngôn ngữ lập trình hướng đối tượng, do vậy không thể dùng Java để viết một chương trình hướng chức năng. JAVA có thể giải quyết hầu hết các công việc mà các ngôn ngữ khác có thể làm được.

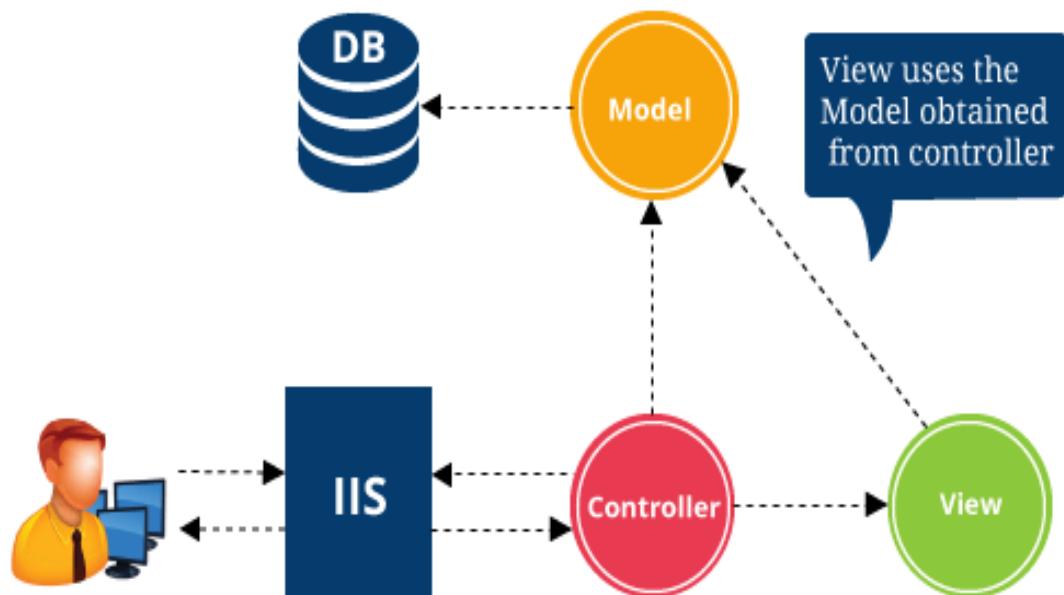
JAVA là ngôn ngữ vừa thông dịch vừa biên dịch. Đầu tiên mã nguồn được biên dịch bằng công cụ JAVAC để chuyển thành dạng ByteCode. Sau đó được thực thi trên

từng loại máy cũ thèn chờ chương trình thông dịch JAVA. Mục tiêu của các nhà thiết kế JAVA là cho phép người lập trình viết chương trình một lần nhưng có thể chạy trên bất cứ phần cứng cụ thể, thế nên khẩu hiệu của nhà thiết kế là “ Write One, Run Any Where”.

JAVA là ngôn ngữ lập trình mã nguồn mở phía server được thiết kế để dễ dàng xây dựng các trang web và ứng dụng di động. JAVA cho phép xây dựng ứng dụng web trên mạng internet tương tác với mọi cơ sở dữ liệu như: MySQL, Oracle,... Ngôn ngữ lập trình JAVA được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C, C++, C#. Nó là ngôn ngữ lập trình phổ biến nhất thế giới và được rất nhiều các công ty hàng đầu thế giới ứng dụng vào dự án của họ.

### ❖ Mô hình MVC trong JAVA

Mô hình MVC là viết tắt của Model – View – Controller. Là một khái niệm quen thuộc trong thiết kế web và có thể được xem như là một tiêu chuẩn quan trọng không thể thiếu đối với các nhà lập trình website khi thiết lập nên những sản phẩm chất lượng, giải quyết các công việc một cách nhanh gọn và dễ dàng hơn. Nói cho dễ hiểu, nó là mô hình phân bổ source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.



Hình 1.1: Mô hình MVC

– Thành phần:

- + **Controller:** Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model.
- + **Model:** Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý...
- + **View:** Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images... Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

### ❖ Spring Framework

Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson. Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Với sự phát triển và có một cộng đồng lớn, spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.

Để ngăn chặn sự phức tạp trong phát triển các ứng dụng, Spring Framework thường dựa trên các quan điểm như sau:

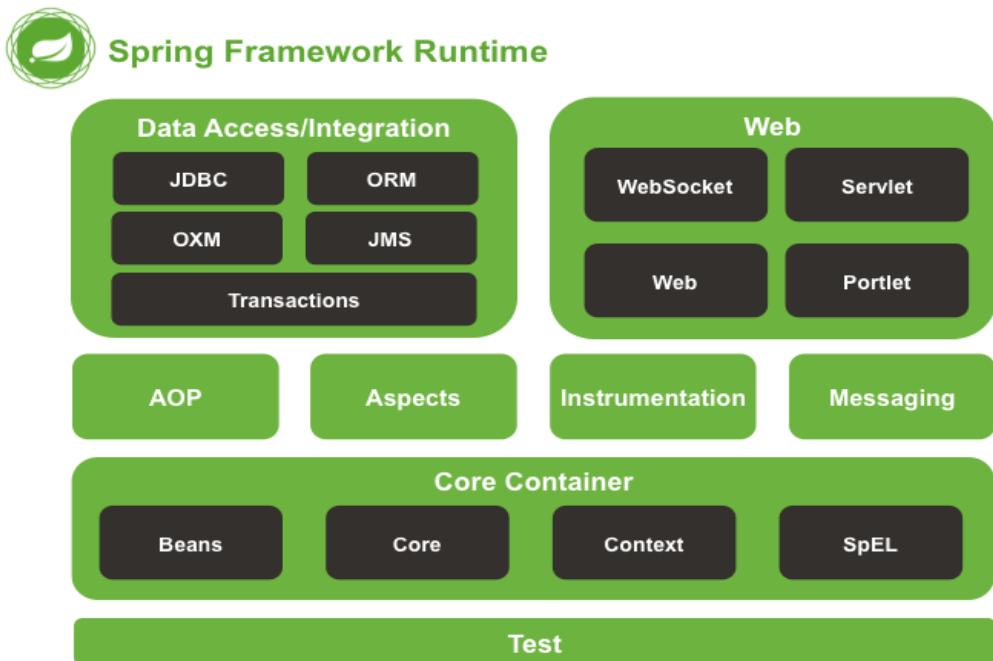
- Đơn giản hóa công việc phát triển thông qua việc sử dụng các POJO “Plain Old Java Object”.
  - Nới lỏng ràng buộc giữa các thành phần thông qua việc sử dụng Dependency Injection.
- Kiến trúc, các thành phần của Spring Framework

Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng một trong các module đó.

Nhiều tính năng được tổ chức thành khoảng 20 module. Các module này được phân nhóm thành Core Container, Data Access/Integration, Web, AOP (Aspect Oriented Programming), Instrumentation, Messaging, và Test.

- + **Spring Core:** là thành phần cốt lõi của Spring Framework. Đây chính là nền tảng để xây dựng nên các thành phần khác trong hệ sinh thái của Spring

Framework. Lược đồ dưới đây mô tả mối quan hệ giữa Spring Core với các thành phần khác trong Spring Framework.



Hình 1.2: Mô hình các thành phần trong spring freamwork

- + **Spring Bean**: là trung tâm của Spring Core và là trái tim của một ứng dụng Spring. Trái ngược với EJB, Spring Framework được thiết kế từ lõi bằng cách sử dụng các POJO hay các Spring Bean. Spring Bean có thể được hiểu là các đối tượng Java đơn giản.
- + **Spring Context**: mang mọi thứ lại với nhau. Spring Context kế thừa các tính năng của Spring Bean và bổ sung các hỗ trợ cho internationalization (ví dụ như các resource bundle), event propagation, resource loading... Ngoài ra, Spring Context cũng hỗ trợ các tính năng của Java EE như EJB, JMX và RMI. Interface ApplicationContext chính là trọng tâm của Spring Context.
- + **Data Access/Integration**: bao gồm các module JDBC, ORM,... và Transaction.
  - **spring-jdbc**: cung cấp một lớp JDBC-abstraction để loại bỏ những code tẻ nhạt cả JDBC và phân tích những mã lỗi cụ thể của database.
  - **spring orm**: cung cấp lớp tích hợp với các orm API phổ biến như JPA, JDO và Hibernate.

- **spring-transaction:** hỗ trợ quản lý giao dịch theo chương trình và khai báo cho các lớp mà thực hiện các giao diện đặc biệt và cho tất cả các POJO của bạn.
- + **Web:** bao gồm các module spring-web module spring-web, spring-webmvc, spring-websocket và springwebmvc-portlet.
  - **spring-webmvc:** bao gồm việc triển khai Model-View-Controller (MVC) của Spring cho các ứng dụng web.
  - **spring-websocket:** cung cấp hỗ trợ cho WebSocket-based, giao tiếp hai chiều giữa client và server trong các ứng dụng web.
  - **spring-web:** cung cấp tính năng tích hợp web theo định hướng cơ bản như chức năng tải lên nhiều phần dữ liệu và khởi tạo các container IoC sử dụng servlet và một bối cảnh ứng dụng web theo định hướng.
  - **springwebmvc-portlet:** cung cấp việc triển khai MVC được sử dụng trong môi trường portlet và ánh xạ chức năng của module Web-Servlet.
- + **Test:** hỗ trợ việc kiểm tra các thành phần mùa xuân với JUnit hoặc TestNG khuôn khổ.

## ❖ Spring Boot

Spring boot là một dự án nổi bật trong hệ sinh thái Spring Framework, là cách nhanh nhất để tạo ra một REST service độc lập. Spring boot đơn giản hóa việc cấu hình, cụ thể ở đây là Spring Boot tự cấu hình tất cả bằng cách cung cấp các hành vi cụ thể. Nó đơn giản hóa việc deploy, bằng cách đóng gói app của chúng ta thành gói jar, để có thể dễ dàng tích hợp vào các web container. Nếu như trước đây, công đoạn khởi tạo một dự án Spring khá vất vả từ việc khai báo các dependency trong file pom.xml cho đến cấu hình bằng XML hoặc các annotation phức tạp, thì giờ đây với Spring Boot chúng ta có thể tạo ứng dụng Spring một cách nhanh chóng và cấu hình cũng đơn giản hơn.

Các lợi ích của Spring Boot:

- Dễ dàng để phát triển các ứng dụng dựa trên Spring với Java.
- Giảm thiểu thời gian phát triển và tăng năng xuất.

- Tránh việc phải viết nhiều mã nguyên mẫu (boilerplate Code), Annotations và các cấu hình XML.
- Dễ dàng để bạn tương tác các ứng dụng Spring Boot với các hệ sinh thái của Spring như Spring JDBC, Spring ORM, Spring Data, Spring Security.
- Đi theo cách tiếp cận “Nguyên tắc cấu hình mặc định” để giảm thiểu thời gian và nỗ lực cần thiết để phát triển ứng dụng.
- Cung cấp các Server nhúng như là Tomcat để phát triển và test các ứng dụng web nhanh chóng và dễ dàng.
- Cung cấp rất nhiều các plugin để phát triển và test các ứng dụng Spring Boot nhanh chóng sử dụng các công cụ Build như Maven và Gradle.
- Cung cấp nhiều plugin để làm việc với các cơ sở dữ liệu nhúng và các cơ sở dữ liệu lưu trữ trên bộ nhớ một cách dễ dàng.

### ❖ Spring Security

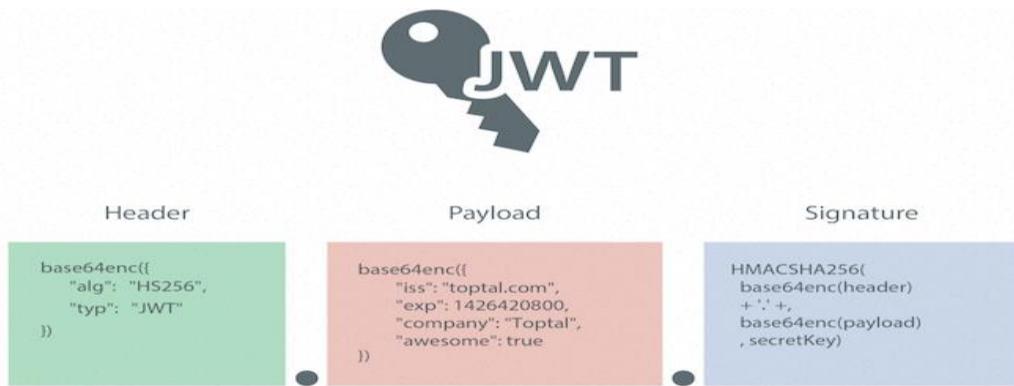
Spring Security là một framework tập trung vào việc cung cấp khả năng xác thực và phân quyền cho ứng dụng Java, một dự án nổi bật trong hệ sinh thái Spring. Spring Security cung cấp các dịch vụ bảo mật toàn diện cho các ứng dụng doanh nghiệp có nền tảng Java EE.

Spring Security cung cấp 2 cơ chế cơ bản:

- **Xác thực** ( Authentication ): là tiến trình thiết lập một principal. Principal có thể hiểu là một người, hoặc một thiết bị, hoặc một hệ thống nào đó có thể thực hiện một hành động trong ứng dụng của bạn.
- **Phân quyền** ( Authorization hay Access-control ): là tiến trình quyết định xem một principal có được phép thực hiện một hành động trong ứng dụng của bạn hay không. Trước khi diễn tiến tới Authorization, principal cần phải được thiết lập bởi Authentication.

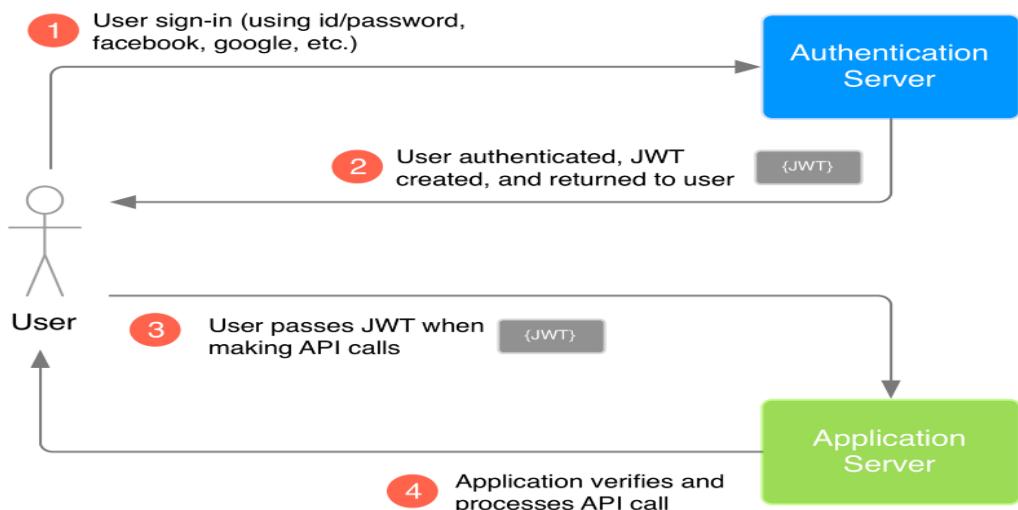
Cơ chế hoạt động của Spring Security kết hợp Json Web Token

- + **Json Web Token (JWT)** : là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server , các thông tin trong chuỗi JWT được định dạng bằng Json . Trong đó chuỗi Token phải có 3 phần là header , phần payload và phần signature được ngăn bằng dấu “.”



Hình 1.3: Mô hình cấu trúc json web token

+ Cơ chế hoạt động :



Hình 1.4: Mô hình hoạt động spring security kết hợp jwt

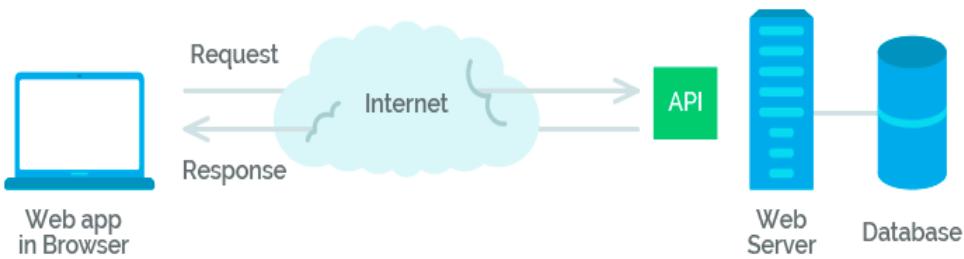
- Nhìn vào sơ đồ, ta có thể thấy luồng đi như sau :

- User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server.
- Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
- Người dùng nhận được JWT do Authentication Server vừa mới trả về làm "chìa khóa" để thực hiện các "lệnh" tiếp theo đối với Application Server.
- Application Server trước khi thực hiện lệnh được gọi từ phía User, sẽ verify JWT gửi lên. Nếu OK, tiếp tục thực hiện lệnh được gọi.

### 1.3. TỔNG QUAN RESTFULL API

#### ❖ Định nghĩa:

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



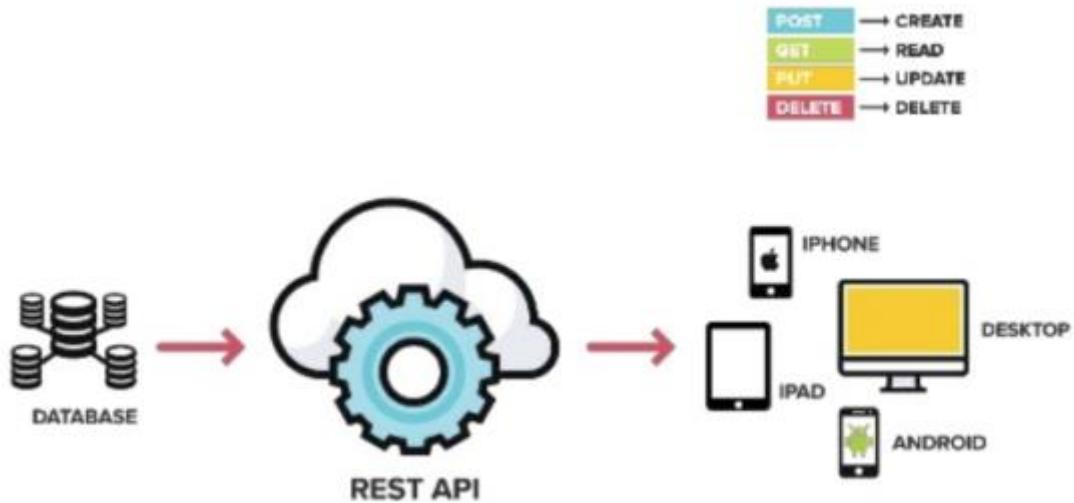
Hình 1.5: Mô tả RESTful API

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

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

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

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



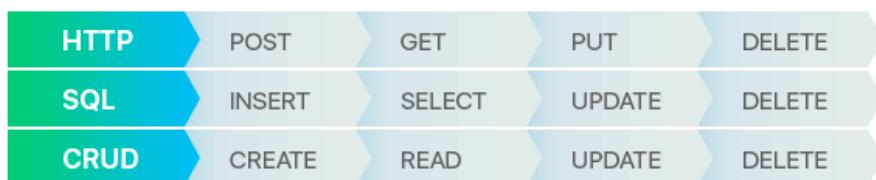
Hình 1.6: Các phương thức HTTP của RESTful API

#### ❖ Hoạt động

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

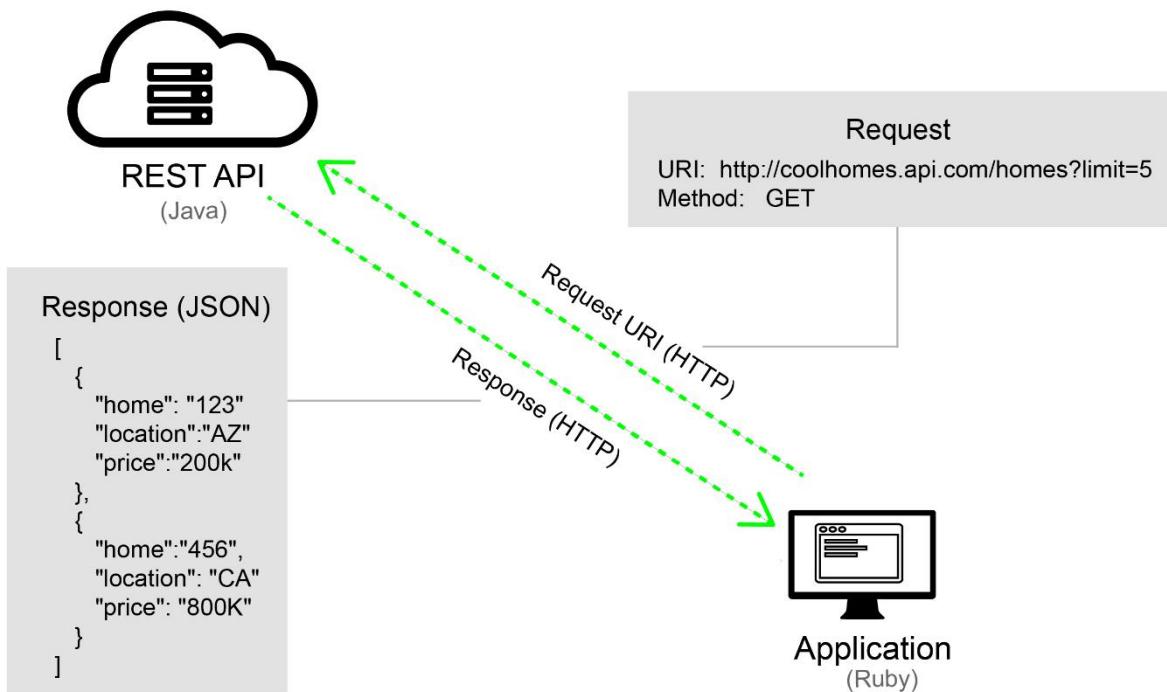
- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xoá một Resource.

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



Hình 1.7: Bảng tương quan giữa phương thức HTTP, CRUD và các lệnh SQL.

## REST API model



Hình 1.8: Hoạt động của RESTful API

Sau quá trình gọi API việc trả về những mã trạng thái cụ thể khi nhận và phiên dịch một HTTP Request, Server sẽ gửi tín hiệu phản hồi là một HTTP Response là một trong số các loại trạng thái sau:

- 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
- 201 Created – Trả về khi một Resouce vừa được tạo thành công.
- 204 No Content – Trả về khi Resource xoá thành công.
- 304 Not Modified – Client có thể sử dụng dữ liệu cache.
- 400 Bad Request – Request không hợp lệ
- 401 Unauthorized – Request cần có auth.
- 403 Forbidden – bị từ chối không cho phép.
- 404 Not Found – Không tìm thấy resource từ URI
- 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
- 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
- 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
- 422 Unprocessable Entity – Dữ liệu không được xác thực
- 429 Too Many Requests – Request bị từ chối do bị giới hạn

## 1.4. TỔNG QUAN VỀ REACTJS

React (Hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi *Jordan Walker*, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013.

### – Đặc tính của ReactJS:

- + Tu tướng của ReactJS là xây dựng lên các **components** có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp chúng ta dễ dàng quản lý, mở rộng hệ thống, điều này nếu là Angular JS thì đòi hỏi cấu trúc, cách viết code phải thật tối ưu.
- + ReactJS luôn giữ các components ở trạng thái stateless (nhiều nhất có thể) khiến ta dễ dàng quản lí bởi nó chẳng khác gì một trang static HTML. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiển thị ra dựa vào các đầu vào đó, điều này giải thích tại sao nó mang tính tái sử dụng và dễ dàng cho việc test.

### ➤ Các đặc trưng của ReactJS:

#### - Component:

- + Là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng.
- + Một ứng dụng React có thể sử dụng hàng chục, hoặc hàng trăm các component liên kết và tương tác với nhau.
- + Ta có thể tạo ra một component bằng cách extends từ Component như sau:

```
1 import React, { Component } from 'react'
2
3 export default class Component extends Component {
4     render() {
5         return [
6             <div>
7                 |     <h1>Welcome to ReactJs ! My Name is Trinh</h1>
8             </div>
9         ]
10    }
11 }
12 }
```

Hình 1.10: Component trong ReactJS.

- + Phương thức quan trọng nhất là render(), phương thức này được trigger khi component đã sẵn sàng để được render lên trên trang. Trong hàm này, sẽ trả về mô tả cho việc bạn muốn React render cái gì lên trên trang. Hàm render chính là mô tả cụ thể của giao diện người dùng tại bất cứ thời điểm nào. Vì thế nếu dữ liệu thay đổi, React sẽ quan tâm đến việc cập nhật lại giao diện với dữ liệu tương ứng.
- **Props:**
  - + Thông tin được truyền từ component này sang component khác được gọi là props.
  - + Props của một component có thể là một object, function, string, number .... Nó chứa thông tin về thành phần đó.

- **State:**

- + Một component trong React có 2 cách để lấy thông tin props và state. Không như props, state của 1 component không được truyền từ bên ngoài vào. Một component sẽ tự quyết định state của chính nó. Để tạo ra state cho component, ta sẽ phải khai báo nó trong hàm constructor.

```

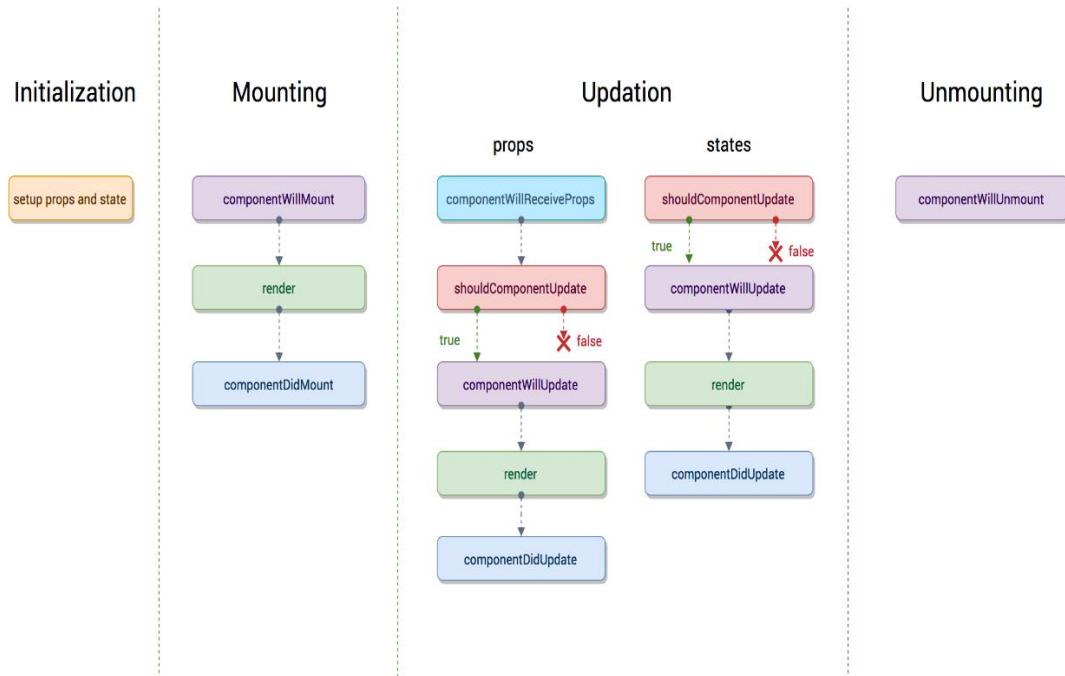
1 import React, { Component } from 'react'
2
3 export default class Component extends Component {
4   constructor(props){
5     super(props);
6     this.state = {
7       name : "Vo Van Trinh"
8     }
9   }
10  render() {
11    return (
12      <div>
13        <h1>Welcome to ReactJs ! My Name is {this.state.name}</h1>
14      </div>
15    )
16  }
17}

```

Hình 1.11: Sử dụng state trong Component

- + This.state ở đây là một đối tượng. Để truy cập đến các state ta sử dụng cú pháp: this.state.name\_of\_property. Và cách phổ biến để thay đổi state của một component là dùng hàm setState().

- **Lifecycle:**



Hình 1.12: Sơ đồ vòng đời của component trong ReactJS

- + Một thành phần đặc trưng không kém Props và State của React đó là LifeCycle. LifeCycle là những phương thức được viết sẵn truy cập đến từng thời điểm trong một vòng đời của component, từ đó chúng ta có thể triển khai ý tưởng một cách dễ dàng mà linh động.
- + **constructor(props):**  
Hàm này là hàm khởi tạo, là hàm chạy ngay sau khi component được tạo ra, đầu vào của hàm này là props (tức là thuộc tính của component được dùng để truyền dữ liệu từ bên ngoài vào). Trong hàm này chúng ta cũng có thể khởi tạo state cho component.
- + **componentWillMount:**  
Trước khi render thì sẽ gọi đến hàm này. Ở đây chúng ta có thể đăng ký các biến toàn cục và có thể dựa vào props tính toán và setState nếu có.
- + **render**  
Là hàm bắt buộc, vì đây là đầu ra của component. Nó chịu trách nhiệm trả về một đối tượng JSX những gì chúng ta muốn hiển thị ra bên ngoài.

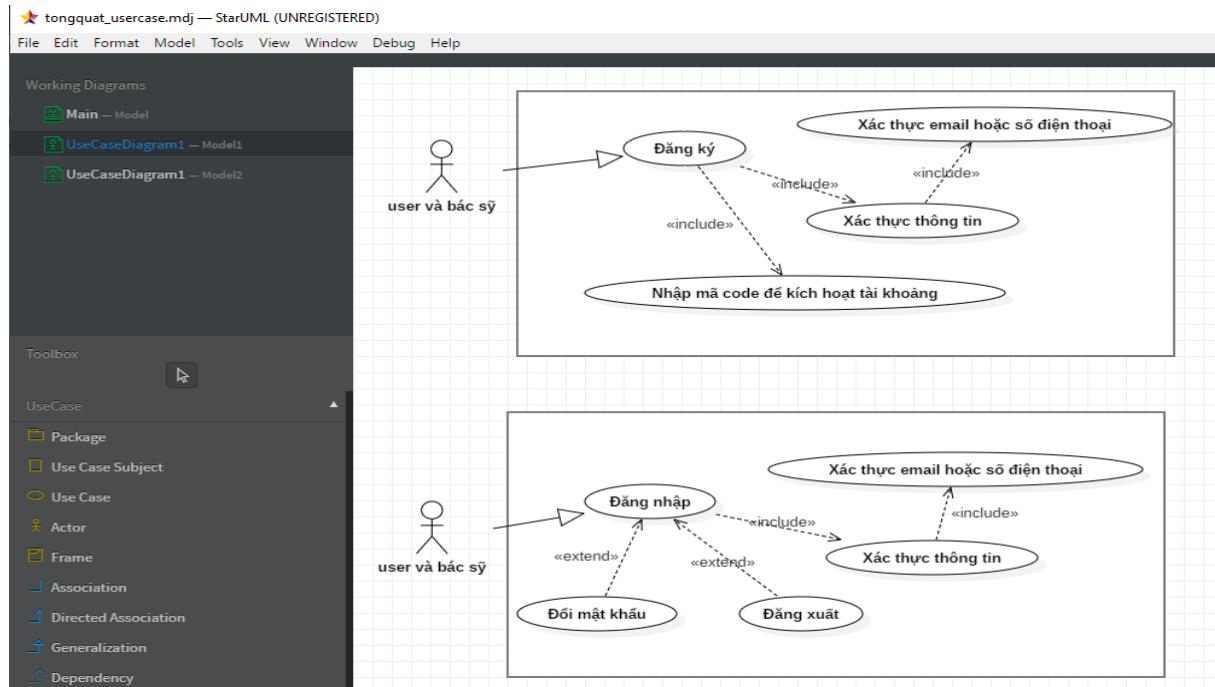
- + **componentDidMount:** Là hàm được gọi ngay sau khi giao diện được render xong, có nghĩa là khi giao diện hiện lên sẽ gọi hàm này. Trong hàm này chúng ta có thể dùng để lấy dữ liệu từ api, server... và setState dữ liệu ra.
  - + **componentWillReceiveProps(nextProps):**  
Hàm này được gọi đến mỗi khi props thay đổi. Do vậy được sử dụng để thay đổi trạng thái của các component phụ thuộc vào props và sử dụng các kết quả, khởi tạo các biến có tính chất bất đồng bộ.
  - + **shouldComponentUpdate(nextProps, nextState):**  
Hàm này được gọi khi state và props thay đổi và sẽ trả về kết quả true/false, chúng ta sử dụng hàm này để xử lý xem có cần update component hay không. Mặc định hàm này sẽ trả về true.
  - + **componentWillUpdate(nextProps, nextState):**  
Hàm này được gọi dựa vào kết quả của hàm shouldComponentUpdate, nếu trả về false thì hàm này sẽ không được gọi.
  - + **componentDidUpdate(prevProps, prevState):**  
Hàm này được gọi ngay sau khi component được render lại.
  - + **componentWillUnmount:**  
Hàm này được gọi khi component loại bỏ DOM, thực hiện các thao tác dọn dẹp, loại bỏ những phần tử dư thừa và hủy các timer...
- **Ưu điểm và nhược điểm của ReactJS:**
- **Ưu điểm:**
    - + ReactJs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật len DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
    - + ReactJs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
    - + Reactjs có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ

dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường

- + Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về
- + Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- + Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi
- **Nhược điểm:**
  - + ReactJs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
  - + Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
  - + React khá nặng nề so với các framework khác, React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.

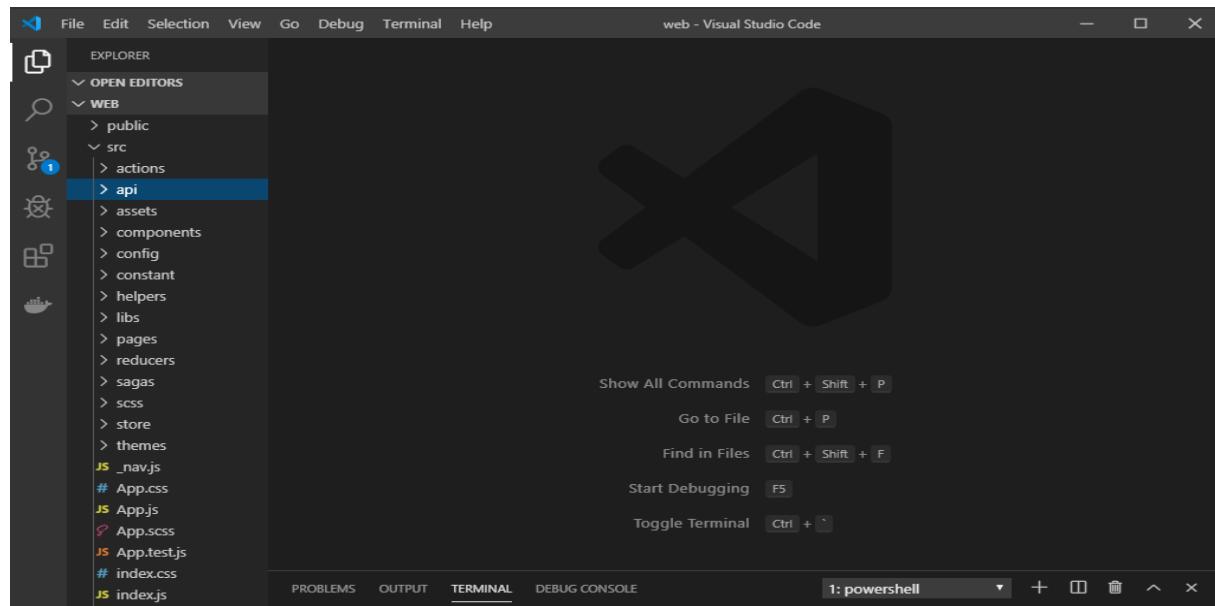
## 1.5. CÔNG CỤ SỬ DỤNG

- Quá trình phân tích và thiết kế UML thì công cụ **starUML**. Công cụ này giúp chúng ta có thể dễ dàng thao tác và sử dụng để tạo các sơ đồ ca sử dụng, sơ đồ hoạt động,....



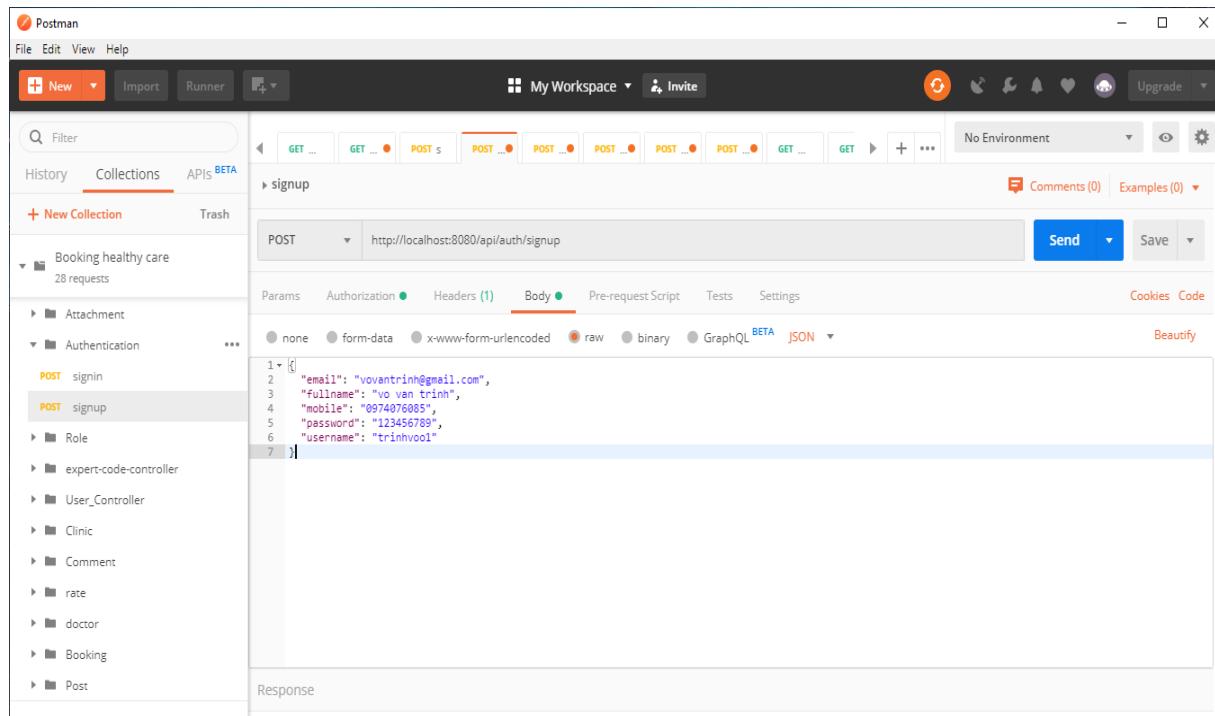
Hình 1.13: Công cụ draw.io

- Quá trình xây dựng mã chương trình, công cụ, editor được dùng là VSCode, đặc điểm nổi bật là đơn giản, gọn nhẹ, dễ dàng cài đặt, hỗ trợ nhiều hệ điều hành cũng như nhiều ngôn ngữ lập trình khác nhau.



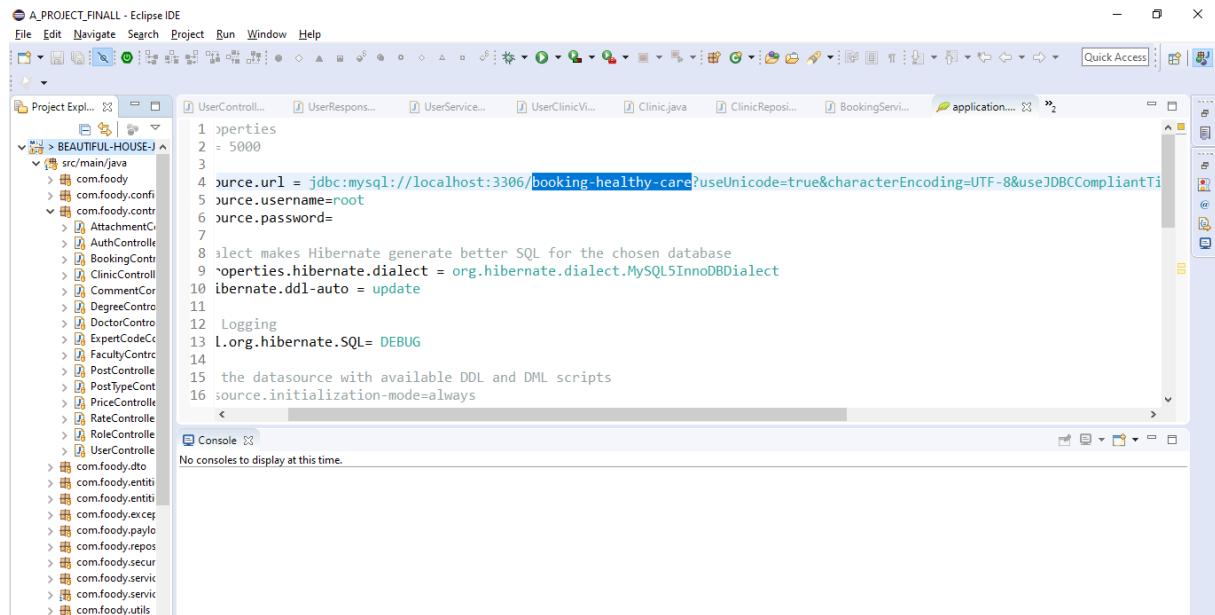
Hình 1.14: Công cụ Editor VSCode

- Quá trình kiểm tra, test và chạy thử các API đã được viết chương trình trước khi áp dụng để gọi từ các request ở website, công cụ được sử dụng là Postman.



Hình 1.15: Công cụ Postman

- Quá trình xây dựng mã nguồn Java ta sử dụng trình soạn thảo IDE eclipse. Eclipse với sự tiện lợi, có nhiều chức năng hỗ trợ nhiều cho việc xây dựng ứng dụng.



Hình 1.16: Công cụ Editor Eclipse

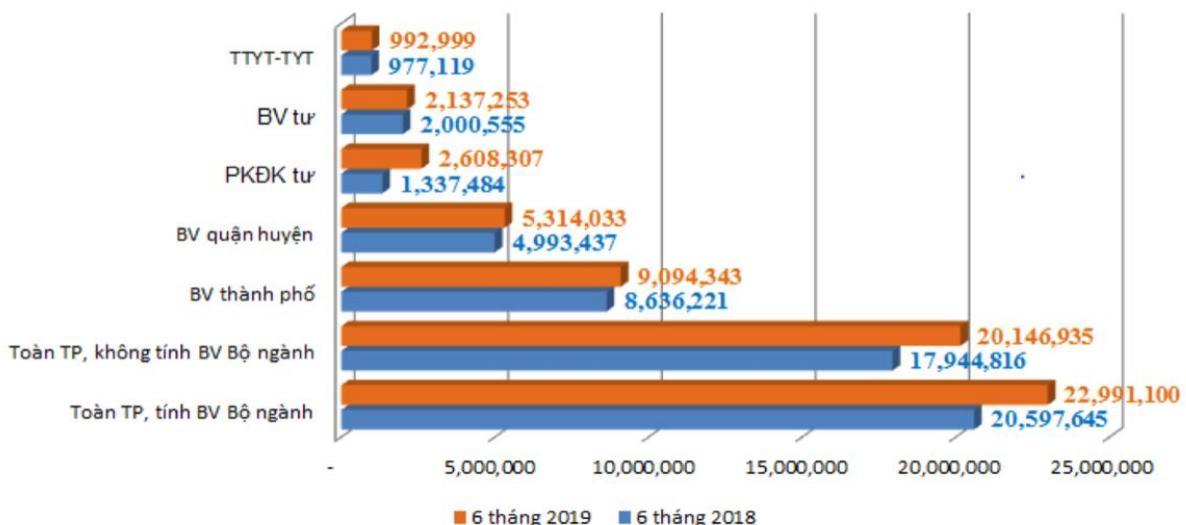
## CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. KHẢO SÁT BÀI TOÁN THỰC TẾ

Hiện nay số lượng lượt khám bệnh ngày càng tăng cao tại các phòng khám tư nhân và bệnh viện. Để làm rõ hơn nhu cầu khám chữa bệnh của người dân. Chúng ta có những số liệu chính xác về 6 tháng và 9 tháng đầu năm 2019 trên địa bàn thành phố Hồ Chí Minh như sau :

#### ➢ **Tình hình khám bệnh 6 tháng đầu năm 2019 trên địa bàn TP Hồ Chí Minh**

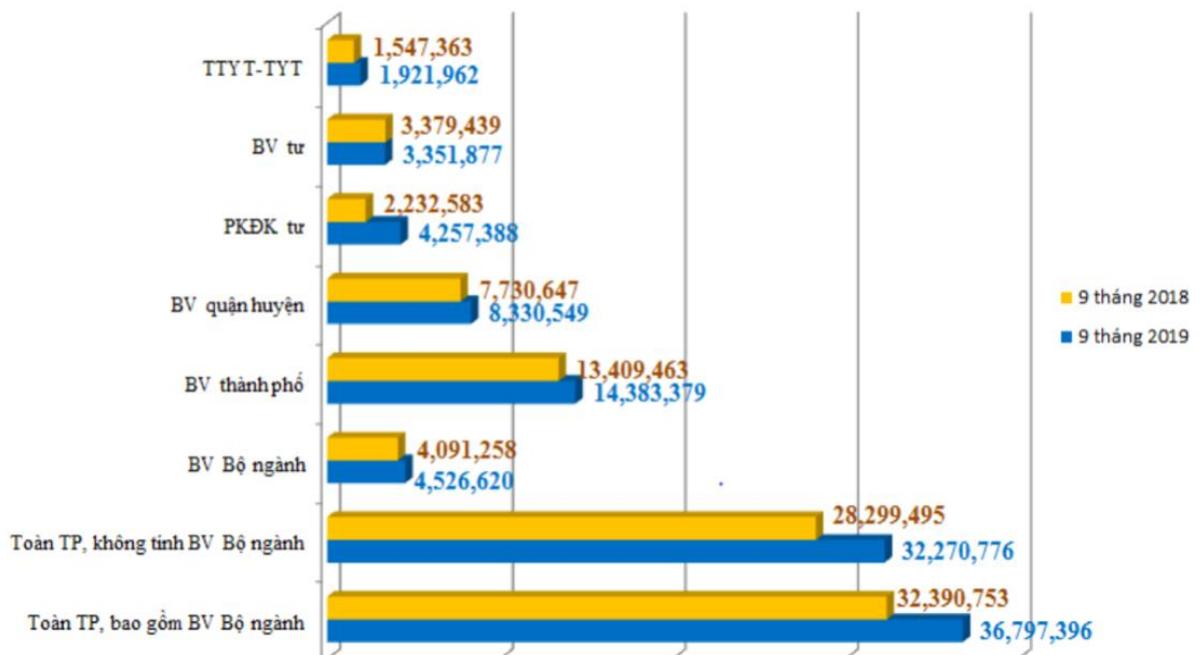
Trên toàn địa bàn thành phố tổng cộng có 22.991.100 lượt, tăng 11,6% so với cùng kỳ. Trong đó, số lượt khám tại các bệnh viện Bộ, Ngành tăng 11,6%, %, bệnh viện thành phố tăng 7,2%, bệnh viện quận, huyện tăng 6,4%, bệnh viện tư tăng 6,8%. Riêng phòng khám đa khoa tư nhân tăng đáng kể xấp xỉ 50%, số lượt khám tại phòng khám của trung tâm y tế và trạm y tế tăng 1,6%.



Hình 2.1: Thống kê lượt khám bệnh 6 tháng đầu năm 2019 tại TP Hồ Chí Minh

#### ➢ **Tình hình khám bệnh 9 tháng đầu năm 2019 trên địa bàn TP Hồ Chí Minh**

Phân tích số liệu khám, chữa bệnh qua 9 tháng cho thấy có những tín hiệu mới, khác với nhiều năm trước đây số lượt khám tiếp tục tăng. Với hơn 36 triệu lượt khám ngoại trú tại các bệnh viện và các cơ sở y tế trên địa bàn thành phố trong 9 tháng đầu năm 2019 cho thấy số lượt khám có xu hướng tăng qua các năm, cùng kỳ trú tại các bệnh viện và các cơ sở y tế trên địa bàn thành phố trong 9 tháng đầu năm 2019 cho thấy số lượt khám có xu hướng tăng qua các năm, cùng kỳ năm 2018 là hơn 32 triệu lượt vẫn chưa có điểm dừng. Tổng số lượt khám tăng 13,6%, trong đó các BV bộ/ngành trên địa bàn thành phố tăng 10% (chưa tính số liệu của BV Giao thông vận tải), BV thành phố tăng 7,1%, BV quận/huyện tăng 7,7%, BV tư giảm nhẹ nhưng PKDK tăng đáng kể (tăng 90%), TYT và phòng khám của TTYT quận/huyện tăng (24%).



Hình 2.2: Thống kê lượt khám bệnh 9 tháng đầu năm 2019 tại TP Hồ Chí Minh

Qua các số liệu thống kê trên thì nhu cầu để có một ứng dụng đặt lịch khám bệnh cho bệnh nhân là khả thi và phù hợp với nhu cầu thực tế.

## 2.2. PHÂN TÍCH CHỨC NĂNG

**Ứng dụng đặt chỗ khám bệnh** cung cấp cho tất cả các thành viên, các bệnh nhân có thể dễ dàng đặt lịch khám bệnh tại các phòng khám tư nhân hoặc bệnh viện. Đồng thời có thể giúp các bệnh nhân cũng như các bác sĩ chủ động trong việc đặt lịch để khám bệnh. Các bệnh nhân có thể tiết kiệm thời gian khám bệnh tại các cơ sở y tế. Ngoài ra ứng dụng còn cung cấp một số chức năng như đánh giá bác sĩ qua số lượng sao, bình luận, tìm kiếm theo vùng miền và theo chuyên khoa.

Hệ thống phục vụ cho 4 loại người dùng: Người quản trị, Bác sĩ, Thành viên và Khách. Các chức năng theo từng nhóm người dùng sẽ được mô tả dưới đây.

### 2.2.1. Khách có các chức năng sau

#### – Chức năng đăng ký

Chức năng này cho phép khách có thể đăng ký tài khoản vào ứng dụng để sử dụng, và được chia làm hai trường hợp:

- + Trường hợp thứ nhất là đăng ký tài khoản để trở thành thành viên và đặt lịch.
- + Trường hợp thứ hai là đăng ký thành viên và mong muốn tài khoản của mình là bác sĩ thì liên hệ người quản trị hệ thống để được đăng ký.

– **Chức năng gửi liên hệ, yêu cầu**

Chức năng này cho phép khách có thể gửi bất kỳ câu hỏi, thắc mắc liên quan đến hệ thống để quản trị viên xử lý, khắc phục.

– **Chức năng xem thông tin tất cả các bác sĩ**

Chức năng này cho phép khách có thể xem được chi tiết các thông tin bác sĩ, xem số lượng đánh giá, số lượng bình luận và số lượng lượt đặt khám bệnh.

– **Chức năng xem thông tin phòng khám bệnh**

Chức năng này cho phép khách có thể xem được chi tiết các bài viết liên quan đến thông tin phòng khám bệnh như : xem thông tin, trang thiết bị, chi tiết giá và quy trình khám bệnh của phòng khám.

– **Chức năng xem thông tin bác sĩ và bình luận của bác sĩ**

Chức năng này cho phép khách có thể xem đầy đủ thông tin bác sĩ và những bình luận mà các thành viên bình luận cho chính bác sĩ đó.

– **Chức năng tìm kiếm theo địa chỉ vùng miền các phòng khám và bác sĩ**

Chức năng này cho phép khách có thể tìm kiếm các phòng khám theo địa chỉ mà mình cần tìm kiếm. Mục đích chức năng này giúp cho việc tìm kiếm dễ dàng hơn theo vùng miền.

### 2.2.2. Thành viên có các chức năng sau

– **Chức năng đăng nhập/đăng xuất**

Chức năng này cho phép thành viên có thể đăng nhập vào ứng dụng để thực hiện các chức năng của mình và đăng xuất khỏi ứng dụng khi không còn sử dụng ứng dụng.

– **Chức năng quản lý tài khoản**

Chức năng này cho phép thành viên có thể xem và chỉnh sửa các thông tin cá nhân đăng ký trong hệ thống như Họ tên, ngày sinh, ảnh đại diện, .... Và quản lý bảo mật cá nhân thông qua việc thay đổi, cập nhật mật khẩu đăng nhập.

– **Chức năng bình luận**

Chức năng này cho phép thành viên có thể bình luận vào bất kỳ bác sĩ nào mà mình thích hoặc đã từng đặt lịch để đưa ra những góp ý cho chính bác sĩ đó.

– **Chức năng đánh giá**

Chức năng này cho phép thành viên đánh giá bác sĩ qua việc cho số lượng sao từ 1 đến 5 sao. Chức năng này để đánh giá độ yêu thích của bệnh nhân đối với bác sĩ.

– **Chức năng đặt lịch khám bệnh**

Chức năng này cho phép thành viên đánh giá bác sĩ qua việc cho số lượng sao từ 1 đến 5 sao. Chức năng này để đánh giá độ yêu thích của bệnh nhân đối với bác sĩ.

– **Chức năng xem lại lịch sử đặt lịch khám bệnh**

Chức năng này cho phép thành viên có thể xem lại tất cả các bác sĩ nào mình đã đặt lịch.

– **Chức năng đăng ký để trở thành bác sĩ**

Chức năng này cho phép thành viên có thể đăng ký để trở thành bác sĩ nếu có sự cho phép của người quản trị hệ thống.

– **Chức năng nhận email khi đặt lịch khám bệnh**

Chức năng này cho phép thành viên có thể nhận email khi đặt lịch khám bệnh với nội dung chi tiết về thông tin phòng khám, bác sĩ và thời gian khám bệnh.

### **2.2.3. Bác sĩ có các chức năng sau**

– **Chức năng đăng ký phòng khám hoặc bệnh viện**

Chức năng này cho phép bác sĩ đăng ký phòng khám của mình trên ứng dụng.

– **Chức năng tạo lịch khám bệnh**

Chức năng này cho phép bác sĩ đăng ký lịch phù hợp với thời gian của mình để khám bệnh.

– **Chức năng tạo các bài viết về phòng khám**

Chức năng này cho phép bác sĩ có thể tạo các bài viết về các thông tin của phòng khám như : bài viết chi tiết về phòng khám, trang thiết bị, quy trình khám bệnh.

– **Chức năng tạo bảng giá cho phòng khám**

Chức năng này cho phép các bác sĩ có thể tạo ra danh sách các bảng giá cho phòng khám.

– **Chức năng báo cáo những tài khoản nào đặt lịch mà không đến khám**

Chức năng này cho phép các bác sĩ báo cáo tài khoản đặt lịch mà không đến khám về hệ thống, nếu số lượng báo cáo lớn hơn 2 lần thì tài khoản đó sẽ bị khóa.

– **Chức năng xem lịch đã đặt khám**

Chức năng này cho phép các bác sĩ có thể theo dõi và xem lịch bệnh nhân đến khám.

– **Chức năng nhận email khi có bệnh nhân đặt khám**

Chức năng này cho phép các bác sĩ nhận email khi có bệnh nhân đặt lịch với nội dung chi tiết về thông tin người đặt lịch và bệnh nhân.

### **2.2.4. Quản trị viên có các chức năng sau**

– **Chức năng quản lý người dùng**

Chức năng này cho phép quản trị viên quản lý được tất cả người dùng, đã đăng ký tồn tại trong hệ thống: Xác nhận, chỉnh sửa thông tin, xóa, khóa một tài khoản bất kì.

– **Chức năng quản lý mã code**

Chức năng này cho phép quản trị viên tạo mới những mã code để cung cấp cho các thành viên mong muốn trở thành vai trò bác sĩ trong ứng dụng. Quản trị có thể thêm sửa, xóa mã code.

– **Chức năng quản lý phòng khám**

Chức năng này cho phép quản trị xem chi tiết thông tin các phòng khám.

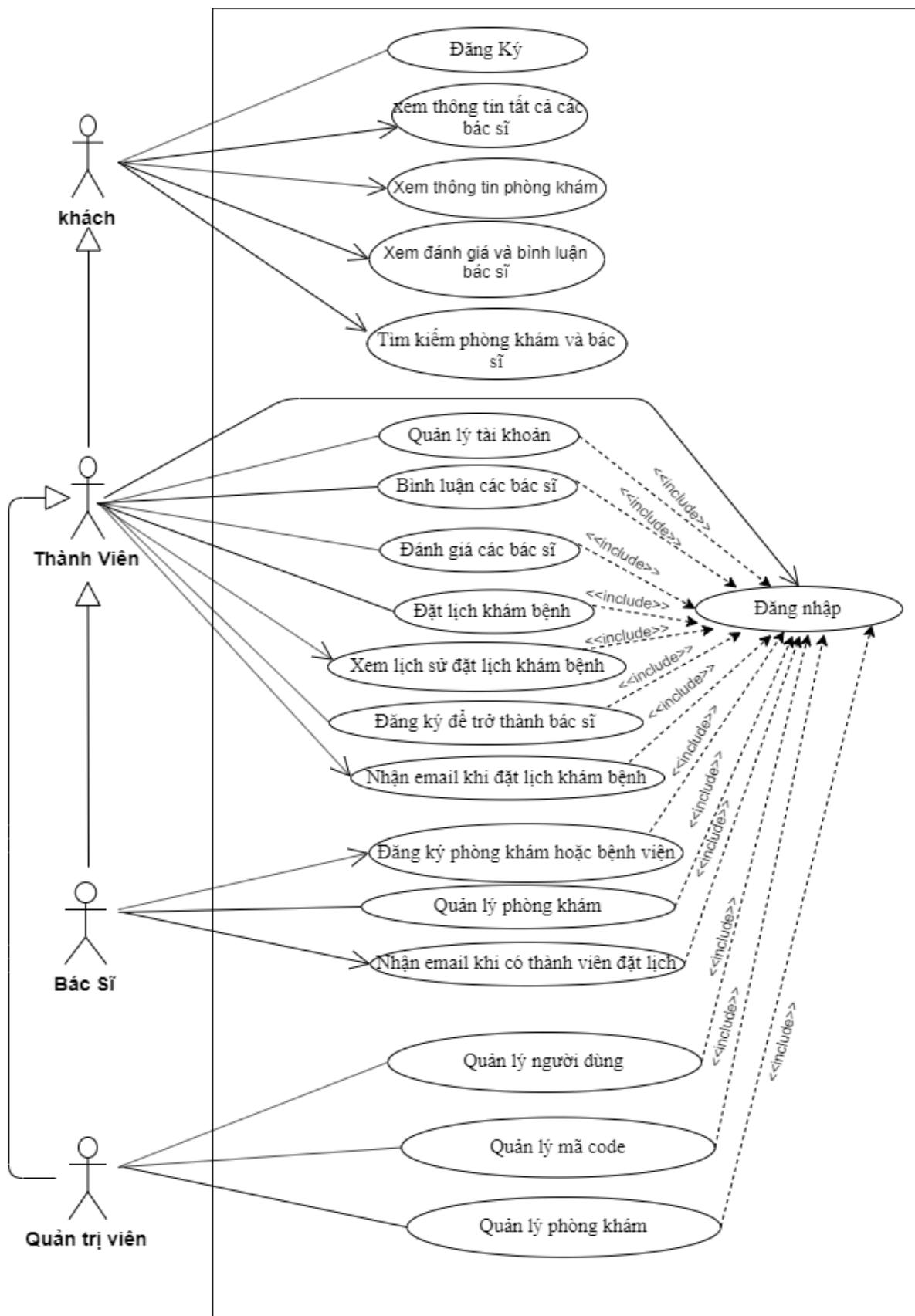
## 2.3. SƠ ĐỒ CA SỬ DỤNG

### 2.3.1. Các tác nhân trong hệ thống

Hệ thống có 4 tác nhân là Quản trị viên, Bác sĩ, Thành viên và Khách:

- Quản trị viên : Là người có mọi quyền trong hệ thống, quản lý người dùng, quản lý mã code, quản lý phòng khám và bác sĩ.
- Bác sĩ : Là người đại diện cho các phòng khám, có chức năng đăng ký phòng khám hoặc bệnh viện, tạo lịch khám bệnh, tạo các bài viết cho phòng khám báo cáo tài khoản khi không đến khám bệnh, chức năng xem lịch đặt khám bệnh, chức năng nhận email khi có người đặt lịch.
- Thành viên : Là người đã có tài khoản hợp lệ trong hệ thống, là người có thể thực hiện việc đặt lịch, bình luận và đánh giá các bác sĩ.
- Khách : Là người chưa có tài khoản trong hệ thống, ghé thăm tìm hiểu về hệ thống, tìm kiếm các bác sĩ qua việc lựa chọn khu vực phù hợp.

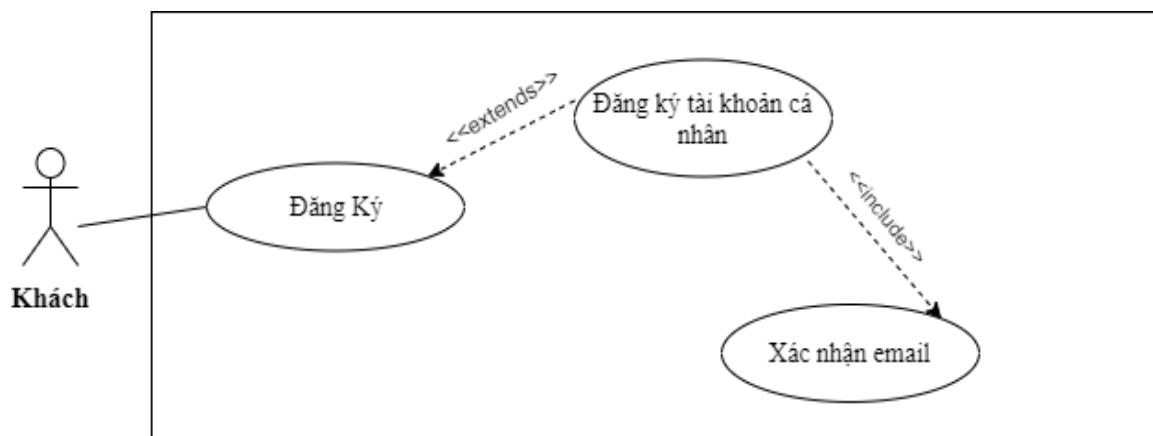
### 2.3.2. Sơ đồ ca sử dụng tổng quát của hệ thống



Hình 2.3: Sơ đồ ca sử dụng tổng quát của hệ thống

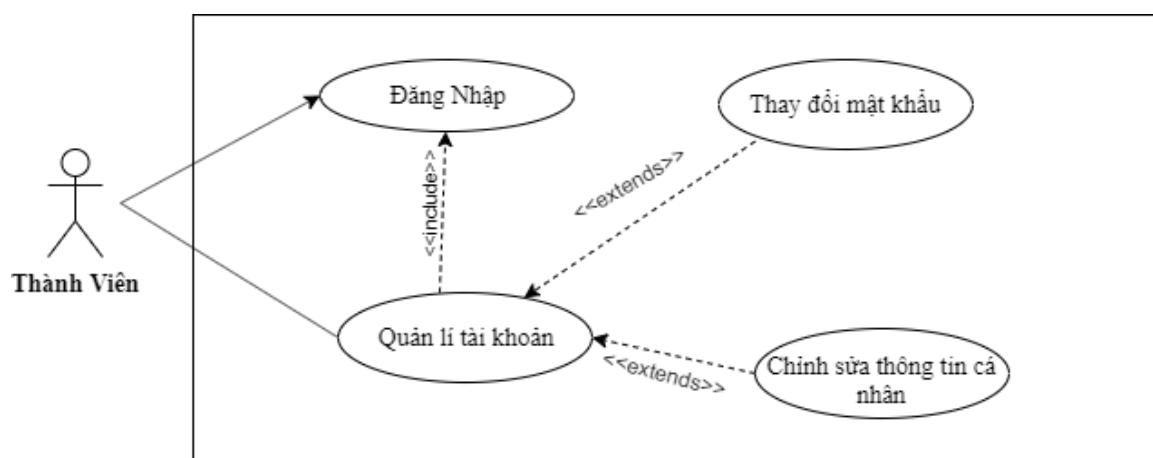
### 2.3.3. Sơ đồ ca sử dụng đăng ký tài khoản của khách

➤ Đăng ký thành viên

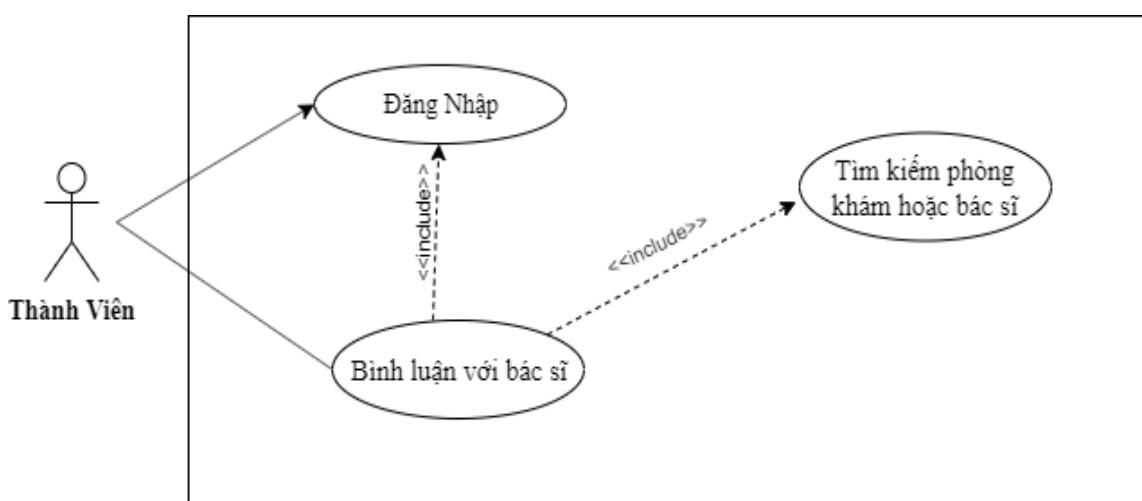


Hình 2.4: Sơ đồ ca sử dụng đăng ký tài khoản của Khách

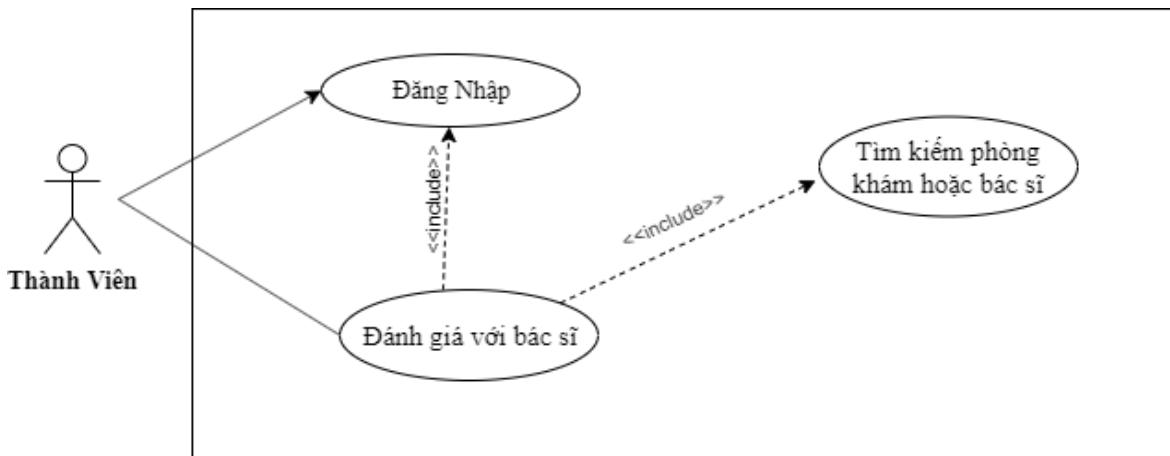
### 2.3.4. Sơ đồ ca sử dụng của thành viên



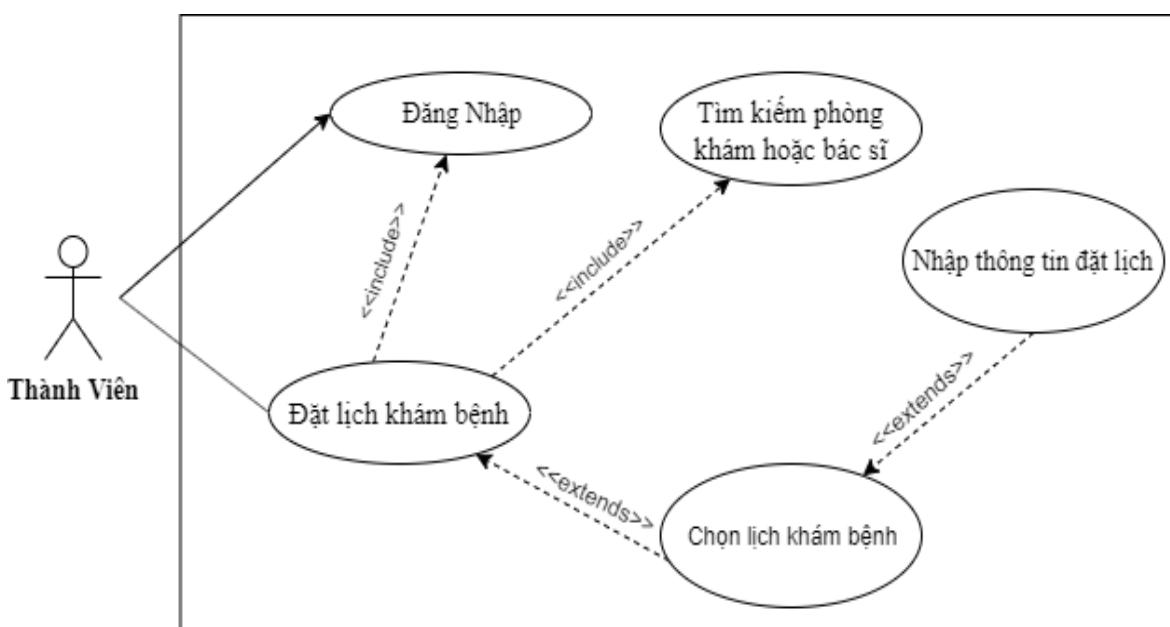
Hình 2.5: Sơ đồ ca sử dụng quản lý tài khoản trong hệ thống



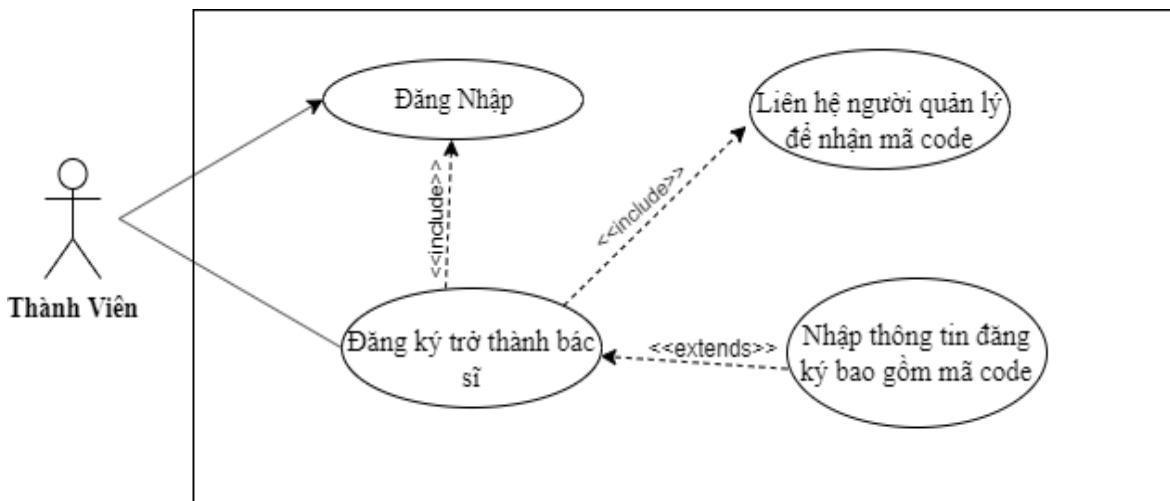
Hình 2.6: Sơ đồ ca sử dụng bình luận với bác sĩ.



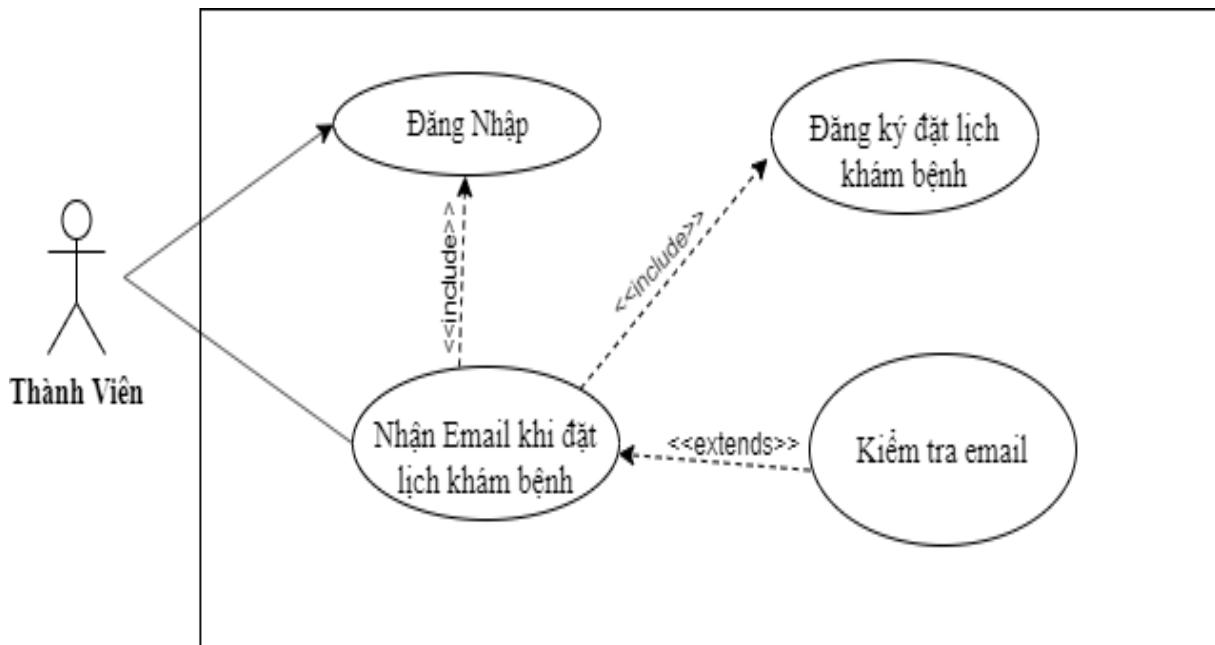
Hình 2.7: Sơ đồ ca sử dụng đánh giá với bác sĩ.



Hình 2.8: Sơ đồ ca sử dụng đặt lịch khám bệnh.

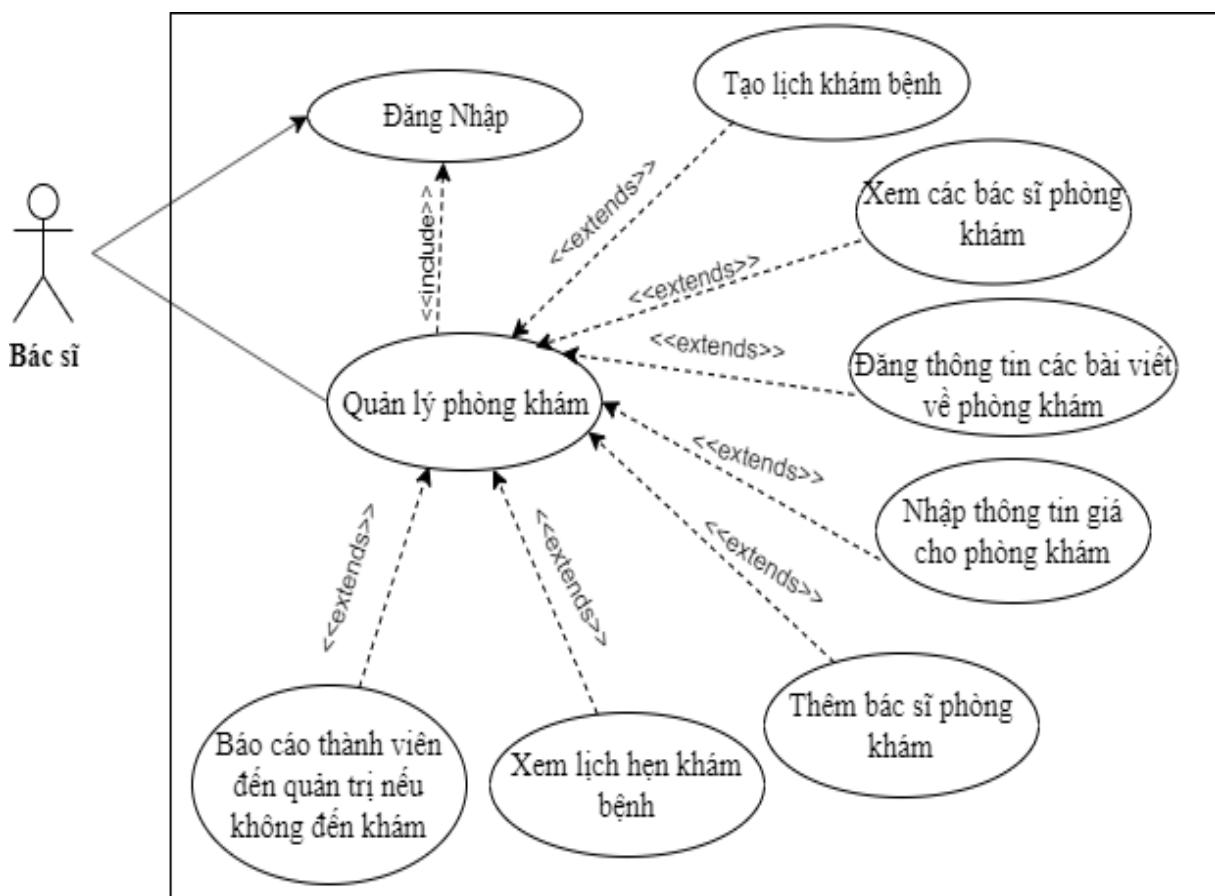


Hình 2.9: Sơ đồ ca sử dụng đăng ký trở thành bác sĩ

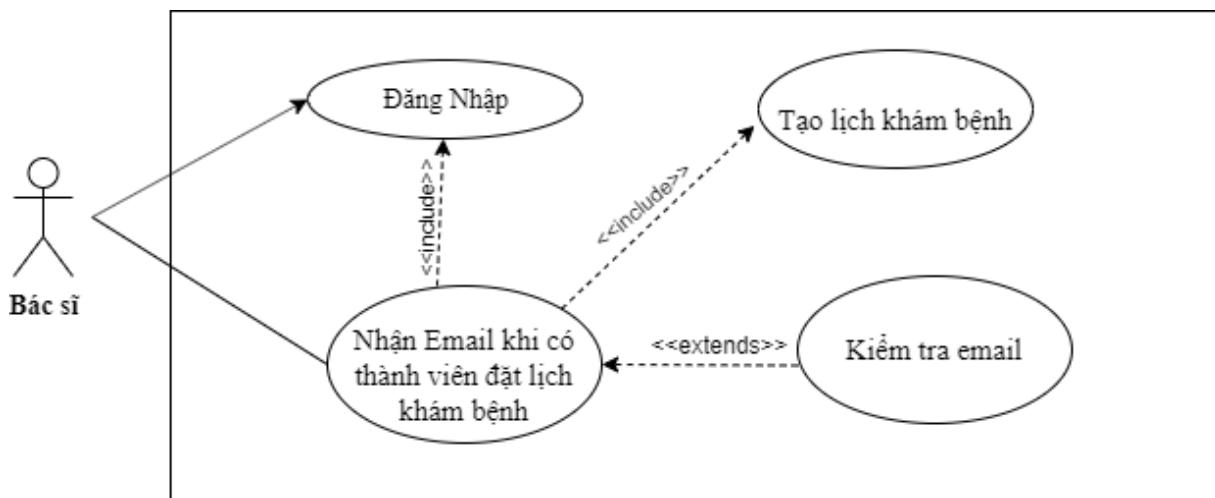


Hình 2.10: Sơ đồ ca sử dụng nhận email khi đăng ký lịch khám bệnh.

### 2.3.5. Sơ đồ ca sử dụng của bác sĩ

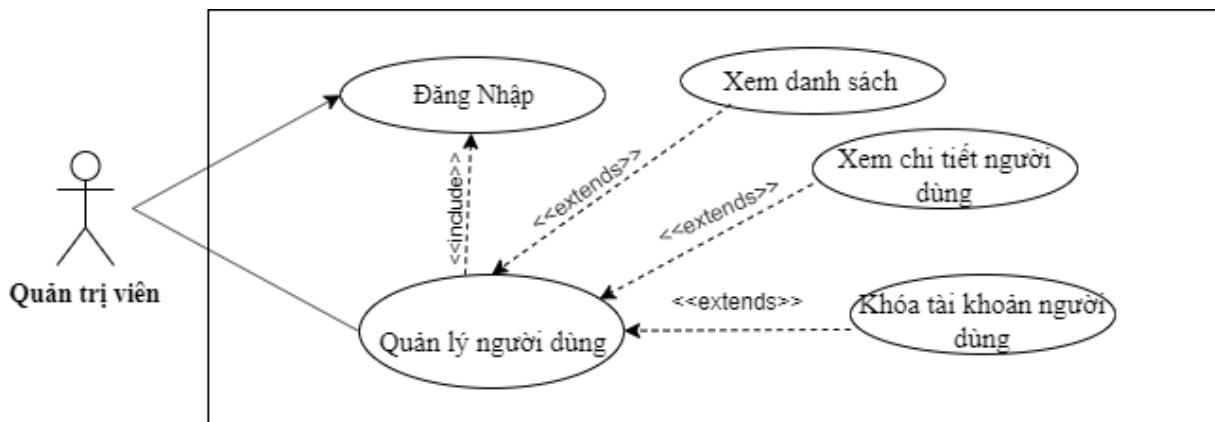


Hình 2.11: Sơ đồ ca sử dụng quản phòng khám

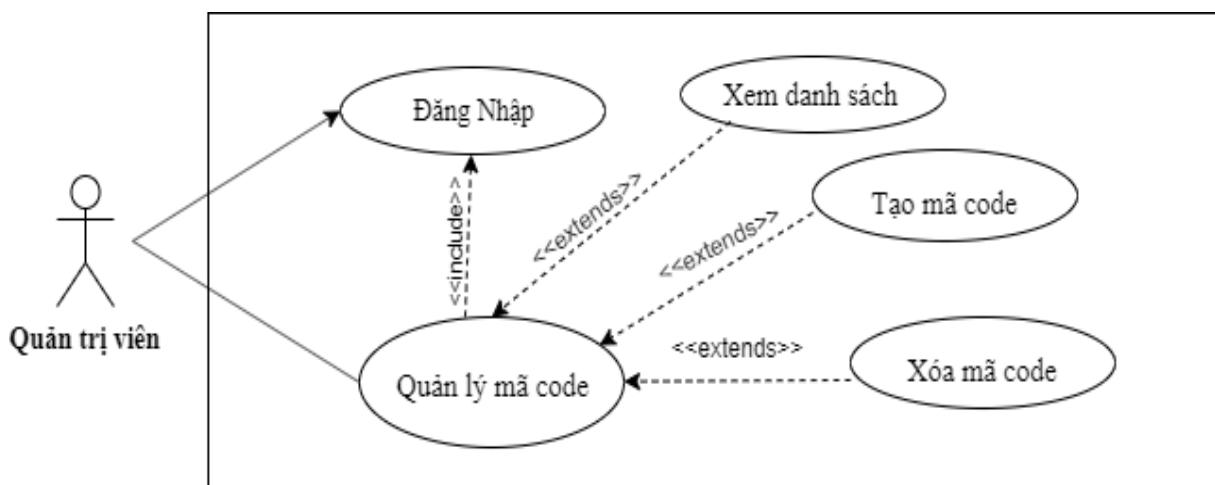


Hình 2.12: Sơ đồ ca sử dụng nhận email khi có thành viên đặt lịch

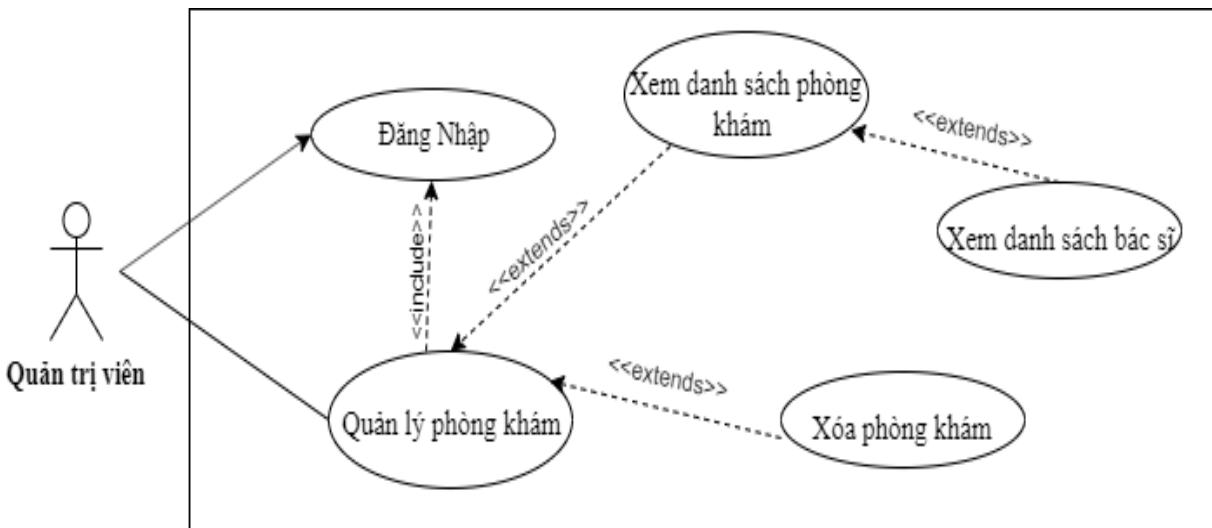
### 2.3.6. Sơ đồ ca sử dụng của quản trị viên



Hình 2.13: Sơ đồ ca sử dụng quản lý người dùng trong hệ thống



Hình 2.14: Sơ đồ ca sử dụng quản lý mã code để cấp phép tạo bác sĩ.



Hình 2.15: Sơ đồ ca sử dụng quản lý phòng khám

## 2.4. ĐẶC TẢ CA SỬ DỤNG

### 2.4.1. Đặc tả ca sử dụng đăng ký

Bảng 2.1: Đặc tả ca sử dụng đăng ký

Tên ca sử dụng	Đăng ký
Tác nhân	Khách
Mô tả	Cho phép khách đăng ký để trở thành thành viên của ứng dụng
Điều kiện kích hoạt	Người sử dụng truy cập hệ thống nhưng chưa có tài khoản. Chọn mục <b>Đăng ký</b> .
Các bước thực hiện	<ol style="list-style-type: none"> <li>Khách nhập các thông tin đăng ký: email, tên, số điện thoại, và mật khẩu, xác nhận mật khẩu.</li> <li>Bấm vào button <b>Đăng ký</b>.</li> <li>Hệ thống kiểm tra dữ liệu.</li> <li>Thông báo kết quả đăng ký.</li> </ol>
Kết quả	Nếu các thông tin hợp lệ, thông báo đăng ký thành công. Gửi yêu cầu xác nhận thông qua email đã đăng ký.
Trường hợp lỗi	<ol style="list-style-type: none"> <li>Người dùng nhập các thông tin đăng ký không hợp lệ</li> <li>Hệ thống kiểm tra dữ liệu nhập vào là sai.</li> <li>Các thông tin trùng lặp, đã tồn tại trong hệ thống.</li> <li>Thông báo lỗi ra màn hình.</li> </ol>

#### 2.4.2. Đặc tả ca sử dụng đăng nhập

Bảng 2.2: Đặc tả ca sử dụng đăng nhập

Tên ca sử dụng	Đăng nhập
Tác nhân	Thành viên, Bác sĩ, Quản trị viên
Mô tả	Cho phép thành viên, quản lý hoặc quản trị viên đăng nhập vào hệ thống để thực hiện các chức năng của mình.
Điều kiện kích hoạt	Thành viên, quản lý hoặc quản trị viên truy cập hệ thống nhưng chưa đăng nhập. Chọn mục <b>Đăng nhập</b> .
Các bước thực hiện	<ol style="list-style-type: none"> <li>1. Người dùng nhập các thông tin email và mật khẩu.</li> <li>2. Bấm nút <b>Đăng nhập</b>.</li> <li>3. Hệ thống kiểm tra dữ liệu nhập là đúng.</li> <li>4. Đưa đến trang chủ.</li> </ol>
Kết quả	Đưa đến trang chủ của hệ thống.
Trường hợp lỗi	<ol style="list-style-type: none"> <li>1. Người dùng không nhập các thông tin email và mật khẩu.</li> <li>2. Hệ thống kiểm tra dữ liệu nhập email và mật khẩu sai.</li> <li>3. Thông báo lỗi ra màn hình.</li> </ol>

#### 2.4.3. Đặc tả ca sử dụng quản lý tài khoản

Bảng 2.3: Đặc tả ca sử dụng quản lý tài khoản

Tên ca sử dụng	Quản lý tài khoản
Tác nhân	Thành viên, Quản lý, Quản trị viên
Mô tả	Cho phép thành viên, quản lý hoặc quản trị viên đăng nhập vào hệ thống để quản lý tài khoản cá nhân của mình bao gồm chỉnh sửa thông tin cá nhân và mật khẩu.
Điều kiện kích hoạt	Thành viên, quản lý hoặc quản trị viên đăng nhập vào hệ thống. Chọn mục <b>Profile/Tài khoản cá nhân</b> .
Các bước thực hiện	<ol style="list-style-type: none"> <li>1. Người dùng nhập các thông tin cá nhân và mật khẩu muốn chỉnh sửa.</li> <li>2. Bấm nút <b>Update/Cập nhật</b></li> <li>3. Hệ thống kiểm tra dữ liệu nhập là đúng.</li> <li>4. Thực hiện cập nhật thông tin.</li> </ol>
Kết quả	Đưa đến trang quản lý tài khoản.

<b>Trường hợp lỗi</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhập không chính xác các thông tin cá nhân và mật khẩu.</li> <li>2. Hệ thống kiểm tra dữ liệu nhập vào là sai.</li> <li>3. Thông báo lỗi ra màn hình.</li> </ol>
-----------------------	---

#### 2.4.4. Đặc tả ca sử dụng đăng ký trở thành bác sĩ

Bảng 2.4: Đặc tả ca sử dụng đăng ký thành bác sĩ

Tên ca sử dụng	Đăng ký thành bác sĩ
<b>Tác nhân</b>	Thành viên.
<b>Mô tả</b>	Cho phép thành viên có thể đăng ký trở thành bác sĩ và tạo phòng khám hoặc bệnh viện hoặc có thể được thêm vào phòng khám khác.
<b>Điều kiện kích hoạt</b>	Thành viên phải liên hệ qua email của hệ thống để quản trị viên cấp mã code và đăng nhập vào hệ thống. Chọn Icon đăng ký thành bác sĩ.
<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>1. Thành viên liên hệ đến quản trị hệ thống để nhận mã code thông qua email ứng dụng.</li> <li>2. Nhập đầy đủ thông tin trong form và kèm theo mã code.</li> <li>3. Chọn button thêm mới.</li> </ol>
<b>Kết quả</b>	Thông báo kết quả thành công ra màn hình.
<b>Trường hợp lỗi</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhập các thông tin đề thi không hợp lệ</li> <li>2. Hệ thống kiểm tra dữ liệu nhập vào là sai</li> <li>3. Mã code không tồn tại.</li> <li>4. Thông báo lỗi ra màn hình.</li> </ol>

#### 2.4.5. Đặc tả ca sử dụng đặt lịch khám bệnh

Bảng 2.5: Đặc tả ca sử dụng đặt lịch khám bệnh

Tên ca sử dụng	Tạo bài thi
<b>Tác nhân</b>	Thành viên.
<b>Mô tả</b>	Cho phép thành viên có thể đặt lịch khám bệnh
<b>Điều kiện kích hoạt</b>	<p>Người quản lý đăng nhập vào hệ thống.</p> <p>Tìm kiếm bác sĩ và chọn lịch phù hợp.</p>

<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>Người dùng đăng nhập vào ứng dụng.</li> <li>Sau đó tìm kiếm bác sĩ.</li> <li>Chọn lịch phù hợp với lịch khám bệnh của mình.</li> <li>Nhập đầy đủ thông tin để đặt lịch.</li> </ol>
<b>Kết quả</b>	Thông báo xuất hiện trên màn hình và hệ thống gửi email thông báo đã đặt lịch.
<b>Trường hợp lỗi</b>	<ol style="list-style-type: none"> <li>Thông báo lịch không tồn tại.</li> </ol>

#### 2.4.6. Đặc tả ca sử dụng đánh giá bác sĩ

Bảng 2.6: Đặc tả ca sử dụng đánh giá bác sĩ

<b>Tên ca sử dụng</b>	<b>Làm bài thi</b>
<b>Tác nhân</b>	Thành viên.
<b>Mô tả</b>	Cho phép thành viên có thể đánh giá bằng sao bác sĩ mình yêu thích.
<b>Điều kiện kích hoạt</b>	<p>Người quản lý đăng nhập vào hệ thống. Tìm kiếm bác sĩ.</p>
<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>Người dùng đăng nhập vào ứng dụng.</li> <li>Sau đó tìm kiếm bác sĩ.</li> <li>Đánh giá bác sĩ qua số lượng sao 5/5 sao.</li> </ol>
<b>Kết quả</b>	Sau khi thành viên đánh giá thì bác sĩ sẽ cập nhật lại trung bình số lượng sao.
<b>Trường hợp lỗi</b>	Thông báo chưa đăng nhập ra màn hình.

#### 2.4.7. Đặc tả ca sử dụng bình luận bác sĩ

Bảng 2.7: Đặc tả ca sử dụng bình luận bác sĩ

<b>Tên ca sử dụng</b>	<b>Quản lý kết quả các bài thi</b>
<b>Tác nhân</b>	Thành viên.
<b>Mô tả</b>	Cho phép thành viên có thể bình luận bác sĩ mình yêu thích.
<b>Điều kiện kích hoạt</b>	<p>Người quản lý đăng nhập vào hệ thống. Tìm kiếm bác sĩ. Chọn button bình luận.</p>

<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>1. Người dùng đăng nhập vào ứng dụng.</li> <li>2. Sau đó tìm kiếm bác sĩ.</li> <li>3. Chọn button Bình luận.</li> <li>4. Nhập nội dung bình luận vào khung nội dung bình luận</li> <li>5. Nhấn button Đăng</li> </ol>
<b>Kết quả</b>	Hiển thị nội dung mình vừa bình luận lên danh sách nội dung bình luận.
<b>Trường hợp lỗi</b>	Thông báo chưa đăng nhập ra màn hình.

#### 2.4.8. Đặc tả ca sử dụng quản lý người dùng

Bảng 2.8: Đặc tả ca sử dụng quản lý người dùng

Tên ca sử dụng	Quản lý người dùng
<b>Tác nhân</b>	Người quản trị viên
<b>Mô tả</b>	Cho phép người quản trị viên quản lý được toàn bộ thông tin về các thành viên, người dùng trong hệ thống.
<b>Điều kiện kích hoạt</b>	Người quản trị viên đăng nhập vào hệ thống. Chọn mục <b>Người dùng</b> ở thanh sidebar.
<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>1. Người chọn mục Users để xem toàn bộ danh sách các tài khoản trong hệ thống.</li> <li>2. Chọn vào từng người dùng để thực hiện các thay đổi.</li> <li>3. Quản trị viên có thể xóa hoặc khóa một tài khoản bất kỳ.</li> <li>3. Lưu lại thông tin đã chỉnh sửa.</li> </ol>
<b>Kết quả</b>	Đưa đến trang danh sách người dùng.
<b>Trường hợp lỗi</b>	<ol style="list-style-type: none"> <li>1. Lỗi do không có dữ liệu để hiển thị.</li> <li>2. Thông báo lỗi ra màn hình.</li> </ol>

#### 2.4.9. Đặc tả ca sử dụng quản lý mã code để đăng ký thành bác sĩ

Bảng 2.9: Đặc tả ca sử dụng quản lý mã code

Tên ca sử dụng	Quản lý các bài viết, mèo
<b>Tác nhân</b>	Người quản trị viên
<b>Mô tả</b>	Cho phép người quản trị viên tạo ra danh sách các mã code duy nhất trong hệ thống.

<b>Điều kiện kích hoạt</b>	Người quản trị viên đăng nhập vào hệ thống. Chọn mục <b>Quản lý mã code</b> ở thanh sidebar.
<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>Quản trị viên chọn mục Quản lý mã code để xem toàn bộ danh sách mã code.</li> <li>Quản trị viên chọn Thêm mới để thêm mã code mới.</li> <li>Quản trị viên có thể chỉnh sửa thông tin bài viết, xóa hoặc thay đổi trạng thái của danh sách mã code.</li> </ol>
<b>Kết quả</b>	Đưa đến trang danh sách quản lý mã code.
<b>Trường hợp lỗi</b>	<ol style="list-style-type: none"> <li>Lỗi do không có dữ liệu để hiển thị.</li> <li>Thông báo lỗi ra màn hình.</li> </ol>

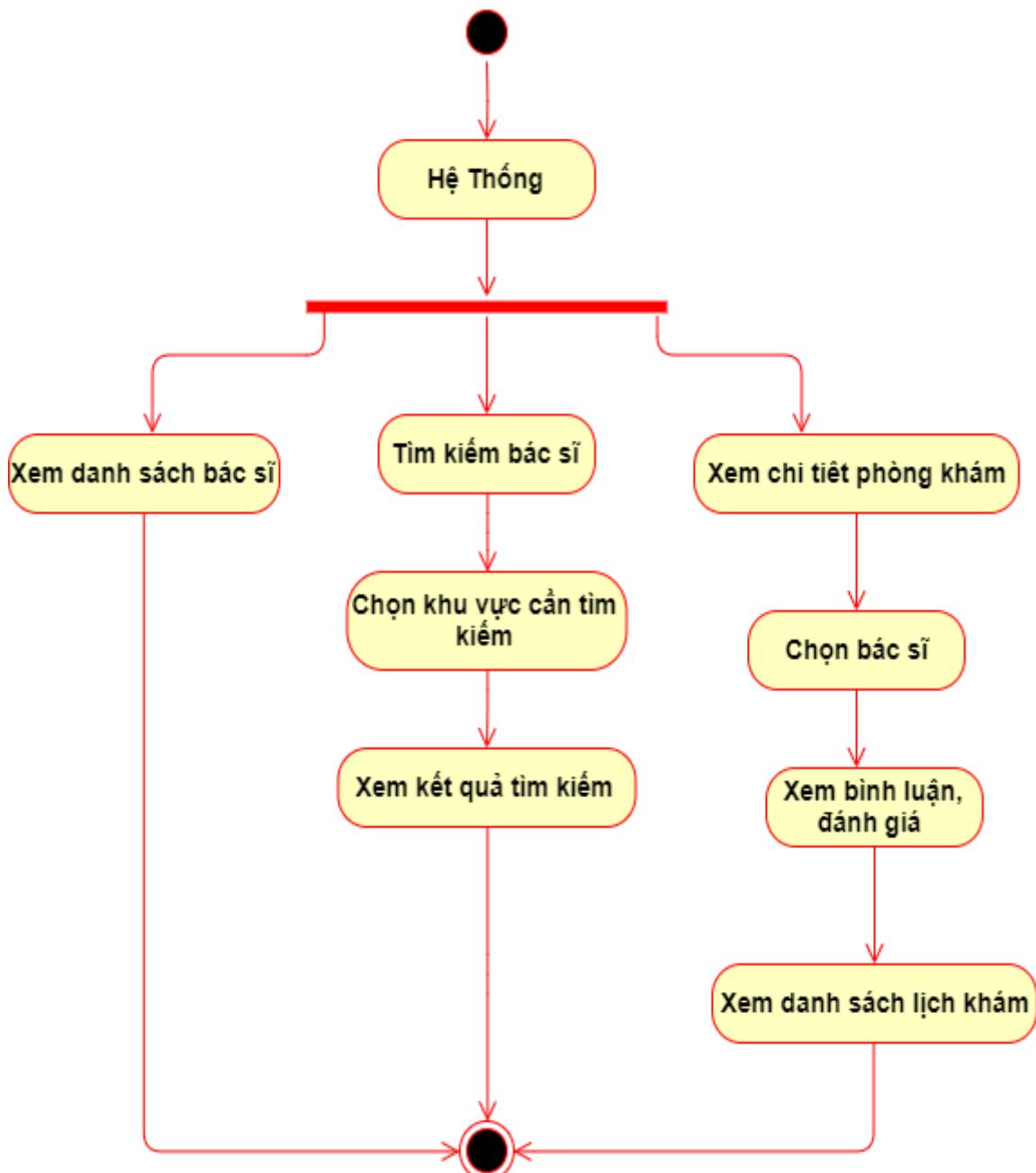
#### 2.4.10. Đặc tả ca sử dụng quản lý phòng khám

Bảng 2.10: Đặc tả ca sử dụng quản lý phòng khám

Tên ca sử dụng	Quản lý liên hệ
<b>Tác nhân</b>	Người quản trị viên
<b>Mô tả</b>	Cho phép người quản trị viên quản lý tất cả các phòng khám trong hệ thống.
<b>Điều kiện kích hoạt</b>	Người quản trị viên đăng nhập vào hệ thống. Chọn mục <b>Quản lý phòng khám</b> ở thanh sidebar
<b>Các bước thực hiện</b>	<ol style="list-style-type: none"> <li>Quản trị viên chọn mục Quản lý phòng khám để xem toàn bộ danh sách phòng khám của ứng dụng.</li> <li>Quản trị viên có thể chỉnh sửa thông tin bài viết, xóa hoặc thay đổi danh sách phòng khám.</li> </ol>
<b>Kết quả</b>	Đưa đến trang danh sách quản lý phòng khám.
<b>Trường hợp lỗi</b>	<ol style="list-style-type: none"> <li>Lỗi do không có dữ liệu để hiển thị</li> <li>Thông báo lỗi ra màn hình.</li> </ol>

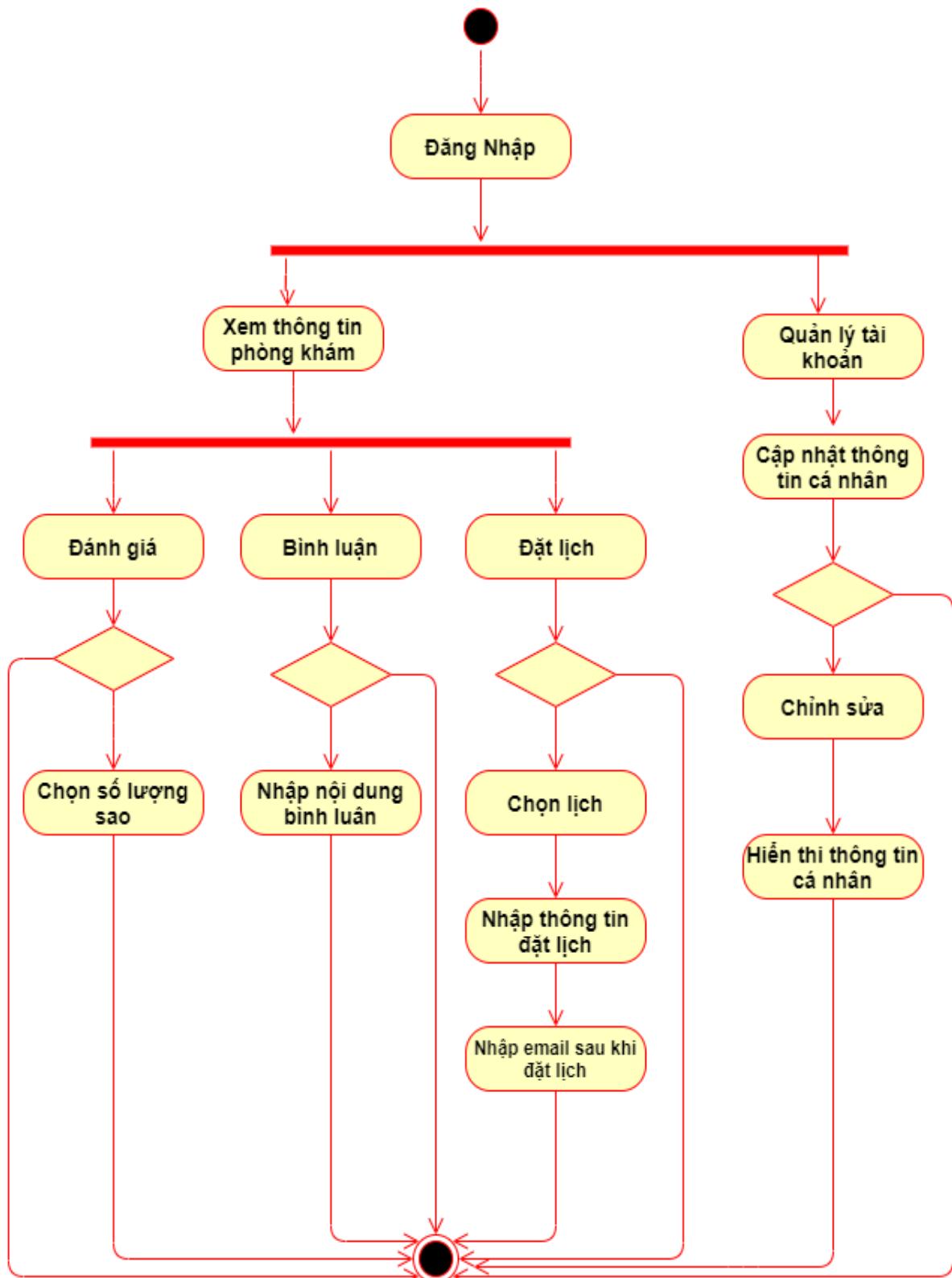
## 2.5. SƠ ĐỒ HOẠT ĐỘNG

### 2.5.1. Sơ đồ hoạt động của khách



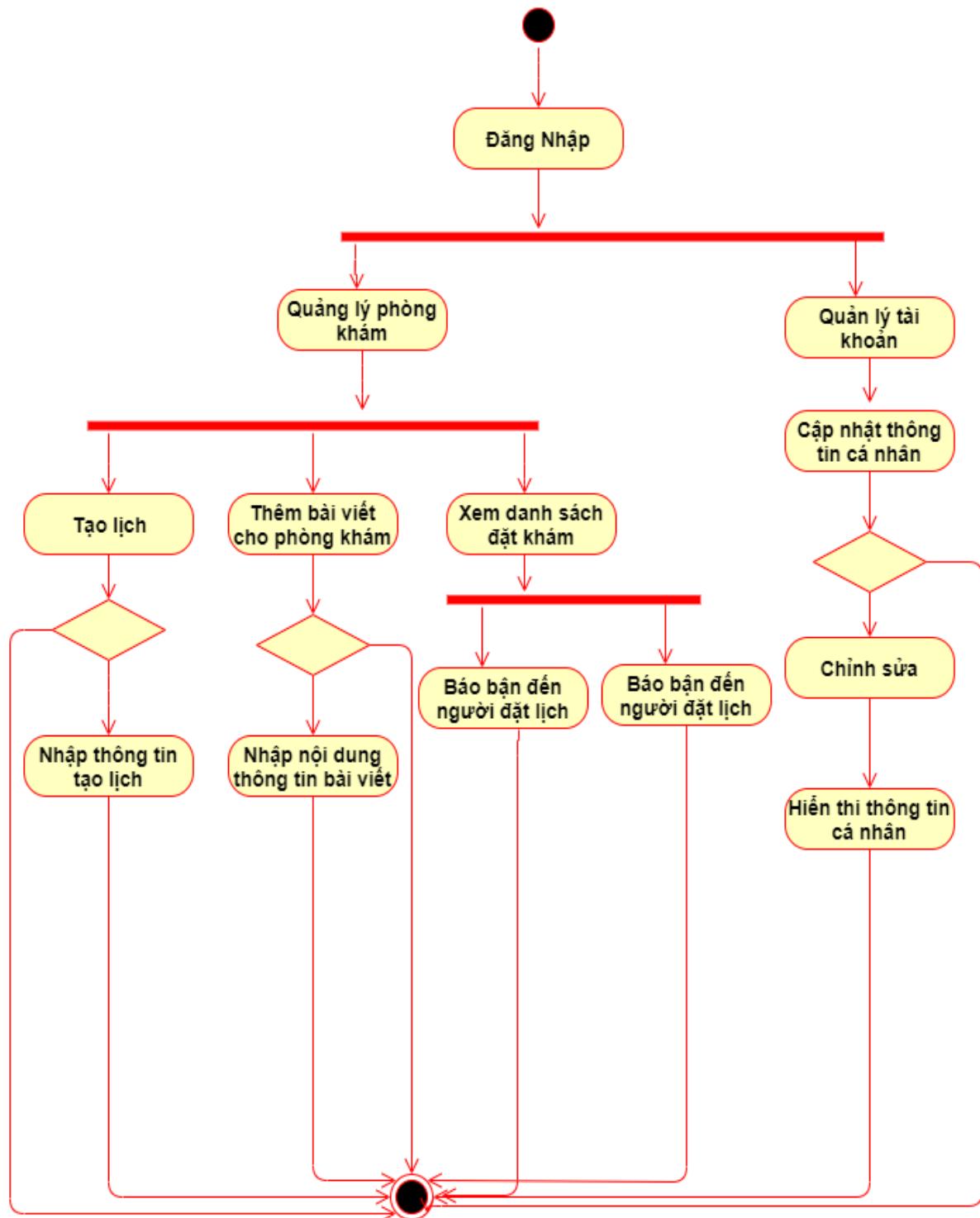
Hình 2.16: Sơ đồ hoạt động các chức năng của khách.

### 2.5.2. Sơ đồ hoạt động của thành viên



Hình 2.17: Sơ đồ hoạt động các chức năng của bác sĩ

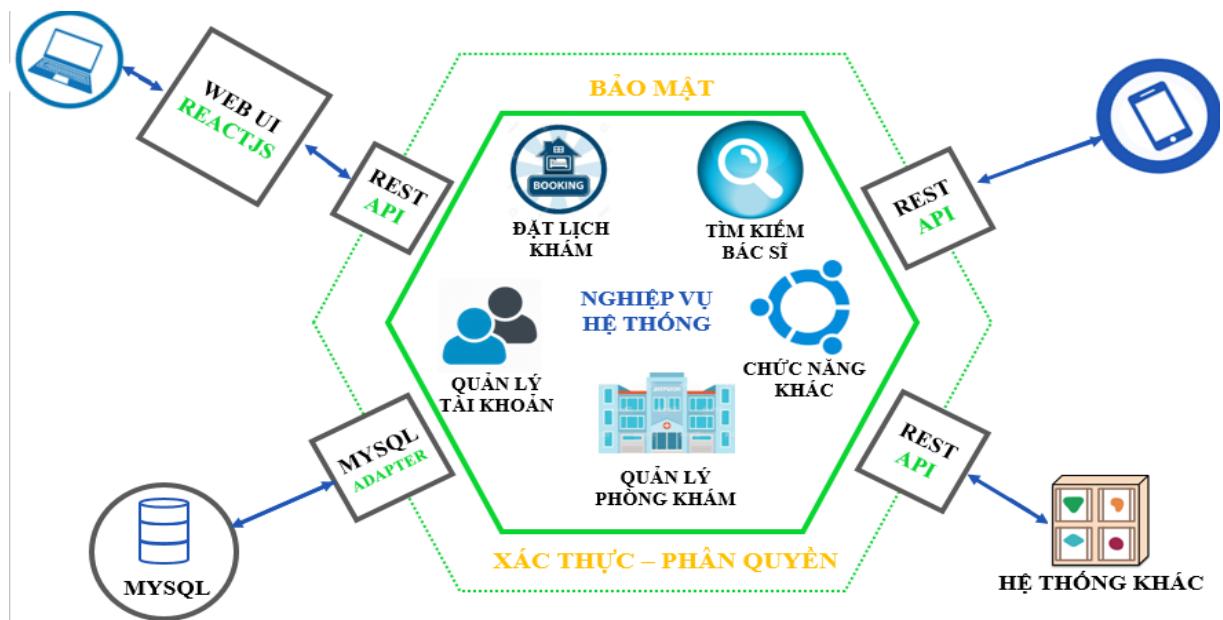
### 2.5.3. Sơ đồ hoạt động của quản trị viên



Hình 2.18: Sơ đồ hoạt động các chức năng của quản trị viên

## 2.6. KIẾN TRÚC HỆ THỐNG

Đây là ứng dụng có sự tương tác trên môi trường web và hệ thống sẽ được xây dựng theo mô hình client-server và dựa trên kiến trúc Monolithic. Trong đó server nắm vai trò chính trong xử lý logic nghiệp vụ và quản trị cơ sở dữ liệu, phía clients có trình duyệt web.

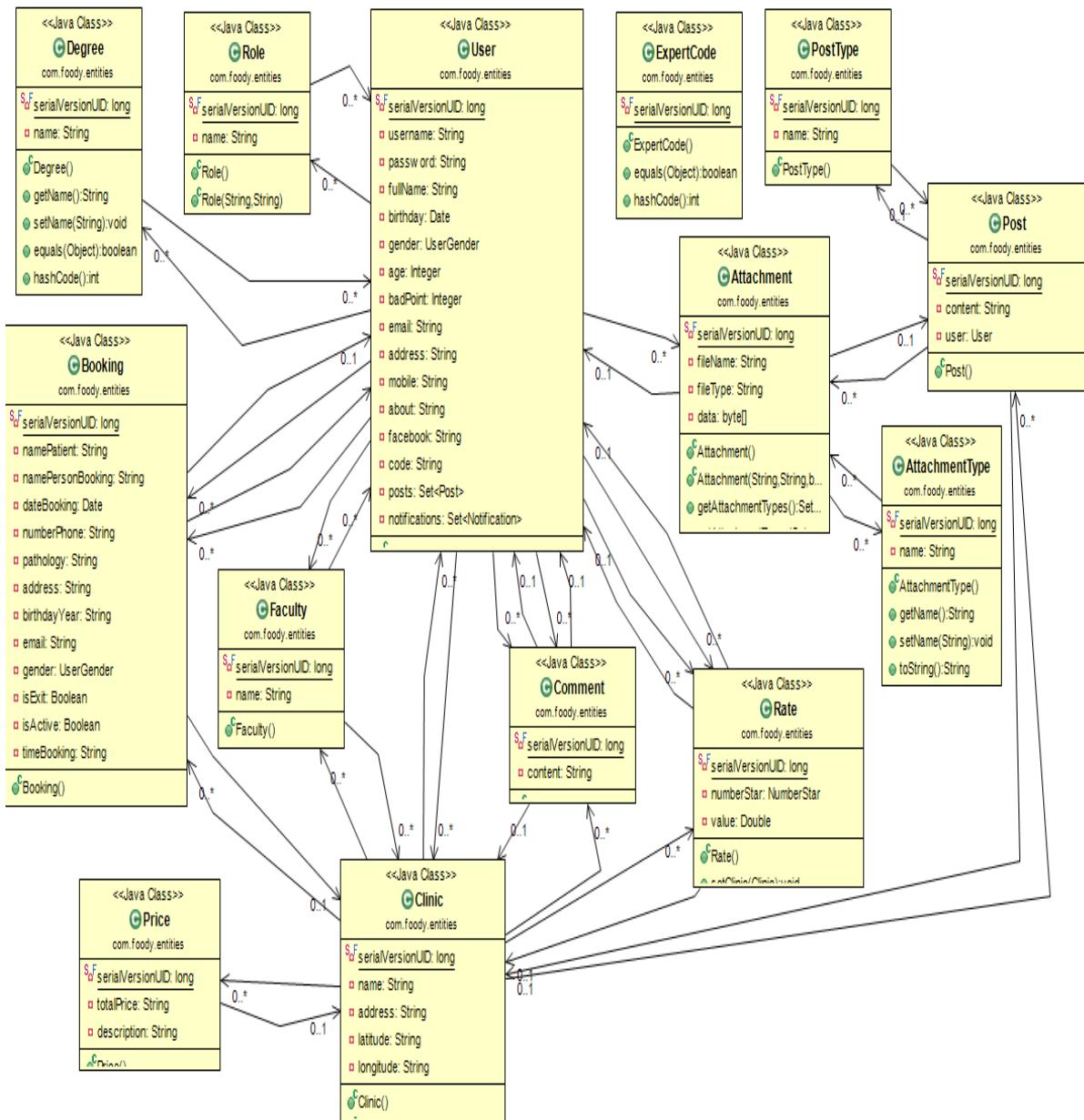


Hình 2.19: Sơ đồ kiến trúc hệ thống

Nhờ vào kiến trúc như vậy, chúng ta có thể triển khai môi trường client trên nhiều nền tảng khác nhau, các xử lý liên quan đến nghiệp vụ tập trung. Nó có thể tích hợp các ứng dụng khác thông qua REST API, đồng thời có thể xử lý các yêu cầu dạng HTTP, thực hiện các logic nghiệp vụ, truy cập cơ sở dữ liệu và có thể trao đổi cơ sở dữ liệu với hệ thống khác.

## 2.7. SƠ ĐỒ LỚP

Một lớp (class) là một tập hợp các đối tượng có cùng cấu trúc, cùng hành vi... Sơ đồ lớp thể hiện các khái của bất kỳ hệ thống hướng đối tượng nào. Sơ đồ lớp mô tả cái nhìn tĩnh của mô hình hoặc một phần của mô hình, miêu tả những thuộc tính và hành vi một cách chi tiết. Dựa vào những phân tích về chức năng, hệ thống có những thành phần và quan hệ được mô tả trong sơ đồ dưới đây:



Hình 2.20: Sơ đồ lớp ứng dụng đặt lịch khám bệnh

## 2.8. THIẾT KẾ CƠ SỞ DỮ LIỆU

### 2.8.1. Chi tiết các bảng cơ sở dữ liệu

- User:** là bảng dữ liệu người dùng, chứa thông tin của người dùng trong hệ thống.

Bảng 2.11: Bảng dữ liệu user

User				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	Bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
username	Varchar	255	Duy nhất	Tài khoản đăng nhập mật khẩu
fullname	Varchar	255		Tên đầy đủ của người dùng
Email	Varchar	255	Duy nhất	Email
password	Varchar	255		Mật khẩu
gender	Int	int		Giới tính
mobile	Varchar	255		Số điện thoại
birthday	Date			Ngày tháng năm sinh
Age	Int	int		Số tuổi người dùng
address	Varchar	255		Địa chỉ người dùng
Code	Varchar	255		Chứa mã code để xác thực khi đăng ký tài khoản.
bad_point	Int	11		Đếm số lượng khi người dùng không đến khám bệnh, nếu lớn hơn 3 lần thì tài khoản sẽ khóa
facebook	Varchar	255		Địa chỉ facebook người dùng

- **Role:** là bảng dữ liệu phân quyền, chứa quyền người dùng truy cập.

Bảng 2.12: Bảng dữ liệu role

Role				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
is_active	Bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Name	VARCHAR	255		Tên vai trò trong ứng dụng

- **User\_Role:** là bảng liên kết quan hệ n – n giữa bảng User và Role, chứa khóa chính của bảng User và Role.

Bảng 2.13: Bảng dữ liệu user\_role

User_Role				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
is_active	Bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
id_user	VARCHAR	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_role	VARCHAR	255	Khóa ngoại, liên kết với bảng Role	Mã bảng role

- **Degree:** là bảng dữ liệu chứa thông tin học hàm học vị của bác sĩ.

Bảng 2.14: Bảng dữ liệu degree

Degree				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	DATE			Ngày khởi tạo
update_at	DATE			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
is_active	BIT		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Name	VARCHAR	255		Tên học hàm, học vị trong ứng dụng

- **User\_Degree:** là bảng liên kết quan hệ n – n giữa bảng User và Degree, chứa khóa chính của bảng User và Degree.

Bảng 2.15: Bảng dữ liệu user\_degree

User_Degree				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	DATE			Ngày khởi tạo
update_at	DATE			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
id_user	VARCHAR	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_degree	VARCHAR	255	Khóa ngoại, liên kết với bảng Degree	Mã bảng học hàm, học vị

- **Faculty:** là bảng dữ liệu thông tin chuyên khoa của bác sĩ và phòng khám.

Bảng 2.16: Bảng dữ liệu faculty

Faculty				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	Bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Name	Varchar	255		Tên chuyên khoa khám bệnh trong ứng dụng

- **User\_Faculty:** là bảng liên kết quan hệ n – n giữa bảng User và Faculty, chứa khóa chính của bảng User và Faculty.

Bảng 2.17: Bảng dữ liệu user\_faculty

User_Faculty				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
id_user	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_faculty	Varchar	255	Khóa ngoại, liên kết với bảng Faculty	Mã chuyên khoa khám bệnh

- **Clinic:** là bảng dữ liệu chứa thông tin phòng khám bệnh.

Bảng 2.18: Bảng dữ liệu clinic

Clinic				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
is_active	Bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Name	VARCHAR	255		Tên phòng khám bệnh trong ứng dụng
address	VARCHAR	255		Địa chỉ phòng khám bệnh trong ứng dụng

- **User\_Clinic :** là bảng liên kết quan hệ n – n giữa bảng User và Clinic, chứa khóa chính của bảng User và Clinic.

Bảng 2.19: Bảng dữ liệu user\_clinic

User_Clinic				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
id_user	VARCHAR	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_clinic	VARCHAR	255	Khóa ngoại, liên kết với bảng Clinic	Mã phòng khám bệnh

- Clinic\_Faculty**: là bảng liên kết quan hệ n – n giữa bảng Clinic và Faculty, chứa khóa chính của bảng Clinic và Faculty.

Bảng 2.20: Bảng dữ liệu clinic\_faculty

Clinic_Faculty				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
id_faculty	Varchar	255	Khóa ngoại, liên kết với bảng Faculty	Mã chuyên khoa khám bệnh của bác sĩ và phòng khám
id_clinic	Varchar	255	Khóa ngoại, liên kết với bảng Clinic	Mã phòng khám bệnh

- Price**: là bảng dữ liệu chi tiết giá của phong khám.

Bảng 2.21: Bảng dữ liệu price

Price				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
description	Longtext			Mô tả chi tiết thông tin giá
total_price	Varchar	255		Giá khám
id_clinic	Varchar	255	Khóa ngoại, liên kết với bảng Clinic	Mã tên khoa khám

- Booking:** là bảng dữ liệu về thông tin lịch khám bệnh và người đặt lịch khám bệnh.

Bảng 2.22: Bảng dữ liệu booking

Booking				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
address	Varchar	255		Địa chỉ người đặt lịch khám bệnh
birthday_year	Varchar	255		Năm sinh bệnh nhân
date_booking	Date			Ngày tạo lịch đặt khám bệnh
Email	Varchar	255		Email người đặt lịch khám bệnh
is_exit	Bit		1: Đã đặt lịch, 0: Chưa đặt lịch	Trạng thái đã đặt lịch hay chưa của lịch khám
name_patient	Varchar	255		Tên đầy đủ bệnh nhân
name_person_booking	Varchar	255		Tên đầy đủ người đặt lịch
number_phone	Varchar	255		Số điện thoại người đặt lịch
name_person_booking	Varchar	255		Tên đầy đủ người đặt lịch
pathology	Varchar	255		Triệu chứng của bệnh nhân
time_booking	Varchar	255		Thời gian lịch khám bệnh
id_clinic	Varchar	255	Khóa ngoại, liên kết với bảng Clinic	Mã tên khoa khám
id_user	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_expert	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã bác sĩ

- **Rate:** là bảng dữ liệu người dùng đánh giá cho bác sĩ.

Bảng 2.23: Bảng dữ liệu rate

Rate				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.
number_start	Int	11		Số lượng người dùng đánh giá từ 1 đến 5 sao
id_clinic	VARCHAR	255	Khóa ngoại, liên kết với bảng Clinic	Mã tên khoa khám
id_user	VARCHAR	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_expert	VARCHAR	255	Khóa ngoại, liên kết với bảng User	Mã bác sĩ

- **Expert\_Code:** là bảng chứa mã dữ liệu mã code cấp cho người dùng để đăng ký trở thành bác sĩ.

Bảng 2.24: Bảng dữ liệu expert\_code

Expert_Code				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	VARCHAR	255	Khóa chính, Duy nhất	Khóa chính của bảng, cũng chính là mã code
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	VARCHAR	255		Chứa khóa chính của người khởi tạo.
update_by	VARCHAR	255		Chứa khóa chính của người cập nhật.
delete_by	VARCHAR	255		Chứa khóa chính của người xóa.

- Comment:** là bảng dữ liệu người dùng bình luận cho bác sĩ.

Bảng 2.25: Bảng dữ liệu comment

Comment				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
content	Varchar	255		Nội dung bình luận cho bác sĩ
id_clinic	Varchar	255	Khóa ngoại, liên kết với bảng Clinic	Mã tên khoa khám
id_user	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_expert	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã bác sĩ

- Post\_Type:** là bảng dữ liệu chứa thông tin kiểu của bảng Post như: thông tin phòng khám, quy trình khám bệnh, trang thiết bị.

Bảng 2.26: Bảng dữ liệu post\_type

Post_Type				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải Thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Name	Varchar	255		Tên kiểu bài viết cho phòng khám

- **Post:** là bảng về chi tiết bài viết về phòng khám để giới thiệu thông tin phòng khám.

Bảng 2.27: Bảng dữ liệu post

Post				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
content	Varchar	255		Nội dung bài viết
id_clinic	Varchar	255	Khóa ngoại, liên kết với bảng Clinic	Mã tên khoa khám
id_user	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_post_type	Varchar	255	Khóa ngoại, liên kết với bảng Post_type	Mã của kiểu bài, thể loại bài viết

- **Attachment\_Type:** là bảng dữ liệu chứa thông tin kiểu của bảng Post như: thông tin phòng khám, quy trình khám bệnh, trang thiết bị.

Bảng 2.28: Bảng dữ liệu attachment\_type

Attachment-Type				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Name	Varchar	255		Tên kiểu ảnh

- **Attachment:** là bảng dữ liệu ảnh của người dùng và phòng khám.

Bảng 2.29: Bảng dữ liệu attachment

Attachment				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
is_active	bit		1: Mở, 0: Khóa	Trạng thái khóa hay còn hoạt động
Data	Longblob			Chứa dữ liệu byte code của hình ảnh
file_name	Varchar	255		Tên của file, hình ảnh
file_type	Varchar	255		Kiểu định dạng hình ảnh
id_user	Varchar	255	Khóa ngoại, liên kết với bảng User	Mã người dùng
id_post	Varchar	255	Khóa ngoại, liên kết với bảng Post	Mã của bài viết

- **Province:** là bảng chứa tên các tỉnh, thành phố của Việt Nam.

Bảng 2.30: Bảng dữ liệu province

Province				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
_code	Varchar	255		Mã của tỉnh hoặc thành phố.
_name	Varchar	255		Tên của tỉnh hoặc thành phố.

- **Attachment\_AttachmentType:** là bảng liên kết quan hệ n – n giữa bảng Attachment và Attachment\_Type, chứa khóa chính của bảng Attachment và Attachment\_Type.

Bảng 2.31: Bảng dữ liệu attachment\_attachmenttype

Attachment_AttachmentType				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
id_attachment	Varchar	255	Khóa ngoại, liên kết với bảng Attachment	Mã của file đính kèm
id_attachmenttype	Varchar	255	Khóa ngoại, với bảng Attachment_Type	Mã của kiểu file đính kèm

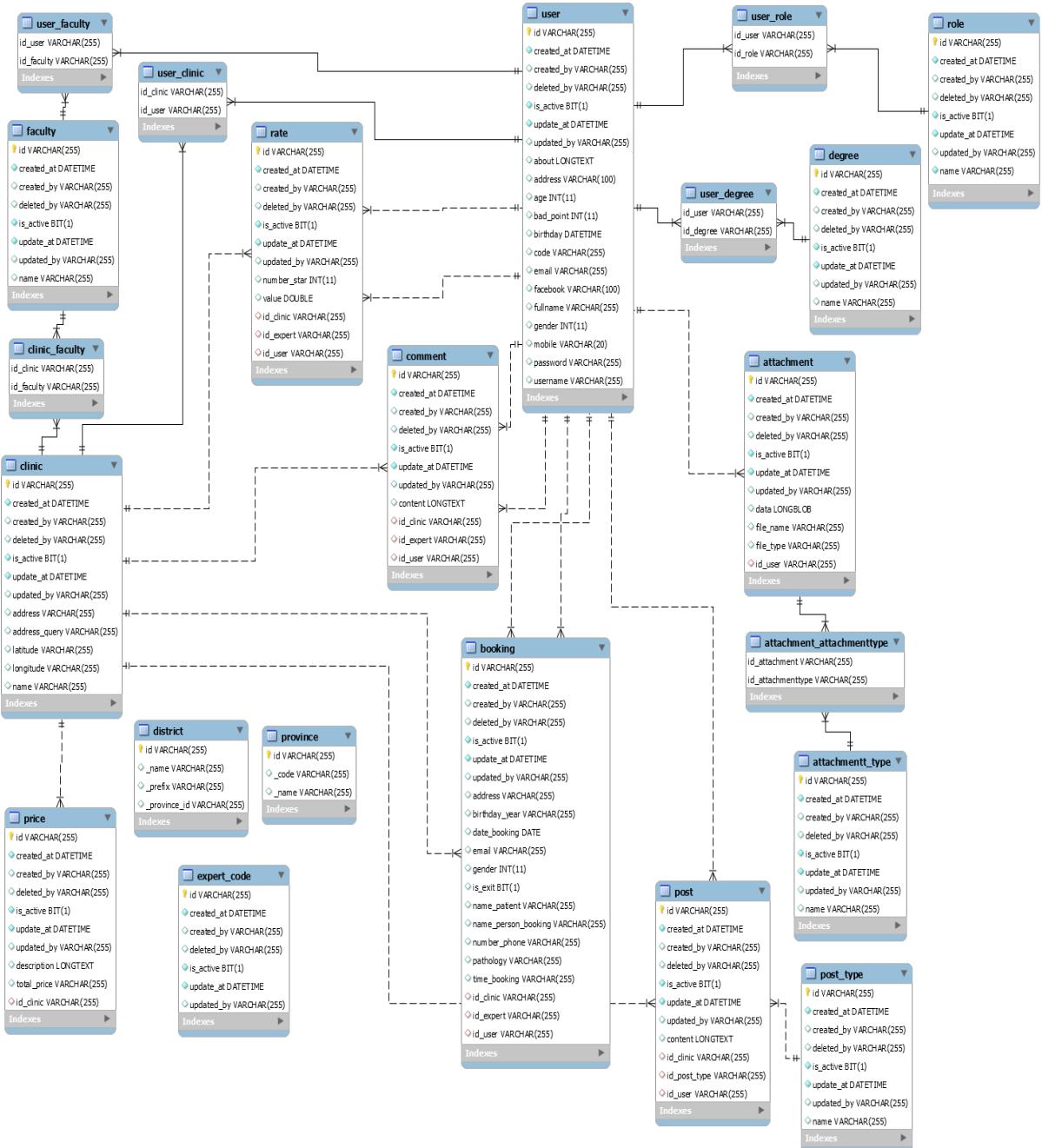
- **Province:** là bảng chứa tên các tỉnh, thành phố của Việt Nam.

Bảng 2.32: Bảng dữ liệu distrist

Distrist				
Tên cột	Kiểu dữ liệu	Độ dài tối đa	Ghi chú	Giải thích
<b>Id</b>	Varchar	255	Khóa chính, Duy nhất	Khóa chính của bảng
created_at	Date			Ngày khởi tạo
update_at	Date			Ngày cập nhật
created_by	Varchar	255		Chứa khóa chính của người khởi tạo.
update_by	Varchar	255		Chứa khóa chính của người cập nhật.
delete_by	Varchar	255		Chứa khóa chính của người xóa.
_name	Varchar	255		Tên của quận hoặc huyện của Việt Nam
_prefix	Varchar	255		Ký hiệu là quận hay huyện
_province_id	Varchar	255	Khóa ngoại, với bảng Province	Mã của bảng Pr

### 2.8.2. Sơ đồ quan hệ cơ sở dữ liệu

Mỗi quan hệ giữa các bảng dữ liệu được thể hiện dưới đây:



Hình 2.21: Sơ đồ quan hệ cơ sở dữ liệu

## CHƯƠNG 3: TRIỂN KHAI VÀ CHẠY THỬ NGHIỆM

### 3.1. MÔI TRƯỜNG TRIỂN KHAI

#### 3.1.1. Quản lý mã nguồn

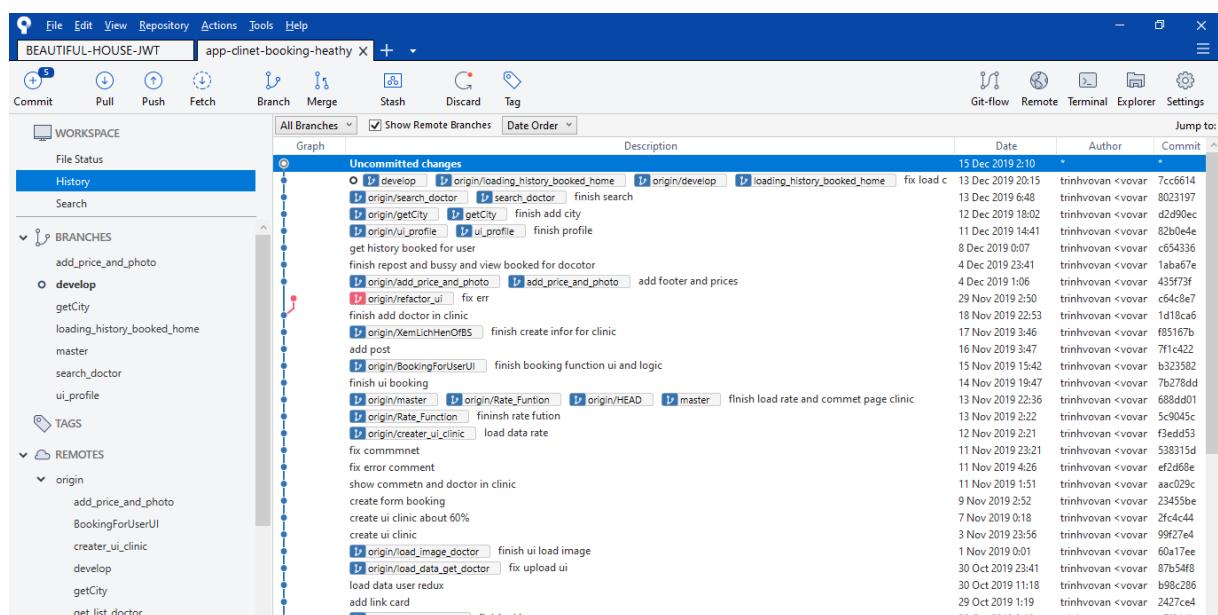
Trong quy trình phát triển phần mềm chuyên nghiệp, việc quản lý các phiên bản và lưu trữ mã nguồn là rất quan trọng hỗ trợ đắc lực cho lập trình viên. Trong đề tài này em đã sử dụng Sourcetree để lưu trữ mã nguồn kết hợp cùng với Git để quản lý các phiên bản.

##### ➤ Git

Git là một hệ thống VCS (Version Control System) dùng để quản lý và kiểm tra các phiên bản source code khác nhau trong quá trình phát triển. Trên Git, có thể lưu trạng thái của file khi có nhu cầu dưới dạng lịch sử cập nhật.

##### ➤ SourceTree

Sourcetree là phần mềm giúp phát triển những dự án lập trình có sử dụng hệ thống kiểm soát Git revision.



Hình 3.1: Lưu trữ mã nguồn trên sourcetree

#### 3.1.2. Môi trường cài đặt

- Môi trường máy để chạy chương trình cả font-end và cả backend cần đáp ứng:
  - JDK (Java Development Kit) : là một bộ phần mềm cung cấp môi trường phát triển ứng dụng viết bằng ngôn ngữ Java. JDK bao gồm cả Java Runtime Environment giúp lập trình viên có thể chạy thử để kiểm tra ứng dụng trong quá trình phát triển ứng dụng.

- Npm: là chương trình quản lý thư viện ngầm định trong môi trường Javascript Node.js hiện tại đang dùng version 6.10.2



Hình 3.2: Trình quản lý thư viện npm

- ❖ Back-end là web API được xây dựng bằng ngôn ngữ Java sử dụng Spring framework



Hình 3.3: Ngôn ngữ back-end sử dụng

- ❖ Front-end là web application được sử dụng HTML, CSS, JavaScript với thư viện ReactJS



Hình 3.4: Ngôn ngữ font-end sử dụng

- ❖ Hệ quản trị cơ sở dữ liệu là MySQL



Hình 3.5: Hệ quản trị cơ sở dữ liệu

- ❖ Triển khai hệ thống trên heroku:

Heroku là nền tảng đám mây cho phép các lập trình viên xây dựng, triển khai, quản lý và mở rộng ứng dụng (PaaS – Platform as a service). Nó rất linh hoạt và dễ sử dụng, cung cấp cho một con đường đơn giản nhất để đưa sản phẩm tiếp cận người dùng.



Hình 3.6: Môi trường heroku deploy bản production

### 3.2. KẾT QUẢ DEMO VÀ HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

Source code được lưu trữ ở source tree, để chạy chương trình hoàn chỉnh chúng ta thực hiện theo các bước:

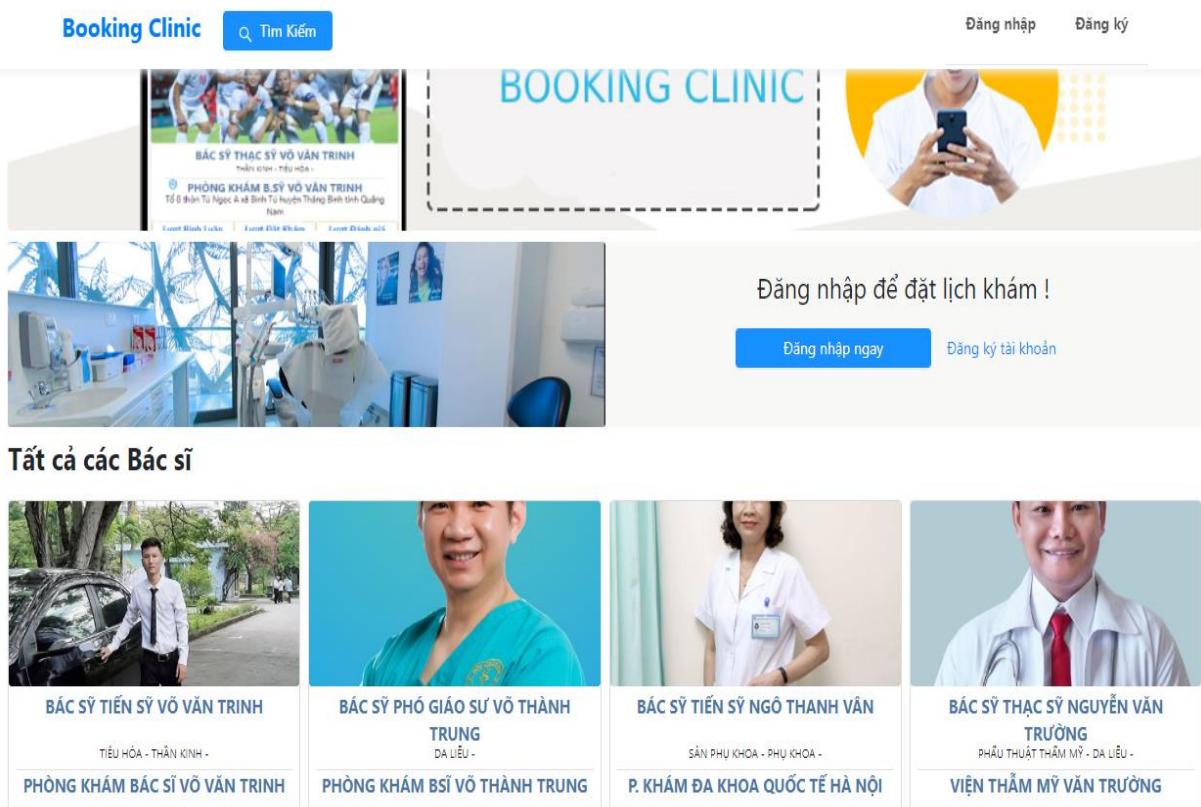
- Backend: Sử dụng lệnh **git clone** để tải toàn bộ mã chương trình về máy. Tiếp theo mở Xamp để khởi động cơ sở dữ liệu. Tiếp theo mở trình soạn thảo mã nguồn Java IDE eclipse, mở thư mục chứa mã nguồn và nhấn nút start trên IDE eclipse để chạy ứng dụng bên server.
- Frontend: Sử dụng lệnh **git clone** để tải toàn bộ mã chương trình web application về máy. Tiếp theo ở thư mục gốc của mã nguồn chạy lệnh **npm install** để cài đặt những gói, thư viện liên quan cần thiết. Cuối cùng chạy lệnh **npm start** để chạy chương trình lên.

Sau khi hoàn thành các bước chuẩn bị trên chúng ta sẽ chạy chương trình cụ thể và sử dụng ứng dụng.

### 3.2.1 Màn hình chức năng của khách

#### ❖ Đăng ký

Ở phần header tại màn hình chính của trang web sẽ xuất hiện 2 liên kết là đăng ký và đăng nhập. Với trường hợp chưa có tài khoản chúng ta sẽ nhấp vào liên kết đăng ký.



Hình 3.7 Màn hình chính của hệ thống

Sau đó hệ thống sẽ hiện lên màn hình đăng ký thành viên.

The screenshot shows the 'Sign Up' registration form. The form fields are as follows:

- \* Họ và Tên: (Last Name and First Name) - Placeholder: Hãy nhập tên đầy đủ của bạn!
- \* Username: - Placeholder: Hãy nhập tài khoản đăng nhập của bạn!
- \* Số điện thoại: - Placeholder: Hãy nhập số điện thoại của bạn!
- \* E-mail: - Placeholder: Hãy nhập email của bạn !
- \* Password: - Placeholder: Hãy nhập mật khẩu của bạn !
- \* Confirm Password: - Placeholder: Nhập lại mật khẩu !

At the bottom of the form is a large blue 'Đăng ký' (Register) button.

Hình 3.8: Màn hình đăng ký dành cho khách

### ❖ Đăng nhập

Sau khi đã đăng ký tài khoản và xác thực hợp lệ thì người dùng dùng đăng nhập vào hệ thống bằng tài khoản đó để thực hiện các chức năng của mình.

Hình 3.9: Màn hình đăng nhập hệ thống

### ❖ Xem danh sách các bác sĩ

Màn hình này cung cấp tất cả các bác sĩ của ứng dụng giúp các tác nhân trong hệ thống có thể xem thông tin tất cả các bác sĩ cũng như bước tiền đề để thực hiện các chức năng chính.

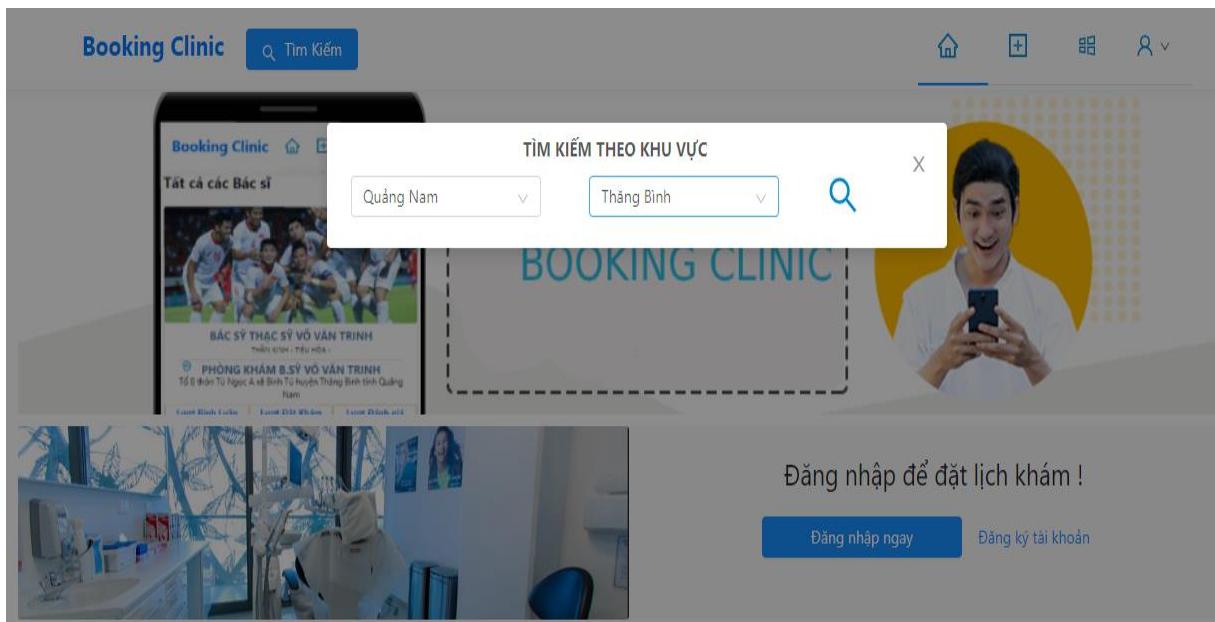
<b>BÁC SỸ TIẾN SỸ VÕ VĂN TRINH</b> TIỀU HÓA - THẨM KINH - <b>PHÒNG KHÁM BÁC SĨ VÕ VĂN TRINH</b>	<b>BÁC SỸ PHÓ GIÁO SƯ VÕ THÀNH TRUNG</b> DA LIỄU - <b>PHÒNG KHÁM BSÍ VÕ THÀNH TRUNG</b>	<b>BÁC SỸ TIẾN SỸ NGÔ THANH VÂN</b> SẢN PHỤ KHOA - PHỤ KHOA - <b>P. KHÁM ĐA KHOA QUỐC TẾ HÀ NỘI</b>	<b>BÁC SỸ THẠC SỸ NGUYỄN VĂN TRƯỜNG</b> PHẪU THUẬT THẨM MỸ - DA LIỄU - <b>VIỆN THẨM MỸ VĂN TRƯỜNG</b>
---	---	---	---

Hình 3.10: Màn hình chính chứa tất cả danh sách các bác sĩ

### ❖ Tìm kiếm bác sĩ theo khu vực

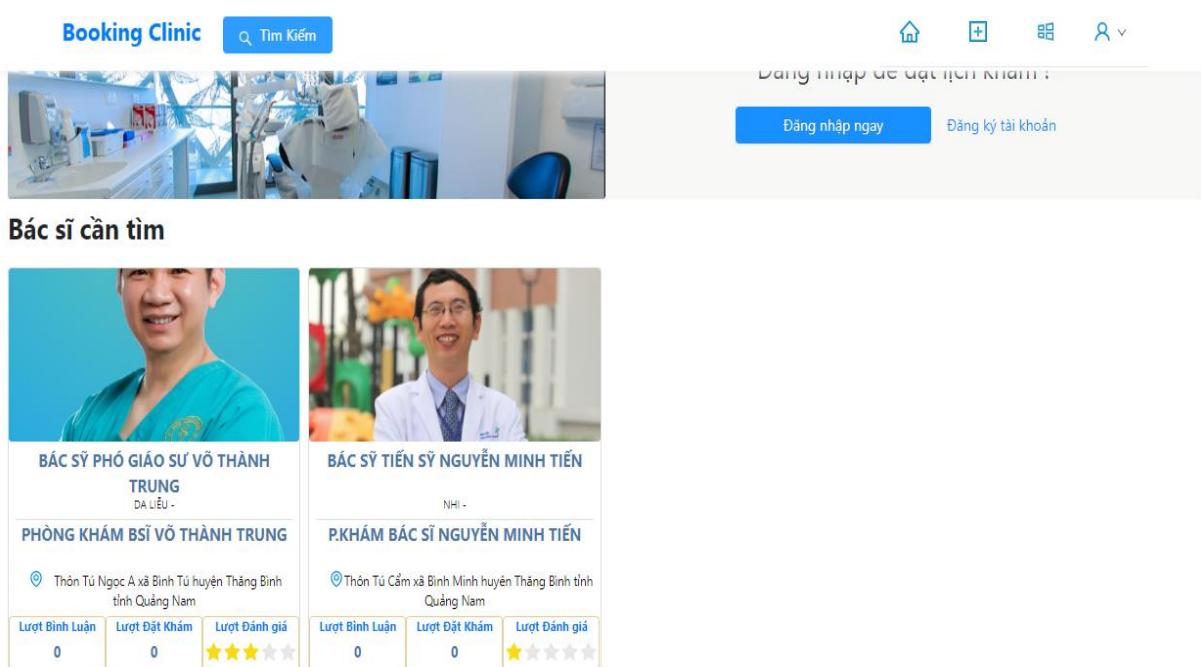
Ở phần header tại màn hình chính của trang web sẽ xuất hiện nút tìm kiếm, khi chọn vào nút đó thì sẽ hiện ra form để chọn khu vực phù hợp với thành viên để tìm kiếm bác sĩ.

Màn hình trước khi tìm kiếm.



Hình 3.11: Màn hình trước khi tìm kiếm

Màn hình kết quả sau khi tìm kiếm.



Hình 3.12: Màn hình kết quả sau khi tìm kiếm

### 3.2.2 Màn hình chức năng của thành viên

Thành viên cũng có đầy đủ các chức năng của khách và có thêm những chức năng riêng được hệ thống cung cấp.

#### ❖ Đăng ký trở thành bác sĩ

Sau khi đăng nhập thành công *header* sẽ xuất hiện các icon mỗi icon là một chức năng. Trong đó có icon đăng ký thành bác sĩ. Khi chọn icon này form đăng ký sẽ xuất hiện. Để đăng ký được thành viên phải liên hệ đến quản trị viên để được cấp mã code.

Màn hình trước khi đăng ký.

**Đăng Ký Trở Thành Bác Sỹ**

Họ và Tên : NGUYỄN ĐỨC HÀ

Ngày sinh: 16/12/2019 Giới tính: NAM

Học Hàm - Học vị : THẠC SỸ X BÁC SỸ X

Ảnh đại diện

Mã code để được xác nhận : 402881e46eff700b016eff8b6e1b000b

Khoa : TIM MẠCH X NHI X Y HỌC GIA ĐÌNH X

Địa chỉ : Đường 105 Thành Thái, Phường 14, Quận 10 thành phố Hồ Chí Minh

Hình 3.13: Màn hình đăng ký phòng khám

#### ❖ Thực hiện bình luận cho bác sĩ

Sau khi đăng ký thành viên thành công, thành viên có thể bình luận những bác sĩ mình yêu thích hoặc đã đặt khám. Trong mục bình luận bác sĩ, phải thực hiện chức năng tìm kiếm bác sĩ và lựa chọn bác sĩ cần bình luận, sẽ có nút bình luận để thành viên có thể thực hiện chức năng bình luận.

## Xây dựng ứng dụng đặt chỗ khám bệnh

Màn hình trước khi bình luận.

**BÁC SỸ TIẾN SỸ VÕ VĂN TRINH**

Giáo sư, Tiến sĩ, Bác sĩ chuyên ngành Tiêu hóa - Gan mật. Nguyên Giám đốc bệnh viện Đại Học Y Hà Nội. Nguyên Tổng thư ký hội khoa học tiêu hóa Việt Nam. Nguyên Trưởng khoa tiêu hóa Bệnh viện Bạch Mai. Nguyên Giám đốc Trung tâm Nghiên cứu Bệnh viện Đại học Y Hà Nội. Tốt nghiệp Đại học Y Hà Nội (1983). Giáo sư nhận điều trị các bệnh lý khó về Tiêu hóa, hậu môn, trực tràng, gan mật. Bác sĩ chuyên khoa Nội khám và kiểm tra tình trạng chung của người bệnh

Tầng 10, Tháp VCCI, Số 9 Đào Duy Anh, quận Đống Đa thành phố Hà Nội

**Hướng dẫn đặt lịch**

Lượt đánh giá  
★★★★★

Đánh giá : ★★★★★ Bình luận

**Liên kết**

- Facebook
- Youtube
- Google+

**Bản di động**

- Download ứng dụng App Store
- Download ứng dụng Google Play

**Hỗ trợ khách hàng**

- Trung tâm trợ giúp
- An toàn mua bán
- Quy định cần biết
- Liên hệ hỗ trợ

**Chứng nhận**

ĐÃ ĐĂNG KÝ  
BỘ CÔNG THƯƠNG

CÔNG TY TNHH BOOKING CLINIC - Địa chỉ: Phòng 1808, Tầng 18, Mê Linh Point Tower, 02 Ngõ Đức Kế, Phường Bến Nghé, Quận 1, TP Hồ Chí Minh

Hình 3.14: Màn hình trước khi bình luận

Màn hình kết quả sau khi bình luận

**Bình Luận**

Nhân viên nhiệt tình. Bác sĩ tư vấn thiếu nhiệt tình, tác phong vội vàng

Nhân viên nhiệt tình. Bác sĩ tư vấn thiếu nhiệt tình, tác phong vội vàng

Bác sĩ bận nên chưa trao đổi được nhiều.

Bác sĩ chẩn đoán liệt dây thần kinh số VII. Đang điều trị theo đơn của bác sĩ và tình trạng có tiến triển tốt hơn.

Bác sỹ vã trợ lý rất nhiệt tình tư vấn, kết quả điều trị thì chưa rõ, tôi hài lòng với sự nhiệt tình của đội ngũ bác sỹ

Bác sĩ tư vấn nhiều, quan tâm bệnh nhân. Bác sĩ cho tôi số điện thoại để liên hệ trong quá trình sử dụng đơn thuốc. Không phải chờ đợi. Thủ tục nhanh gọn, thoải mái khi đi khám bệnh. Sự hỗ trợ nhân viên bệnh viện rất nhiệt tình và tận tình.

Bác sĩ tư vấn nhiệt tình, tuy nhiên tôi quên mang phim chụp đến nên có 1 số vấn đề vẫn chưa trao đổi được nhiều. Nhân viên nhiệt tình, chu đáo

Xem thêm 1 bình luận

Bác sĩ tốt thuốc tốt, trang thiết bị tốt

Đăng

**BÌNH LUẬN**

**Cá Nhân**

**BÉNH:** 2019-12-13

**No Data**

**Hỗ trợ khách hàng**

**Chứng nhận**

ĐÃ ĐĂNG KÝ  
BỘ CÔNG THƯƠNG

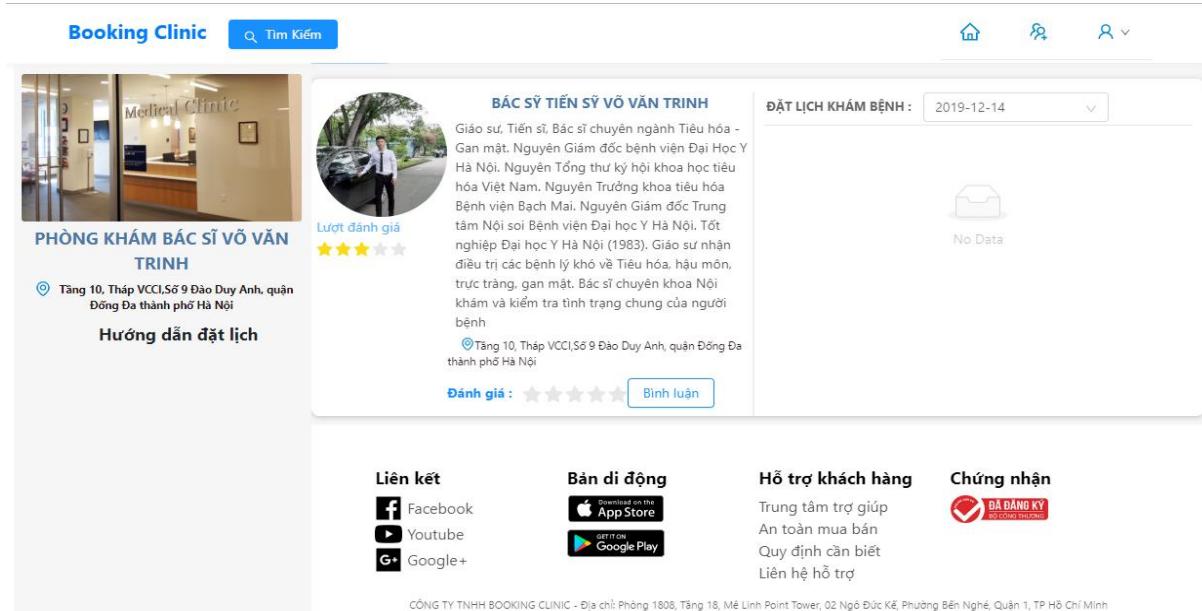
CÔNG TY TNHH BOOKING CLINIC - Địa chỉ: Phòng 1808, Tầng 18, Mê Linh Point Tower, 02 Ngõ Đức Kế, Phường Bến Nghé, Quận 1, TP Hồ Chí Minh

Hình 3.15: Màn hình kết quả sau khi bình luận

### ❖ Thực hiện đánh giá cho bác sĩ

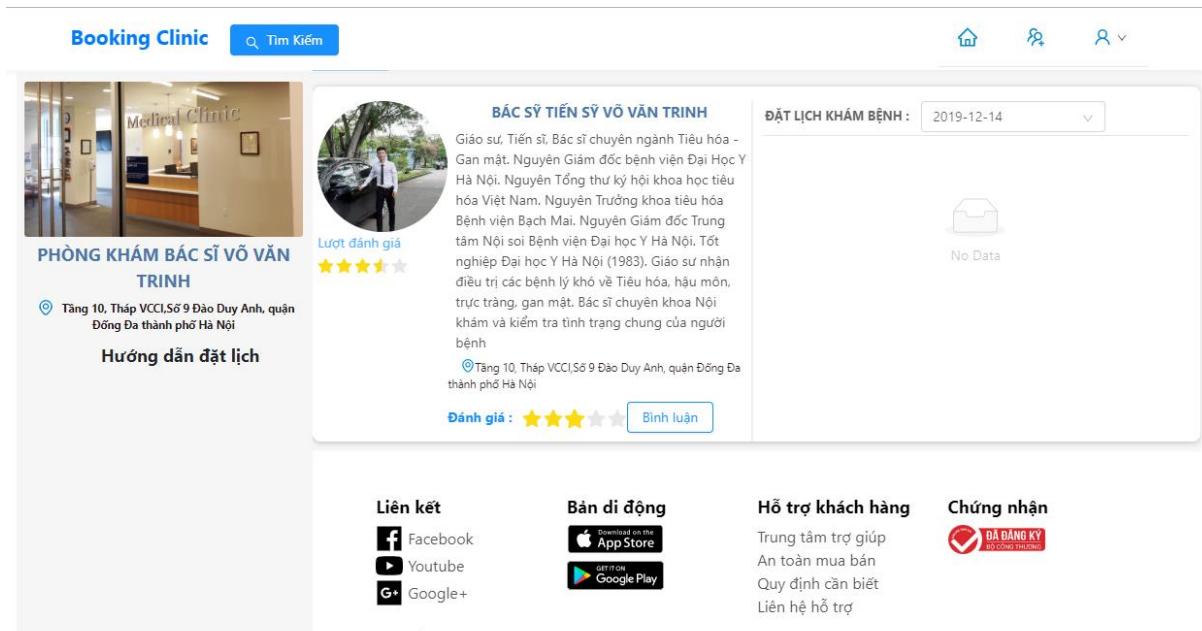
Trong mục đánh giá bác sĩ, phải thực hiện chức năng tìm kiếm bác sĩ và lựa chọn bác sĩ cần đánh giá, màn hình sẽ có 5 hình ngôi sao để thành viên chọn độ yêu thích bác sĩ phụ thuộc vào số lượng sao thành viên cho tối đa 5 sao và tối thiểu 1 sao. Càng nhiều sao thì bác sĩ càng chất lượng.

Màn hình trước khi đánh giá.



Hình 3.16: Màn hình trước khi đánh giá

Màn hình kết quả sau khi đánh giá.



Hình 3.17: Màn hình kết quả sau khi đánh giá

### ❖ Chức năng đặt lịch khám bệnh

Trong mục đặt lịch khám bệnh, phải thực hiện chức năng tìm kiếm bác sĩ và lựa chọn bác sĩ cần đặt lịch, sẽ có danh sách các giờ để bạn đặt lịch do bác sĩ tạo ra. Lựa chọn ngày, giờ phù hợp và điền đầy đủ thông tin trong form để đặt lịch khám

Màn hình chính trước khi đặt lịch.

The screenshot shows the main booking interface for 'Booking Clinic'. At the top, there's a search bar with 'Tim Kiếm' (Search) and icons for home, user, and search. Below the search bar is a photo of a medical clinic interior. To the right of the photo is a doctor's profile: 'BÁC SỸ TIẾN SỸ VÕ VĂN TRINH'. The profile includes a photo of the doctor standing outdoors, a star rating of 4.5, and a link to his reviews ('Lượt đánh giá'). Below the profile is a detailed biography of Dr. Vo Van Trinh, mentioning his education at the University of Medicine and Pharmacy, Hanoi, and his specialties in endocrinology and metabolism. It also lists his office address at Tháp VCCI, Số 9 Đào Duy Anh, Quận Đống Đa, Hà Nội. A booking calendar for December 14, 2019, shows available slots from 07:00 to 10:00. At the bottom of the profile section is a feedback button ('Đánh giá') and a comment box ('Bình luận'). Below the main content are links for 'Liên kết' (Facebook), 'Bản đồ di động' (Google Maps), 'Hỗ trợ khách hàng' (Customer Support), and 'Chứng nhận' (Certifications), including a 'ĐÃ ĐĂNG KÝ' (Registered) badge.

Hình 3.18: Màn hình chính trước khi đặt lịch

Màn hình form để đặt lịch.

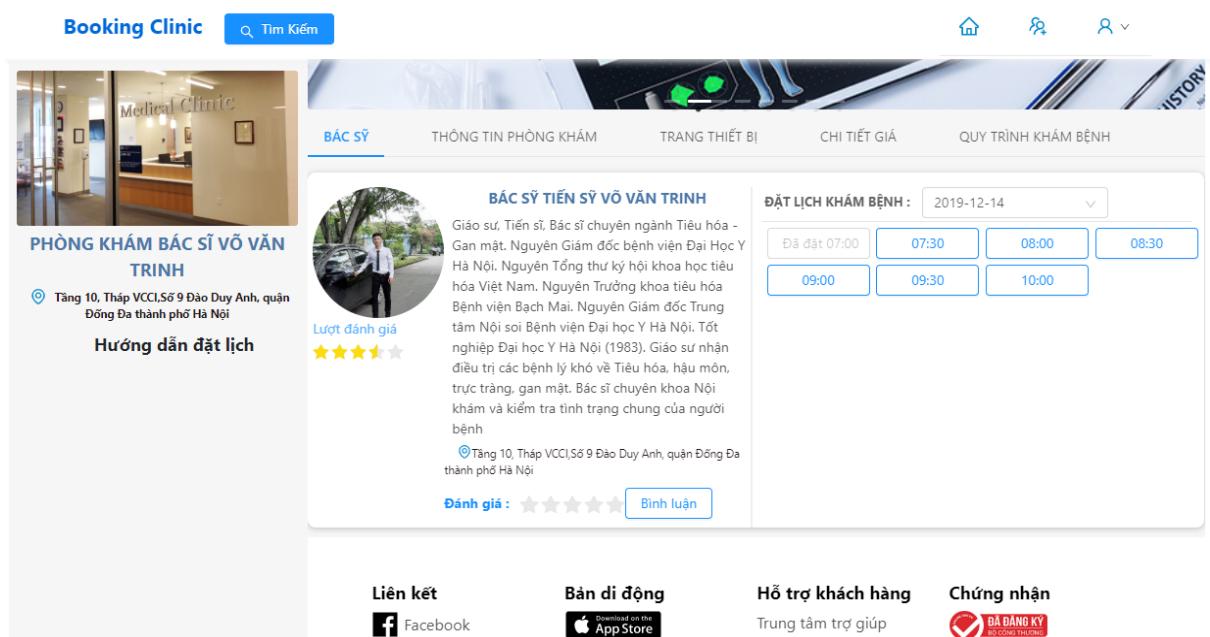
The screenshot shows the booking form interface. At the top left is the 'Booking Clinic' logo and search bar. The main area features a doctor's profile and appointment details: 'PHÒNG KHÁM BÁC SĨ VÕ VĂN TRINH', 'NGÀY ĐẶT LỊCH : 2019-12-14', and 'THỜI GIAN : 07:00'. Below this are sections for 'THÔNG TIN NGƯỜI ĐẶT LỊCH' (Patient Information) and 'THÔNG TIN NGƯỜI BỆNH' (Patient Information). The 'THÔNG TIN NGƯỜI ĐẶT LỊCH' section contains fields for 'Tên Người Đặt Khám', 'Email Người Đặt Khám', and 'Số điện thoại'. The 'THÔNG TIN NGƯỜI BỆNH' section contains fields for 'Tên Bệnh Nhân', 'Ngày sinh Bệnh Nhân', 'Giới tính Bệnh Nhân', and 'Địa chỉ'. On the right side of the form, there's a preview of the appointment schedule for December 14, 2019, with time slots from 07:30 to 10:00. At the bottom right are 'Hỗ trợ khách hàng' (Customer Support) and 'Chứng nhận' (Certifications) links, including a 'ĐÃ ĐĂNG KÝ' (Registered) badge.

Hình 3.19: Màn hình form đăng ký khi đặt lịch

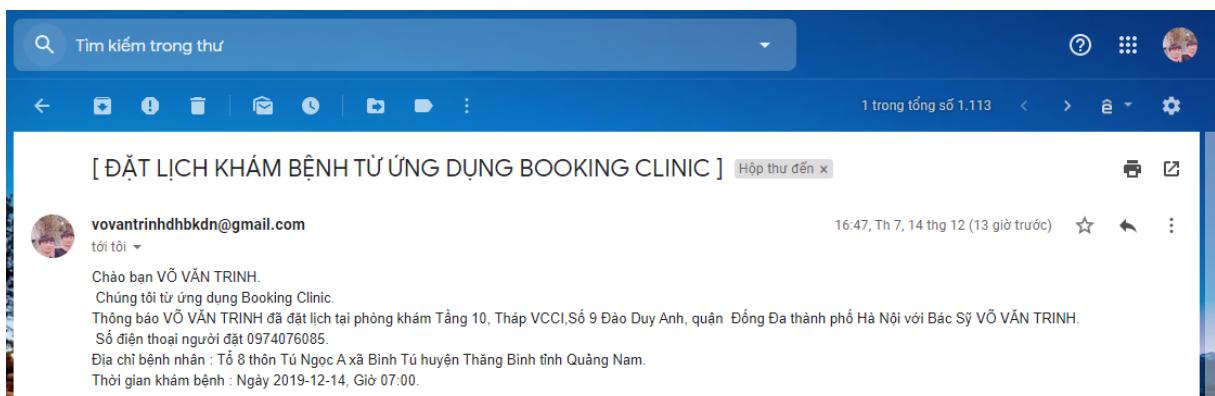
## Xây dựng ứng dụng đặt chỗ khám bệnh

Kết quả sau khi đặt lịch.

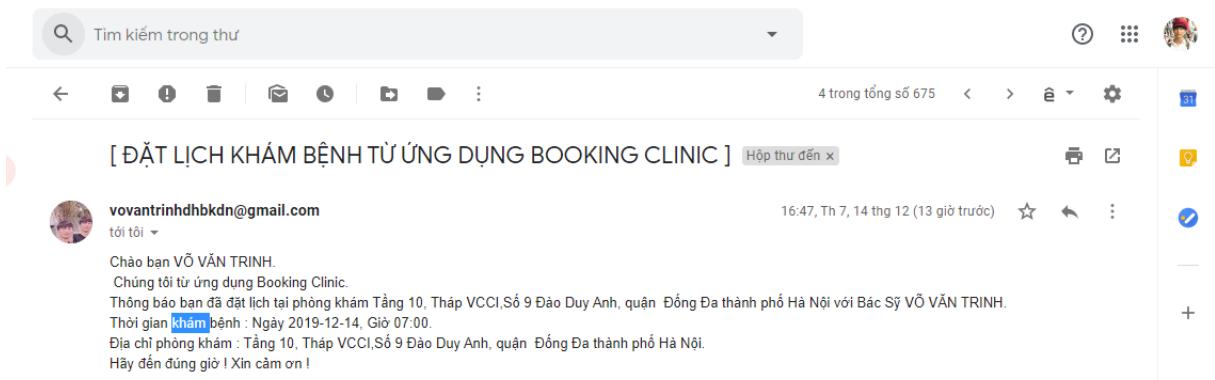
Hệ thống sẽ gửi email với nội dung thông báo đến bác sĩ và người đặt lịch thông qua email nhắc nhở bác sĩ và người đặt lịch



Hình 3.20: Màn hình chính sau khi đã được đặt lịch



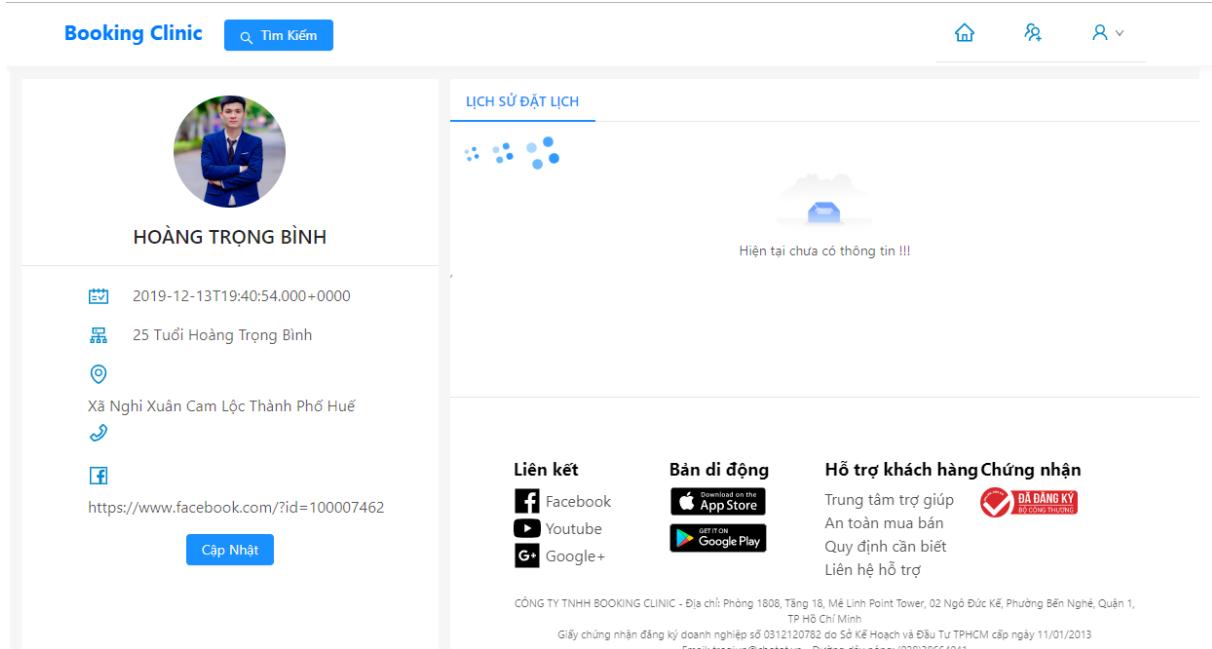
Hình 3.21: Màn hình thông báo đến bác sĩ khi có người đặt lịch



Hình 3.22: Màn hình thông báo đến người đã đặt lịch

### ❖ Chức năng chỉnh sửa thông tin cá nhân

Ở phần header tại màn hình chính của trang web sẽ xuất hiện icon người dùng, khi chọn nó sẽ hiện mục profile. Khi chọn màn hình trang profile xuất hiện, tại màn hình này người dùng có thể thay đổi thông tin cá nhân, cập nhật ảnh đại diện. Ngoài ra nó còn hiện lịch sử mà bạn đã đặt lịch khám.



Hình 3.23: Màn hình chính trang profile

Màn hình form cập nhật thông tin cá nhân

This screenshot shows the 'Cập Nhật' (Update) form for the same user. It includes fields for birth date (16/12/2019), profile picture, address (105 Thành Thái, Phường 14, Quận 10 (Đối diện BV 115)), phone number (055585565), Facebook link (<https://www.facebook.com/trinhhung20>), and a bio section. The bio field contains redacted text about a doctor's visit and treatment. A blue 'Cập Nhật' button is at the bottom. The right side of the screen shows the same support section as in the previous screenshot.

Hình 3.24: Màn hình form cập nhật thông tin cá nhân

### 3.2.3 Màn hình chức năng của bác sĩ

Bác sĩ cũng có đầy đủ các chức năng của khách, thành viên và có thêm những chức năng riêng được hệ thống cung cấp.

#### ❖ Đăng ký phòng khám

Sau khi đăng ký bác sĩ thành công, header có sự thay đổi trong đó có icon đăng ký phòng khám. Chọn icon này form đăng ký phòng khám sẽ xuất hiện.

Màn hình đăng ký phòng khám.

Tên phòng khám :

PHÒNG KHÁM B.SĨ NGUYỄN ĐỨC HÀ

Ảnh phòng khám :

Logo phòng khám :

HỒNG TÂM  
Cardiology Clinic

Tỉnh:

Hà Nội

Quận / Huyện:

BA ĐÌNH

Hình 3.25: Màn hình đăng ký phòng khám

Màn hình kết quả sau khi đăng ký phòng khám thành công.

Đường số 105 Thành Thái, Phường 14, Quận 10 thành phố Hồ Chí Minh

THÊM THÔNG TIN PHÒNG KHÁM

THÊM GIÁ PHÒNG KHÁM

THÊM BÁC SỸ VÀO PHÒNG KHÁM

Hướng dẫn đặt lịch

BÁC SỸ

THÔNG TIN PHÒNG KHÁM

TRANG THIẾT BỊ

CHI TIẾT GIÁ

QUY TRÌNH KHÁM BỆNH

BÁC SỸ THẠC SỸ NGUYỄN ĐỨC HÀ

Phó Giáo sư Nguyễn Văn Liệu có gần 30 năm kinh nghiệm công tác tại Khoa Thần kinh - Bệnh viện Bạch Mai, kiêm giảng dạy tại Đại học Y khoa Hà Nội. Phó Giáo sư đã trên 30 bài báo và nhiều tài liệu sau đại học đã được xuất bản. Ông không ngừng cập nhật nâng cao kiến thức thông qua các chương trình đào tạo, hội thảo trong nước và quốc tế. Khám các bệnh Đau nhức đầu, Mắt ngủ kéo dài, Suy nhược thần kinh. Đau thần kinh chi dưới, đau thắt lưng cùng

ĐẶT LỊCH KHÁM BỆNH : 2019-12-17

Lượt đánh giá

Đánh giá : ★★★★★

Bình luận

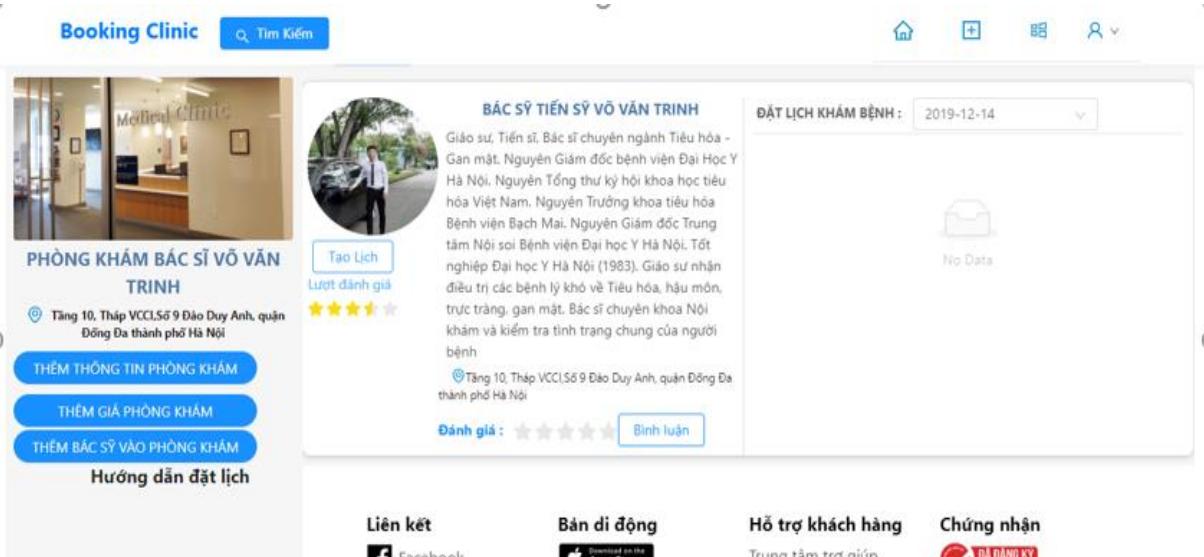
No Data

Hình 3.26: Màn hình kết quả sau đăng ký phòng khám

### ❖ Chức năng tạo lịch khám bệnh của bác sĩ

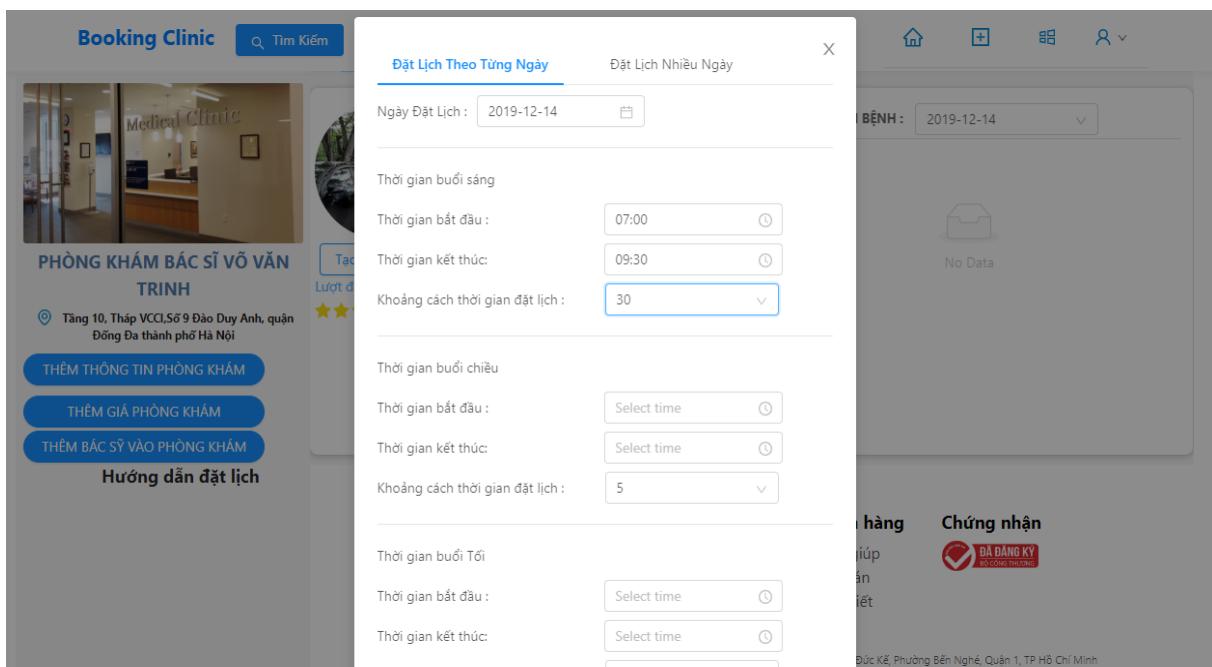
Bác sĩ lựa chọn phòng khám của mình trên header. Khi nó sẽ xuất hiện màn hình phòng khám và bác sĩ đứng đầu danh sách phòng khám chính là bác sĩ đang ở trong hệ thống. Sẽ có button tạo lịch, khi nhấn tạo lịch thì sẽ hiện form để chon ngày và giờ để tạo lịch.

Màn hình chính trước khi tạo lịch.



Hình 3.27: Màn hình chính trước khi tạo lịch

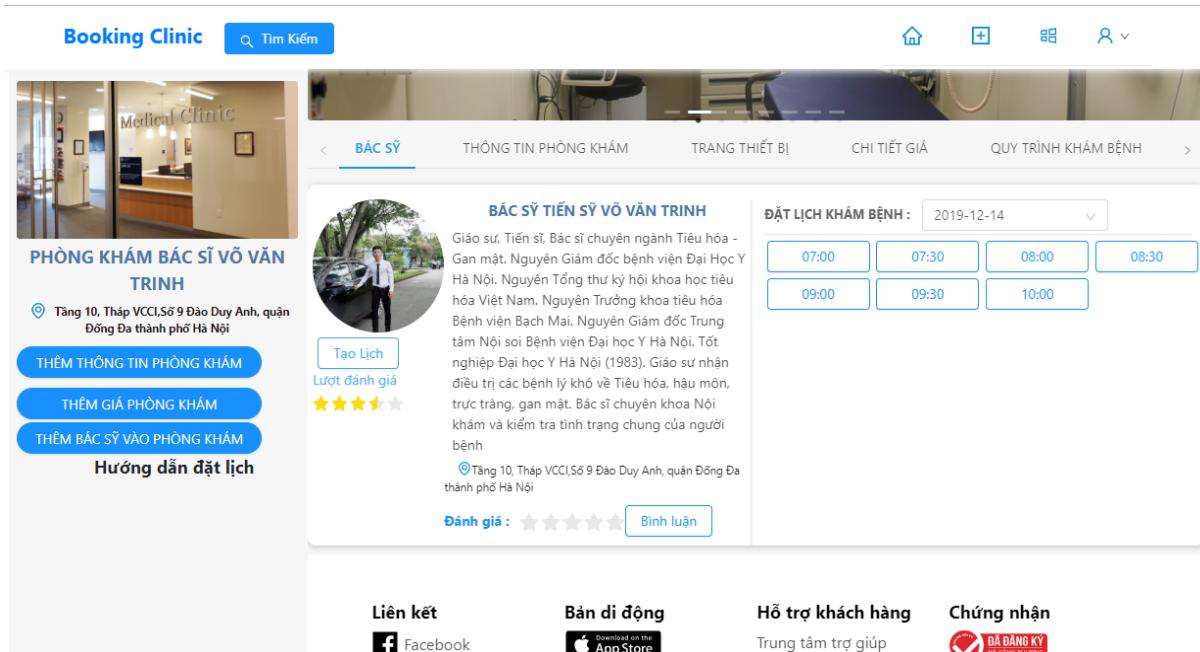
Màn hình form đăng ký lịch khám.



Hình 3.28: Màn hình form khởi tạo lịch

## Xây dựng ứng dụng đặt chỗ khám bệnh

Màn hình kết quả sau khi tạo lịch.

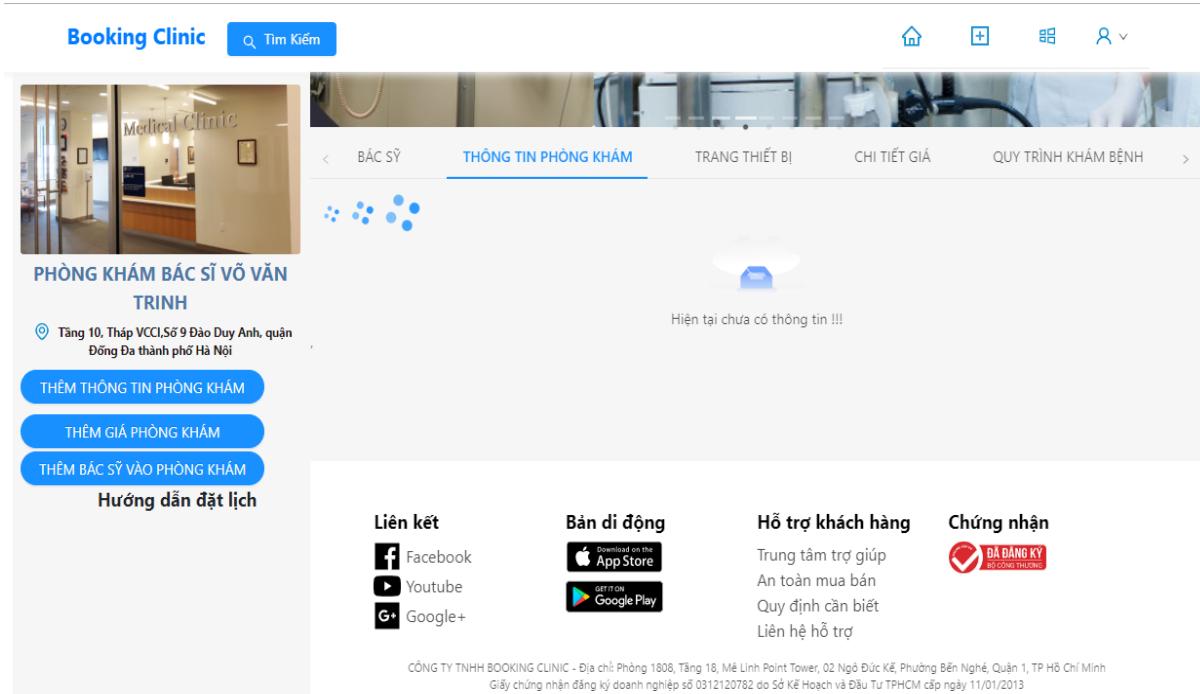


Hình 3.29: Màn hình kết quả sau khi đặt lịch

### ❖ Chức năng tạo bài viết cho phòng khám của bác sĩ

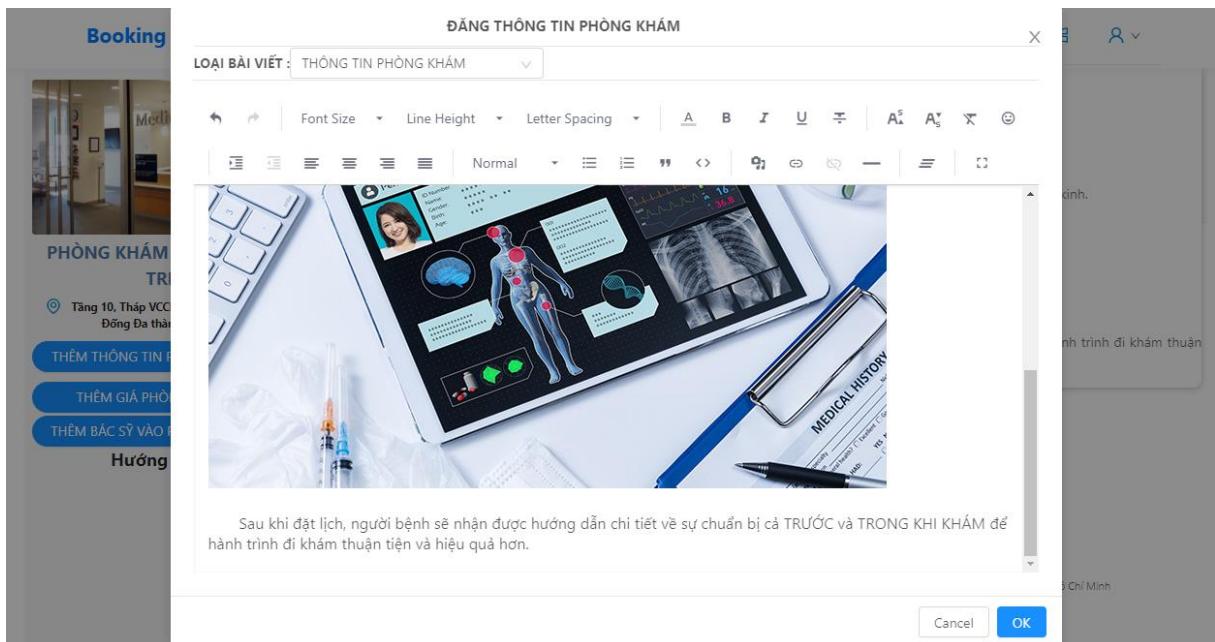
Bác sĩ lựa chọn phòng khám của mình trên header. Khi nó sẽ xuất hiện màn hình phòng khám, sẽ có button tạo bài viết, sau khi chọn sẽ hiện form để nhập thông tin và đăng bài viết cho phòng khám.

Màn hình trước khi đăng thông tin.



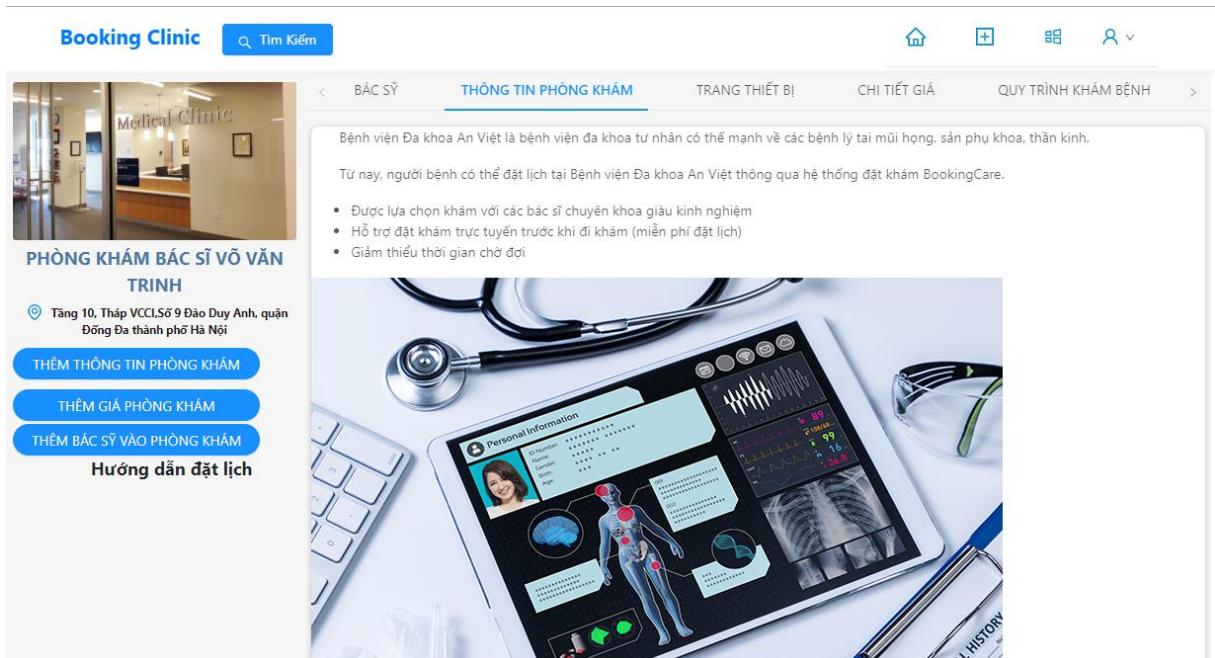
Hình 3.30: Màn hình trước khi khởi tạo bài viết

Màn hình form tạo bài viết.



Hình 3.31: Màn form để khởi tạo bài viết

Màn hình kết quả sau khi tạo bài viết.

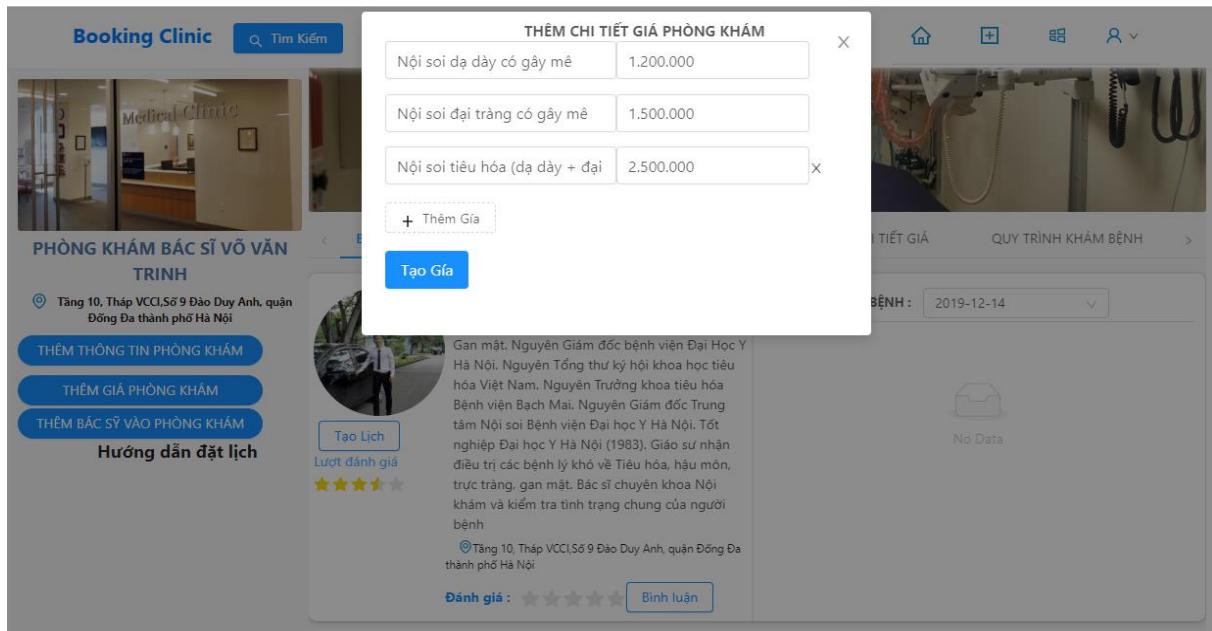


Hình 3.32: Màn hình kết quả sau khi tạo thành công bài viết

#### ❖ Chức năng tạo danh sách giá cho phòng khám cho phòng khám của bác sĩ

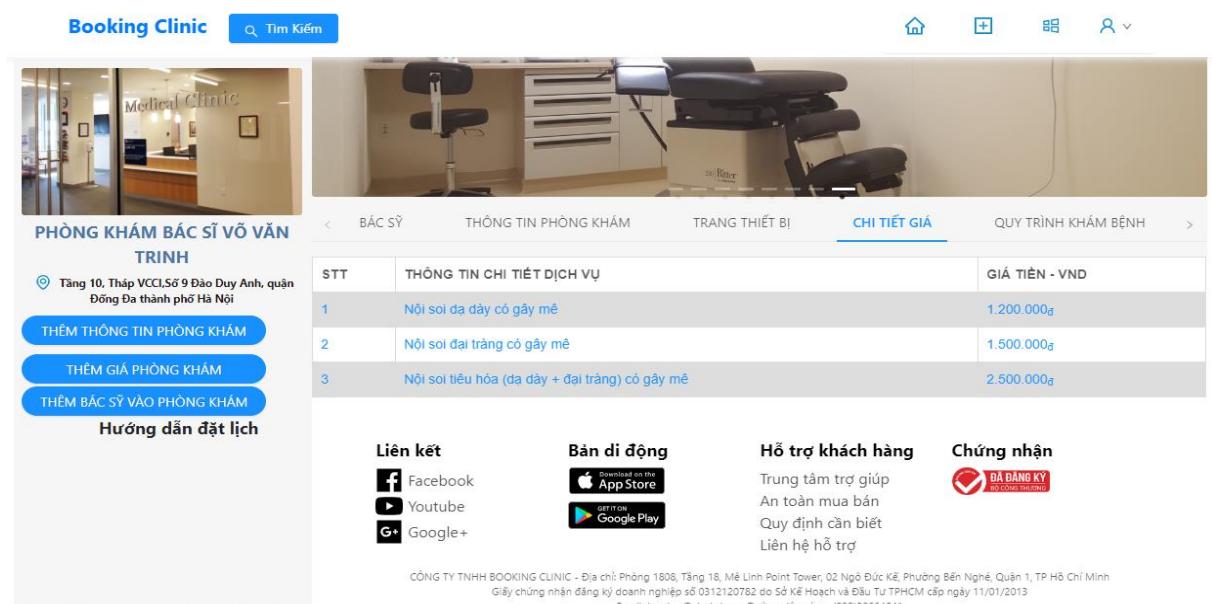
Bác sĩ lựa chọn phòng khám của mình trên header. Khi nó sẽ xuất hiện màn hình phòng khám, sẽ có button thêm giá phòng khám, sau khi chọn sẽ hiện form để nhập thông tin chi tiết giá.

Màn hình chính trước khi thêm.



Hình 3.33: Màn hình form tạo danh sách giá

Kết quả sau khi thêm danh sách giá cho phòng khám



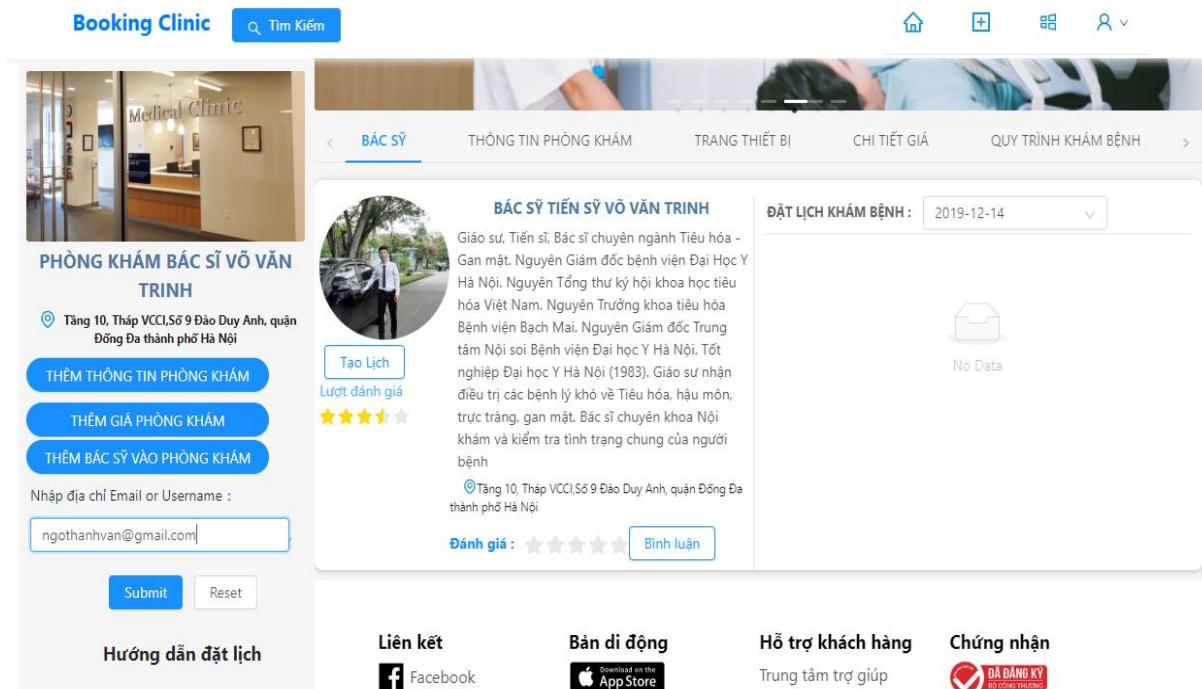
Hình 3.34: Màn hình danh sách giá của phòng khám

#### ❖ Thêm bác sĩ vào phòng khám

Bác sĩ tạo phòng khám mới có thể thêm bác sĩ vào phòng khám của mình. Bác sĩ lựa chọn phòng khám của mình trên header. Khi nó sẽ xuất hiện màn hình phòng khám, nếu bác sĩ đang sử dụng là chủ phòng khám, người đầu tiên tạo phòng khám thì sẽ hiện button thêm bác sĩ. Sau đó, form thêm bác sĩ sẽ hiện ra, nhập email bác sĩ cần thêm để thêm vào phòng khám.

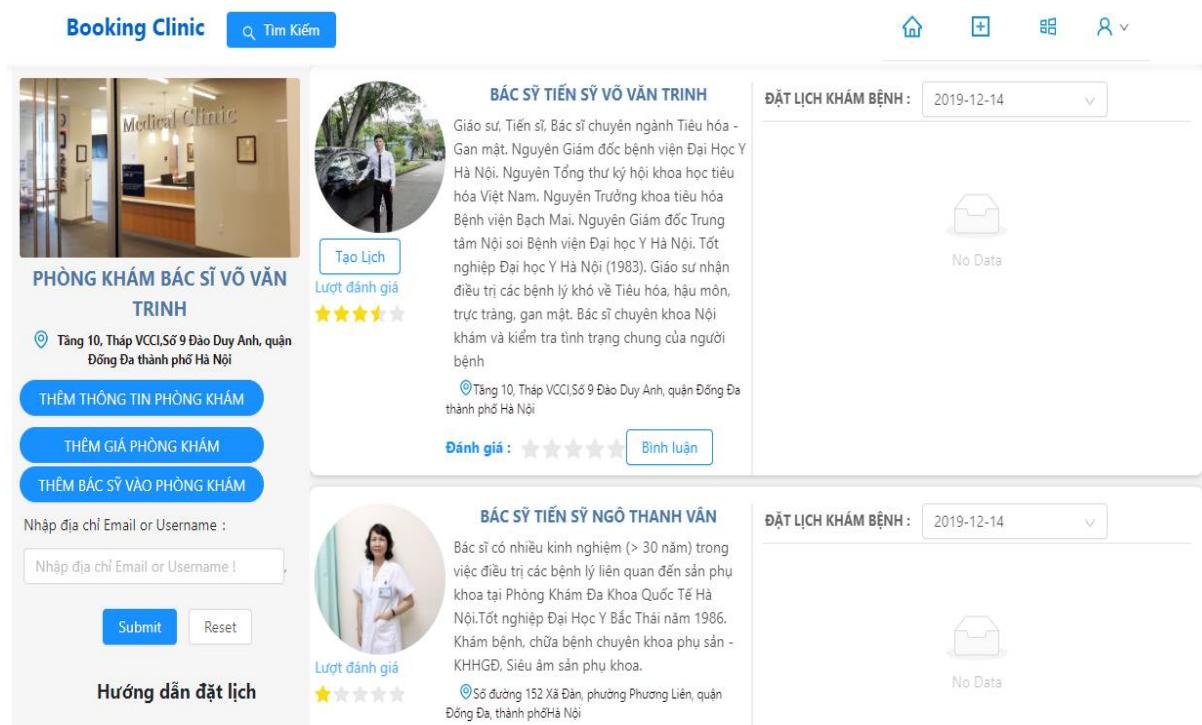
## Xây dựng ứng dụng đặt chỗ khám bệnh

Màn hình chính trước khi thêm.



Hình 3.35: Màn hình trước khi thêm bác sĩ vào phòng khám

Kết quả sau khi thêm bác sĩ vào phòng khám

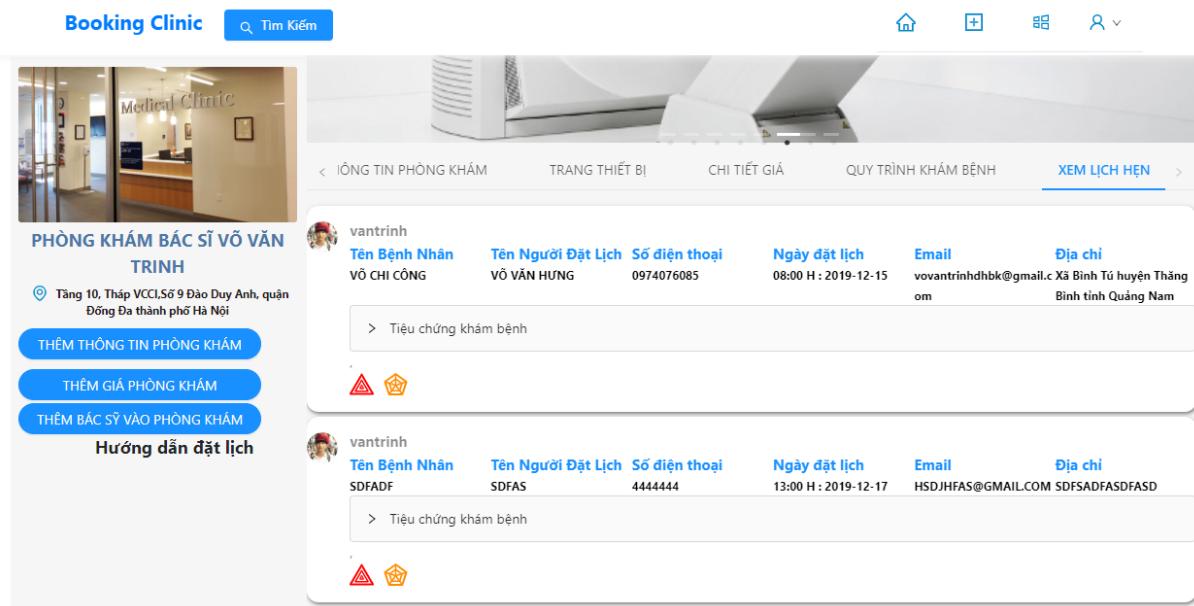


Hình 3.36: Màn hình kết quả sau khi thêm bác sĩ vào phòng khám

❖ **Báo bận cho người đặt lịch nếu bác sĩ bận và report tài khoản nếu người đặt lịch không đến khám.**

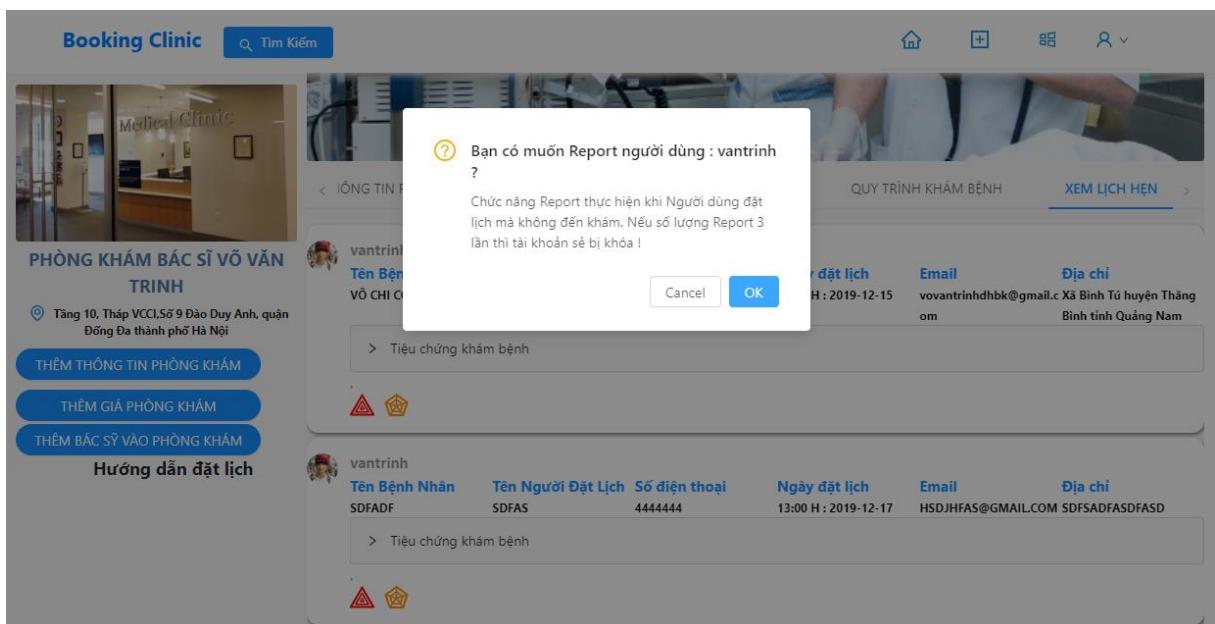
Khi vào phòng khám của mình bác sĩ sẽ thấy mục xem lịch hẹn, tại đây sẽ chứa danh sách lịch khám bệnh. Tại mỗi lịch khám bệnh sẽ có hai icon báo bận và report. Để thực hiện chức năng bạn cần chọn icon và chọn ok chức năng sẽ thực hiện.

Màn hình danh sách lịch khám bệnh của bác sĩ.



Hình 3.37: Màn hình danh sách lịch khám bệnh của bác sĩ

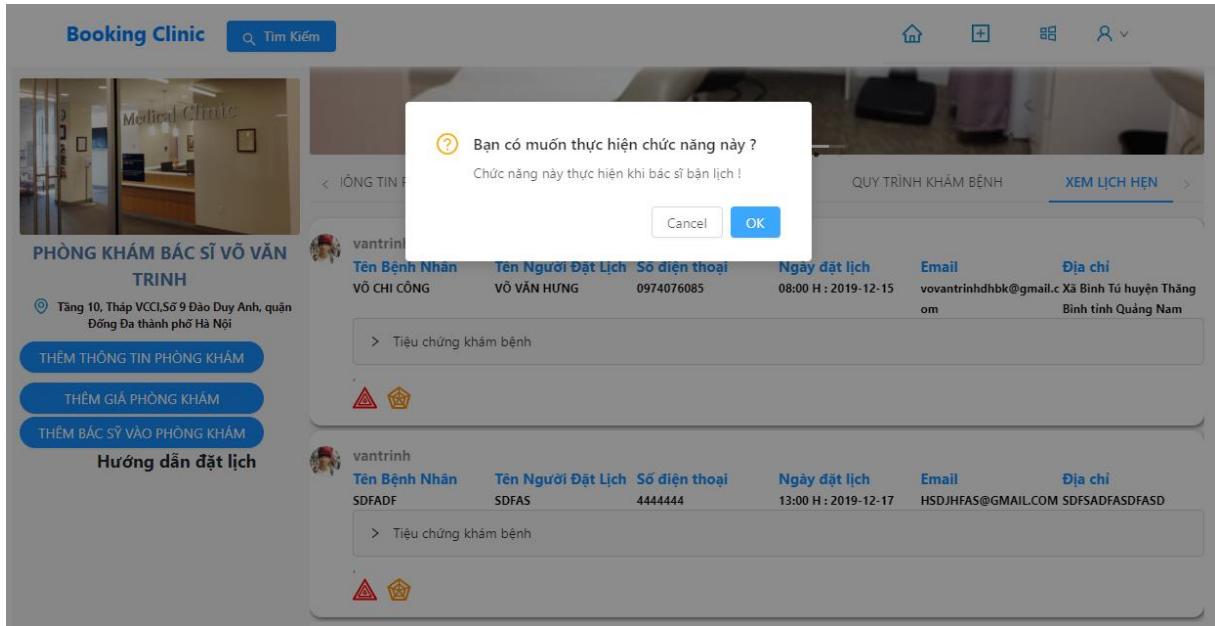
Màn hình thực hiện chức năng report.



Hình 3.38: Màn hình chức năng report tài khoản

## Xây dựng ứng dụng đặt chỗ khám bệnh

Màn hình thực hiện chức năng báo bận khi bác sĩ bận lịch.



Hình 3.39: Màn hình chức năng thông báo bận khi bác sĩ bận lịch

## KẾT LUẬN

### 1. KẾT QUẢ ĐẠT ĐƯỢC

Sau khi thực hiện đồ án, em đã thu được những thành công nhất định thông qua việc tìm hiểu, nghiên cứu và xây dựng ứng dụng thực tế. Những phần chính mà em đã nghiên cứu và ứng dụng là:

- ❖ Hiểu thêm về các giai đoạn trong việc xây dựng và phát triển website.
- ❖ Rèn luyện kỹ năng mềm như kỹ năng lập kế hoạch, kỹ năng viết báo cáo...
- ❖ Về mặt kỹ thuật, hiểu sâu và vận dụng được ngôn ngữ lập trình JAVA, Javascript, framework Spring, ReactJS để xây dựng “Ứng dụng đặt chỗ khám bệnh”. Áp dụng công nghệ quản lý phiên bản mã nguồn phân tán bằng Git và Sourcetree. Biết được cách triển khai một dự án từ môi trường phát triển local lên môi trường internet. Vận dụng các kiến thức được học thời gian học tập tại trường như: Mô hình MVC, Mô hình Client – Server, Phân tích thiết kế hướng đối tượng, Phân tích thiết kế Hệ thống thông tin, phân tích và thiết kế CSDL... để đi từ lý thuyết đến thực tế.
- ❖ Về mặt thực tiễn, sản phẩm đã đạt được mục đích đề ra, chạy ổn định các chức năng phân tích ban đầu bao gồm:
  - ✓ Người dùng có thể truy cập vào hệ thống để xem và tìm kiếm bác sĩ theo khu vực vùng miền, xem thông tin về bác sĩ, đánh giá, bình luận và lịch khám bệnh của bác sĩ.
  - ✓ Sau khi đăng nhập người dùng có thể thực hiện chức năng đánh giá bác sĩ qua việc cho số lượng sao cũng như bình luận.
  - ✓ Sau khi đăng nhập người dùng có thể thực hiện chức năng đặt lịch khám bệnh và nhận email nếu có đặt lịch.
  - ✓ Tạo Thực hiện được chức năng tạo phòng khám, tạo lịch khám cho bác sĩ, xem lịch sử lịch khám. Thông báo qua email cho bác sĩ có người đặt lịch.
  - ✓ Bác sĩ có thể báo cáo đến hệ thống nếu người bệnh nhân không đến khám để khóa tài khoản người đặt lịch.
  - ✓ Bác sĩ có thể đăng các bài viết, danh sách giá khám để tạo độ y tín cho phòng khám.

- ✓ Giao diện thân thiện, dễ dàng sử dụng.

## 2. NHỮNG VẤN ĐỀ CHUA GIẢI QUYẾT

Ngoài những mặt đã đạt được, ứng dụng vẫn còn một số vấn đề chưa được giải quyết như sau:

Hệ thống cần xây dựng một chương trình đưa ra những gợi ý bác sĩ phù hợp với tình trạng cũng như triệu chứng của người bệnh để phù hợp với chuyên ngành chuyên khoa của bác sĩ.

## 3. HƯỚNG PHÁT TRIỂN

Với những vấn đề chưa được giải quyết, hệ thống cần có những cải thiện trong tương lai như sau:

- Xây dựng ứng dụng trên nền tảng di động.
- Nghiên cứu về học máy, trí tuệ nhân tạo để xây dựng hệ thống đưa ra những gợi ý chính xác.
- Thu thập dữ liệu về y học, lọc phân tách dữ liệu để phục vụ cho quá trình nghiên cứu và phát triển hệ thống gợi ý bác sĩ.

## TÀI LIỆU THAM KHẢO

### TÀI LIỆU TIẾNG VIỆT

- [1] Lê Thị Mỹ Hạnh, *Giáo trình môn công nghệ phần mềm, Khoa Công nghệ Thông tin, Đại học Bách Khoa, Đại học Đà Nẵng.*
- [2] Thiết kế RESTful APIs, <https://www.codehub.vn/RESTful-API-Cho-Nguoi-Bat-Dau>

### TÀI LIỆU TIẾNG ANH

- [1] Spring, <https://spring.io/guides>
- [2] ReactJs, <https://reactjs.org>
- [3] Javascript, <https://vi.wikipedia.org/wiki/JavaScript>
- [4] Nguyen Thanh Binh, Vo Duc An, Le Thi My Hanh, *Object Oriented Analysis and Design, Faculty of Infomation and Technology, Da Nang University of Technology: Book.*
- [5] Java, <https://docs.oracle.com/en/java/index.html>
- [6] Microservices, <https://microservices.io/patterns/monolithic.html>