

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC**

**ĐỀ TÀI:  
Hệ thống quản lý quy trình đặt lịch khám chữa bệnh  
trực tuyến cho phòng khám**

**Giảng viên hướng dẫn : TS. NGUYỄN TẮT THẮNG**

**Sinh viên thực hiện : Nguyễn Ngọc Thiện**

**Lớp : D20HTTT03**

**Mã sinh viên : B20DCCN667**

**Hệ : ĐẠI HỌC CHÍNH QUY**

*Hà Nội, tháng 12 năm 2024*

## LỜI CẢM ƠN

(GỢI Ý Ở DƯỚI, CÓ THỂ CHỈNH SỬA)

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy TS. Nguyễn Tất Thắng, người đã trực tiếp hướng dẫn tận tình, chu đáo, chia sẻ những ý kiến và kinh nghiệm quý báu trong suốt quá trình em thực tập và thực hiện đồ án tốt.

Sau đó, em xin gửi lời cảm ơn các thầy, cô trong Học Viện nói chung và khoa CNTT1 nói riêng đã luôn nhiệt huyết, tận tình trong từng bài giảng và tạo điều kiện thuận lợi nhất cho em trong thời gian học tập và nghiên cứu tại trường Học Viện Công Nghệ Bưu Chính Viễn Thông.

Con xin được gửi lời cảm ơn tới bố mẹ và những người thân yêu đã luôn lo lắng, động viên, ủng hộ và tạo điều kiện cho con được học tập tốt. Là chỗ dựa tinh thần và những người tiếp sức cho con có được thành công trong cuộc sống.

Cuối cùng, tôi xin gửi lời chúc tốt đẹp nhất đến những người bạn của tôi và các thầy cô tham gia đợt bảo vệ tốt nghiệp trong khóa này. Chúc cho mọi người luôn vui vẻ và thành công trong cuộc sống.

*Em xin chân thành cảm ơn!*

*Hà Nội, tháng 12 năm 2024*

Sinh viên thực hiện

**Nguyễn Ngọc Thiện**

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

## CÁN BỘ GIẢNG VIÊN HƯỚNG DẪN

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

**Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?**  
 ....., ngày..... tháng..... năm 202...

---

iii

## MỤC LỤC

LỜI CẢM ƠN.....	i
NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM .....	ii
NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM .....	iii
MỤC LỤC.....	iv
BẢNG VIẾT TẮT VÀ THUẬT NGỮ .....	vi
DANH SÁCH HÌNH VẼ .....	vii
DANH SÁCH BẢNG.....	ix
MỞ ĐẦU .....	1
CHƯƠNG I. GIỚI THIỆU HỆ THỐNG.....	2
1.1 Khảo sát thực trạng .....	2
1.1.1 (Chèn tên mục viết thường).....	2
1.1.2 .....	3
1.2 .....	8
1.2.1 .....	8
1.2.2 .....	9
1.3 .....	11
1.3.1 .....	11
1.3.2 .....	12
1.3 Kết luận Chương I .....	12
CHƯƠNG II. (CHÈN TÊN CHƯƠNG VIẾT HOA).....	13
2.1 .....	13
2.1.1 .....	13
2.1.2 .....	15
2.2 .....	18
2.2.1 .....	18
2.2.2 .....	19
2.3 Kết luận Chương II .....	19
CHƯƠNG III. (CHÈN TÊN CHƯƠNG VIẾT HOA) .....	20
3.1 .....	20
3.2 .....	26

3.3 Kết luận Chương III .....	26
CHƯƠNG .....	41
.....	<b>Error! Bookmark not defined.</b>
.....	<b>Error! Bookmark not defined.</b>
3.3 Kết luận Chương .....	46
CHƯƠNG ... KẾT LUẬN.....	47
...1 Kết quả đạt được.....	47
...2 Hạn chế của hệ thống .....	47
...3 Định hướng phát triển hệ thống .....	47
DANH MỤC TÀI LIỆU THAM KHẢO .....	48

*(Phần này chỉ cần chọn toàn bộ các dòng và click chuột trái, chọn Update Field, text tự động cập nhật, nếu sai format font thì format lại font: Times New Roman size 13. Để làm tự động như thế thì ở tiêu đề các đề mục, phải chọn đúng Heading. Tên chương là Heading 1, đề mục 2 số là Heading 2, 3 số là Heading 3 v.v.)*

**BẢNG VIẾT TẮT VÀ THUẬT NGỮ**

<b>TỪ VIẾT TẮT VÀ THUẬT NGỮ</b>	<b>Ý NGHĨA</b>

## DANH SÁCH HÌNH VẼ

Hình 1.1 (Chèn tên hình vào đây, copy dòng này cho các hình khác nếu bổ sung thêm hình, format Heading 7).....	3
Hình 1.(...) (Chèn tên hình vào đây, copy dòng này cho các hình khác nếu bổ sung thêm hình, format Heading 7).....	12
Hình 2.1 (Chèn tên hình vào đây) .....	18
Hình 2.2 .....	19
Hình 2.3 (Chèn tên hình vào đây) .....	19
Hình 2.4 (Chèn tên hình vào đây) .....	19
Hình 3.1 Use case Tổng quan hệ thống.....	20
Hình 3.2 Use case Đăng nhập .....	21
Hình 3.3 Use case Tìm kiếm bác sĩ .....	21
Hình 3.4 Use case Đặt lịch khám .....	22
Hình 3.5 Use case Quản lý lịch hẹn.....	22
Hình 3.6 Use case Quản lý thông tin cá nhân .....	23
Hình 3.7 Use case Quản lý bình luận.....	23
Hình 3.8 Use case Quản lý lịch hẹn.....	24
Hình 3.9 Use case Quản lý bệnh nhân .....	24
Hình 3.10 Use case Quản lý dịch vụ phòng khám.....	25
Hình 3.11 Use case Quản lý tài khoản người dùng.....	25
Hình 3.12 Use case Quản lý bình luận.....	25
Hình 3.13 Use case Quản lý lịch hẹn (NVTN).....	26
Hình 3.14 Use case Quản lý hóa đơn.....	26
Hình 3.15 Biểu đồ hoạt động Đăng nhập .....	27
Hình 3.16 Biểu đồ hoạt động Đăng ký.....	27
Hình 3.17 Biểu đồ hoạt động Quên mật khẩu .....	27
Hình 3.18 Biểu đồ hoạt động Tìm kiếm bác sĩ.....	28
Hình 3.19 Biểu đồ hoạt động Đặt lịch khám.....	28
Hình 3.20 Biểu đồ hoạt động Quản lý lịch khám .....	29
Hình 3.21 Biểu đồ hoạt động Quản lý thông tin cá nhân.....	29
Hình 3.22 Biểu đồ hoạt động Quản lý thông tin cá nhân.....	30
Hình 3.23 Biểu đồ hoạt động Quản lý bình luận.....	30



Hình 3.24 Biểu đồ hoạt động Đổi mật khẩu .....	31
Hình 3.25 Biểu đồ hoạt động Quản lý dịch vụ.....	31
Hình 3.26 Biểu đồ hoạt động Quản lý tài khoản người dùng.....	32
Hình 3.27 (Biểu đồ lớp phân tích).....	39
Hình 3.28 (Biểu đồ Quan hệ thực thể) .....	40
Hình 4.1 Cài đặt NodeJS bước 1 .....	41
Hình 4.2 Cài đặt NodeJS bước 2 .....	41
Hình 4.3 Cài đặt NodeJS bước 3 .....	42
Hình 4.4 Cài đặt NodeJS bước 4 .....	42
Hình 4.5 Cài đặt NodeJS bước 5 .....	43
Hình 4.6 Cài đặt NodeJS bước 6 .....	43
Hình 4.7 Cài đặt NodeJS thành công.....	44
Hình 4.8 Khởi tạo dự án ReactJS .....	44
Hình 4.9 Khởi tạo dự án ReactJS thành công .....	45
Hình 4.10 Chạy dự án.....	45
Hình 4.11 Chạy dự án thành công.....	46

*(Phần này chỉ cần chọn toàn bộ các dòng và click chuột trái, chọn Update Field, text tự động cập nhật, nếu sai format font thì format lại font: Times New Roman size 13. Để làm tự động như thế thì ở các hình, tên hình phải chọn là Heading 7)*

## DANH SÁCH BẢNG

Bảng 1.1 (Chèn tên bảng vào đây nếu có bảng, copy dòng này cho các bảng khác nếu bổ sung thêm bảng, format Heading 8)..... 4

*(Phần này chỉ cần chọn toàn bộ các dòng và click chuột trái, chọn Update Field, text tự động cập nhật, nếu sai format font thì format lại font: Times New Roman size 13. Để làm tự động như thế thì ở các bảng, tên bảng phải chọn là Heading 8)*

## MỞ ĐẦU

Copy -> Edit/Paste Special (chọn Unformatted text).

...hoặc viết chèn vào đây.

...

Nội dung của đồ án ... bao gồm các phần sau:

### **Chương I: ...**

(Sơ lược Chương 1)

...

### **Chương II: ...**

(Sơ lược Chương 2)

...

### **Chương III: ...**

(Sơ lược Chương 3)

...

### **Kết luận**

(Sơ lược Kết luận)

...

## CHƯƠNG I. GIỚI THIỆU HỆ THỐNG

(Chèn giới thiệu mở chương ở đây).

### 1.1 Khảo sát thực trạng

#### 1.1.1 Đặt vấn đề

Chăm sóc sức khỏe là một trong những ngành công nghiệp phát triển nhanh chóng nhất trên khắp thế giới. Trước đây, việc đặt lịch khám bệnh chủ yếu được tiến hành thông qua các đường dây điện thoại hoặc bằng cách đến trực tiếp tại bệnh viện. Quy trình này đòi hỏi sự có mặt của bệnh nhân, do đó khả năng đặt lịch khám bị hạn chế bởi sự sẵn có của bộ phận sắp lịch, đường dây điện thoại hoặc sự có mặt trực tiếp của bệnh nhân. Không chỉ vậy, việc quản lý thông tin y tế của bệnh nhân cũng gặp nhiều khó khăn. Thông tin y tế của bệnh nhân thường được lưu trữ trên giấy hoặc trong các hệ thống riêng lẻ, dẫn đến việc truy cập, chia sẻ và quản lý thông tin này giữa các bác sĩ và các khoa trong bệnh viện trở nên phức tạp và tốn thời gian. Điều này không chỉ ảnh hưởng đến việc đưa ra quyết định nhanh chóng và chính xác trong quá trình điều trị mà còn tạo ra các rủi ro liên quan đến sai sót y tế.

Theo thời gian, mọi người bắt đầu tìm kiếm dịch vụ chăm sóc sức khỏe hiệu quả và tiết kiệm thời gian vì việc đặt lịch thủ công (yêu cầu cả hai bên phải có mặt) và việc xếp hàng dài gây bất tiện cho các cơ sở y tế. Đồng thời, nhu cầu quản lý thông tin y tế một cách chính xác, an toàn và dễ dàng truy cập ngày càng trở nên quan trọng. Vì vậy, chúng ta cần một hệ thống y tế tích hợp có thể cung cấp các dịch vụ liền mạch cho cả bệnh nhân ngoại trú và nội trú, đồng thời tối ưu hóa việc quản lý thông tin y tế.

Sự ra đời của hệ thống đặt lịch trực tuyến đã cho phép tiếp cận hiệu quả các dịch vụ y tế mà không bị hạn chế về thời gian, và sự phát triển của các hệ thống quản lý thông tin y tế điện tử (EMR) đã cải thiện khả năng truy cập, lưu trữ và quản lý dữ liệu y tế của bệnh nhân một cách an toàn và chính xác. Nhờ đó, các bác sĩ có thể nắm bắt được tiền sử bệnh lý của bệnh nhân trước khi khám, giúp nâng cao chất lượng chăm sóc và đưa ra quyết định điều trị phù hợp hơn.

Ngày nay, có rất nhiều công cụ đặt lịch hẹn trực tuyến và quản lý thông tin y tế điện tử trên thị trường, đang dần trở nên phổ biến vì dễ cài đặt và chi phí cũng không quá đắt. Hệ thống đặt lịch trực tuyến và EMR cung cấp các dịch vụ giá trị gia tăng và mang lại nhiều lợi ích cho bác sĩ và bệnh nhân. Nó tăng sự hài lòng trong trải nghiệm của cả bệnh nhân và bác sĩ bằng cách rút ngắn thời gian chờ đợi, đồng thời đảm bảo việc quản lý thông tin y tế diễn ra thuận lợi và an toàn.

...

(Chèn Hình vào đây nếu có)

*Hình 1.1 (Chèn tên hình vào đây, copy dòng này cho các hình khác nếu bổ sung thêm hình, format Heading 7)*

### **1.1.2 Đặc điểm của hệ thống quản lý phòng khám**

Đối với một hệ thống quản lý phòng khám thì cần phải có những đặc điểm và tính năng cần thiết để đảm bảo hoạt động của phòng khám diễn ra trơn tru và hiệu quả. Dưới đây là 1 số đặc điểm chính của hệ thống:

- Đối với phòng khám tư nhân, quy mô nhỏ

Hệ thống quản lý phòng khám cho các cơ sở quy mô nhỏ thường được thiết kế để đáp ứng nhu cầu cơ bản và thiết yếu của một phòng khám độc lập với những đặc điểm:

+ Quản lý lịch hẹn linh hoạt: Phù hợp với lượng bệnh nhân ít hơn và lịch làm việc đơn giản hơn của phòng khám nhỏ.

+ Hồ sơ bệnh án điện tử cơ bản: Đáp ứng nhu cầu lưu trữ thông tin cơ bản của bệnh nhân mà không cần các tính năng phức tạp.

+ Thanh toán và xuất hóa đơn đơn giản: Phù hợp với quy mô giao dịch nhỏ và ít phức tạp hơn.

+ Giao diện người dùng thân thiện: Quan trọng cho nhân viên có thể không chuyên về công nghệ thông tin.

- Đối với phòng khám đa khoa, quy mô lớn

Đối với các phòng khám quy mô lớn, phần mềm quản lý cần đáp ứng nhu cầu phức tạp và đa dạng của một cơ sở y tế lớn:

+ Hệ thống quản lý bệnh nhân toàn diện: Cần thiết để xử lý lượng lớn bệnh nhân và thông tin phức tạp.

+ Hồ sơ bệnh án điện tử nâng cao: Cần thiết để lưu trữ và quản lý thông tin chi tiết từ nhiều khoa khác nhau.

+ Quản lý lịch hẹn phức tạp: Đáp ứng nhu cầu sắp xếp lịch cho nhiều bác sĩ, nhiều khoa và các dịch vụ đa dạng.

+ Hệ thống báo cáo và phân tích dữ liệu chuyên sâu: Cần thiết cho việc ra quyết định dựa trên dữ liệu trong một tổ chức lớn.

+ Khả năng mở rộng và nâng cấp: Quan trọng để đáp ứng nhu cầu phát triển của phòng khám lớn.

+ Những tính năng này giúp tối ưu hóa hoạt động, nâng cao chất lượng chăm sóc, và quản lý hiệu quả nguồn lực trong các cơ sở y tế quy mô khác nhau, đảm bảo sự vận hành trơn tru và đáp ứng nhu cầu đa dạng của bệnh nhân.

*Bảng 1.1 (Chèn tên bảng vào đây nếu có bảng, copy dòng này cho các bảng khác nếu bổ sung thêm bảng, format Heading 8)*


### **1.1.3 Một số hệ thống quản lý phòng khám phổ biến hiện nay**

**1.1.3.1. CloudHOS** - Giải pháp hàng đầu hiện nay cho bệnh viện và phòng khám, tự động hóa quản trị bán hàng và chăm sóc khách hàng, giúp gia tăng trải nghiệm và sự hài lòng của khách hàng cho doanh nghiệp.

**- Tính năng nổi bật của hệ thống:**

- + Xây dựng kho dữ liệu khách hàng tập trung: Cung cấp góc nhìn 360 độ về khách hàng với thông tin chi tiết như liên hệ, lịch sử khám bệnh, và lịch sử chăm sóc.
- + Kết nối đa kênh: Tập trung dữ liệu khách hàng từ nhiều kênh vào một nơi duy nhất, giúp phân quyền bảo mật dữ liệu tối đa và phân loại khách hàng theo nhiều tiêu chí.
- + Quản lý marketing và bán hàng: Đo lường và tối ưu hiệu quả các chiến dịch quảng cáo, thực hiện remarketing qua SMS, email, Zalo ZNS, và hỗ trợ bán hàng cho cả khách cá nhân và tổ chức.
- + Quản lý lịch hẹn và quy trình khám: Tự động nhắc nhở lịch hẹn qua SMS, ZNS, quản lý lịch làm việc của bác sĩ, và kết nối quy trình khám chữa bệnh với hệ thống HIS để đồng bộ thông tin bệnh nhân.
- + Chăm sóc khách hàng đa kênh: Tương tác với khách hàng qua nhiều kênh như gọi điện, SMS, email, Zalo, và Messenger, xây dựng chu kỳ chăm sóc tự động và xử lý khiếu nại nhanh chóng.

+ Báo cáo phân tích: Cung cấp dashboard tập trung để theo dõi báo cáo, tạo báo cáo động theo nhu cầu, và truy cập báo cáo mọi lúc, mọi nơi.

- ***Ưu điểm của phần mềm:***

Phát triển chuyên sâu theo nghiệp vụ y tế: Được thiết kế đặc biệt cho ngành y, đáp ứng đầy đủ các yêu cầu nghiệp vụ của phòng khám và bệnh viện tại Việt Nam.

Triển khai cho nhiều đơn vị: Đã được áp dụng thành công cho nhiều phòng khám và bệnh viện trên toàn quốc, khẳng định tính ứng dụng thực tế và hiệu quả.

Khả năng tùy chỉnh linh hoạt: Hệ thống dễ dàng tùy chỉnh để phù hợp với các đặc thù riêng của từng cơ sở y tế, giúp tối ưu hóa quy trình làm việc.

Đội ngũ tư vấn giàu kinh nghiệm: Đội ngũ chuyên gia có kinh nghiệm sâu rộng, luôn sẵn sàng hỗ trợ nhanh chóng, đảm bảo giải quyết mọi vấn đề phát sinh.

Dịch vụ hỗ trợ sau triển khai: Dịch vụ hậu mãi được đảm bảo, hỗ trợ liên tục sau khi triển khai, giúp duy trì và tối ưu hóa hệ thống trong suốt quá trình sử dụng.

1.1.3.2. Phần mềm quản lý phòng khám **TCSOFT**

TCSOFT là một giải pháp phần mềm quản lý phòng khám toàn diện, được phát triển bởi công ty TNHH TCSOFT, một doanh nghiệp công nghệ Việt Nam. Phần mềm này được thiết kế để đáp ứng nhu cầu quản lý của các phòng khám và cơ sở y tế vừa và nhỏ tại Việt Nam.

- ***Tính năng nổi bật:***

+ Quản lý thông tin bệnh nhân: Lưu trữ và truy xuất hồ sơ bệnh án, lịch sử khám chữa bệnh.

+ Lập lịch hẹn và quản lý lịch khám: Hỗ trợ đặt lịch tự động, gửi tin nhắn nhắc lịch hẹn.

+ Kê đơn thuốc và quản lý thuốc: Hệ thống cảnh báo tương tác thuốc, quản lý kho thuốc.

+ Quản lý tài chính và hóa đơn: Tích hợp với hệ thống kế toán, xuất hóa đơn điện tử.

+ Báo cáo thống kê: Tạo báo cáo tự động về hoạt động phòng khám, doanh thu, bệnh nhân.

+ Tích hợp với thiết bị y tế: Kết nối với các thiết bị như máy xét nghiệm, máy chụp X-quang.

+ Bảo mật dữ liệu: Hệ thống mã hóa và phân quyền truy cập.

- ***Ưu điểm:***

- + Giao diện thân thiện, được thiết kế phù hợp với người dùng Việt Nam.
- + Phù hợp với quy trình làm việc của các phòng khám Việt Nam, tuân thủ các quy định của Bộ Y tế.
- + Hỗ trợ kỹ thuật từ đội ngũ trong nước, dễ dàng liên hệ và được hỗ trợ nhanh chóng.
- + Phần mềm liên tục cải tiến và bổ sung tính năng mới.

- **Nhược điểm:**

- + Thiếu một số tính năng chuyên sâu cho các phòng khám lớn hoặc bệnh viện
- + Khả năng tích hợp với các hệ thống khác có thể hạn chế
- + Giao diện chưa được tối ưu cho thiết bị di động.
- + Khả năng mở rộng có thể bị giới hạn khi phòng khám phát triển nhanh.

### 1.1.3.3. Giải pháp quản lý phòng khám **VNPT Home & Clinic**

VNPT Home & Clinic là một giải pháp phần mềm quản lý phòng khám toàn diện được phát triển bởi Tập đoàn Bưu chính Viễn thông Việt Nam (VNPT). Đây là một phần của hệ sinh thái các giải pháp công nghệ thông tin trong lĩnh vực y tế của VNPT.

- **Tính năng nổi bật:**

- + Quản lý hồ sơ bệnh án điện tử: Lưu trữ và truy xuất thông tin bệnh nhân an toàn, bảo mật.
- + Đặt lịch khám trực tuyến: Cho phép bệnh nhân đặt lịch qua web hoặc ứng dụng di động.
- + Kê đơn và quản lý thuốc: Hệ thống cảnh báo tương tác thuốc, quản lý kho thuốc tự động.
- + Quản lý doanh thu và chi phí: Tích hợp với hệ thống kế toán, báo cáo tài chính.
- + Tích hợp với các thiết bị y tế: Kết nối và lưu trữ dữ liệu từ các thiết bị chẩn đoán.
- + Hỗ trợ telemedicine: Tính năng tư vấn và khám bệnh từ xa.
- + Phân tích dữ liệu và AI: Hỗ trợ ra quyết định lâm sàng dựa trên dữ liệu lớn.
- + Tích hợp BHYT: Kết nối với hệ thống bảo hiểm y tế quốc gia.

- **Ưu điểm**

- + Được phát triển bởi tập đoàn lớn, có độ tin cậy cao
- + Tích hợp tốt với các dịch vụ khác của VNPT, tạo hệ sinh thái hoàn chỉnh.



- + Hỗ trợ đa nền tảng, sử dụng được trên máy tính, điện thoại, tablet.
- + Cập nhật liên tục, đáp ứng nhanh với các thay đổi trong quy định y tế.
- + Hỗ trợ kỹ thuật 24/7.

- **Nhược điểm:**

- + Chi phí cao hơn so với các giải pháp khác, phù hợp với các phòng khám lớn hoặc bệnh viện.
- + Quy trình cài đặt và triển khai khá phức tạp, đòi hỏi thời gian và nguồn lực để triển khai.
- + Có nhiều tính năng không cần thiết cho phòng khám nhỏ.

#### 1.1.3.4. Giải pháp quản lý phòng khám **STD Clinic**

STD Clinic là phần mềm quản lý phòng khám chuyên biệt được thiết kế cho các cơ sở y tế chuyên điều trị bệnh lây truyền qua đường tình dục (STD - Sexually Transmitted Diseases). Phần mềm này tập trung vào việc quản lý thông tin nhạy cảm và quy trình điều trị đặc thù cho các bệnh STD

- **Tính năng nổi bật:**

- + Quản lý hồ sơ bệnh nhân bảo mật: Mã hóa dữ liệu, phân quyền truy cập chặt chẽ.
- + Theo dõi quá trình điều trị: Lịch sử điều trị, đánh giá tiến triển bệnh.
- + Quản lý xét nghiệm và kết quả: Tích hợp với các thiết bị xét nghiệm STD.
- + Báo cáo dịch tễ học: Tạo báo cáo tự động cho cơ quan y tế về tình hình dịch bệnh.
- + Hỗ trợ tư vấn và giáo dục bệnh nhân: Cung cấp tài liệu giáo dục sức khỏe tình dục.
- + Quản lý liên hệ bạn tình: Hỗ trợ theo dõi và thông báo cho các đối tác có nguy cơ.
- + Tích hợp với hệ thống giám sát quốc gia: Báo cáo số liệu cho cơ quan y tế.
- + Lịch hẹn tái khám và nhắc nhở: Tự động gửi thông báo cho bệnh nhân.

- **Ưu điểm:**

- + Chuyên biệt cho phòng khám STD, tối ưu hóa quy trình làm việc đặc thù.
- + Đảm bảo tính bảo mật cao cho thông tin nhạy cảm.
- + Hỗ trợ báo cáo cho cơ quan y tế
- + Tích hợp các hướng dẫn điều trị STD mới nhất.
- + Cung cấp dữ liệu ẩn danh cho nghiên cứu dịch tễ học.

- **Nhược điểm:**

- + Chỉ phù hợp cho loại phòng khám cụ thể, không phù hợp cho phòng khám đa khoa.
- + Chi phí có thể cao do tính chuyên biệt.
- + Khả năng tích hợp với các hệ thống quản lý bệnh viện tổng quát có thể hạn chế.
- + Đòi hỏi đào tạo chuyên sâu cho nhân viên sử dụng, cần thời gian để làm quen với các tính năng đặc thù.

#### 1.1.3.5. Công cụ quản lý phòng khám **NANOSOFT**

NANOSOFT là một công ty chuyên cung cấp giải pháp phần mềm quản lý phòng khám, bán hàng, thú y và bảo hành. Họ tập trung vào việc cung cấp các giải pháp ứng dụng và giải pháp về công nghệ thông tin cho doanh nghiệp.

- **Tính năng nổi bật:**

- + Quản lý bán hàng cho các phòng khám, cơ sở thú y.
- + Tích hợp quét/in mã vạch, quét QR Code CCCD, import/export dữ liệu từ file Excel.
- + Quản lý thông tin bệnh nhân và hồ sơ bệnh án
- + Kê đơn thuốc và quản lý kho thuốc

- **Ưu điểm:**

- + Giao diện thân thiện, dễ sử dụng
- + Phù hợp với quy trình làm việc của các phòng khám Việt Nam
- + Hỗ trợ kỹ thuật từ đội ngũ trong nước

- **Nhược điểm:**

- + TCSOFT chưa tối ưu trên smartphone và tablet, gây khó khăn khi cần truy cập nhanh.
- + Thiếu tính năng đặt lịch trực tuyến, gây bất tiện cho bệnh nhân trong việc đăng ký khám từ xa.

## 1.2 Mục tiêu, phạm vi, chức năng của hệ thống

### 1.2.1 Mục tiêu

Mục tiêu chính của đề tài là tạo ra được 1 hệ thống có thể đáp ứng được các tiêu chí sau:

- Tối ưu hóa quy trình đặt lịch khám bệnh trực tuyến: Cung cấp cho bệnh nhân và bác sĩ một nền tảng tiện lợi để đặt lịch khám, giúp giảm thiểu thời gian chờ đợi, tránh tình trạng quá tải tại phòng khám và nâng cao trải nghiệm của người dùng.
- Số hóa và quản lý thông tin y tế một cách hiệu quả: Lưu trữ và quản lý hồ sơ bệnh nhân, lịch sử khám bệnh, và thông tin y tế một cách an toàn, dễ truy cập và chính xác, giúp bác sĩ đưa ra các quyết định điều trị nhanh chóng và chính xác hơn.
- Cải thiện khả năng quản lý và vận hành của phòng khám: Hỗ trợ quản trị viên và nhân viên y tế quản lý lịch làm việc, tài nguyên, và dữ liệu bệnh nhân hiệu quả hơn, tối ưu hóa quy trình điều phối và tăng cường tính bảo mật cho thông tin y tế.

### 1.2.2 Chức năng hệ thống

Hệ thống này được triển khai cho tất cả những cá nhân muốn được đặt lịch khám và điều trị tại phòng khám. Người dùng chỉ có thể tham gia nếu họ đã tạo tài khoản thông qua biểu mẫu đăng ký và cung cấp lịch sử y tế của mình

Nội dung đề tài sẽ bao gồm:

- Đặt lịch khám bệnh trực tuyến:
  - + Cho phép bệnh nhân tự đặt lịch khám với các bác sĩ dựa trên thời gian trống, chuyên khoa, và dịch vụ y tế mong muốn.
  - + Gửi thông báo xác nhận lịch hẹn qua email của bệnh nhân
- Quản lý lịch làm việc của bác sĩ, hiển thị lịch trống
- Quản lý thông tin bệnh nhân:
  - + Lưu trữ và quản lý thông tin cá nhân của bệnh nhân (tên, tuổi, giới tính, địa chỉ, số điện thoại).
  - + Quản lý hồ sơ y tế của bệnh nhân, bao gồm lịch sử khám bệnh, kết quả xét nghiệm, đơn thuốc, và quá trình điều trị.
- Quản lý thông tin bác sĩ và lịch làm việc:
  - + Cung cấp thông tin chi tiết về các bác sĩ bao gồm chuyên khoa, kinh nghiệm, và lịch làm việc.
  - + Cho phép quản trị viên và bác sĩ quản lý lịch khám bệnh, hủy lịch hoặc điều chỉnh thời gian khám.
- Bảo mật thông tin và quyền truy cập:

+ Bảo vệ thông tin cá nhân và y tế của bệnh nhân bằng các phương pháp mã hóa, xác thực và phân quyền truy cập cho từng vai trò trong hệ thống (bệnh nhân, bác sĩ, quản trị viên).

- Báo cáo và thống kê:

+ Hệ thống sẽ cung cấp các báo cáo và thống kê liên quan đến lịch khám bệnh, số lượng bệnh nhân, và kết quả khám chữa bệnh.

- ***Chức năng chi tiết cho từng vai trò***

- Bệnh nhân:

+ Đăng ký và đăng nhập hệ thống.

+ Đặt lịch khám và theo dõi lịch sử khám bệnh.

+ Xem đơn thuốc, dịch vụ đã thực hiện và hóa đơn.

+ Quản lý thông tin cá nhân

- Bác sĩ:

+ Xem danh sách bệnh nhân và điều chỉnh lịch hẹn.

+ Chẩn đoán, kê đơn, chỉ định dịch vụ và cập nhật hồ sơ bệnh án.

+ Xem và tải lên tài liệu y tế (như ảnh chụp X-quang, kết quả xét nghiệm).

+ Quản lý thông tin cá nhân

- Nhân viên:

+ Quản lý lịch hẹn và xác nhận đặt lịch.

+ Quản lý thanh toán, tạo và cập nhật hóa đơn.

- Quản trị viên:

+ Quản lý tài khoản của tất cả các vai trò trong hệ thống.

+ Xem báo cáo thống kê

**1.2.3 Phạm vi hệ thống**

- Giới hạn về nghiệp vụ

+ Chỉ hỗ trợ khám bệnh ngoại trú: Hệ thống chỉ tập trung vào việc quản lý khám và điều trị cho bệnh nhân ngoại trú, không hỗ trợ quản lý bệnh nhân nội trú.

+ Hạn chế quản lý chuyên sâu về hồ sơ y tế: Hồ sơ bệnh án sẽ lưu trữ chẩn đoán, đơn thuốc và các dịch vụ đã sử dụng, nhưng không lưu thông tin chi tiết về quá trình điều trị dài hạn hoặc bệnh sử phức tạp.

- Giới hạn về loại hình dịch vụ y tế: Chỉ cung cấp các dịch vụ phổ biến (khám tổng quát, xét nghiệm máu, X-quang), không bao gồm các dịch vụ y tế chuyên khoa phức tạp hoặc phẫu thuật.

- Giới hạn về chức năng

+ Chỉ hỗ trợ một phòng khám duy nhất: Hệ thống được thiết kế để quản lý một phòng khám cố định và không hỗ trợ nhiều phòng khám.

+ Không tích hợp bảo hiểm y tế: Hệ thống sẽ không xử lý bảo hiểm y tế hoặc liên kết với các tổ chức bảo hiểm khác; chi phí và thanh toán sẽ được quản lý riêng lẻ cho mỗi bệnh nhân.

+ Giới hạn về thông báo: Thông báo nhắc lịch hẹn hoặc tái khám sẽ chỉ được gửi qua email hoặc SMS (không hỗ trợ ứng dụng di động hoặc các phương thức liên lạc khác).

+ Không bao gồm chức năng thống kê nâng cao: Hệ thống chỉ cung cấp báo cáo cơ bản về số lần khám và thu phí, không hỗ trợ báo cáo phân tích dữ liệu nâng cao hoặc AI để phân tích dữ liệu y tế.

- Giới hạn về kỹ thuật

Bảo mật cơ bản: Hệ thống sử dụng bảo mật thông qua phân quyền và xác thực tài khoản cơ bản, không có các phương pháp bảo mật nâng cao như xác thực hai yếu tố hoặc mã hóa dữ liệu phức tạp.

+ Lưu trữ hình ảnh và tài liệu ở mức cơ bản: Hệ thống cho phép lưu trữ các tài liệu như ảnh chụp X-quang và xét nghiệm nhưng không hỗ trợ lưu trữ với dung lượng lớn hoặc tối ưu hóa tốc độ truy cập file phức tạp.

+ Không hỗ trợ tích hợp API của bên thứ ba: Hệ thống không tích hợp với các hệ thống y tế khác hoặc các API bên ngoài như thanh toán trực tuyến hoặc tích hợp với ứng dụng lịch.

### 1.3

#### 1.3.1 ...

(Nội dung)

...

### **1.3.2 ...**

(Nội dung)

...

(Chèn Hình vào đây nếu có)

*Hình 1.(...) (Chèn tên hình vào đây, copy dòng này cho các hình khác nếu bổ sung thêm hình, format Heading 7)*

## **1.4 Kết luận Chương I**

...

## CHƯƠNG II. CƠ SỞ LÝ THUYẾT

(Chèn giới thiệu mở chương ở đây).

### 2.1 Công nghệ sử dụng

...

...

#### 2.1.1 React JS

##### - *ReactJS là gì:*

ReactJS được phát triển bởi Facebook và được giới thiệu lần đầu tiên vào năm 2011. Ban đầu, ReactJS được phát triển để xây dựng giao diện người dùng trên trang web Facebook, nhằm cải thiện tốc độ và hiệu suất của ứng dụng web.

Tuy nhiên, ReactJS không được công bố cho cộng đồng phát triển cho đến năm 2013, khi Facebook công bố mã nguồn mở của nó và giới thiệu cho cộng đồng lập trình viên. Từ đó, ReactJS nhanh chóng trở thành một trong những thư viện phổ biến nhất để phát triển các ứng dụng web động.

Sau đó, vào năm 2015, Facebook giới thiệu phiên bản React Native, một framework phát triển ứng dụng di động sử dụng ReactJS. React Native cho phép các nhà phát triển xây dựng các ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn, tương tự như ReactJS trên web.

Hiện nay, ReactJS đã trở thành một trong những thư viện phát triển web phổ biến nhất, được sử dụng rộng rãi bởi các công ty lớn và nhỏ trên toàn thế giới. Facebook cũng tiếp tục đầu tư phát triển và nâng cấp ReactJS để đáp ứng nhu cầu của cộng đồng phát triển.

##### - *ReactJS mang đến nhiều lợi ích cho việc phát triển ứng dụng web, bao gồm:*

**Hiệu suất cao:** ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.

**Tái sử dụng:** ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

**Dễ dàng quản lý trạng thái:** ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.

**Hỗ trợ tốt cho SEO:** ReactJS cho phép các nhà phát triển xây dựng ứng dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.

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

**- *ReactJS có nhiều tính năng hữu ích cho việc phát triển ứng dụng web:***

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

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

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

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

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

**Hỗ trợ đa nền tảng:** ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các



nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

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

### 2.1.2 Java Spring Boot

#### - *Spring Boot là gì*

Spring Boot được xây dựng dựa trên nền của Spring Framework, phát triển bởi ngôn ngữ Java. Nó là một phiên bản có tính tự động hoá cao hơn và đơn giản hơn của Spring. Spring Boot khiến cho việc tạo và khởi chạy ứng dụng chỉ diễn ra trong vòng vài phút.

#### - *Lịch sử phát triển của Spring Boot*

Spring Boot được phát triển bởi Pivotal Software, với phiên bản đầu tiên (Spring Boot 1.0) được phát hành vào tháng 4 năm 2014. Mục tiêu ban đầu của Spring Boot là giải quyết những hạn chế và phức tạp trong việc cấu hình Spring Framework truyền thống. Spring Framework, mặc dù rất mạnh mẽ và linh hoạt, nhưng yêu cầu nhiều cấu hình thủ công và có thể trở nên cồng kềnh khi xây dựng các ứng dụng phức tạp.

Trước khi Spring Boot ra đời, việc cấu hình một ứng dụng Spring thường đòi hỏi nhiều file XML hoặc các class Java cấu hình, gây khó khăn cho việc bảo trì và phát triển. Spring Boot được giới thiệu như một cách tiếp cận đơn giản và hiệu quả hơn, với các cấu hình tự động và các công cụ hỗ trợ như Spring Initializr, giúp nhà phát triển bắt đầu dự án mới nhanh chóng chỉ với một vài cú nhấp chuột.

Từ khi ra mắt, Spring Boot đã nhanh chóng trở thành một trong những framework phổ biến nhất trong cộng đồng Java, được sử dụng rộng rãi trong nhiều loại ứng dụng khác nhau, từ các ứng dụng web đơn giản đến các hệ thống microservices phức tạp. Sự phát triển và cải tiến liên tục của Spring Boot đã giúp nó trở thành một phần không thể thiếu trong hệ sinh thái Spring, giúp các nhà phát triển xây dựng các ứng dụng hiện đại một cách dễ dàng và nhanh chóng hơn.

#### - *Sự khác biệt giữa Spring và Spring Boot*

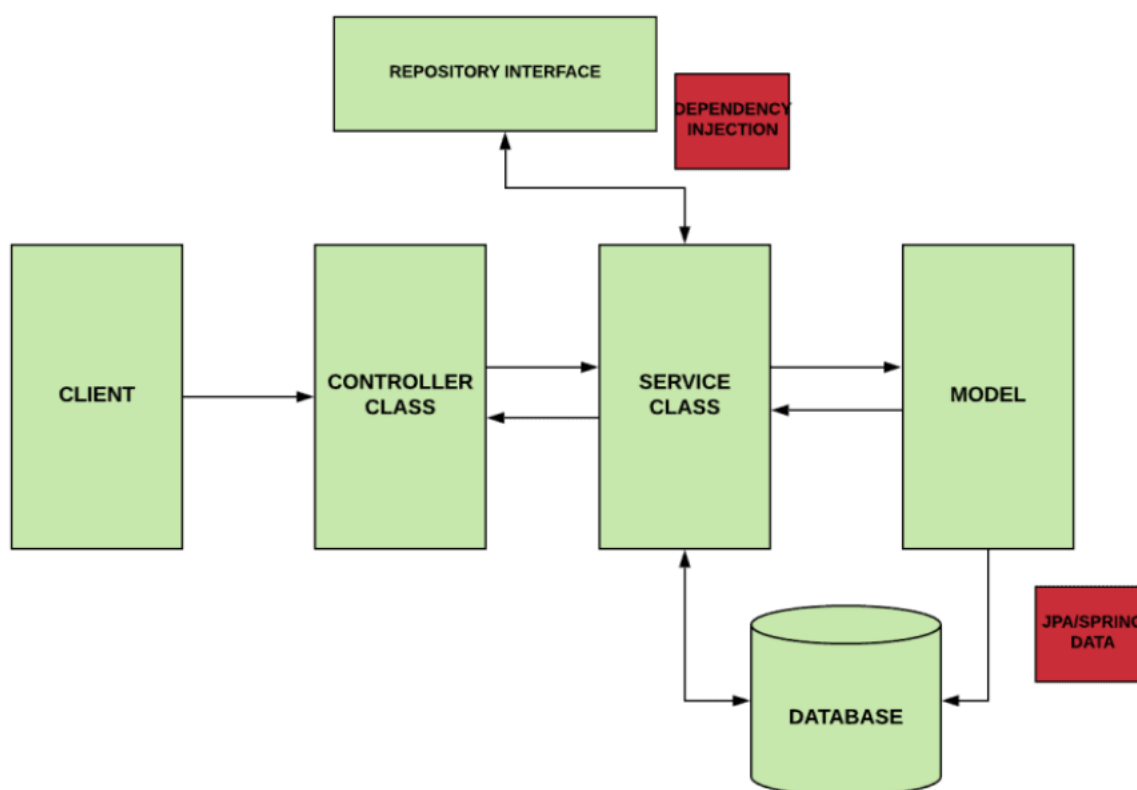
Spring là một tập hợp các framework con, bao gồm nhiều module khác nhau như Spring MVC, Spring JDBC, Spring AOP, Spring ORM và Spring Test. Tính năng cốt lõi của Spring Framework chính là Dependency Injection (DI), cho phép các đối tượng được tiêm

(injected) vào nhau một cách linh hoạt mà không cần phải tạo mới hoặc quản lý chúng trực tiếp. Điều này làm cho ứng dụng trở nên dễ dàng quản lý và mở rộng hơn.

Mặc dù Spring đã giúp cho việc phát triển các ứng dụng Java trở nên dễ dàng hơn, nhưng vẫn còn đó sự phức tạp và độ rộng lớn mà framework này mang lại. Việc học Spring Framework từ đầu có thể là một hành trình dài và đầy thách thức đối với nhiều lập trình viên. Đó cũng chính là lúc Spring Boot tỏa sáng với những cải tiến vượt trội.

Spring Boot ra đời với mục tiêu đơn giản hóa việc phát triển các ứng dụng Java bằng cách giảm thiểu cấu hình thủ công và cung cấp các thiết lập mặc định hợp lý. Thay vì phải cấu hình một cách phức tạp và chi tiết như trong Spring, Spring Boot giúp lập trình viên khởi tạo và chạy ứng dụng một cách nhanh chóng chỉ với một vài dòng mã. Spring Boot còn tích hợp sẵn các công cụ như máy chủ nhúng (embedded server), hỗ trợ cấu hình bằng YAML, và tự động hóa nhiều tác vụ mà trước đây cần phải làm thủ công.

#### - *Workflow của 1 dự án Spring Boot*



Workflow của Spring Boot hoạt động như sau:

1. Client tạo một HTTP request.
2. Class Controller nhận HTTP request.
3. Controller hiểu loại request và xử lý nó.
4. Nếu cần, nó gọi class Service.

5. Class Service sẽ xử lý logic nghiệp vụ, với dữ liệu từ database
6. Nếu mọi thứ hoạt động tốt, ta sẽ trả về một trang JSP.

### 2.1.3 PostgreSQL

#### - *PostgreSQL là gì*

PostgreSQL được ứng dụng phổ biến xây dựng như một hệ cơ sở quản trị dữ liệu mở. Hệ cơ sở dữ liệu này giữ vai trò tiên phong mở đường cho khái niệm hệ quản trị dữ liệu thương mại ra đời sau này.

Thời kỳ đầu, PostgreSQL hoạt động của giống như nền tảng điều hành Unix. Về sau, PostgreSQL dần thay đổi để trở nên linh hoạt và có thể khởi chạy trên một số nền tảng như Windows, Solaris, Mac OS. Đồng thời, tích hợp thêm nhiều tính năng vượt trội khác.

Về mặt kỹ thuật, PostgreSQL chính là một mã nguồn mở hoàn toàn miễn phí, xây dựng theo quy chuẩn SQL99. Nhờ đó, tất cả người dùng đều có quyền sử dụng, tham gia chỉnh sửa.

Khác với phần lớn cơ sở dữ liệu khác, PostgreSQL hoạt động rất ổn định không yêu cầu bảo trì cầu kỳ. Nó dễ dàng để triển khai trên nhiều ứng dụng khác nhau với phí đầu tư hợp lý

#### - *Vai trò chính của PostgreSQL*

PostgreSQL hoạt động như một hệ quản trị dữ liệu mang tính mở, phù hợp để mọi doanh nghiệp khai thác. Đây là một hệ quản trị tương thích với hầu hết các nền tảng, ứng dụng đa ngôn ngữ, phần mềm phổ biến.

Ngày nay, PostgreSQL là một phần quan trọng của ngành dữ liệu GIS cho tổ chức chính phủ, ngành ngân hàng, thương mại, công nghệ web,.. PostgreSQL hoàn toàn không bị quản lý bởi bất kỳ tổ chức nào. Tuy vậy, việc không bị quản lý bởi bất kỳ tổ chức nào đôi khi lại là yếu điểm của PostgreSQL.

#### - *Một vài tính năng cơ bản của PostgreSQL*

PostgreSQL mang đến người dùng vô số chức năng hữu ích. Cùng với đó là khả năng vận hành ổn định, tốc độ tối ưu.

Cung cấp nhiều kiểu dữ liệu: PostgreSQL cung cấp đa dạng kiểu dữ liệu như nguyên hàm, cấu trúc, hình học,..

Bảo đảm toàn vẹn dữ liệu: Dữ liệu trong PostgreSQL luôn được đảm bảo tính toàn vẹn.

Tính năng thiết lập linh hoạt: PostgreSQL cho phép người dùng thiết lập danh mục từ cơ bản đến nâng cao, tối ưu ưu hóa tốc độ truy cập, hỗ trợ thống kê trên nhiều cột,.. Cùng với đó là vô số thiết lập khác.

Chức năng bảo mật: PostgreSQL hỗ trợ xây dựng hàng rào bảo mật, xác thực mạnh.

Khả năng mở rộng: Người dùng có thể thực hiện mở rộng hệ thống qua các phức thức lưu trữ, kết nối cơ sở dữ liệu.

Chức năng tìm kiếm văn bản: PostgreSQL cung cấp tính năng tìm kiếm văn bản đầy đủ, hệ thống hóa ký tự theo cách khoa học.

Ngoài ra, PostgreSQL còn tích hợp chức năng cùng lúc quản lý số lượng lớn người dùng.

### - *Ưu điểm và nhược điểm của PostgreSQL*

PostgreSQL là hệ quản trị dữ liệu sở hữu nhiều ưu điểm nổi trội. Tuy nhiên song song với đó, hệ quản trị này vẫn còn tồn tại không ít nhược điểm.

#### - Ưu điểm

Có khả năng chạy trang web, ứng dụng web động.

Lưu lại nhật ký, hình thành cơ sở dữ liệu hỗ trợ sửa lỗi.

Mã nguồn PostgreSQL luôn sẵn sàng để người dùng chỉnh sửa nâng cấp.

Hỗ trợ khách hàng theo từng vùng địa lý.

Cách thức sử dụng đơn giản.

#### - Hạn chế

PostgreSQL không chịu sự quản lý của bất kỳ tổ chức nào gây khó khăn để người tiếp cận với đầy đủ tính năng.

Có vô số ứng dụng ứng dụng mã nguồn mở không phải ứng dụng nào cũng được hỗ trợ tốt.

Hiệu suất hoạt động chậm hơn so với MySQL

## 2.2 Một số khái niệm trong ngành y tế

### 2.2.1 ...

...

...

(Chèn Hình vào đây nếu có)

*Hình 2.1 (Chèn tên hình vào đây)*

(Chèn Hình vào đây nếu có)

*Hình 2.2 ...*

...

...

### **2.2.2 ...**

...

...

(Chèn Hình vào đây nếu có)

*Hình 2.3 (Chèn tên hình vào đây)*

...

...

(Chèn Hình vào đây nếu có)

*Hình 2.4 (Chèn tên hình vào đây)*

....

....

## **2.3 Kết luận Chương II**

....

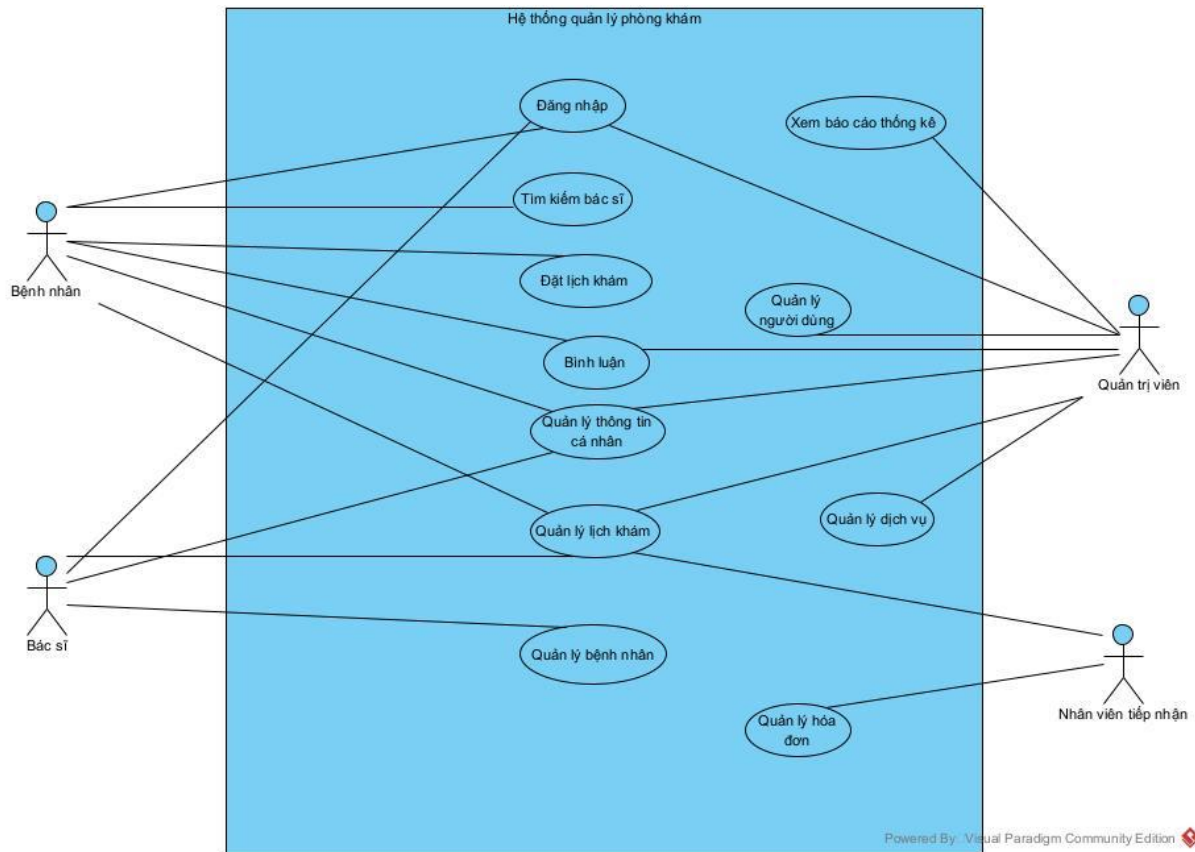
....

## CHƯƠNG III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

(Chèn giới thiệu mở chương ở đây).

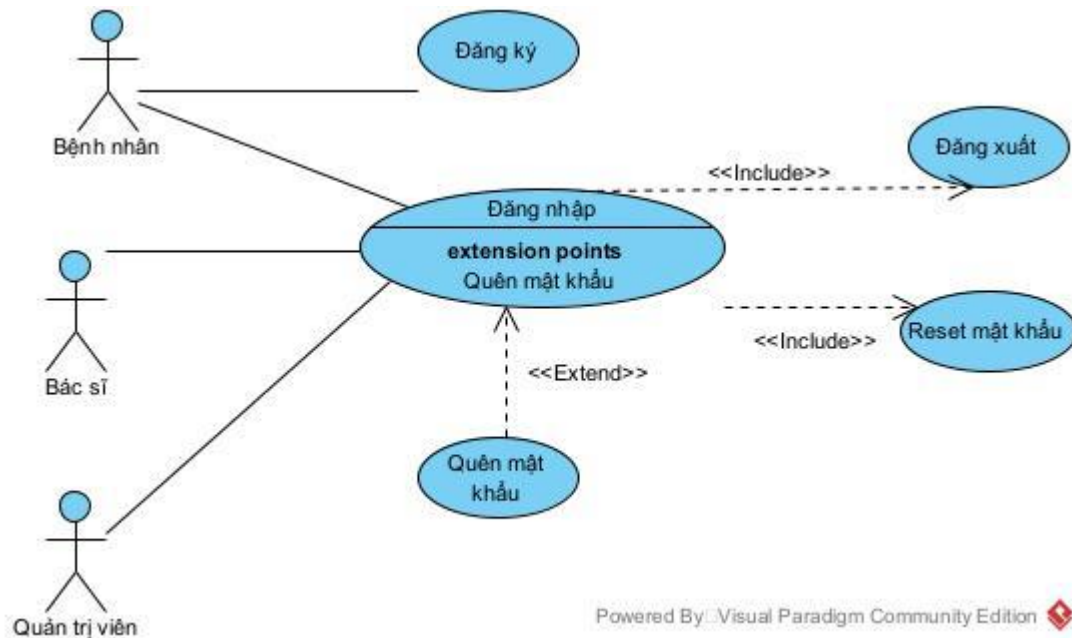
### 3.1 Biểu đồ Use case (Usecase diagram)

#### 3.1.1 Use case Tổng quan hệ thống



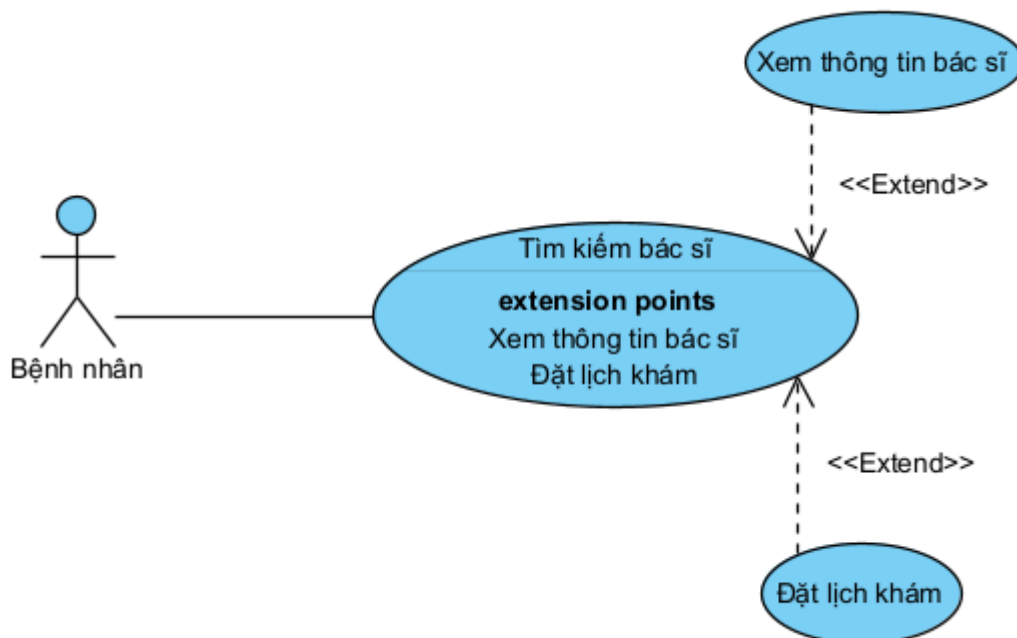
Hình 3.1 Use case Tổng quan hệ thống

### 3.1.2 Use case Đăng nhập



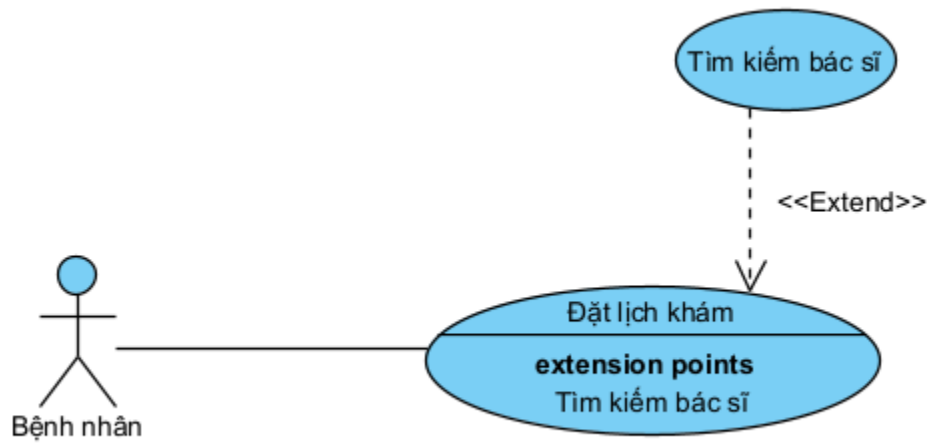
Hình 3.2 Use case Đăng nhập

### 3.1.3 Use case Tìm kiếm bác sĩ (Bệnh nhân)



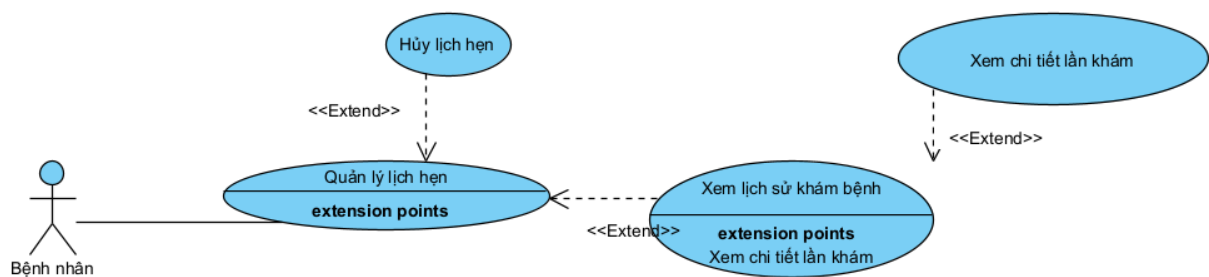
Hình 3.3 Use case Tìm kiếm bác sĩ

### 3.1.4 Use case Đặt lịch khám (Bệnh nhân)



Hình 3.4 Use case Đặt lịch khám

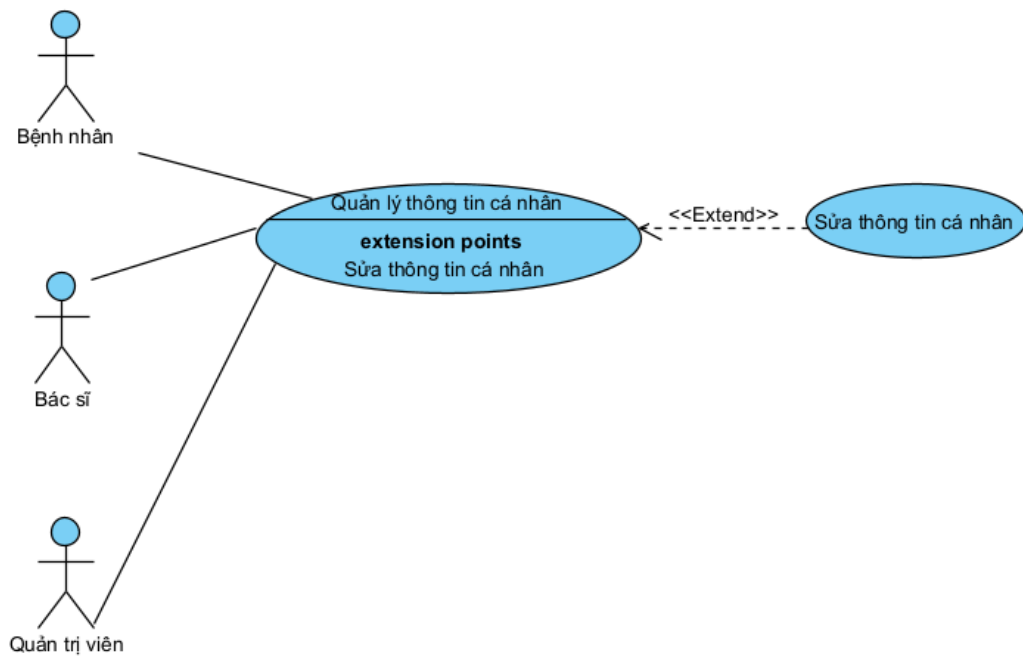
### 3.1.5 Use case Quản lý lịch hẹn (Bệnh nhân)



Hình 3.5 Use case Quản lý lịch hẹn

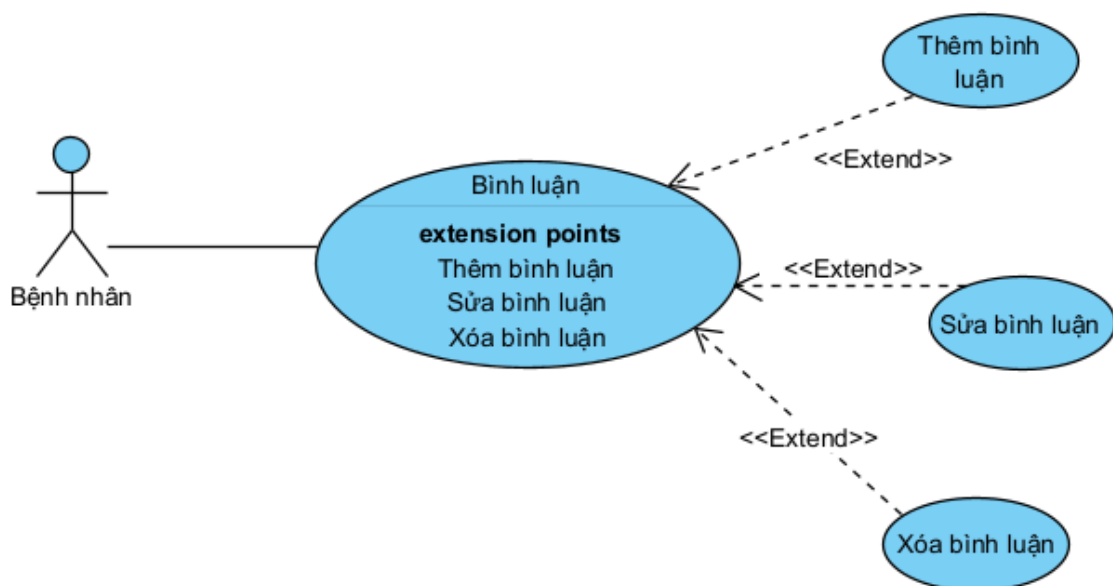


### 3.1.6 Use case Quản lý thông tin cá nhân



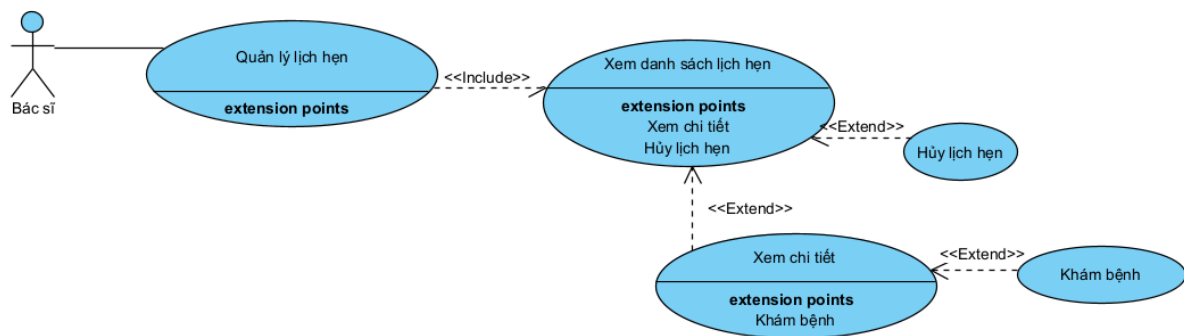
Hình 3.6 Use case Quản lý thông tin cá nhân

### 3.1.7 Use case Quản lý bình luận (Bệnh nhân)



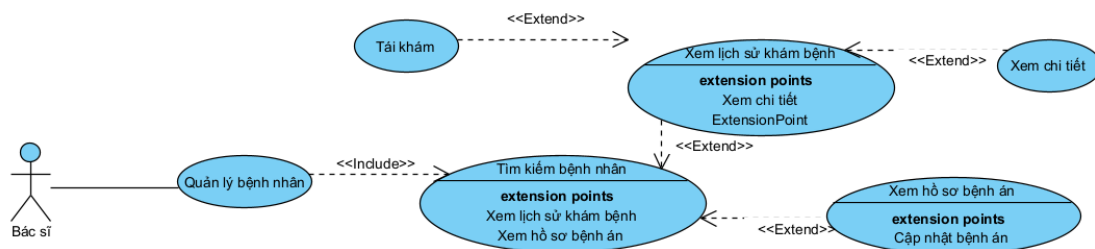
Hình 3.7 Use case Quản lý bình luận

### 3.1.8 Use case Quản lý lịch hẹn (Bác sĩ)



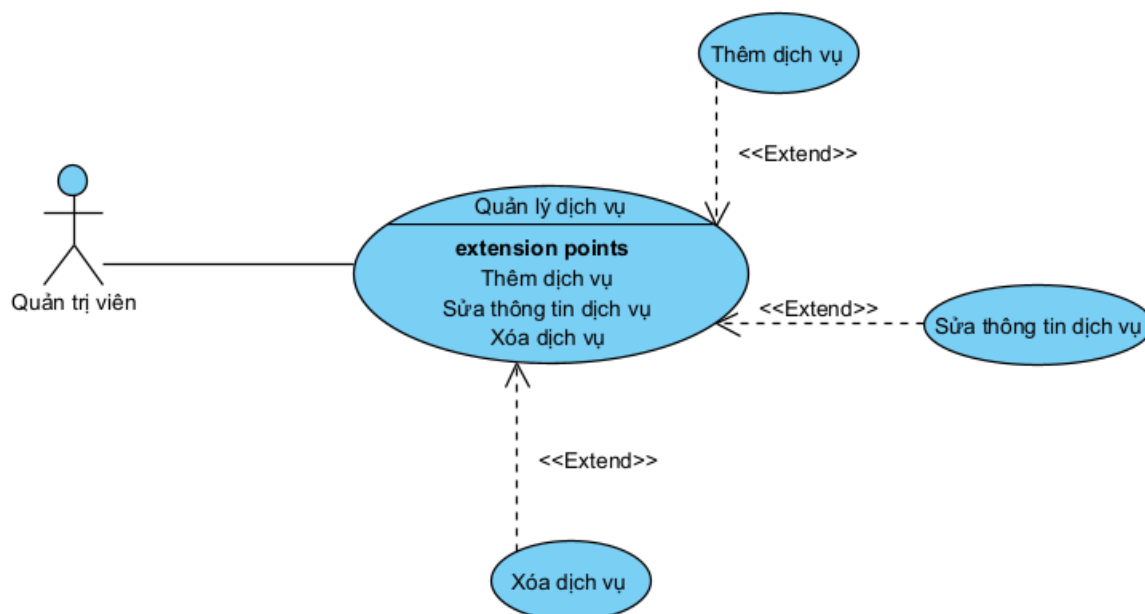
Hình 3.8 Use case Quản lý lịch hẹn

### 3.1.9 Use case Quản lý bệnh nhân (Bác sĩ)



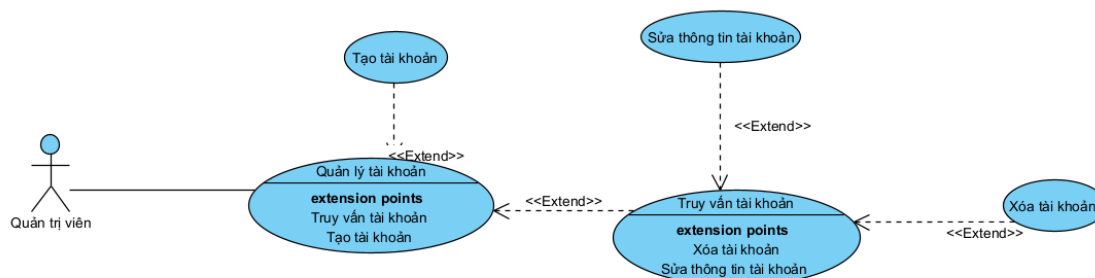
Hình 3.9 Use case Quản lý bệnh nhân

### 3.1.10 Use case Quản lý dịch vụ phòng khám (Quản trị viên)



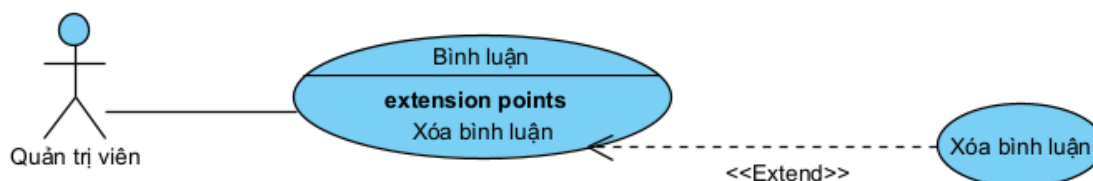
Hình 3.10 Use case Quản lý dịch vụ phòng khám

### 3.1.11 Use case Quản lý tài khoản người dùng (Quản trị viên)



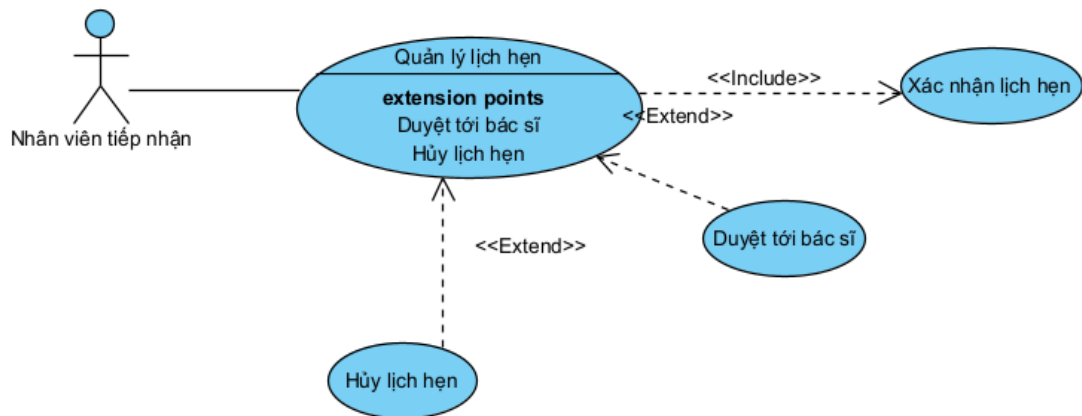
Hình 3.11 Use case Quản lý tài khoản người dùng

### 3.1.12 Use case Quản lý bình luận (Quản trị viên)



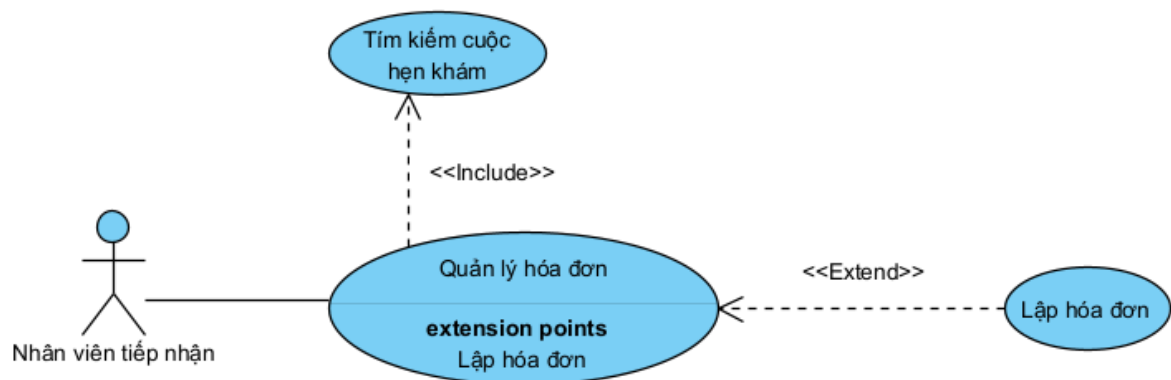
Hình 3.12 Use case Quản lý bình luận

### 3.1.13 Use case Quản lý lịch hẹn (NVTN)



Hình 3.13 Use case Quản lý lịch hẹn (NVTN)

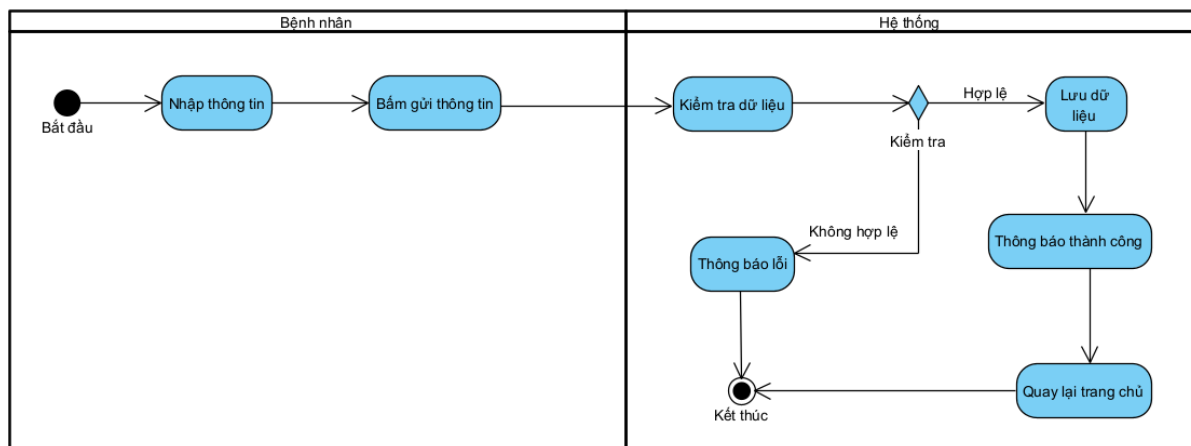
### 3.1.14 Use case Quản lý hóa đơn (NVTN)



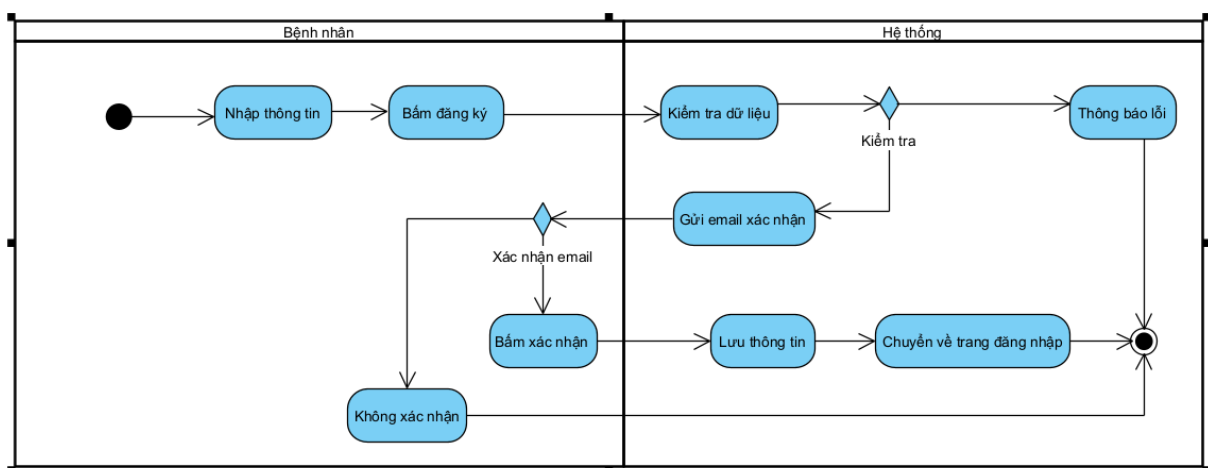
Hình 3.14 Use case Quản lý hóa đơn

## 3.2 Biểu đồ hoạt động (Activity Diagram)

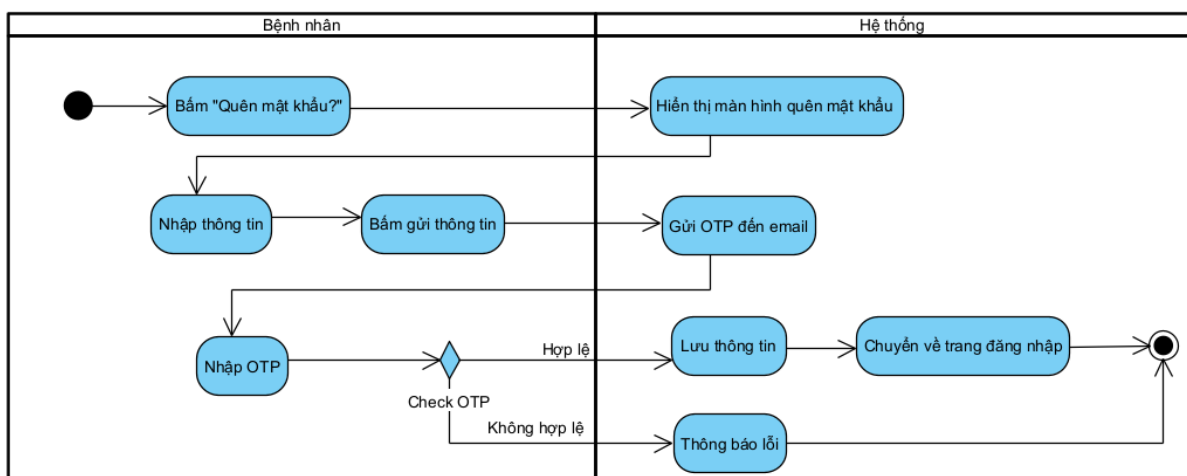
Các biểu đồ hoạt động trong hệ thống:



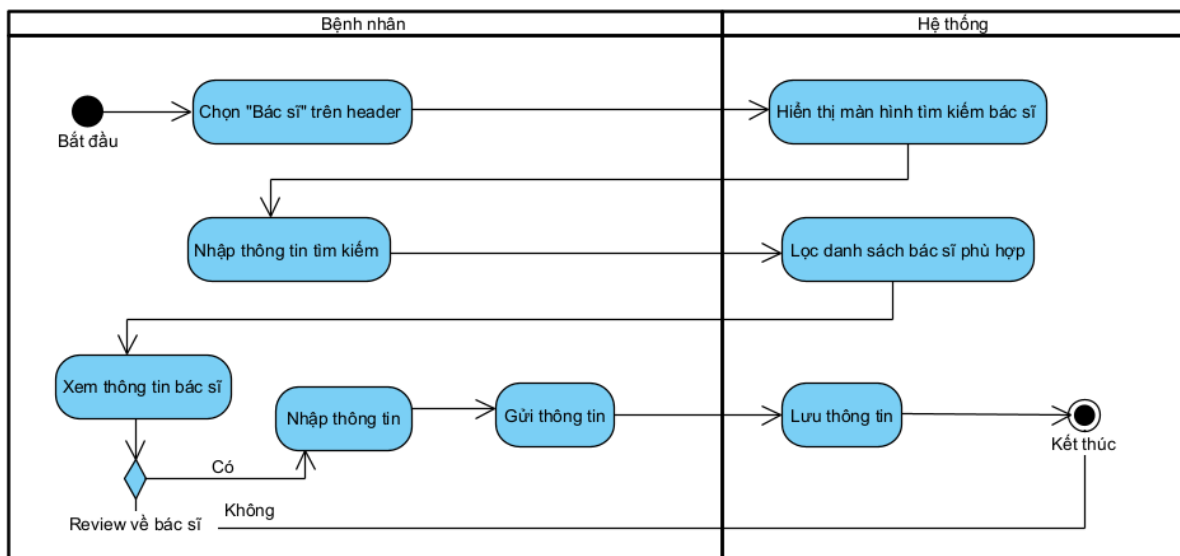
Hình 3.15 Biểu đồ hoạt động Đăng nhập



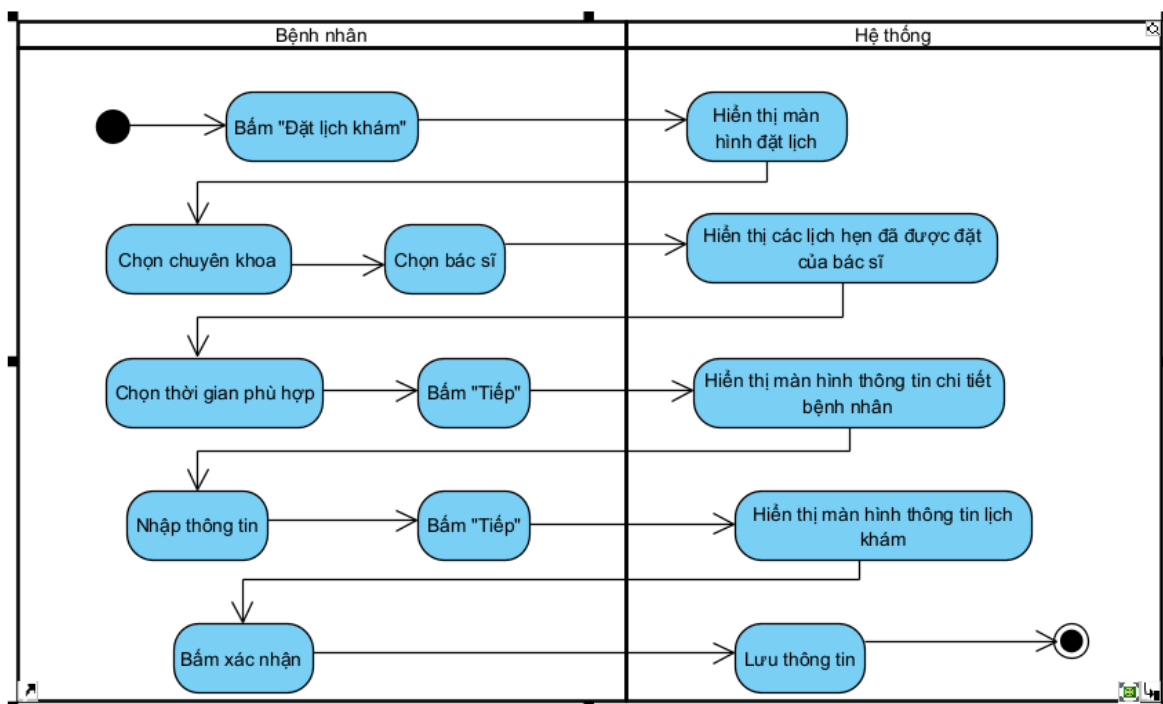
Hình 3.16 Biểu đồ hoạt động Đăng ký



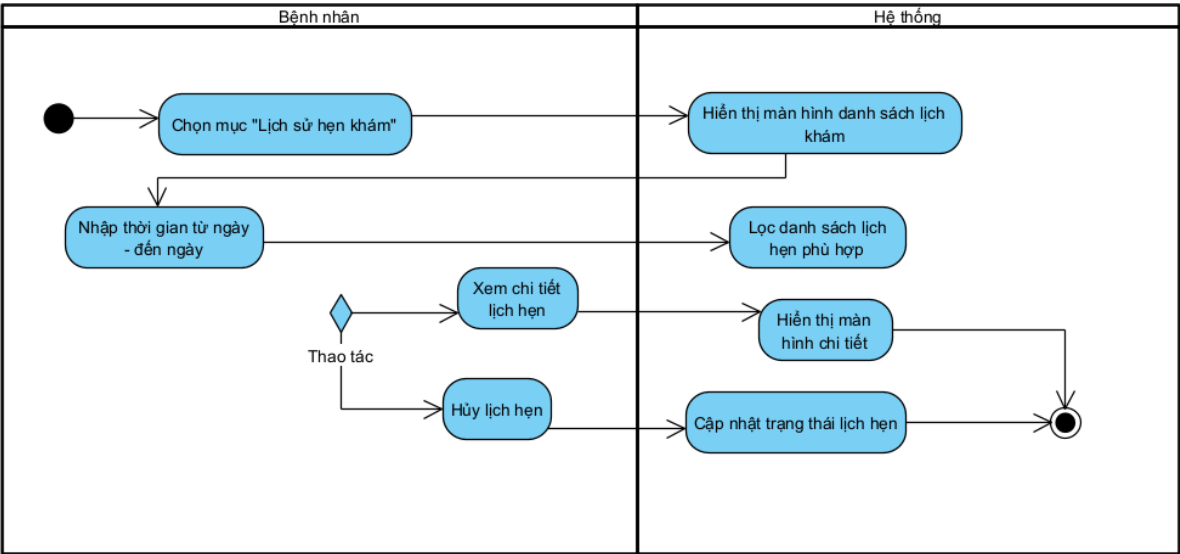
Hình 3.17 Biểu đồ hoạt động Quên mật khẩu



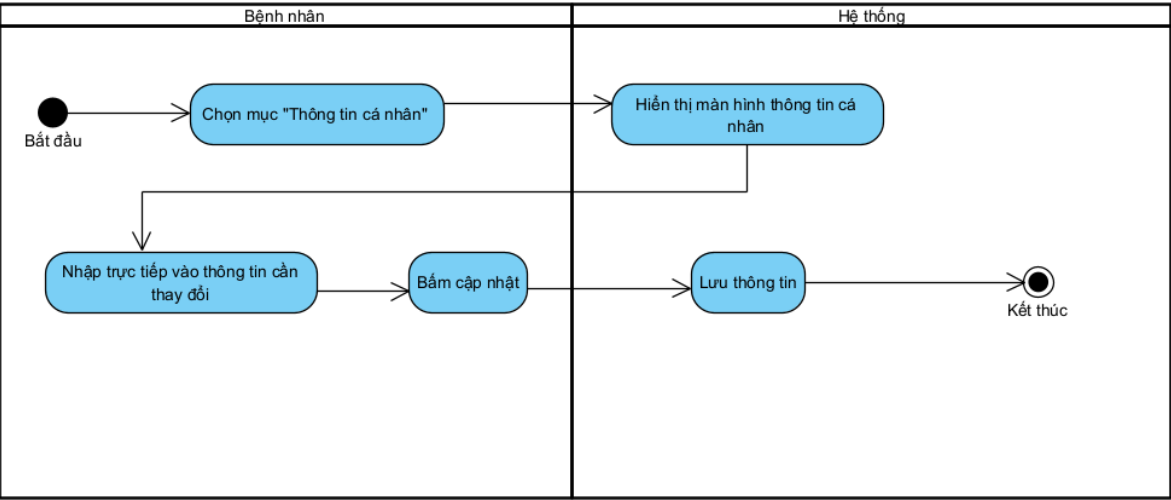
Hình 3.18 Biểu đồ hoạt động Tìm kiếm bác sĩ



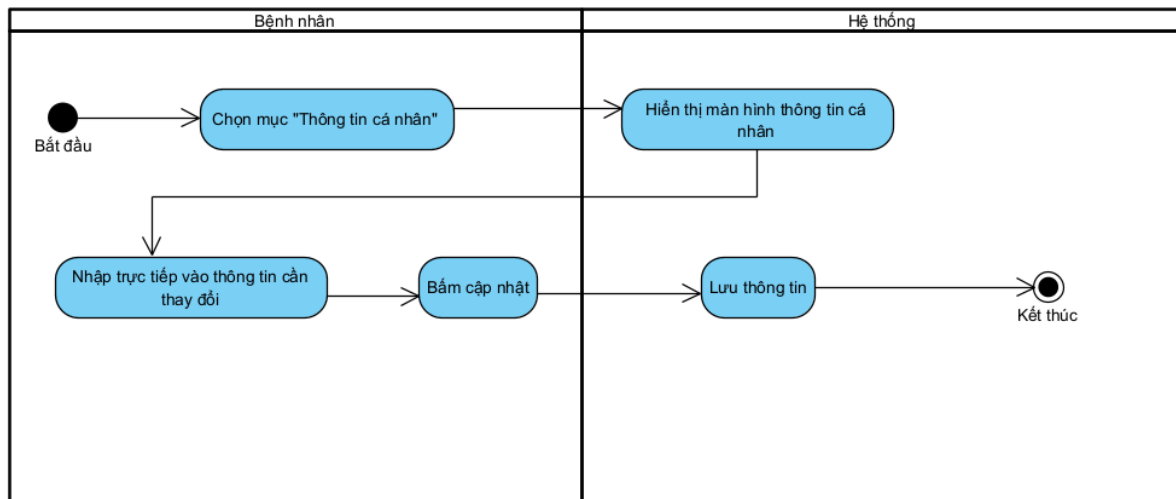
Hình 3.19 Biểu đồ hoạt động Đặt lịch khám



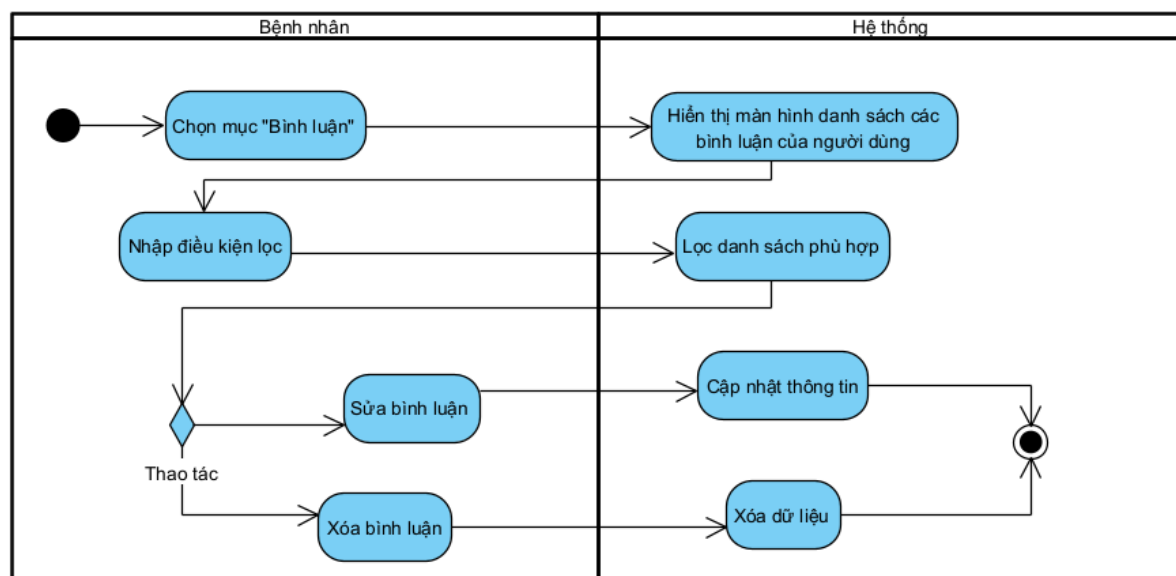
Hình 3.20 Biểu đồ hoạt động Quản lý lịch khám



Hình 3.21 Biểu đồ hoạt động Quản lý thông tin cá nhân

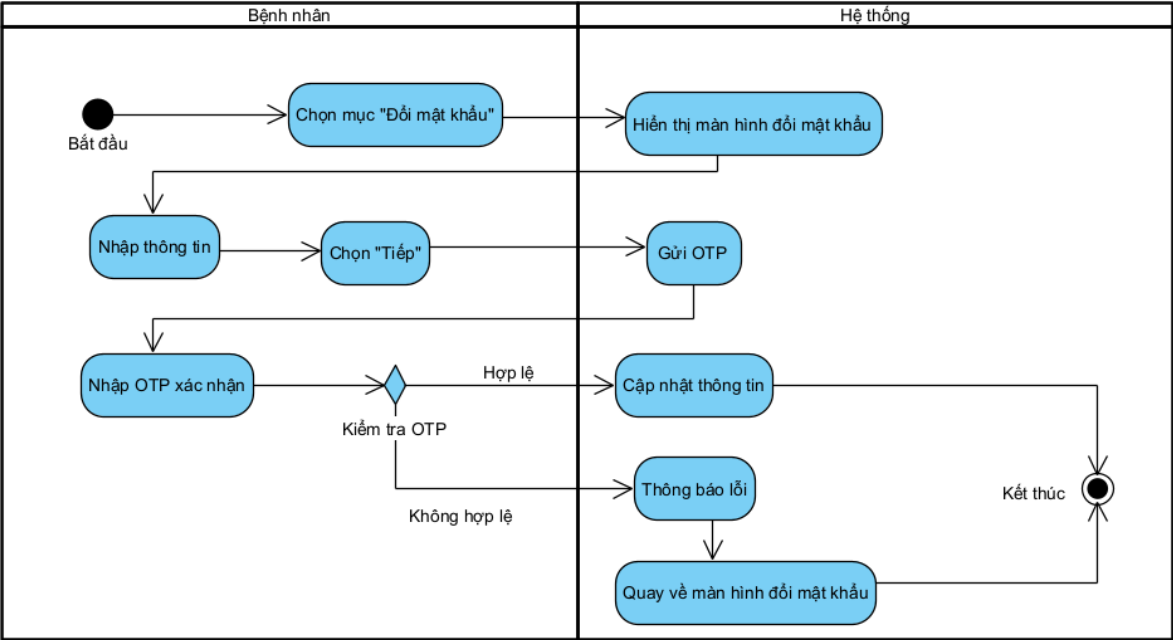


Hình 3.22 Biểu đồ hoạt động Quản lý thông tin cá nhân

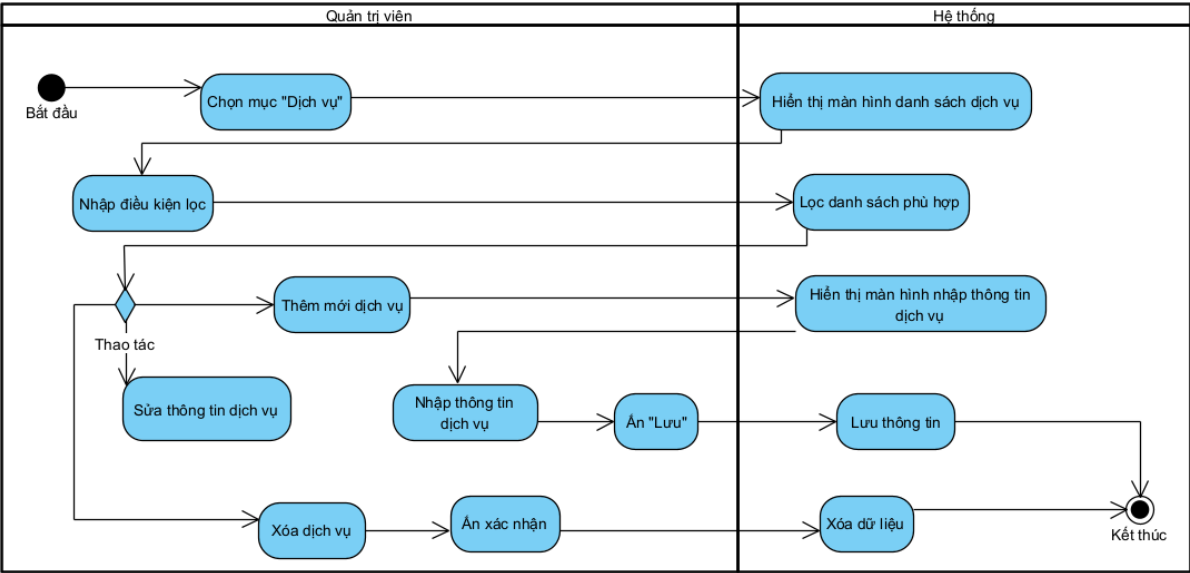


Hình 3.23 Biểu đồ hoạt động Quản lý bình luận

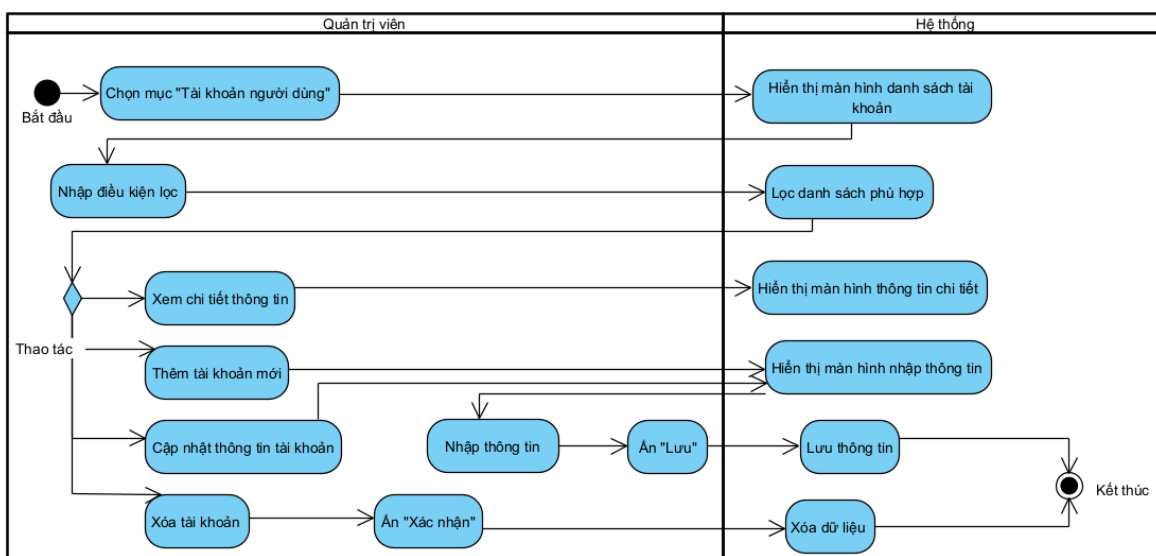




Hình 3.24 Biểu đồ hoạt động Đổi mật khẩu



Hình 3.25 Biểu đồ hoạt động Quản lý dịch vụ



Hình 3.26 Biểu đồ hoạt động Quản lý tài khoản người dùng

### 3.3 Kịch bản chuẩn và ngoại lệ

Use case Đăng ký:

Tên Use Case	Đăng ký
Actor	Bệnh nhân
Mục tiêu	Khi người dùng có nhu cầu truy cập để thực hiện các chức năng trên hệ thống nhưng chưa có tài khoản, người dùng cần đăng ký tài khoản
Tiền điều kiện	Người dùng có email và số điện thoại
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị giao diện ký</li> <li>2. Người dùng nhập các thông tin của tài khoản: tên, email, số điện thoại, mật khẩu</li> <li>3. Hệ thống kiểm tra định dạng các thông tin</li> <li>4. Người dùng sửa các thông tin cho đúng định dạng và gửi thông tin</li> <li>5. Hệ thống kiểm tra thông tin và gửi email xác nhận</li> <li>6. Người dùng truy cập email và xác nhận</li> <li>7. Hệ thống điều hướng về trang đăng nhập, lưu thông tin tài khoản vào CSDL</li> </ol>

Ngoại lệ	5.1 Hệ thống báo lỗi do email đã được đăng ký trước đó 5.2 Quay lại bước 2 6.1 Người dùng không xác nhận email hoặc xác nhận khi quá thời hạn 6.2 Quay lại bước 2
----------	--

Bảng 3.1 Kịch bản use case Đăng ký

Use case Đăng nhập:

Tên Use Case	Đăng nhập
Actor	Bệnh nhân/Bác sĩ/Nhân viên tiếp nhận/Quản trị viên
Mục tiêu	Khi người dùng có nhu cầu truy cập để thực hiện các chức năng trên hệ thống, người dùng phải đăng nhập vào hệ thống để xác thực
Tiền điều kiện	Người dùng phải có tài khoản truy cập hệ thống
Luồng hoạt động cơ bản	1. Hệ thống hiển thị giao diện đăng nhập 2. Người dùng nhập email và password 3. Hệ thống kiểm tra email và password

	4.Hệ thống thông báo thành công và người dùng đăng nhập thành công vào hệ thống, hiển thị trang chủ hoặc màn hình trước đó có thông báo lỗi do chưa đăng nhập nên bị đẩy ra màn đăng nhập
Ngoại lệ	4.1 Hệ thống báo lỗi do người dung nhập sai thông tin đăng nhập 4.2 Quay lại bước 2

Bảng 3.2 Kịch bản use case Đăng nhập

Use case Quên mật khẩu:

Tên Use Case	Quên mật khẩu
Actor	Bệnh nhân/ Bác sĩ/ Nhân viên tiếp nhận
Mục tiêu	Khi người dùng đã có tài khoản nhưng quên mật khẩu, người dùng có thể tạo lại mật khẩu mới
Tiền điều kiện	Người dùng đã có tài khoản được đăng ký trước đó
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị giao diện quên mật khẩu</li> <li>2. Người dùng nhập email và bấm gửi OTP</li> <li>3. Hệ thống gửi OTP đến email</li> <li>4. Người dùng lấy OTP và nhập</li> <li>5. Hệ thống kiểm tra OTP</li> <li>6. Người dùng nhập mật khẩu mới và bấm cập nhật</li> <li>7. Hệ thống cập nhật mật khẩu mới và điều hướng về trang đăng nhập</li> </ol>

Ngoại lệ	2.1 Người dùng nhập email chưa được đăng ký trước đó 2.2 Hệ thống thông báo lỗi và quay lại bước 2 4.1 Người dùng nhập OTP đã quá hạn 4.2 Hệ thống thông báo lỗi và quay lại bước 2
----------	--

*Bảng 3.3 Kịch bản use case Quên mật khẩu*

*Use case Tìm kiếm bác sĩ:*

Tên Use Case	Tìm kiếm bác sĩ
Actor	Bệnh nhân
Mục tiêu	Người dùng muốn tra cứu thông tin của bác sĩ
Tiền điều kiện	Người dùng truy cập hệ thống
Luồng hoạt động cơ bản	1. Hệ thống hiển thị giao diện tra cứu bác sĩ 2. Người dùng có thể chọn các điều kiện tra cứu để lọc danh sách bác sĩ 3. Hệ thống lọc danh sách phù hợp với điều kiện 4. Người dùng xem thông tin chi tiết của bác sĩ 5. Hệ thống hiển thị màn hình thông tin chi tiết
Ngoại lệ	

*Bảng 3.4 Kịch bản use case Tìm kiếm bác sĩ*

*Use case Đặt lịch khám:*

Tên Use Case	Đặt lịch khám
Actor	Bệnh nhân
Mục tiêu	Người dùng đặt lịch hẹn khám thành công
Tiền điều kiện	Người dùng đã đăng nhập và hệ thống
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị giao diện đặt lịch khám</li> <li>2. Người dùng chọn chuyên khoa phù hợp với bệnh cần khám</li> <li>3. Hệ thống lọc danh sách bác sĩ của chuyên khoa đó</li> <li>4. Người dùng chọn bác sĩ</li> <li>5. Hệ thống hiển thị khung lịch hẹn có ngày tháng và thời gian, các lịch hẹn đã được đặt của bác sĩ đó</li> <li>6. Người dùng chọn khung thời gian phù hợp và ấn tiếp tục</li> <li>7. Hệ thống hiển thị màn hình thông tin chi tiết của người dùng</li> <li>8. Người dùng chỉnh sửa thông tin và ấn tiếp tục</li> <li>9. Hệ thống chuyển sang màn hình xác nhận lịch hẹn</li> <li>10. Người dùng kiểm tra lại thông tin lịch hẹn và ấn xác nhận (Có thể quay lại các trang trước đó để sửa thông tin)</li> <li>11. Hệ thống thông báo thành công và lưu thông tin lịch hẹn vào CSDL</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>4.1 Người dùng đặt lịch hẹn từ chức năng tìm kiếm bác sĩ</li> <li>4.2 Hệ thống chuyển sang màn hình đặt lịch khám và hiển thị khung lịch hẹn có ngày tháng và thời gian, các lịch hẹn đã được đặt của bác sĩ đó</li> <li>4.3 Tiếp tục bước 6</li> </ol>

*Bảng 3.5 Kịch bản use case Đặt lịch khám*

*Use case Đổi mật khẩu:*

Tên Use Case	Đổi mật khẩu
Actor	Bệnh nhân/ Bác sĩ/ Nhân viên tiếp nhận/ Quản trị viên
Mục tiêu	Người dùng đổi mật khẩu thành công
Tiền điều kiện	Người dùng đã đăng nhập và hệ thống
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị giao diện đổi mật khẩu</li> <li>2. Người dùng nhập email và bấm gửi OTP</li> <li>3. Hệ thống gửi OTP đến email</li> <li>4. Người dùng lấy OTP và nhập</li> <li>5. Hệ thống kiểm tra OTP, nếu hợp lệ chuyển màn hình nhập mật khẩu</li> <li>6. Người dùng nhập mật khẩu cũ, mật khẩu mới và bấm cập nhật</li> <li>7. Hệ thống kiểm tra 2 mật khẩu cũ và mới, nếu khớp thì cập nhật vào CSDL</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>3.1 Hệ thống báo lỗi email không đúng với email đang đăng nhập hiện tại</li> <li>3.2 Quay lại bước 2</li> <li>5.1 Hệ thống báo lỗi do OTP nhập không đúng hoặc hết hạn</li> <li>5.2 Quay lại bước 2</li> <li>7.1 hệ thống báo lỗi do 2 mật khẩu không khớp nhau</li> <li>7.2 Quay lại bước 6</li> </ol>

*Bảng 3.6 Kịch bản use case Đổi mật khẩu*

*Use case Cập nhật thông tin cá nhân:*

Tên Use Case	Cập nhật thông tin cá nhân
Actor	Bệnh nhân/ Bác sĩ/ Nhân viên tiếp nhận/ Quản trị viên
Mục tiêu	Người dùng cập nhật lại thông tin cá nhân của mình
Tiền điều kiện	Người dùng đã đăng nhập và hệ thống
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình thông tin cá nhân chi tiết của người dùng</li> <li>2. Người dùng sửa các thông tin mong muốn và ấn cập nhật</li> <li>3. Hệ thống cập nhật lại thông tin vào trong CSDL</li> </ol>
Ngoại lệ	

*Bảng 3.7 Kịch bản use case Cập nhật thông tin cá nhân**Use case Quản lý bình luận:*

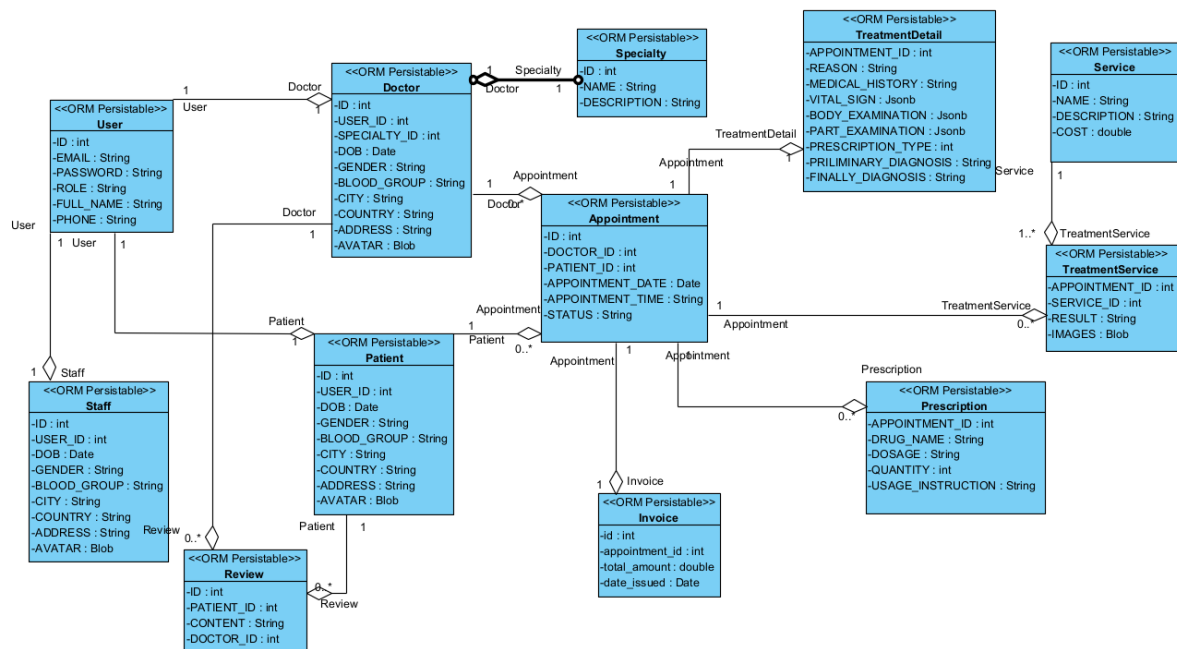
Tên Use Case	Quản lý bình luận
Actor	Bệnh nhân
Mục tiêu	Người dùng xem, sửa, xóa các bình luận của mình
Tiền điều kiện	Người dùng đã đăng nhập và hệ thống
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình danh sách các bình luận của người dùng được phân loại review về phòng khám và review về bác sĩ</li> <li>2. Người dùng có thể nhập điều kiện để lọc danh sách</li> <li>3. Hệ thống lọc danh sách phù hợp điều kiện</li> <li>4. Người dùng có thể thực hiện thao tác với bình luận: <ol style="list-style-type: none"> <li>4.1 Người dùng chọn Sửa bình luận <ol style="list-style-type: none"> <li>4.1.1 Hệ thống hiển thị popup thông tin bình luận cho phép sửa</li> <li>4.1.2 Người dùng sửa nội dung và ấn lưu</li> <li>4.1.3 Hệ thống đóng popup, lưu thông tin vào CSDL</li> </ol> </li> <li>4.2 Người dùng chọn Xóa bình luận <ol style="list-style-type: none"> <li>4.2.1 Hệ thống hiện thông báo có chắc chắn muốn xóa</li> <li>4.2.2 Người dùng ấn Đồng ý/Quay lại</li> <li>4.2.3 Hệ thống xóa dữ liệu nếu người dùng ấn Đồng ý</li> </ol> </li> </ol> </li> <li>5. Hệ thống cập nhật lại danh sách</li> </ol>



Ngoại lệ	
----------	--

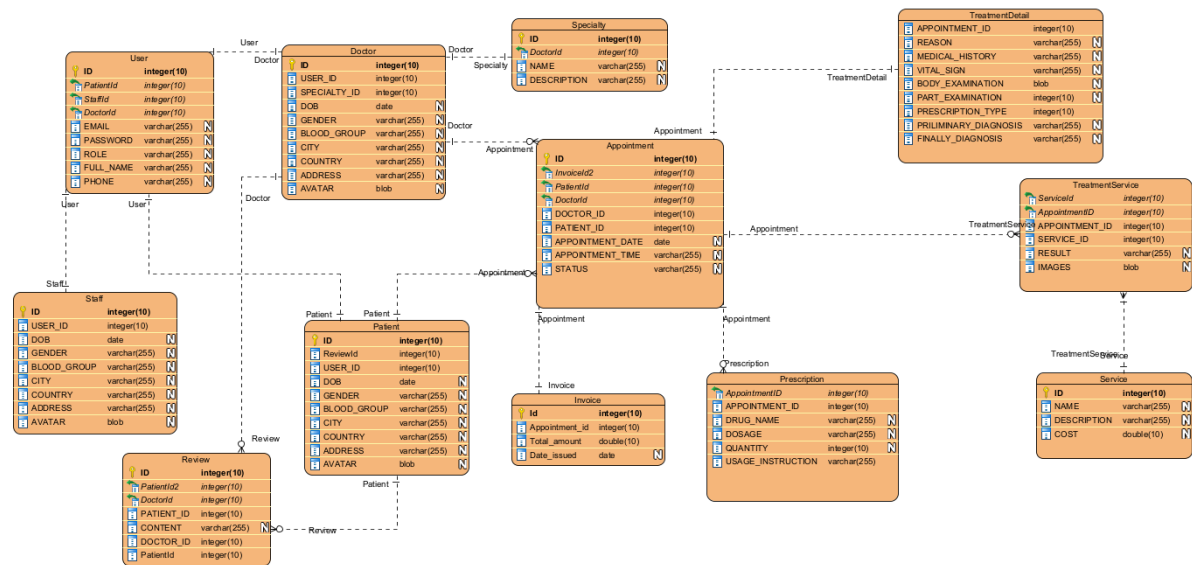
Bảng 3.8 Kịch bản use case Quản lý bình luận

3.4 Biểu đồ lớp phân tích (Class Diagram)



Hình 3.27 (Biểu đồ lớp phân tích)

### 3.5 Biểu đồ quan hệ thực thể (Entity-Relationship Diagram)



Hình 3.28 (Biểu đồ Quan hệ thực thể)

### 3.6 Kết luận chương III

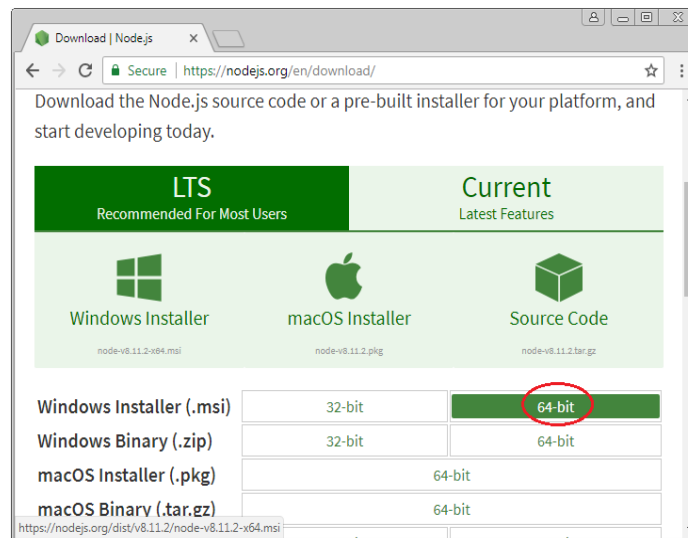
## CHƯƠNG IV. CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG

(Chèn giới thiệu mở chương ở đây).

### 4.1 Môi trường triển khai ứng dụng

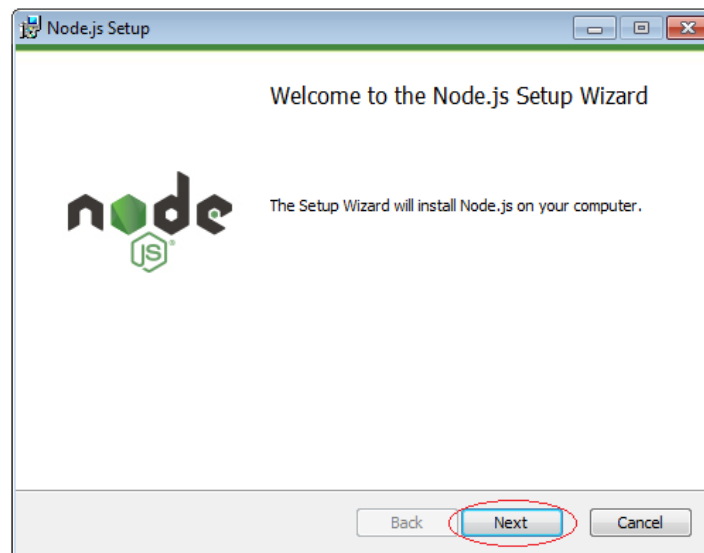
#### 4.1.1 Cài đặt ReactJS cho giao diện

- Cài đặt NodeJS: Link download <https://nodejs.org/en/download>

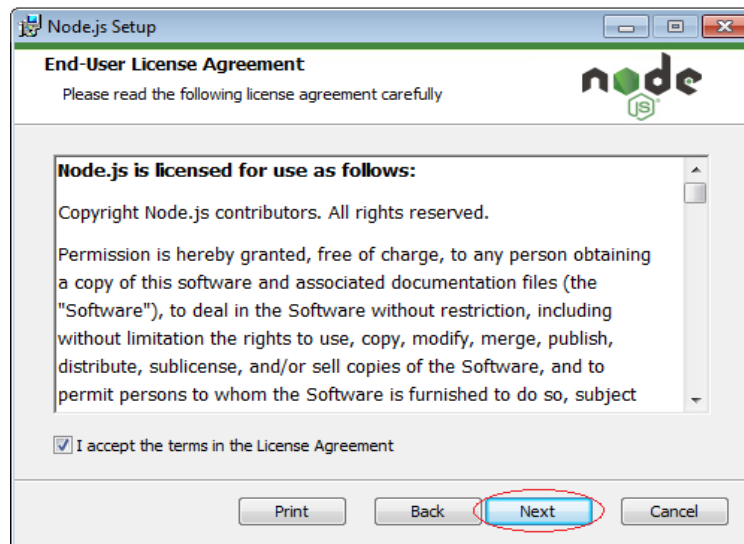


Hình 4.1 Cài đặt NodeJS bước 1

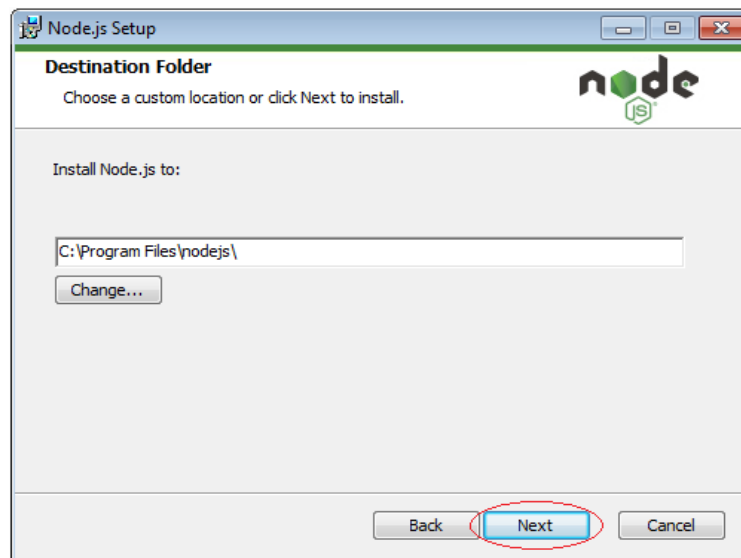
Sau khi tải xuống thành công, mở file vừa tải và cài đặt theo hướng dẫn.



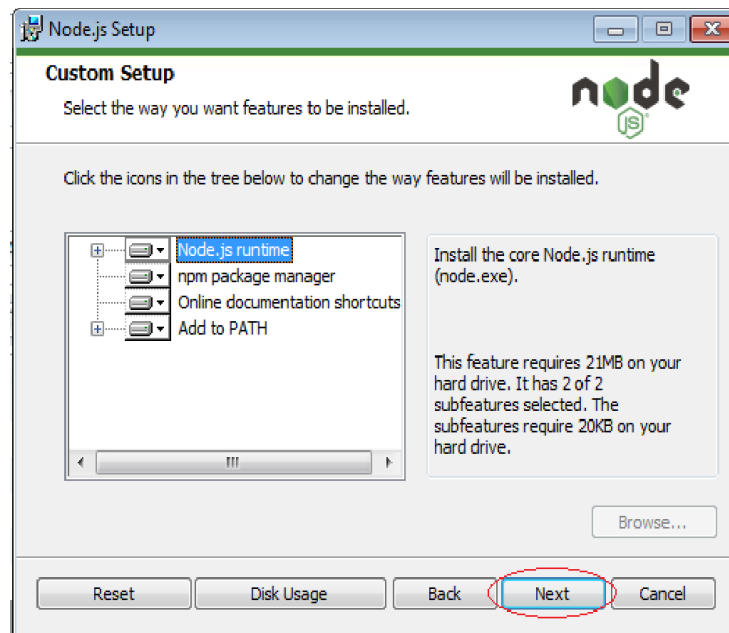
Hình 4.2 Cài đặt NodeJS bước 2



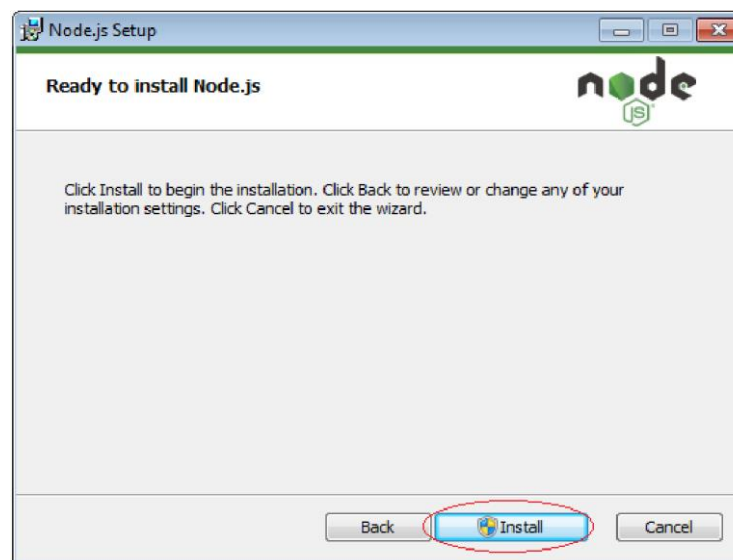
Hình 4.3 Cài đặt NodeJS bước 3



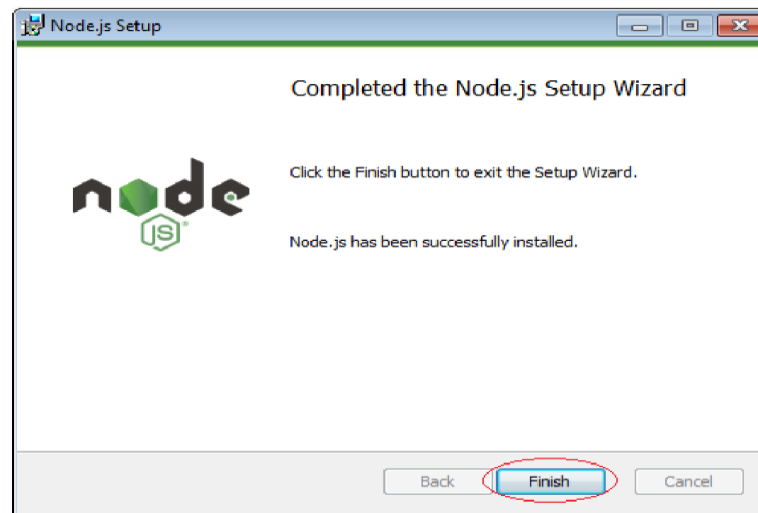
Hình 4.4 Cài đặt NodeJS bước 4



Hình 4.5 Cài đặt NodeJS bước 5



Hình 4.6 Cài đặt NodeJS bước 6

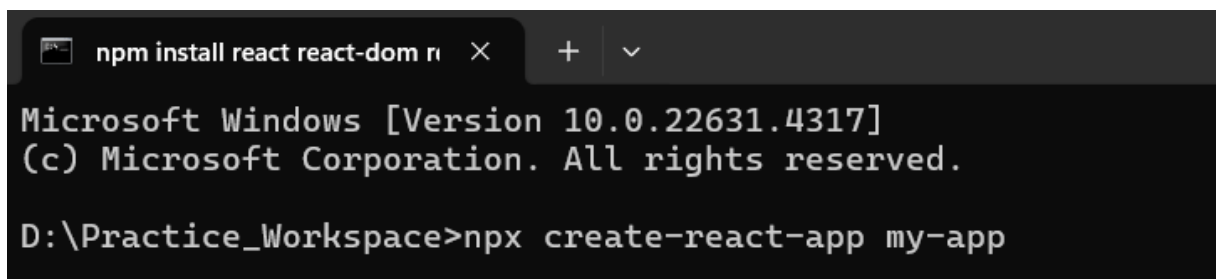


Hình 4.7 Cài đặt NodeJS thành công

- Khởi tạo dự án ReactJS:

Trước tiên, vào thư mục chứa dự án, mở terminal lên và gõ dòng lệnh:

`npx create-react-app my-app` (Trong đó my-app là tên dự án muốn khởi tạo)



Hình 4.8 Khởi tạo dự án ReactJS

Sau khi tạo dự án thành công terminal sẽ hiển thị như hình:

```
Success! Created my-app at D:\Practice_Workspace\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!
D:\Practice_Workspace>df
```

Hình 4.9 Khởi tạo dự án ReactJS thành công

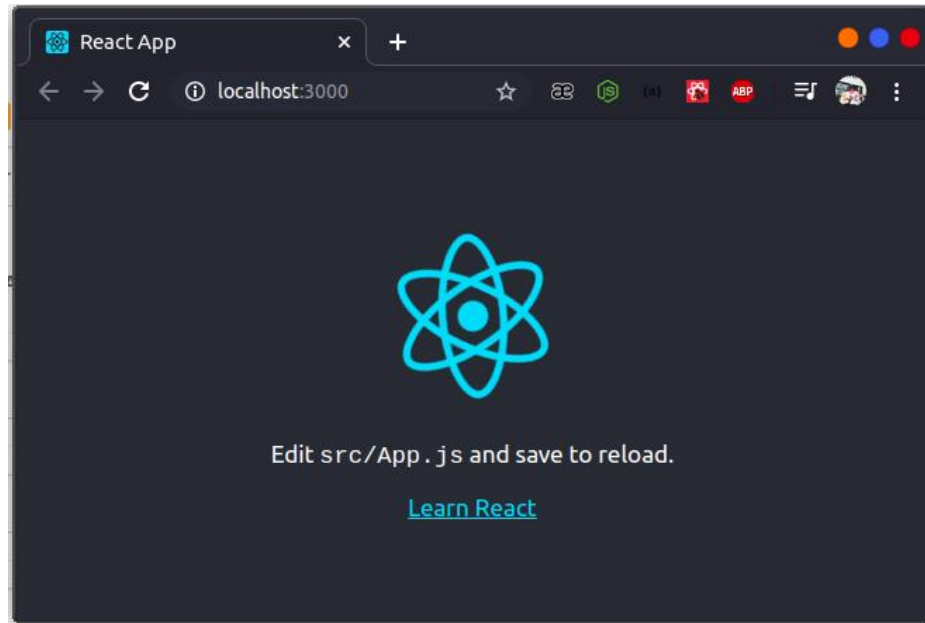
- Sau khi tạo xong dự án, mở dự án lên và chạy dự án bằng lệnh: `npm run start`

```
Hi@LAPTOP-THIENNN MINGW64 /d/Practice_Workspace/my-app (master)
$ npm run start

> my-app@0.1.0 start
> react-scripts start
```

Hình 4.10 Chạy dự án

- Lúc này, ReactJS sẽ khởi chạy dự án mặc định ở port 3000, chúng ta có thể mở trình duyệt và truy cập đường dẫn <http://localhost:3000>



Hình 4.11 Chạy dự án thành công

#### 4.1.2 Cài đặt Java Spring Boot cho server

#### 4.1.3 Cài đặt PostgreSQL cho cơ sở dữ liệu

### 4.2 Kiểm thử hệ thống

....

....



## **CHƯƠNG ... KẾT LUẬN**

### **...1 Kết quả đạt được**

....

....

### **...2 Hạn chế của hệ thống**

....

....

### **...3 Định hướng phát triển hệ thống**

....

....

## DANH MỤC TÀI LIỆU THAM KHẢO

### Tài liệu, giáo trình:

[1] Ví dụ: PGS.TS Trần Đình Quế, Phân tích và thiết kế hệ thống thông tin, Học viện Công nghệ Bưu chính Viễn thông, 2014.

[...] ...

[...] ...

### Trang web:

[1] Ví dụ: <https://www.abc.xyz>

[2] ...

[3] ...