

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 phòng khám:
Số hóa dịch vụ y tế và lịch hẹn

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

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc tới thầy TS. Nguyễn Tất Thắng, người đã tận tình hướng dẫn và đồng hành cùng em trong suốt quá trình thực hiện đồ án. Những ý kiến và kinh nghiệm quý báu từ thầy không chỉ giúp em hoàn thiện đồ án mà còn mang lại những bài học ý nghĩa cho hành trình học tập và sự nghiệp sau này.

Em cũng xin bày tỏ lòng biết ơn tới các thầy, cô trong Học viện nói chung và khoa Công nghệ Thông tin 1 nói riêng. Sự nhiệt huyết và tận tâm trong từng bài giảng đã truyền cảm hứng và tạo điều kiện tốt nhất để em có thể học tập, nghiên cứu và phát triển bản thân trong suốt thời gian tại Học viện Công nghệ Bưu chính Viễn thông.

Con xin chân thành cảm ơn bố mẹ và gia đình, những người luôn là điểm tựa vững chắc, đã không ngừng động viên, ủng hộ và tạo mọi điều kiện để con tập trung vào việc học tập. Sự yêu thương và hỗ trợ của mọi người chính là nguồn động lực lớn lao giúp con vượt qua mọi khó khăn.

Cuối cùng, em xin gửi những lời chúc tốt đẹp nhất tới các thầy cô và bạn bè trong khóa bảo vệ tốt nghiệp lần này. Chúc mọi người luôn gặp nhiều niềm vui, may mắn và gặt hái nhiều thành công trong tương lai.

Em xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2024

Sinh viên thực hiện

Nguyễn Ngọc Thiện

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 Thực trạng hiện nay	2
1.1.2 Đặc điểm của hệ thống quản lý phòng khám	3
1.1.3 Một số hệ thống quản lý phòng khám phổ biến hiện nay	4
1.2 Mục tiêu, phạm vi, chức năng của hệ thống	7
1.2.1 Mục tiêu	7
1.2.2 Chức năng hệ thống	7
1.3 Công nghệ sử dụng trong hệ thống	9
1.3.1 React JS	10
1.3.2 Java Spring Boot	11
1.3.3 PostgreSQL.....	12
1.4 Kết luận Chương I	14
CHƯƠNG II. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	15
2.1 Mô tả bài toán	15
2.1.1 Tổng quan quy trình khám bệnh của hệ thống:	15
2.1.2 Quy trình đặt lịch khám của bệnh nhân:.....	16
2.1.3 Quy trình khám bệnh của bác sĩ:.....	16
2.2 Biểu đồ Use case (Usecase diagram)	16
2.2.1 Use case Tổng quan hệ thống.....	16
2.2.2 Use case Đăng nhập	18
2.2.3 Use case Tìm kiếm bác sĩ (Bệnh nhân)	18
2.2.4 Use case Đặt lịch khám (Bệnh nhân)	19
2.2.5 Use case Quản lý lịch hẹn (Bệnh nhân).....	19
2.2.6 Use case Quản lý thông tin cá nhân.....	20

2.2.7 Use case Quản lý bình luận (Bệnh nhân)	20
2.2.8 Use case Quản lý lịch hẹn (Bác sĩ)	20
2.2.9 Use case Quản lý bệnh nhân (Bác sĩ)	21
2.2.10 Use case Quản lý danh mục (QTV)	21
2.2.11 Use case Quản lý người dùng (QTV)	22
2.2.12 Use case Quản lý bình luận (QTV)	22
2.2.13 Use case Quản lý lịch hẹn (NVTN)	23
2.2.14 Use case Quản lý hóa đơn (NVTN)	23
2.3 Biểu đồ hoạt động (Activity Diagram)	23
2.4 Kịch bản chuẩn và ngoại lệ	29
2.5 Biểu đồ lớp phân tích (Class Diagram)	35
2.6 Biểu đồ quan hệ thực thể (Entity-Relationship Diagram)	35
2.7 Kết luận chương II	37
CHƯƠNG III. CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG	38
3.1 Môi trường triển khai ứng dụng	38
3.1.1 Cài đặt ReactJS cho giao diện	38
3.1.2 Cài đặt Java Spring Boot cho server	42
3.1.3 Cài đặt PostgreSQL cho cơ sở dữ liệu	45
3.2 Giao diện hệ thống	46
CHƯƠNG IV. KẾT LUẬN	47
4.1 Kết quả đạt được	47
4.2 Hạn chế của hệ thống	47
4.3 Định hướng phát triển hệ thống	48
DANH MỤC TÀI LIỆU THAM KHẢO	49

(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

TỪ VIẾT TẮT VÀ THUẬT NGỮ	Ý NGHĨA
CSDL	Cơ sở dữ liệu
NVTV	Nhân viên tiếp nhận
QTV	Quản trị viên

DANH SÁCH HÌNH VẼ

Hình 1.1 Thách thức trong quy trình chăm sóc sức khỏe	3
Hình 1.2 Kiến trúc hệ thống	10
Hình 1.3 Chức năng chính của ReactJS trong hệ thống	11
Hình 1.4 Chức năng chính của Spring Boot trong hệ thống	12
Hình 2.1 Quy trình khám bệnh.....	15
Hình 2.2 Use case Tổng quan hệ thống.....	17
Hình 2.3 Use case Đăng nhập	18
Hình 2.4 Use case Tìm kiếm bác sĩ	18
Hình 2.5 Use case Đặt lịch khám	19
.....	19
Hình 2.6 Use case Quản lý lịch hẹn (Bệnh nhân).....	19
Hình 2.7 Use case Quản lý thông tin cá nhân	20
Hình 2.8 Use case Quản lý bình luận (Bệnh nhân).....	20
Hình 2.9 Use case Quản lý lịch hẹn (Bác sĩ).....	21
Hình 2.10 Use case Quản lý bệnh nhân.....	21
Hình 2.11 Use case Quản lý danh mục	22
Hình 2.12 Use case Quản lý tài khoản người dùng.....	22
Hình 2.13 Use case Quản lý bình luận (QTV)	22
Hình 2.14 Use case Quản lý lịch hẹn (NVTN).....	23
Hình 2.15 Use case Quản lý hóa đơn.....	23
Hình 2.16 Biểu đồ hoạt động Đăng nhập	24
Hình 2.17 Biểu đồ hoạt động Đăng ký.....	24
Hình 2.18 Biểu đồ hoạt động Quên mật khẩu	24
Hình 2.19 Biểu đồ hoạt động Tìm kiếm bác sĩ.....	25
Hình 2.20 Biểu đồ hoạt động Đặt lịch khám.....	25
Hình 2.21 Biểu đồ hoạt động Quản lý lịch khám	26
Hình 2.22 Biểu đồ hoạt động Quản lý thông tin cá nhân.....	26
Hình 2.23 Biểu đồ hoạt động Quản lý thông tin cá nhân.....	27
Hình 2.24 Biểu đồ hoạt động Quản lý bình luận.....	27
Hình 2.25 Biểu đồ hoạt động Đổi mật khẩu	28

Hình 2.26 Biểu đồ hoạt động Quản lý dịch vụ.....	28
Hình 2.27 Biểu đồ hoạt động Quản lý tài khoản người dùng.....	29
Hình 2.28 (Biểu đồ lớp phân tích).....	35
Hình 2.29 (Biểu đồ Quan hệ thực thể).....	36
Bảng 2.9 Mô tả cơ sở dữ liệu.....	36
Hình 3.1 Cài đặt NodeJS bước 1	38
Hình 3.2 Cài đặt NodeJS bước 2	39
Hình 3.3 Cài đặt NodeJS bước 3	39
Hình 3.4 Cài đặt NodeJS bước 4.....	39
Hình 3.5 Cài đặt NodeJS bước 5	40
Hình 3.6 Cài đặt NodeJS bước 6.....	40
Hình 3.7 Cài đặt NodeJS thành công.....	41
Hình 3.8 Chạy dự án.....	41
Hình 3.9 Chạy dự án thành công.....	41
Hình 3.10 Tải file JDK.....	42
Hình 3.11 Tải file maven.....	42
Hình 3.12 Tìm kiếm Environment variable.....	43
Hình 3.13 Bảng System Properties.....	43
Hình 3.14 Tạo Variable M2_HOME.....	44
Hình 3.15 Edit Path.....	44
Hình 3.16 Tạo Path Maven Home.....	45
Hình 3.17 Ứng dụng khởi chạy PostgreSQL	45
Hình 3.17 Giao diện PostgreSQL.....	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 Chi tiết chức năng cho từng vai trò.....	8
Bảng 2.1 Kịch bản use case Đăng ký.....	29
Bảng 2.2 Kịch bản use case Đăng nhập	30
Bảng 2.3 Kịch bản use case Quên mật khẩu	31
Bảng 2.4 Kịch bản use case Tìm kiếm bác sĩ.....	31
Bảng 2.5 Kịch bản use case Đặt lịch khám.....	32
Bảng 2.6 Kịch bản use case Đổi mật khẩu	33
Bảng 2.7 Kịch bản use case Cập nhật thông tin cá nhân	34
Bảng 2.8 Kịch bản use case Quản lý bình luận.....	34

(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

Ứng dụng quản lý phòng khám đóng vai trò quan trọng trong việc nâng cao hiệu quả hoạt động và chất lượng dịch vụ y tế tại các cơ sở khám chữa bệnh.

Ứng dụng giúp tự động hóa các quy trình quản lý như đặt lịch khám, theo dõi hồ sơ bệnh án, và xử lý hóa đơn, giảm thiểu sự phụ thuộc vào các phương pháp thủ công truyền thống. Các tính năng như tra cứu thông tin bác sĩ, nhắc nhở lịch hẹn, và quản lý tài khoản người dùng giúp bệnh nhân và nhân viên y tế thao tác dễ dàng, mang lại sự chính xác và minh bạch trong vận hành.

Nội dung của đồ án xây dựng hệ thống quản lý phòng khám bao gồm các phần sau:

Chương I: Giới thiệu hệ thống

Nội dung chương I là khảo sát các nhu cầu sử dụng, các chức năng chính cần có trong hệ thống quản lý phòng khám và xác định các công nghệ sử dụng trong hệ thống

Chương II: Phân tích thiết kế hệ thống

Nội dung chương II trình bày phân tích hệ thống quản lý phòng khám bao gồm biểu đồ use case, biểu đồ hoạt động, kịch bản từng chức năng và trích lớp thực thể, trình bày việc thiết kế hệ thống theo chuẩn UML

Chương III: Cài đặt và kiểm thử hệ thống

Nội dung chương III là hướng dẫn cài đặt hệ thống, chạy và kiểm thử hệ thống

Kết luận

Đánh giá các kết quả đạt được, đưa ra các hạn chế của hệ thống và định hướng phát triển hệ thống trong tương lai

CHƯƠNG I. GIỚI THIỆU HỆ THỐNG

Đối với việc xây dựng bất kỳ một hệ thống nào thì việc đầu tiên cần làm là cần phải khảo sát hiện trạng trong thực tế để biết được nhu cầu người dùng, khảo sát các ứng dụng tương tự hiện nay từ đó xác định được mục tiêu, phạm vi và các chức năng cơ bản cần có trong hệ thống

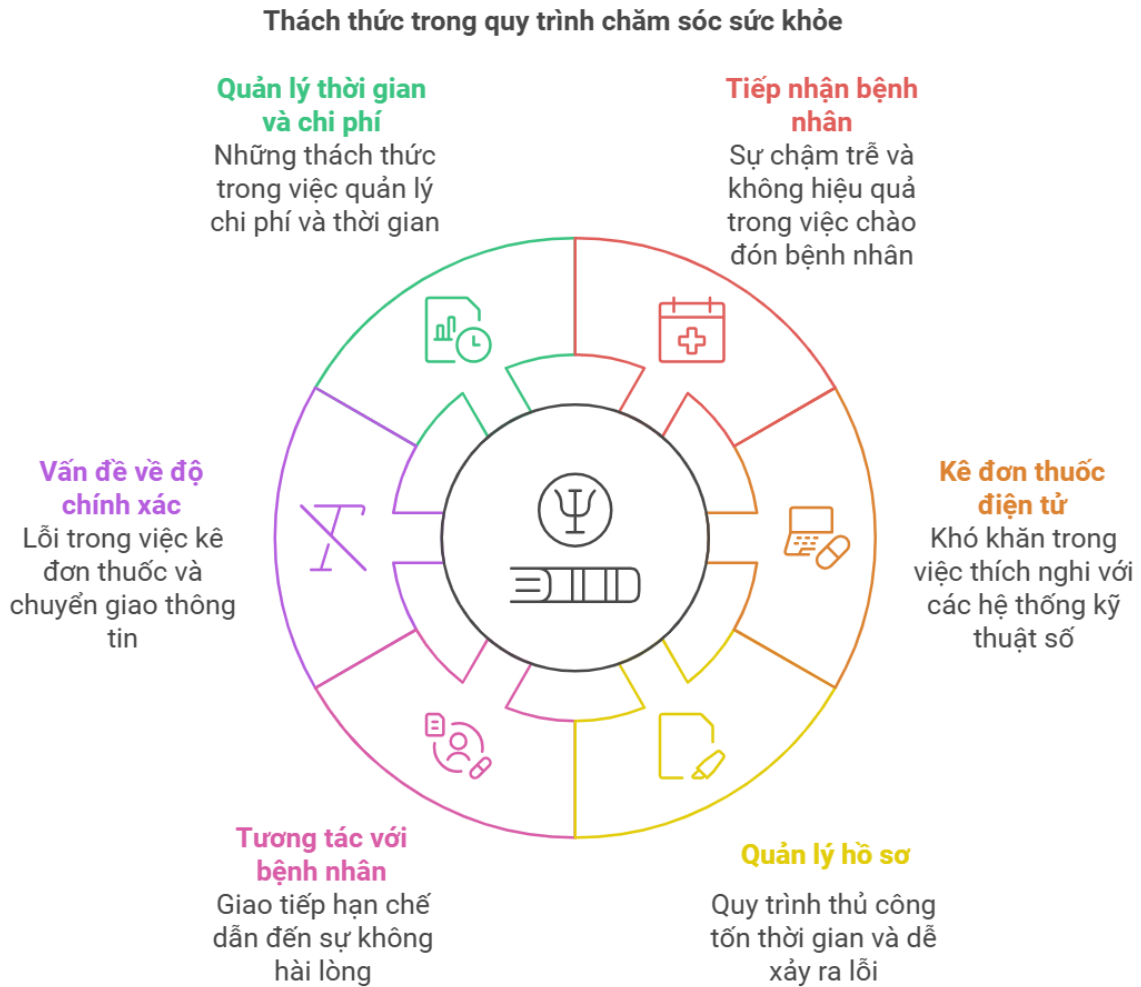
1.1 Khảo sát thực trạng

1.1.1 Thực trạng hiện nay

Bên cạnh thách thức từ làn sóng cạnh tranh khốc liệt trên thị trường, các phòng khám cũng đang phải đối mặt với nhiều vướng mắc trong hoạt động quản lý và vận hành. Quản lý phòng khám theo phong cách truyền thống rườm rà kém hiệu quả là một trong những rào cản lớn nhất tại các phòng khám phát triển mạnh mẽ hơn.

Khi số lượng bệnh nhân tại phòng khám gia tăng, việc ghi chép hoàn toàn bằng thủ công sẽ khiến quy trình tiếp đón và đăng ký khám bệnh trở nên rườm rà, kéo dài thời gian chờ đợi của bệnh nhân. Đối với các bệnh nhân đến tái khám cũng không ngoại lệ. Việc tìm kiếm lại các thông tin bệnh án tốn nhiều thời gian, nhất là khi thất lạc các hồ sơ sổ sách khiến việc tìm kiếm, nhập liệu và ghi chú lại phức tạp. Tình trạng này kéo theo hệ lụy quá tải tại hầu hết các phòng khám, khiến phòng khám hoạt động không hết công suất, tốn nhiều chi phí liên quan không cần thiết.

Kể cả các phòng khám tư nhân hay của công đều đang tập trung quá nhiều thời gian cho các thao tác thủ công làm giảm đi thời gian cần thiết cho hoạt động chẩn đoán và chăm sóc bệnh nhân, khiến năng lực khám và điều trị không đạt được hiệu quả cao \Rightarrow Bệnh nhân không hài lòng với dịch vụ khám chữa của phòng khám.



Hình 1.1 Thách thức trong quy trình chăm sóc sức khỏe

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.

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.

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.

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.

- + Kế 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.

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.

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

+ Kế đơn thuốc và quản lý kho thuốc

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: Phát triển một nền tảng tiện lợi và dễ sử dụng, cho phép bệnh nhân đặt lịch khám nhanh chóng và bác sĩ quản lý lịch hẹn hiệu quả. Hệ thống sẽ giúp giảm thiểu thời gian chờ đợi, hạn chế tình trạng quá tải tại phòng khám, và mang lại trải nghiệm tốt nhất cho người dùng.

- Số hóa và quản lý thông tin y tế một cách hiệu quả: Cung cấp giải pháp lưu trữ và quản lý hồ sơ bệnh nhân, lịch sử khám chữa bệnh, kết quả xét nghiệm và các thông tin y tế khác một cách bảo mật, chính xác và dễ dàng truy cập. Hệ thống sẽ hỗ trợ bác sĩ đưa ra các quyết định điều trị kịp thời, đồng thời tạo điều kiện cho bệnh nhân theo dõi hồ sơ y tế của mình một cách thuận tiện.

- Cải thiện khả năng quản lý và vận hành của phòng khám: Trang bị cho quản trị viên và nhân viên y tế công cụ quản lý lịch làm việc, tài nguyên y tế và dữ liệu bệnh nhân một cách hiệu quả. Hệ thống sẽ tối ưu hóa quy trình điều phối thông tin, giảm thiểu sai sót trong vận hành và hỗ trợ phòng khám nâng cao năng suất làm việc.

- Hỗ trợ ra quyết định thông qua báo cáo và thống kê: Cung cấp các công cụ phân tích và báo cáo chi tiết giúp quản trị viên theo dõi hoạt động của phòng khám, từ đó đưa ra các quyết định chiến lược nhằm nâng cao hiệu quả vận hành và chất lượng dịch vụ.

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.

- Quản lý lịch làm việc của bác sĩ, hiển thị lịch hẹn theo ngày

- 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, doanh thu, ...
- **Chức năng chi tiết cho từng vai trò**

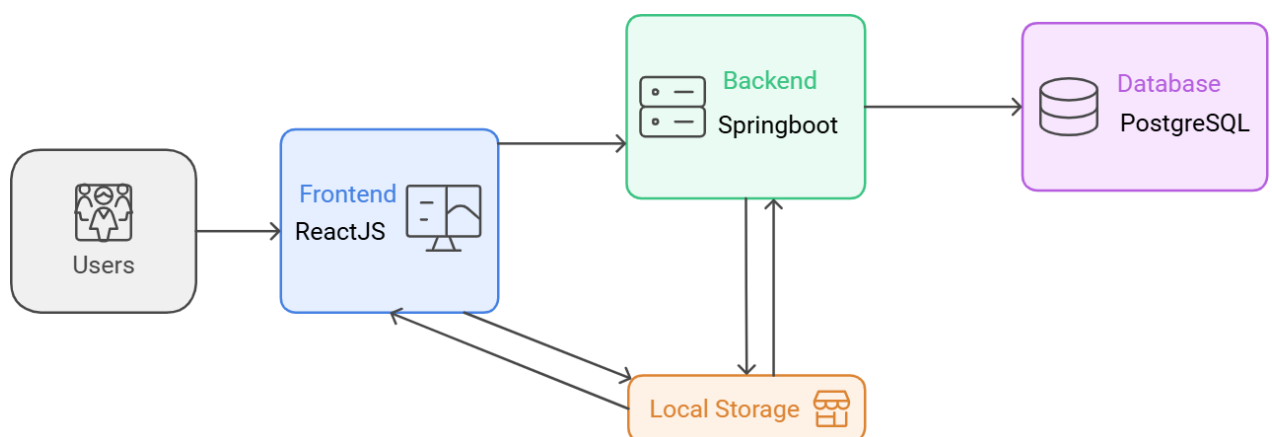
Bảng 1.1 Chi tiết chức năng cho từng vai trò

Vai trò	Chức năng
Bệnh nhân	+ Đăng ký và đăng nhập hệ thống. + Đặt lịch khám và theo dõi lịch sử khám bệnh. + Xem bệnh án và hóa đơn cho từng lần khám. + Quản lý thông tin cá nhân, tài khoản + Bình luận về bác sĩ hoặc phòng khám
Bác sĩ	+ Quản lý lịch hẹn khám

	<ul style="list-style-type: none"> + Khám bệnh: chẩn đoán, kê đơn, chỉ định dịch vụ và cập nhật hồ sơ bệnh án. + Quản lý bệnh nhân: quản lý hồ sơ bệnh án cho từng bệnh nhân + Quản lý thông tin cá nhân, tài khoản + Quản lý bình luận
Nhân viên tiếp nhận	<ul style="list-style-type: none"> + Quản lý lịch hẹn và xác nhận đặt lịch. + Quản lý thanh toán, tạo và cập nhật hóa đơn. + Quản lý thông tin cá nhân, tài khoản
Quản trị viên	<ul style="list-style-type: none"> + Quản lý lịch hẹn + Quản lý tài khoản của tất cả các vai trò trong hệ thống. + Quản lý các danh mục như chuyên khoa, dịch vụ xét nghiệm + Quản lý bình luận + Xem báo cáo thống kê

1.3 Công nghệ sử dụng trong hệ thống

Mô hình kiến trúc hệ thống:



*Hình 1.2 Kiến trúc hệ thống***1.3.1 React JS****- Lý do lựa chọn ReactJS cho phần Frontend**

+ Hiệu suất cao: ReactJS sử dụng Virtual DOM, giúp tăng hiệu suất trong việc cập nhật giao diện và phản hồi người dùng.

+ Tính linh hoạt và dễ mở rộng: Với cấu trúc component-based, ReactJS cho phép xây dựng giao diện theo cách dễ quản lý và tái sử dụng. Điều này giúp dễ dàng mở rộng hoặc thay đổi giao diện khi hệ thống phát triển.

+ Cộng đồng lớn: ReactJS có một cộng đồng lớn và nhiều thư viện hỗ trợ, giúp đẩy nhanh quá trình phát triển.

- Mô tả ReactJS:

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.

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.

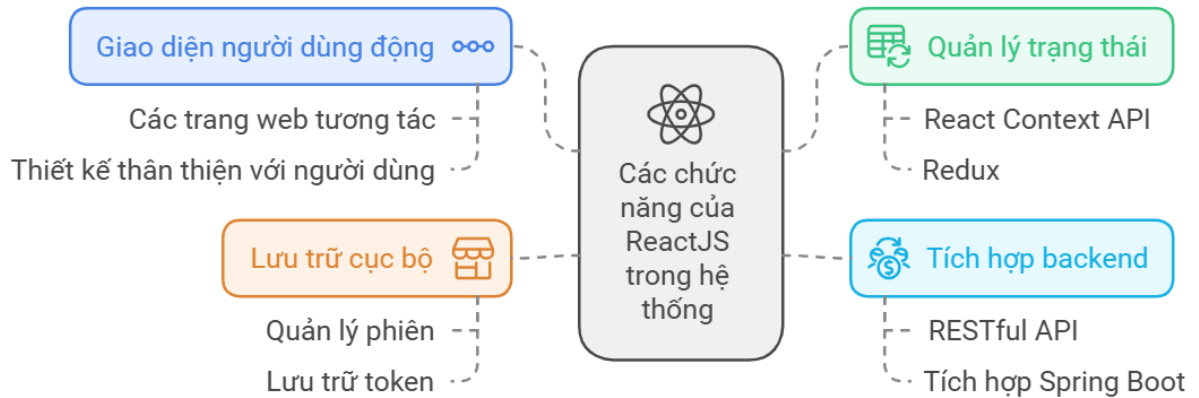
- Chức năng chính của ReactJS trong hệ thống

+Tạo giao diện người dùng động: ReactJS chịu trách nhiệm xây dựng các trang web tương tác và thân thiện với người dùng, như trang đặt lịch khám, quản lý thông tin cá nhân, và tra cứu hồ sơ bệnh nhân.

+Quản lý trạng thái ứng dụng: Sử dụng các công cụ như React Context API hoặc Redux (nếu cần) để quản lý trạng thái ứng dụng, đảm bảo dữ liệu được đồng bộ giữa các thành phần giao diện.

+Tích hợp với backend: ReactJS giao tiếp với backend Spring Boot thông qua các RESTful API, đảm bảo dữ liệu được lấy và hiển thị đúng cách.

+Sử dụng Local Storage: Tạm lưu trữ các thông tin cần thiết (như token đăng nhập) trong local storage để hỗ trợ các tính năng như duy trì phiên đăng nhập.



Hình 1.3 Chức năng chính của ReactJS trong hệ thống

1.3.2 Java Spring Boot

- Lý do chọn Spring Boot cho backend:

+ Hiệu suất và khả năng mở rộng: Spring Boot cung cấp một nền tảng mạnh mẽ để phát triển các ứng dụng backend với hiệu suất cao, phù hợp với hệ thống quản lý phòng khám có thể mở rộng trong tương lai.

+ Cấu hình đơn giản: Spring Boot giảm bớt sự phức tạp trong cấu hình bằng cách cung cấp các cấu hình mặc định (opinionated defaults) và các công cụ tự động cấu hình.

+ Tích hợp dễ dàng: Spring Boot hỗ trợ tích hợp với nhiều công cụ và thư viện khác như PostgreSQL, RESTful APIs, và các framework bảo mật như Spring Security.

+ Cộng đồng và tài liệu phong phú: Spring Boot có một cộng đồng lớn và hệ thống tài liệu phong phú, giúp dễ dàng tìm kiếm giải pháp khi gặp vấn đề.

- Mô tả Spring Boot:

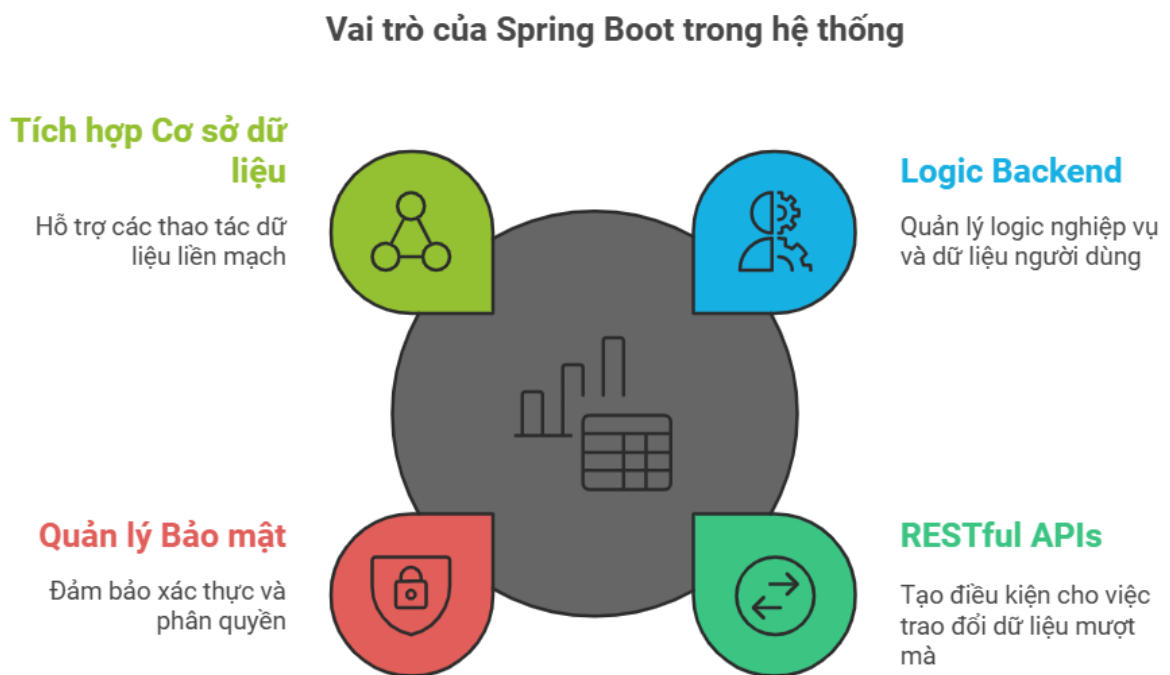
Spring Boot được phát triển bởi Pivotal Software, với phiên bản đầu tiên (Spring Boot 1.0) được phát hành vào tháng 4 năm 2014. Mục tiêu ban đầu của Spring Boot là giải quyết những hạn chế và phức tạp trong việc cấu hình Spring Framework truyền thống. Spring Framework, mặc dù rất mạnh mẽ và linh hoạt, 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.

- Chức năng chính của Spring Boot trong hệ thống
 - + Xử lý backend logic: Spring Boot đảm nhận việc xử lý logic nghiệp vụ của hệ thống, bao gồm quản lý thông tin bệnh nhân, lịch hẹn, và tài khoản người dùng.
 - + Cung cấp RESTful APIs: Các RESTful API được xây dựng bằng Spring Boot để giao tiếp giữa backend và frontend ReactJS, đảm bảo trao đổi dữ liệu mượt mà.
 - + Quản lý bảo mật: Sử dụng Spring Security để triển khai các cơ chế bảo mật như xác thực (authentication) và phân quyền (authorization) cho các vai trò khác nhau trong hệ thống (bệnh nhân, bác sĩ, quản trị viên).
 - + Tích hợp với cơ sở dữ liệu: Spring Boot hỗ trợ tích hợp liền mạch với PostgreSQL thông qua Spring Data JPA, giúp thực hiện các thao tác CRUD nhanh chóng và dễ dàng.



Hình 1.4 Chức năng chính của Spring Boot trong hệ thống

1.3.3 PostgreSQL

- Lý do chọn PostgreSQL để làm database:
 - + Hiệu suất cao và đáng tin cậy: PostgreSQL được biết đến là một hệ quản trị cơ sở dữ liệu mạnh mẽ với hiệu suất cao, xử lý tốt các hệ thống có lượng dữ liệu lớn và giao dịch phức tạp.

- + Hỗ trợ tính năng nâng cao: PostgreSQL hỗ trợ các tính năng mạnh mẽ như quản lý JSON, truy vấn phức tạp, và cơ chế khóa dữ liệu (locking) tiên tiến, phù hợp với nhu cầu lưu trữ và quản lý dữ liệu y tế.

- + Khả năng mở rộng: PostgreSQL có khả năng mở rộng cả theo chiều dọc (vertical scaling) và chiều ngang (horizontal scaling), đảm bảo đáp ứng được nhu cầu khi hệ thống phát triển.

- + Tích hợp tốt với Spring Boot: PostgreSQL tích hợp dễ dàng với Spring Boot thông qua Spring Data JPA, giúp triển khai và quản lý cơ sở dữ liệu thuận lợi hơn.

- + Nguồn mở và miễn phí: Là một giải pháp nguồn mở, PostgreSQL không yêu cầu chi phí bản quyền, giúp giảm chi phí vận hành hệ thống.

- Mô tả PostgreSQL:

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.

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.

- Chức năng chính của PostgreSQL trong hệ thống:

- + Lưu trữ toàn bộ dữ liệu của hệ thống

- + Tích hợp với backend: Kết hợp với Spring Boot thông qua Spring Data JPA để thực hiện các thao tác CRUD (Create, Read, Update, Delete) nhanh chóng và hiệu quả.

1.4 Kết luận Chương I

Trong chương 1, đồ án đã khảo sát nghiệp vụ và thu thập yêu cầu từ thực tế được rút ra từ nhiều hệ thống quản lý phòng khám. Từ những dữ liệu đó, chương này đưa ra mục tiêu, phạm vi và các chức năng chính của ứng dụng và xác định các công nghệ framework sẽ được sử dụng trong ứng dụng

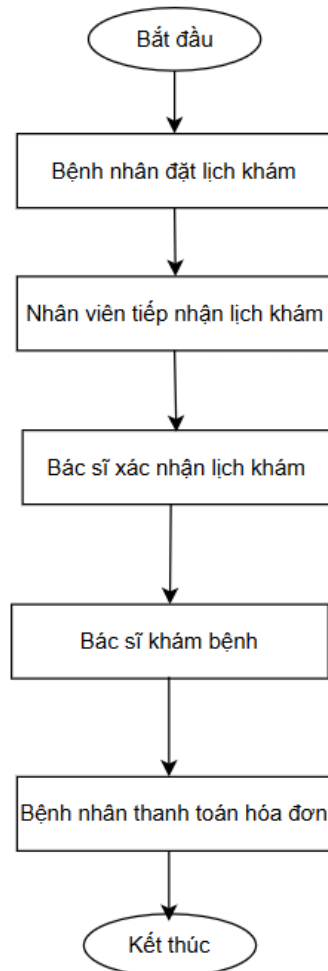
CHƯƠNG II. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương I đã đưa ra một cái nhìn tổng quan về hệ thống, sang tới chương II sẽ đi sâu vào phân tích, thiết kế các chức năng của hệ thống

2.1 Mô tả bài toán

2.1.1 Tổng quan quy trình khám bệnh của hệ thống:

1. Bệnh nhân bắt đầu đặt lịch khám
2. Sau khi bệnh nhân đặt lịch khám sẽ được nhân viên tiếp nhận lịch
3. Các lịch hẹn đã được tiếp nhận sẽ được bác sĩ xác nhận
4. Bác sĩ xác nhận lịch hẹn và bắt đầu khám bệnh, đưa ra các chẩn đoán cũng như kê đơn cho bệnh nhân
5. Bệnh nhân được khám xong thì thanh toán hóa đơn



Hình 2.1 Quy trình khám bệnh

2.1.2 Quy trình đặt lịch khám của bệnh nhân:

Bệnh nhân truy cập hệ thống bằng tài khoản đã được đăng ký. Sau khi đăng nhập thành công bệnh nhân có thể bắt đầu đặt lịch khám bệnh. Khi đặt lịch, tùy theo nhu cầu và lý do khám bệnh mà bệnh nhân chọn chuyên khoa khám cho phù hợp. Bệnh nhân chọn bác sĩ khám thuộc chuyên khoa đó có lịch trống phù hợp với thời gian mà bệnh nhân muốn hẹn khám. Sau khi đã chọn xong bác sĩ, thời gian phù hợp bệnh nhân nhập các thông tin cá nhân cần thiết và xác nhận lại lịch hẹn. Lịch hẹn đã được xác nhận và đặt thành công

2.1.3 Quy trình khám bệnh của bác sĩ:

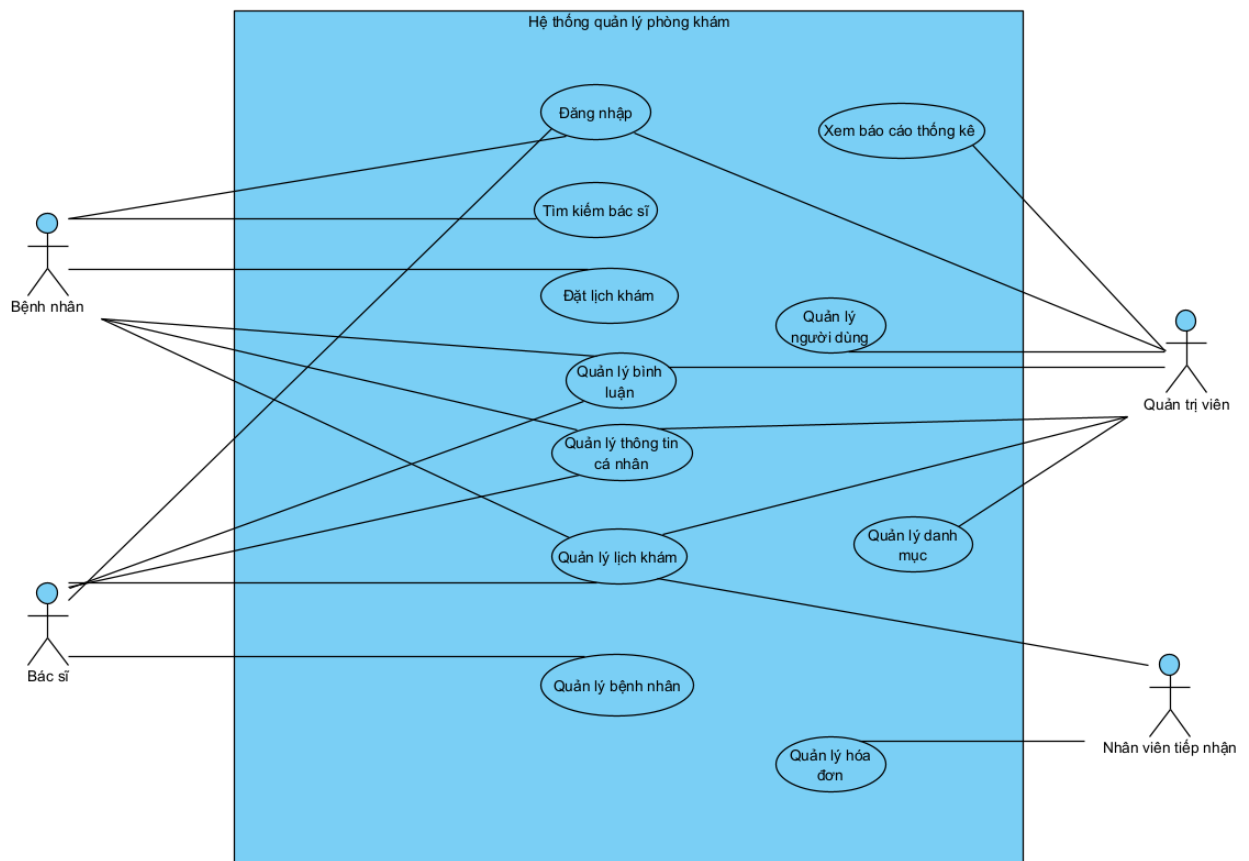
Bác sĩ truy cập hệ thống bằng tài khoản đã được đăng ký. Sau khi đăng nhập thành công bác sĩ có thể xác nhận lịch hẹn và bắt đầu khám bệnh. Quá trình khám bệnh gồm 3 phần:

- Phần 1: Bác sĩ xác định lý do khám bệnh, bệnh sử của bệnh nhân, xác định các chỉ số sinh tồn (nhịp tim, huyết áp, ...), nhập kết quả khám tổng quát, khám bộ phận từ đó đưa ra chẩn đoán sơ bộ
- Phần 2: Bác sĩ xác định các dịch vụ xét nghiệm mà bệnh nhân cần thực hiện
- Phần 3: Bác sĩ nhập các kết quả xét nghiệm, đưa ra chẩn đoán xác định cuối cùng và kê đơn (không bắt buộc)

2.2 Biểu đồ Use case (Usecase diagram)

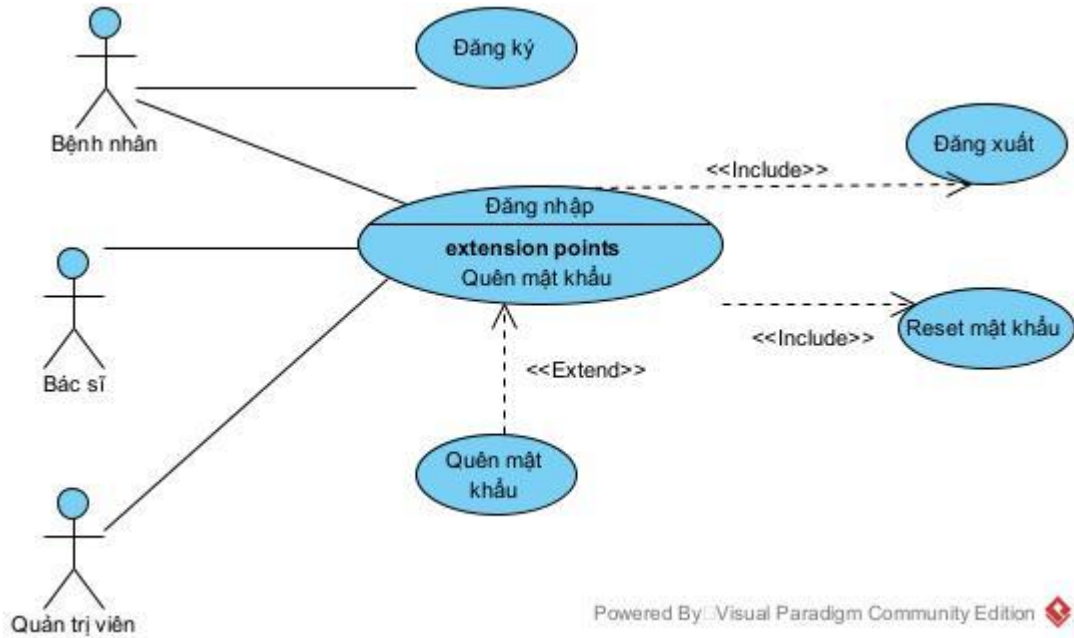
2.2.1 Use case Tổng quan hệ thống

Hệ thống gồm 4 actor chính là Bệnh nhân, Bác sĩ, Nhân viên tiếp nhận và Quản trị viên



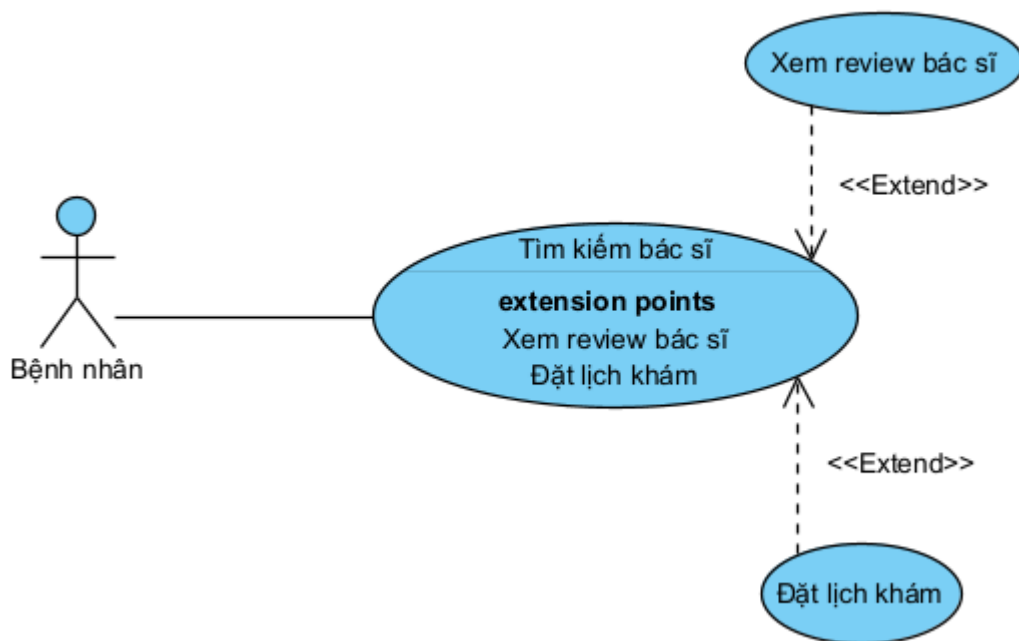
Hình 2.2 Use case Tổng quan hệ thống

2.2.2 Use case Đăng nhập



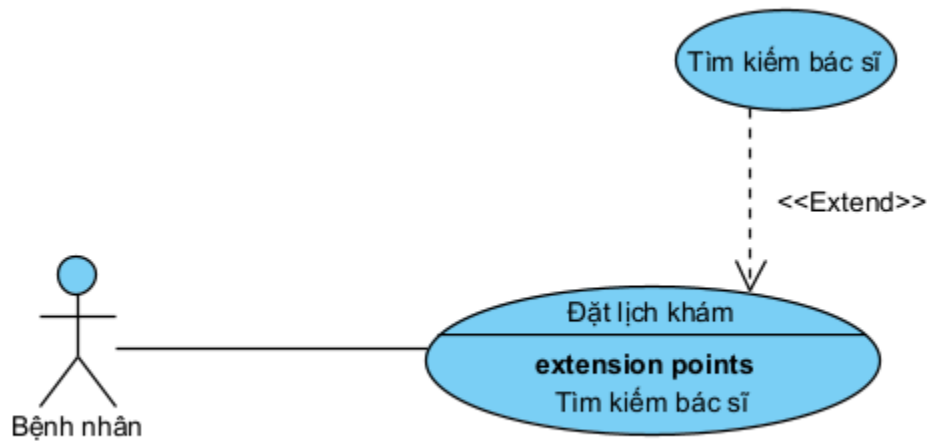
Hình 2.3 Use case Đăng nhập

2.2.3 Use case Tìm kiếm bác sĩ (Bệnh nhân)



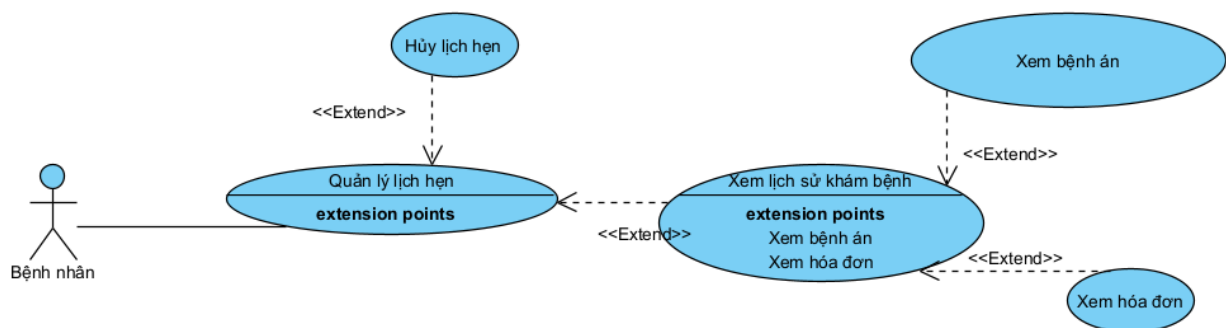
Hình 2.4 Use case Tìm kiếm bác sĩ

2.2.4 Use case Đặt lịch khám (Bệnh nhân)



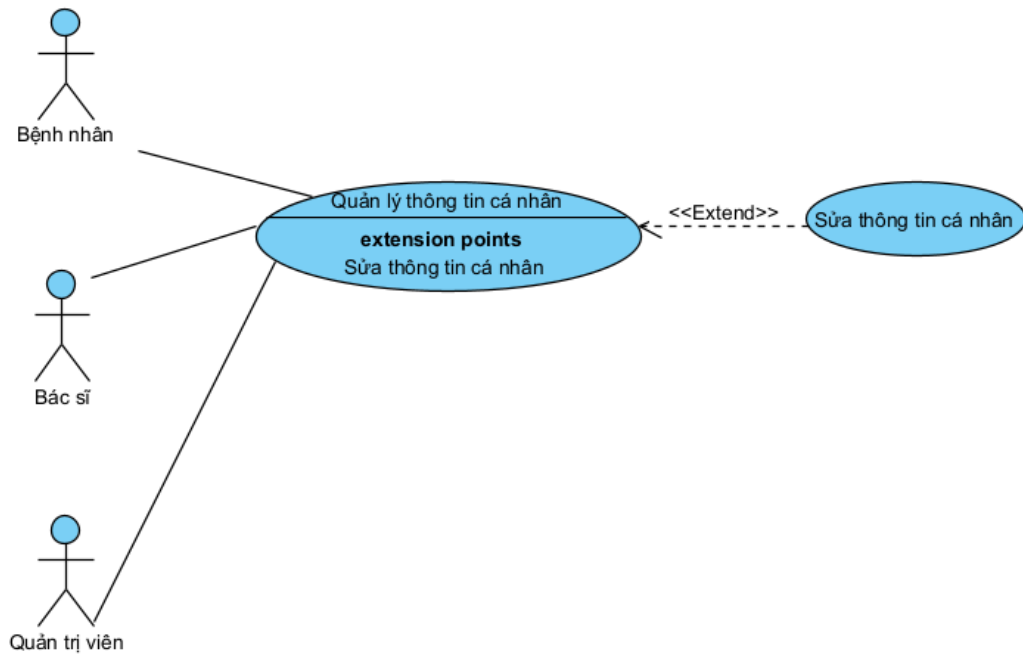
Hình 2.5 Use case Đặt lịch khám

2.2.5 Use case Quản lý lịch hẹn (Bệnh nhân)



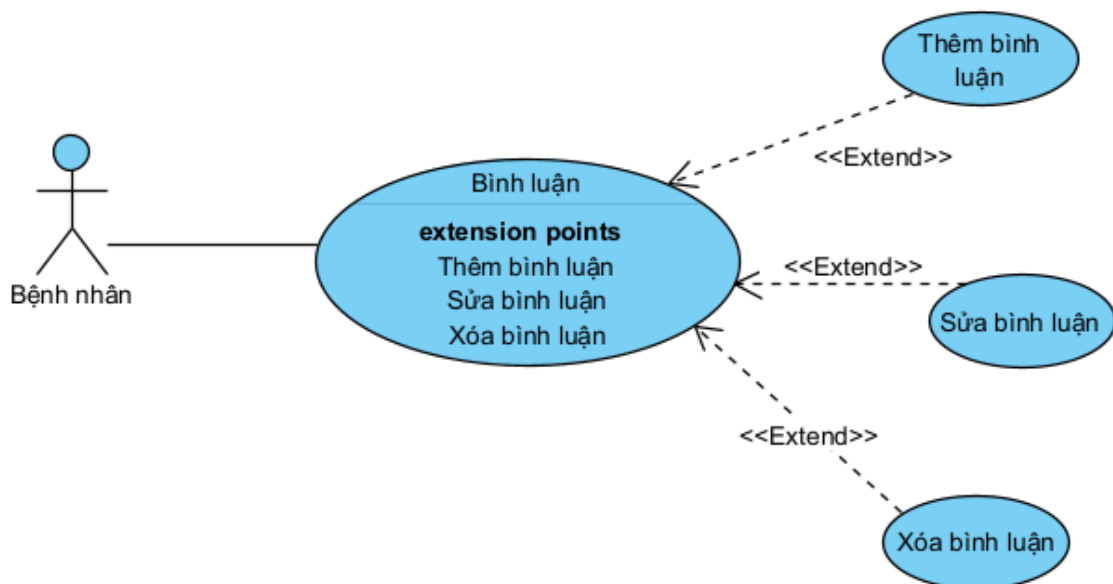
Hình 2.6 Use case Quản lý lịch hẹn (Bệnh nhân)

2.2.6 Use case Quản lý thông tin cá nhân



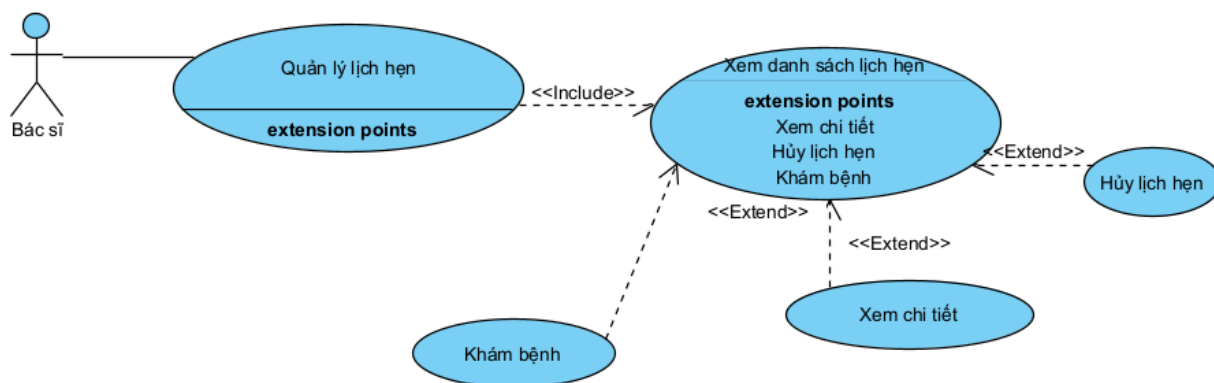
Hình 2.7 Use case Quản lý thông tin cá nhân

2.2.7 Use case Quản lý bình luận (Bệnh nhân)



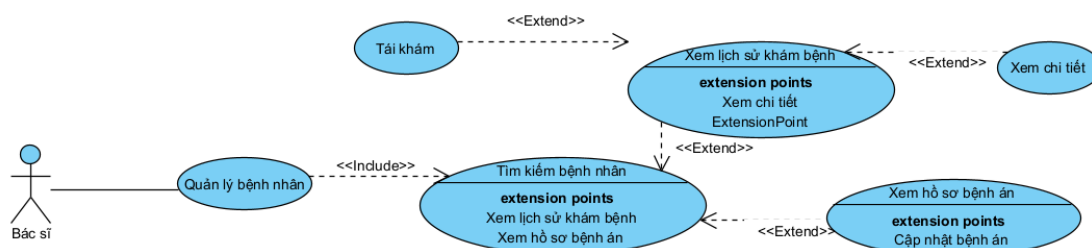
Hình 2.8 Use case Quản lý bình luận (Bệnh nhân)

2.2.8 Use case Quản lý lịch hẹn (Bác sĩ)



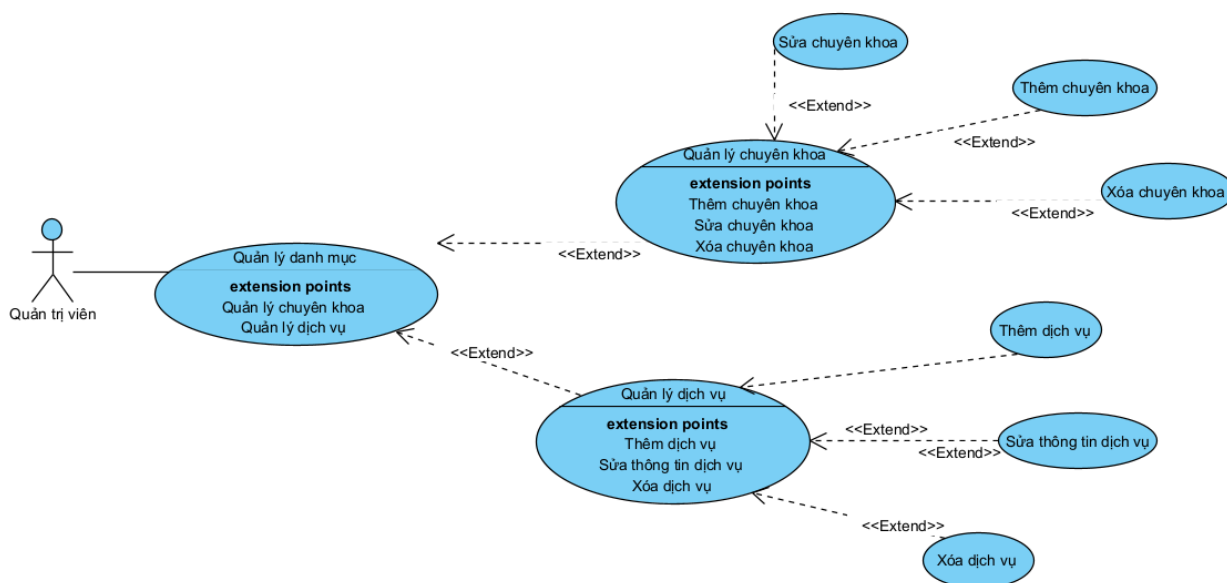
Hình 2.9 Use case Quản lý lịch hẹn (Bác sĩ)

2.2.9 Use case Quản lý bệnh nhân (Bác sĩ)



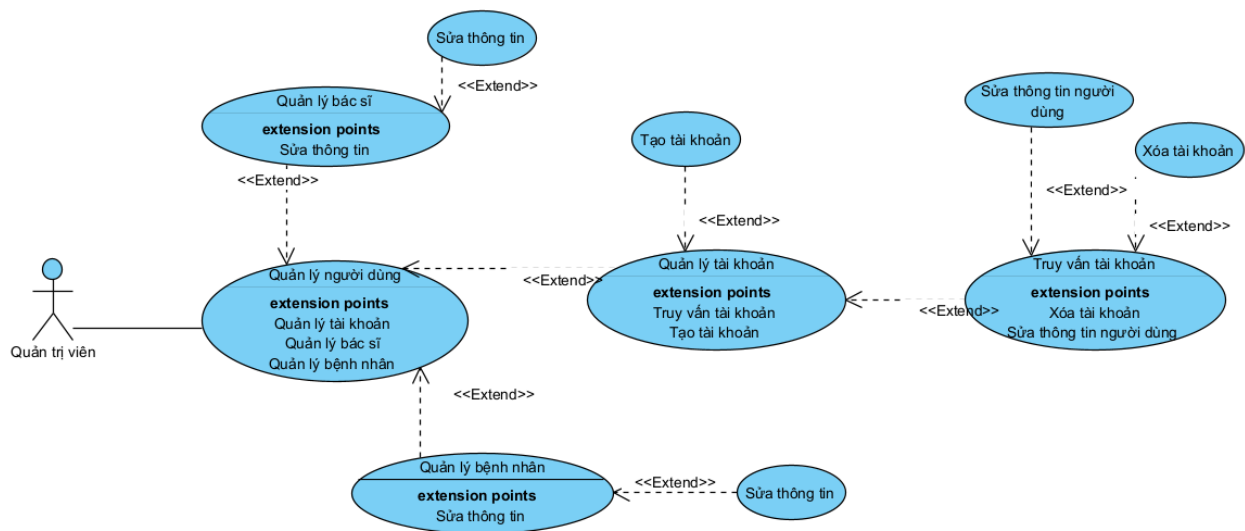
Hình 2.10 Use case Quản lý bệnh nhân

2.2.10 Use case Quản lý danh mục (QTV)



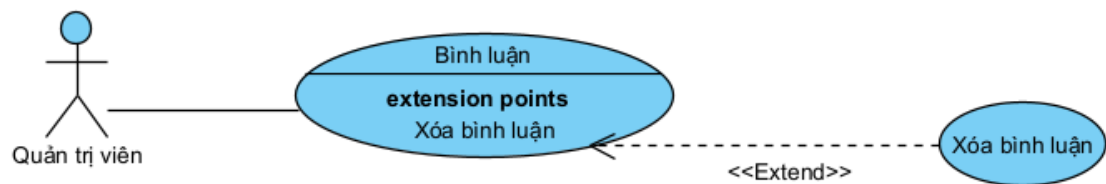
Hình 2.11 Use case Quản lý danh mục

2.2.11 Use case Quản lý người dùng (QTV)



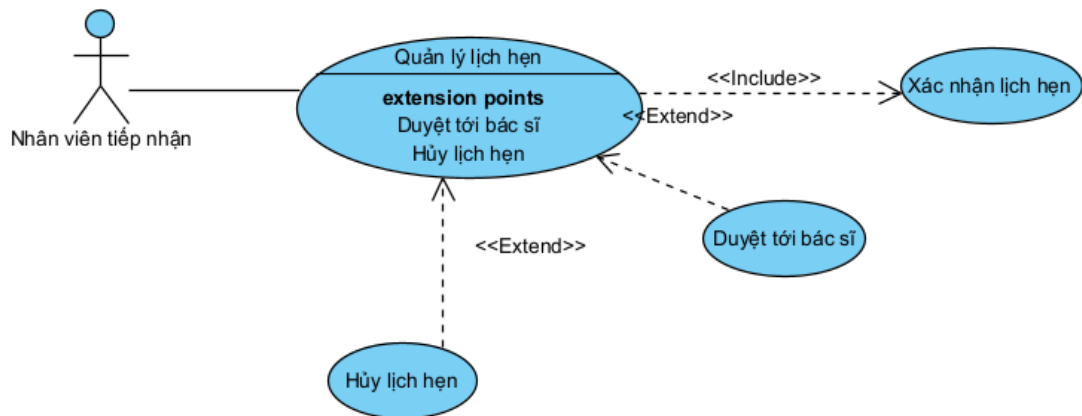
Hình 2.12 Use case Quản lý tài khoản người dùng

2.2.12 Use case Quản lý bình luận (QTV)



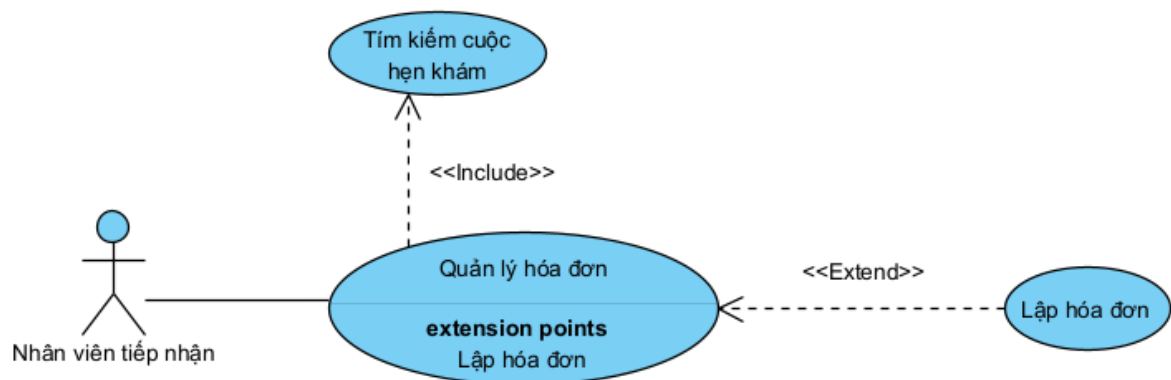
Hình 2.13 Use case Quản lý bình luận (QTV)

2.2.13 Use case Quản lý lịch hẹn (NVTN)



Hình 2.14 Use case Quản lý lịch hẹn (NVTN)

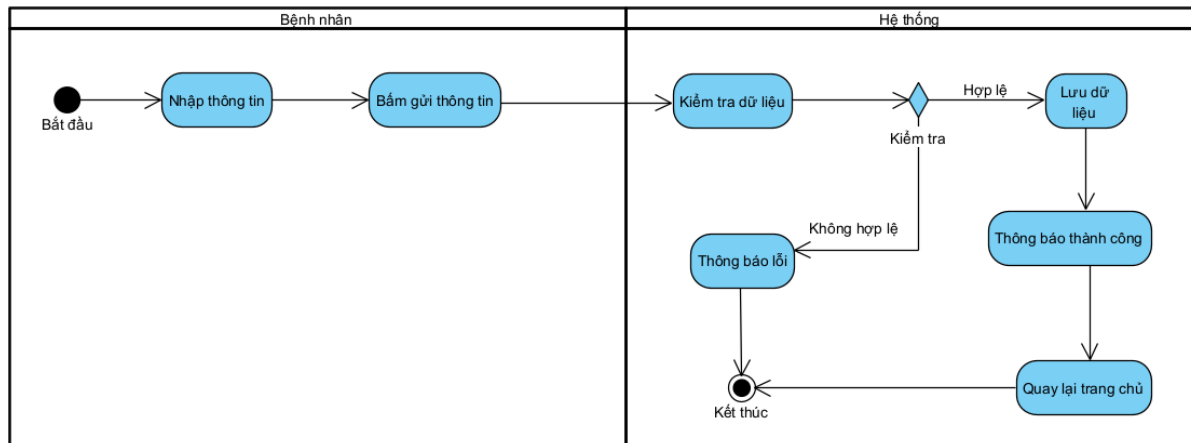
2.2.14 Use case Quản lý hóa đơn (NVTN)



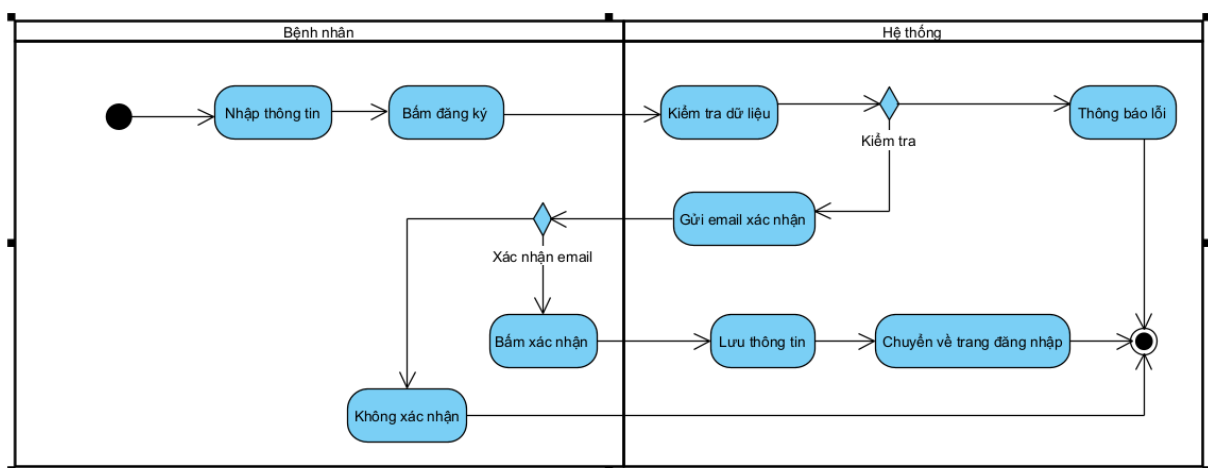
Hình 2.15 Use case Quản lý hóa đơn

2.3 Biểu đồ hoạt động (Activity Diagram)

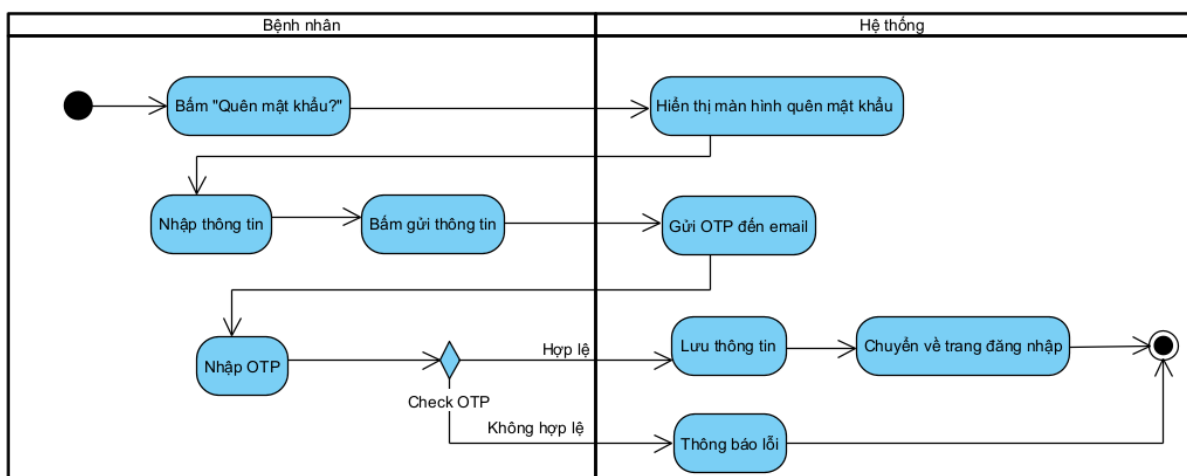
Các biểu đồ hoạt động trong hệ thống:



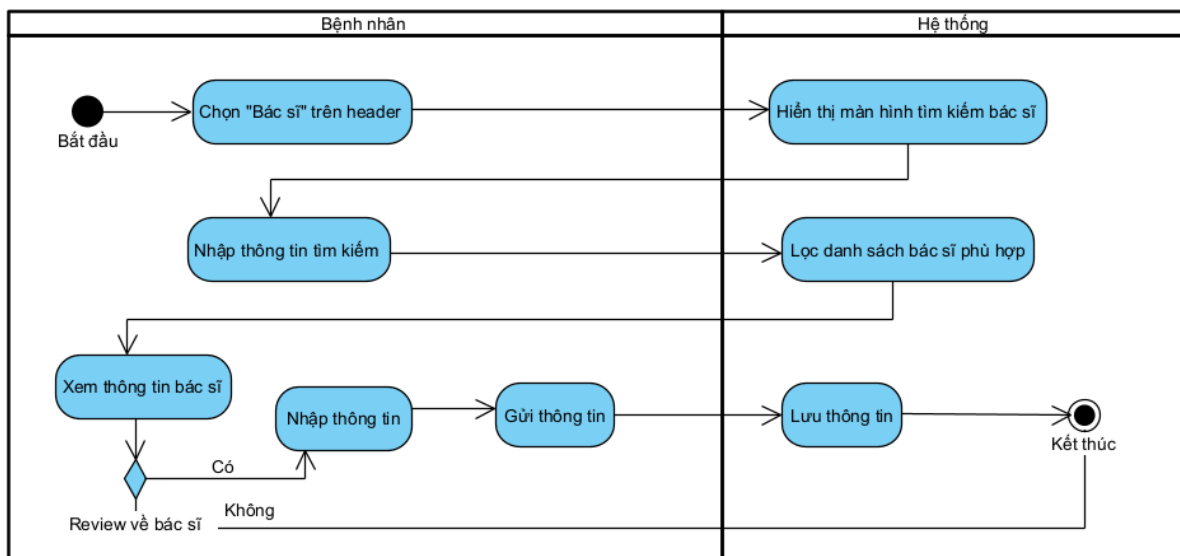
Hình 2.16 Biểu đồ hoạt động Đăng nhập



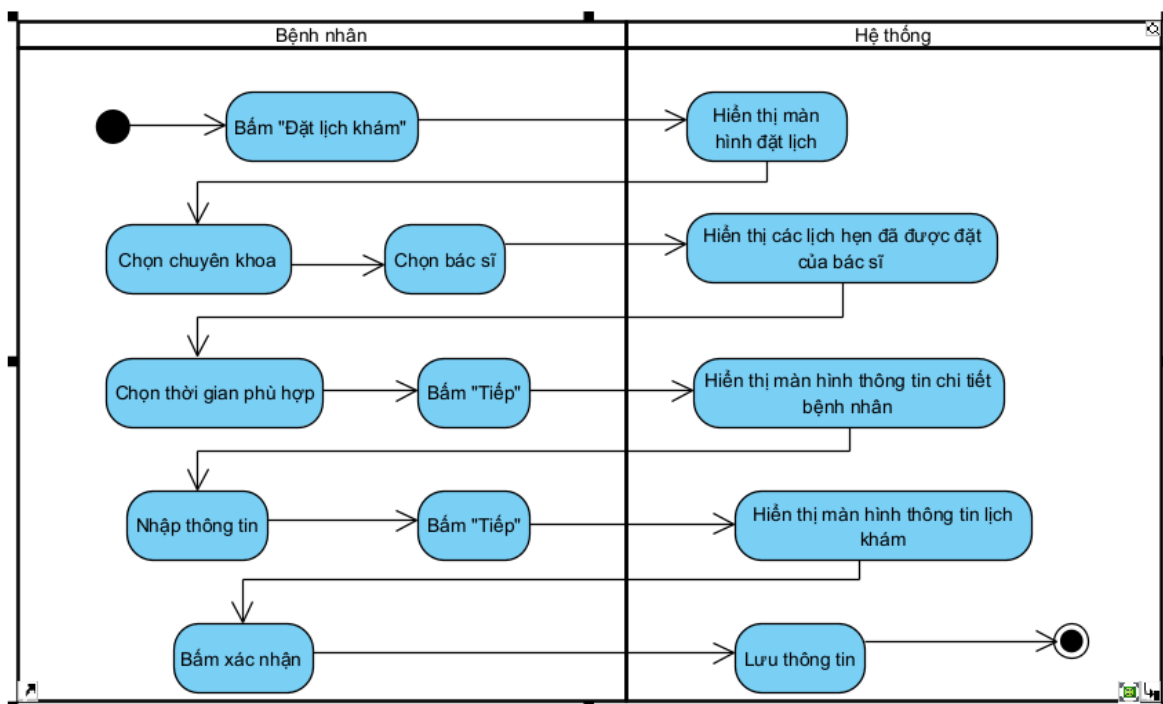
Hình 2.17 Biểu đồ hoạt động Đăng ký



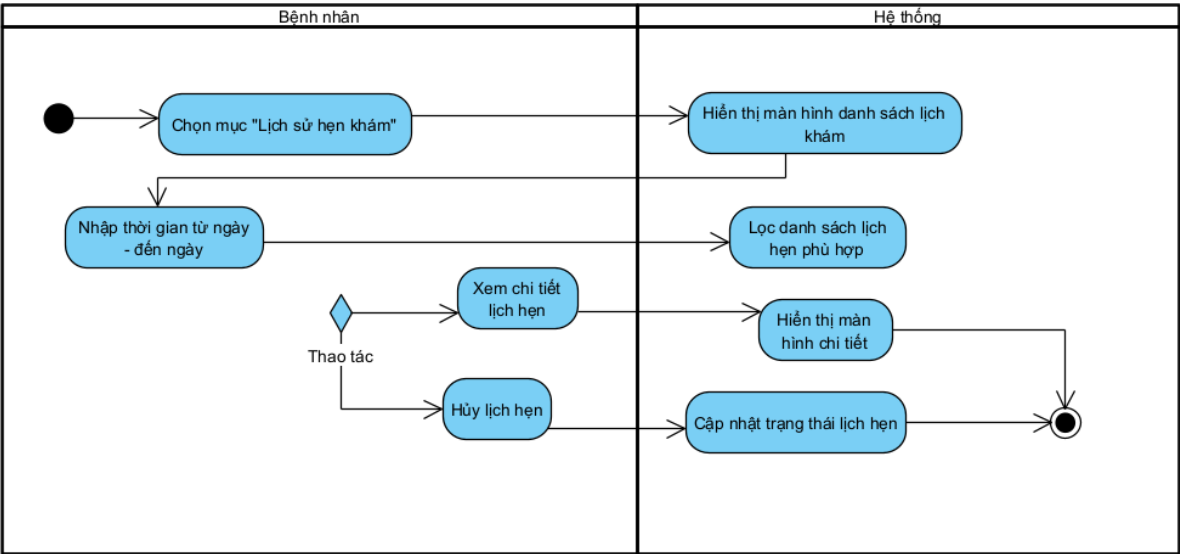
Hình 2.18 Biểu đồ hoạt động Quên mật khẩu



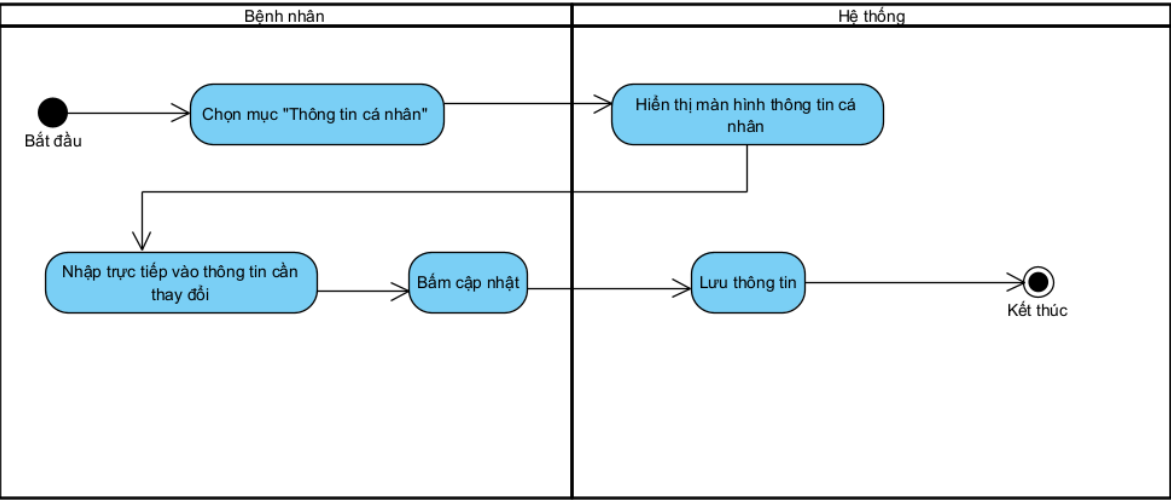
Hình 2.19 Biểu đồ hoạt động Tìm kiếm bác sĩ



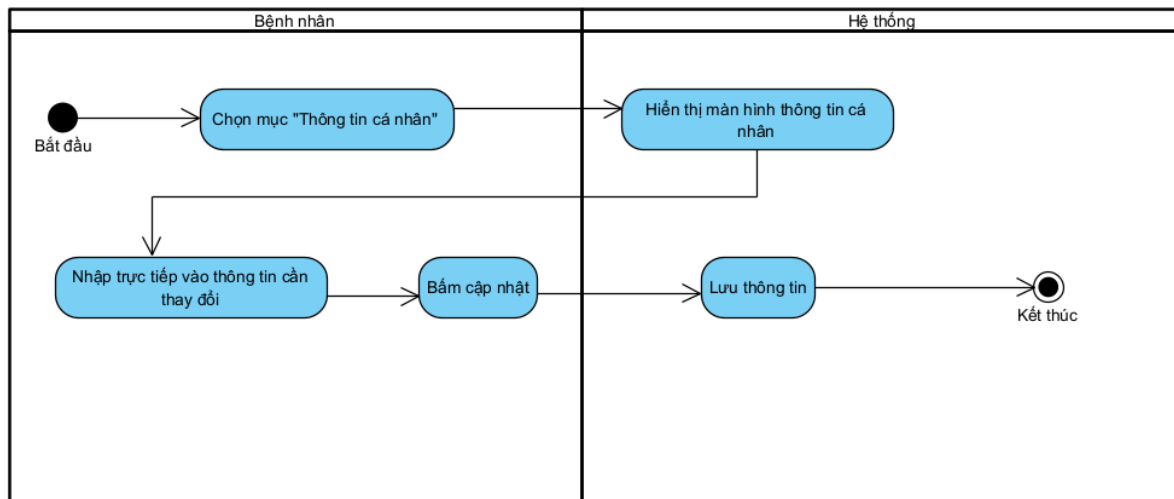
Hình 2.20 Biểu đồ hoạt động Đặt lịch khám



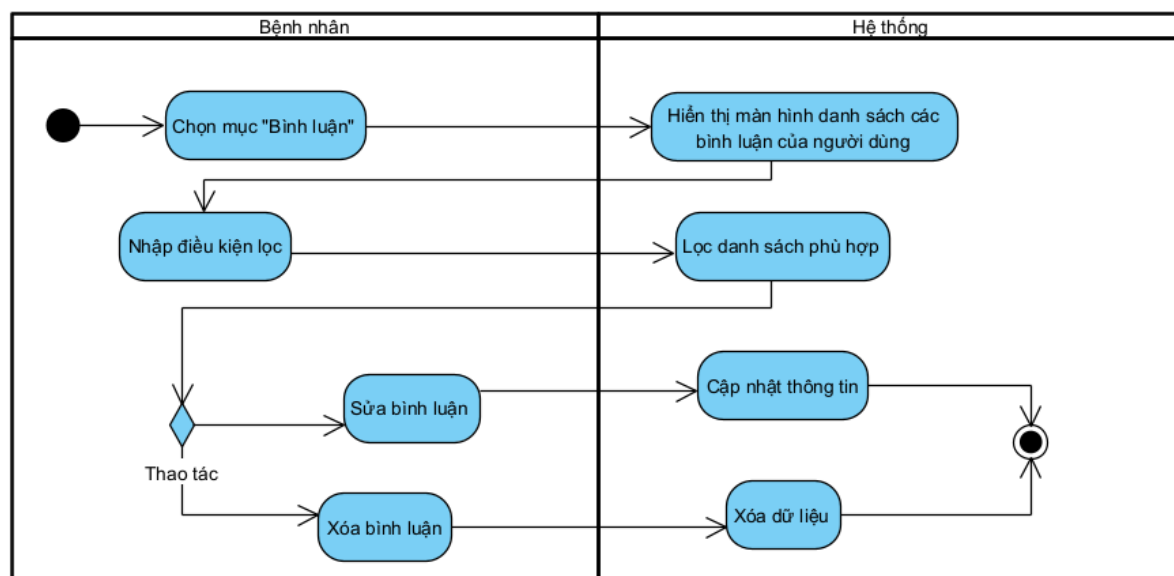
Hình 2.21 Biểu đồ hoạt động Quản lý lịch khám



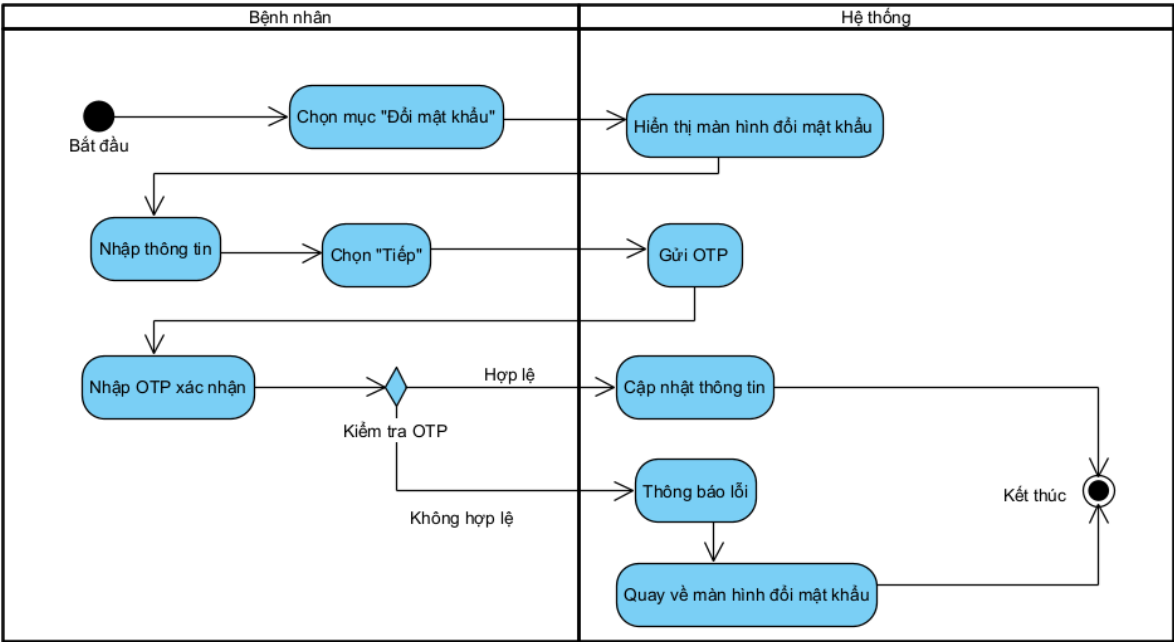
Hình 2.22 Biểu đồ hoạt động Quản lý thông tin cá nhân



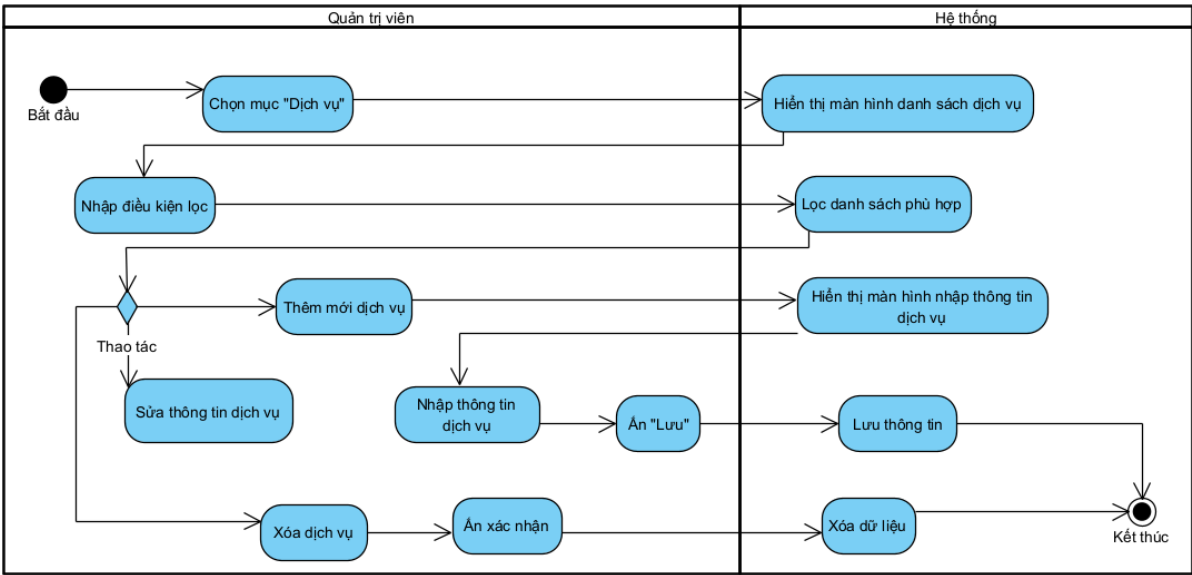
Hình 2.23 Biểu đồ hoạt động Quản lý thông tin cá nhân



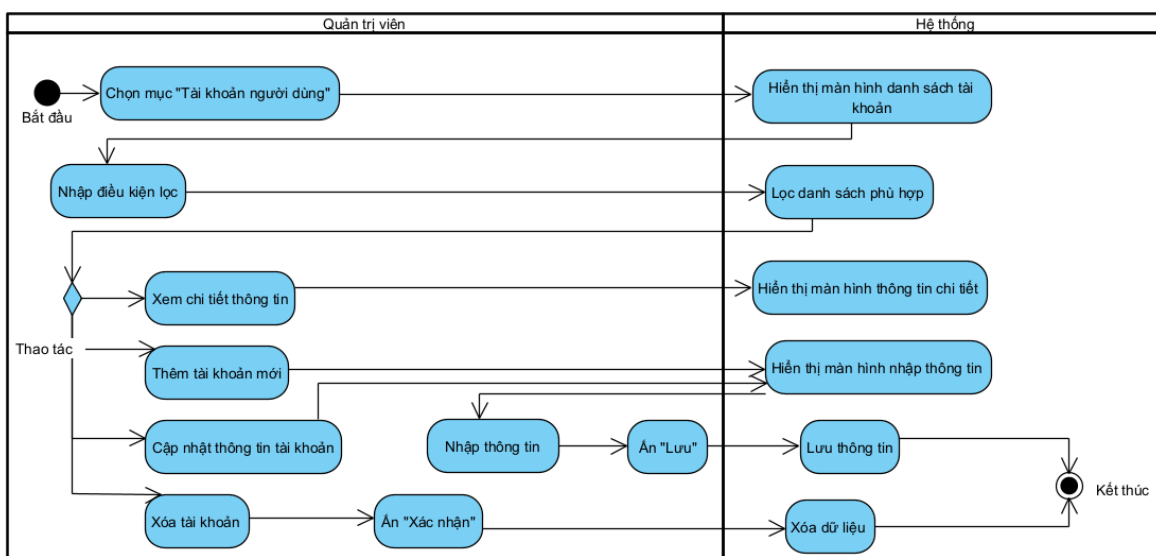
Hình 2.24 Biểu đồ hoạt động Quản lý bình luận



Hình 2.25 Biểu đồ hoạt động Đổi mật khẩu



Hình 2.26 Biểu đồ hoạt động Quản lý dịch vụ



Hình 2.27 Biểu đồ hoạt động Quản lý tài khoản người dùng

2.4 Kịch bản chuẩn và ngoại lệ

Use case Đăng ký:

Bảng 2.1 Kịch bản 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"> 1. Hệ thống hiển thị giao diện ký 2. Người dùng nhập các thông tin của tài khoản: tên, email, số điện thoại, mật khẩu 3. Hệ thống kiểm tra định dạng các thông tin 4. Người dùng sửa các thông tin cho đúng định dạng và gửi thông tin 5. Hệ thống kiểm tra thông tin và gửi email xác nhận 6. Người dùng truy cập email và xác nhận

	7. Hệ thống điều hướng về trang đăng nhập, lưu thông tin tài khoản vào CSDL
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 8.6.2 Quay lại bước 2

Use case Đăng nhập:

Bảng 2.2 Kịch bản 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 dùng nhập sai thông tin đăng nhập 4.2 Quay lại bước 2

*Use case Quên mật khẩu:**Bảng 2.3 Kịch bản 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"> 1. Hệ thống hiển thị giao diện quên mật khẩu 2. Người dùng nhập email và bấm gửi OTP 3. Hệ thống gửi OTP đến email 4. Người dùng lấy OTP và nhập 5. Hệ thống kiểm tra OTP 6. Người dùng nhập mật khẩu mới và bấm cập nhật 7. Hệ thống cập nhật mật khẩu mới và điều hướng về trang đăng nhập
Ngoại lệ	<ol style="list-style-type: none"> 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

*Use case Tìm kiếm bác sĩ:**Bảng 2.4 Kịch bản 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	<ol style="list-style-type: none"> 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	

Use case Đặt lịch khám:

Bảng 2.5 Kịch bản 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"> 1. Hệ thống hiển thị giao diện đặt lịch khám 2. Người dùng chọn chuyên khoa phù hợp với bệnh cần khám 3. Hệ thống lọc danh sách bác sĩ của chuyên khoa đó 4. Người dùng chọn bác sĩ 5. Hệ thống hiển thị khung lịch hẹn có ngày tháng và thời gian, các lịch hẹn đã được đặt của bác sĩ đó 6. Người dùng chọn khung thời gian phù hợp và ấn tiếp tục 7. Hệ thống hiển thị màn hình thông tin chi tiết của người dùng 8. Người dùng chỉnh sửa thông tin và ấn tiếp tục 9. Hệ thống chuyển sang màn hình xác nhận lịch hẹn 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) 11. Hệ thống thông báo thành công và lưu thông tin lịch hẹn vào CSDL
Ngoại lệ	4.1 Người dùng đặt lịch hẹn từ chức năng tìm kiếm bác sĩ

	<p>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</p> <p>4.3 Tiếp tục bước 6</p>
--	---

Use case Đổi mật khẩu:

Bảng 2.6 Kịch bản 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	<p>1. Hệ thống hiển thị giao diện đổi mật khẩu</p> <p>2. Người dùng nhập email và bấm gửi OTP</p> <p>3. Hệ thống gửi OTP đến email</p> <p>4. Người dùng lấy OTP và nhập</p> <p>5. Hệ thống kiểm tra OTP, nếu hợp lệ chuyển màn hình nhập mật khẩu</p> <p>6. Người dùng nhập mật khẩu cũ, mật khẩu mới và bấm cập nhật</p> <p>7. Hệ thống kiểm tra 2 mật khẩu cũ và mới, nếu khớp thì cập nhật vào CSDL</p>
Ngoại lệ	<p>3.1 Hệ thống báo lỗi email không đúng với email đang đăng nhập hiện tại</p> <p>3.2 Quay lại bước 2</p> <p>5.1 Hệ thống báo lỗi do OTP nhập không đúng hoặc hết hạn</p> <p>5.2 Quay lại bước 2</p> <p>7.1 hệ thống báo lỗi do 2 mật khẩu không khớp nhau</p> <p>7.2 Quay lại bước 6</p>

Use case Cập nhật thông tin cá nhân:

Bảng 2.7 Kịch bản use case Cập nhật thông tin cá nhân

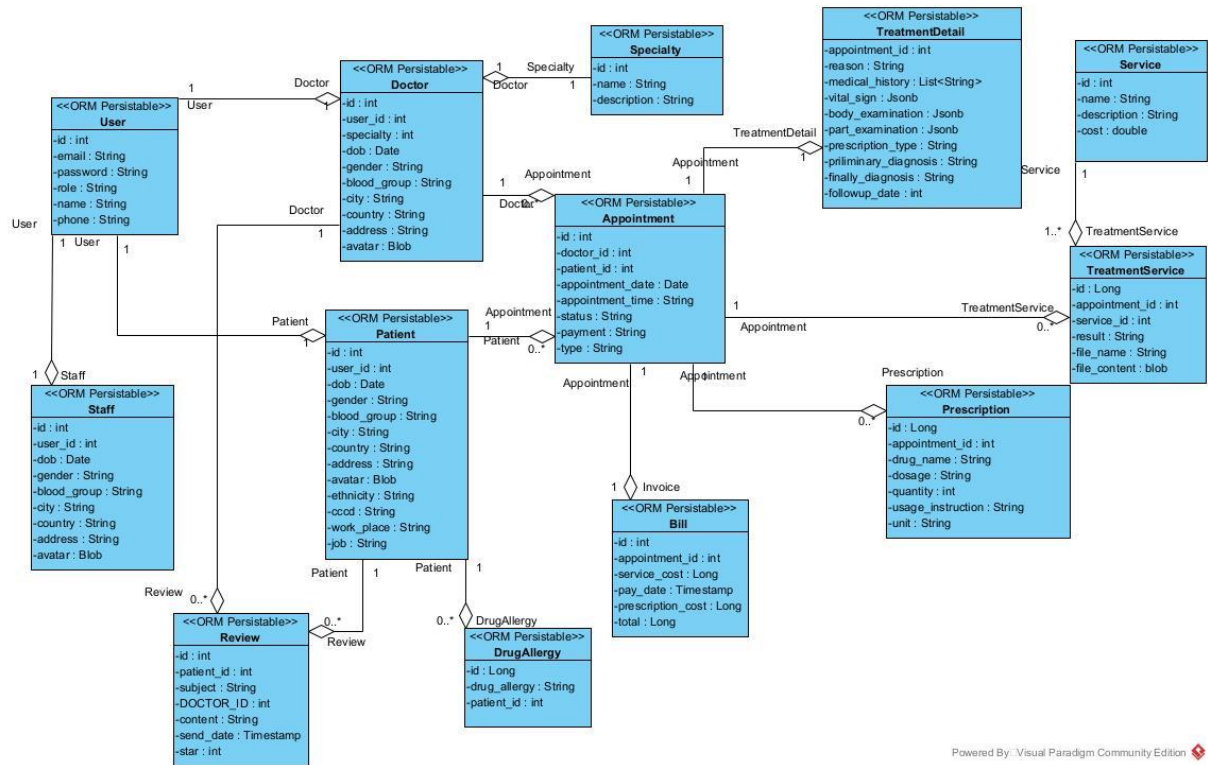
Tên Use Case	Cập nhật thông tin cá nhân
Actor	Bệnh nhân/ Bác sĩ/ Nhân viên tiếp nhận/ Quản trị viên
Mục tiêu	Người dùng cập nhật lại thông tin cá nhân của mình
Tiền điều kiện	Người dùng đã đăng nhập và hệ thống
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình thông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

Use case Quản lý bình luận:

Bảng 2.8 Kịch bản use case Quản lý bình luận

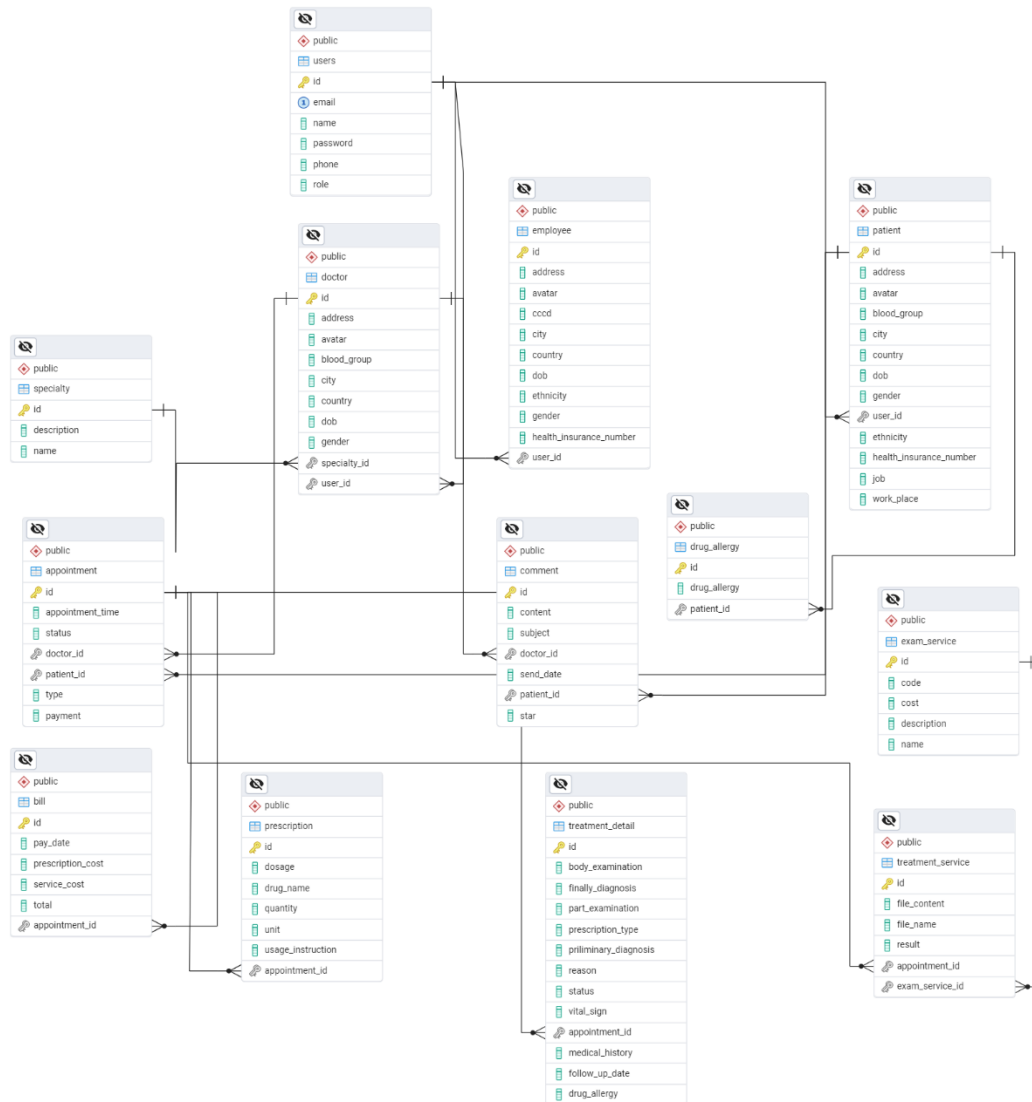
Tên Use Case	Quản lý bình luận
Actor	Bệnh nhân
Mục tiêu	Người dùng xem, sửa, xóa các bình luận của mình
Tiền điều kiện	Người dùng đã đăng nhập và hệ thống
Luồng hoạt động cơ bản	<ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình danh sách các bình luận của người dùng được phân loại review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

2.5 Biểu đồ lớp phân tích (Class Diagram)



Hình 2.28 (Biểu đồ lớp phân tích)

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



Hình 2.29 (Biểu đồ Quan hệ thực thể)

Bảng 2.9 Mô tả cơ sở dữ liệu

STT	Tên bảng	Mô tả
1	users	Lưu tài khoản người dùng
2	patient	Lưu thông tin cá nhân của bệnh nhân
3	doctor	Lưu thông tin cá nhân của bác sĩ
4	employee	Lưu thông tin cá nhân của nhân viên
5	appointment	Lưu thông tin lịch hẹn khám
6	treatment_detail	Lưu kết quả khám chi tiết

7	treatment_service	Lưu kết quả xét nghiệm
8	prescription	Lưu thông tin đơn thuốc
9	drug_allergy	Lưu thông tin tương tác thuốc của bệnh nhân
10	bill	Lưu thông tin hóa đơn
11	comment	Lưu nội dung bình luận
12	specialty	Lưu danh mục các chuyên khoa
13	exam_service	Lưu danh mục các dịch vụ xét nghiệm

2.7 Kết luận chương II

Trong chương này đã hoàn thành các bước phân tích và thiết kế hệ thống quản lý phòng khám, bao gồm việc xây dựng các biểu đồ use case, biểu đồ hoạt động, trích xuất lớp thực thể, và thiết kế cơ sở dữ liệu. Những nội dung này là cơ sở quan trọng để đảm bảo tính logic, hiệu quả và khả năng triển khai của hệ thống. Tiếp theo, chương III sẽ trình bày chi tiết các bước cài đặt và kiểm thử hệ thống quản lý phòng khám.

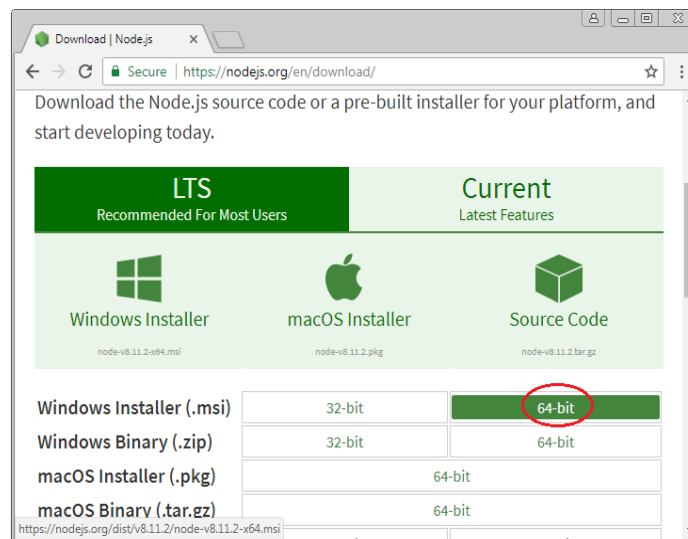
CHƯƠNG III. CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG

Từ những bước phân tích và thiết kế hệ thống chi tiết đã trình bày ở chương trước, chương này tập trung vào việc triển khai ứng dụng quản lý phòng khám. Giai đoạn này bao gồm việc lựa chọn môi trường phát triển, thực hiện lập trình, chạy thử hệ thống, và kiểm thử các chức năng chính. Cuối cùng, chương sẽ đánh giá kết quả đạt được, chỉ ra những ưu điểm, hạn chế, và đề xuất định hướng phát triển hệ thống trong tương lai.

3.1 Môi trường triển khai ứng dụng

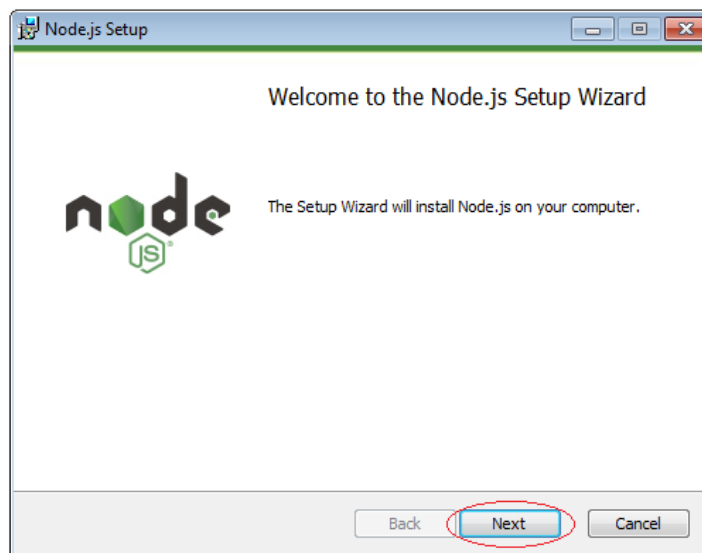
3.1.1 Cài đặt ReactJS cho giao diện

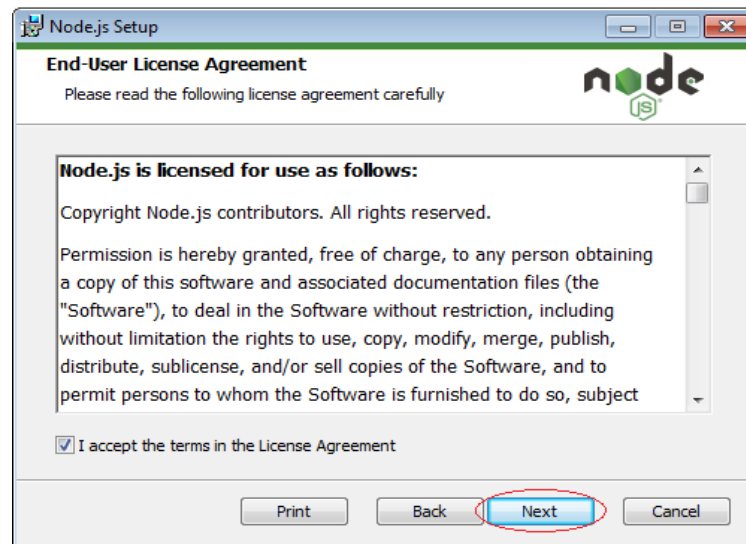
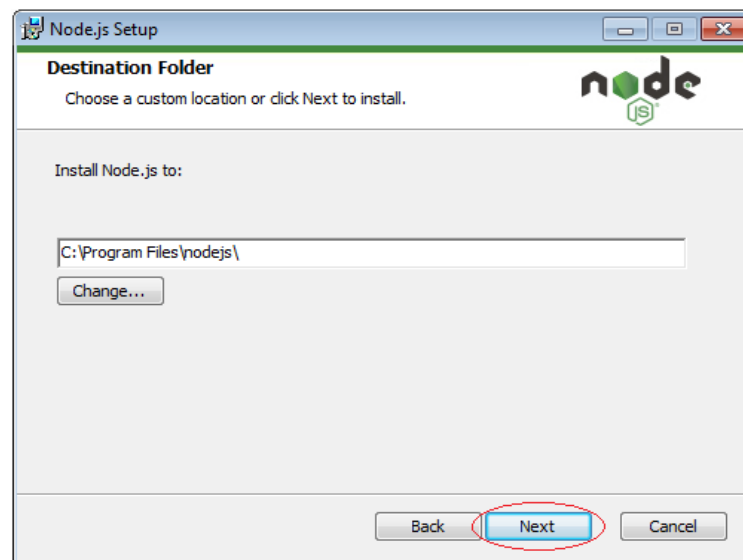
- Cài đặt NodeJS: Link download <https://nodejs.org/en/download>

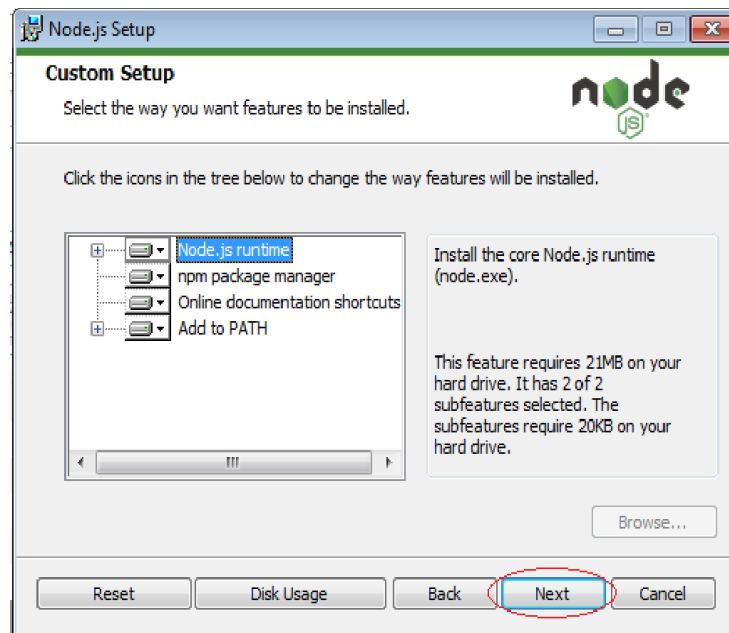


Hình 3.1 Cài đặt NodeJS bước 1

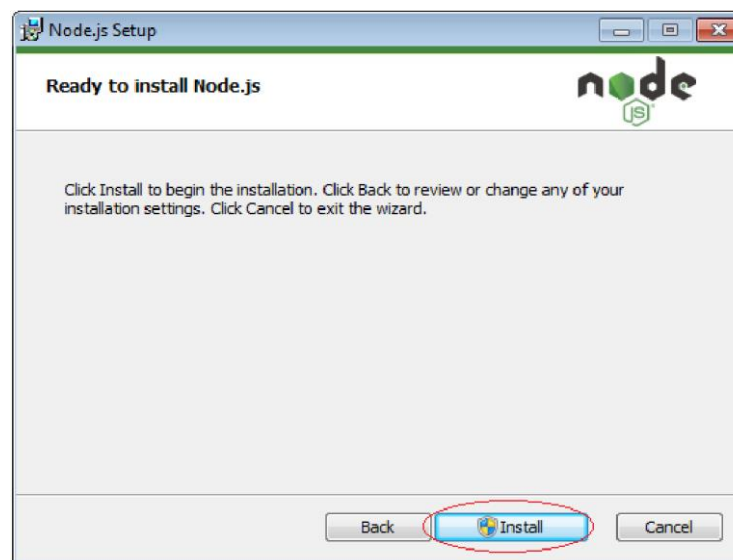
Sau khi tải xuống thành công, mở file vừa tải và cài đặt theo hướng dẫn.



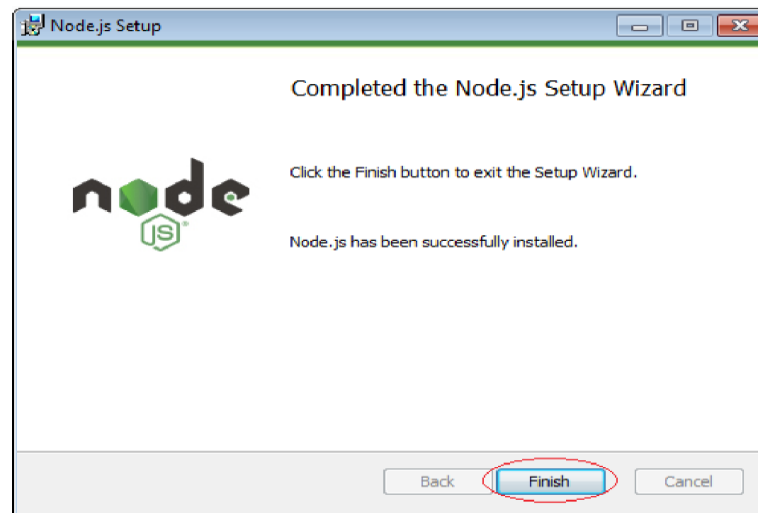
Hình 3.2 Cài đặt NodeJS bước 2*Hình 3.3 Cài đặt NodeJS bước 3**Hình 3.4 Cài đặt NodeJS bước 4*



Hình 3.5 Cài đặt NodeJS bước 5



Hình 3.6 Cài đặt NodeJS bước 6



Hình 3.7 Cài đặt NodeJS thành công

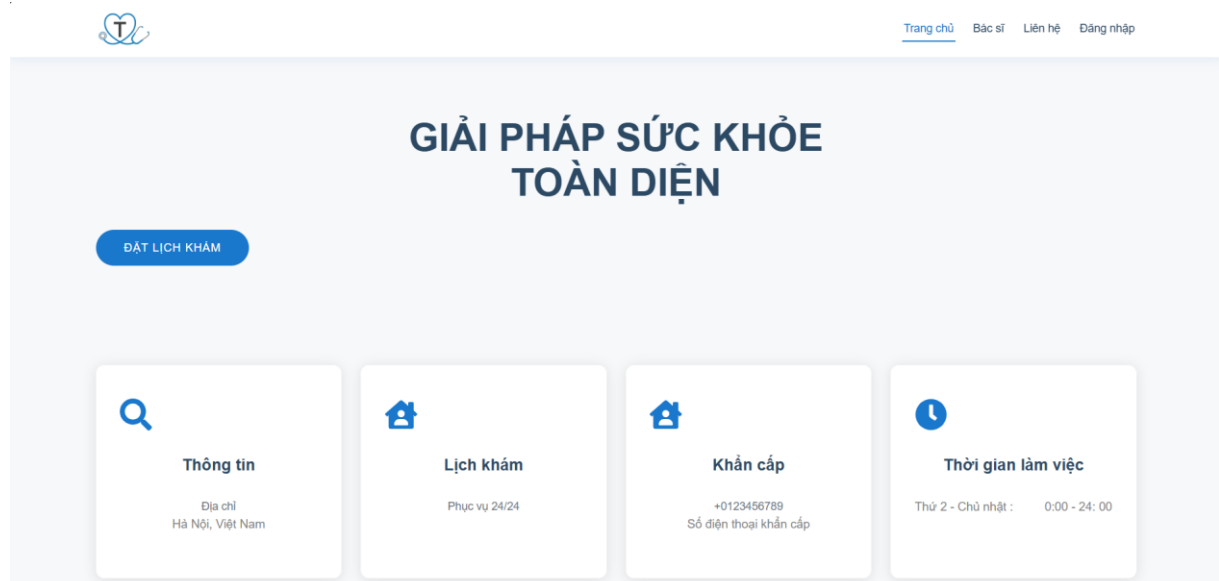
- Mở dự án lên và chạy dự án bằng lệnh: `npm run start`

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

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

Hình 3.8 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 3.9 Chạy dự án thành công

3.1.2 Cài đặt Java Spring Boot cho server

- Cài đặt Java Runtime Environment (JRE):

Java Runtime Environment (JRE) là môi trường thực thi tập lệnh chạy ứng dụng Java. Công nghệ bao gồm các thư viện class của Java, trình quản lý bộ nhớ, trình điều khiển cơ sở dữ liệu và các file hỗ trợ cần thiết khác.

Link cài đặt: https://www.java.com/en/download/windows_offline.jsp

- Cài đặt Java Development Kit (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.

JDK Development Kit 21.0.2 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the [limited free grants](#) of the OTN license will require a fee.

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	185.52 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	163.91 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	162.07MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)

Hình 3.10 Tải file JDK

- Cài đặt Maven:

+ Bước 1: Tải và giải nén file (Link tải: <https://maven.apache.org/download.cgi>)

Downloading Apache Maven 3.9.4

Apache Maven 3.9.4 is the latest release. It is the recommended version for all users.

System Requirements

Requirement	Details
Java Development Kit (JDK)	Maven 3.0+ requires JDK 8 or above to execute. It will allow you to build against 1.3 and other JDK versions for using m2plugins.
Memory	No minimum requirement.
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 300MB.
Operating System	No minimum requirement. Start-up scripts are included as shell scripts (based on many Unix flavors) and Windows batch files.

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public GPG's used by the Apache Maven developers.

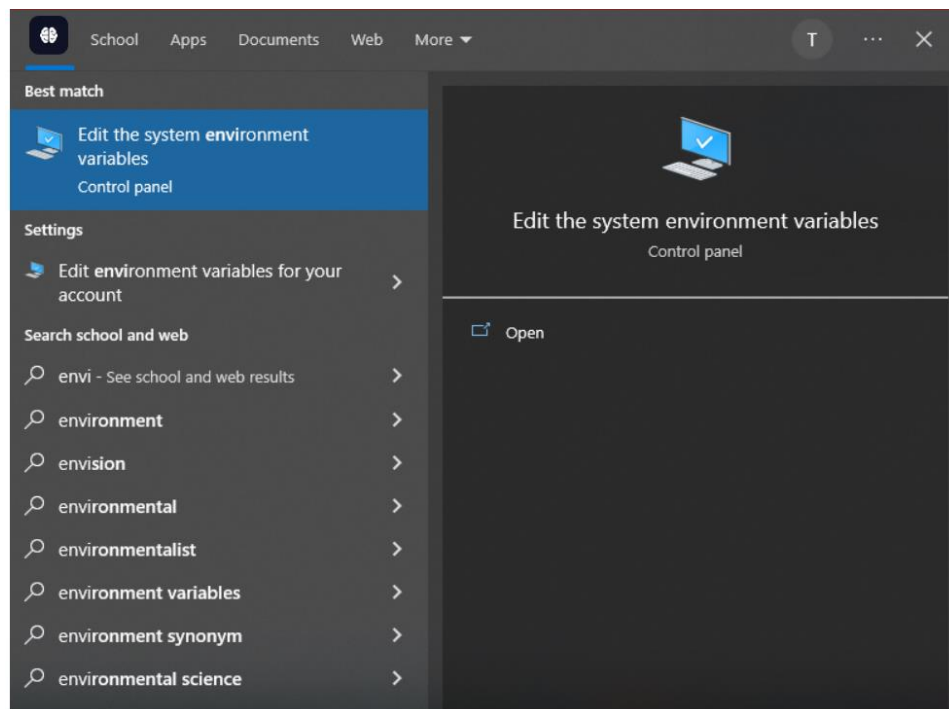
	LINK	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.4-bin.tar.gz	apache-maven-3.9.4-bin.tar.gz.sha1	apache-maven-3.9.4-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.4-bin.zip	apache-maven-3.9.4-bin.zip.sha1	apache-maven-3.9.4-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.4-src.tar.gz	apache-maven-3.9.4-src.tar.gz.sha1	apache-maven-3.9.4-src.tar.gz.asc
Source zip archive	apache-maven-3.9.4-src.zip	apache-maven-3.9.4-src.zip.sha1	apache-maven-3.9.4-src.zip.asc

• 3.9.4 Release Notes and Release Information Documentation
 • Used under the Apache License, version 2.0
 • Distributed under the Apache License, version 2.0
 • Other:
 • Apache Maven Website As Documentation Archive
 • All current release sources (plugins, source bundles, ...) available at <https://www.apache.org/maven/>

Hình 3.11 Tải file maven

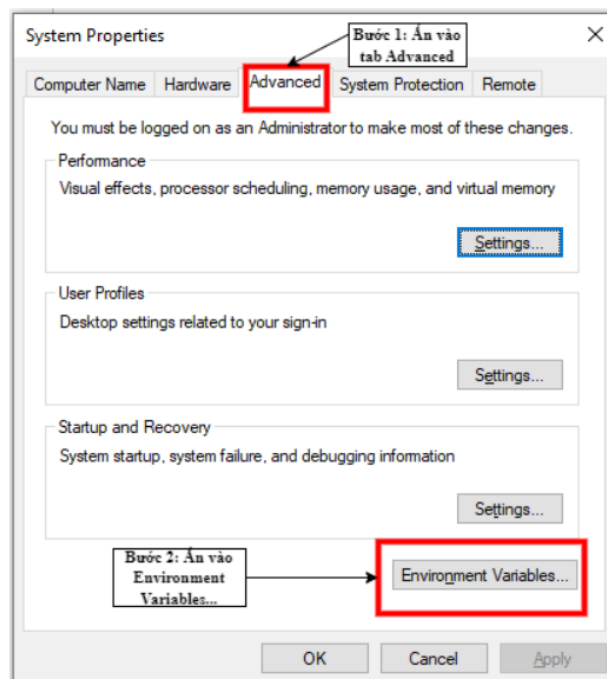
+ Bước 2: Tạo Maven_Home:

Search trên thanh window “environment variable”:



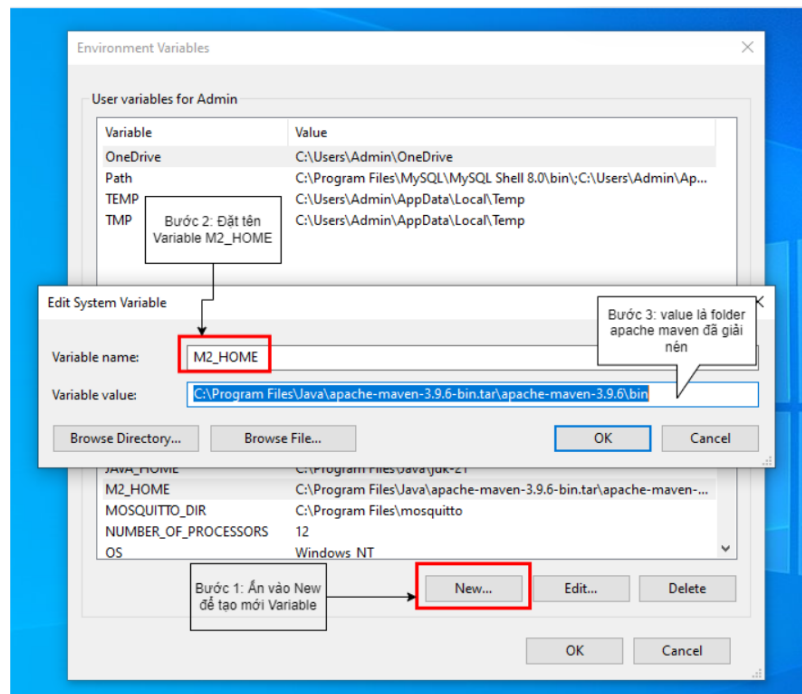
Hình 3.12 Tìm kiếm Environment variable

Sau khi hiện bảng System Properties, ta chọn Advanced và dưới thanh tab Advanced, click vào Environment Variables...



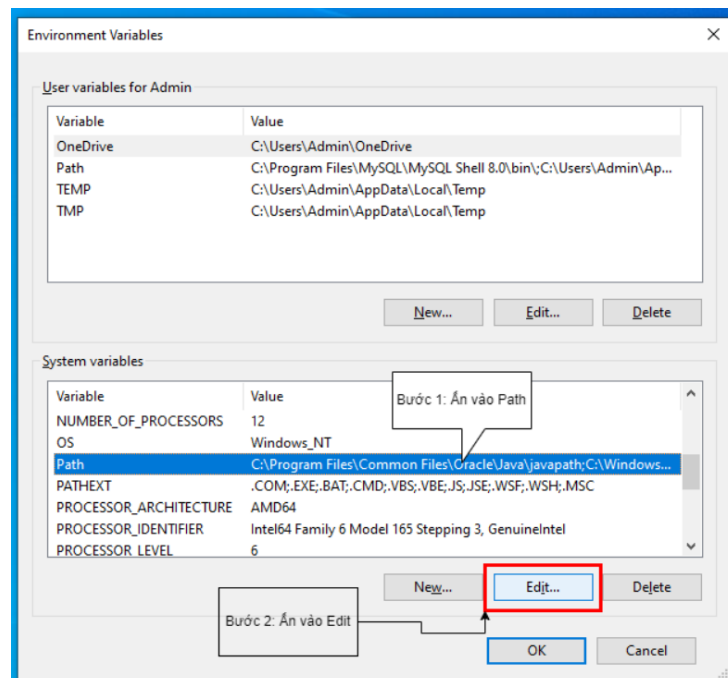
Hình 3.13 Bảng System Properties

Tạo biến Variable name là M2_Home với value là folder apache maven đã giải nén:



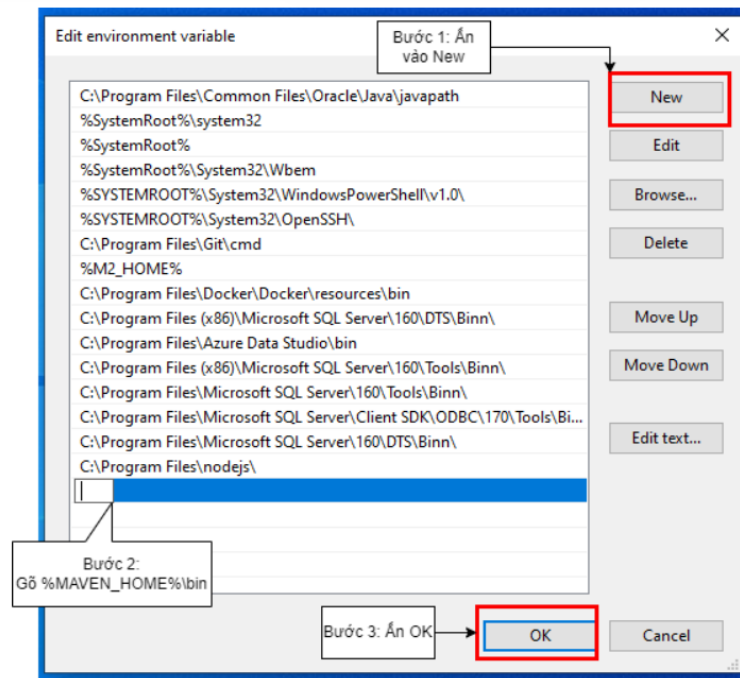
Hình 3.14 Tạo Variable M2_HOME

Sau khi đã tạo System Variable M2_HOME, chọn Path dưới System variable và ấn edit



Hình 3.15 Edit Path

Sau khi ấn Edit và hiện ra bảng Edit environment variable, ta chọn New ở thanh bên phải và gõ %MAVEN_HOME%\bin rồi nhấn Ok.



Hình 3.16 Tạo Path Maven Home

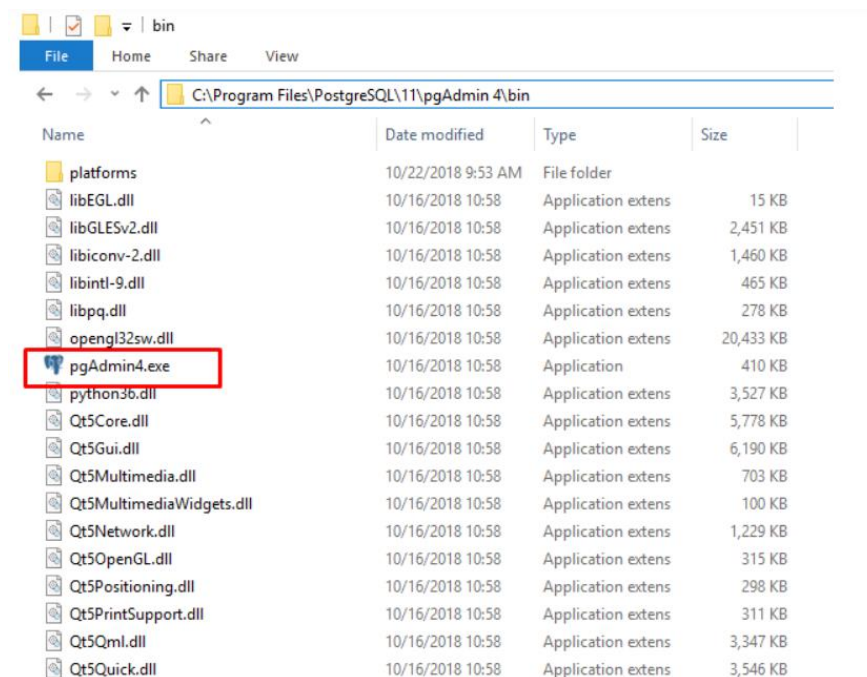
- Cài đặt IDE (IntelliJ) : Link tải

<https://www.jetbrains.com/idea/download/?section=windows>

3.1.3 Cài đặt PostgreSQL cho cơ sở dữ liệu

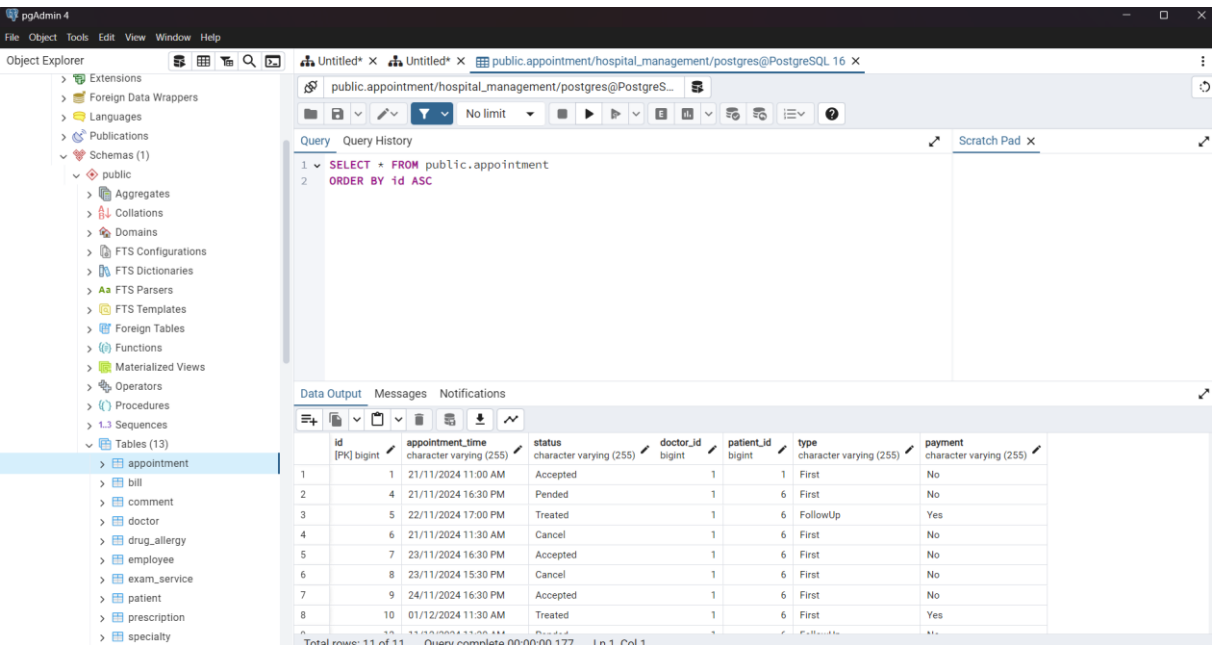
Link download PostgreSQL cho windows: <https://www.postgresql.org/download/windows/>

Sau khi tải và cài đặt thì phần mềm chạy sẽ nằm trong thư mục bin:



Hình 3.17 Ứng dụng khởi chạy PostgreSQL

Khi mở lên sẽ có giao diện giống như sau:



Hình 3.17 Giao diện PostgreSQL

3.2 Giao diện hệ thống

....

....

CHƯƠNG IV. KẾT LUẬN

4.1 Kết quả đạt được

....

....

4.2 Hạn chế của 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

4.3 Định hướng phát triển hệ thống

Nâng cấp giao diện người dùng để tăng tính thân thiện và hấp dẫn.

Tích hợp công nghệ AI/ML để phân tích dữ liệu y tế, đưa ra gợi ý điều trị hoặc quản lý hiệu quả hơn.

Phát triển thêm tính năng thông báo lịch hẹn qua email hoặc SMS để hỗ trợ người dùng.

Mở rộng tích hợp với các thiết bị IoT như máy đo huyết áp, đo đường huyết, để hỗ trợ theo dõi sức khỏe từ xa.

Triển khai hệ thống trên môi trường cloud để tăng tính sẵn sàng và khả năng mở rộng.

DANH MỤC TÀI LIỆU THAM KHẢO

Tài liệu, giáo trình:

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

[...] TS Nguyễn Đình Hóa, Slide môn cơ sở dữ liệu, Học viện Công nghệ Bưu chính Viễn thông

[...] ...

Trang web:

[1] <https://ailab.siu.edu.vn/article/46/huong-dan-cai-at-va-thuc-hanh-spring-boot-phan-1>

[2] <https://stackjava.com/postgresql/huong-dan-cai-dat-va-cau-hinh-postgresql-tren-windows.html>

[3] NodeJs Với Express FrameWork