

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống đặt lịch khám bệnh trực tuyến

Phạm Xuân Duy

duy.px200114@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: PGS.TS. Đặng Văn Chuyết

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 12/2024

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành đến các thầy cô trong Trường Đại học Bách Khoa Hà Nội, những người đã truyền đạt cho em những kiến thức quý báu trong suốt quá trình học tập tại trường.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến giảng viên hướng dẫn **PGS.TS. Đặng Văn Chuyết** – người thầy đã tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện đồ án. Những lời góp ý, định hướng và sự hỗ trợ nhiệt tình của thầy là nguồn động lực to lớn giúp em hoàn thành tốt đồ án này.

Xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn ở bên cạnh, động viên và hỗ trợ em trong quá trình học tại Đại học Bách Khoa nói chung và quá trình thực hiện Đồ án tốt nghiệp nói riêng.

Dù đã cố gắng hết sức, song đồ án không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ thầy cô và các bạn để hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Đồ án "*Hệ thống đặt lịch khám bệnh trực tuyến*" tập trung vào việc phát triển hệ thống đặt lịch khám bệnh trực tuyến, giúp kết nối giữa bệnh nhân và cơ sở y tế, bác sĩ một cách thuận tiện, nhanh chóng. Hệ thống bao gồm các chức năng chính:

Quản lý thông tin tài khoản người dùng: Đăng ký, đăng nhập và cập nhật thông tin cá nhân cho cả bệnh nhân và bác sĩ.

Tìm kiếm: cho phép bệnh nhân tìm kiếm bác sĩ phù hợp theo chuyên khoa, cơ sở y tế,...

Đặt lịch khám: Cho phép bệnh nhân tra cứu lịch trống của bác sĩ và đặt lịch khám theo thời gian mong muốn.

Hiển thị thông tin chi tiết: Cho phép người dùng xem được thông tin chi tiết của bác sĩ, chuyên khoa, cơ sở y tế, bài viết,... hỗ trợ cho người dùng chọn đúng bác sĩ, cơ sở y tế phù hợp mà mình muốn đến khám.

Hệ thống được thiết kế trên kiến trúc Client-Server với frontend sử dụng [ReactJS] và backend dựa trên [NodeJS/ExpressJS]. Dữ liệu được lưu trữ và quản lý bởi cơ sở dữ liệu [MySQL].

Hệ thống đặt lịch khám bệnh không chỉ nâng cao trải nghiệm người dùng mà còn giảm thiểu thời gian chờ đợi và tối ưu hóa quy trình quản lý tại các cơ sở y tế.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

MỤC LỤC

| | |
|-----------------------------------------------------------------------|----------|
| CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI..... | 1 |
| 1.1 Đặt vấn đề..... | 1 |
| 1.2 Mục tiêu và phạm vi đề tài..... | 2 |
| 1.3 Bố cục đồ án | 3 |
| CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU..... | 4 |
| 2.1 Khảo sát hiện trạng | 4 |
| 2.2 Tổng quan chức năng | 6 |
| 2.2.1 Biểu đồ ca sử dụng phân rã “Tìm kiếm” | 7 |
| 2.2.2 Biểu đồ ca sử dụng phân rã “Bình luận” | 7 |
| 2.2.3 Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bệnh nhân”..... | 8 |
| 2.2.4 Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bác sĩ” | 8 |
| 2.2.5 Biểu đồ ca sử dụng phân rã “Quản lý thông tin cơ sở y tế” | 9 |
| 2.2.6 Biểu đồ ca sử dụng phân rã “Quản lý thông tin chuyên khoa”..... | 9 |
| 2.2.7 Biểu đồ ca sử dụng phân rã “Quản lý bài viết” | 10 |
| 2.2.8 Biểu đồ ca sử dụng phân rã “Quản lý lịch hẹn” | 10 |
| 2.3 Quy trình nghiệp vụ | 11 |
| 2.4 Đặc tả chức năng | 13 |
| 2.4.1 Đặc tả ca sử dụng “Đăng ký tài khoản bác sĩ” | 13 |
| 2.4.2 Đặc tả ca sử dụng “Xác nhận tài khoản bác sĩ”..... | 14 |
| 2.4.3 Đặc tả ca sử dụng “Tìm kiếm bác sĩ” | 15 |
| 2.4.4 Đặc tả ca sử dụng “Cập nhật lịch làm việc” | 16 |
| 2.5 Yêu cầu phi chức năng | 17 |

| | |
|------------------------------------------------------|-----------|
| CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG..... | 18 |
| 3.1 Thiết kế kiến trúc..... | 18 |
| 3.1.1 Sơ đồ gói..... | 18 |
| 3.1.2 Mối quan hệ giữa các thư mục..... | 21 |
| 3.2 Thiết kế cơ sở dữ liệu | 22 |
| 3.2.1 Bảng "Users" | 23 |
| 3.2.2 Bảng "DoctorInfos" | 24 |
| 3.2.3 Bảng "Specialties" | 25 |
| 3.2.4 Bảng "DoctorSpecialties" | 25 |
| 3.2.5 Bảng "Clinics" | 26 |
| 3.2.6 Bảng "Notebooks" | 27 |
| 3.2.7 Bảng "NotebookAuthors" | 27 |
| 3.2.8 Bảng "NotebookCensors" | 28 |
| 3.2.9 Bảng "Schedules" | 28 |
| 3.2.10 Bảng "Bookings" | 29 |
| 3.2.11 Bảng "Comments" | 30 |
| CHƯƠNG 4. CÀI ĐẶT TRIỂN KHAI VÀ KẾT QUẢ | 31 |
| 4.1 Công nghệ sử dụng | 31 |
| 4.1.1 Công nghệ sử dụng cho Back-end..... | 31 |
| 4.1.2 Công nghệ sử dụng cho Front-end | 31 |
| 4.1.3 Công nghệ sử dụng trong Database..... | 32 |
| 4.2 Cài đặt triển khai | 34 |
| 4.3 Kết quả đạt được..... | 36 |
| 4.3.1 Chức năng "Đăng ký tài khoản bác sĩ" | 36 |
| 4.3.2 Chức năng "Cập nhật lịch làm việc"..... | 37 |
| 4.3.3 Chức năng "Quản lý lịch hẹn khám bệnh" | 37 |

| | |
|------------------------------------------------------------|-----------|
| 4.3.4 Chức năng "Hiển thị thông tin chi tiết bác sĩ" | 38 |
| 4.3.5 Chức năng "Đặt lịch khám bệnh" | 39 |
| 4.3.6 Chức năng "Quản lý tài khoản bệnh nhân" | 40 |
| 4.3.7 Chức năng "Quản lý tài khoản bác sĩ" | 40 |
| 4.3.8 Chức năng "Quản lý thông tin cơ sở y tế" | 40 |
| CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 42 |
| 5.1 Kết luận | 42 |
| 5.1.1 Nhận Định Về Thị Trường Hiện Tại | 42 |
| 5.1.2 Giá Trị Của Hệ Thống Được Phát Triển | 42 |
| 5.1.3 Những Vấn Đề Chưa Giải Quyết..... | 42 |
| 5.1.4 Bài Học Kinh Nghiệm | 42 |
| 5.2 Hướng phát triển trong tương lai | 43 |
| 5.2.1 Phát Triển Tính Năng Tìm Kiếm Nâng Cao | 43 |
| 5.2.2 Mở Rộng Phạm Vi Hoạt Động..... | 43 |
| 5.2.3 Tối Ưu Hóa Trải Nghiệm Người Dùng..... | 43 |
| 5.2.4 Cập Nhật Tình Trạng Lịch Theo Thời Gian Thực | 43 |

DANH MỤC HÌNH VẼ

| | | |
|-----------|---------------------------------------------------------------------------------|----|
| Hình 2.1 | Sếp hàng lấy số khám bệnh tại bệnh viện Bạch Mai - Hà Nội . | 4 |
| Hình 2.2 | Biểu đồ ca sử dụng tổng quát | 6 |
| Hình 2.3 | Biểu đồ ca sử dụng phân rã “Tìm kiếm” | 7 |
| Hình 2.4 | Biểu đồ ca sử dụng phân rã “Bình luận” | 7 |
| Hình 2.5 | Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bệnh nhân” . . | 8 |
| Hình 2.6 | Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bác sĩ” | 8 |
| Hình 2.7 | Biểu đồ ca sử dụng phân rã “Quản lý thông tin cơ sở y tế” . . . | 9 |
| Hình 2.8 | Biểu đồ ca sử dụng phân rã “Quản lý thông tin chuyên khoa” . | 9 |
| Hình 2.9 | Biểu đồ ca sử dụng phân rã “Quản lý bài viết” | 10 |
| Hình 2.10 | Biểu đồ ca sử dụng phân rã “Quản lý lịch hẹn” | 10 |
| Hình 2.11 | Biểu đồ ca sử dụng phân rã “CRUD bình luận” | 11 |
| Hình 3.1 | Biểu đồ phụ thuộc gói phần máy chủ Nodejs | 18 |
| Hình 3.2 | Biểu đồ phụ thuộc gói phần giao diện | 20 |
| Hình 3.3 | Sơ đồ tổng quan cơ sở dữ liệu | 22 |
| Hình 4.1 | Mô hình client - server | 34 |
| Hình 4.2 | Giao diện thể hiện các trường thông tin bác sĩ cần điền để đăng ký | 36 |
| Hình 4.3 | Giao diện quản lý lịch làm việc | 37 |
| Hình 4.4 | Giao diện quản lý lịch hẹn khám bệnh | 37 |
| Hình 4.5 | Thông tin chi tiết bác sĩ | 38 |
| Hình 4.6 | Chức năng "Đặt lịch khám bệnh" | 39 |
| Hình 4.7 | Chức năng "Quản lý tài khoản bệnh nhân" | 40 |
| Hình 4.8 | Chức năng "Quản lý tài khoản bác sĩ" | 40 |
| Hình 4.9 | Chức năng "Quản lý thông tin cơ sở y tế" | 41 |

DANH MỤC BẢNG BIỂU

| | | |
|----------|---------------------------------------------------------|----|
| Bảng 2.1 | Đặc tả ca sử dụng “Đăng ký tài khoản bác sĩ” | 13 |
| Bảng 2.2 | Đặc tả ca sử dụng “Xác nhận tài khoản bác sĩ” | 14 |
| Bảng 2.3 | Đặc tả ca sử dụng “Tìm kiếm bác sĩ” | 15 |
| Bảng 2.4 | Đặc tả ca sử dụng “Cập nhật đơn đặt phòng” | 16 |

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Đặt lịch khám bệnh tại các bệnh viện và phòng khám đang trở thành một vấn đề được quan tâm ngày càng nhiều trong bối cảnh sự gia tăng dân số và nhu cầu chăm sóc sức khỏe không ngừng tăng cao. Hiện nay, người dân thường phải đối mặt với nhiều khó khăn khi tiếp cận dịch vụ y tế, bao gồm thời gian chờ đợi dài, quá tải tại các bệnh viện lớn, và khó khăn trong việc sắp xếp lịch hẹn khám phù hợp với lịch làm việc cá nhân. Điều này không chỉ ảnh hưởng đến chất lượng chăm sóc sức khỏe mà còn tạo ra sự bất tiện và mất thời gian cho cả bệnh nhân lẫn đội ngũ y tế.

Thực tế, hệ thống y tế ở nước ta, dù đã có những cải tiến đáng kể, nhưng vẫn còn tồn tại nhiều hạn chế trong việc tổ chức và quản lý lịch khám bệnh một cách hiệu quả. Nhiều bệnh viện và phòng khám chưa ứng dụng triệt để công nghệ để tối ưu hóa quy trình đặt lịch, dẫn đến tình trạng xếp hàng dài, khó kiểm soát số lượng bệnh nhân, và đôi khi là việc phải đến trực tiếp để đặt lịch hẹn. Những bất cập này đã làm giảm sự hài lòng của bệnh nhân và ảnh hưởng đến chất lượng dịch vụ y tế.

Vì vậy, vấn đề đặt ra là làm thế nào để xây dựng một hệ thống đặt lịch khám bệnh tiện lợi, nhanh chóng và hiệu quả, đáp ứng được nhu cầu của người dân trong thời đại công nghệ số hiện nay. Sự kết hợp giữa công nghệ thông tin và y tế có thể mang lại giải pháp giúp giảm thiểu thời gian chờ đợi, cải thiện trải nghiệm của bệnh nhân và nâng cao chất lượng dịch vụ y tế tại Hà Nội.

1.2 Mục tiêu và phạm vi đề tài

Mục tiêu

- Hỗ trợ bệnh nhân:
 - Giúp bệnh nhân dễ dàng đặt lịch khám qua mạng mọi lúc, mọi nơi.
 - Tiết kiệm thời gian so với việc đến trực tiếp cơ sở y tế lấy số và chờ khám bệnh.
 - Cung cấp thông tin chi tiết về các bác sĩ, cơ sở y tế, dịch vụ khám và lịch trống.
- Hỗ trợ cơ sở y tế:
 - Tự động hóa quy trình quản lý lịch khám, giảm tải công việc thủ công.
 - Tối ưu hóa sắp xếp lịch khám và phân bổ tài nguyên y tế.
 - Nâng cao chất lượng dịch vụ và sự hài lòng của bệnh nhân.

Phạm vi

- Chức năng chính:
 - Đăng ký/Đăng nhập tài khoản cho bệnh nhân, bác sĩ và quản trị viên.
 - Xem thông tin bác sĩ, dịch vụ khám bệnh và lịch làm việc.
 - Đặt lịch khám trực tuyến và quản lý lịch hẹn.
- Đối tượng sử dụng:
 - Bệnh nhân: Xem thông tin bác sĩ, cơ sở y tế, chuyên khoa,..., đặt lịch, xem lịch sử khám.
 - Bác sĩ: Cập nhật lịch làm việc, quản lý lịch khám, xác nhận lịch hẹn.
 - Quản trị viên: Quản lý thông tin tài khoản, quản lý thông tin bác sĩ, bệnh viên, chuyên khoa, bài viết,...
- Giới hạn:
 - Chỉ triển khai trên nền tảng web.
 - Phạm vi áp dụng cho một hoặc một số cơ sở y tế cụ thể.
 - Không bao gồm việc phát triển hệ thống quản lý chuyên sâu hồ sơ bệnh án (trừ thông tin cơ bản).

1.3 Bố cục đề án

- **Chương 2** trình bày về khảo sát hiện trạng từ đó phân tích yêu cầu đặt ra, quy trình nghiệp vụ thực tế để đưa ra tổng quan về các chức năng cũng như các yêu cầu phi chức năng của hệ thống.
- **Chương 3** sẽ trình bày về phân tích thiết kế hệ thống bao gồm thiết kế kiến trúc, thiết kế chi tiết từ đó xây dựng hệ thống.
- **Chương 4** cài đặt triển khai và nêu kết quả của đề tài, các khó khăn khi triển khai dự án và cách giải quyết. Đưa ra các kết quả đạt được cũng như quá trình kiểm thử và triển khai hệ thống trên thực tế.
- Trong **Chương 5**, em sẽ trình bày kết luận về đề án tốt nghiệp và hướng phát triển trong thời gian sắp tới để cải thiện hệ thống trở nên tối ưu hơn.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng



Hình 2.1: Sếp hàng lấy số khám bệnh tại bệnh viện Bạch Mai - Hà Nội

Hiện nay, tại Việt Nam, quy trình khám bệnh tại nhiều cơ sở y tế, đặc biệt là bệnh viện công lập, vẫn yêu cầu người bệnh phải đến trực tiếp để lấy số thứ tự. Ta có thể thấy tình trạng này diễn ra ở tất cả các bệnh viện, đặc biệt bệnh viện lớn như bệnh viện Bạch Mai thường xảy ra tình trạng này như hình 2.1 trên.

Điều này thường dẫn đến tình trạng xếp hàng dài hoặc chen lấn, đặc biệt trong giờ cao điểm hay tại các bệnh viện lớn. Thời gian chờ đợi thường kéo dài hàng giờ, gây mệt mỏi và bất tiện, nhất là đối với người già, trẻ nhỏ hoặc bệnh nhân nặng. Bên cạnh đó, sự quá tải tại các bệnh viện lớn khiến không gian chờ đợi trở nên chật chội và khó chịu. Việc quản lý số thứ tự vẫn chủ yếu thủ công, chưa có sự hỗ trợ từ các công cụ hiện đại như hệ thống đặt lịch trực tuyến, dẫn đến lãng phí thời gian và nguồn lực. Hiện trạng này không chỉ làm giảm trải nghiệm của người bệnh mà còn ảnh hưởng đến hiệu quả vận hành của các cơ sở y tế, đặc biệt là các bệnh viện tuyến trên.

Từ những bất cập trong quy trình khám bệnh hiện tại, có thể thấy rằng việc cải thiện trải nghiệm của người bệnh và tối ưu hóa quy trình quản lý tại các cơ sở y tế là vô cùng cần thiết. Hệ thống đặt lịch khám bệnh trực tuyến là một giải pháp hiệu

quả nhằm giải quyết những hạn chế tồn tại, như việc phải xếp hàng dài để lấy số thứ tự, thời gian chờ đợi kéo dài, và tình trạng quá tải tại các bệnh viện lớn. Với hệ thống này, người bệnh có thể chủ động tra cứu thông tin, đặt lịch khám phù hợp với thời gian cá nhân, giúp tiết kiệm thời gian và tránh được những phiền toái khi phải chờ đợi tại bệnh viện.

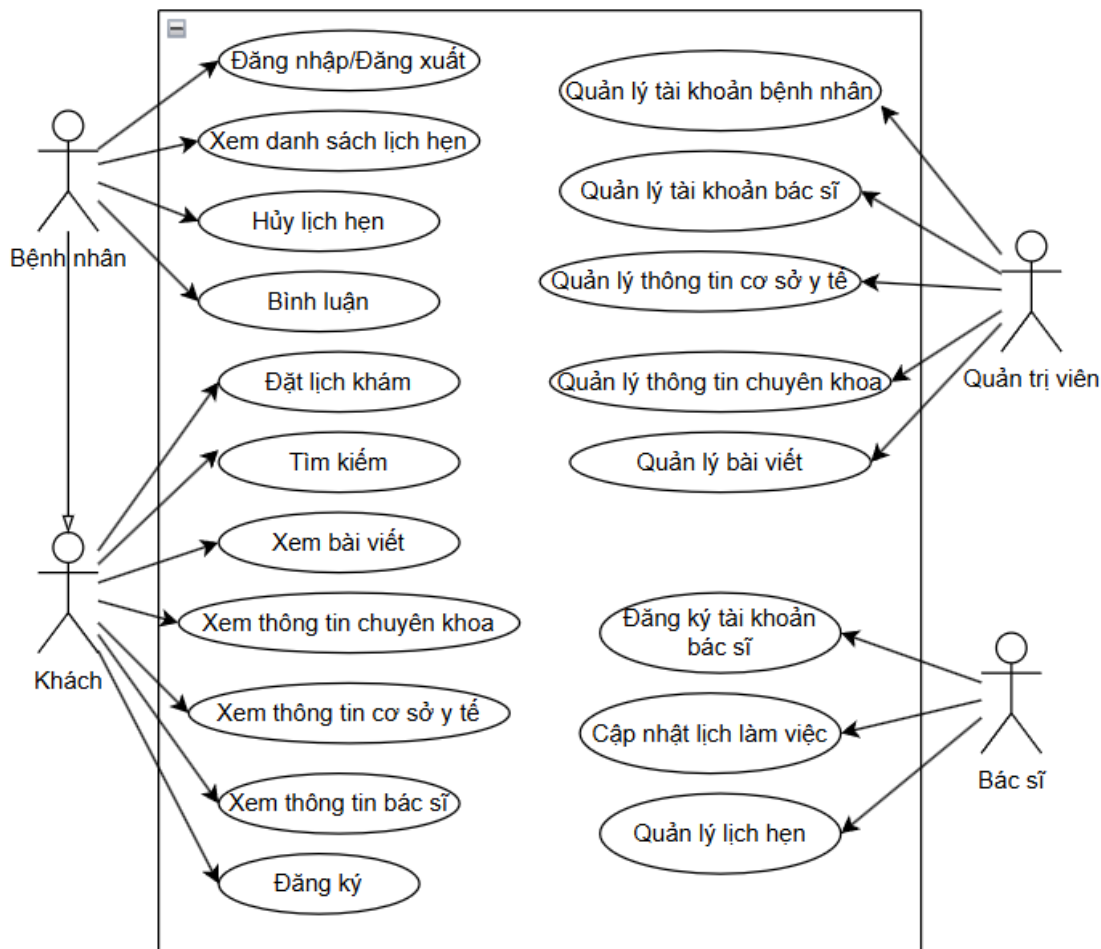
Đặc biệt, trong bối cảnh chuyển đổi số đang được thúc đẩy mạnh mẽ, hệ thống đặt lịch khám bệnh trực tuyến không chỉ là một bước tiến quan trọng trong việc hiện đại hóa ngành y tế mà còn mang lại những giá trị lâu dài cho xã hội. Giải pháp này không chỉ giải quyết những vấn đề trước mắt mà còn định hình một quy trình khám bệnh chuyên nghiệp, tiện lợi và hiệu quả hơn, góp phần nâng cao sự hài lòng của người bệnh và cải thiện hiệu suất hoạt động của các cơ sở y tế.

2.2 Tổng quan chức năng

Các tác nhân tham gia trong hệ thống bao gồm :

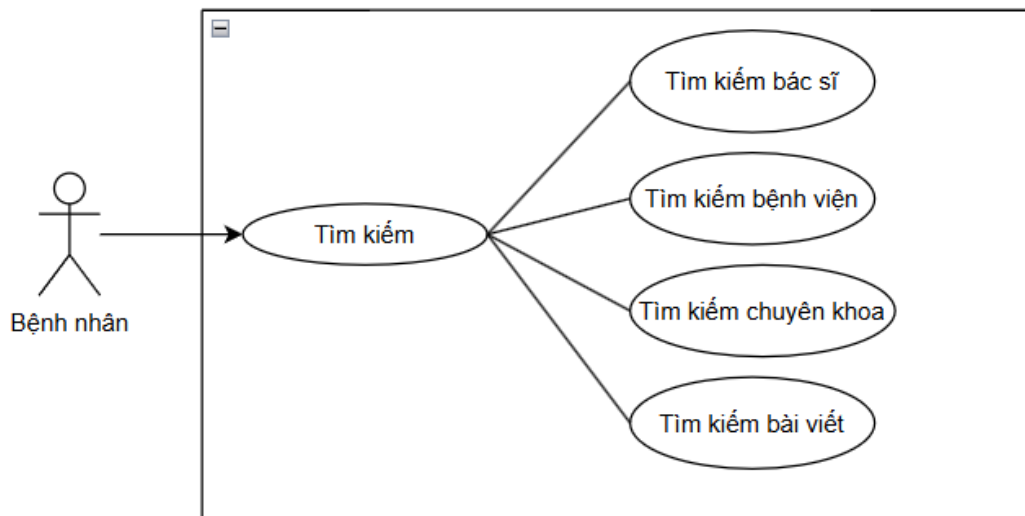
- **Khách:** Là người sử dụng hệ thống để tìm kiếm thông tin và đặt lịch khám bệnh.
- **Bệnh nhân:** Là khách đã đăng ký tài khoản hoặc đã đặt lịch khám bệnh qua hệ thống.
- **Bác sĩ:** Là người thực hiện đăng ký tài khoản với vai trò bác sĩ ở các bệnh viện, cơ sở y tế. Bác sĩ có thể thực hiện chức năng cập nhập lịch làm việc và quản lý lịch hẹn.
- **Quản trị hệ thống:** Là người liên hệ với các bệnh viện. Quản trị viên có vai trò quản lý thông tin bệnh viện, quản lý tài khoản, quản lý thông tin chuyên khoa, quản lý bài viết.

Trên đây là hình 2.2, thể hiện biểu đồ ca sử dụng tổng quát.



Hình 2.2: Biểu đồ ca sử dụng tổng quát

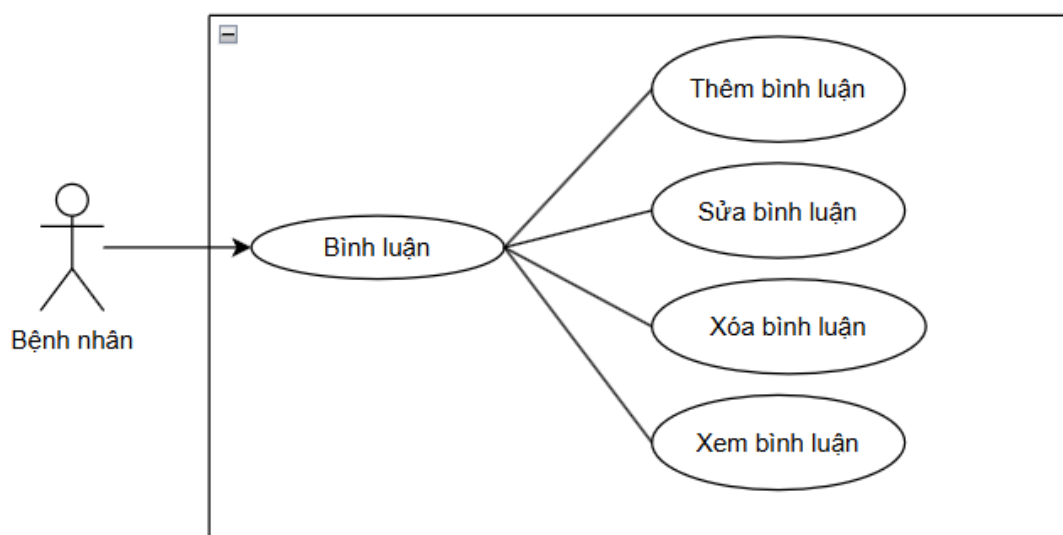
2.2.1 Biểu đồ ca sử dụng phân rã “Tìm kiếm”



Hình 2.3: Biểu đồ ca sử dụng phân rã “Tìm kiếm”

Trên đây là hình 2.3, biểu đồ phân rã chức năng của “tìm kiếm” có các chức năng như tìm kiếm bác sĩ, tìm kiếm chuyên khoa, tìm kiếm cơ sở y tế, tìm kiếm bài viết.

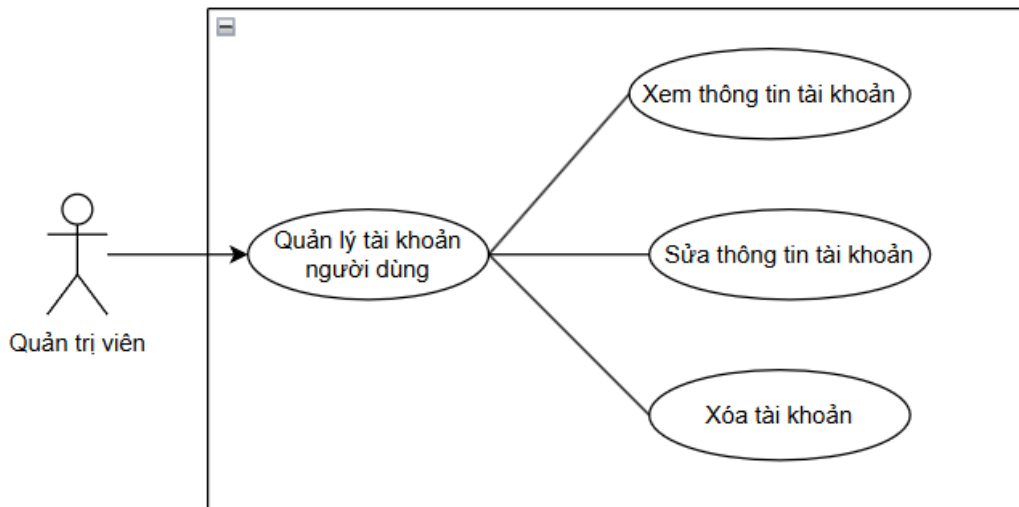
2.2.2 Biểu đồ ca sử dụng phân rã “Bình luận”



Hình 2.4: Biểu đồ ca sử dụng phân rã “Bình luận”

Trên đây là hình 2.4, biểu đồ phân rã chức năng “Bình luận” có các chức năng như xem, thêm, sửa, xóa bình luận.

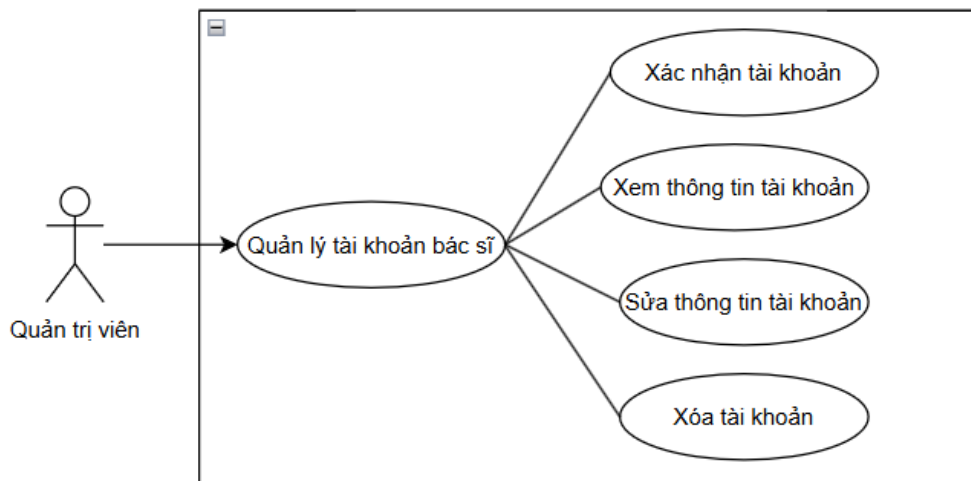
2.2.3 Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bệnh nhân”



Hình 2.5: Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bệnh nhân”

Trên đây là hình 2.5, biểu đồ phân rã chức năng của “Quản lý tài khoản bệnh nhân” có các chức năng như xem, sửa, xóa tài khoản bệnh nhân.

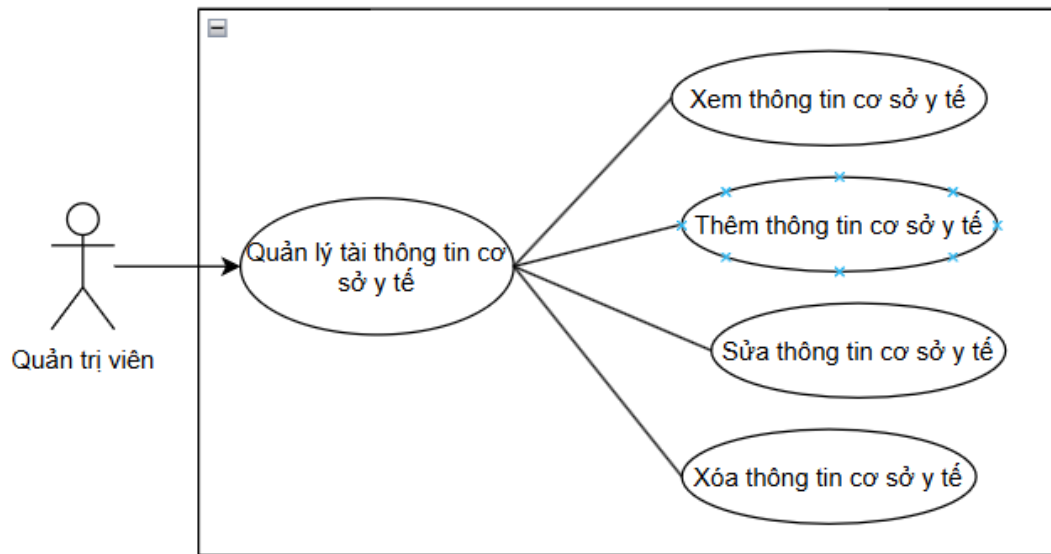
2.2.4 Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bác sĩ”



Hình 2.6: Biểu đồ ca sử dụng phân rã “Quản lý tài khoản bác sĩ”

Trên đây là hình 2.6, biểu đồ phân rã chức năng của “Quản lý tài khoản bác sĩ” có các chức năng như xác nhận, xem, sửa, xóa tài khoản bác sĩ.

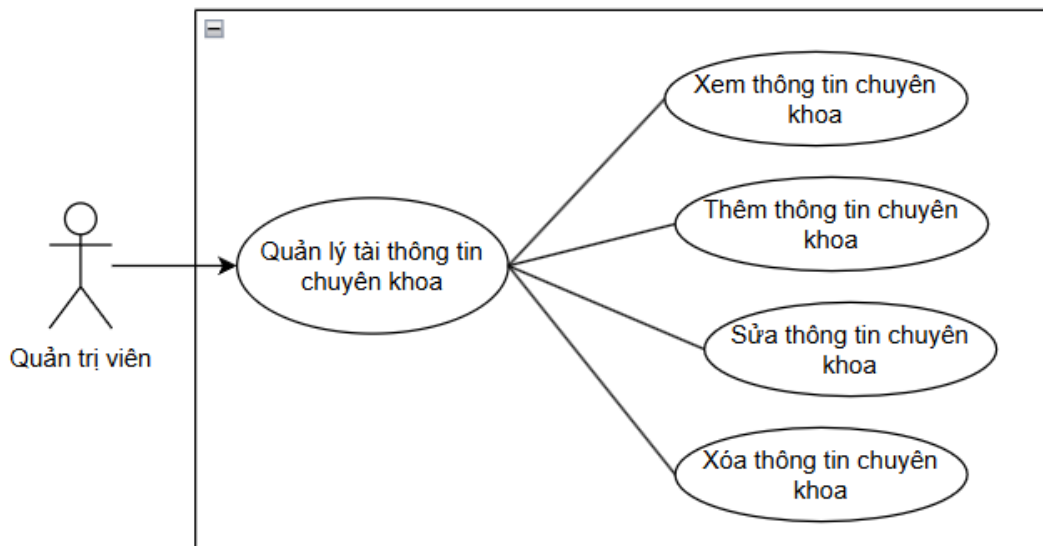
2.2.5 Biểu đồ ca sử dụng phân rã “Quản lý thông tin cơ sở y tế”



Hình 2.7: Biểu đồ ca sử dụng phân rã “Quản lý thông tin cơ sở y tế”

Trên đây là hình 2.7, biểu đồ phân rã chức năng của “Quản lý thông tin cơ sở y tế” có các chức năng như thêm, xem, sửa, xóa thông tin cơ sở y tế.

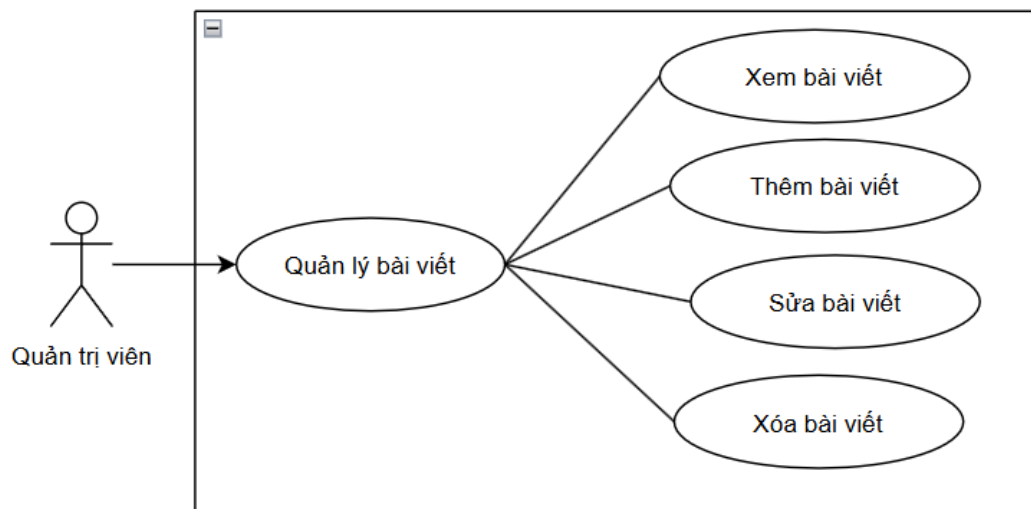
2.2.6 Biểu đồ ca sử dụng phân rã “Quản lý thông tin chuyên khoa”



Hình 2.8: Biểu đồ ca sử dụng phân rã “Quản lý thông tin chuyên khoa”

Trên đây là hình 2.8, biểu đồ phân rã chức năng của “Quản lý thông tin chuyên khoa” có các chức năng như thêm, xem, sửa, xóa thông tin chuyên khoa.

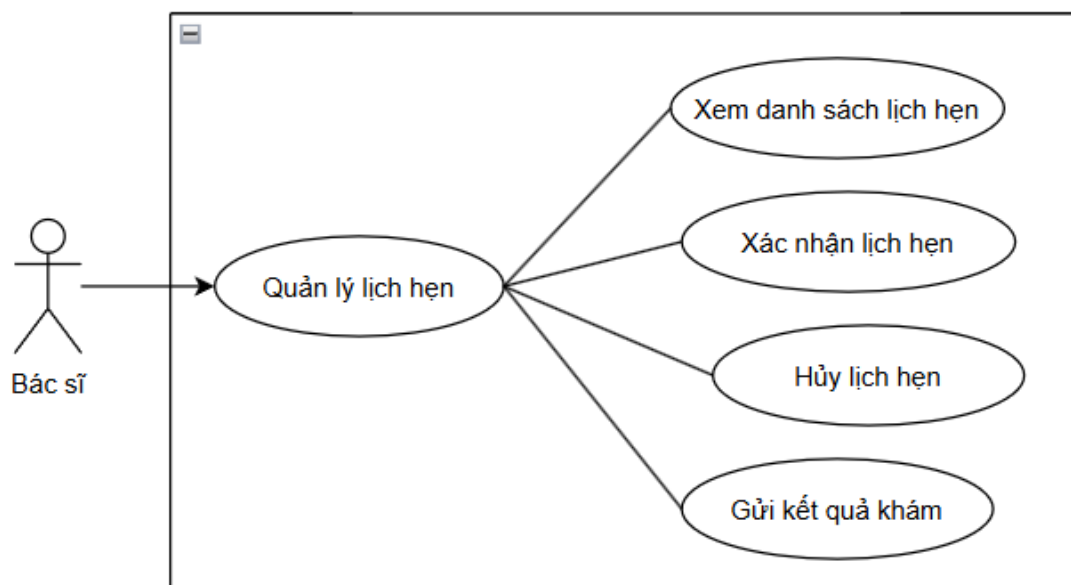
2.2.7 Biểu đồ ca sử dụng phân rã “Quản lý bài viết”



Hình 2.9: Biểu đồ ca sử dụng phân rã “Quản lý bài viết”

Trên đây là hình 2.9, biểu đồ phân rã chức năng của “Quản lý bài viết” có các chức năng như thêm, xem, sửa, xóa bài viết.

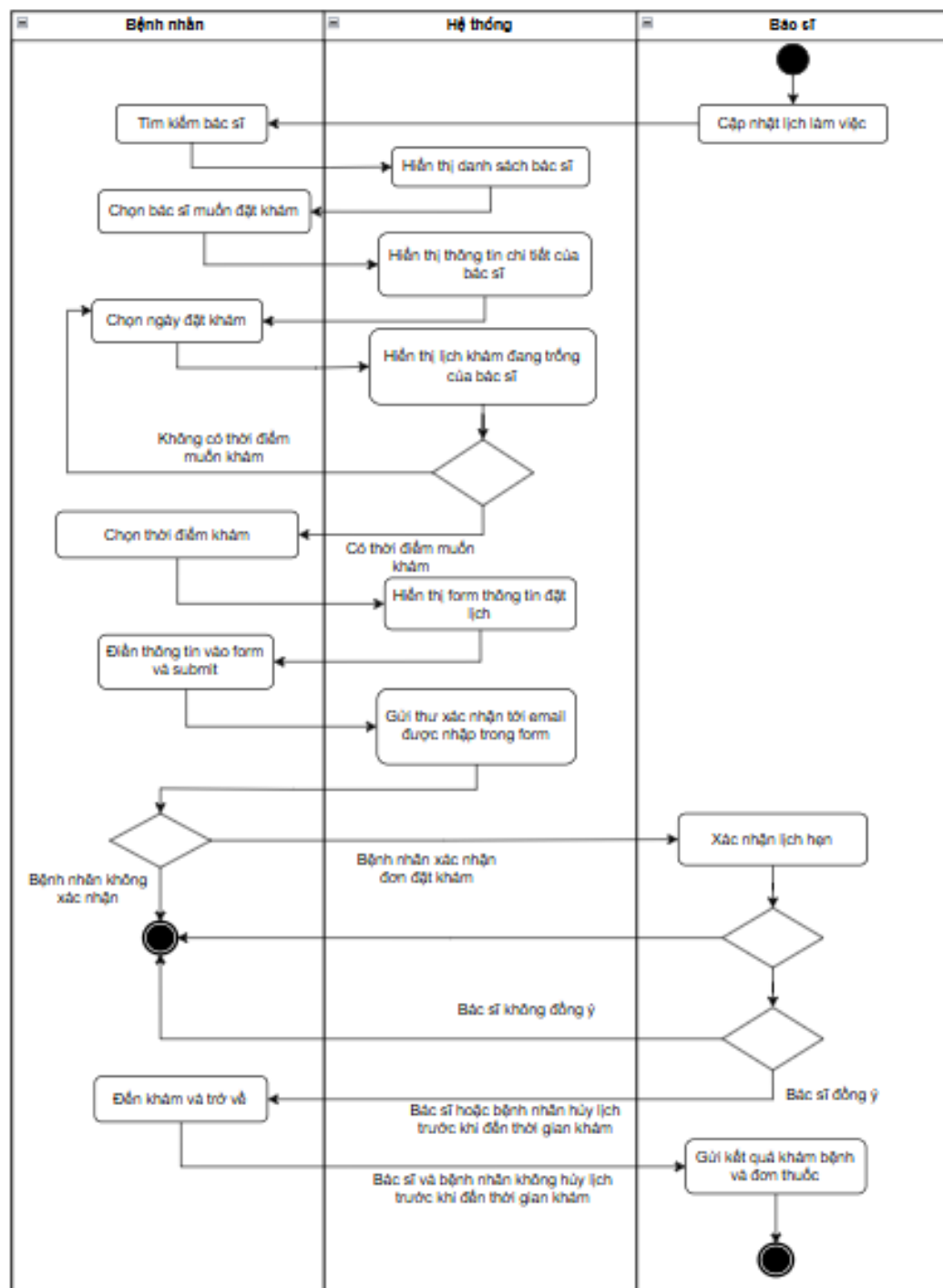
2.2.8 Biểu đồ ca sử dụng phân rã “Quản lý lịch hẹn”



Hình 2.10: Biểu đồ ca sử dụng phân rã “Quản lý lịch hẹn”

Trên đây là hình 2.10, biểu đồ phân rã chức năng của “Quản lý lịch hẹn” có các chức năng như thêm, xem, sửa, xóa bài viết.

2.3 Quy trình nghiệp vụ



Hình 2.11: Biểu đồ ca sử dụng phân rã “CRUD bình luận”

Hình 2.11 thể hiện quy trình nghiệp vụ, có các bước như sau:

- **Bước 1:** Bác sĩ cập nhật lịch làm việc của mình vào hệ thống.
- **Bước 2:** Bệnh nhân tìm kiếm bác sĩ muốn khám dựa theo các tiêu chí như bệnh viện, chuyên khoa,...
- **Bước 3:** Hệ thống hiển thị danh sách bác sĩ.

- **Bước 4:** Bệnh nhân chọn bác sĩ muốn đặt lịch khám.
- **Bước 5:** Hệ thống hiển thị thông tin bác sĩ kèm theo lịch làm việc trống của bác sĩ.
- **Bước 6:** Bệnh nhân chọn thời gian muốn đặt khám và gửi form thông tin đặt lịch đã điền.
- **Bước 7:** Bệnh nhân tới email trong form đăng ký khám bệnh để xác thực.
 - Nếu như bệnh nhân không xác nhận thì quy trình kết thúc và lịch hẹn khám sẽ không được lên ở phía bác sĩ.
- **Bước 8:** Bác sĩ tới danh sách lịch hẹn khám đang chờ xác nhận để xác nhận lịch hẹn.
 - Nếu bác sĩ chọn từ chối thì sẽ có mail gửi đến bệnh nhân để thông báo và lịch hẹn bị hủy.
- **Bước 9:** Bệnh nhân đến khám và ra về.
- **Bước 10:** Bác sĩ gửi kết quả khám bệnh và đơn thuốc.

2.4 Đặc tả chức năng

Tôi lựa chọn trình bày một số ca sử dụng trong hệ thống là:

2.4.1 Đặc tả ca sử dụng “Đăng ký tài khoản bác sĩ”

| | | | |
|------------------------|----------------------------------------------------------------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------|
| Mã use case | UC001 | Tên use case | Đăng ký tài khoản bác sĩ |
| Tác nhân | Bác sĩ | | |
| Mục đích ca sử dụng | Có tài khoản sử dụng cho bác sĩ để bác sĩ cập nhật lịch làm việc lên hệ thống và quản lý lịch hẹn. | | |
| Tiền điều kiện | Bác sĩ đã có url tới trang đăng ký | | |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động |
| | 1 | Bác sĩ | Bác sĩ vào url được cấp và điền đầy đủ thông tin vào form, đặc biệt cần chứng chỉ hành nghề; nộp form và đợi kết quả. |
| | 2 | Hệ thống | Lưu thông tin tài khoản bác sĩ và cập nhật tới cho admin để xác thực. |
| Luồng sự kiện thay thế | | | |
| Hậu điều kiện | | | |

Bảng 2.1: Đặc tả ca sử dụng “Đăng ký tài khoản bác sĩ”

2.4.2 Đặc tả ca sử dụng “Xác nhận tài khoản bác sĩ”

| | | | |
|------------------------|------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------------|
| Mã use case | UC002 | Tên use case | Xác nhận tài khoản bác sĩ |
| Tác nhân | Quản trị viên | | |
| Mục đích ca sử dụng | Xác thực thông tin bác sĩ, từ đó đồng ý hoặc từ chối với tài khoản đó. | | |
| Tiền điều kiện | Khách hàng đã lọc khu nghỉ dưỡng. | | |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động |
| | 1 | Quản trị viên | Vào trang danh sách tài khoản bác sĩ cần xác nhận |
| | 2 | Hệ thống | Hệ thống truy vấn thông tin chi danh sách tài khoản bác sĩ chưa xác nhận và hiển thị ra màn hình. |
| | 3 | Quản trị viên | Thực hiện việc xác thực các thông tin mà bác sĩ đã đăng ký và xác nhận đồng ý. |
| | 4 | Hệ thống | Cập nhật lại trạng thái tài khoản và gửi thư kết quả tới email của bác sĩ. |
| Luồng sự kiện thay thế | STT | Thực hiện bởi | Hành động |
| | 3a | Quản trị viên | Thực hiện việc xác thực các thông tin mà bác sĩ đã đăng ký và xác nhận từ chối. |
| | 4a | Hệ thống | Xóa tài khoản bác sĩ và gửi thư kết quả tới cho bác sĩ qua email. |
| Hậu điều kiện | | | |

Bảng 2.2: Đặc tả ca sử dụng “Xác nhận tài khoản bác sĩ”

2.4.3 Đặc tả ca sử dụng “Tìm kiếm bác sĩ”

| | | | |
|------------------------|------------------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mã use case | UC003 | Tên use case | Tìm kiếm bác sĩ |
| Tác nhân | Khách/Bệnh nhân | | |
| Mục đích ca sử dụng | Bệnh nhân tìm được bác sĩ phù hợp để đặt khám. | | |
| Tiền điều kiện | | | |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động |
| | 1 | Khách/Bệnh nhân | Khách/Bệnh nhân vào trang chủ hệ thống |
| | 2 | Hệ thống | Hiển thị ra các thông tin như thanh navbar, banner, danh sách bác sĩ, danh sách bệnh viện, danh sách chuyên khoa, danh sách bài viết,.. |
| | 3 | Khách/Bệnh nhân | Khách/Bệnh nhân có thể tìm kiếm bác sĩ thông qua các hạng mục trên trang chủ như danh sách bác sĩ nổi bật, tìm theo chuyên khoa,... |
| | 4 | Hệ thống | Hiển thị thông tin chi tiết của của bác sĩ bao gồm: tên, học vị, chuyên khoa, nơi làm việc, lịch làm việc còn trống của bác sĩ, giá khám, các bệnh khám và điều trị, quá trình công tác, quá trình đào tạo và các thông tin thêm khác. |
| Luồng sự kiện thay thế | STT | Thực hiện bởi | Hành động |
| | 3a-1 | Khách/Bệnh nhân | Tới trang tìm kiếm và tìm kiếm bác sĩ theo từ khóa. |
| | 3a-2 | Hệ thống | Hiển thị danh sách bác sĩ hợp lệ. |
| | 3a-3 | Khách/Bệnh nhân | Chọn bác sĩ |
| Hậu điều kiện | | | |

Bảng 2.3: Đặc tả ca sử dụng “Tìm kiếm bác sĩ”

2.4.4 Đặc tả ca sử dụng “Cập nhật lịch làm việc”

| | | | |
|------------------------|---------------------------------------------------------------------------------------------|---------------|------------------------------------------------------|
| Mã use case | UC004 | Tên use case | Cập nhật lịch làm việc |
| Tác nhân | Bác sĩ | | |
| Mục đích ca sử dụng | Bác sĩ cập nhật lịch làm việc của mình lên hệ thống để bệnh nhân có thể xem và đặt lịch hẹn | | |
| Tiền điều kiện | Bác sĩ đã đăng nhập vào hệ thống | | |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động |
| | 1 | Bác sĩ | Chọn ngày muốn cập nhật lịch làm việc. |
| | 2 | Hệ thống | Hiển thị các thời điểm bác sĩ đăng ký lịch làm việc. |
| | 3 | Bác sĩ | Thay đổi lịch làm việc của bác sĩ |
| | 3 | Hệ thống | Cập nhật lịch làm việc của bác sĩ vào database. |
| Luồng sự kiện thay thế | | | |
| Hậu điều kiện | Hệ thống lưu trữ đơn đặt phòng mới vào cơ sở dữ liệu | | |

Bảng 2.4: Đặc tả ca sử dụng “Cập nhật đơn đặt phòng”

2.5 Yêu cầu phi chức năng

Ngoài các yêu cầu về chức năng thì hệ thống cũng cần đáp ứng các yêu cầu phi chức năng hiệu năng, độ tin cậy, tính dễ dùng, tính dễ bảo trì, hoặc các yêu cầu về mặt kỹ thuật như về cơ sở dữ liệu, công nghệ sử dụng.

Hiệu năng:

- Hệ thống phải phản hồi các yêu cầu của người dùng trong vòng 2 giây hoặc ít hơn.
- Các thao tác thường dùng như tìm kiếm, đặt phòng cần được thực hiện trong vòng 1 giây hoặc ít hơn.

Tính dễ dùng:

- Giao diện cần được thiết kế đơn giản để dễ sử dụng và thân thiện với người dùng.
- Các chức năng và tính năng cần được thiết kế một cách logic và trực quan để người dùng dễ sử dụng, bao gồm cả người mới sử dụng.

Độ tin cậy:

- Hệ thống cần hoạt động 24/7 với thời gian chết tối thiểu.
- Bảo vệ thông tin cá nhân của khách hàng và nhân viên là ưu tiên hàng đầu.
- Hệ thống cần tuân thủ các quy định về bảo mật dữ liệu.

Cơ sở dữ liệu:

- Cần sử dụng các biện pháp bảo mật phù hợp để bảo vệ cơ sở dữ liệu khỏi truy cập trái phép, thay đổi hoặc xóa bỏ.

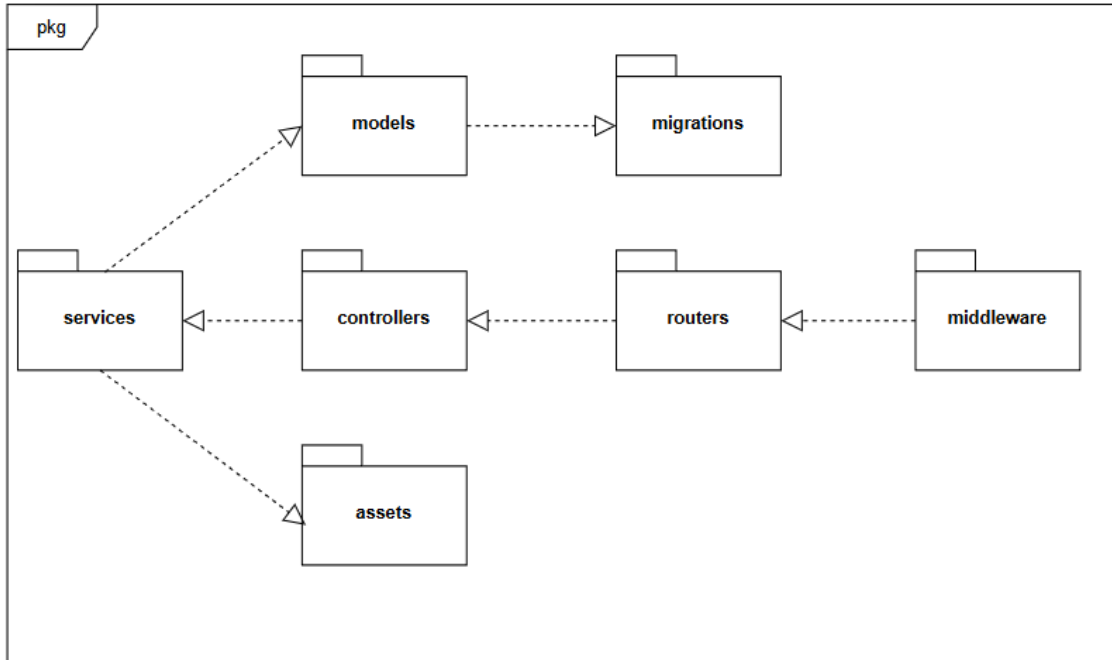
Công nghệ sử dụng:

- Sử dụng các công nghệ đảm bảo tính tương thích, có khả năng mở rộng và cập nhật dễ dàng khi hệ thống có sự thay đổi.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Thiết kế kiến trúc

3.1.1 Sơ đồ gói



Hình 3.1: Biểu đồ phụ thuộc gói phần máy chủ Nodejs

Nhìn vào hình 3.1, các thành phần của biểu đồ phụ thuộc gói có ý nghĩa như sau:

– Models:

- * Chứa các mô hình dữ liệu (models), đại diện cho các thực thể (entities) trong ứng dụng.
- * Định nghĩa cấu trúc dữ liệu, các thuộc tính và phương thức xử lý liên quan đến dữ liệu.
- * Ví dụ: Patient, Doctor, Appointment.

* **Mối quan hệ:**

- **Migrations:** Models liên hệ với migrations để tạo hoặc thay đổi cấu trúc cơ sở dữ liệu.
- **Services:** Được sử dụng trong services để đọc, ghi, và xử lý dữ liệu.

– Migrations:

- * Quản lý các thay đổi trong cơ sở dữ liệu thông qua các tệp migration.

- * Đồng bộ cơ sở dữ liệu với mã nguồn.

- * **Mối quan hệ:**

- **Models:** Dựa trên thông tin từ models để tạo cấu trúc cơ sở dữ liệu.

- **Services:**

- * Xử lý các logic nghiệp vụ chính của ứng dụng.

- * Tính toán, xử lý dữ liệu phức tạp, hoặc tương tác với tài nguyên bên ngoài.

- * Ví dụ: Logic kiểm tra lịch bác sĩ trong hệ thống đặt lịch khám.

- * **Mối quan hệ:**

- **Models:** Tương tác với cơ sở dữ liệu thông qua models.

- **Controllers:** Gửi dữ liệu đã xử lý cho controllers.

- **Assets:** Liên kết tài nguyên tĩnh như email templates hoặc báo cáo.

- **Controllers:**

- * Là cầu nối giữa người dùng và logic nghiệp vụ của ứng dụng.

- * Nhận yêu cầu từ người dùng, gọi services và trả kết quả.

- * **Mối quan hệ:**

- **Routers:** Nhận yêu cầu qua routers.

- **Services:** Xử lý logic nghiệp vụ thông qua services.

- **Routers:**

- * Quản lý định tuyến trong ứng dụng.

- * Cấu hình các endpoint API như `/appointments` hoặc `/patients`.

- * **Mối quan hệ:**

- **Controllers:** Điều hướng yêu cầu đến đúng controller.

- **Middlewares:** Xử lý yêu cầu trước khi chuyển đến controller.

- **Middlewares:**

- * Cung cấp chức năng trung gian như xác thực, ghi log hoặc kiểm tra dữ liệu.

- * Đảm bảo các yêu cầu hợp lệ trước khi vào hệ thống.

- * **Mối quan hệ:**

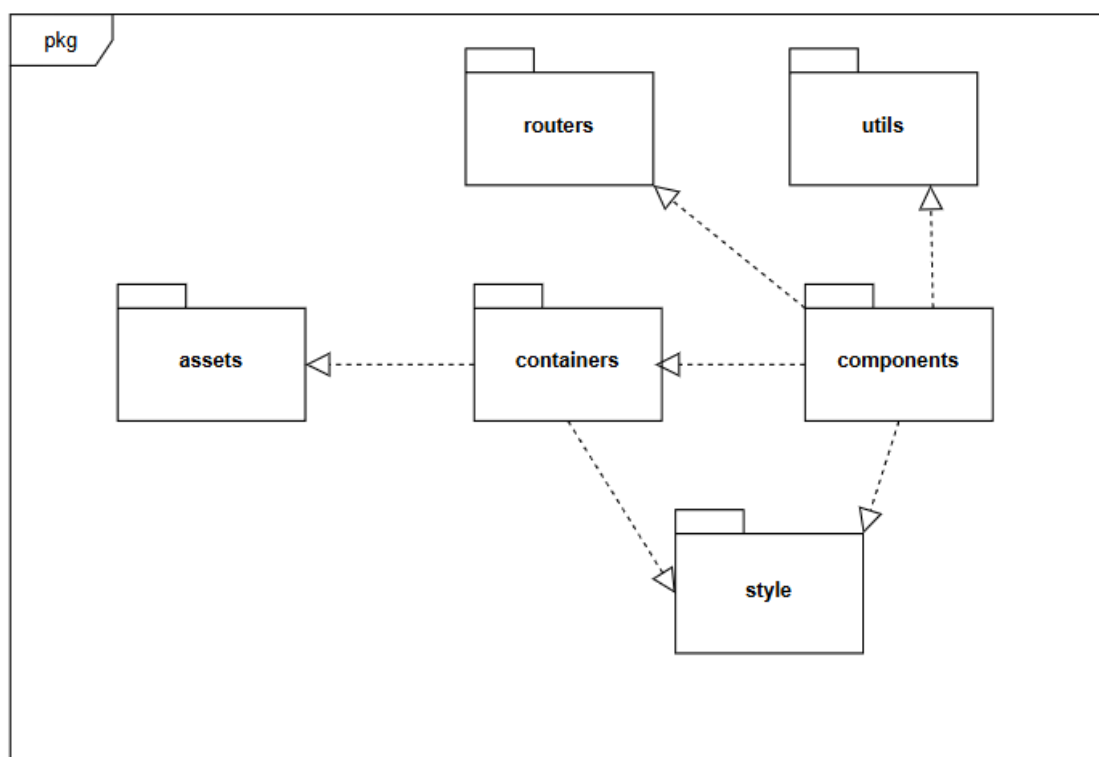
- **Routers:** Xử lý yêu cầu trong quá trình định tuyến.
- **Controllers:** Chuyển yêu cầu hợp lệ đến controllers.

– **Assets:**

- * Lưu trữ tài nguyên tĩnh như hình ảnh, CSS, hoặc file JavaScript.
- * Sử dụng trong giao diện hoặc các dịch vụ khác.

* **Mối quan hệ:**

- **Services:** Tải hình ảnh hoặc email template trong logic nghiệp vụ.



Hình 3.2: Biểu đồ phụ thuộc gói phần giao diện

Hình 3.2 biểu thị các thành phần của biểu đồ phụ thuộc gói phần giao diện:

- **assets:** Lưu trữ các tài nguyên tĩnh như hình ảnh, biểu tượng, hoặc tệp CSS/JS liên quan.
- **routers:** Định nghĩa các tuyến (route) trong ứng dụng, đảm bảo điều hướng giữa các trang hoặc thành phần.
- **containers:** Chứa các thành phần chính trong ứng dụng, thường là các module lớn đảm nhận cả logic và giao diện.
- **components:** Chứa các thành phần giao diện nhỏ, tái sử dụng được như button, input, hoặc card.

- **utils**: Bao gồm các module hỗ trợ, hàm tiện ích chung như xử lý dữ liệu, định dạng ngày giờ, hoặc các thư viện tùy chỉnh.
- **style**: Quản lý các tệp định dạng giao diện như CSS, SCSS, hoặc styled-components, đảm bảo giao diện đồng nhất.

3.1.2 Mỗi quan hệ giữa các thư mục

- **assets**:
 - Được sử dụng bởi **containers** và **components** để hiển thị tài nguyên.
- **routers**:
 - Điều hướng yêu cầu tới các **containers**.
- **containers**:
 - Kết nối với **components** để xây dựng giao diện phức tạp.
 - Có thể sử dụng trực tiếp các tệp từ **style**.
- **components**:
 - Sử dụng các tiện ích từ **utils** để xử lý logic nhỏ.
 - Liên kết với **style** để áp dụng kiểu dáng.
- **utils**:
 - Cung cấp các hàm tiện ích cho tất cả các thành phần khác như **components**, **containers**, hoặc **routers**.
- **style**:
 - Được sử dụng bởi **containers** và **components** để thiết kế giao diện.

3.2.1 Bảng "Users"

Bảng Users lưu trữ thông tin tài khoản người dùng hệ thống.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|--------------------|
| id | NOT NULL | bigint | increment | ID người dùng |
| email | NOT NULL | String[255] | unique | Email người dùng |
| password | NOT NULL | String[255] | none | Mật khẩu |
| username | NOT NULL | String[255] | none | Tên người dùng |
| birthdate | NULL | date | none | Ngày sinh |
| address | NULL | String[255] | none | Địa chỉ |
| phonenummer | NULL | String[255] | none | Số điện thoại |
| image | NULL | String[255] | none | Ảnh đại diện |
| gender | NULL | String[255] | none | Giới tính |
| roleId | NULL | String[255] | none | Vai trò |
| refreshToken | NULL | String[255] | none | Refresh token |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.2 Bảng "DoctorInfos"

Bảng DoctorInfos lưu trữ thông tin về bác sĩ.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|-----------------|-----------|--------------|-----------|-------------------------------------------------------|
| id | NOT NULL | bigint | increment | ID thông tin bác sĩ |
| doctorId | NOT NULL | bigint | unique | ID bác sĩ, khóa ngoài tham chiếu tới id của bảng user |
| priceId | NULL | varchar(255) | none | ID giá khám |
| positionId | NULL | varchar(255) | none | ID chức danh |
| paymentId | NULL | varchar(255) | none | ID phương thức thanh toán |
| clinicId | NULL | bigint | none | ID phòng khám |
| profile | NULL | varchar(255) | none | Hồ sơ bác sĩ |
| certificate | NULL | varchar(255) | none | Chứng chỉ hành nghề |
| descriptionHTML | NULL | text | none | Mô tả dạng HTML |
| descriptionText | NULL | text | none | Mô tả dạng văn bản |
| statusId | NULL | varchar(255) | none | ID trạng thái |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.3 Bảng "Specialties"

Bảng Specialties lưu trữ thông tin về các chuyên khoa.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|-----------------|-----------|--------------|-----------|-------------------------------|
| id | NOT NULL | bigint | increment | ID chuyên khoa |
| name | NOT NULL | String[255] | none | Tên chuyên khoa |
| image | NULL | String[255] | none | Đường dẫn tới ảnh chuyên khoa |
| descriptionHTML | NULL | text | none | Mô tả dạng HTML |
| descriptionText | NULL | text | none | Mô tả dạng văn bản |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.4 Bảng "DoctorSpecialties"

Bảng DoctorSpecialties là bảng liên kết giữa hai bảng DoctorInfos và Specialties với mối quan hệ nhiều nhiều (Bác sĩ và chuyên khoa).

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|----------------------|
| id | NOT NULL | bigint | increment | ID chuyên môn bác sĩ |
| doctorId | NOT NULL | bigint | none | ID bác sĩ |
| specialtyId | NOT NULL | bigint | none | ID chuyên khoa |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.5 Bảng "Clinics"

Bảng Clinics lưu trữ thông tin về cơ sở y tế.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|-----------------|-----------|--------------|-----------|-----------------------|
| id | NOT NULL | bigint | increment | ID cơ sở y tế |
| name | NOT NULL | varchar(255) | none | Tên cơ sở y tế |
| address | NOT NULL | varchar(255) | none | Địa chỉ |
| phonenumber | NULL | varchar(12) | none | Số điện thoại |
| email | NULL | varchar(50) | none | Email |
| website | NULL | varchar(50) | none | Website |
| districtId | NULL | varchar(10) | none | ID quận, huyện |
| provinceId | NULL | varchar(10) | none | ID tỉnh, thành phố |
| image | NULL | varchar(255) | none | Đường dẫn tới logo |
| background | NULL | varchar(255) | none | Đường dẫn tới ảnh nền |
| descriptionHTML | NULL | text | none | Mô tả dạng HTML |
| descriptionText | NULL | text | none | Mô tả dạng văn bản |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.6 Bảng "Notebooks"

Bảng Notebooks lưu trữ các bài viết của trang.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|-----------------------|
| id | NOT NULL | bigint | increment | ID số tay |
| title | NOT NULL | varchar(255) | none | Tiêu đề |
| contentHTML | NULL | text | none | Nội dung dạng HTML |
| contentText | NULL | text | none | Nội dung dạng văn bản |
| image | NULL | varchar(255) | none | Đường dẫn tới ảnh |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.7 Bảng "NotebookAuthors"

Bảng NotebookAuthors là bảng liên kết giữa hai bảng Notebooks với Users với mối quan hệ nhiều nhiều (Bài viết và tác giả).

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|--------------------|
| id | NOT NULL | bigint | increment | Mã định danh |
| authorId | NOT NULL | bigint | none | ID tác giả |
| notebookId | NOT NULL | bigint | none | ID số tay |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.8 Bảng "NotebookCensors"

Bảng NotebookCensors là bảng liên kết giữa hai bảng Notebooks với Users với mối quan hệ nhiều nhiều (Bài viết và kiểm duyệt viên).

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|--------------------|
| id | NOT NULL | bigint | increment | Mã định danh |
| censorId | NOT NULL | bigint | none | ID kiểm duyệt viên |
| notebookId | NOT NULL | bigint | none | ID sổ tay |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.9 Bảng "Schedules"

Bảng Schedules lưu trữ lịch làm việc của bác sĩ.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|-----------------------|
| id | NOT NULL | bigint | increment | ID lịch trình |
| doctorId | NOT NULL | bigint | none | ID bác sĩ |
| date | NOT NULL | date | none | Ngày làm việc |
| timeId | NULL | varchar(10) | none | ID thời gian |
| currentPatient | NOT NULL | int | none | Số bệnh nhân hiện tại |
| maxPatient | NOT NULL | int | none | Số bệnh nhân tối đa |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.10 Bảng "Bookings"

Bảng Bookings lưu trữ thông tin các đơn đặt lịch hẹn khám.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|----------------------------------------------------------------------------------------------|-----------|--------------------------|
| id | NOT NULL | bigint | increment | ID đặt lịch |
| patientId | NOT NULL | bigint | none | ID bệnh nhân |
| doctorId | NOT NULL | bigint | none | ID bác sĩ |
| address | NULL | varchar(255) | none | Địa chỉ |
| phone | NULL | varchar(255) | none | Số điện thoại |
| relativeName | NULL | varchar(255) | none | Tên người thân |
| relativePhone | NULL | varchar(255) | none | Số điện thoại người thân |
| date | NOT NULL | date | none | Ngày đặt lịch |
| timeId | NULL | varchar(10) | none | ID thời gian |
| reason | NULL | text | none | Lý do đặt lịch |
| result | NULL | varchar(255) | none | Kết quả khám |
| status | NULL | varchar(20) | none | Trạng thái |
| createdAt | NOT NULL | datetime  | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

3.2.11 Bảng "Comments"

Bảng Comments lưu trữ các bình luận của người dùng với bác sĩ hoặc cơ sở y tế.

| Trường dữ liệu | Ràng buộc | Kiểu dữ liệu | Mặc định | Mô tả |
|----------------|-----------|--------------|-----------|--------------------|
| id | NOT NULL | bigint | increment | ID bình luận |
| patientId | NOT NULL | bigint | none | ID bệnh nhân |
| doctorId | NULL | bigint | none | ID bác sĩ |
| clinicId | NULL | bigint | none | ID phòng khám |
| content | NOT NULL | text | none | Nội dung bình luận |
| parentId | NULL | bigint | none | ID bình luận gốc |
| createdAt | NOT NULL | datetime | now() | Thời điểm tạo |
| updatedAt | NULL | datetime | none | Thời điểm cập nhật |

CHƯƠNG 4. CÀI ĐẶT TRIỂN KHAI VÀ KẾT QUẢ

4.1 Công nghệ sử dụng

4.1.1 Công nghệ sử dụng cho Back-end

Node.js

Node.js là một nền tảng mã nguồn mở được xây dựng trên JavaScript và chạy trên nền tảng Chrome V8 Engine. Node.js được phát triển bởi Ryan Dahl vào năm 2009 với mục tiêu đưa JavaScript ra khỏi môi trường trình duyệt và ứng dụng trong các ứng dụng server-side. Node.js cho phép lập trình viên sử dụng JavaScript để:

- Xây dựng các ứng dụng server-side.
- Xử lý các yêu cầu HTTP.
- Cập nhật dữ liệu và thực hiện các tác vụ liên quan đến server-side.

Một số công nghệ tương tự Node.js bao gồm Deno, ASP.NET, và Ruby on Rails. Tuy nhiên, tôi lựa chọn Node.js vì nó rất phổ biến trong cộng đồng phát triển web, hỗ trợ tốt cho nhiều thư viện và framework, đảm bảo khả năng mở rộng và phát triển dễ dàng.

Express.js

Express.js là một framework phổ biến của Node.js, được sử dụng để xây dựng các ứng dụng web và API. Framework này giúp quá trình phát triển trở nên đơn giản và nhanh chóng nhờ cung cấp nhiều tính năng hữu ích như:

- **Middleware:** Express.js tích hợp sẵn nhiều middleware, giúp thêm các tính năng bổ sung như xác thực, xử lý định tuyến, bảo mật, v.v.
- **Các phương pháp xử lý yêu cầu:** Express.js hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE để lấy, tạo mới, cập nhật và xóa dữ liệu.
- **Hỗ trợ module:** Cung cấp nhiều thư viện hỗ trợ giúp xây dựng các ứng dụng web và API mạnh mẽ.

Tôi chọn Express.js vì nó là một framework đơn giản, dễ sử dụng, và giúp xây dựng các ứng dụng web hoặc API hiệu quả. Middleware và các thư viện hỗ trợ mạnh mẽ của Express.js cũng là một lợi thế lớn.

4.1.2 Công nghệ sử dụng cho Front-end

React.js

React.js là một thư viện JavaScript mã nguồn mở do Facebook phát triển, được

thiết kế để xây dựng giao diện người dùng (UI) linh hoạt và tương tác cao. React.js cung cấp các tính năng nổi bật như:

- **Component tái sử dụng:** Giúp việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn.
- **Quản lý trạng thái linh hoạt:** React.js cung cấp cách tiếp cận hiện đại và hiệu quả để quản lý trạng thái của ứng dụng.
- **Tối ưu hóa hiệu suất:** React.js sử dụng Virtual DOM để cải thiện hiệu suất khi cập nhật giao diện.

Trong đồ án này, React.js được sử dụng để xây dựng giao diện người dùng cho hệ thống. Các component được thiết kế để hỗ trợ tìm kiếm và tư vấn đặt phòng một cách linh hoạt. Lựa chọn React.js bởi vì nó là một thư viện phổ biến, mạnh mẽ, và dễ tích hợp trong các ứng dụng web hiện đại.

4.1.3 Công nghệ sử dụng trong Database

MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở được sử dụng rộng rãi trong các ứng dụng web. Các tính năng nổi bật của MySQL bao gồm:

- **Tính phổ biến:** MySQL được hỗ trợ rộng rãi bởi cộng đồng và có tài liệu phong phú.
- **Khả năng mở rộng:** Phù hợp với các hệ thống từ quy mô nhỏ đến lớn.
- **Hiệu suất cao:** Hỗ trợ truy vấn và lưu trữ dữ liệu hiệu quả.

Trong đồ án này, tôi đã lựa chọn MySQL làm hệ quản trị cơ sở dữ liệu vì nó đáp ứng tốt nhu cầu lưu trữ và truy vấn dữ liệu. Ngoài ra, MySQL có khả năng tích hợp tốt với các công nghệ khác như Node.js và React.js, giúp hệ thống hoạt động ổn định và hiệu quả.

Docker

Docker là một nền tảng mã nguồn mở cho phép xây dựng, triển khai và quản lý các ứng dụng bên trong các container. Container hóa cơ sở dữ liệu MySQL thông qua Docker mang lại các lợi ích như:

- **Tính di động:** Cơ sở dữ liệu có thể được triển khai trên nhiều môi trường khác nhau mà không cần cấu hình lại.
- **Tính nhất quán:** Docker đảm bảo rằng môi trường cơ sở dữ liệu luôn đồng nhất giữa các môi trường phát triển, kiểm thử và triển khai.
- **Dễ quản lý:** Docker hỗ trợ các công cụ giám sát và quản lý container hiệu quả.

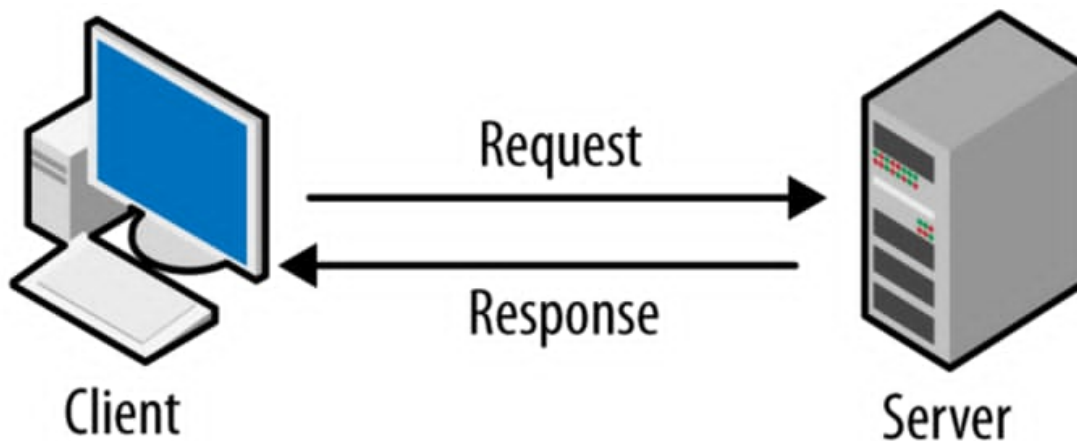
Trong đồ án này, tôi sử dụng Docker để container hóa cơ sở dữ liệu MySQL, đảm bảo tính nhất quán và dễ dàng quản lý khi triển khai trên các môi trường khác nhau.

4.2 Cài đặt triển khai

Từ các khảo sát, em sử dụng lựa chọn mô hình kiến trúc client - server để cài đặt hệ thống. Mô tả kiến trúc Client - Server :

- Mô hình client server mô hình mạng máy tính gồm có 2 thành phần chính là client và server (tức là máy khách – máy chủ). Client sẽ là bên yêu cầu dịch vụ cài đặt cũng như lưu trữ tài nguyên từ phía server
- Khi máy client gửi yêu cầu dữ liệu đến server thông qua Internet, server sẽ chấp nhận quy trình được yêu cầu. Sau đó gửi các gói dữ liệu được yêu cầu trở lại client. Client không chia sẻ bất kỳ tài nguyên nào của họ.

Đây là một cấu trúc ứng dụng phân tán, nó phân vùng các nhiệm vụ hay workload giữa các nhà cung cấp tài nguyên hoặc dịch vụ, gọi là server, và người yêu cầu dịch vụ (Client).



Hình 4.1: Mô hình client - server

Client

Client là một máy tính (Host), tức là có khả năng nhận thông tin hoặc sử dụng một dịch vụ cụ thể từ các nhà cung cấp dịch vụ (Server).

Server

Trong lĩnh vực công nghệ thì Server là một máy tính từ xa. Nó cung cấp các thông tin (dữ liệu) hoặc quyền truy cập vào các dịch vụ cụ thể.

Ưu điểm:

- Tập trung: Dễ dàng quản lý và kiểm soát tài nguyên, dữ liệu từ một vị trí duy nhất.
- Bảo mật: Kiểm soát truy cập chặt chẽ, dễ dàng sao lưu và phục hồi dữ liệu.
- Khả năng mở rộng: Linh hoạt thêm tài nguyên (client, server) khi cần thiết.

- Khả năng truy cập: Cho phép truy cập từ nhiều nền tảng và vị trí khác nhau.

Nhược điểm:

- Tắc nghẽn lưu lượng: Hiệu suất giảm khi có quá nhiều yêu cầu từ client.
- Độ bền: Sự cố server ảnh hưởng đến toàn bộ hệ thống.
- Chi phí: Thiết lập và bảo trì server tốn kém.
- Bảo trì: Đòi hỏi chuyên môn quản lý và bảo trì server.
- Tài nguyên: Giới hạn khả năng sử dụng tài nguyên trên server.

4.3 Kết quả đạt được

4.3.1 Chức năng "Đăng ký tài khoản bác sĩ"

Hình 4.2 mô tả form đăng ký tài khoản bác sĩ, bao gồm các trường thông tin quan trọng phục vụ cho quá trình xác thực thông tin bác sĩ, tạo độ tin cậy,...

Đăng ký tài khoản Bác sĩ

Email:

Mật khẩu

Nhập lại mật khẩu

Nhập email

Nhập mật khẩu

Nhập mật khẩu

Họ và tên

Số điện thoại

Nhập họ và tên

Nhập số điện thoại

Địa chỉ

Nhập địa chỉ

Ngày sinh

Giới tính

Hình ảnh

30/12/2024

Chọn giới tính

Choose File

Nơi làm việc

Chuyên khoa

Chọn cơ sở y tế

Chọn chuyên khoa

Chức danh

Giá khám

Hình thức thanh toán

Chọn chức danh

Chọn giá khám

Chọn hình thức thanh

Profile

Chứng chỉ hành nghề

Choose File

No file chosen

Choose File

No file chosen

Đăng ký

Hình 4.2: Giao diện thể hiện các trường thông tin bác sĩ cần điền để đăng ký

4.3.2 Chức năng "Cập nhật lịch làm việc"

Hình 4.3 mô tả giao diện cập nhật lịch làm việc của bác sĩ bao gồm chọn ngày và khung giờ làm việc; từ đó để thể hiện ra giao diện chi tiết bác sĩ cho người dùng lựa chọn thời điểm phù hợp.

The screenshot shows a web interface for a doctor's work schedule management. On the left, there's a sidebar with a logo 'MEDICAL BOOKING' and two buttons: 'Lịch làm việc' (Work Schedule) and 'Lịch hẹn' (Appointment). The main content area is titled 'Xin chào Bác sĩ' (Hello Doctor) and 'Quản lý lịch làm việc' (Manage Work Schedule). It features a date picker set to '30/12/2024' and a grid of time slots for selection. The time slots are arranged in two rows: 8:00 - 9:00, 9:00 - 10:00, 10:00 - 11:00, 11:00 - 12:00, 13:00 - 14:00 in the first row, and 14:00 - 15:00, 15:00 - 16:00, 16:00 - 17:00 in the second row. A green 'Cập nhật' (Update) button is located at the top right of the main area.

Hình 4.3: Giao diện quản lý lịch làm việc

4.3.3 Chức năng "Quản lý lịch hẹn khám bệnh"

Hình 4.4 mô tả giao diện quản lý lịch hẹn của bác sĩ, bao gồm các lịch hẹn được filter theo status và ngày hẹn.


The screenshot shows a web interface for managing appointments. On the left, there's a sidebar with a logo 'MEDICAL BOOKING' and two buttons: 'Lịch làm việc' (Work Schedule) and 'Lịch hẹn' (Appointment). The main content area is titled 'Xin chào Bác sĩ' (Hello Doctor) and 'Quản lý lịch hẹn' (Manage Appointment Schedule). It includes a date picker set to '30/12/2024' and a dropdown menu set to 'Đã qua' (Over). Below this is a table with the following data:

| STT | Ngày hẹn | Thời gian | Khách hàng | Lý do khám | Hành động |
|-----|------------|--------------|---------------|------------------------------------------------------------------------------------------------------------------|--------------|
| 1 | 2024-12-21 | 9:00 - 10:00 | Phạm Xuân Duy | Cảm thấy đau đầu thường xuyên kèm theo chóng mặt, mắt tập trung và cảm giác buồn nôn kéo dài trong vài ngày qua. | Xem chi tiết |
| 2 | 2024-12-22 | 8:00 - 9:00 | Phạm Xuân Duy | Sốt kéo dài. | Xem chi tiết |

Hình 4.4: Giao diện quản lý lịch hẹn khám bệnh

4.3.4 Chức năng "Hiển thị thông tin chi tiết bác sĩ"


Hình 4.5 mô tả giao diện thông tin chi tiết bác sĩ, bao gồm các trường thông tin quan trọng về bác sĩ như tên, học vị, nơi làm việc, quá trình đào tạo, quá trình làm việc,... Ngoài ra tại đây bệnh nhân có thể xem được lịch làm việc trống của bác sĩ để chọn thời điểm đặt khám phù hợp.


MEDICAL BOOKING


Chuyên khoa
Tim bác sĩ theo chuyên khoa

Cơ sở y tế
Chọn bệnh viện phòng khám

Bác sĩ
Chọn bác sĩ giới


Lịch hẹn

[/ Bác sĩ](#)



Phó Giáo sư, Tiến sĩ, Bác sĩ Nguyễn Thị Hoài An
Phó Giáo sư, Tiến sĩ chuyên ngành Tai Mũi Họng
 Nguyên Trưởng khoa Tai Mũi Họng trẻ em, Bệnh viện Tai Mũi Họng Trung ương
 Trên 25 năm công tác tại Bệnh viện Tai Mũi Họng Trung ương
Hà Nội

Hôm nay - 30/12

Lịch khám
 Không có lịch làm việc

ĐỊA CHỈ KHÁM
 Hệ thống Y tế Thu Cúc TCI
 Cơ sở Thụy Khuê: 286 Thụy Khuê, Tây Hồ, Hà Nội
 GIÁ KHÁM: 400000 VND
 HÌNH THỨC THANH TOÁN: Tất cả

Phó Giáo sư, Tiến sĩ, Bác sĩ Nguyễn Thị Hoài An

- Phó Giáo sư, Tiến sĩ chuyên ngành Tai Mũi Họng
- Nguyên Trưởng khoa Tai Mũi Họng trẻ em, Bệnh viện Tai Mũi Họng Trung ương
- Trên 25 năm công tác tại Bệnh viện Tai Mũi Họng Trung ương
- Bác sĩ Nội trú chuyên ngành Tai Mũi Họng
- Bác sĩ đã từng tu nghiệp Cộng hòa Pháp về Tai Mũi Họng

Khám & điều trị

- Chuyên khám và điều trị các bệnh lý Tai Mũi Họng người lớn
- Chuyên khám và điều trị các bệnh lý Tai Mũi Họng trẻ em
- Nội soi Tai Mũi Họng
- Thực hiện các qui trình kỹ thuật Tai Mũi Họng

Các bệnh về tai

- Ù tai, nghe kém, điếc đột ngột
- Chảy mủ tai, viêm tai giữa cấp, mạn
- Vá màng nhĩ nội soi
- Phát hiện sớm và điều trị tốt bệnh viêm tai giữa màng nhĩ đóng kín, không chảy mủ ra ngoài


Các bệnh mũi xoang




- Viêm mũi xoang dị ứng, viêm mũi vãn mạch
- Viêm mũi ngạt tắc mũi mạn tính
- Viêm đa xoang mạn lâu ngày khó khỏi, polyp mũi xoang
- Nấm mũi xoang
- Đau đầu mạn tính do mũi xoang...

Các bệnh về họng thanh quản

- Ở trẻ em viêm VA, viêm mũi họng mạn tính
- Các biến chứng của viêm VA (như viêm tai thanh dịch, viêm tai giữa cấp, viêm thanh khí phế quản...)
- Viêm amidan cấp, mạn
- Nào VA
- Cắt Amidan

Bình luận


MEDICAL BOOKING

Hình 4.5: Thông tin chi tiết bác sĩ

4.3.5 Chức năng "Đặt lịch khám bệnh"

Hình 4.6 mô tả form thông tin đặt lịch khám bệnh, bao gồm các trường thông tin cơ bản của bệnh nhân; ngoài đặt lịch cho bản thân, người dùng còn có thể đặt lịch cho người thân.

The screenshot displays the 'ĐẶT LỊCH KHÁM' (Book Appointment) screen of the 'MEDICAL BOOKING' application. The header bar is light green and contains the app logo, navigation links for 'Chuyên khoa' (Specialty), 'Cơ sở y tế' (Medical Facility), and 'Bác sĩ' (Doctor), a search bar, and a 'Lịch hẹn' (Appointment) icon. The main content area has a light gray background. It features a circular profile picture of a doctor, followed by the text 'ĐẶT LỊCH KHÁM', 'Phó giáo sư, Bác sĩ Nguyễn Thị Hoài An', '13:00 - 14:00 - Thứ ba - 31/12', 'Hệ thống Y tế Thu Cúc TCI', and 'Cơ sở Thụy Khuê: 286 Thụy Khuê, Tây Hồ, Hà Nội'. Below this, a yellow box shows 'GIÁ KHÁM: 400000 VND'. There are two radio buttons for 'Đặt lịch cho bản thân' (selected) and 'Đặt lịch cho người thân'. A section titled 'Thông tin bệnh nhân' (Patient Information) includes input fields for 'Họ và tên bệnh nhân (Bắt buộc)' (required), gender selection ('Nam' selected, 'Nữ'), 'Số điện thoại (Bắt buộc)' (required), 'Địa chỉ email', 'Địa chỉ', and a text area for 'Lý do khám'. Below the form, a link 'Hình thức thanh toán' (Payment Method) indicates 'Thanh toán tại cơ sở y tế' (Pay at the medical facility). A large yellow button at the bottom says 'Xác nhận đặt khám' (Confirm appointment). The footer is light green, showing '@ MEDICAL BOOKING' and social media icons for Facebook, YouTube, and Instagram.

Hình 4.6: Chức năng "Đặt lịch khám bệnh"

4.3.6 Chức năng "Quản lý tài khoản bệnh nhân"

Hình 4.7 mô tả giao diện quản lý tài khoản bệnh nhân, bao gồm một bảng chứa thông tin các tài khoản bệnh nhân và các button xem và xóa. Khi click button xem có thể xem thông tin chi tiết của bệnh nhân và chỉnh sửa nó.

| STT | Email | Tên người dùng | Số điện thoại | Ngày sinh | Địa chỉ | Giới tính | Hành động |
|-----|--------------------------|----------------|---------------|------------|-----------|-----------|-----------|
| 1 | duymoc06112002@gmail.com | Phạm Xuân Duy | 0966632070 | 2002-11-06 | Hải Dương | Nam | Xem Xóa |

Hình 4.7: Chức năng "Quản lý tài khoản bệnh nhân"

4.3.7 Chức năng "Quản lý tài khoản bác sĩ"

Hình 4.8 mô tả giao diện quản lý tài khoản bác sĩ, bao gồm một bảng chứa thông tin các tài khoản bác sĩ. Tại đây chia thành hai trang: trang thứ nhất chứa thông tin các bác sĩ đã được xác thực và phê duyệt, trang thứ hai chứa các thông tin các bác sĩ đang chờ xác thực và phê duyệt.

| STT | Email | Tên bác sĩ | SĐT | Chuyên khoa | Nơi làm việc | Học vị | Số bệnh nhân | Hành động |
|-----|---------------------------|-----------------------|------------|-------------------------|--------------------------------------------------------------------|-------------|--------------|-----------|
| 1 | gaugau@gmail.com | Nguyễn Thị Hằng | 0987654321 | Cơ xương khớp, Cột sống | Trung tâm Khám sức khỏe định kỳ, Bệnh viện Trung ương Quân đội 108 | Phó giáo sư | 0 | Xem Xóa |
| 2 | phamduy06112002@gmail.com | Nguyễn Thị Hoài An | 0987654321 | Tai mũi họng, Nhi khoa | Hệ thống Y tế Thu Cúc TCI | Phó giáo sư | 2 | Xem Xóa |
| 3 | doctor2@gmail.com | Trần Thị Hoài Hương | 0967687652 | Da liễu | Trung tâm Khám sức khỏe định kỳ, Bệnh viện Trung ương Quân đội 108 | Bác sĩ | 0 | Xem Xóa |
| 4 | doctor3@gmail.com | Hà Văn Quyết | 0987656789 | Tiêu hóa | Bệnh viện Chợ Rẫy | Giáo sư | 0 | Xem Xóa |
| 5 | doctor4@gmail.com | Nguyễn Thị Thanh Thủy | 0897657891 | Cơ xương khớp | Trung tâm Khám sức khỏe định kỳ, Bệnh viện Trung ương Quân đội 108 | Bác sĩ | 0 | Xem Xóa |
| 6 | doctor5@gmail.com | Lê Tấn Lợi | 0987678789 | Sức khỏe tâm thần | Bệnh viện Chợ Rẫy | Thạc sĩ | 0 | Xem Xóa |
| 7 | doctor7@gmail.com | Trần Minh Khuyến | 0986867524 | Cột sống, Cơ xương khớp | Bệnh viện Chợ Rẫy | Bác sĩ | 0 | Xem Xóa |

Hình 4.8: Chức năng "Quản lý tài khoản bác sĩ"

4.3.8 Chức năng "Quản lý thông tin cơ sở y tế"

Hình 4.9 mô tả giao diện quản lý thông tin cơ sở y tế, bao gồm một bảng chứa thông tin các cơ sở y tế. Tại đây là nơi quản lý thông tin các cơ sở y tế, đồng thời là nơi điều chỉnh giao diện thông tin chi tiết cơ sở y tế.

| STT | Tên cơ sở y tế | Địa chỉ | Số điện thoại | Email | Website | Hành động |
|-----|--------------------------------------------------------------------|-------------------------------------------------------|-------------------------|---------------------------|-------------------------------------------------------------------------|-----------|
| 1 | Trung tâm Khám sức khỏe định kỳ, Bệnh viện Trung ương Quân đội 108 | Số 1B Trần Hưng Đạo, Quận Hai Bà Trưng, Hà Nội | 096.775.1616 | bvtuqd108@benhvien108.vn | https://www.benhvien108.vn/gioi-thieu-dich-vu-kham-suc-khoe-dinh-ky.htm | Xem Xóa |
| 2 | Hệ thống Y tế Thu Cúc TCI | Cơ sở Thụy Khuê: 286 Thụy Khuê, Tây Hồ, Hà Nội | 0936 388 288 | contact@thucuchospital.vn | https://benhvienthucuc.vn/ | Xem Xóa |
| 3 | Bệnh viện Chợ Rẫy | 201B Nguyễn Chí Thanh, Phường 12, Quận 5, Hồ Chí Minh | 02839555880-02838554137 | truyenthong@bvchoray.com | http://choray.vn/ | Xem Xóa |

Hình 4.9: Chức năng "Quản lý thông tin cơ sở y tế"

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1 Nhận Định Về Thị Trường Hiện Tại

- **Tính năng đa dạng nhưng khó tìm kiếm:** Các hệ thống đặt lịch khám bệnh hiện nay tích hợp nhiều dịch vụ như chọn bác sĩ, phòng khám, và các khóa khám. Tuy nhiên, điều này khiến người dùng phải xử lý nhiều thông tin trước khi chọn lựa phòng khám phù hợp.
- **Thiếu tính đồng bộ giữa các hệ thống:** Hệ thống đặt lịch hiện chưa được đồng bộ với tình trạng lịch khám, dẫn đến khó khăn trong việc đảm bảo lịch hẹn chính xác.

5.1.2 Giá Trị Của Hệ Thống Được Phát Triển

- **Tập trung vào nhu cầu cụ thể của bệnh nhân:** Hệ thống được xây dựng nhằm giúp bệnh nhân dễ dàng tìm bác sĩ, phòng khám, hoặc dịch vụ y tế phù hợp.
- **Dịch vụ tư vấn và hỗ trợ trực tiếp:** Thông qua kênh chat hoặc hotline, bệnh nhân có thể nhận được sự hỗ trợ nhanh chóng, đảm bảo quy trình đặt lịch không gây trở ngại.
- **Tính năng nhắc nhở và theo dõi:** Hệ thống cung cấp những nhắc nhở tự động, giúp bệnh nhân không bỏ lỡ các cuộc hẹn khám bệnh quan trọng.

5.1.3 Những Vấn Đề Chưa Giải Quyết

- **Thiếu tích hợp thanh toán trực tuyến:** Hệ thống chưa hỗ trợ thanh toán trực tuyến, điều này giới hạn sự tiện lợi trong giao dịch.
- **Chưa cập nhật tình trạng lịch theo thời gian thực:** Dẫn đến việc xảy ra tình trạng lịch hẹn trùng lặp hoặc không đồng bộ.

5.1.4 Bài Học Kinh Nghiệm

- **Hiểu biết sâu hơn về nhu cầu y tế:** Quá trình phát triển giúp nâng cao kiến thức về hệ thống y tế và nhu cầu của bệnh nhân.
- **Tập trung vào người dùng:** Sản phẩm thành công khi giải quyết được vấn đề của người dùng.
- **Tích hợp công nghệ hiện đại:** Sử dụng công nghệ tiên tiến và liên tục tối ưu hóa trải nghiệm người dùng.

5.2 Hướng phát triển trong tương lai

5.2.1 Phát Triển Tính Năng Tìm Kiếm Nâng Cao

- **Tìm kiếm theo tiêu chí cụ thể:** Cho phép bệnh nhân lọc theo tiêu chí như chuyên khoa, bác sĩ, giờ lịch khả dụng.
- **Tích hợp AI:** Đề xuất lựa chọn bác sĩ/phòng khám dựa trên thói quen đặt lịch của người dùng.

5.2.2 Mở Rộng Phạm Vi Hoạt Động

- **Hỗ trợ đa ngôn ngữ:** Phục vụ bệnh nhân quốc tế.
- **Kết nối với nhiều hệ thống y tế:** Tích hợp thông tin từ bệnh viện và phòng khám lớn.

5.2.3 Tối Ưu Hóa Trải Nghiệm Người Dùng

- **Phát triển ứng dụng di động:** Tích hợp đặt lịch, nhắc nhở, và thanh toán trực tuyến.
- **Tối ưu giao diện:** Cải thiện UI/UX cho cả website và ứng dụng.
- **Tích hợp chatbot hỗ trợ:** Giải đáp nhanh các thắc mắc của bệnh nhân.

5.2.4 Cập Nhật Tình Trạng Lịch Theo Thời Gian Thực

- **Tích hợp với hệ thống quản lý phòng khám:** Đảm bảo tình trạng được đồng bộ.
- **Thông báo tự động:** Gửi những nhắc nhở quan trọng cho bệnh nhân.
- **Cập nhật theo thời gian thực:** Hiển thị tình trạng còn trống ngay khi bệnh nhân đăng ký.

TÀI LIỆU THAM KHẢO

- [1] NodeJS. [Online]. Available: <https://nodejs.org/en/>, visited on 22/02/2022.
- [2] ReactJS. [Online]. Available: <https://reactjs.org/>, visited on 28/02/2022.
- [3] MongoDB. [Online]. Available: <https://www.mongodb.com/>, visited on 12/02/2022.
- [4] Firebase. [Online]. Available: <https://firebase.google.com/>, visited on 18/02/2022
- [5] NPM. [Online]. Available: <https://www.npmjs.com/>, visited on 10/02/2022
- [6] Yarn. [Online]. Available: <https://yarnpkg.com/>, visited on 20/02/2022