|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **NGUYỄN CHÍ DŨNG** | | **CÔNG NGHỆ THÔNG TIN** | | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**  ĐỒ ÁN TỐT NGHIỆP  NGÀNH CÔNG NGHỆ THÔNG TIN  **XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH BẰNG REACTJS VÀ NODEJS**  **CBHD: ThS. An Văn Minh**  **Sinh viên:  Nguyễn Chí Dũng**  **Mã số sinh viên: 2020606337**  Hà Nội – Năm 2024 |

**LỜI CẢM ƠN**

Trong suốt quá trình học tập, nghiên cứu và làm đồ án tốt nghiệp, em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các thầy, các cô trong Khoa Công nghệ thông tin cùng với sự giúp đỡ của gia đình và bạn bè.

Lời đầu tiên em xin được bày tỏ lòng biết ơn sâu sắc tới Ban giám hiệu Trường Đại học Công Nghiệp Hà Nội, Ban chủ nhiệm khoa cùng các thầy cô giáo trong Khoa Công nghệ thông tin đã tận tình dạy dỗ, chỉ bảo em trong suốt quá trình học tập tại trường.

Đặc biệt em xin tỏ lòng biết ơn chân thành tới thầy giáo **Th.S An Văn Minh**, người đã trực tiếp hướng dẫn và giúp đỡ em hoàn thành khoá luận đồ án này.

Em cũng xin tỏ lòng biết ơn sâu sắc đến gia đình và bạn bè đã luôn động viên, đồng hành, ủng hộ và giúp đỡ em hoàn thiện đồ án tốt nghiệp này.

***Em xin chân thành cảm ơn!***

MỤC LỤC

LỜI MỞ ĐẦU………………………………………………………………....1

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI…………………………………………..2

* 1. Giới thiệu đề tài……………………………………………………..2
  2. Mục đích làm đề tài………………………………………………....2
  3. Lý do chọn đề tài…………………………………………………....2
  4. Cơ sở lý thuyết……………………………………………………...3
  5. Cơ sở thực tiễn………………………………………………………3

CHƯƠNG 2: KIẾN THỨC TỔNG QUAN……………………………………4

2.1 Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu………………………….4

2.1.1 Cơ sở dữ liệu……………………………………………....4

2.1.2 Cơ sở dữ liệu quan hệ…………………………...................4

2.1.3 Hệ quản trị cơ sở dữ liệu…………………………………..5

2.1.4 Hệ quản trị cơ sở dữ liệu MySQL………………………....7

2.2 Lý thuyết về phân tích và thiết kế hệ thống…………………………7

2.3 Ngôn ngữ lập trình……………………………………………….....8

2.4 Thư viện sử dụng……………………………………………………9

2.4.1 ReactJS………………………………………………….....9

2.4.2 Bootstrap………………………………………………....10

2.4 Môi trường thời gian chạy NodeJS………………………………..10

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG…………………...11

3.1 Khảo sát hệ thống………………………………………………....11

3.1.1 Mục tiêu…………………………………………………..11

3.1.2 Phương pháp thực hiện …………………………………..11

3.1.3 Kết quả sơ bộ……………………………………………..12

3.2 Mô tả yêu cầu chức năng……………………………………….....14

3.2.1 Các yêu cầu chức năng……………………………………14

3.2.2 Mô tả use case Đặt lịch khám bệnh………………………14

3.2.3 Mô tả use case Xem thông tin bác sĩ……………………..15

3.2.4 Mô tả use case Xem thông tin chuyên khoa………………16

3.2.5 Mô tả use case Xem thông tin mô tả cơ sở y tế…………..17

3.2.6 Mô tả use case Tìm kiếm………………………………...17

3.2.7 Mô tả use case Quản lý thời gian khám bệnh……………18

3.2.8 Mô tả use case Quản lý người dùng………………………19

3.2.9 Mô tả use case Quản lý chuyên khoa………………….....21

3.2.10 Mô tả use case Quản lý thông tin mô tả cơ sở y tế……..22

3.2.11 Mô tả use case Đăng nhập………………………………24

3.3 Phân tích chi tiết một số use case………………………………….25

3.3.1 Phân tích use case Quản lý người dùng………………......25

3.3.2 Phân tích use case Quản lý chuyên khoa…………………27

3.3.3 Phân tích use case Quản lý thông tin mô tả cơ sở y tế……29

3.3.4 Phân tích use case Quản lý bài viết……………………….31

3.3.5 Phân tích use case Quản lý lịch khám…………………….33

3.3.6 Phân tích use case Xem thông tin bác sĩ………………….34

3.3.7 Phân tích use case Xem thông tin chuyên khoa……….....35

3.3.8 Phân tích use case Xem thông tin mô tả cơ sở y tế………36

3.3.9 Phân tích use case Xem bài viết……………………….....37

3.3.10 Phân tích use case Đăng nhập…………………………..38

3.3.12 Phân tích use case Đặt lịch khám bệnh…………………39

3.3.13 Phân tích use case Tìm kiếm……………………………40

3.4 Mô hình cơ sở dữ liệu……………………………………………..41

3.4.1 Thiết kế bảng……………………………………………..42

3.4.2 Biểu đồ quan hệ thực thể…………………………………45

CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG……………………………………46

4.1 Công cụ sử dụng…………………………………………………..46

4.2 Một số giao diện của trang web……………………………………46

KẾT LUẬN...………………………………………………………………..54

TÀI LIỆU THAM KHẢO……………………………………………………55

DANH MỤC BẢNG

[Bảng 3.1: Lịch khám đã được đặt 42](#_Toc15559)

[Bảng 3.2: Thông tin mô tả cơ sở y tế 42](#_Toc17571)

[Bảng 3.3: Thông tin chi tiết bác sĩ 43](#_Toc28401)

[Bảng 3.4: Tiểu sử và công việc khám chữa bệnh của bác sĩ 43](#_Toc22930)

[Bảng 3.5: Bài viết y tế 44](#_Toc3970)

[Bảng 3.6: Lịch khám của bác sĩ 44](#_Toc13545)

[Bảng 3.7: Thông tin chuyên khoa 44](#_Toc13562)

[Bảng 3.8: Thông tin người dùng 45](#_Toc21029)

[Bảng 3.9: Lý do khám 45](#_Toc6136)

DANH MỤC HÌNH ẢNH

Hình 2.1: Ví dụ về cơ sở dữ liệu quan hệ……………………….................5

Hình 2.2: Cấu trúc của hệ quản trị cơ sở dữ liệu…………………………..6

Hình 2.3: Biểu tượng của ReactJS………………………………………....9

Hình 3.1: Biểu đồ use case tổng quát……………………………………..14

Hình 3.2: Biểu đồ trình tự Quản lý người dùng…………………………..25

Hình 3.3: Biểu đồ lớp phân tích Quản lý người dùng……………………..26

Hình 3.4: Biểu đồ trình tự Quản lý chuyên khoa…………………………27

Hình 3.5: Biểu đồ lớp phân tích Quản lý chuyên khoa…………………...28

Hình 3.6: Biểu đồ trình tự Quản lý thông tin mô tả cơ sở y tế……………29

Hình 3.7: Biểu đồ lớp phân tích Quản lý thông tin mô tả cơ sở y tế……...30

Hình 3.8: Biểu đồ trình tự Quản lý bài viết……………………………….31

Hình 3.9: Biểu đồ lớp phân tích Quản lý bài viết…………………………32

Hình 3.10: Biểu đồ trình tự Quản lý lịch khám…………………………...33

Hình 3.11: Biểu đồ lớp phân tích Quản lý lịch khám……………………..33

Hình 3.12: Biểu đồ trình tự Xem thông tin bác sĩ…………………………34

Hình 3.13: Biểu đồ lớp phân tích Xem thông tin bác sĩ…………………...34

Hình 3.14: Biểu đồ trình tự Xem thông tin chuyên khoa………………….35

Hình 3.15: Biểu đồ lớp phân tích Xem thông tin chuyên khoa……………35

Hình 3.16: Biểu đồ trình tự Xem thông tin mô tả cơ sở y tế………………36

Hình 3.17: Biểu đồ lớp phân tích Xem thông tin mô tả cơ sở y tế………...36

Hình 3.18: Biểu đồ trình tự Xem bài viết………………………………….37

Hình 3.19: Biểu đồ lớp phân tích Xem bài viết……………………………37

Hình 3.20: Biểu đồ trình tự Đăng nhập……………………………………38

Hình 3.21: Biểu đồ lớp phân tích Đăng nhập……...………………………38

Hình 3.22: Biểu đồ trình tự Đặt lịch khám bệnh…………………………..39

Hình 3.23: Biểu đồ lớp phân tích Đặt lịch khám bệnh…………………….39

Hình 3.24: Biểu đồ trình tự Tìm kiếm……………………………………..40

Hình 3.25: Biểu đồ lớp phân tích Tìm kiếm……………………………….40

Hình 3.26: Biểu đồ quan hệ thực thể………………………………………45

Hình 4.1: Ảnh trang chủ website…………………………………………..46

Hình 4.2: Ảnh màn hình chuyên khoa……………………………………..47

Hình 4.3: Ảnh màn hình cơ sở y tế………………………………………...47

Hình 4.4: Ảnh màn hình bác sĩ…………………………………………….48

Hình 4.5: Ảnh màn hình bài viết…………………………………………..48

Hình 4.6: Ảnh màn hình đặt lịch khám bệnh………………………………49

Hình 4.7: Ảnh màn hình tìm kiếm…………………………………………49

Hình 4.8: Ảnh màn hình đăng nhập của bác sĩ và người quản trị…………50

Hình 4.9: Ảnh màn hình quản lý người dùng……………………………...50

Hình 4.10: Ảnh màn hình quản lý thông tin mô tả cơ sở y tế……………...51

Hình 4.11: Ảnh màn hình quản lý chuyên khoa…………………………...51

Hình 4.12: Ảnh màn hình quản lý bài viết…………………………………52

Hình 4.13: Ảnh màn hình quản lý thông tin chi tiết bác sĩ………………...52

Hình 4.14: Ảnh màn hình quản lý kế hoạch khám bệnh của bác sĩ………..53

Hình 4.15: Ảnh màn hình quản lý bệnh nhân của bác sĩ…………………..53

**LỜI MỞ ĐẦU**

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức cũng như các công ty, doanh nghiệp hay cá nhân. Nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.

Cùng với sự phát triển về kỹ thuật máy tính và mạng điện tử, công nghệ thông tin cũng được những công nghệ có đẳng cấp cao lần lượt chinh phục hết đỉnh cao này đến đỉnh cao khác. Mạng Internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên không thể thiếu, là một nền tảng chính cho sự truyền tải, trao đổi thông tin trên toàn cầu.

Hiện nay, trên thế giới thương mại điện tử đang phát triển rất mạnh mẽ. Kỹ thuật số giúp chúng ta tiết kiệm đáng kể các chi phí nhờ chi phí vận chuyển trung gian, chi phí giao dịch và đặc biệt là giúp tiết kiệm thời gian để ta tham gia vào các hoạt động khác. Hơn nữa thương mại điện tử còn giúp chúng ta có thể tìm kiếm tự động theo nhiều mục đích khác nhau, tự động cung cấp thông tin theo nhu cầu và sở thích cá nhân.

Cùng với đó, để đáp ứng nhu cầu chăm sóc và bảo vệ sức khoẻ của cộng đồng nhiều trang web y tế đã được tạo ra với vai trò giúp cho bác sĩ và bệnh nhân có thể gặp mặt và tiếp xúc với nhau nhiều hơn nhằm cải thiện sức khoẻ cho mọi người.

Chính vì vậy, em đã chọn thực hiện đề tài: “Xây dựng website đặt lịch khám bệnh bằng ReactJS và NodeJS”.

Trong đề tài này, em sẽ trình bày quá trình nghiên cứu, phân tích và triển khai hệ thống website đặt lịch khám bệnh, cũng như các kỹ thuật và công nghệ được áp dụng để đảm bảo tính linh hoạt, an toàn và tiện lợi cho người dùng.

**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

* 1. **Giới thiệu đề tài**

Tên đề tài: Xây dựng website đặt lịch khám bệnh bằng ReactJS và NodeJS.

Đây là một dự án nhằm giúp người bệnh có thể đặt lịch khám bệnh trực tuyến một cách dễ dàng và thuận tiện. Website sẽ cung cấp cho người dùng thông tin về các bệnh viện, chuyên khoa và bác sĩ có sẵn để khám bệnh. Người dùng có thể tìm kiếm và chọn lựa bác sĩ phù hợp với nhu cầu của mình, sau đó đặt lịch khám bệnh trực tuyến thông qua website.

**1.2 Mục đích làm đề tài**

Nhằm mục đích tổng hợp và áp dụng những kiến thức đã học trong quá trình đào tạo như lập trình web, quản lý dữ liệu, phân tích và thiết kế hệ thống, … vào một dự án thực tế. Đồng thời củng cố và mở rộng hiểu biết của bản thân về các khía cạnh cụ thể của ngành công nghệ thông tin. Ngoài ra, cá nhân em sẽ được rèn luyện, phát triển kỹ năng, kỹ thuật và tăng cường khả năng tự học, tìm hiểu, giải quyết vấn đề để thiết kế và xây dựng một hệ thống đáp ứng các yêu cầu đã đề ra.

**1.3 Lý do chọn đề tài**

Nhằm vận dụng lý thuyết đã học trên lớp vào một dự án thực tế cùng với việc nhằm đáp ứng nhu cầu khám chữa bệnh đang ngày càng cấp thiết hơn trong đời sống hiện nay. Cùng với đó, để đáp ứng nhu cầu chăm sóc và bảo vệ sức khoẻ ở mọi nơi mọi lúc của con người trang web y tế này đã được tạo ra với vai trò giúp cho bác sĩ và bệnh nhân có thể gặp mặt và tiếp xúc với nhau nhiều hơn nhằm cải thiện sức khoẻ của mọi người.

Với website này người bệnh có thể đặt lịch khám trong khoảng thời gian được cho phép của bác sĩ mà không phải thông qua những cuộc gọi điện thoại hay những cuộc gặp mặt trực tiếp giúp tiết kiệm thời gian đi lại và công sức của người bệnh.

* 1. **Cơ sở lý thuyết**

- Quy trình xây dựng website đặt lịch khám bệnh: Tìm hiểu về các bước cần thiết để xây dựng website đặt lịch khám bệnh từ việc khảo sát và phân tích đến triển khai và vận hành.

- Cơ sở dữ liệu: Xác định cấu trúc của cơ sở dữ liệu bao gồm các bảng, mỗi bảng đại diện cho một loại thông tin và quan hệ giữa chúng. Thiết kế cơ sở dữ liệu để lưu trữ thông tin về bệnh nhân, bác sĩ, thông tin lịch hẹn,...

- Giao diện người dùng: Xác định các trang và thành phần giao diện cần thiết. Thiết kế giao diện người dùng thân thiện và dễ sử dụng.

- Ngôn ngữ lập trình: Tìm hiểu về các ngôn ngữ lập trình được sử dụng để cài đặt website.

- Các thư viện sử dụng: Tìm hiểu về các thư viện được sử dụng để quản lý trạng thái ứng dụng và tạo các thành phần có thể tái sử dụng.

- Công cụ sử dụng: Tìm hiểu các công cụ dùng để xây dựng website đặt lịch khám bệnh bằng ReactJS và NodeJS.

* 1. **Cơ sở thực tiễn**

Việc xây dựng website đặt lịch khám bệnh mang lại rất nhiều lợi ích thực tế như:

* Tạo ra sự tiện lợi cho bệnh nhân: Bệnh nhân có thể dễ dàng đặt lịch hẹn vào bất kỳ thời gian nào mà họ muốn, thay vì phải đến trực tiếp phòng khám hoặc gọi điện thoại để đặt lịch.
* Tiết kiệm thời gian: Bằng cách sử dụng trang web đặt lịch, bệnh nhân không cần phải chờ đợi lâu tại phòng khám để đặt lịch hẹn. Thay vào đó, họ có thể chọn lịch hẹn trống từ trang web.
* Tăng hiệu suất của cơ sở y tế: Hệ thống đặt lịch trực tuyến giúp cơ sở y tế quản lý lịch hẹn của họ hiệu quả hơn và giảm thiểu thời gian không hoạt động.

**CHƯƠNG 2: KIẾN THỨC TỔNG QUAN**

**2.1 Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu**

**2.1.1 Cơ sở dữ liệu**

Cơ sở dữ liệu (Database) là một tập hợp các dữ liệu được tổ chức và lưu trữ trên một hệ thống máy tính. Các dữ liệu này có thể được truy cập, thêm mới, sửa đổi hoặc xóa bỏ bằng các phương thức và công cụ phù hợp. Ngoài ra, cơ sở dữ liệu thường được lưu trữ và truy cập thông qua hệ thống quản lý cơ sở dữ liệu (Database Management System - DBMS).

Chức năng chính của cơ sở dữ liệu bao gồm:

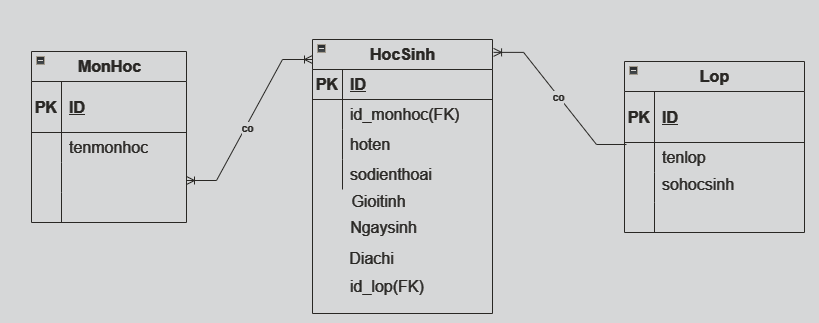
1. **Lưu trữ dữ liệu**: Cơ sở dữ liệu giúp tổ chức và lưu trữ dữ liệu một cách có tổ chức và an toàn. Dữ liệu có thể là thông tin về khách hàng, sản phẩm, đơn hàng, tài liệu, hoặc bất kỳ loại dữ liệu nào khác cần được lưu trữ và quản lý.
2. **Truy xuất dữ liệu**: Người dùng có thể truy cập dữ liệu từ cơ sở dữ liệu để lấy thông tin cụ thể, thực hiện phân tích, hoặc thực hiện các thao tác khác như thêm, sửa đổi hoặc xóa dữ liệu.
3. **Cập nhật dữ liệu**: Cơ sở dữ liệu cho phép người dùng cập nhật dữ liệu mới, sửa đổi dữ liệu hiện có hoặc xóa dữ liệu không cần thiết.
4. **Bảo mật dữ liệu**: Cơ sở dữ liệu cung cấp các cơ chế bảo mật để bảo vệ dữ liệu khỏi sự truy cập trái phép hoặc sửa đổi không được ủy quyền.
5. **Chia sẻ dữ liệu**: Cơ sở dữ liệu cho phép nhiều người dùng cùng truy cập và làm việc với dữ liệu từ xa thông qua mạng Internet hoặc mạng nội bộ.

**2.1.2** **Cơ sở dữ liệu quan hệ**

Cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) là một loại cơ sở dữ liệu được tổ chức dưới dạng các bảng có mối quan hệ với nhau thông qua các khóa chính và khóa ngoài. Đây là một trong những loại cơ sở dữ liệu phổ biến nhất và được sử dụng rộng rãi trong các hệ thống thông tin doanh nghiệp và ứng dụng phần mềm.

Một cơ sở dữ liệu quan hệ bao gồm các thành phần sau:

1. Bảng : Mỗi bảng trong cơ sở dữ liệu quan hệ đại diện cho một tập hợp các dòng và cột. Mỗi dòng trong bảng tương ứng với một bản ghi cụ thể và mỗi cột đại diện cho một thuộc tính.
2. Khóa chính : Mỗi bảng có một hoặc nhiều trường được xác định là khóa chính, đảm bảo rằng mỗi dòng trong bảng là duy nhất.
3. Khóa ngoài : Một trường trong một bảng tham chiếu đến khóa chính của một bảng khác, tạo ra mối quan hệ giữa các bảng.
4. Mối quan hệ : Các mối quan hệ giữa các bảng được xác định thông qua việc sử dụng khóa ngoài. Mối quan hệ này có thể là một-một, một-nhiều hoặc nhiều-nhiều.

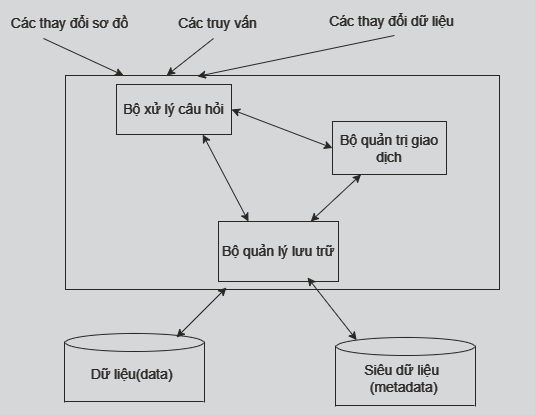
 *Hình 2.1:* *Ví dụ về cơ sở dữ liệu quan hệ*

**2.1.3** **Hệ** **quản trị cơ sở dữ liệu**

Có thể hiểu là hệ thống được thiết kế để quản lí một khối lượng dữ liệu nhất định một cách tự động và có trật tự. Các hành động quản lý này bao gồm chỉnh sửa, xóa, lưu thông tin và tìm kiếm (truy xuất thông tin) trong một nhóm dữ liệu nhất định. Một số hệ quản trị cơ sở dữ liệu phổ biến hiện nay là: SQL Server, Oracle, MySQL,…

Các chức năng chính của một hệ quản trị cơ sở dữ liệu bao gồm:

1. **Quản lý dữ liệu**: Lưu trữ dữ liệu vào cơ sở dữ liệu, xóa, cập nhật và truy vấn dữ liệu một cách hiệu quả.
2. **Bảo mật dữ liệu**: Đảm bảo tính bảo mật của dữ liệu bằng cách quản lý quyền truy cập và thiết lập các biện pháp bảo mật khác nhau.
3. **Khôi phục dữ liệu**: Cung cấp khả năng sao lưu dữ liệu định kỳ và khôi phục dữ liệu trong trường hợp có sự cố xảy ra.
4. **Quản lý và tối ưu hóa hiệu suất**: Tối ưu hóa cơ sở dữ liệu để đảm bảo hiệu suất tốt nhất, bao gồm quản lý chỉ mục, tối ưu hóa truy vấn và cải thiện hiệu suất hệ thống.
5. **Đảm bảo tính toàn vẹn của dữ liệu**: Đảm bảo rằng dữ liệu trong cơ sở dữ liệu luôn đúng và không bị mất mát hoặc hỏng hóc.



*Hình 2.2: Cấu trúc của hệ quản trị cơ sở dữ liệu*

Cấu trúc:

- Bộ Quản lý lưu trữ

* Lưu trữ và truy xuất dữ liệu trên các thiết bị nhớ.
* Tổ chức tối ưu dữ liệu trên thiết bị nhớ.
* Tương tác hiệu quả với bộ quản lý tệp.

- Bộ xử lý câu hỏi

* Tìm kiếm dữ liệu trả lời cho một yêu cầu truy vấn.
* Biến đổi truy vấn ở mức cao thành các yêu cầu có thể hiểu được bởi hệ cơ sở dữ liệu.
* Lựa chọn một kế hoạch tốt nhất để trả lời truy vấn này.

- Bộ Quản trị giao dịch

* Định nghĩa giao dịch: một tập các thao tác được xử lý như một đơn vị không chia căt được.
* Đảm bảo tính đúng đắn và tính nhất quán của dữ liệu.
* Quản lý điều khiển tương tranh.
* Phát hiện lỗi và phục hồi cơ sở dữ liệu

**2.1.4 Hệ quản trị cơ sở dữ liệu MySQL**

Là một trong những [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u" \o "Hệ quản trị cơ sở dữ liệu) [nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), MacOS,...

**MySQL** là một trong những ví dụ rất cơ bản về hệ quản trị cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL).

**MySQL** được sử dụng cho việc bổ trợ NodeJS, [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl) và nhiều ngôn ngữ khác, là nơi lưu trữ những thông tin trên các trang web viết bằng NodeJS, PHP hay Perl,...

**2.2** **Lý thuyết về phân tích và thiết kế hệ thống**

1. **Phân tích hệ thống:** Là quá trình nghiên cứu, hiểu và xác định các yêu cầu, chức năng và cấu trúc của một hệ thống để đảm bảo rằng nó có thể đáp ứng được nhu cầu và mục tiêu của tổ chức hoặc người sử dụng. Nó bao gồm các bước sau:
   * **Thu thập yêu cầu:** Ghi nhận và hiểu yêu cầu từ các bên liên quan, bao gồm cả người dùng cuối và các nhóm liên quan khác.
   * **Phân tích yêu cầu:** Phân tích và phân loại yêu cầu để hiểu rõ nhu cầu và mong muốn của người dùng.
   * **Xác định rủi ro:** Phân tích và đánh giá các rủi ro có thể xảy ra trong quá trình phát triển và triển khai hệ thống.
2. **Thiết kế hệ thống:** Là quá trình tạo ra một bản thiết kế chi tiết và cụ thể cho một hệ thống, bao gồm cả các phần mềm và phần cứng, dựa trên các yêu cầu và thông tin đã thu thập từ quá trình phân tích hệ thống. Nó bao gồm các bước sau:
   * **Kiến trúc hệ thống:** Xác định cấu trúc và các thành phần cơ bản của hệ thống.
   * **Thiết kế dữ liệu:** Xác định cách dữ liệu được tổ chức, lưu trữ và truy cập trong hệ thống.
   * **Thiết kế giao diện người dùng:** Thiết kế các giao diện người dùng để đảm bảo sự tương tác dễ dàng và hiệu quả.
   * **Thiết kế hệ thống phần mềm:** Xác định cách triển khai các tính năng và chức năng của hệ thống thông qua việc phát triển các mô hình và thiết kế phần mềm.

**2.3 Ngôn ngữ lập trình**

Ngôn ngữ để cài đặt website:

- HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng các trang web. Nó định nghĩa cấu trúc và định dạng nội dung trên trang web bằng cách sử dụng các thẻ và các phần tử. HTML không thực hiện các tác vụ lập trình phức tạp, mà chỉ mô tả cấu trúc và hiển thị của các thành phần trên trang web.

- CSS (Cascading Style Sheets) là ngôn ngữ định dạng kiểu dáng cho các tài liệu HTML và XML. Nó được sử dụng để điều khiển cách các phần tử HTML hiển thị trên trang web, bao gồm màu sắc, phông chữ, độ rộng, độ cao, vị trí, hiệu ứng và nhiều thuộc tính khác. CSS giúp tách biệt nội dung (HTML) và kiểu dáng (CSS) trong một trang web, giúp việc thiết kế và bảo trì trang web trở nên dễ dàng hơn.

- JavaScript (JS) là một ngôn ngữ lập trình phía client, thường được sử dụng để thêm tính năng động và tương tác vào các trang web. JS có khả năng tương tác với người dùng thông qua các sự kiện, thay đổi nội dung trang web, thực hiện kiểm tra và xử lý dữ liệu người dùng, và giao tiếp với máy chủ. Ngoài ra, JavaScript cũng được sử dụng để phát triển ứng dụng web đơn trang (Single-Page Applications) và các ứng dụng di động.

* 1. **Thư viện sử dụng**

**2.4.1 ReactJS**

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện JavaScript được dùng để để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng server mà còn ở dưới client nữa.



*Hình 2.3: Biểu tượng của ReactJS*

ReactJS là một thư viện JavaScript mã nguồn mở và miễn phí để xây dựng giao diện người dùng dựa trên các thành phần giao diện người dùng riêng lẻ. ReactJS có thể được sử dụng làm cơ sở để phát triển các ứng dụng SPA (Single-page), thiết bị di động hoặc ứng dụng được kết xuất bằng máy chủ với các thư viện khác như Next.js.

ReactJS chuyên giúp các nhà phát triển xây dựng giao diện người dùng. Trong lập trình ứng dụng, lập trình viên thường sẽ phải làm việc chính trên hai thành phần sau: giao diện người dùng và xử lý tương tác của người dùng. Giao diện người dùng là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,…

* + 1. **Bootstrap**

Bootstrap là một thư viện HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo một tiêu chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,...

Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, JavaScript hỗ trợ cho việc thiết kế responsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

Bootstrap là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng nên một website. Bootstrap đã xây dựng nên một tiêu chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, chúng ta hay nghe tới một cụm từ rất thông dụng “Thiết kế theo chuẩn Bootstrap”.

**2.5** **Môi trường thời gian chạy NodeJS**

Được phát hành vào năm 2009, NodeJS là môi trường thời gian chạy (runtime environment) JavaScript đa nền tảng và mã nguồn mở. NodeJS cho phép các lập trình viên tạo cả ứng dụng hay chương trình bằng JavaScript.

NodeJS là một nền tảng (platform) phát triển độc lập dựa trên V8 JavaScript engine. Đây là một trình thông dịch thực thi mã JavaScript cho phép tạo các ứng dụng web như video clip và diễn đàn, đặc biệt có thể mở rộng nhanh chóng và dễ dàng cũng như thu hẹp phạm vi hoạt động của các trang mạng mạng xã hội.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**3.1 Khảo sát hệ thống**

**3.1.1 Mục tiêu**

Mục tiêu của việc khảo sát hệ thống là thu thập thông tin và đánh giá các khía cạnh khác nhau của một hệ thống. Ngoài ra nó còn giúp tìm hiểu, nghiên cứu và xây dựng về các yêu cầu nghiệp vụ của hệ thống mà khách hàng mong muốn ứng dụng đạt được, mô tả yêu cầu về chức năng và cách thức hoạt động khi trải nghiệm. Từ đó đưa ra phương hướng xây dựng phù hợp để đảm bảo các yêu cầu khách hàng đề ra.

**3.1.2** **Phương pháp thực hiện**

Để thực hiện khảo sát sơ bộ nhằm thu thập thông tin để xây dựng ứng dụng web, thực hiện phỏng vấn với khách hàng.

|  |  |
| --- | --- |
| Phiếu phỏng vấn | |
| Tên dự án: Xây dựng website đặt lịch khám bệnh | |
| **Câu hỏi** | **Ghi chú** |
| Câu 1: Người được phỏng vấn đã sử dụng website đặt lịch khám bệnh trước đây chưa?  Câu 2: Các đối tượng đã từng sử dụng website đặt lịch khám bệnh có yêu cầu đặc biệt nào khi đặt lịch cho bản thân hoặc cho người khác?  Câu 3: Người sử dụng mong muốn website có những chức năng nào?  Câu 4: Người sử dụng mong muốn website có giao diện như thế nào? (đơn giản, độc đáo,…) | Nắm rõ trình tự hoạt động của từng chức năng.  Tư vấn phong cách phù hợp với lĩnh vực. |
| Câu 6: Người sử dụng có quan tâm đến tính bảo mật của trang web không? Và mong muốn trang web đảm bảo tính bảo mật của người dùng như thế nào?  Câu 7: Trên phương diện quản lý, trang quản trị cần quản lý những gì?  Câu 8: Ngoài các chức năng chính của hệ thống, cần thêm chức năng nào không? | Gợi ý một vài chức năng nếu chưa có. |

**3.1.3 Kết quả sơ bộ**

Yêu cầu chức năng:

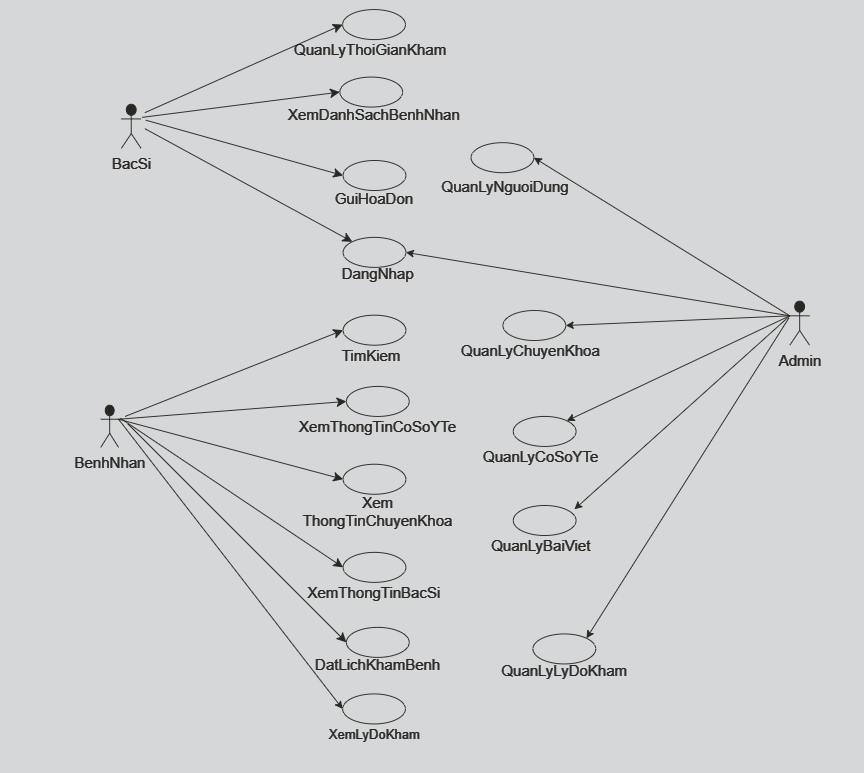
* **Quản lý bác sĩ:** Người quản trị thực hiện các thao tác thêm, cập nhật, xoá bác sĩ khỏi hệ thống.
* **Quản lý chuyên khoa:** Người quản trị thực hiện các thao tác thêm, cập nhật, xoá chuyên khoa khỏi hệ thống.
* **Quản lý thông tin mô tả cơ sở y tế:** Người quản trị thực hiện các thao tác thêm, cập nhật, xoá thông tin mô tả cơ sở y tế khỏi hệ thống.
* **Quản lý bài viết:** Người quản trị thực hiện các thao tác thêm, cập nhật, xoá bài viết khỏi hệ thống.
* **Quản lý lý do khám:** Người quản trị thực hiện các thao tác thêm, cập nhật, xoá lý do khám khỏi hệ thống.
* **Đăng nhập:** Bác sĩ hay người quản trị sử dụng chức năng đăng nhập tài khoản vào hệ thống để sử dụng các chức năng quản trị của hệ thống.
* **Tìm kiếm:** Người dùng cung cấp tên bác sĩ, cơ sở y tế, chuyên khoa hay lý do khám và hệ thống đưa ra kết quả có chứa từ khoá mà người dùng cung cấp.
* **Xem thông tin bác sĩ:** Người dùng có thể xem thông tin mô tả chi tiết của bác sĩ, đồng thời xem lịch khám của bác sĩ đó.
* **Xem thông tin chuyên khoa:** Chức năng cho phép người dùng xem thông tin mô tả của chuyên khoa cũng như bác sĩ thuộc chuyên khoa đó.
* **Xem thông tin mô tả cơ sở y tế:** Chức năng cho phép người dùng xem thông tin mô tả của cơ sở y tế cũng như bác sĩ thuộc cơ sở y tế đó.
* **Xem lý do khám:** Chức năng cho phép người dùng xem danh sách lý do khám phổ biến cũng như bác sĩ chuyên điều trị.
* **Đặt lịch khám bệnh:** Khi người dùng đặt lịch khám thành công, hệ thống sẽ gửi thông tin về email hoặc gọi điện thoại để xác nhận.
* **Xem bài viết:** Chức năng cho phép người dùng xem những bài viết liên quan tới thông tin y tế trong nước hoặc quốc tế.

Yêu cầu phi chức năng:

* Có tính bảo mật thông tin khách hàng, quyền truy cập tới các tài khoản đối với từng nhiệm vụ.
* Tốc độ xử lý nhanh, ổn định, không bị giật hay lỗi khi có nhiều khách hàng truy cập cùng lúc.
* Giao diện đơn giản, dễ sử dụng.
* Các chức năng thể hiện rõ ràng, dễ sử dụng.
* Sử dụng ngôn ngữ Tiếng Việt.
* Khả năng tương thích với các trình duyệt web hiện nay.

**3.2 Mô tả yêu cầu chức năng**

**3.2.1 Các yêu cầu chức năng**

****

*Hình 3.1: Biểu đồ use case tổng quát*

**3.2.2 Mô tả use case Đặt lịch khám bệnh**

**Mô tả vắn tắt:** Use case cho phép bệnh nhân đăng đặt lịch khám bệnh.

- Luồng sự kiện:

+) Luồng cơ bản:

Use case này bắt đầu khi bệnh nhân click vào một ô thời gian khám bệnh bất kỳ ở dưới tiêu đề lịch khám trong màn hình xem chi tiết bác sĩ. Hệ thống hiển thị màn hình yêu cầu người dùng điền các thông tin để đặt lịch.

Người bệnh điền thông tin gồm họ tên (email), số điện thoại, địa chỉ, giới tính, ngày sinh, lý do khám và click vào “Send”. Hệ thống sẽ lưu thông tin vào cơ sở dữ liệu. Hệ thống sẽ gửi mail đến bệnh nhân xác nhận đặt lịch.

Người bệnh kiểm tra gmail của mình rồi nhấn vào nút “Click here” để xác nhận đặt lịch. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt: Không có.

\*) Tiền điều kiện: Không có.

\*) Hậu điều kiện: Nếu xác nhận đặt lịch thành công bệnh nhân sẽ có thể khám bệnh với bác sĩ.

\*) Điểm mở rộng: Không có.

**3.2.3 Mô tả use case Xem thông tin bác sĩ**

**Mô tả vắn tắt**: Use case này cho phép bệnh nhân xem thông tin bác sĩ: tên bác sĩ, ảnh đại diện, thời gian khám, địa chỉ khám, giá khám, chức vụ, tên chuyên khoa, phương thức thanh toán, mô tả.

- Luồng sự kiện:

+) Luồng cơ bản:

Use case này bắt đầu khi bệnh nhân kích vào bác sĩ bất kì. Hệ thống lấy thông tin gồm: tên bác sĩ, địa chỉ khám, giá khám, chức vụ, tên chuyên khoa, phương thức thanh toán, mô tả từ bảng user và doctor\_info. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị lên một thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt: Không có.

\*) Tiền điều kiện: Không có.

\*) Hậu điều kiện: Không có.

\*) Điểm mở rộng: Đặt lịch khám bệnh.

* + 1. **Mô tả use case Xem thông tin chuyên khoa**

**Mô tả vắn tắt**: Use case này cho phép bệnh nhân xem thông tin chuyên khoa: tên chuyên khoa, ảnh, mô tả, danh sách bác sĩ từ chuyên khoa đó.

- Luồng sự kiện:

+) Luồng cơ bản:

Use case này bắt đầu khi bệnh nhân kích vào chuyên khoa bất kì. Hệ thống lấy thông tin gồm: tên chuyên khoa, ảnh, mô tả, danh sách bác sĩ từ chuyên khoa đó từ bảng specialty, user và doctor\_info. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị lên một thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt: Không có.

\*) Tiền điều kiện: Không có.

\*) Hậu điều kiện: Không có.

\*) Điểm mở rộng: Xem thông tin bác sĩ.

* + 1. **Mô tả use case Xem thông tin mô tả cơ sở y tế**

**Mô tả vắn tắt**: Use case này cho phép bệnh nhân xem thông tin mô tả cơ sở y tế: tên cơ sở y tế, ảnh, mô tả, danh sách bác sĩ từ cơ sở y tế đó.

- Luồng sự kiện:

+) Luồng cơ bản:

Use case này bắt đầu khi người bệnh kích vào hình ảnh của một cơ sở y tế bất kì. Hệ thống lấy thông tin gồm: tên cơ sở y tế, ảnh, mô tả, danh sách bác sĩ từ cơ sở y tế đó từ bảng clinic, user và doctor\_info. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị lên một thông báo lỗi và use case kết thúc

\*) Các yêu cầu đặc biệt: Không có.

\*) Tiền điều kiện: Không có.

\*) Hậu điều kiện: Không có.

\*) Điểm mở rộng: Xem thông tin bác sĩ.

**3.2.6 Mô tả use case Tìm kiếm**

**Mô tả vắn tắt**: Use case cho phép bệnh nhân gõ từ khóa để tìm kiếm tên cơ sở y tế, chuyên khoa, bác sĩ hay lý do khám.

- Luồng sự kiện:

+) Luồng cơ bản:

1. Use case này bắt đầu khi bệnh nhân click vào ô “Tìm kiếm” ở thanh tìm kiếm.

2. Bệnh nhân gõ từ khoá cần tìm rồi nhấn Enter. Hệ thống sẽ hiển thị bác sĩ, cơ sở y tế hoặc chuyên khoa có liên quan đến thông tin bệnh nhân tìm kiếm.

3. Bệnh nhân click vào một bác sĩ, cơ sở y tế hoặc chuyên khoa mà mình muốn tìm. Hệ thống hiển thị lên màn hình: tên bác sĩ, cơ sở y tế, chuyên khoa hoặc lý do khám từ bảng doctor\_info, clinic, specialty hoặc reasons. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bước 2 trong luồng cơ bản, khi bệnh nhân nhập từ khóa mà hệ thống không tìm thấy. Hệ thống sẽ hiển thị dòng chữ “0 result”. Quay lại bước 2 trong luồng cơ bản.

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt: Không có.

\*) Tiền điều kiện: Không có.

\*) Hậu điều kiện: Không có.

\*) Điểm mở rộng: Không có.

**3.2.7 Mô tả use case Quản lý thời gian khám bệnh**

**Mô tả vắn tắt:** Use case này cho phép bác sĩ xem, sửa, xóa thông tin trong bảng schedule.

- Luồng sự kiện:

+) Luồng cơ bản:

1. Use case này bắt đầu khi bác sĩ kích vào nút “Quản lý thời gian khám bệnh” trên menu quản trị. Bác sĩ chọn ngày khám bệnh. Hệ thống lấy danh sách các khoảng thời gian trong ngày từ bảng allcode trong cơ sở dữ liệu và hiển thị danh sách các khoảng thời gian lên màn hình.

2. Sửa trạng thái: Người quản trị kích vào nút khoảng thời gian để tạo lịch trong ngày đã chọn rồi nhấn nút “Tạo lịch”. Hệ thống sẽ cập nhật vào bảng schedule.

3. Xoá lịch khám: Người quản trị click vào lại nút khoảng thời gian rồi click nút “cập nhật” . Hệ thống sẽ cập nhật từ bảng schedule.

4. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt:

Use case này chỉ cho phép một số vai trò như bác sĩ, người quản trị thực hiện.

\*) Tiền điều kiện:

Bác sĩ cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

\*) Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin về lịch khám trong ngày sẽ được cập nhập trong cơ sở dữ liệu.

\*) Điểm mở rộng: Không có.

* + 1. **Mô tả use case Quản lý người dùng**

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem, sửa trạng thái, xóa thông tin trong bảng user.

- Luồng sự kiện:

+) Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị kích vào nút “Quản lý người dùng” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các tài khoản gồm: tên tài khoản (email), first name, last name, số điện thoại, địa chỉ từ bảng user trong cơ sở dữ liệu và hiển thị danh sách các tài khoản lên màn hình.

2. Sửa trạng thái:

Bước 2.1: Người quản trị kích vào nút “Sửa” trên một dòng tài khoản. Hệ thống sẽ lấy thông tin cũ của tài khoản được chọn gồm: tên tài khoản (email), first name, last name, số điện thoại, địa chỉ, giới tính, ảnh đại diện, chức vụ và vai trò từ bảng user và hiển thị lên màn hình.

Bước 2.2: Người quản trị nhập thông tin mới cho tài khoản và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của tài khoản được chọn trong bảng user và hiển thị danh sách tài khoản đã cập nhật.

3. Xoá người dùng: Người quản trị click vào nút “Xoá” trên dòng tài khoản. Hệ thống sẽ xoá thông tin người dùng từ bảng user và hiển thị danh sách tài khoản đã cập nhật.

4. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bước 2.2 trong luồng cơ bản nếu người quản trị nhập thông tin tài khoản không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.

Tại bước 2.2 trong luồng cơ bản nếu người quản trị kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác sửa chữa tương ứng và hiển thị danh sách các tài khoản trong bảng user.

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt:

Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

\*) Tiền điều kiện:

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

\*) Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin về tài khoản sẽ được cập nhập trong cơ sở dữ liệu.

\*) Điểm mở rộng: Không có.

**3.2.9 Mô tả use case Quản lý chuyên khoa**

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem, sửa trạng thái, xóa thông tin trong bảng specialty.

- Luồng sự kiện:

+) Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị kích vào nút “Quản lý chuyên khoa” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các chuyên khoa gồm: tên chuyên khoa từ bảng specialty trong cơ sở dữ liệu và hiển thị danh sách các chuyên khoa lên màn hình.

2. Sửa trạng thái:

Bước 2.1: Người quản trị kích vào nút “Sửa” trên một dòng chuyên khoa. Hệ thống sẽ lấy thông tin cũ của chuyên khoa được chọn gồm: tên chuyên khoa, ảnh đại diện và mô tả từ bảng specialty và hiển thị lên màn hình.

Bước 2.2: Người quản trị nhập thông tin mới cho chuyên khoa và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của chuyên khoa được chọn trong bảng specialty và hiển thị danh sách chuyên khoa đã cập nhật.

3. Xoá chuyên khoa: Người quản trị click vào nút “Xoá” trên dòng chuyên khoa. Hệ thống sẽ xoá thông tin chuyên khoa từ bảng specialty và hiển thị danh sách chuyên khoa đã cập nhật.

4. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bước 2.2 trong luồng cơ bản nếu người quản trị nhập thông tin chuyên khoa không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.

Tại bước 2.2 trong luồng cơ bản nếu người quản trị kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác sửa chữa tương ứng và hiển thị danh sách các chuyên khoa trong bảng specialty.

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc…

\*) Các yêu cầu đặc biệt:

Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

\*) Tiền điều kiện:

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

\*) Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin về chuyên khoa sẽ được cập nhập trong cơ sở dữ liệu.

\*) Điểm mở rộng: Không có.

**3.2.10 Mô tả use case Quản lý thông tin mô tả cơ sở y tế**

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem, sửa trạng thái, xóa thông tin trong bảng clinic.

* Luồng sự kiện:

+) Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị kích vào nút “Quản lý cơ sở y tế” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các cơ sở y tế gồm: tên cơ sở y tế từ bảng clinic trong cơ sở dữ liệu và hiển thị danh sách các tên cơ sở y tế lên màn hình.

2. Sửa trạng thái:

Bước 2.1: Người quản trị kích vào nút “Sửa” trên một dòng cơ sở y tế. Hệ thống sẽ lấy thông tin cũ của cơ sở y tế được chọn gồm: tên cơ sở y tế, ảnh đại diện và mô tả từ bảng clinic và hiển thị lên màn hình.

Bước 2.2: Người quản trị nhập thông tin mới cho cơ sở y tế và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của cơ sở y tế được chọn trong bảng clinic và hiển thị danh sách cơ sở y tế đã cập nhật.

3. Xoá cơ sở y tế: Người quản trị click vào nút “Xoá” trên dòng cơ sở y tế. Hệ thống sẽ xoá thông tin cơ sở y tế từ bảng clinic và hiển thị danh sách cơ sở y tế đã cập nhật.

4. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bước 2.2 trong luồng cơ bản nếu người quản trị nhập thông tin cơ sở y tế không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.

Tại bước 2.2 trong luồng cơ bản nếu người quản trị kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác sửa chữa tương ứng và hiển thị danh sách các cơ sở y tế trong bảng clinic.

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc…

\*) Các yêu cầu đặc biệt:

Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

\*) Tiền điều kiện:

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

\*) Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin về cơ sở y tế sẽ được cập nhập trong cơ sở dữ liệu.

\*) Điểm mở rộng: Không có.

**3.2.11 Mô tả use case Đăng nhập**

**Mô tả vắn tắt:** Use case này cho phép bác sĩ hoặc người quản trị đăng nhập vào hệ thống để thực hiện chức năng quản trị.

- Luồng sự kiện:

+) Luồng cơ bản:

1. Use case này bắt đầu khi một bác sĩ hoặc người quản trị muốn đăng nhập vào hệ thống. Hệ thống yêu cầu bác sĩ hoặc người quản trị nhập tên tài khoản (email) và mật khẩu.

2. Bác sĩ hoặc người quản trị nhập tên tài khoản và mật khẩu sau đó kích vào nút “Login”. Hệ thống kiểm tra tên và mật khẩu đã nhập và quay trở lại trang quản trị. Use case kết thúc.

+) Luồng rẽ nhánh:

Tại bước 2 trong luồng cơ bản, nếu bác sĩ hoặc người quản trị nhập tên tài khoản hoặc mật khẩu sai, hệ thống hiển thị một thông báo “Tên tài khoản hoặc mật khẩu sai!”. Bác sĩ hoặc người quản trị có thể nhập lại hoặc bỏ qua thao tác khi đó use case sẽ kết thúc.

Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thi một thông báo lỗi và use case kết thúc.

\*) Các yêu cầu đặc biệt: Không có.

\*) Tiền điều kiện: Không có.

\*) Hậu điều kiện:

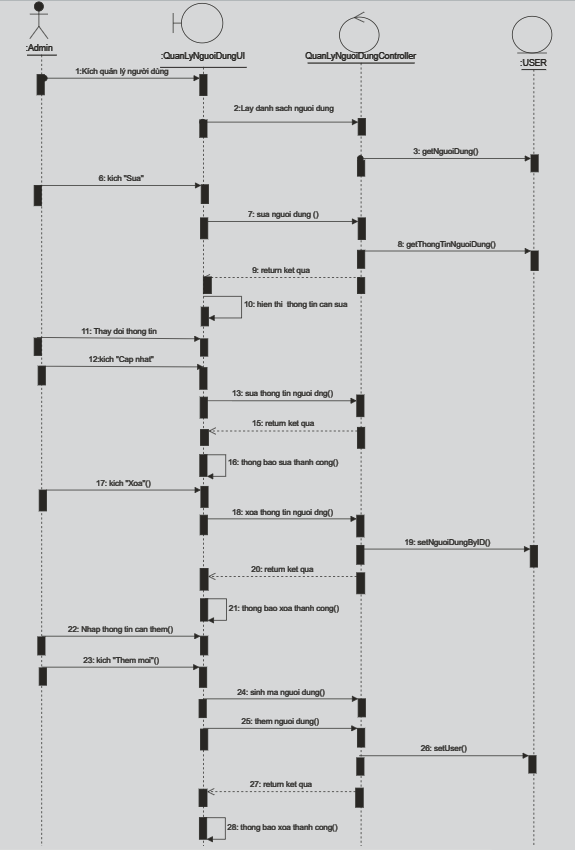
Nếu use case thành công, bác sĩ hoặc người quản trị sẽ đăng nhập được vào hệ thống. Nếu không, trạng thái của hệ thống không thay đổi.

\*) Điểm mở rộng: Không có.

**3.3 Phân tích chi tiết một số use case**

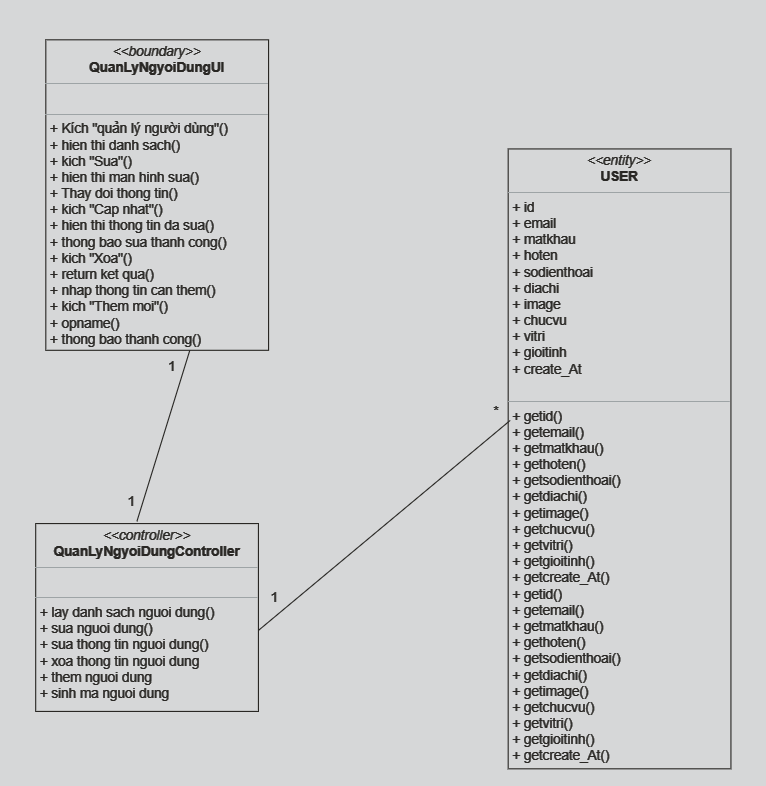
**3.3.1 Phân tích use case Quản lý người dùng**

Biểu đồ trình tự:



*Hình 3.2: Biểu đồ trình tự Quản lý người dùng*

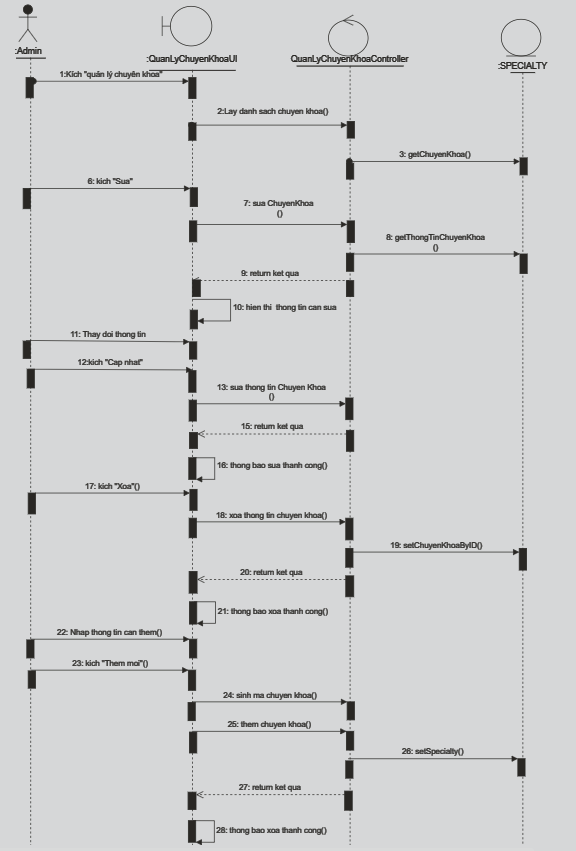
Biểu đồ lớp phân tích:



*Hình 3.3: Biểu đồ lớp phân tích Quản lý người dùng*

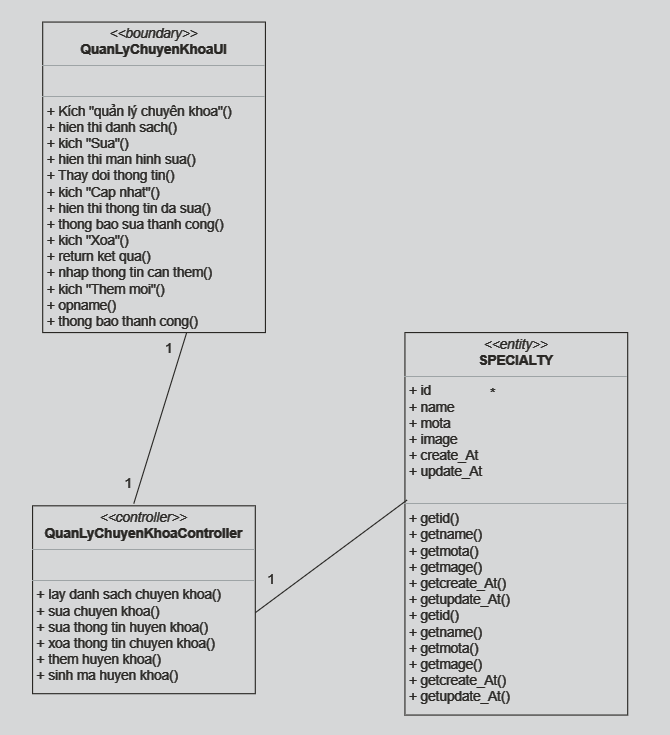
**3.3.2 Phân tích use case Quản lý chuyên khoa**

Biểu đồ trình tự:



*Hình 3.4: Biểu đồ trình tự Quản lý chuyên khoa*

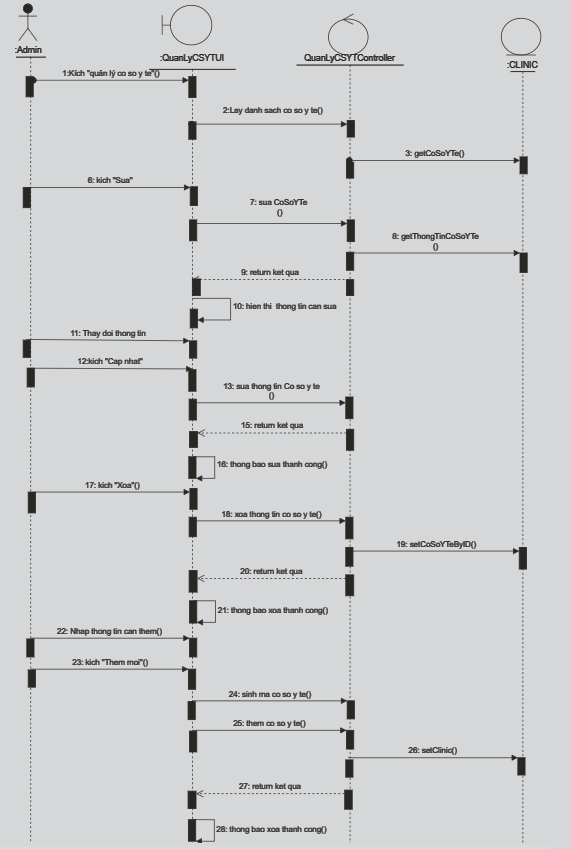
Biểu đồ lớp phân tích:



*Hình 3.5: Biểu đồ lớp phân tích Quản lý chuyên khoa*

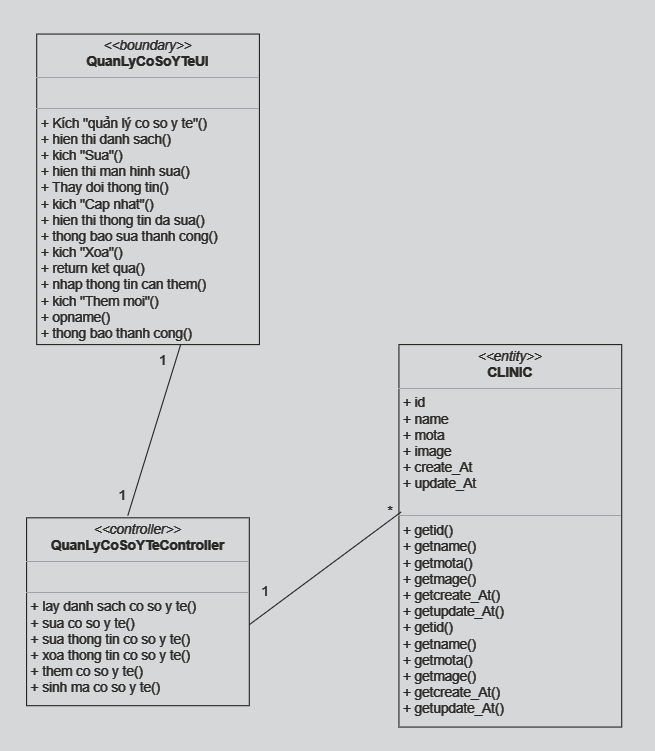
**3.3.3 Phân tích use case Quản lý thông tin mô tả cơ sở y tế**

Biểu đồ trình tự:



*Hình 3.6: Biểu đồ trình tự Quản lý thông tin mô tả cơ sở y tế*

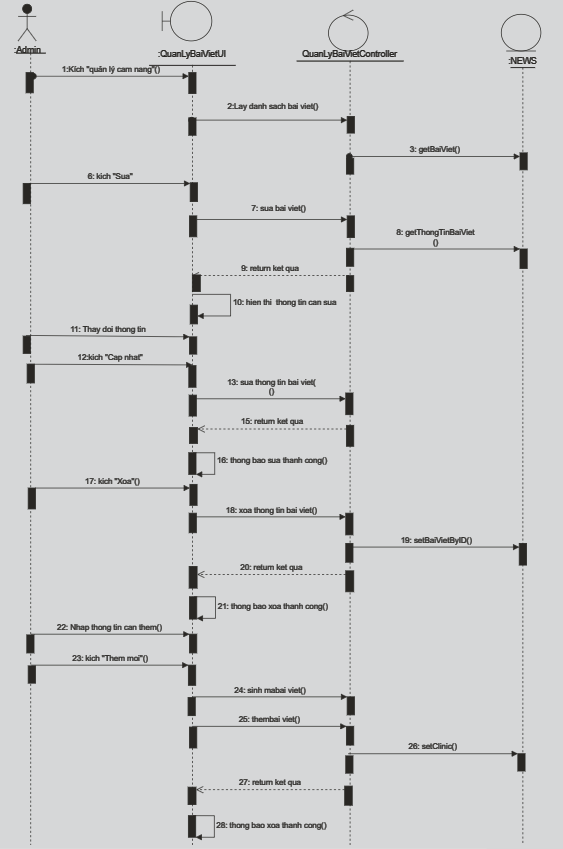
Biểu đồ lớp phân tích:



*Hình 3.7: Biểu đồ lớp phân tích Quản lý thông tin mô tả cơ sở y tế*

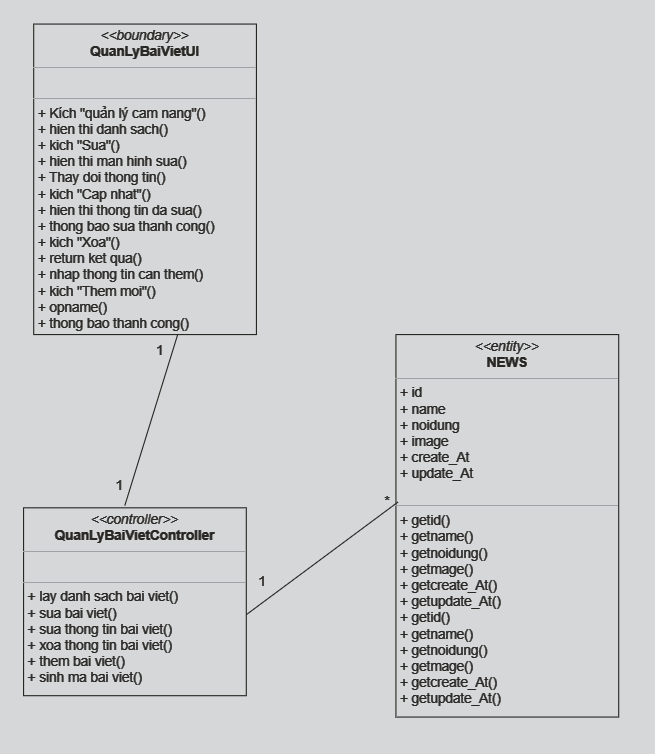
**3.3.4 Phân tích use case Quản lý bài viết**

Biểu đồ trình tự:



*Hình 3.8: Biểu đồ trình tự Quản lý bài viết*

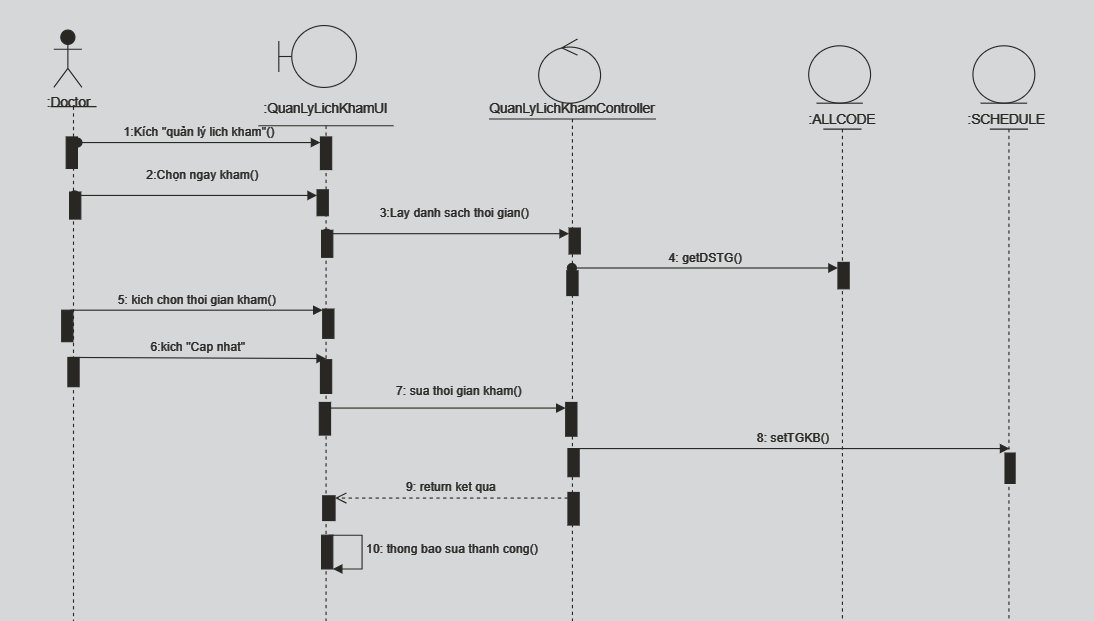
Biểu đồ lớp phân tích:



*Hình 3.9: Biểu đồ lớp phân tích Quản lý bài viết*

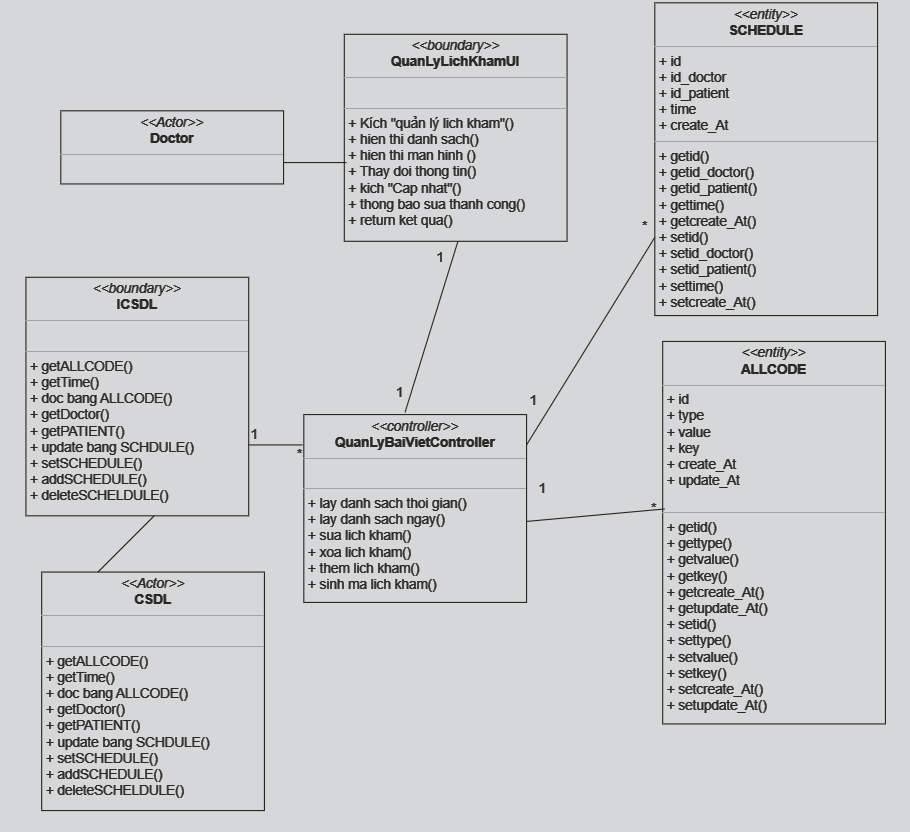
**3.3.5 Phân tích use case Quản lý lịch khám**

Biểu đồ trình tự:



*Hình 3.10: Biểu đồ trình tự Quản lý lịch khám*

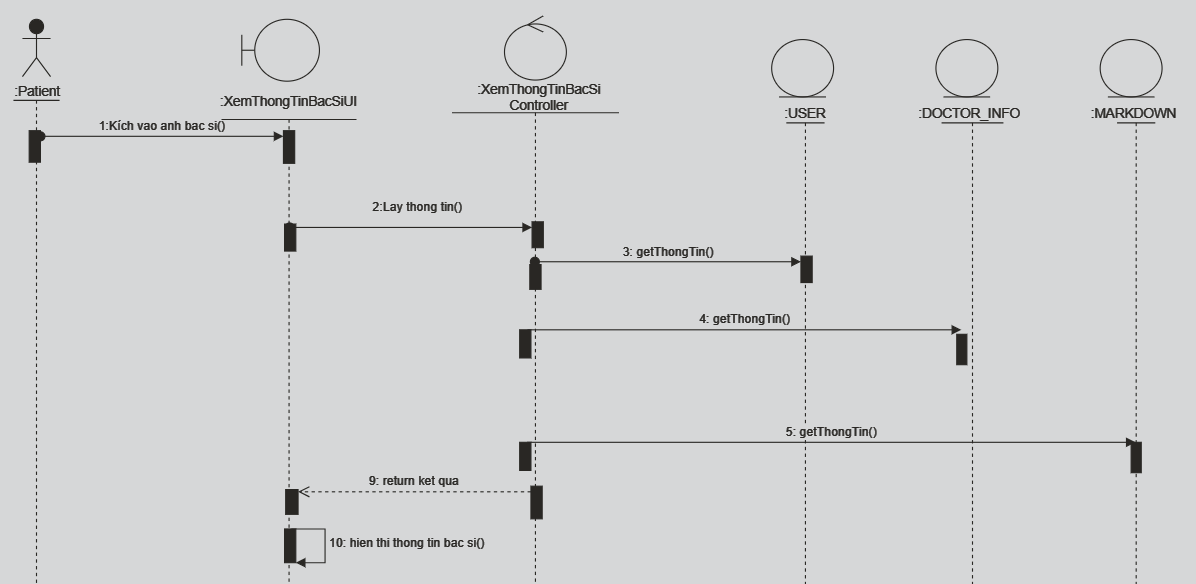
Biểu đồ lớp phân tích:



*Hình 3.11: Biểu lớp phân tích Quản lý lịch khám*

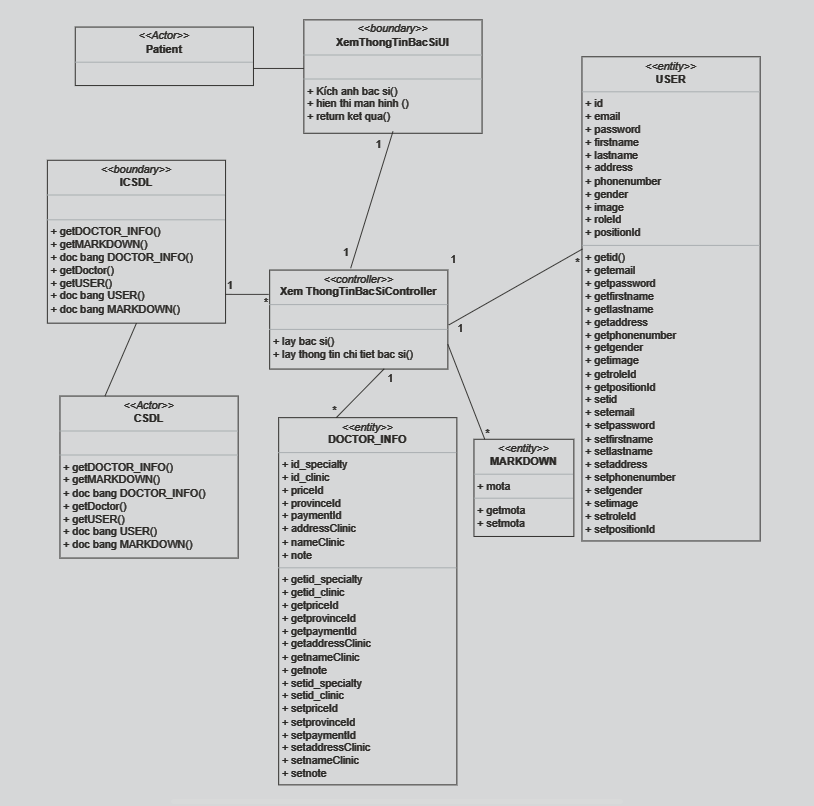
**3.3.6 Phân tích use case Xem thông tin bác sĩ**

Biểu đồ trình tự:



*Hình 3.12: Biểu đồ trình tự Xem thông tin bác sĩ*

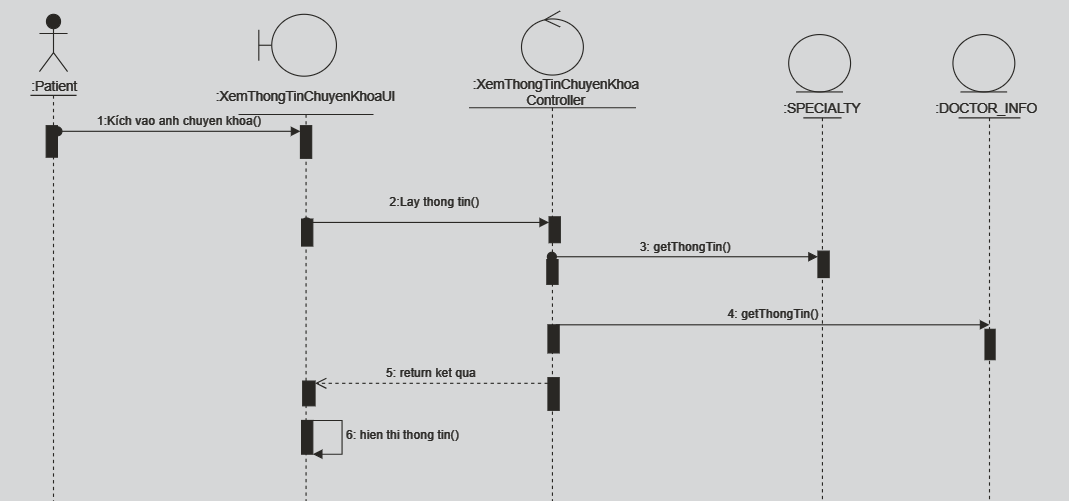
Biểu đồ lớp phân tích:



*Hình 3.13: Biểu đồ lớp phân tích Xem thông tin bác sĩ*

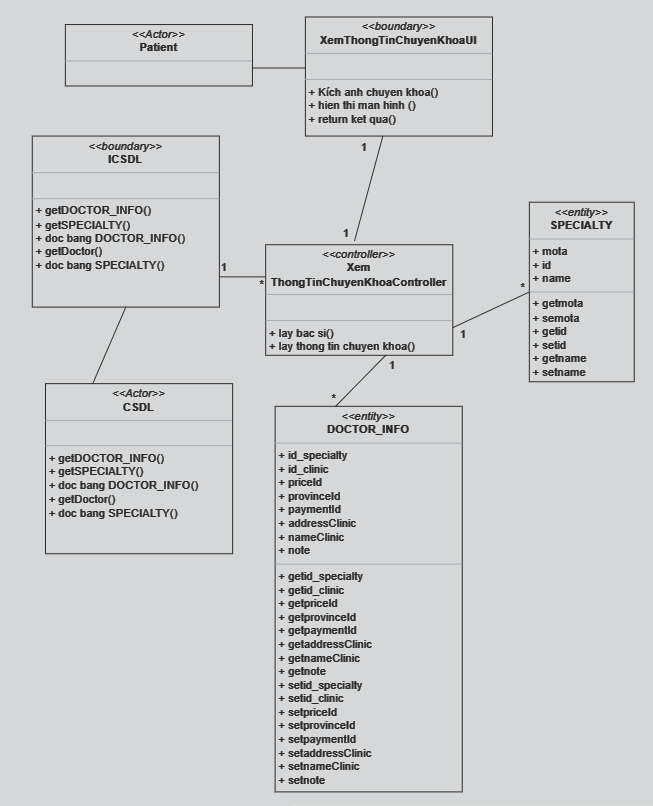
**3.3.7 Phân tích use case Xem thông tin chuyên khoa**

Biểu đồ trình tự:



*Hình 3.14: Biểu đồ trình tự Xem thông tin chuyên khoa*

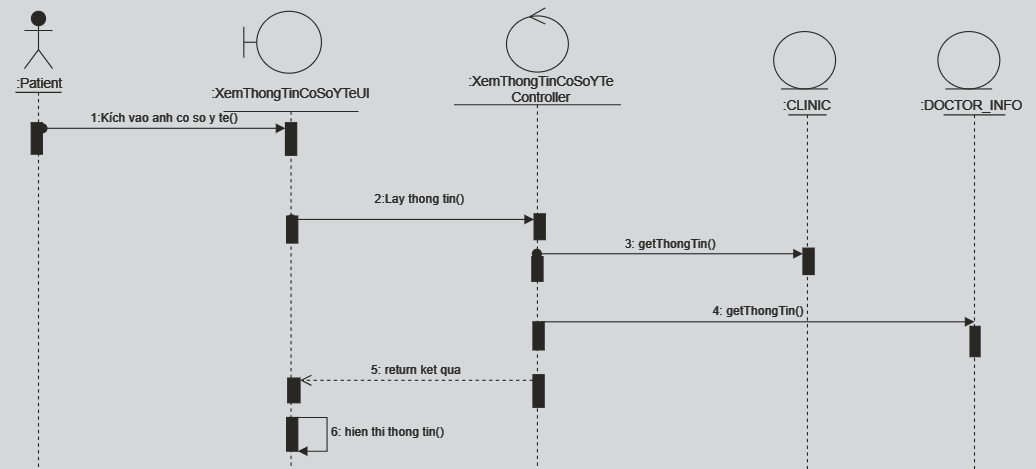
Biểu đồ lớp phân tích:



*Hình 3.15: Biểu đồ lớp phân tích Xem thông tin chuyên khoa*

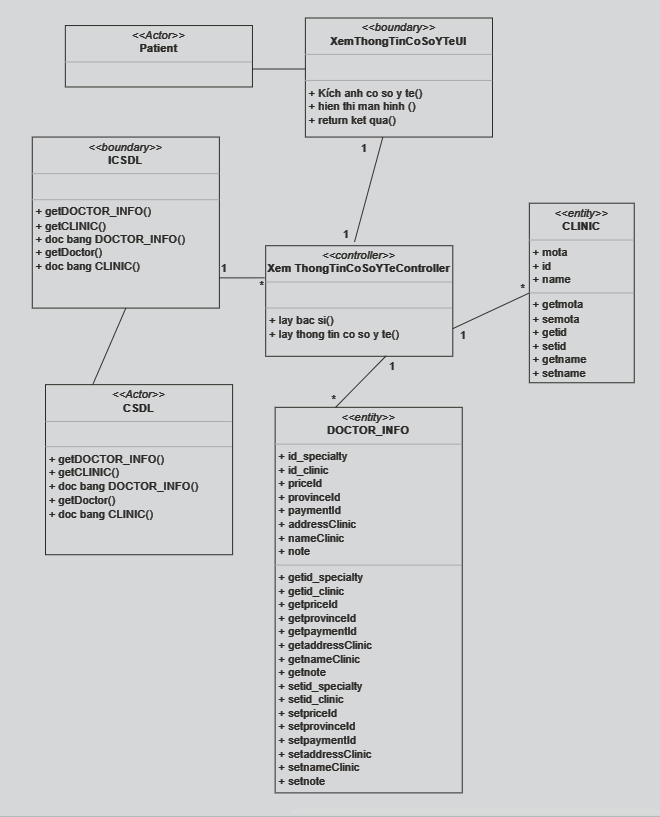
**3.3.8 Phân tích use case Xem thông tin mô tả cơ sở y tế**

Biểu đồ trình tự:



*Hình 3.16: Biểu đồ trình tự Xem thông tin cơ sở y tế*

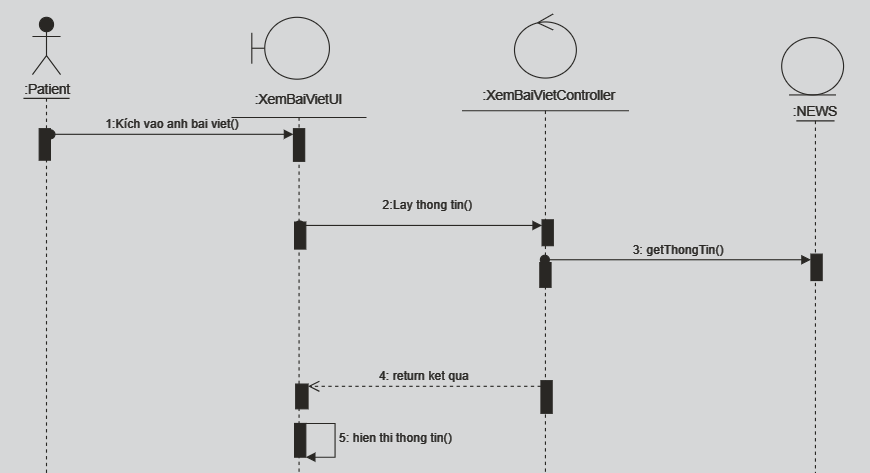
Biểu đồ lớp phân tích:



*Hình 3.17: Biểu đồ lớp phân tích Xem thông tin cơ sở y tế*

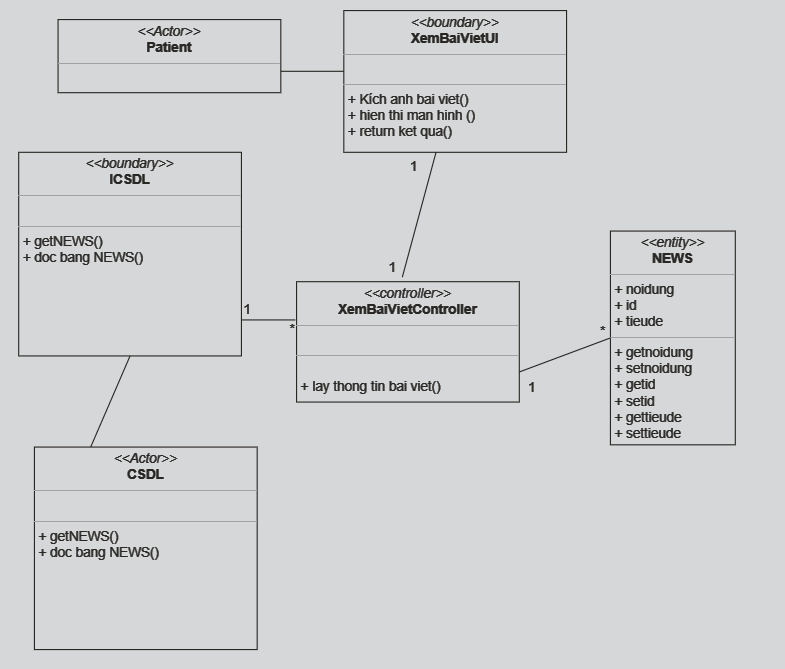
**3.3.9 Phân tích use case Xem bài viết**

Biểu đồ trình tự:



*Hình 3.18: Biểu đồ trình tự Xem bài viết*

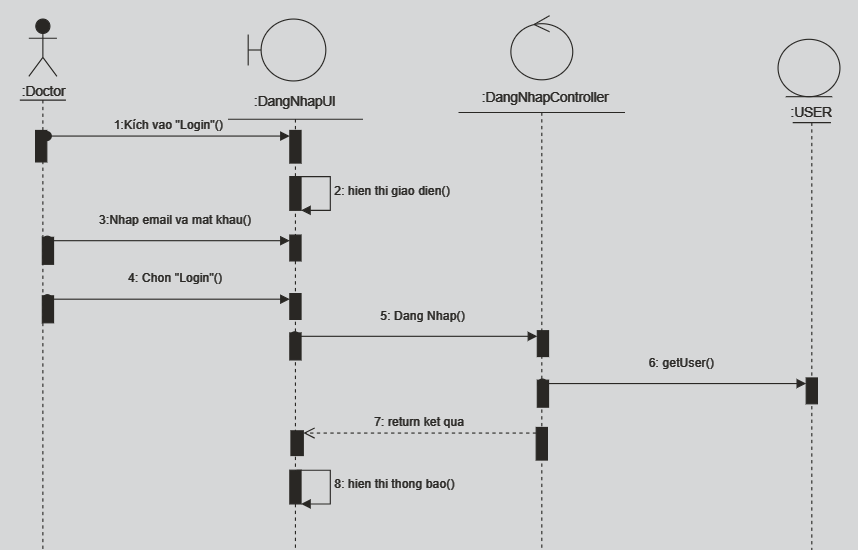
Biểu đồ lớp phân tích:



*Hình 3.19: Biểu đồ lớp phân tích Xem bài viết*

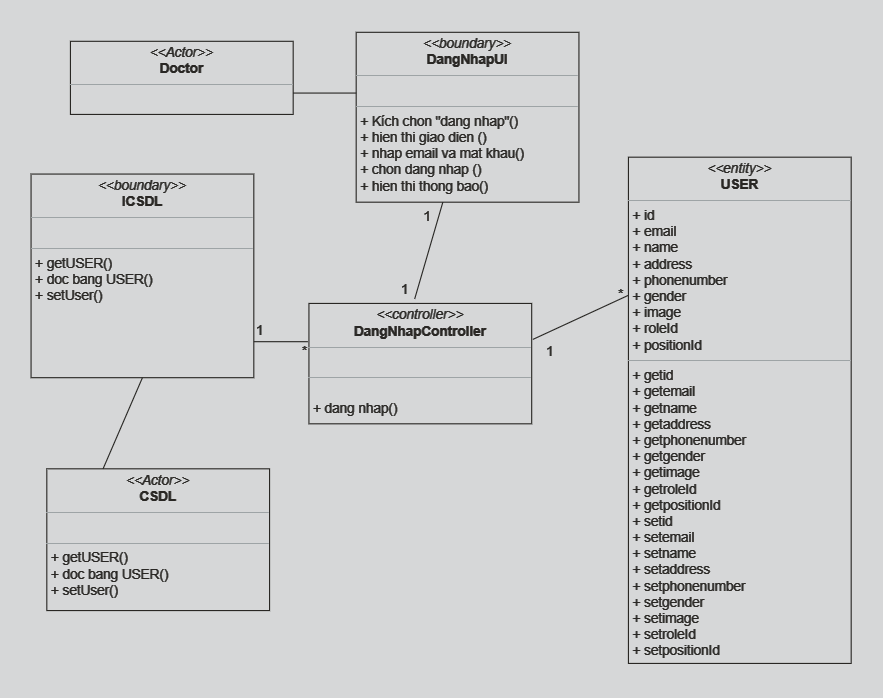
**3.3.10 Phân tích use case Đăng nhập**

Biểu đồ trình tự:



*Hình 3.20: Biểu đồ trình tự Đăng nhập*

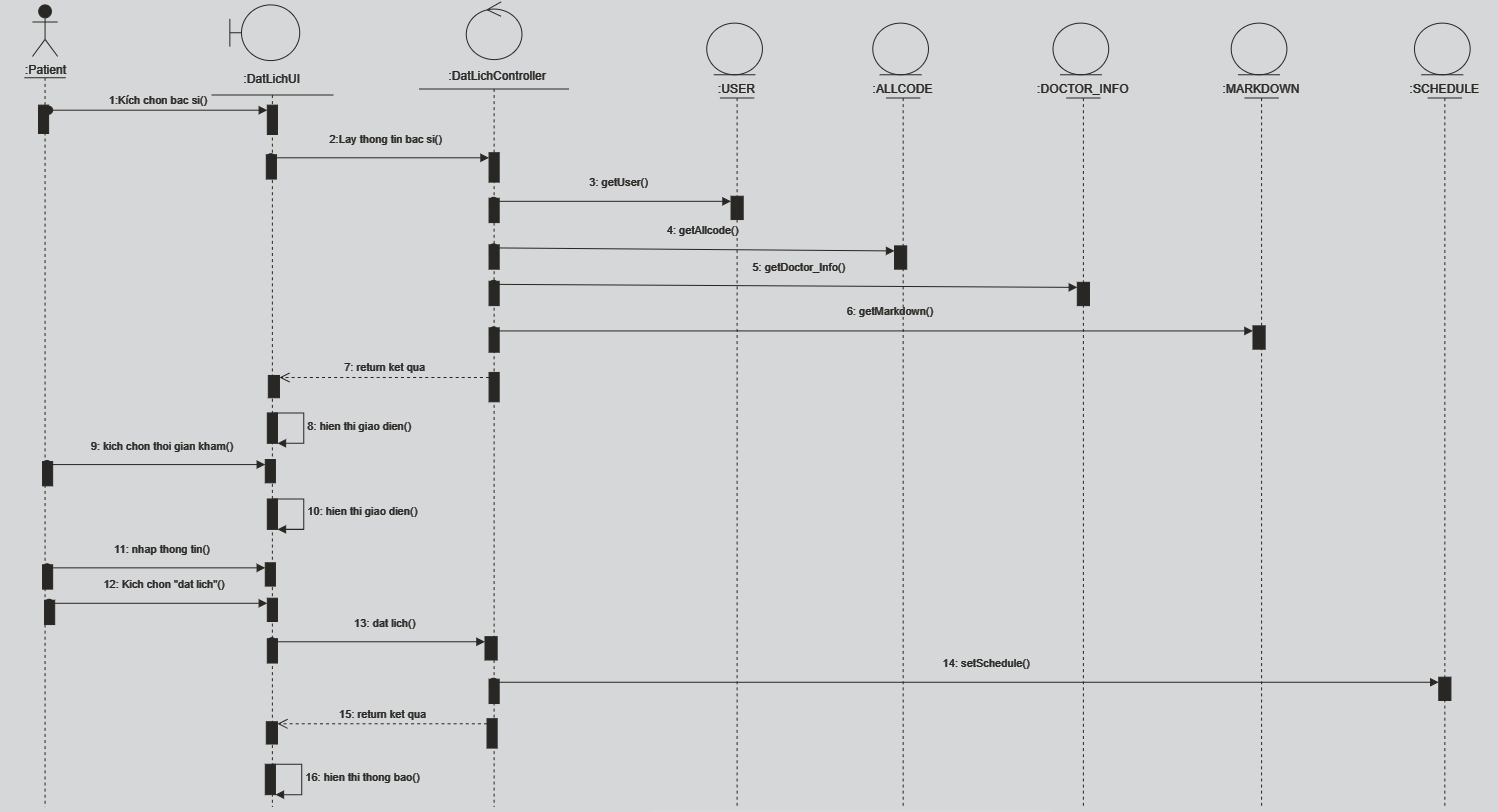
Biểu đồ lớp phân tích:



*Hình 3.21: Biểu đồ lớp phân tích Đăng nhập*

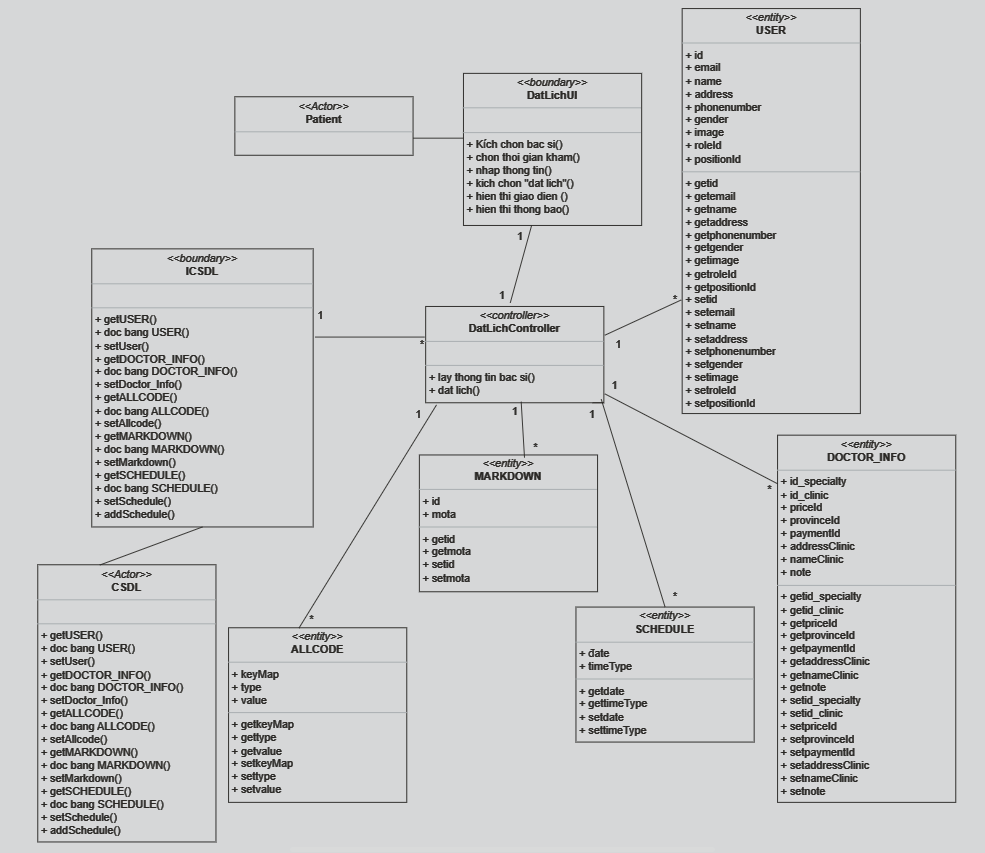
**3.3.11 Phân tích use case Đặt lịch khám bệnh**

Biểu đồ trình tự:



*Hình 3.22: Biểu đồ trình tự Đặt lịch khám bệnh*

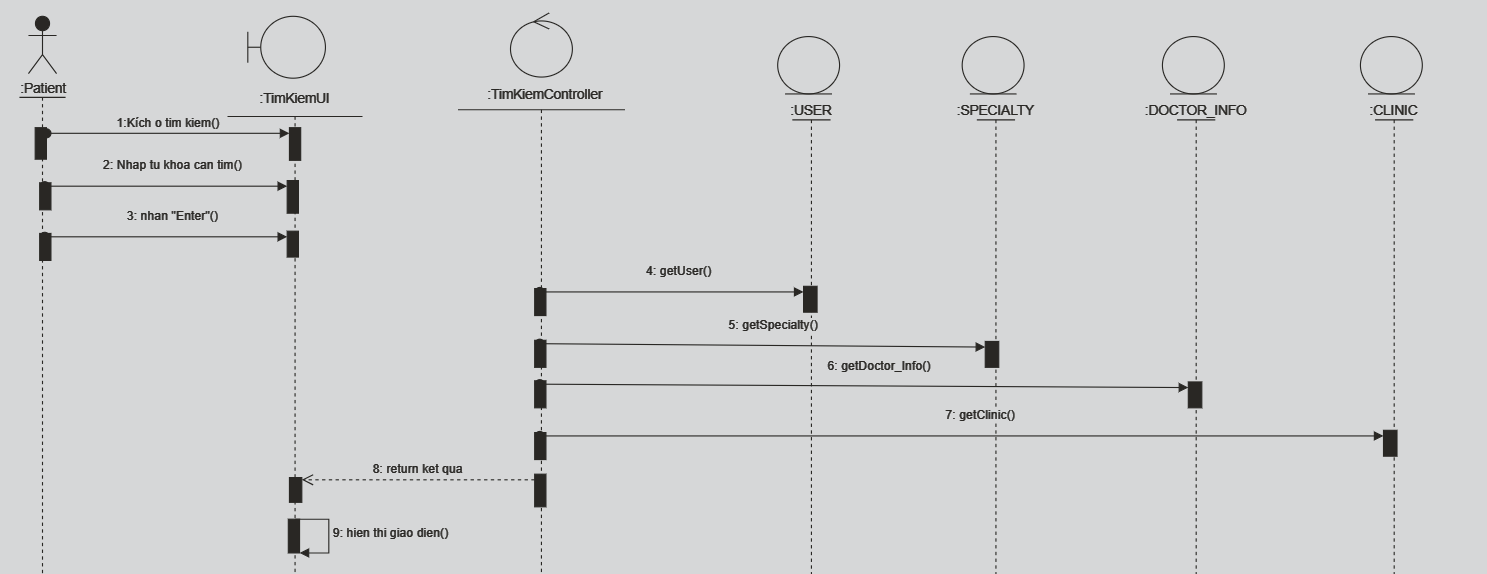
Biểu đồ lớp phân tích:



*Hình 3.23: Biểu đồ lớp phân tích Đặt lịch khám bệnh*

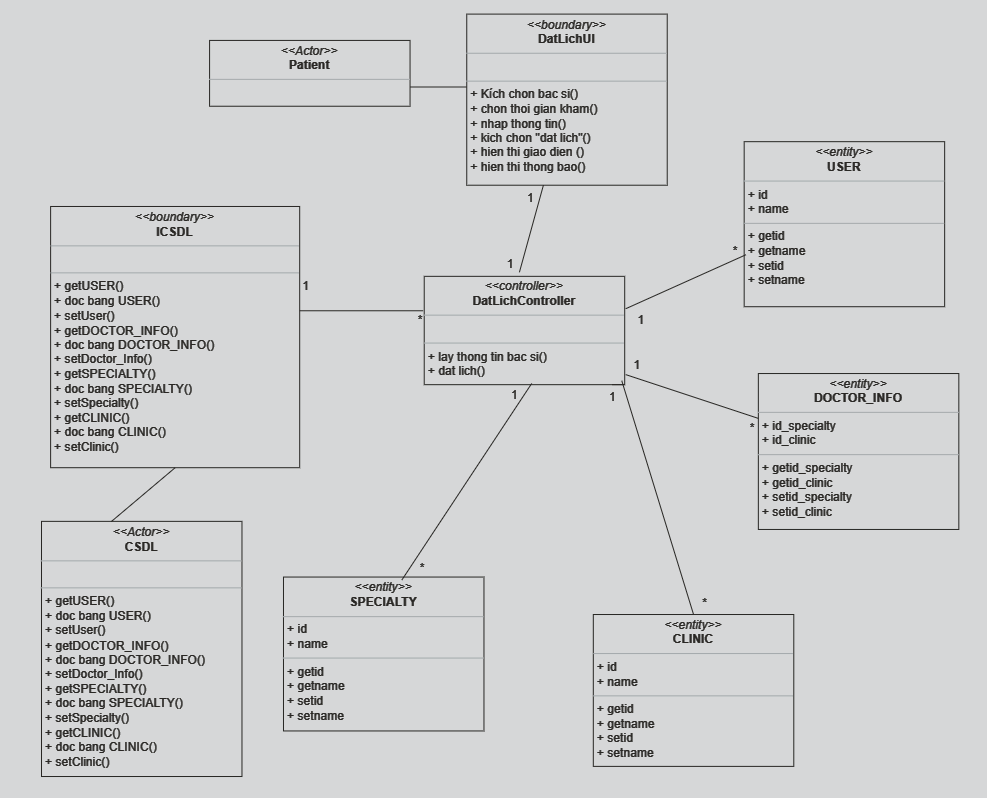
**3.3.12 Phân tích use case Tìm kiếm**

Biểu đồ trình tự:



*Hình 3.24: Biểu đồ trình tự Tìm kiếm*

Biểu đồ lớp phân tích:



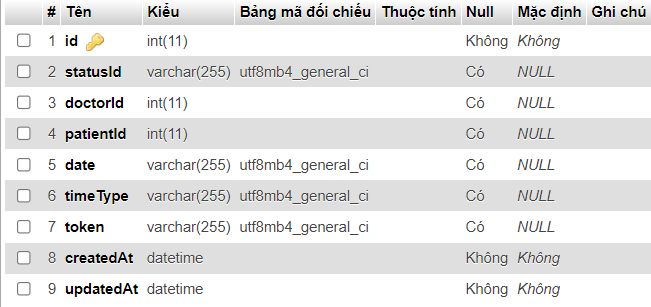
*Hình 3.25: Biểu đồ lớp phân tích Tìm kiếm*

**3.4 Mô hình cơ sở dữ liệu**

**3.4.1 Thiết kế bảng**

\*) Bảng thông tin về lịch khám đã được bệnh nhân đặt lịch.

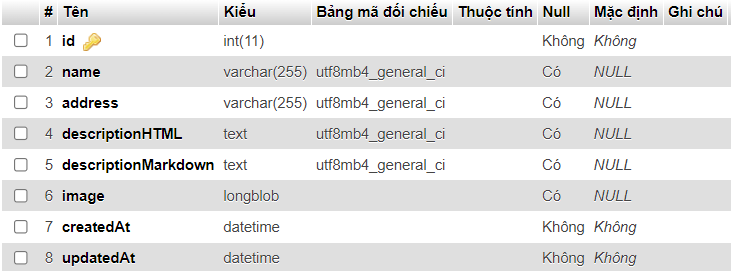
- Tên bảng: booking

**

*Bảng 3.1: Lịch khám đã được đặt*

\*) Bảng thông tin mô tả về cơ sở y tế.

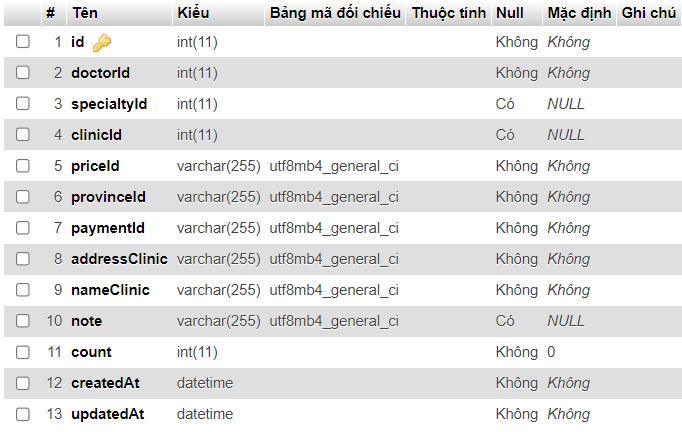
- Tên bảng: clinic

**

*Bảng 3.2: Thông tin mô tả cơ sở y tế*

\*) Bảng thông tin chi tiết về bác sĩ như: chuyên khoa, giá khám, địa chỉ khám…

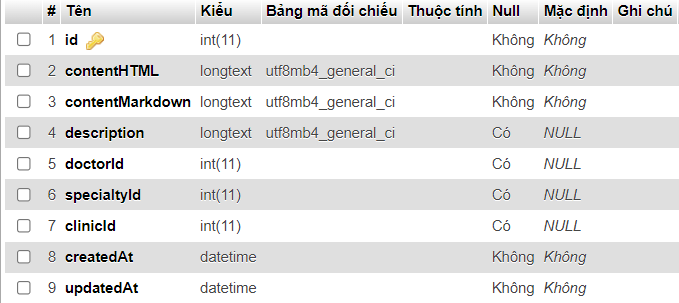
- Tên bảng: doctor\_info

**

*Bảng 3.3: Thông tin chi tiết bác sĩ*

\*) Bảng thông tin về tiểu sử cũng như việc khám và điều trị của bác sĩ.

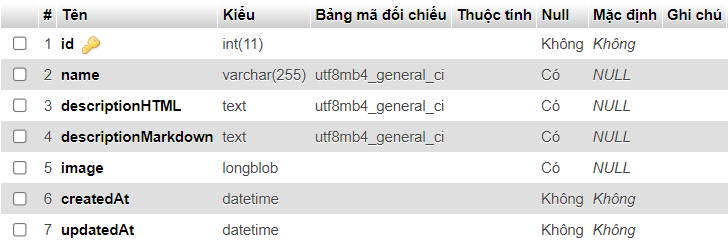
- Tên bảng: markdown

**

*Bảng 3.4: Tiểu sử và công việc khám chữa bệnh của bác sĩ*

\*) Bảng thông tin bài viết về y tế trong nước cũng như quốc tế.

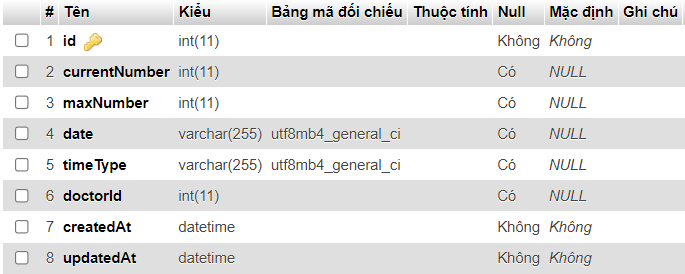
- Tên bảng: new

**

*Bảng 3.5: Bài viết y tế*

\*) Bảng thông tin về những lịch khám đã được bác sĩ sắp xếp để người bệnh có thể đặt lịch.

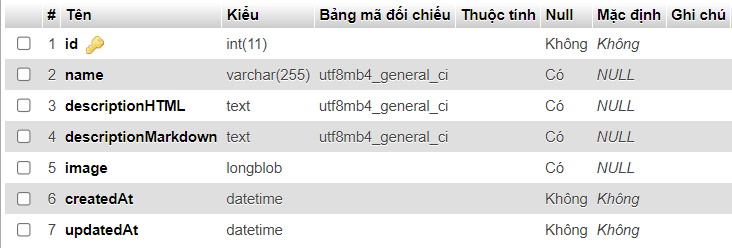
- Tên bảng: schedule

**

*Bảng 3.6: Lịch khám của bác sĩ*

\*) Bảng thông tin mô tả về chuyên khoa.

- Tên bảng: specialty

**

*Bảng 3.7: Thông tin chuyên khoa*

\*) Bảng thông tin cơ bản của người dùng như: họ tên, số điện thoại, địa chỉ,…

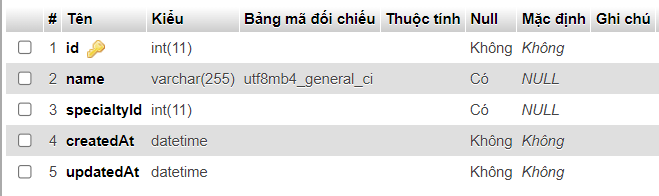
- Tên bảng: user

**

*Bảng 3.8: Thông tin người dùng*

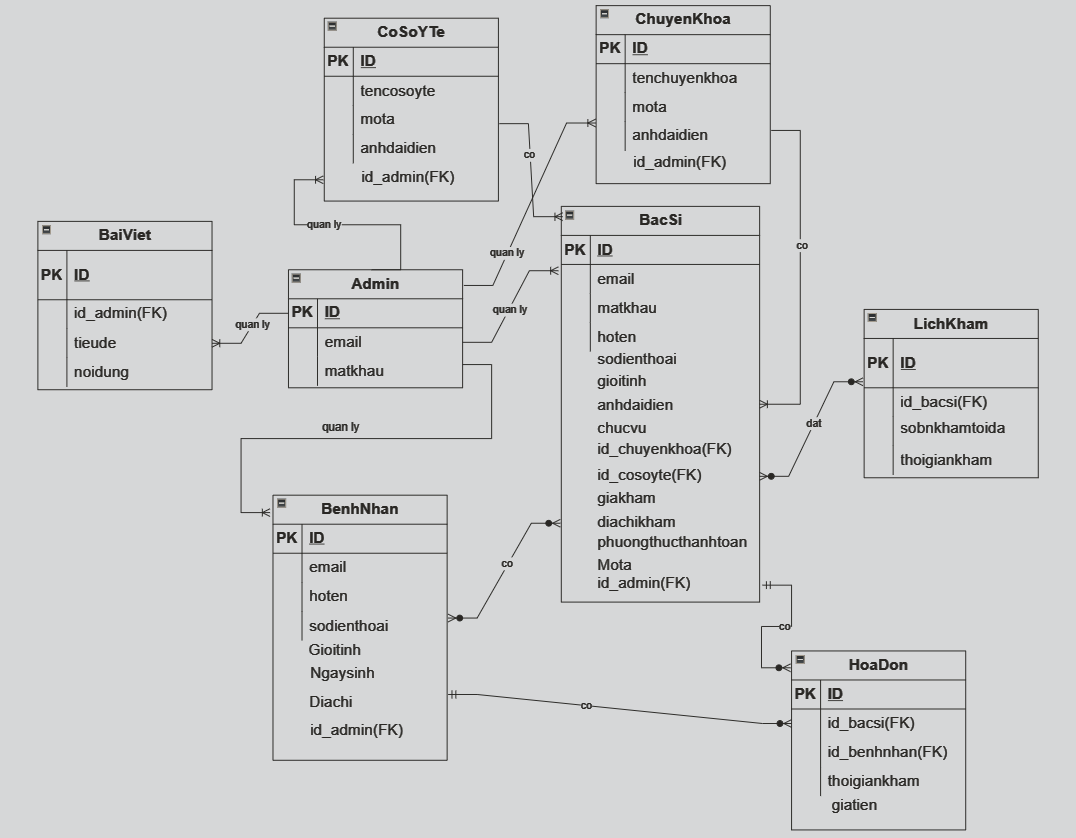
\*) Bảng thông tin về lý do khám phổ biến.

- Tên bảng: reasons

**

*Bảng 3.9: Lý do khám*

**3.4.2 Biểu đồ quan hệ thực thể**



*Hình 3.26: Biểu đồ quan hệ thực thể*

**CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG**

* 1. **Công cụ sử dụng**

**- Visual Studio Code:** Là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và MacOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp, tự hoàn thành mã, snippets và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, thay đổi theme, phím tắt,... Nó miễn phí và là phần mềm mã nguồn mở theo giấy phép MIT.

**- MySQL Server**: MySQL Server là thành phần chính của hệ thống quản lý cơ sở dữ liệu MySQL. Đây là nơi dữ liệu được lưu trữ, xử lý và truy cập thông qua các truy vấn SQL.

**- MySQL Workbench**: MySQL Workbench là một công cụ quản trị cơ sở dữ liệu tích hợp cung cấp giao diện đồ họa để quản lý cơ sở dữ liệu của MySQL. Cung cấp các tính năng như thiết kế cơ sở dữ liệu, thực thi truy vấn SQL, sao lưu, phục hồi dữ liệu và quản lý người dùng.

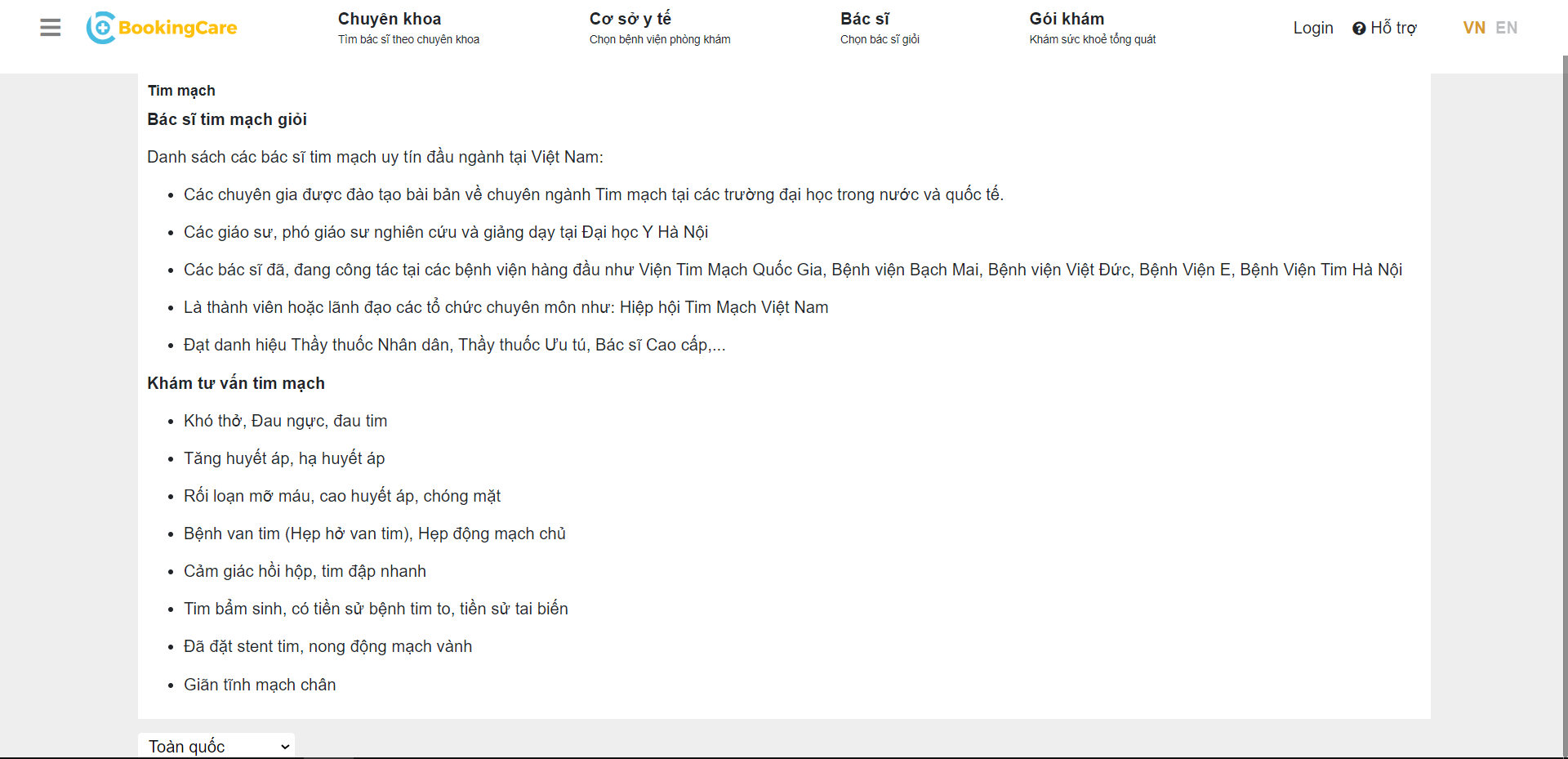
**4.2 Một số giao diện của trang web**



*Hình 4.1: Ảnh trang chủ website*

\*) Giao diện trang chủ

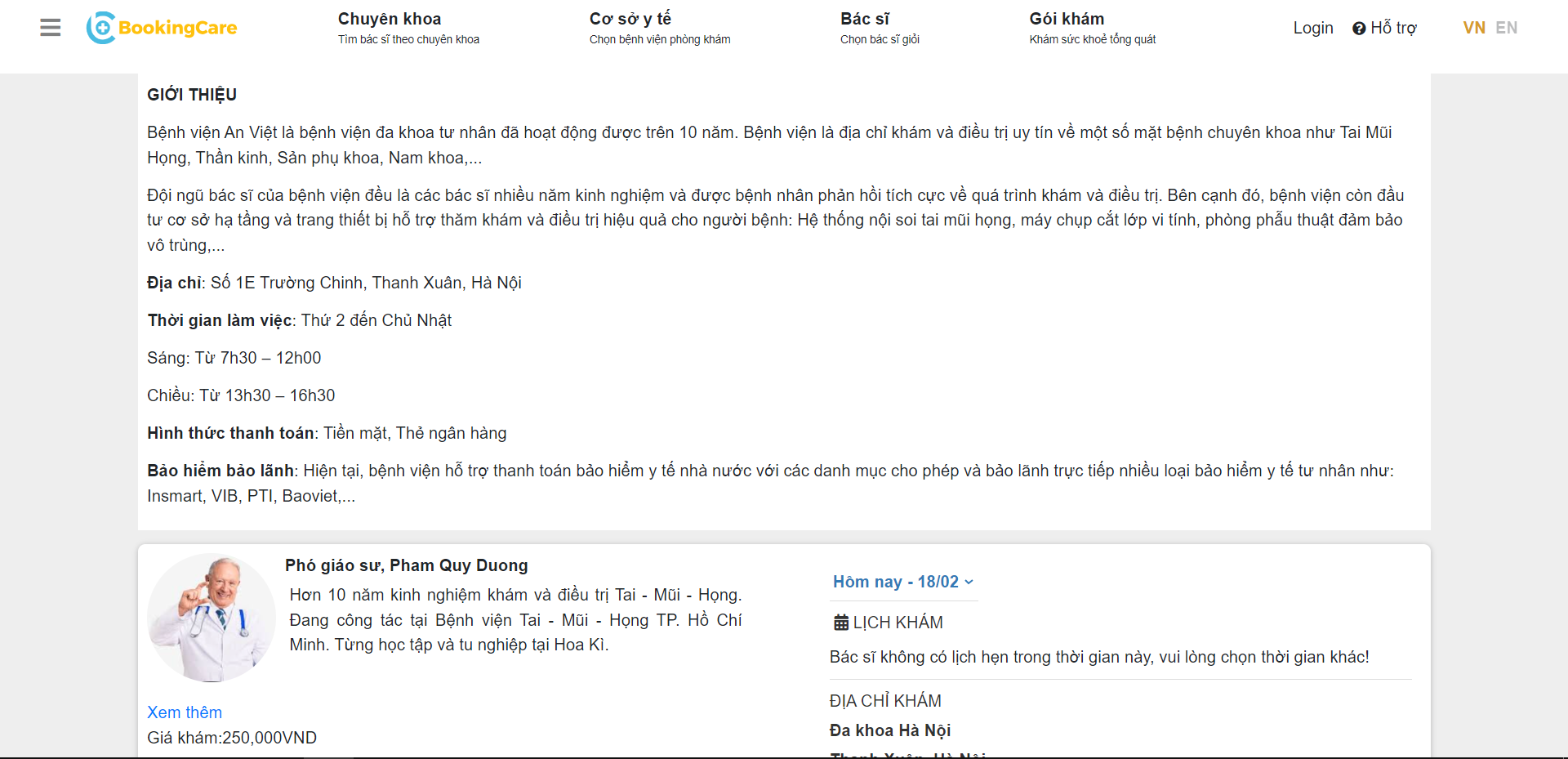
* Trang chủ trang web cho khách hàng gồm các chức năng cơ bản của trang web. Người dùng có thể xem thông tin chuyên khoa, cơ sở y tế, bác sĩ và bài viết từ trang web này.
* Người thực hiện: Khách hàng.



*Hình 4.2: Ảnh màn hình chuyên khoa*

\*) Giao diện trang thông tin chuyên khoa

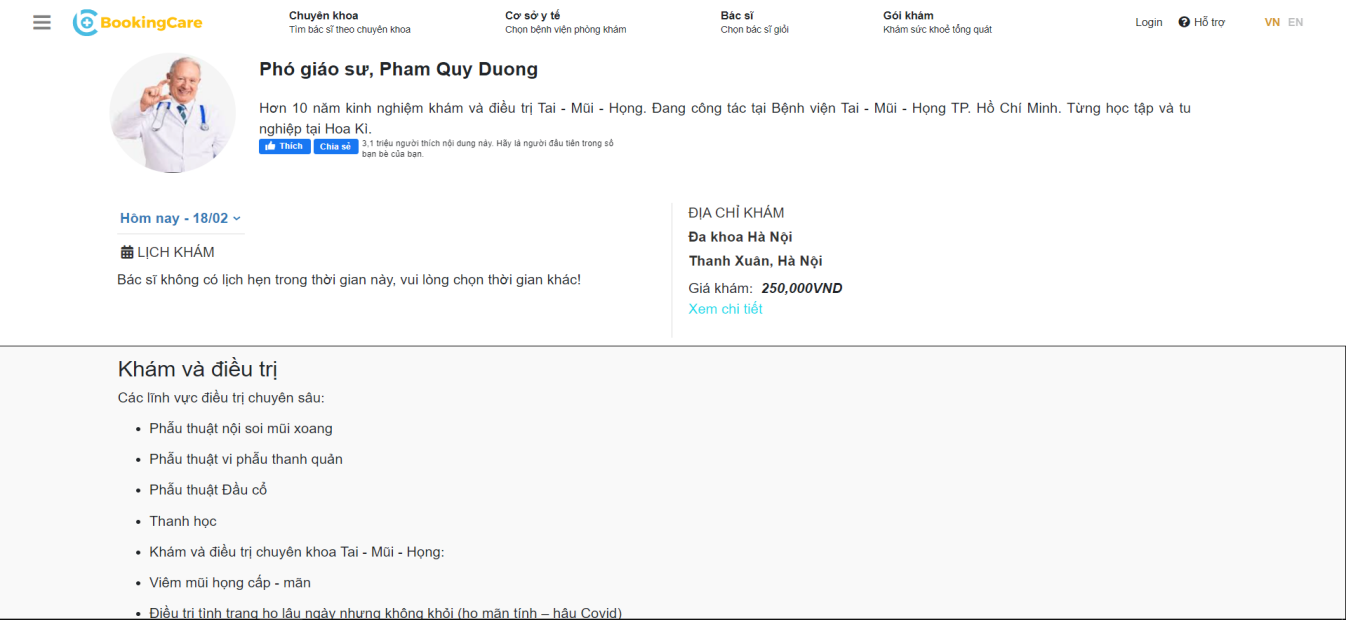
* Trang chuyên khoa chứa thông tin về chuyên khoa và các bác sĩ thuộc chuyên khoa đó. Người dùng có thể xem thông tin chi tiết bác sĩ từ đây. Người dùng kích vào hình ảnh chuyên khoa để vào trang này.
* Người thực hiện: Khách hàng.



*Hình 4.3: Ảnh màn hình cơ sở y tế*

\*) Giao diện trang thông tin cơ sở y tế

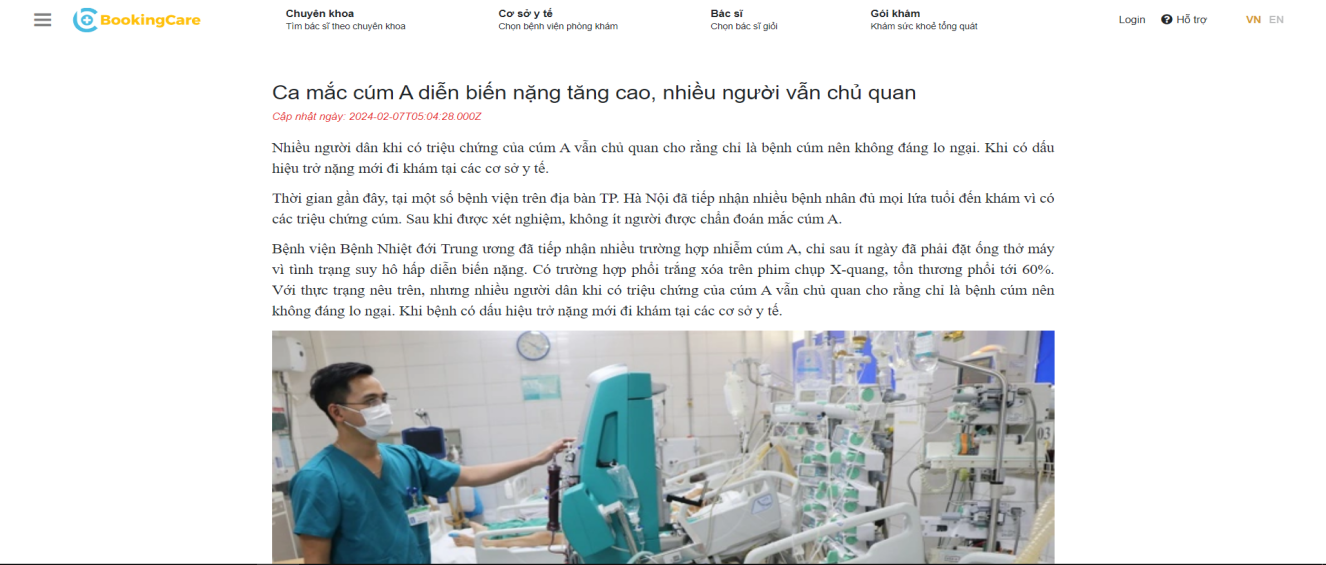
* Trang cơ sở y tế chứa thông tin mô tả về cơ sở y tế và các bác sĩ thuộc cơ sở y tế đó. Người dùng có thể xem thông tin chi tiết bác sĩ từ đây. Người dùng kích vào hình ảnh cơ sở y tế để vào trang này.
* Người thực hiện: Khách hàng.



*Hình 4.4: Ảnh màn hình bác sĩ*

\*) Giao diện chi tiết bác sĩ

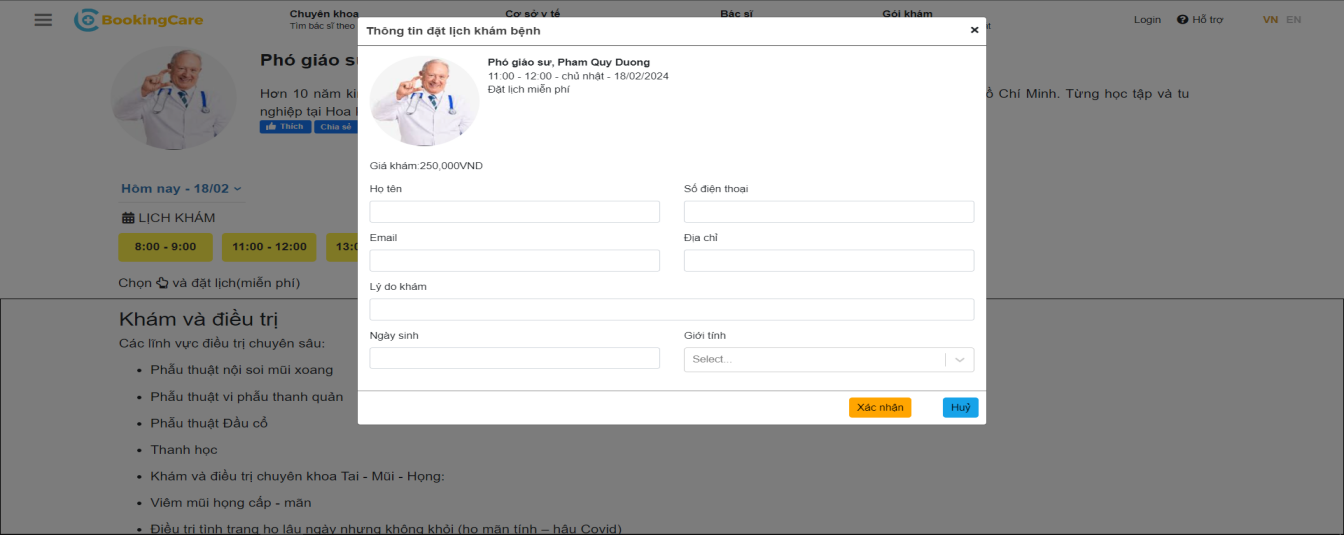
* Trang chi tiết bác sĩ chứa thông tin chi tiết về bác sĩ. Người dùng có thể đặt lịch khám bệnh từ đây. Người dùng kích vào hình ảnh bác sĩ để vào trang này.
* Người thực hiện: Khách hàng.



*Hình 4.5: Ảnh màn hình bài viết*

\*) Giao diện bài viết

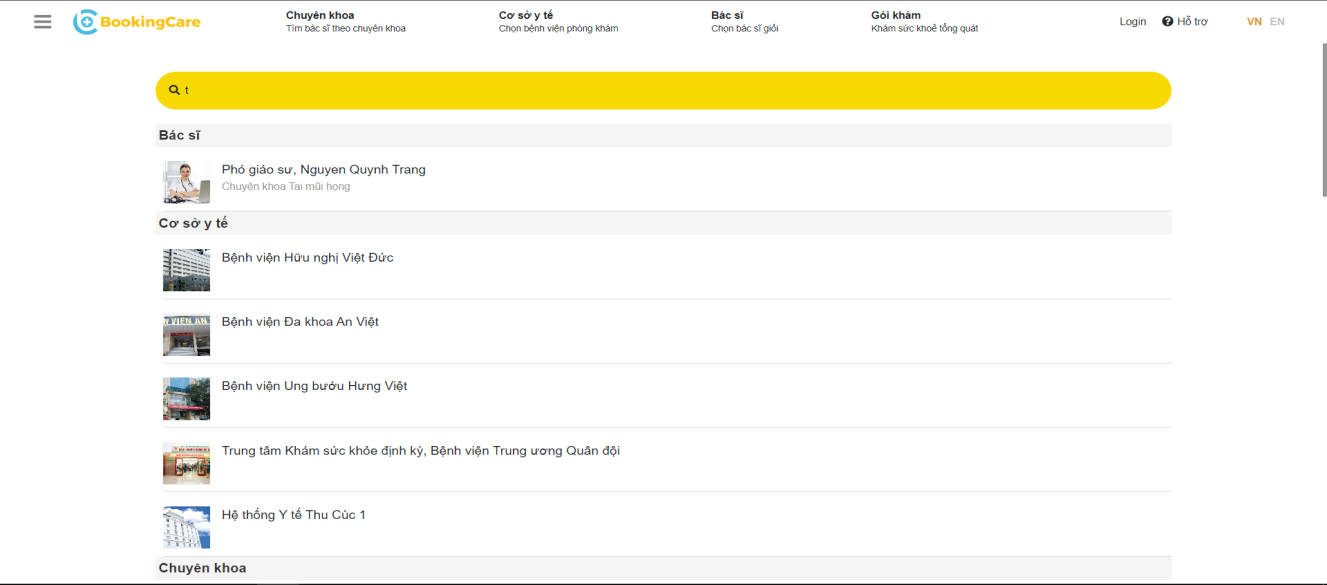
* Trang bài viết chứa thông tin về các tin tức đã xảy ra liên quan đến vấn đề y tế. Người dùng có thể xem thông tin bài viết để biết thêm. Người dùng kích vào hình ảnh bài viết để vào trang này.
* Người thực hiện: Khách hàng.



*Hình 4.6: Ảnh màn hình đặt lịch khám bệnh*

\*) Giao diện đặt lịch khám bệnh

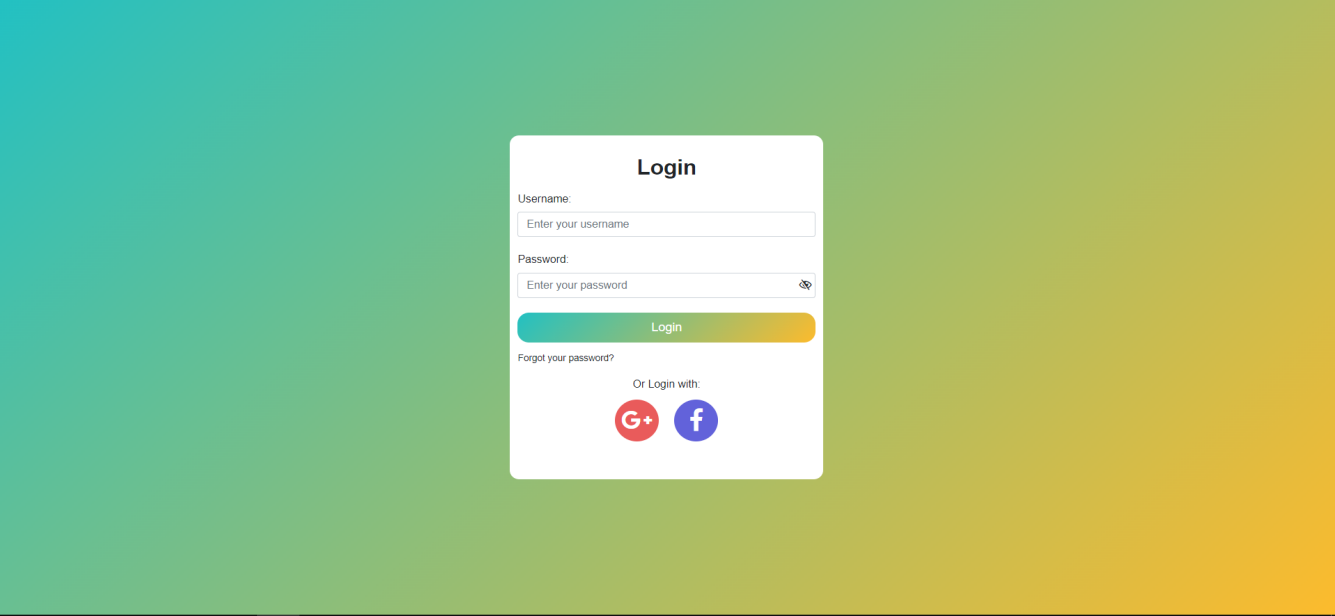
* Trang đặt lịch khám bệnh cho phép người bệnh có thể nhập thông tin đặt lịch khám bệnh. Người dùng nhấn chọn thời gian chỗ lịch khám của bác sĩ để mở trang này.
* Người thực hiện: Khách hàng.



*Hình 4.7: Ảnh màn hình kết quả tìm kiếm*

\*) Giao diện kết quả tìm kiếm

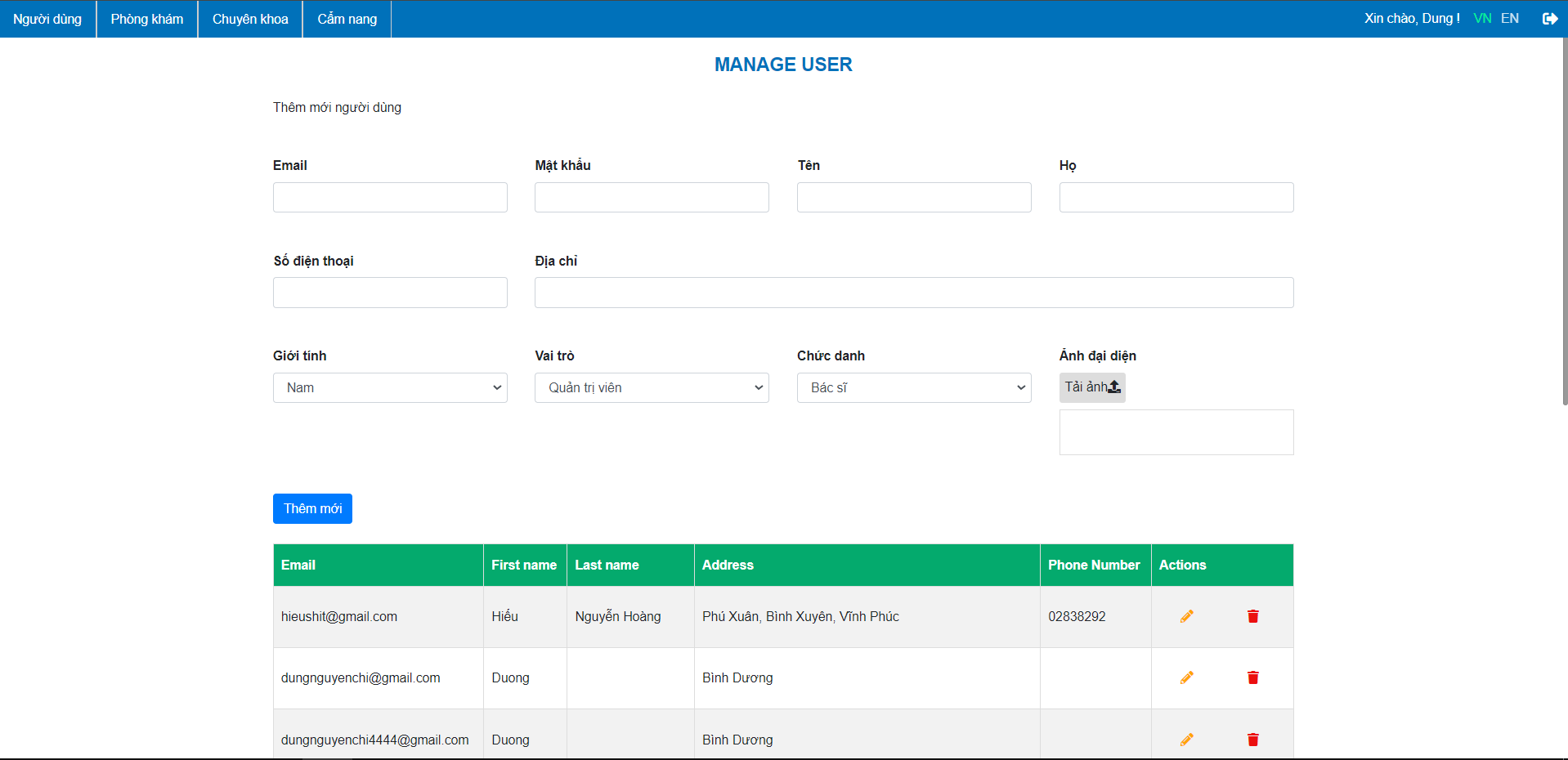
* Chứa thông tin về từ khoá mà người dùng nhập. Người dùng có thể tìm kiếm bác sĩ, chuyên khoa, cơ sở y tế hay lý do khám. Người dùng nhập từ khoá vào ô tìm kiếm rồi nhấn Enter để xem kết quả hiển thị.
* Người thực hiện: Khách hàng.



*Hình 4.8: Ảnh màn hình đăng nhập của bác sĩ và người quản trị*

\*) Giao diện đăng nhập

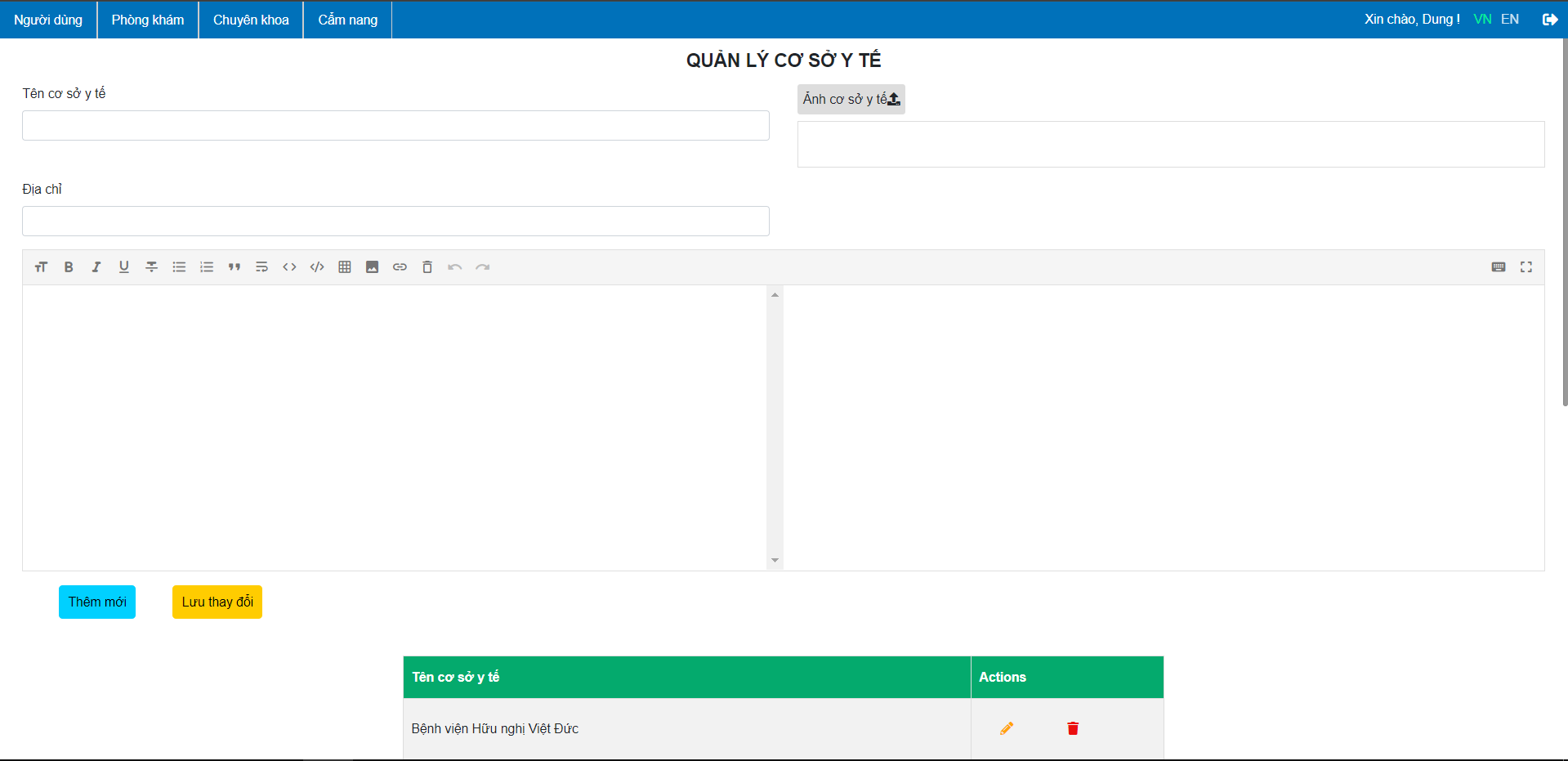
* Trang đăng nhập của người quản trị và bác sĩ. Người quản trị và bác sĩ nhập email và password để đăng nhập vào trang quản trị của mình.
* Người thực hiện: Người quản trị hoặc bác sĩ.



*Hình 4.9: Ảnh màn hình quản lý người dùng*

\*) Giao diện quản lý người dùng

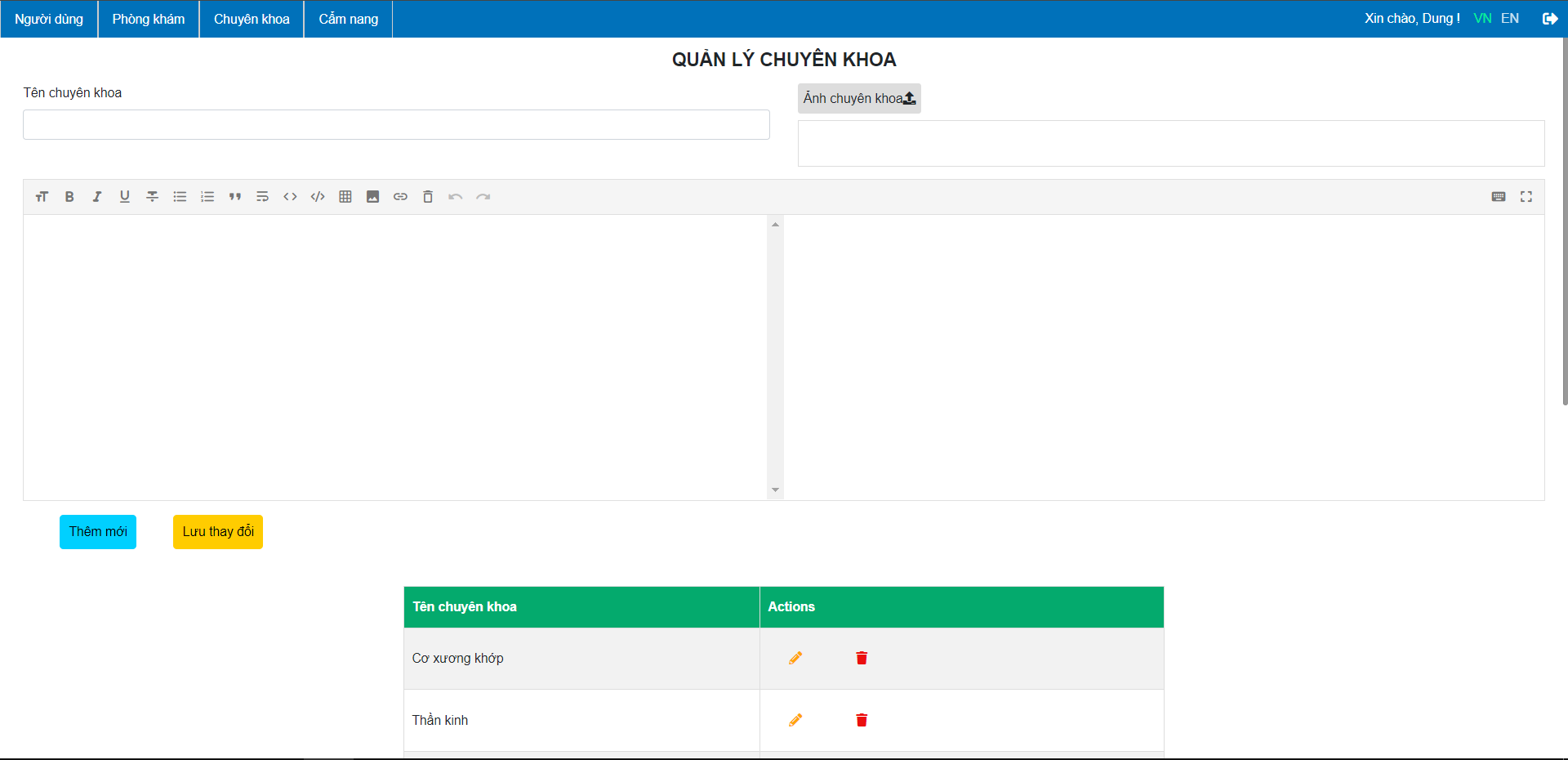
* Trang quản lý người dùng của người quản trị. Người quản trị có thể thêm, sửa, xoá thông tin cơ bản của người dùng tại đây.
* Người thực hiện: Người quản trị.



*Hình 4.10: Ảnh màn hình quản lý thông tin mô tả cơ sở y tế*

\*) Giao diện quản lý thông tin mô tả cơ sở y tế

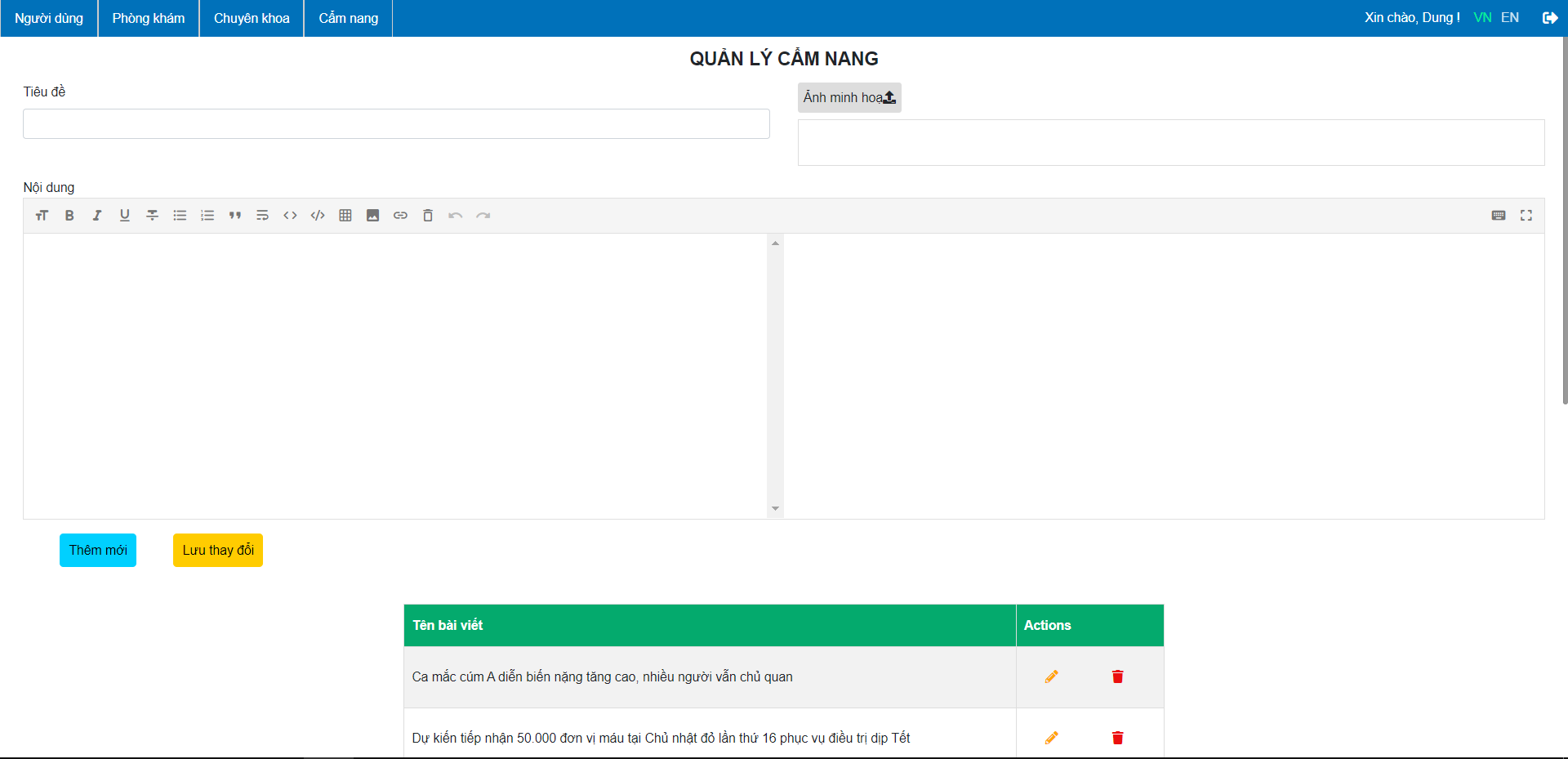
* Trang quản lý thông tin mô tả cơ sở y tế của người quản trị. Người quản trị có thể thêm, sửa, thông tin của cơ sở y tế tại đây.
* Người thực hiện: Người quản trị.



*Hình 4.11: Ảnh màn hình quản lý chuyên khoa*

\*) Giao diện quản lý chuyên khoa

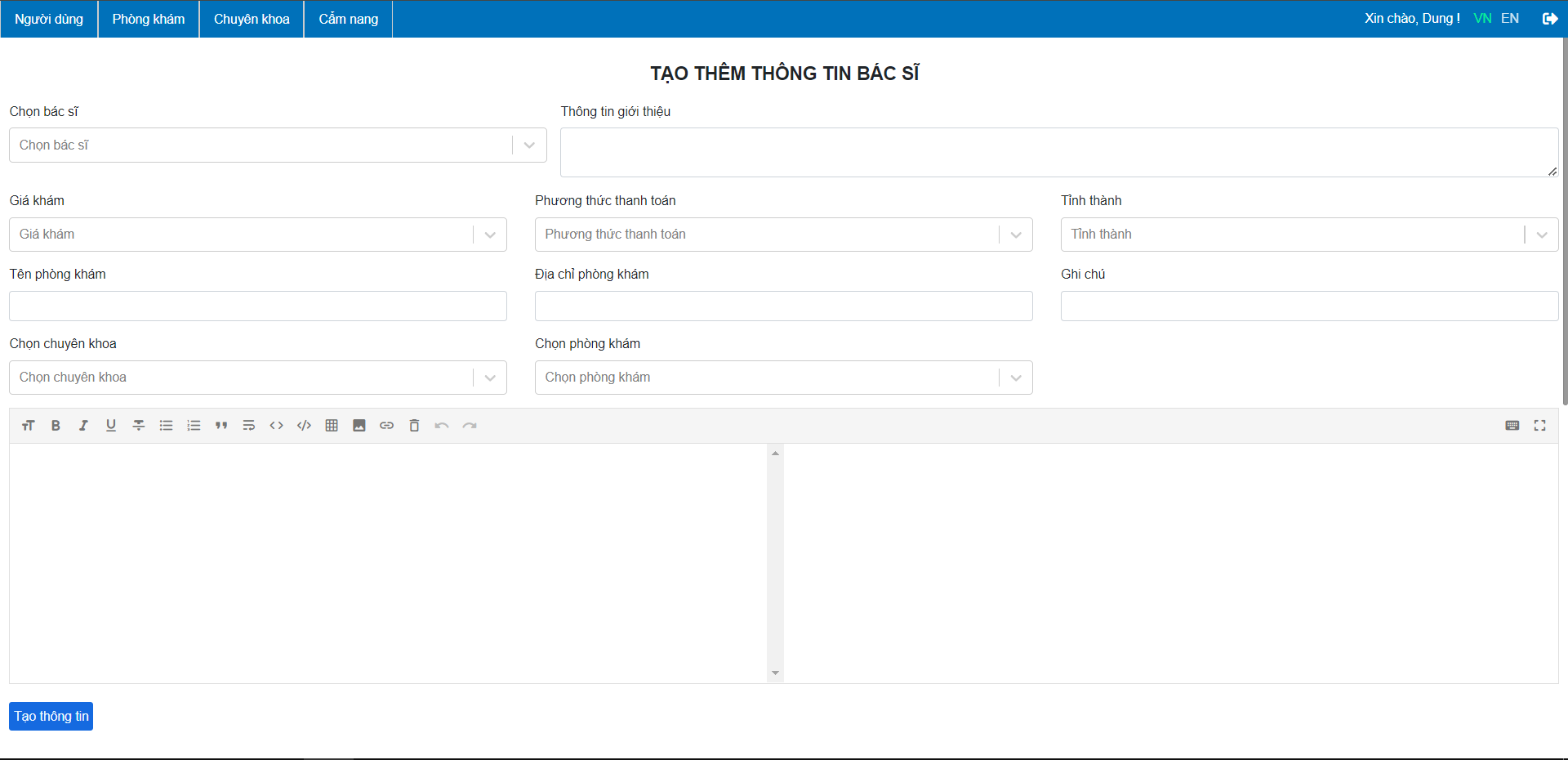
* Trang quản lý chuyên khoa của người quản trị. Người quản trị có thể thêm, sửa, xoá thông tin của chuyên khoa tại đây.
* Người thực hiện: Người quản trị.



*Hình 4.12: Ảnh màn hình quản lý bài viết*

\*) Giao diện quản lý bài viết

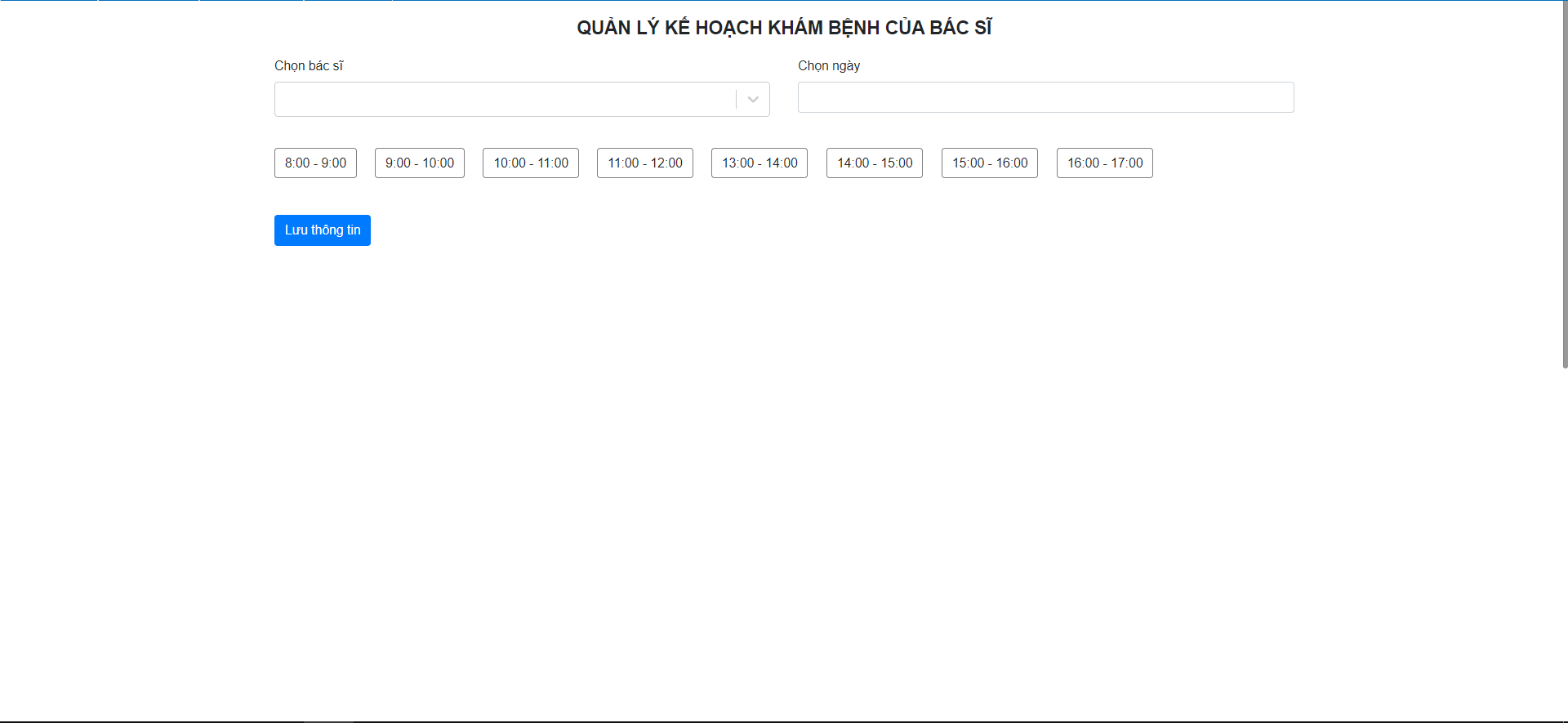
* Trang quản lý bài viết của người quản trị. Người quản trị có thể thêm, sửa, xoá thông tin của bài viết tại đây.
* Người thực hiện: Người quản trị.



*Hình 4.13: Ảnh màn hình quản lý thông tin chi tiết bác sĩ*

\*) Giao diện quản lý thông tin chi tiết bác sĩ

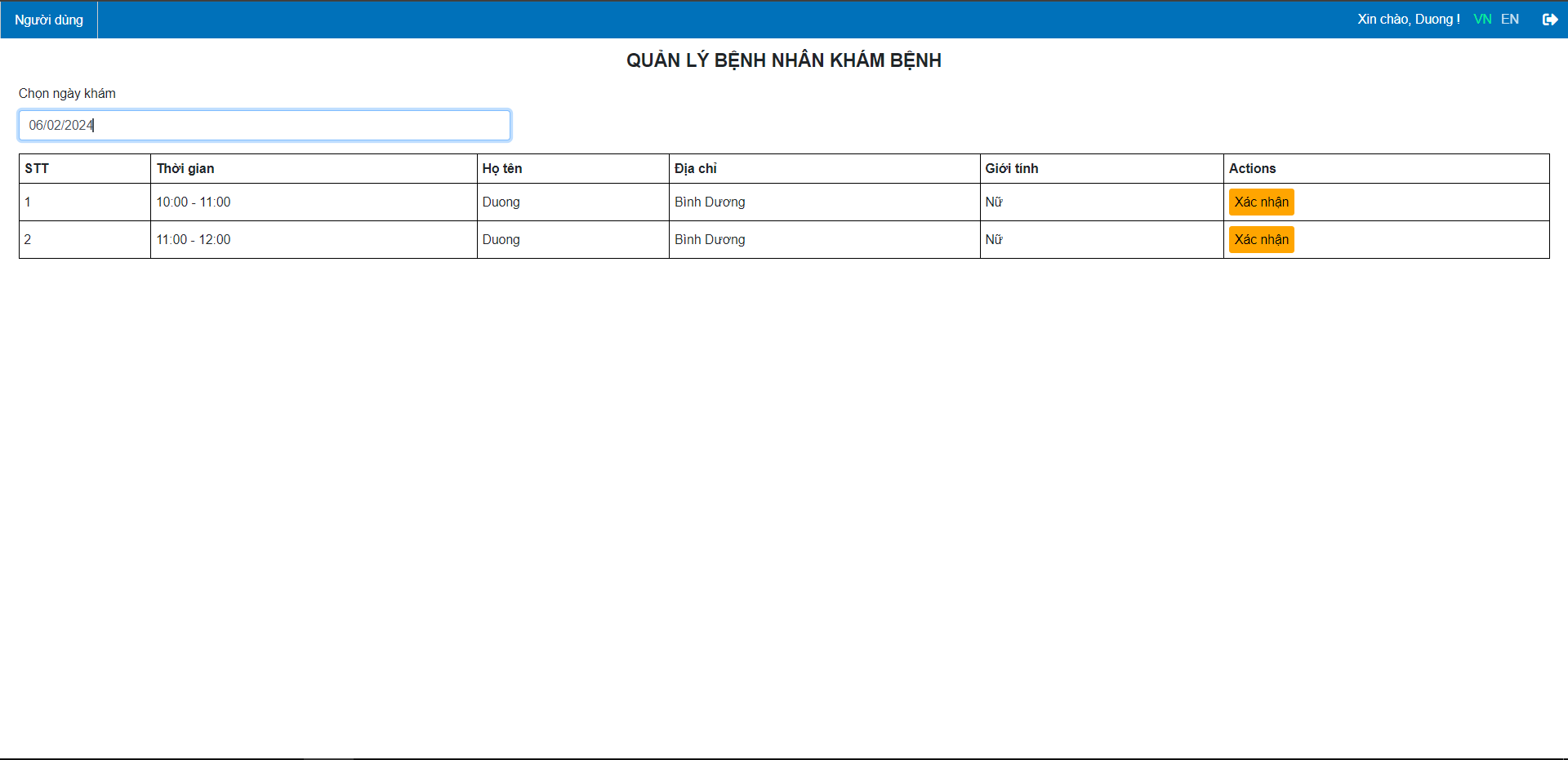
* Trang quản lý thông tin chi tiết bác sĩ của người quản trị. Người quản trị có thể thêm, sửa thông tin của chi tiết bác sĩ tại đây.
* Người thực hiện: Người quản trị.



*Hình 4.14: Ảnh màn hình quản lý kế hoạch khám bệnh của bác sĩ*

\*) Giao diện quản lý kế hoạch khám bệnh của bác sĩ

* Trang quản lý kế hoạch khám bệnh của bác sĩ của mỗi bác sĩ. Bác sĩ có thể thêm, sửa kế hoạch khám bệnh tại đây.
* Người thực hiện: Bác sĩ.



*Hình 4.15: Ảnh màn hình quản lý bệnh nhân của bác sĩ*

\*) Giao diện quản lý bệnh nhân khám bệnh của bác sĩ

* Trang quản lý bệnh nhân khám bệnh của mỗi bác sĩ. Bác sĩ có thể xem thông tin bệnh nhân đã đặt lịch khám và gửi hoá đơn tại đây.
* Người thực hiện: Bác sĩ.

**KẾT LUẬN**

Qua quá trình làm đề tài em đã học hỏi và lĩnh hội được rất nhiều kiến thức và công nghệ mới như: lập trình web bằng ReactJS và NodeJS, xử lý dữ liệu người dùng, giao thức truyền tải dữ liệu,… Đồng thời nó cũng giúp em tổng quát hoá được những kiến thức đã được học trước đây về khảo sát, phân tích và thiết kế hệ thống. Ngoài ra, em cũng có thêm được nhiều kinh nghiệm trong quá trình xây dựng một trang web thực tế.

Sau thời gian thực hiện đề tài, hệ thống mà em xây dựng được có thể đáp ứng được các yêu cầu mà đề tài đặt ra cho một website đặt lịch khám bệnh. Về phần khách hàng: tìm kiếm, xem thông tin mô tả chuyên khoa, bác sĩ,... Về phần người quản trị: có thể thêm, sửa, xóa chuyên khoa, cơ sở y tế, bài viết,… Ngoài ra, website được thiết kế với giao diện thân thiện, dễ thao tác.

Trong quá trình thực hiện đề tài em cũng gặp rất nhiều khó khăn, phần về thời gian thực hiện cũng như phần về kinh nghiệm thực tế còn hạn chế nên website cũng chưa được hoàn chỉnh. Như chưa tích hợp được hình thức thanh toán toán online để quá trình khám chữa bệnh của người bệnh có thể dễ dàng và có nhiều cách lựa chọn hơn, chưa tích hợp được chatbot giúp tư vấn cho người bệnh về thông tin các bác sĩ, chuyên khoa,... Phần tìm kiếm của khách hàng vẫn đang còn chưa được đầy đủ thông tin.

Trên cơ sở những phần đã thực hiện được và những hạn chế còn tồn tại thì đề tài này có các hướng phát triển như: Có thể tích hợp thêm chức năng khám bệnh trực tuyến, giúp bệnh nhân có thể chẩn đoán bệnh tình tại nhà bớt thời gian đi lại và thêm các gói khám chữa bệnh, xét nghiệm để khách hàng có thể kiêm tra các vấn đề sức khoẻ dễ dàng và nhanh chóng.

**TÀI LIỆU THAM KHẢO**

Sách, giáo trình:

1. Nguyễn Văn Hiếu (2020), “*Lập trình web với ReactJS*”, Nhà xuất bản Thống Kê.
2. Nguyễn Văn Hiếu (2017), “*Lập trình web với NodeJS*”, Nhà xuất bản Thống Kê.
3. Jon Duckett (2011), “HTML and CSS: Design and Build Websites”, *John Wiley & Sons.*

[4] Nguyễn Quốc Bảo (2020), “*Phân tích thiết kế hệ thống thông tin*”, Nhà xuất bản Trẻ.

[5] Vũ Thanh Lộc (2016), “*Quản trị cơ sở dữ liệu MySQL*”, Nhà xuất bản Trẻ.

Website tham khảo:

[6] <https://stackoverflow.com/>

[7] <https://bookingcare.vn/>

[8] <https://www.w3schools.com/>