BỘ GIÁO DỤC VÀ ĐÀO TẠO

**ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

------🙚🙘🕮🙚🙘------

**NIÊN LUẬN CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**WEBSITE ĐẶT LỊCH KHÁM BỆNH**

**Người hướng dẫn:**

Nguyễn Minh Trung

**Sinh viên thực hiện:**

Phạm Thanh Phong

**MSSV:** B2014598

**Khóa:** K46

**Cần Thơ, 12/2023**

# Lời cảm ơn

Lời đầu tiên, xin chân thành cảm ơn quý thầy cô của trường Công nghệ Thông tin và Truyền thông – trường Đại học Cần Thơ đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báo trong những năm học vừa qua.

Thứ hai, em xin tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Minh Trung – giảng viên trường Công nghệ Thông tin – trường Công nghệ Thông tin và Truyền thông – trường Đại học Cần Thơ đã tận tình giúp đỡ, góp ý và hướng dẫn kiến thức để em có thể làm được đề tài này.

Tuy hoàn thành đề tài này nhưng đây là lần em tiếp cận với ngôn ngữ mới và công nghệ mới nên không tránh khỏi những thiếu sót, kính mong sự thông cảm của thầy. Cuối cùng em xin kính chúc thầy có nhiều sức khỏe và thành công trong công việc để vững bước trong công việc giảng dạy và nghiên cứu cho thế hệ mai sau.

Em xin chân thành cảm ơn!

Cần Thơ, ngày … tháng 12 năm 2023

Người viết

Phạm Thanh Phong

**Mục Lục**

[Lời cảm ơn 2](#_Toc146657930)

[PHẦN GIỚI THIỆU 5](#_Toc146657931)

[I. Đặt vấn đề 5](#_Toc146657932)

[II. Mục tiêu đề tài 5](#_Toc146657933)

[PHẦN NỘI DUNG 6](#_Toc146657934)

[CHƯƠNG 1: ĐẶC TẢ YÊU CẦU 6](#_Toc146657935)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 7](#_Toc146657936)

[1. NODE.JS 7](#_Toc146657937)

[2. EXPRESS.JS 7](#_Toc146657938)

[3. REACT.JS 9](#_Toc146657939)

[4. MY SQL 9](#_Toc146657940)

[5. SEQUELIZE 10](#_Toc146657941)

[CHƯƠNG 3: THIẾT KẾ GIẢI PHÁP 11](#_Toc146657942)

[1. Sơ đồ use case 11](#_Toc146657943)

[1.1 Sơ đồ usecase tổng quát 11](#_Toc146657944)

[1.2 Sơ đồ usecase khách vãng lai 12](#_Toc146657945)

[1.3 Sơ đồ usecase khách thành viên là thành viên 13](#_Toc146657946)

[1.4 Sơ đồ usecase bác sĩ 14](#_Toc146657947)

[1.5 Sơ đồ usecase quản trị viên 15](#_Toc146657948)

[2. Thiết kế cơ sở dữ liệu 15](#_Toc146657949)

[2.1. Bảng User 15](#_Toc146657950)

[2.2 Bảng doctor\_infor 16](#_Toc146657951)

[2.3 Bảng Clinic 16](#_Toc146657952)

[2.4 Bảng Specialty 16](#_Toc146657953)

[2.5 Bảng Booking 17](#_Toc146657954)

[2.6 Bảng Schedule 17](#_Toc146657955)

[2.7 Bảng History 17](#_Toc146657956)

[2.8 Bảng doctor\_clicnic\_specialty 17](#_Toc146657957)

[2.9 Bảng allCode 18](#_Toc146657958)

[2.10 Bảng markdown 18](#_Toc146657959)

**Mục lục hình**

[Hình 1: Sơ đồ use case tổng quát 11](#_Toc146462739)

[Hình 2: Sơ đồ use case khách vãng lai 12](#_Toc146462740)

[Hình 3: Sơ đồ use case khách thành viên 13](#_Toc146462741)

[Hình 4: Sơ đồ use case bác sĩ 14](#_Toc146462742)

[Hình 5: Sơ đồ use case quản trị viên 15](#_Toc146462743)

# PHẦN GIỚI THIỆU

## I. Đặt vấn đề

Trong thời đại ngày nay, sự tiện lợi và hiệu quả của hệ thống đặt lịch khám bệnh trực tuyến đã trở thành một phần quan trọng trong việc thăm khám bệnh của con người. Việc đặt lịch trực tuyến không chỉ giúp tiết kiệm thời gian mà còn tối ưu hóa sự sắp xếp và quản lí việc khám bênh cho bệnh nhân tại các cơ sở y tế và bệnh viện.

Tuy nhiên, sự phổ biến và sự phát triển của công nghệ thông tin, cùng với nhu cầu về sức khỏe của con người ngày càng tăng, hệ thống đặt lịch khám bệnh cũng phải đối diện với nhiều khó khăn với một số vấn đề quan trọng cần phải được xem xét. Một số vấn đề như: Bảo mật thông tin bệnh nhân, tính khả dụng và sẵn sàng thích ứng với nhiều sự thay đổi, khả năng tương tác và các thông tin cần phải chi tiết và dễ hiểu, khả năng tích hợp và quản lý dữ liệu, hỗ trợ khả năng ngôn ngữ và khả năng truy cập.

Vì vậy, trong bối cảnh công nghệ thông tin phát triển và nhu cầu về sức khỏe ngày càng tăng, việc tối ưu hóa và đảm bảo tính hiệu quả, bảo mật và các tiện ích của website quản lý đặt lịch khám bệnh là một thách thức trong thời đại này.

## II. Mục tiêu đề tài

Nhằm cung cấp cho người dùng một hệ thống website tiện lợi trong việc thăm khám sức khỏe của người bệnh, có tính bảo mật cao, uy tín và chất lượng. Ngoài ra việc tạo ra hệ thống website này, giúp cho việc thực hiện giảng cách xã hội theo nghị định của chính phủ trong mùa dịch Covid – 19 và trong những giai đoạn về sau. Ngoài ra, nó còn giúp cho việc thăm khám sức khỏe trở nên hiệu quả và hiệu suất khám bệnh sẽ tăng lên góp phần tạo ra một xã hội không bệnh tật.

# PHẦN NỘI DUNG

## CHƯƠNG 1: ĐẶC TẢ YÊU CẦU

Hệ thống website đặt lịch khám bệnh là một hệ thống website hỗ trợ việc đặt lịch khám bệnh thay vì phải đến bệnh viện chờ bốc số để khám bệnh.

* Đối với khách hàng:

+ Trang đăng nhập / đăng ký: người dùng đăng ký để được cấp tài khoản để trở thành thành viên của hệ thống và đăng nhập để đặt lịch khám trên hệ thống.

+ Xem thông tin: Xem thông tin về phòng khám, chuyên khoa và bác sĩ.

+ Xem cẩm nang: Xem các cẩm nang của bác sĩ về một số cách trị bệnh đơn giản nhưng có hiệu quả.

+ Đặt lịch khám bệnh: người dùng có thể đặt lịch khám bệnh.

+ Tìm kiếm: tím kiếm bác sĩ, chuyên khoa và phòng khám.

* Đối với bác sĩ:

+ Đăng nhập: mỗi bác sĩ sẽ được cấp một tài khoản để đăng nhập vào hệ thống, thực hiện các chức năng chính của mình.

+ Tạo lịch khám bệnh: cho phép bác sĩ tạo lịch khám bệnh để người dùng có thể đăng ký.

+ Xem lịch khám bệnh của người dùng.

+ Xác nhận khám bệnh thành công.

+ Tạo cẩm nang.

* Đối với người quản trị:

+ Đăng nhập: đăng nhập vào hệ thống bằng tài khoản của mình.

+ Quản lý người dùng hệ thống: thêm, sửa, xóa người dùng.

+ Quản lý chuyên khoa: thêm, sửa, xóa chuyên khoa.

+ Quản lý phòng khám: thêm, sửa, xóa phòng khám.

+ Quản lý cẩm nang: thêm, sửa, xóa cẩm nang.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### NODE.JS

Hình : Công nghệ node.js

Node.js là môi trường runtime mã nguồn mở đa nền tảng, được sử dụng để phát triển các ứng dụng mạng và ứng dụng client - server. Các ứng dụng Node.js được viết bằng JavaScript và có thể chạy Node.js runtime trên OS X, Microsoft window và Linux.

Các tính năng của Node.js:

+ Cực kỳ nhanh chóng: xây dựng trên công cụ JavaScript V8 của Google Chrome, thư viện Node.js có khả năng xử lý lỗi cực kỳ nhanh.

+ Đơn luồng nhưng có khả năng mở rộng vô cùng cao: dùng mô hình đơn luồng với event loop. Cơ chế của event loop cho phép máy chủ phản hồi non – blocking, cho phép xử lý các yêu cầu đồng thời mà không cần chờ đợi các yêu cầu trước đó hoàn tất. Giúp giảm thời gian xử lý và tăng hiệu suất cho ứng dụng.

+ Sử dụng modules: Node.js có hệ thống modules phong phú, cho phép dễ dàng tái sử dụng mã nguồn và cung cấp các chức năng phổ biến như kết nối cơ sở dữ liệu, xử lý tiệp và mã hóa mật khẩu.

Ưu điểm của Node.js:

+ Node.js là một framework nguồn mở.

+ Sử dụng JavaScript để xây dựng toàn bộ các ứng dụng client – server.

+ Framework gọn nhẹ bao gồm nhiều module tối thiểu. Các module có thể được thêm vào theo nhu cầu sử dụng.

+ không đồng bộ theo mặc định, nhờ vậy hoạt động nhanh hơn các framework khác.

+ Framework đa nền tảng chạy trên Window, Mac OS và Linux.

### A logo on a keyboard Description automatically generatedEXPRESS.JS

Hình : Công nghệ Express.js

Express.js là một framework mạnh mẽ và phổ biến được xây dựng trên nền tảng Node.js. Đơn giản hóa quá trình phát triển ứng dụng web bằng các cung cấp các công cụ, thư viện cần thiết để xử lý các yêu cầu và phản hồi HTTP một cách dễ dàng.

Do được xây dựng trên cơ sở Node.js, cho phép sử dụng JavaScript để phát triển trên cả Backend và Frontend. Express.js cũng cho phép ta kết hợp nhiều thư viện và công nghệ khác nhau để xây dựng web động.

Các tính năng của Express.js:

+ Phát triển máy chủ nhanh hơn: tối ưu hóa cú pháp cung cấp phương thức và hàm tiện ích để xây dựng các tác vụ phổ biến trong lập trình.

+ Định tuyến (Routing): cung cấp hệ thống định tuyến mạnh mẽ, cho phép định tuyến các URL và xử lý các yêu cầu HTTP, GET, POST, PUT, DELETE.

+ Middleware: Express hỗ trợ middleware, cho phép thêm các chức năng trung gian vào quy trình xử lý yêu cầu và phản hồi. Giúp xác thực người dùng, ghi log, xử lý lỗi, nén dữ liệu và thực hiện nhiều tác vụ khác một cách linh hoạt.

+ Cấu hình môi trường: Express.js cung cấp cách để cấu hình môi trường phát triển và môi trường sản xuất.

+ Xử lý lỗi: Express.js cung cấp cơ chế xử lý lỗi, giúp kiểm soát và xử lý lỗi trong quá trình xử lý yêu cầu. Có thể tạo ra các middleware để sử lý lỗi và phản hồi với các thông báo.

Ưu điểm của Express.js:

+ Đơn giản và dễ sử dụng: cú pháp đơn giản và dễ hiểu.

+ Linh hoạt: Không áp đặt cấu trúc cụ thể, cho phép tự do tùy chỉnh và xây dựng xây dựng ứng dụng theo ý muốn.

+ Hỗ trợ middleware: cung cấp hệ thống middleware mạnh mẽ, thực hiện các chức năng xác thực, ghi log, nén và xử lý lỗi một cách linh hoạt.

+ Hiệu suất cao: nền tảng có hiệu suất cao, cho phép xử lý nhanh chóng các yêu cầu đồng thời có khả năng mở rộng tốt.

### A blue background with white text and a symbol Description automatically generatedREACT.JS

Hình : Công nghệ React.js

React.js là một thư viện viết bằng javaScript, dùng để xây dựng giao diện người dùng (UI). Mục đích cốt lõi của React.js không chỉ khiến cho trang web trở nên mượt mà, mà còn phải nhanh, khả năng mở rộng cao và đơn giản.

Đặc trưng của React.js:

+ JSX: nó được đánh giá là sử dụng đơn giản hơn JavaScript và cho phép trích dẫn HTML cũng như sử dụng các cú pháp của HTML để render ra các subcomponent.

+ Redux: nó có thể viết các ứng dụng hoạt động một cách nhất quán, chạy trong môi trường khác nhau và dễ dàng test.

+ Virtual DOM: chúng ta không cần thao tác trực tiếp với DOM trên view mà vẫn phản ánh được sự thay đổi đó.

### A logo with a dolphin Description automatically generatedMY SQL

Hình : Công nhệ mySQL

MySQL là một hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển ưa chuộng cho quá trình phát triển ứng dụng.

Ưu điểm của MySQL:

+ Dễ sử dụng: là một ưu điểm nổi bật khiến cho MySQL trở nên phổ biến.

+ Tính bảo mật cao: Đây là một ưu điểm giúp cho MySQL rất khó bị hacker tấn công.

+ Tốc độ truy vấn và tốc độ phản hồi dữ liệu tương đối nhanh chóng.

+ Khả năng được phát triển và mở rộng dễ dàng.

+ Miễn phí: nên được rất nhiều lập trình viên ưa chuộng.

Cách thức hoạt động của MySQL: Hoạt động dựa trên mô hình Client – Server. Máy chủ MySQL là cốt lõi của MySQL, xử lý toàn bộ các hướng dẫn CSDL hoặc các lệnh. Máy chủ MySQL có sẵn, được xem là một chương trình riêng biệt để có thể sử dụng trong môi trường mạng client – server. Còn được xem là một như viện có thể được liên kết với các ứng dụng riêng biệt.

### A logo with blue and white text Description automatically generatedSEQUELIZE

Hình : Công nghệ Sequelize

Sequelize là một Object relational mapping (ORM) dành cho các cơ sở dữ liệu, giúp thao tác với cơ sở dữ liệu một cách dễ dàng.

Các tính năng chính của Sequelize:

+ model: tạo một model ánh xạ một – một với bảng trong cơ sở dữ liệu.

+ DataTypes cho từng Attribute.

+ Tạo bảng trong cơ sở dữ liệu: khi định nghĩa xong các model thì có thể tạo các bảng tương ứng bằng cách chạy phương thức async.

+ Xóa bảng: chỉ cần gọi đến phương thức drop và trả về một promise.

+ Tạo model và migration thông qua sequelize-cli.

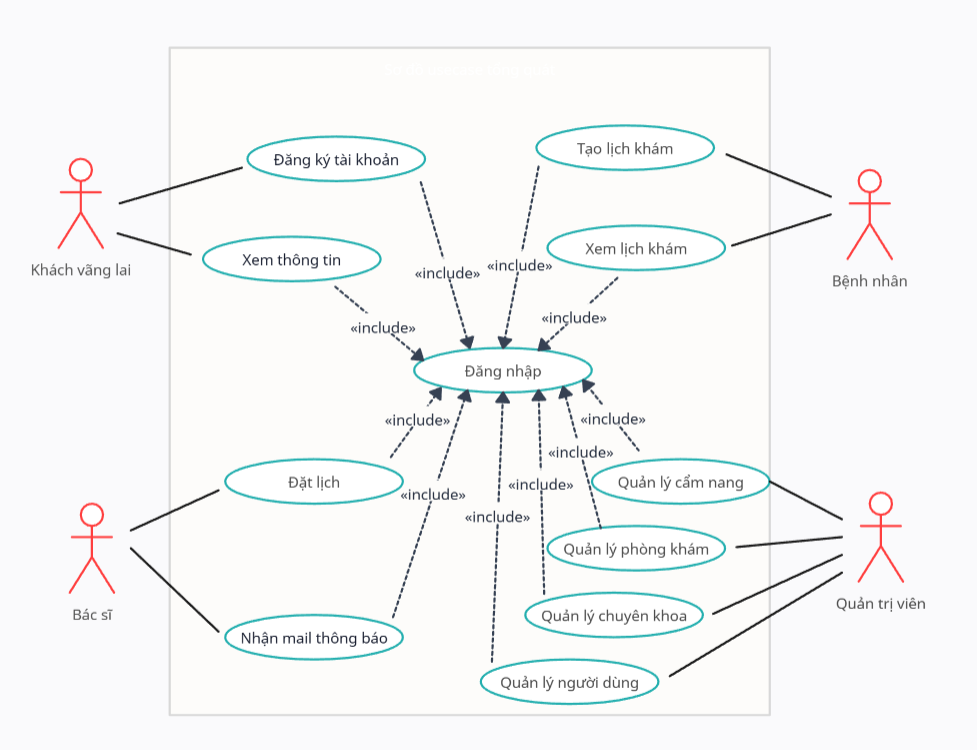
+ Hổ trợ các mối quan hệ: one to one, one to many, …

+ Truy vấn cơ sở dữ liệu: cung cấp đầy đủ các chức năng để truy vấn cơ sở dữ liệu: tạo dữ liệu, lấy, cập nhật và xóa dữ liệu.

## CHƯƠNG 3: THIẾT KẾ GIẢI PHÁP

### 1. Sơ đồ use case

#### 1.1 Sơ đồ usecase tổng quát



Hình : Sơ đồ use case tổng quát

Sơ đồ use case tổn quát được thể hiện như trong *Hình 6,* mô tả cái nhìn tổng quát toàn bộ hệ thống quản lý đặt lịch khám bệnh với những nhóm người dùng của hệ thống như: Khách vãng lai, khách thành viên (bệnh nhân), bác sĩ và quản trị viên. Bên cạnh đó còn thể hiện các chức năng cơ bản của người dùng.

#### 1.2 Sơ đồ usecase khách vãng lai

A diagram of a person with text

Description automatically generated

Hình : Sơ đồ use case khách vãng lai

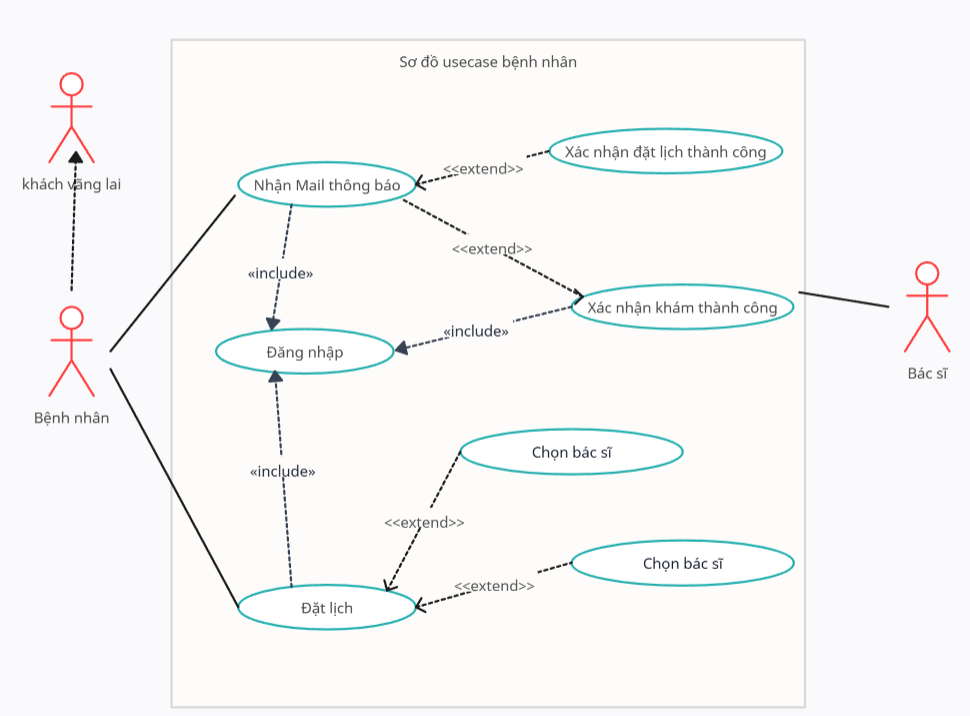
Khách vãng lai là những bệnh nhân không cần đăng nhập vào hệ thống, actor này sẽ có những chức năng được thể hiện như *Hình 7*, cụ thể như sau:

+ Về chức năng đăng ký: cho phép khách vãng lai đăng ký tài khoản để trở thành khách thành viên của hệ thống.

+ Về chức năng xem thông tin: cho phép khách vãng lai có thể xem thông tin về các chuyên khoa, thông tin bác sĩ, xem thông tin phòng khám, và các chuyên mục cẩm nang.

+ Về chức năng tìm kiếm: cho phép actor này tìm kiếm chuyên khoa, tìm kiếm bác sĩ, tìm kiếm phòng khám.

#### 1.3 Sơ đồ usecase khách thành viên là thành viên



Hình : Sơ đồ use case khách thành viên

Khi khách hàng đăng nhập vào hệ thống sẽ được thực hiện các chức năng của mình như *Hình 8*:

+ Về chức năng đăng nhập: cho phép actor này đăng nhập vào hệ thống bằng tài khoản và mật khẩu của mình.

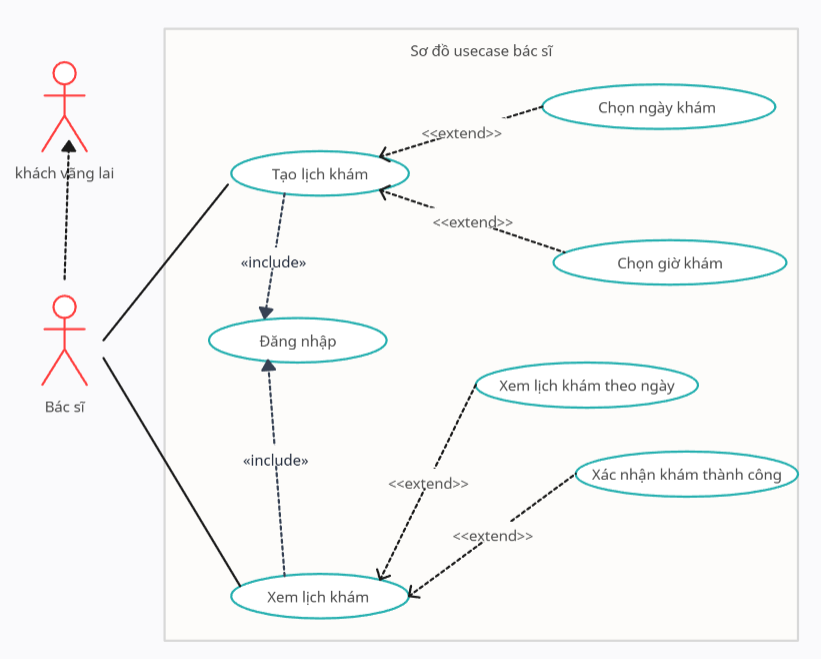
+ Về chức năng đăng xuất: cho phép actor này đăng xuất khỏi hệ thống.

+ Về chức năng đặt lịch: cho phép actor chọn bác sĩ, ngày khám và giờ khám để đặt lịch khám bệnh.

+ Nhận mail thông báo đặt lịch thành công: khách hàng sau khi đặt lịch sẽ được gửi mail xác nhân đến với khách hàng. Nếu không xác nhận mail thì sẽ không hoàn thành đặt lịch.

+ Nhận mail thông báo hoàn thành khám bệnh: Sau khi khám xong, bác sĩ sẽ xác nhận hoàn tất khám và người dùng nhận được mail trong đó có thông tin bác sĩ khám, tình trạng bệnh và đơn thuốc.

#### 1.4 Sơ đồ usecase bác sĩ



Hình : Sơ đồ use case bác sĩ

Khi bác sĩ đăng nhập vào hệ thống sẽ được thực hiện các chức năng như *Hình 9* như sau:

+ Về chức năng đăng nhập: cho phép actor đăng nhập vào hệ thống bằng tài khoản và mật khẩu đucợ cấp.

+ Tạo lịch khám bệnh: Cho phép bác sĩ tạo lịch khám bệnh theo ngày, theo giờ.

+ Về chức năng xem lịch khám: cho phép actor xem lịch khám bệnh của mình.

#### 1.5 Sơ đồ usecase quản trị viên

A diagram of a network

Description automatically generated

Hình : Sơ đồ use case quản trị viên

Người quản trị cần phải đăng nhập để thực hiện một số chức năng như Hình 5:

+ Về chức năng đăng nhập: cho phép actor đăng nhập vào hệ thống và thực hiện các chức năng của mình.

+ về chức năng quản lý khách hàng: cho phép actor quản lý khách hàng của hệ thống. Ngoài ra, actor này còn có chức năng thêm, sửa, xóa khách hàng của hệ thống.

+ Về chức năng quản lý bác sĩ: cho phép actor quản lý bác sĩ trong hệ thống. Ngoài ra, actor còn có chức năng thêm, sửa, xóa bác sĩ trong hệ thống.

+ Về chức năng quản lý chuyên khoa: cho phép actor quản lý các chuyên khoa. Ngoài ra, actor này còn có thể thêm, sửa, xóa các chuyên khoa.

### 2. Xây dựng bảng thực thể liên kết

#### 2.1. Bảng User

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | integer | Primary key | id của người dùng |
| email | String |  | Email người dùng |
| password | String |  | Mật khẩu người dùng |
| firstName | String |  | Họ tên điệm |
| lastName | String |  | Tên người dùng |
| address | Text |  | Địa chỉ người dùng |
| phoneNumber | String |  | Số điện thoại |
| roleId | Boolean |  | Phân quyền người dùng |
| position | String |  | Chức vị người dùng |
| image | blob |  | ảnh đại diện người dùng |

#### 2.2 Bảng doctor\_infor

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | integer | Primary key | Id bác sĩ |
| doctorId | String |  | Mã bác sĩ |
| specialtyId | String |  | Mã chuyên khoa |
| clinicId | String |  | Mã phòng máy |
| priceId | Interger |  | Mã giá |
| provinceId | String |  | Mã tỉnh thành |
| paymentId | String |  | Mã loại thanh toán |
| addressClinic | Text |  | Địa chỉ phòng khám |
| nameClinic | Text |  | Tên phòng khám |
| note | Text |  | Ghi chú |
| count | Integer |  | Hạng bác sĩ |

#### 2.3 Bảng Clinic

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Interger | Primary key | id cơ sở y tế |
| clinicId | String |  | Mã cơ sở y tế |
| Name | String |  | Tên cơ sở y tế |
| Address | Text |  | Địa chỉ |
| DescriptionHTML | Text |  | Mô tả dạng HTML |
| DescriptionMarkdown | Text |  | Mô tả dạng markdown |
| image | String |  | Hình ảnh |

#### 2.4 Bảng Specialty

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã chuyên khoa |
| Name | String |  | Tên chuyên khoa |
| DescriptionHTML | Text |  | Mô tả HTML |
| DescriptionMarkdown | Text |  | Mô tả Markdown |
| image | String |  | Hình ảnh |

#### 2.5 Bảng Booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã đặt |
| DoctorId | String |  | Mã bác sĩ |
| PatientId | String |  | Mã bệnh nhân |
| statusId | String |  | Trạng thái khám |
| Date | String |  | Ngày khám |
| timeType | String |  | Giờ khám |
| token | String |  | Mã token |

#### 2.6 Bảng Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã lịch hẹn |
| CurrentNumber | Integer |  | Số lượng bệnh nhân hiện tại |
| MaxNumber | Integer |  | Số lượng bệnh nhân tối đa |
| Date | String |  | Ngày khám |
| timeType | String |  | Giờ khám |
| doctorId | String |  | Mã bác sĩ |

#### 2.7 Bảng History

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã lịch sử khám |
| PatientId | String |  | Mã bệnh nhân |
| DoctorId | String |  | Mã bác sĩ |
| Description | Text |  | Mô tả |
| files | String |  | File |

#### 2.8 Bảng HandBook

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã số |
| DoctorId | String |  | Mã bác sĩ |
| clicnicId | String |  | Mã cơ sở y tế |
| specialtyId | String |  | Mã chuyên khoa |

#### 2.9 Bảng allCode

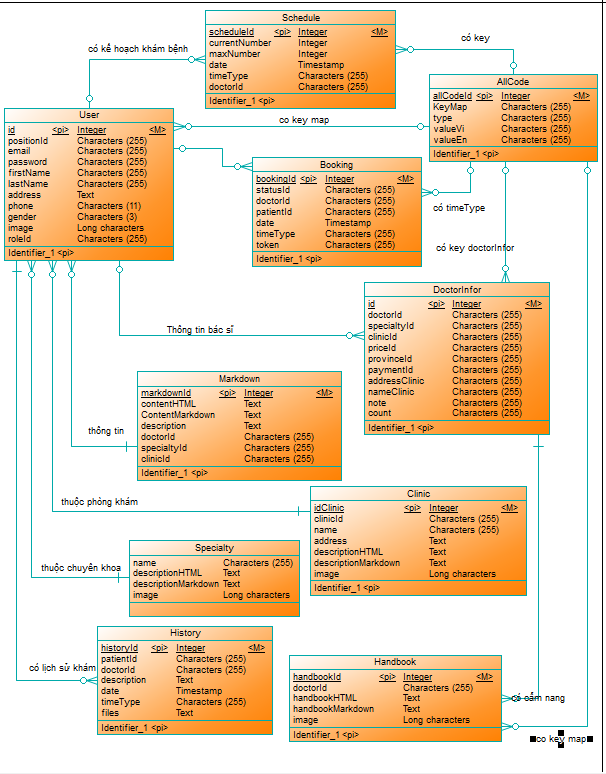
|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã số |
| Keymap | String |  | Diễn giải |
| Type | String |  | Mã loại |
| valueEn | String |  | Tiếng anh |
| valueVi | String |  | Tiếng việt |

#### 2.10 Bảng markdown

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| Id | Integer | Primary key | Mã Id |
| DoctorId | String |  | Mã bác sĩ |
| clicnicId | String |  | Mã phòng khám |
| specialtyId | String |  | Mã chuyên khoa |
| contentHTML | String |  | Nội dung html |
| contentMarkdown | String |  | Nội dung markdown |

### 3. Thiết kế cơ sở dữ liệu

#### 3.1 Mô hình mức quan niệm CDM



# CHƯƠNG 4: CÀI ĐẶT GIẢI PHÁP

## 1. Cài đặt giải pháp

### 1.1 Yêu cầu phầm mềm

Cài đặt xampp để kết nối hệ quản trị cơ sở dữ liệu MySQL.

Cài đặt Visual Studio Code để lập trình javaScript.

### 1.2 Yêu cầu phần cứng

Các thiết bị phần cứng phải thỏa mãn các yêu cầu:

- Bộ xử lý: Pentium III 1.8 Ghz trở lên.

- RAM: 2 GB trở lên.

- HDD: 80 GB trở lên.

- Cấu hình mạng: Có Internet.

- Độ phân giải màn hình: 960 x 640.

Phải đảm bảo rằng các thiết bị phần cứng có đủ khả năng phục vụ hệ thống hoạt động hiệu quả và nhanh chóng.

### 1.3 Hệ điều hành

Hệ thống phải tương tác tốt với các phần mềm sau:

- Hệ điều hành: Mọi hệ điều hành.

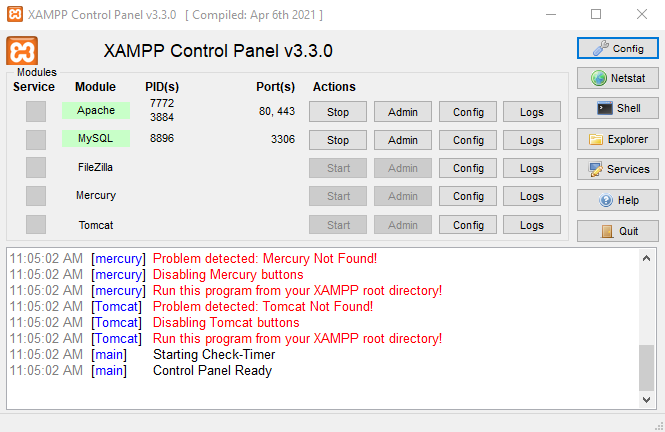
- Webserver: giả lập với Apache.

- Hệ cơ sở dữ liệu MySQL hoạt động trên xampp.

- Ngôn ngữ lập trình: SCSS, HTML, Javascript,...

- Hệ thống có thể chạy tốt với các trình duyệt: Chrome, Cốc Cốc, Firefox...

## 2. Kết quả thực hiện



A screenshot of a computer

Description automatically generated

# CHƯƠNG 5: KIỂM THỬ VÀ ĐÁNH GIÁ

## 1. Giao diện khách vãng lai và bệnh nhân