**ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**

**BÁO CÁO ĐỒ ÁN**

**XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH**

**SVTH: NGUYỄN LÊ NGUYÊN ANH MSSV: 17110094**

**SVTH: HỒ QUỐC ĐẠT MSSV: 17110117**

**Môn học: CÔNG NGHỆ PHẦN MỀM MỚI**

**GVHD: NGUYỄN THANH PHƯỚC**

Tp. Hồ Chí Minh, tháng 12 năm 2020

# LỜI CẢM ƠN

Lời đầu tiên, nhóm thực hiện xin được phép gửi lời cảm ơn đến khoa Đào tạo Chất Lượng Cao – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo điều kiện thuận lợi nhất cho nhóm thực hiện có cơ hội được tiếp cận, tham khảo, nghiên cứu để mở rộng và trau dồi thêm nhiều kiến thức trong lĩnh vực Công nghệ thông tin nói chung và môn học Công nghệ phần mềm mới nói riêng.

Lời cảm ơn trân trọng nhất nhóm thực hiện xin chân thành gửi đến Thầy **Nguyễn Thanh Phước** – người đã dùng nhiều thời gian, tâm huyết để hướng dẫn và tạo điều kiện thuận lợi giúp đỡ cho nhóm trong suốt quá trình thực hiện đồ án môn học. Thầy đã bỏ ra nhiều thời gian để giải đáp những vướng mắc của nhóm, nhắc nhở, điều chỉnh để nhóm đi đúng hướng. Sự nhiệt tình, quan tâm của thầy đã giúp nhóm có thêm động lực thực hiện đề tài, nghiên cứu và mở rộng nhiều kiến thức mới, đặc biệt là trong lĩnh vực xây dựng Website.

**Nhóm sinh viên thực hiện**

# LỜI MỞ ĐẦU

Ngày nay Công Nghệ Thông Tin đã và đang phát triển đạt đến mức độ cao, được ứng dụng vào mọi ngành, mọi lĩnh vực của nền kinh tế đất nước góp phần quan trọng làm cho các ngành này phát triển nhanh hơn, hiệu quả hơn. Việc thiết kế và xây dựng các website trong thời đại **công nghệ 4.0** này nhằm đáp ứng nhu cầu ngày càng cao của người dùng trên nhiều lĩnh vực.

Trong lĩnh vực bệnh viện đa khoa, bệnh nhân khi muốn được tư vấn sức khỏe và khám bệnh thì cần phải đến trực tiếp và làm giấy khám bệnh. Tuy nhiên, số lượng bệnh nhân đến khá đông và việc tư vấn hay đặt lịch sẽ trở nên lâu hơn, bệnh nhân phải chờ đợi trong một khoảng thời gian dài. Vì thế, bệnh nhân muốn được tư vấn và đặt lịch khám bệnh trong thời đại 4.0 này, chúng ta cần phát triển một Website trực tuyến, kết nối bệnh nhân với bác sĩ một cách nhanh chóng nhất và không làm mất thời gian nhiều của bệnh nhân. Từ những khảo sát và nghiên cứu ấy, nhóm đã xây dựng một hệ thống **Website đặt lịch khám bệnh**.

# MỤC LỤC

Contents

[LỜI CẢM ƠN i](#_Toc60618736)

[LỜI MỞ ĐẦU ii](#_Toc60618737)

[MỤC LỤC iii](#_Toc60618738)

[DANH MỤC CÁC CHỮ VIẾT TẮT iv](#_Toc60618739)

[DANH MỤC CÁC BẢNG BIỂU v](#_Toc60618740)

[DANH MỤC CÁC HÌNH ẢNH, BIỂU ĐỒ vi](#_Toc60618741)

[1. Chương 1: TỔNG QUAN 1](#_Toc60618742)

[1.1. Lý do chọn đề tài 1](#_Toc60618743)

[1.2. Mục tiêu nghiên cứu đề tài 1](#_Toc60618744)

[1.3. Công nghệ nghiên cứu cho đề tài. 1](#_Toc60618745)

[**1.3.1.** **Công nghệ sử dụng.** 1](#_Toc60618746)

[**1.3.2.** **Lý do chọn công nghệ.** 2](#_Toc60618747)

[1.4. Danh sách các trang dự tính 2](#_Toc60618748)

[2. Chương 2: CƠ SỞ LÝ THUYẾT 4](#_Toc60618749)

[2.1. NodeJS 4](#_Toc60618750)

[2.1.1. Giới thiệu về nodeJS 4](#_Toc60618751)

[2.1.2. Tại sao cần sử dụng nodeJS 4](#_Toc60618752)

[2.2. ReactJS 5](#_Toc60618753)

[2.2.1. Giới thiệu về ReactJS 5](#_Toc60618754)

[2.2.2. Tại sao cần sử dụng React 6](#_Toc60618755)

[2.3. MySQL 6](#_Toc60618756)

[2.3.1. Giới thiệu về mySQL 6](#_Toc60618757)

[2.3.2. Tại sao nên dùng mySQL 7](#_Toc60618758)

[*2.4.* *RESTFUL API* 9](#_Toc60618759)

[**2.4.1.** **Định nghĩa RESTFUl API** 9](#_Toc60618760)

[**2.4.2.** **Cách hoạt động chính.** 9](#_Toc60618761)

[3. Chương 3: THIẾT KẾ WEBSITE 11](#_Toc60618762)

[3.1. Đặc tả hệ thống 11](#_Toc60618763)

[3.1.1. Class Diagram 11](#_Toc60618764)

[3.1.2. Mô tả các class 12](#_Toc60618765)

[3.2. Thiết kế Prototype giao diện chương trình. 15](#_Toc60618766)

[**3.2.1.** **Prototype trang chủ** 15](#_Toc60618767)

[**3.2.2.** **Prototype đăng nhập** 16](#_Toc60618768)

[**3.2.3.** **Prototype đăng ký** 17](#_Toc60618769)

[**3.2.4.** **Prototype đặt lịch khám** 17](#_Toc60618770)

[**3.2.5.** **Prototype thông tin liên hệ** 18](#_Toc60618771)

[**3.2.6.** **Prototype thông tin người dùng** 19](#_Toc60618772)

[4. Chương 4: THỰC HIỆN CHỨC NĂNG HỆ THỐNG 21](#_Toc60618773)

[4.1. Các chức năng cần thiết cho hệ thống 21](#_Toc60618774)

[**4.1.1.** **Đăng nhập và phân quyền** 21](#_Toc60618775)

[**4.1.2.** **Đăng ký** 21](#_Toc60618776)

[**4.1.3.** **Đặt lịch khám và xác thực** 21](#_Toc60618777)

[**4.1.4.** **Hiển thị danh sách bệnh** 21](#_Toc60618778)

[4.2. Các phi chức năng của hệ thống. 22](#_Toc60618779)

[**4.2.1.** **Mã hóa mật khẩu tài khoản để lưu vào mySQL (bảo mật)** 22](#_Toc60618780)

[**4.2.2.** **Khả năng mở rộng chương trình.** 22](#_Toc60618781)

[5. Chương 5: KẾT LUẬN 23](#_Toc60618782)

[5.1. Tổng kết giao diện 23](#_Toc60618783)

[5.1.1. Giao diện đăng nhập 23](#_Toc60618784)

[5.1.2. Giao diện đăng ký 24](#_Toc60618785)

[**5.1.3.** **Giao diện đặt lịch khám** 24](#_Toc60618786)

[**5.1.4.** **Giao diện thông tin người dùng** 24](#_Toc60618787)

[**5.1.5.** **Giao diện danh sách** 25](#_Toc60618788)

[**5.1.6.** **Thông tin liên hệ** 25](#_Toc60618789)

[5.2. Hạn chế 26](#_Toc60618790)

[5.3. Hướng phát triển 26](#_Toc60618791)

[TÀI LIỆU THAM KHẢO 27](#_Toc60618792)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **CHỮ VIẾT TẮT** | **NGHĨA** |
| **UI** | **User Interface** |
| **JSX** | **Javascript + XML** |
| **DOM** | **Document Object Model** |
| **JS** | **Javascript** |
| **npm** | **Node package manager** |
| **DB** | **Database** |
| **RDBMS** | **Relational Database Management System** |
| **HTTP** | **HyperText Transfer Protocol** |
| **API** | **Application Programming Interface** |
| **REST** | **REpresentational State Transfer** |
| **CRUD** | **Create, Read, Update, Delete** |

# DANH MỤC CÁC BẢNG BIỂU

# DANH MỤC CÁC HÌNH ẢNH, BIỂU ĐỒ

# Chương 1: TỔNG QUAN

## Lý do chọn đề tài

Hiện nay, với sự phát triển vượt bậc của Công nghệ xây dựng Website, đặt lịch Online đã được áp dụng tốt trong các lĩnh vực như đặt vé xem phim, đặt vé xe khách, … Để thời gian của bệnh nhân tiếp cận bác sĩ một cách tối ưu nhất, phát hiện bệnh và chữa trị kịp thời, **Website đặt lịch khám bệnh** cho bệnh nhận được nghiên cứu và xây dựng.

Từ đó, bệnh nhân có thể được tư vấn sức khỏe và đặt lịch khám bệnh tại Bệnh viện một cách nhanh chóng, tiết kiệm thời gian và sức khỏe cho người bệnh. Nhờ vào công nghệ Mạng được phát triển rộng rãi, việc tư vấn và đặt lịch khám không còn là nỗi lo cho mọi người nói chung và bệnh nhân nói riêng.

## Mục tiêu nghiên cứu đề tài

Thiết kế và xây dựng được một website đặt lịch khám bệnh với đầy đủ các tính năng như: Cập nhật thông tin liên hệ tới các bác sĩ tư vấn, đặt lịch khám bệnh trực tuyến, giúp bệnh nhân tự bảo vệ sức khỏe thông qua các bài viết chia sẽ. Ngoài ra hỗ trợ các ban quản lý của bênh viện dễ dàng quản lý bệnh nhân đặt lịch khám, có thể dễ dàng theo dõi doanh thu và kiểm soát số lượng người bệnh. Bên cạnh đó, các y bác sĩ dễ dàng tiếp cận được bệnh nhân, chuẩn đoán và chữa trị nhanh nhất.

Xây dựng giao diện người dùng gần gũi, dễ sử dụng và đặt lịch nhanh chống, dễ tiếp cận với nhiều y bác sĩ từ các chuyên khoa khác nhau. Với giao diện người quản lý có đầy đủ chức năng để quản trị và kiểm soát số lượng lịch khám.

## Công nghệ nghiên cứu cho đề tài.

* + 1. **Công nghệ sử dụng.**
* Với sự phát triển mạnh mẽ của công nghệ xây dựng Website như: PHP, ASP.NET, … Nhóm thực hiện quyết định chọn NodeJS và ReactJS để xây dựng website một cách đơn giản và dễ dàng mở rộng.
  + 1. **Lý do chọn công nghệ.**
* NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh. Có rất nhiều sự hỗ trợ liên quan đến NodeJS
* Tốc độ xử lý nhanh: nhờ cơ chế xử lý bất đồng độ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
* Dễ dàng mở rộng: khi có nhu cầu phát triển website thì tính năng dễ dàng mở rộng của NodeJS là một lợi thế cực kỳ quan trọng.

## Danh sách các trang dự tính

* Các trang cho người dùng:
* Trang chủ: cho phép người dùng xem các giới thiệu hiện có, đăng nhập, đăng ký tài khoản, xem thông tin liên hệ và xem danh sách đội ngũ bac sĩ.
* Trang đăng ký: cho phép người dùng có thể đăng ký tài khoản bệnh nhân với tên, địa chỉ gmail và mật khẩu.
* Trang đăng nhập: bệnh nhân sau khi đăng ký tài khoản có thể dùng tên và mật khẩu để đăng nhập vào hệ thống.
* Trang đặt lịch khám bệnh: cho phép bênh nhân xem các thông tin như: danh sách các phòng khám, các bác sĩ, thời gian có thể đặt lịch khám. Sau đó có thể lựa chọn phù hợp nhất để đặt lịch đến bệnh viện.
* Trang bản đồ: cho phép bệnh nhân có thể xem địa chỉ của bệnh viện, giúp bệnh nhân tìm đúng nơi khám sau khi đặt lịch.
* Trang thông tin liên hệ: cho phép bệnh nhân có thể tư vấn trực tiếp đến các y bác sĩ, các quản lý của bệnh viện.
  1. ***Phân công công việc***

|  |  |  |
| --- | --- | --- |
| Stt | Tên sinh viên | Khái quát mảng công việc thực hiện trong đồ án |
| 1 | Hồ Quốc Đạt | * Thiết kế prototype. * Thiết kế Front-End cho website * Thiết kế Usecase |
| 2 | Nguyễn Lê Nguyên Anh | * Thiết kế cơ sở dữ liệu * Thiết kế Back-End cho website |

# Chương 2: CƠ SỞ LÝ THUYẾT

## NodeJS

### Giới thiệu về nodeJS

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.[1]

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

NPM là một công cụ tạo và quản lý các thư viện lập trình Javascript cho Node.js. Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework.

### Tại sao cần sử dụng nodeJS

Ưu điểm khi sử dụng NodeJS:

* NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh.
* Tốc độ xử lý nhanh. Nhờ cơ chế xử lý bất đồng độ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
* Dễ dàng mở rộng.

## ReactJS

### Giới thiệu về ReactJS

React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web. Nếu như AngularJS là một Framework cho phép nhúng code Javasscript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một thư viện cho phép nhúng code html trong code javascript nhờ vào JSX, có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn. [2]

Virtual DOM giúp tăng hiệu năng cho ứng dụng. ReactJS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề về tốc độ xử lý. Virtual DOM là một ‘object Javascript’, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM. React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện.

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript, thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Ngược với Javascript, JSX là kiểu statically- typed, nghĩa là được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng

Component: giúp phân chia các UI phức tạp thành các thành phần nhỏ và được phát triển và duy trì bởi Facebook.

Props: giúp các component tương tác với nhau, component nhận đầu vào là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.

State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật UI.

Không chỉ được xây dựng bên phía clients mà render trên server và có thể kết nối với nhau. Như một công cụ JavaScript open source, đi trước cả như Angular và Bootstrap.

### Tại sao cần sử dụng React

Ưu điểm khi sử dụng React:

* Dễ sử dụng.
* Hỗ trợ Reusable Component trong Java.
* Viết component dễ dàng hơn: sử dụng tiện ích mở rộng JSX, về cơ bản là sự kết hợp tuyệt vời giữa HTML và JavaScript.
* Hiệu suất tốt hơn với Virtual DOM.
* Thân thiện với SEO.

.

## MySQL

### Giới thiệu về mySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu theo hình thức quản lý các mối liên hệ giữa chúng.[3]

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. Các ứng dụng web lớn nhất như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

Database là tập hợp dữ liệu theo cùng một cấu trúc. Cơ sở dữ liệu là nơi chứa và sắp đặt dữ liệu. Dữ liệu được đặt trong một bộ dữ liệu chung, dataset, được tổ chức sắp xếp giống như một bảng tính. Mỗi bảng này có liên hệ với nhau theo cách nào đó. Vì vậy từ Relational (liên hệ) trong RDBMS có ý nghĩa như vậy. Nếu phần mềm không hỗ trợ mô hình dữ liệu quan hệ với nhau như vậy thì gọi là DBMS.

* *Mô hình Client-server:* Máy tính cài đặt và chạy phần mềm RDBMS được gọi là client (máy khách). Mỗi khi chúng cần truy cập dữ liệu, chúng kết nối tới máy chủ (server). Cách thức này chính là mô hình “client-server”.
* *MySQL Server*: là máy tính hay một hệ các máy tính cài đặt phần mềm MySQL dành cho server để giúp bạn lưu trữ dữ liệu trên đó, để máy khách có thể truy cập vào quản lý. Dữ liệu này được đặt trong các bảng, và các bảng có mối liên hệ với nhau. MySQL server nhanh, an toàn, đáng tin cậy. Phần mềm MySQL cũng miễn phí và được phát triển, phân phối và hỗ trợ bởi Oracle Corporation.
* *MySQL Client:* không hẵn phải cài phần mềm MySQL của Oracle mà là nói chung của mọi phần mềm có thể thực hiện truy vấn lên một MySQL server và nhận kết quả trả về. Phpmyadmin cũng là một MySQL client có giao diện người dùng.

### Tại sao nên dùng mySQL

MySQL không phải là hệ quản lý cơ sở dữ liệu (RDBMS) duy nhất trên thị trường, nhưng nó đích thực phổ biến nhất và chỉ xếp sau Oracle Database khi xét đến những thông số chính như số lượng tìm kiếm, profile người dùng trên LinkedIn, và lượng thảo luận trên các diễn đàn internet.

* Ưu điểm:
* Linh hoạt và dễ sử dụng.
* Hiệu năng cao.
* Được tin cậy và sử dụng nhiều qua các năm.
* An toàn.
  1. ***RESTFUL API***
     1. **Định nghĩa RESTFUl API**
* RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.
* API là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.
* REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.
* RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.
* Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API. [4]
  + 1. **Cách hoạt động chính.**

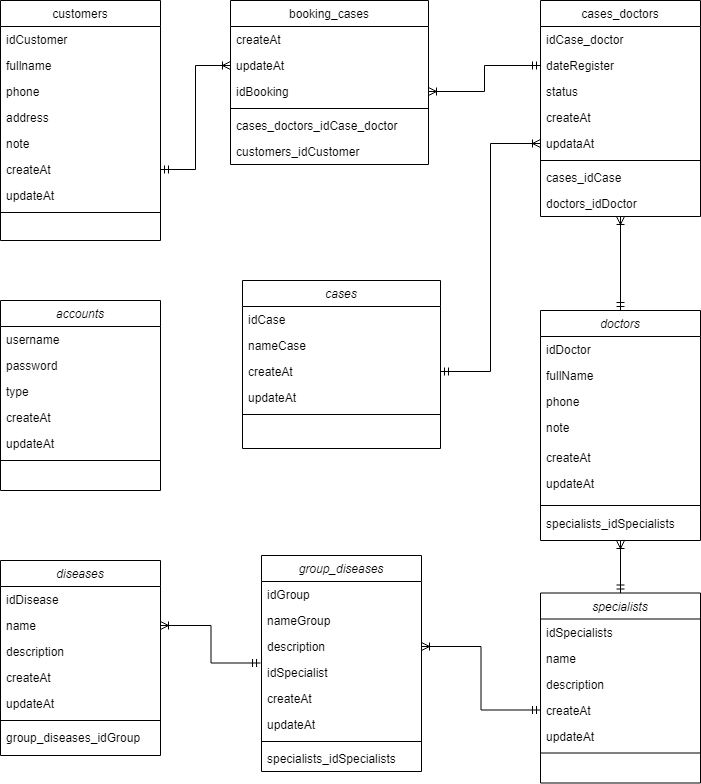


* REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.
* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.
* Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

# Chương 3: THIẾT KẾ WEBSITE

## Đặc tả hệ thống

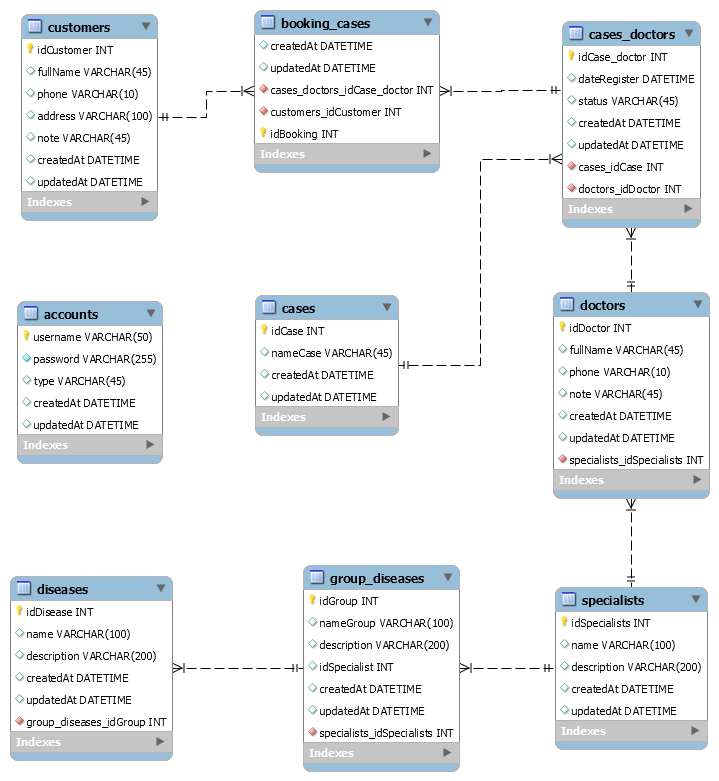
### Class Diagram



### Mô tả các class

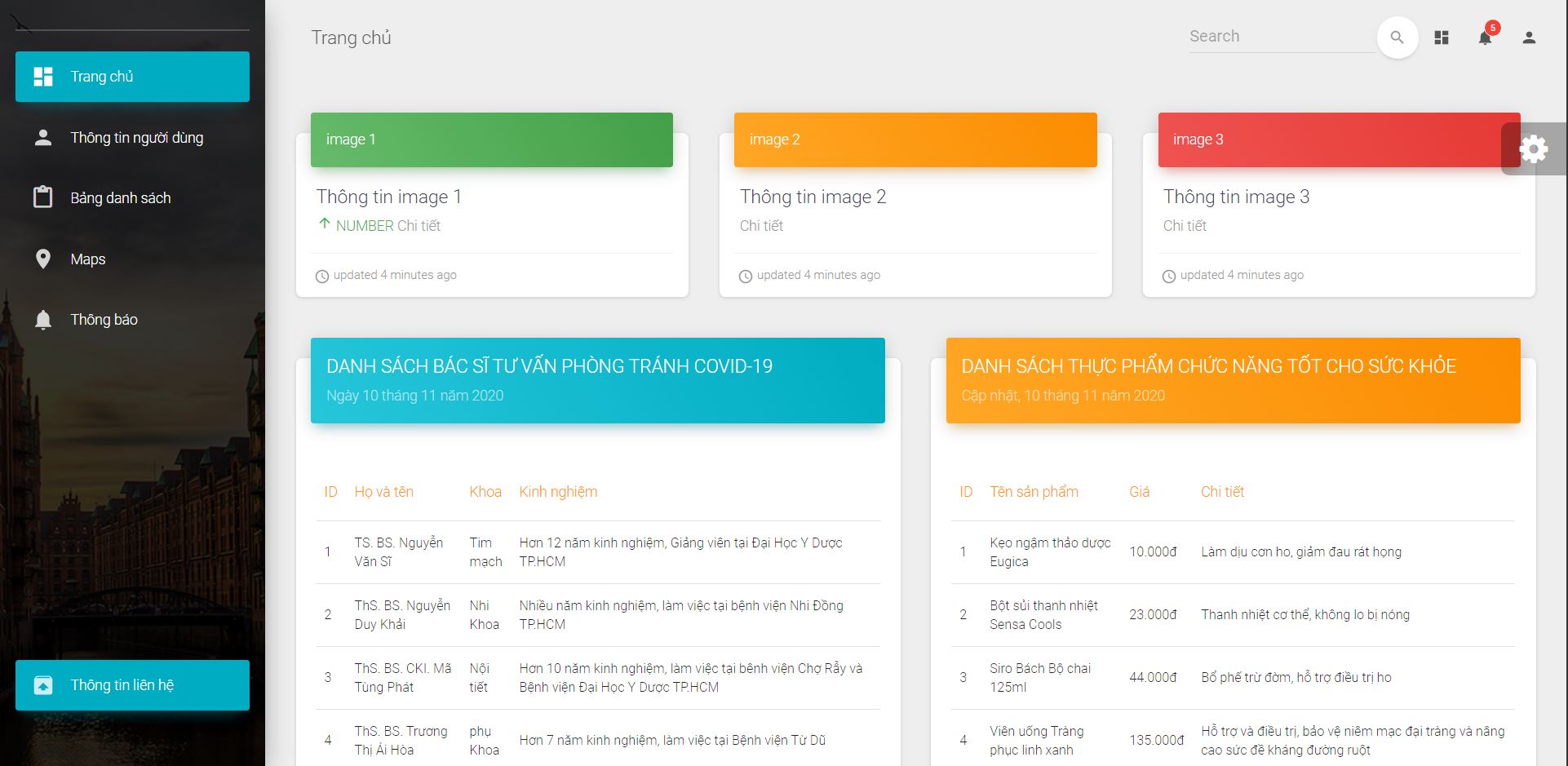
* Class Account:
* Username: tên người dùng đăng nhập.
* Password: mật khẩu người dùng.
* Type: loại người dùng. (bệnh nhân hay Admin).
* createAt: ngày đăng ký
* updateAt: ngày chỉnh sửa. thông tin tài khoản.
* Class Cases:
* idCase: mã số Ca khám bệnh
* nameCase: tên Ca
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class Diseases:
* idDisease: mã bệnh
* name: tên bệnh
* description: mô tả
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class group\_diseases:
* idGroup: mã số nhóm bệnh
* nameGroup: tên nhóm bệnh
* description: mô tả nhóm bệnh
* idSpecialist: mã số chuyên khoa
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class Specialists:
* idSpecialist: mã số chuyên khoa
* name: tên chuyên khoa
* description: mô tả chuyên khoa
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class Customers
* idCustomer: mã số khách hàng (bệnh nhân)
* fullName: tên bệnh nhân
* phone: số điện thoại
* address: địa chỉ
* note: ghi chú từ bệnh nhân
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class booking\_case:
* idBooking: mã số ca được đặt
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class Case\_doctors:
* idCase\_doctor: mã số ca trực của bác sĩ
* dateRegister: ngày đăng ký
* status: trạng thái
* createAt: ngày thêm vào
* updateAt: ngày chỉnh sửa
* Class Doctors:
* idDoctor: mã bác sĩ
* fullname: Tên bác sĩ
* phone: Số điện thoại
* note: ghi chú
* CreateAt: ngày đăng ký
* updateAt: ngày chỉnh sửa

ERD diagram

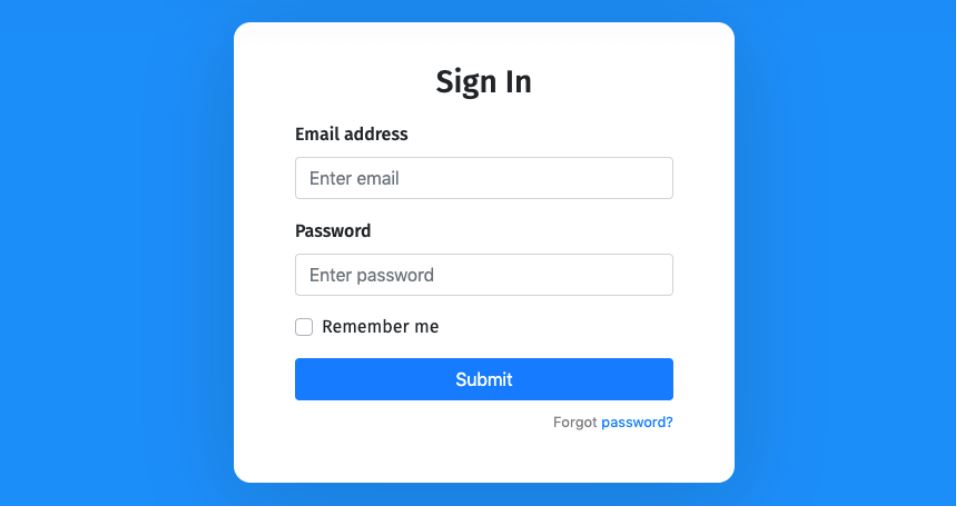


## Thiết kế Prototype giao diện chương trình.

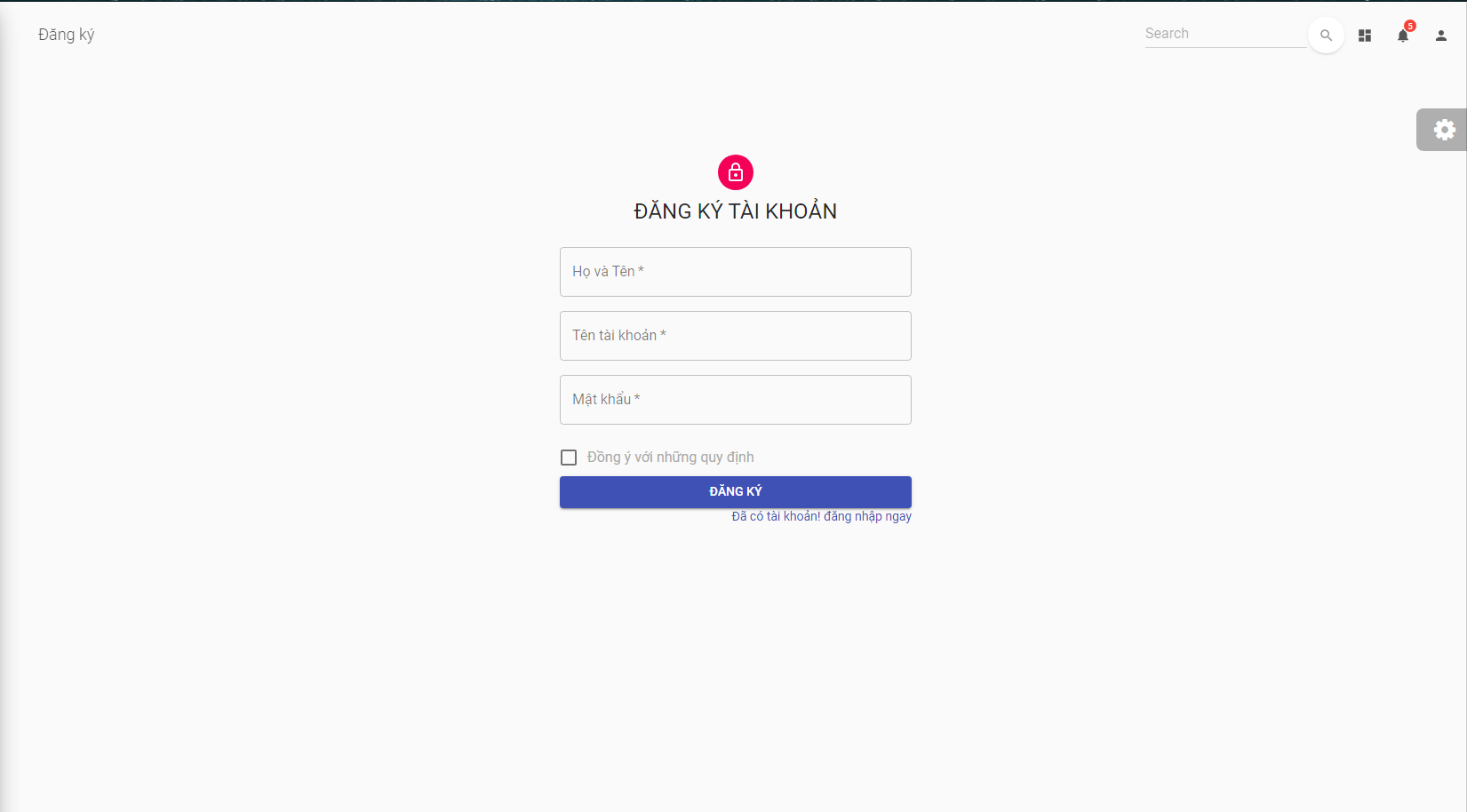
* + 1. **Prototype trang chủ**

****

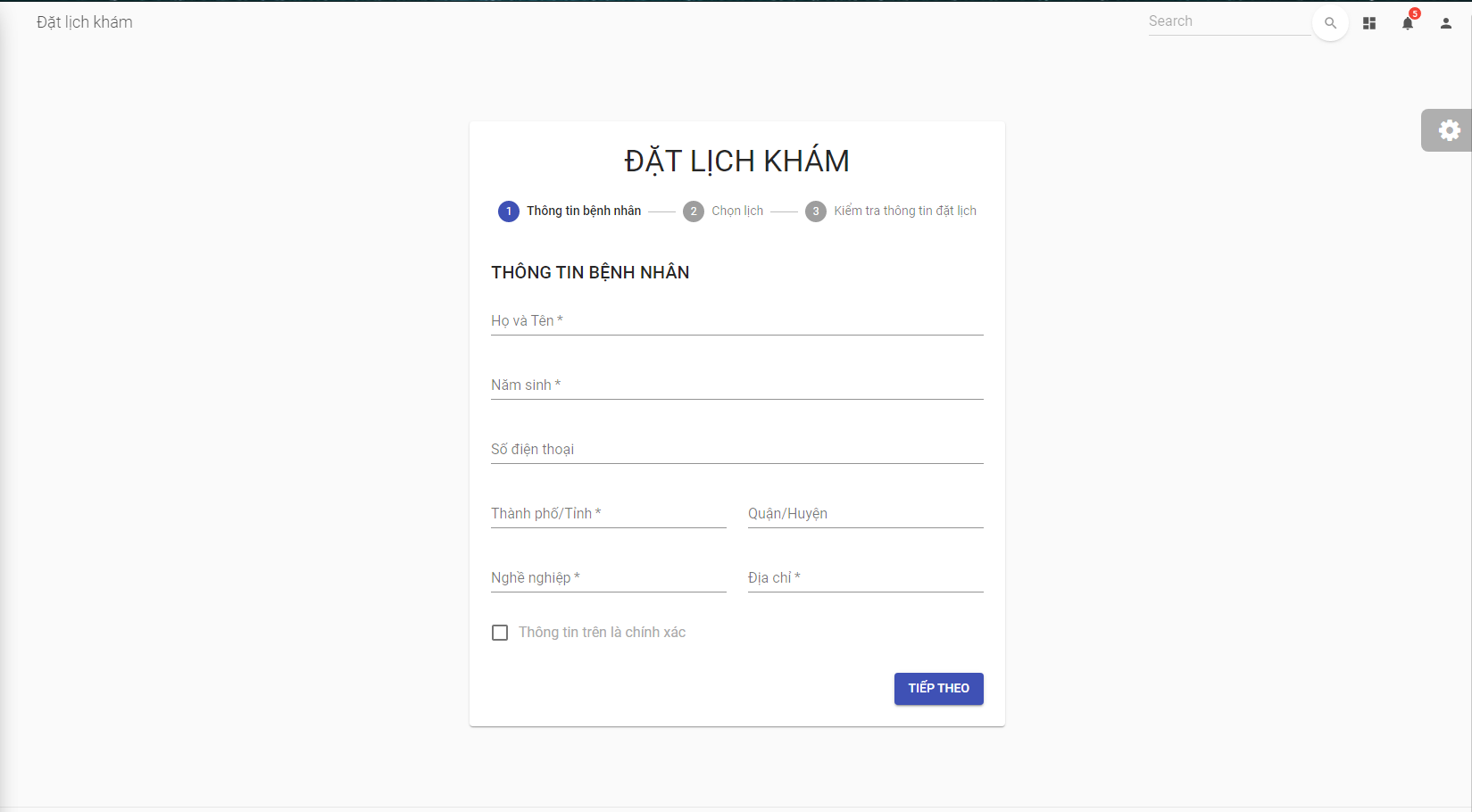
* Giao diện cho phép người dùng xem các thông tin mà bệnh viện sẽ thông báo công khai trên website.
* Người dùng có thể xem trực tiếp mà không cần phải đăng nhập.
  + 1. **Prototype đăng nhập**

****

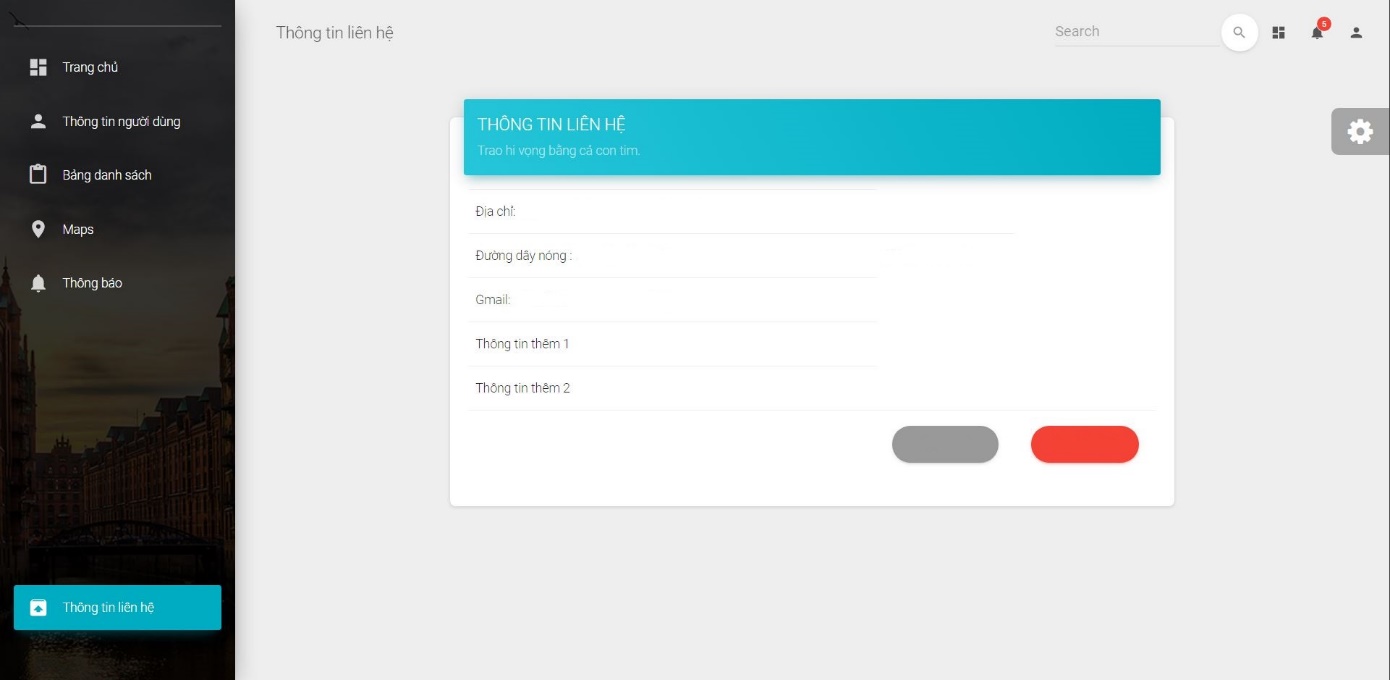
* Giao diện cho phép người dùng đăng nhập vào website, lưu thông tin người dùng.
* Cần đăng nhập để được đăng kí lịch khám tại trang đặt lịch khám.
* Nếu người dùng chưa có tài khoản để đăng nhập, có thể đăng ký để hệ thống tạo cho người dùng.
  + 1. **Prototype đăng ký**

****

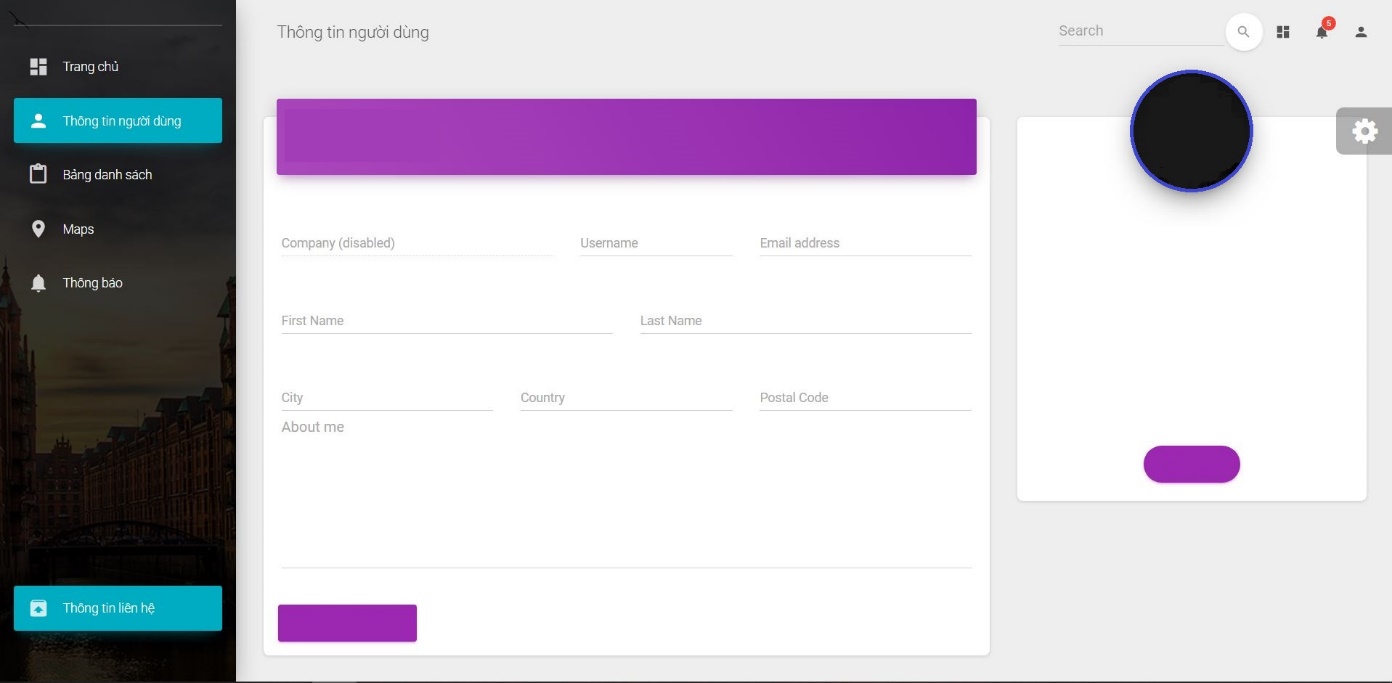
* Giao diện cho phép người dùng có thể đăng ký tài khoản riêng cho mình.
  + 1. **Prototype đặt lịch khám**

****

* Giao diện cho phép bệnh nhân có thể đặt lịch khám của mình tại bệnh viện.
* Bệnh nhân cần thực hiện 2 bước để đặt lịch thành công:
* Bước 1: thực hiện điền thông tin của bệnh nhân.
* Bước 2: thực hiện chọn loại triệu chứng, chọn tên bác sĩ, chọn lịch khám phù hợp.
  + 1. **Prototype thông tin liên hệ**

****

* Trang cho phép người dùng có thể xem thông tin cần liên hệ đến bệnh viện để được tư vấn và hướng dẫn.
* Trang không cần đăng nhập cũng có thể xem được.
  + 1. **Prototype thông tin người dùng**

****

* Trang cho phép người dùng có thể xem và chỉnh sửa thông tin tài khoản của mình.
* Trang cần phải đăng nhập để có thể thực hiện chỉnh sửa khi cần thiết.

# Chương 4: THỰC HIỆN CHỨC NĂNG HỆ THỐNG

## Các chức năng cần thiết cho hệ thống

* + 1. **Đăng nhập và phân quyền**
* Chức năng đăng nhập cho phép hệ thống có thể được bảo mật thông tin cơ bản nhất.
* Hệ thống cho phép người dùng đăng nhập với quyền là khách để được thực hiện các lệnh dành riêng.
* Hệ thống cho phép nhân viên y tế hoặc quản trị website tại bệnh viện để có thể thêm xóa sửa các thông tin cần thông báo đến người truy cập.
  + 1. **Đăng ký**
* Chức năng đăng kí cho phép hệ thống có thể cấp quyền cho nhân viên hoặc bệnh nhân có thể đăng ký tài khoản cho riêng mình.
* Hệ thống sẽ khởi tạo một tài khoản riêng cho người đăng ký để có thể thực hiện được các quyền truy cập.
  + 1. **Đặt lịch khám và xác thực**
* Chức năng đặt lịch khám cho phép bệnh nhân có thể tự đặt lịch để được khám theo những khung giờ trống trong thời gian nhanh nhất.
* Bệnh nhân có thể nắm được lịch bác sĩ và thời gian khám dành riêng cho mình.
* Theo các cách khám truyền thống, bệnh nhân phải đến trực tiếp bệnh viện gần nhất để được sắp xếp thứ tự khám, có khi bệnh nhân phải mất cả ngày để đến lượt khám cho mình. Bệnh nhân sẽ mất một khoản thời gian và sức khỏe. Dựa vào những khó khăn gặp phải của bệnh nhân khi muốn đăng ký khám. Chức năng này được hoàn thiện để thực hiện điều đó.
  + 1. **Hiển thị danh sách bệnh**
* Chức năng cho phép hệ thống thực hiện hiển thị danh sách các bệnh, các loại bệnh thuộc các chuyên khoa.
* Bệnh nhân có thể xem thông tin này và nắm được bệnh viện có thật sự đáp ứng những gì mà bệnh nhân muốn.

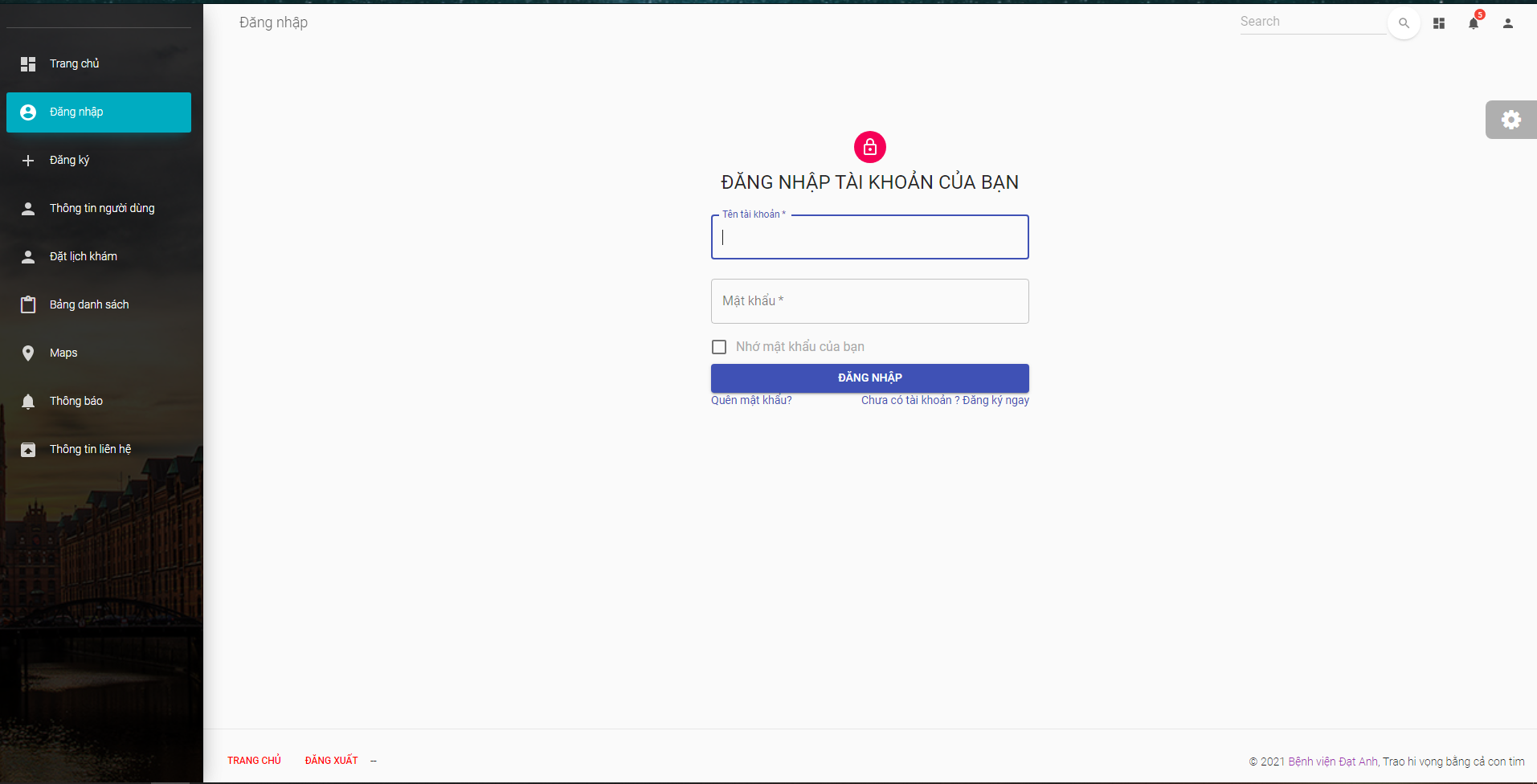
## Các phi chức năng của hệ thống.

* + 1. **Mã hóa mật khẩu tài khoản để lưu vào mySQL (bảo mật)**
* Mật khẩu sau khi được đăng ký thành công sẽ được mã hóa mà lưu xuống cơ sở dữ liệu.
* Tăng khả năng bảo mật cao cho tài khoản của mình.
  + 1. **Khả năng mở rộng chương trình.**
* Chương trình website có thể được mở rộng thêm nhiều tính năng đáp ứng nhu cầu sử dụng của khách hàng.

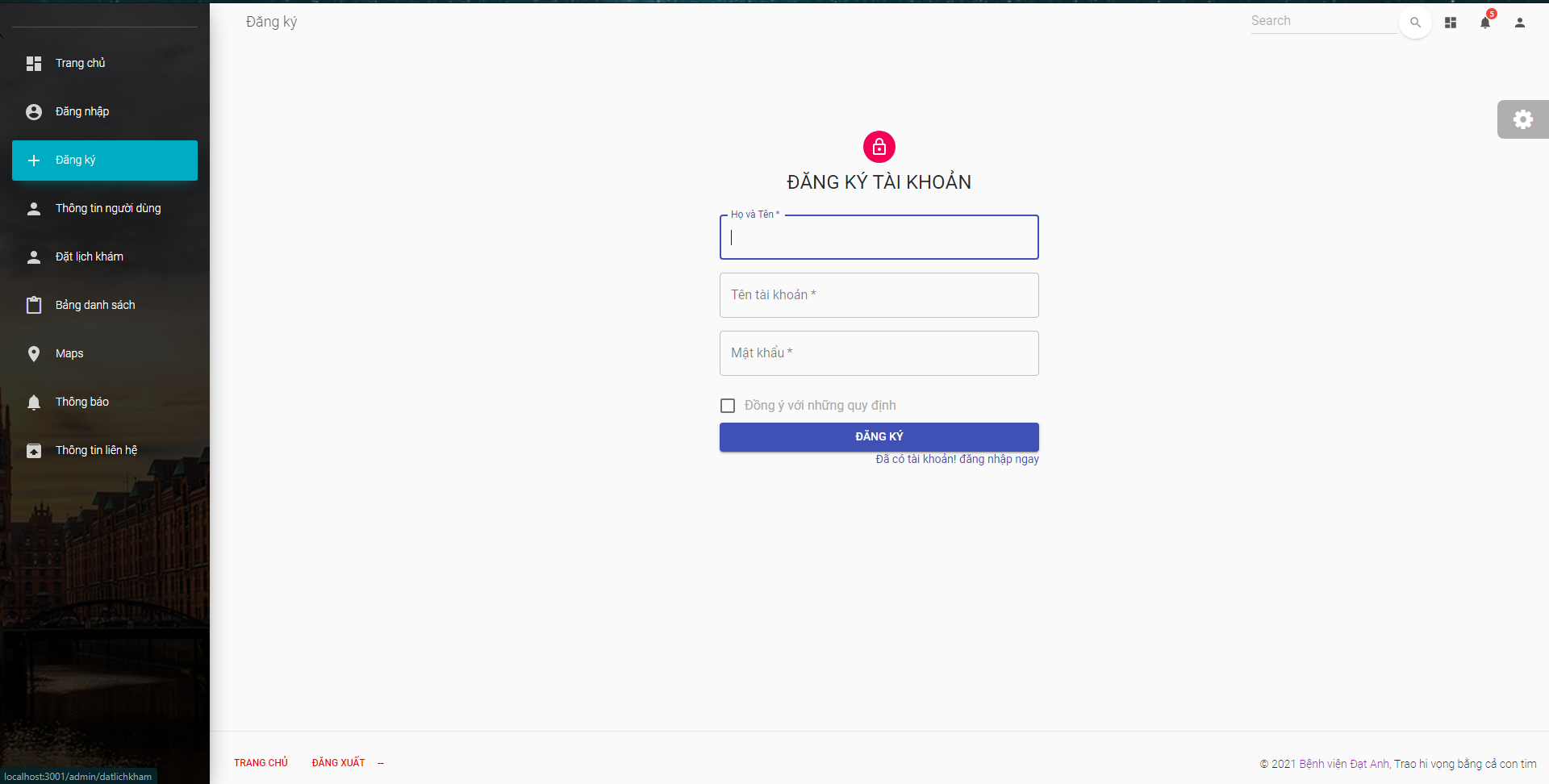
# Chương 5: KẾT LUẬN

## Tổng kết giao diện

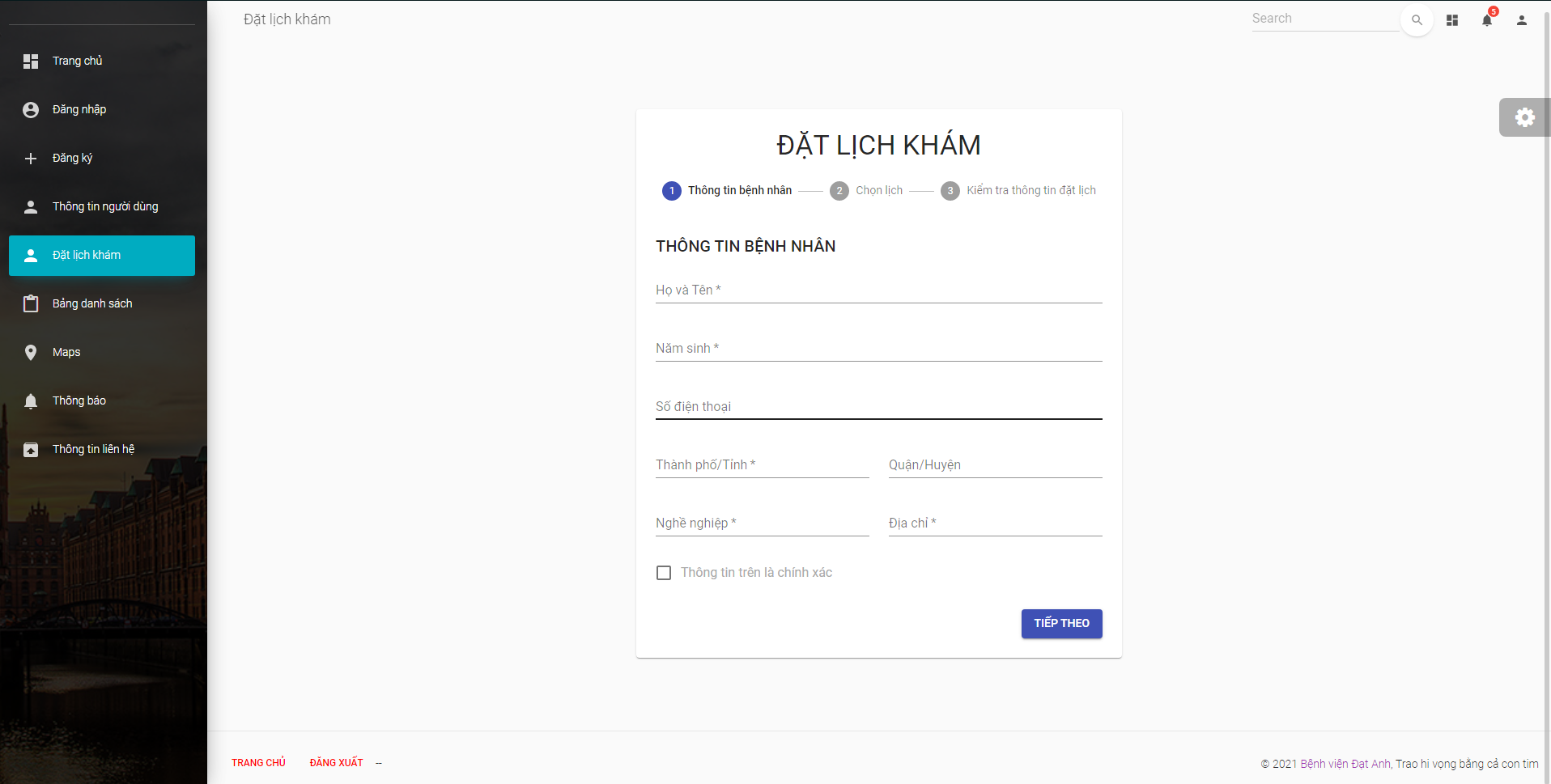
### Giao diện đăng nhập



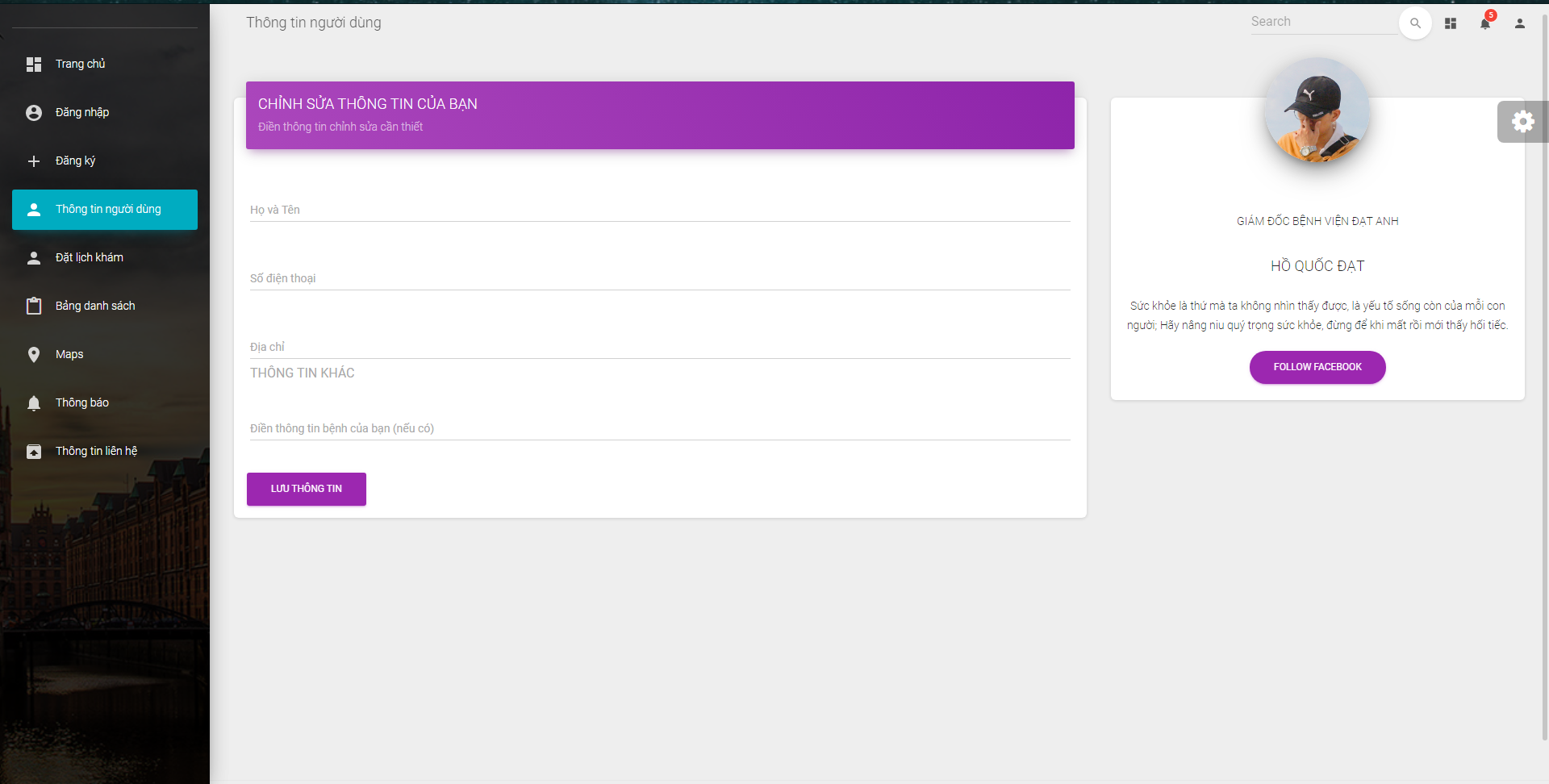
### Giao diện đăng ký



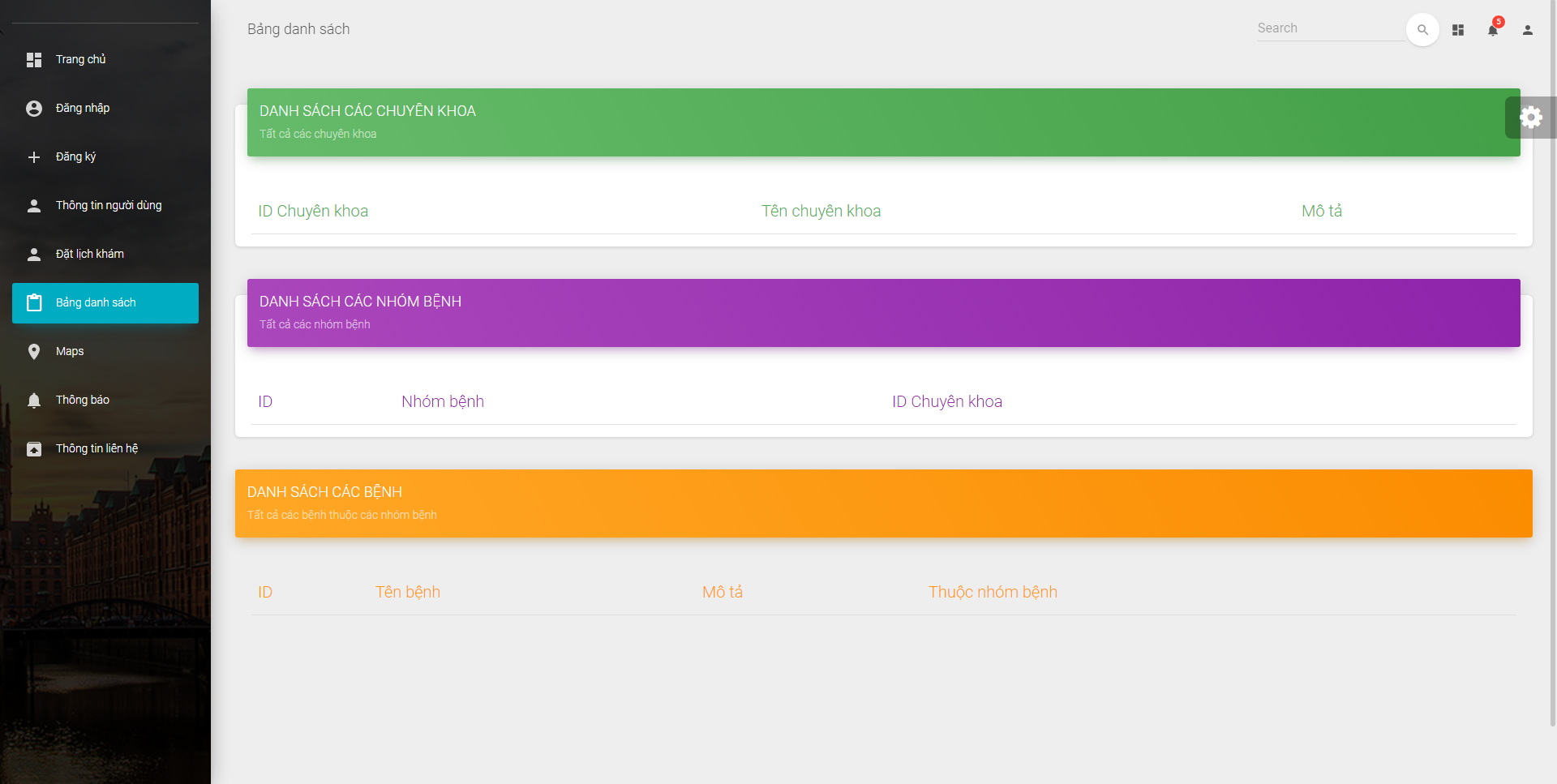
* + 1. **Giao diện đặt lịch khám**

****

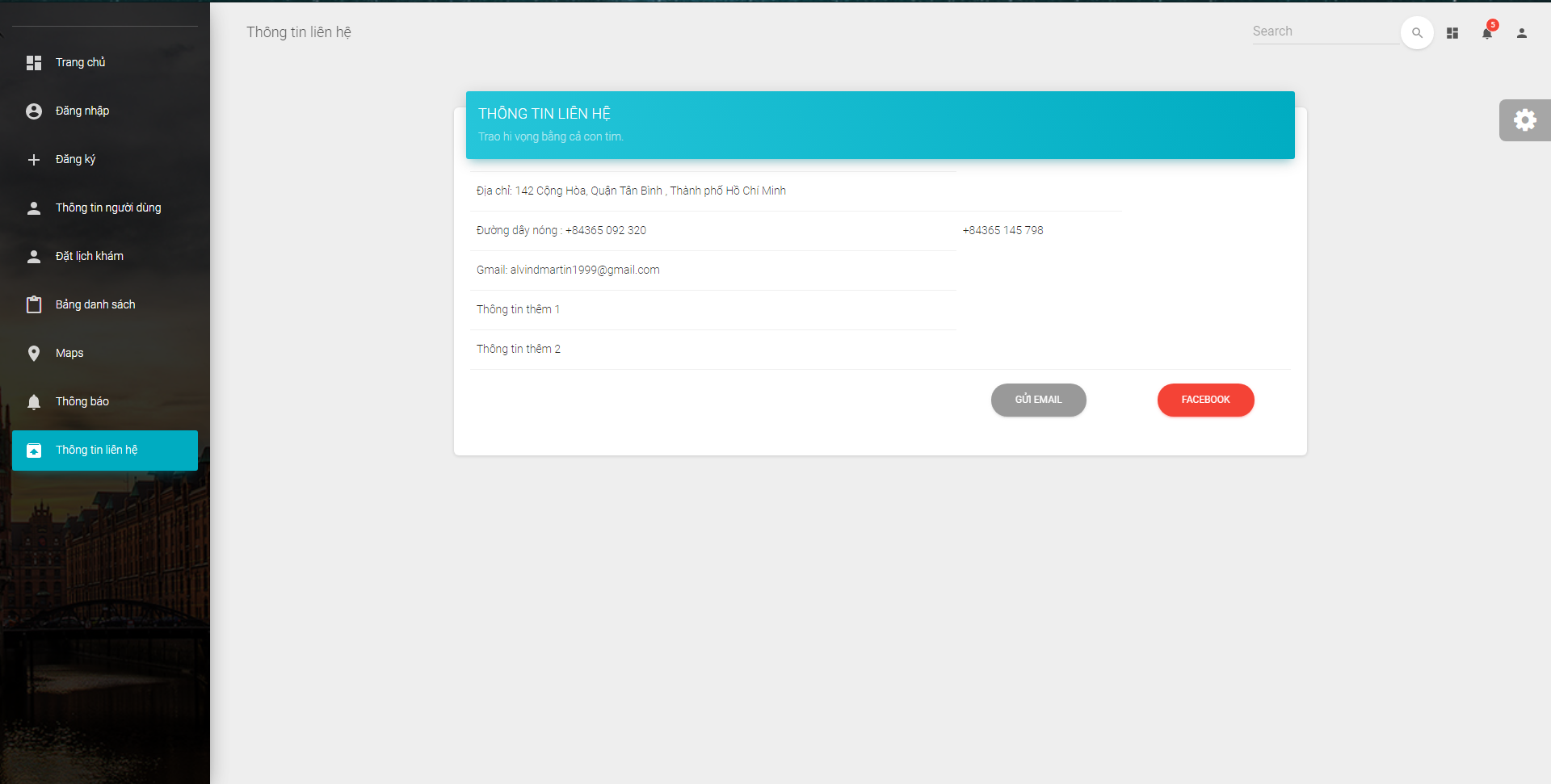
* + 1. **Giao diện thông tin người dùng**

****

* + 1. **Giao diện danh sách**

****

* + 1. **Thông tin liên hệ**

****

## Hạn chế

* Chương trình vẫn còn gặp nhiều lỗi khôgn mong muốn.

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

[1] <https://topdev.vn/blog/node-js-la-gi/>

[2] <https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7>

[3] <https://www.hostinger.vn/huong-dan/mysql-la-gi/>

[4] https://topdev.vn/blog/restful-api-la-gi/