**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**⋅⋅⋅⋅⋅🙣🕮🙡⋅⋅⋅⋅**

**A picture containing logo

Description automatically generated**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**XÂY DỰNG** **WEBSITE ĐẶT LỊCH KHÁM CHO PHÒNG KHÁM “NHA KHOA THỦ ĐÔ”**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **TS. Nguyễn Văn Tỉnh** |
| **Sinh viên thực hiện:** | **Nguyễn Xuân Duy – 2020604950** |
| **Khóa:** | **15** |

**Hà Nội – Năm 2024**

|  |  |
| --- | --- |
| NGUYỄN XUÂN DUY    NGÀNH KỸ THUẬT PHẦN MỀMẦN MỀM | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**        ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  *Ngành Kỹ Thuật Phần Mềm*    **XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM CHO PHÒNG KHÁM “NHA KHOA THỦ ĐÔ”**    **CBHD :TS. Nguyễn Văn Tỉnh**  **Sinh viên : Nguyễn Xuân Duy**  **Mã số sinh viên : 2020604950**  Hà Nội - 2024 |

**MỤC LỤC**

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT 5](#_Toc167104383)

[DANH MỤC BẢNG VÀ HÌNH ẢNH 6](#_Toc167104384)

[LỜI CẢM ƠN 8](#_Toc167104385)

[MỞ ĐẦU 10](#_Toc167104386)

[1. Lý do chọn đề tài. 10](#_Toc167104387)

[2. Đối tượng và phạm vi nghiên cứu. 10](#_Toc167104388)

[3. Kết quả mong muốn. 10](#_Toc167104389)

[4. Cấu trúc đồ án. 11](#_Toc167104390)

[CHƯƠNG 1. TỔNG QUAN VỀ ASP.NET FRAMEWORK VÀ MÔ HÌNH MVC 12](#_Toc167104391)

[1.1. Tìm hiểu về ASP.NET Framework. 12](#_Toc167104392)

[1.1.1. ASP.NET Framework 12](#_Toc167104393)

[1.1.2. Một số tính năng của ASP.NET Framework. 12](#_Toc167104394)

[1.1.3. Lý do nên sử dụng ASP.NET Framework 13](#_Toc167104395)

[1.1.4. Ưu, nhược điểm của ASP.NET Framework. 14](#_Toc167104396)

[1.2. Tìm hiểu về mô hình MVC. 16](#_Toc167104397)

[1.2.1. Mô hình MVC 16](#_Toc167104398)

[1.2.2. Luồng đi trong mô hình MVC. 17](#_Toc167104399)

[1.2.3. Ưu nhược điểm của mô hình MVC. 17](#_Toc167104400)

[1.2.4. Ứng dụng mô hình MVC vào lập trình. 18](#_Toc167104401)

[CHƯƠNG 2. XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH ONLINE 19](#_Toc167104402)

[2.1. Tổng quan về bài toán. 19](#_Toc167104403)

[2.1.1. Quy trình xây dựng sản phẩm. 19](#_Toc167104404)

[2.1.2. Kết quả đạt được. 20](#_Toc167104405)

[2.2. Khảo sát hệ thống. 20](#_Toc167104406)

[2.2.1. Khảo sát sơ bộ. 20](#_Toc167104407)

[2.2.2. Tài liệu đặc tả yêu cầu. 23](#_Toc167104408)

[2.3. Phân tích hệ thống. 24](#_Toc167104409)

[2.3.1. Mô hình Use case. 24](#_Toc167104410)

[2.3.2 Biểu đồ lớp của hệ thống 30](#_Toc167104411)

[2.3.3 Hiện thực hóa các Use case 30](#_Toc167104412)

[2.3.4 Mô hình hóa dữ liệu 38](#_Toc167104413)

[2.3.5 Thiết kế các bảng dữ liệu 39](#_Toc167104414)

[2.3.6 Thiết kế giao diện 44](#_Toc167104415)

[CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC. 50](#_Toc167104416)

[3.1. Kết quả đạt được. 50](#_Toc167104417)

[3.2. Kiểm thử chức năng hệ thống. 51](#_Toc167104418)

[3.2.1 Kiểm thử chức năng phía người dùng. 51](#_Toc167104419)

[3.2.2 Kiểm thử chức năng phía người quản lý. 52](#_Toc167104420)

[KẾT LUẬN 54](#_Toc167104421)

[TÀI LIỆU THAM KHẢO 55](#_Toc167104422)

# **DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt, ký hiệu** | **Giải thích** |
| 1 | MVC | Model – View – Controller |

**DANH MỤC BẢNG VÀ HÌNH ẢNH**

[Hình 2. 1 Biểu đồ Use Case 24](#_Toc167104443)

[Hình 2. 2 Mô tả chi tiết Use case Đặt lịch 24](#_Toc167104444)

[Hình 2. 3 Mô tả chi tiết Use case Sửa lịch 25](#_Toc167104445)

[Hình 2. 4 Mô tả chi tiết Use case Hủy lịch 26](#_Toc167104446)

[Hình 2. 5 Mô tả chi tiết Use case Quản lý nhân viên 26](#_Toc167104447)

[Hình 2. 6 Mô tả chi tiết Use case Quản lý nha sĩ 28](#_Toc167104448)

[Hình 2. 7 Biểu đồ lớp 30](#_Toc167104449)

[Hình 2. 8 Biểu đồ lớp Use case Đặt lịch 31](#_Toc167104450)

[Hình 2. 9 Biểu đồ trình tự Use case Đặt lịch 31](#_Toc167104451)

[Hình 2. 10 Biểu đồ hoạt động Use case Đặt lịch 32](#_Toc167104452)

[Hình 2. 11 Biểu đồ lớp Use case Sửa lịch 33](#_Toc167104453)

[Hình 2. 12 Biểu đồ trình tự Use case Sửa lịch 33](#_Toc167104454)

[Hình 2. 13 Biểu đồ hoạt động Use case Sửa lịch 34](#_Toc167104455)

[Hình 2. 14 Biểu đồ lớp Use case Hủy lịch 35](#_Toc167104456)

[Hình 2. 15 Biểu đồ Use case Hủy lịch 35](#_Toc167104457)

[Hình 2. 16 Biểu đồ hoạt động Use case Hủy lịch 36](#_Toc167104458)

[Hình 2. 17 Biểu đồ lớp Use case Quản lý nhân viên 36](#_Toc167104459)

[Hình 2. 18 Biểu đồ trình tự Use case Quản lý nhân viên 37](#_Toc167104460)

[Hình 2. 19 Biểu đồ lớp Use case Quản lý nha sĩ 37](#_Toc167104461)

[Hình 2. 20 Biểu đồ Use case Quản lý nha sĩ 38](#_Toc167104462)

[Hình 2. 21 Biểu đồ thực thể liên kết 39](#_Toc167104463)

[Hình 2. 22 Thiết kế Bảng LichKham 39](#_Toc167104464)

[Hình 2. 23 Thiết kế bảng KhachHang 40](#_Toc167104465)

[Hình 2. 24 Thiết kế bảng NhaSi 40](#_Toc167104466)

[Hình 2. 25 Thiết kế bảng NhanVien 40](#_Toc167104467)

[Hình 2. 26 Thiết kế bảng LichSuKhamBenh 41](#_Toc167104468)

[Hình 2. 27 Thiết kế bảng LichSuThayDoi 41](#_Toc167104469)

[Hình 2. 28 Thiết kế bảng Thuoc 41](#_Toc167104470)

[Hình 2. 29 Thiết kế bảng DonThuoc 42](#_Toc167104471)

[Hình 2. 30 Thiết kế bảng DichVu 42](#_Toc167104472)

[Hình 2. 31 Thiết kế bảng CaKham 43](#_Toc167104473)

[Hình 2. 32 Thiết kế bảng HoaDon 43](#_Toc167104474)

[Hình 2. 33 Thiết kế bảng PhanHoi 44](#_Toc167104475)

[Hình 2. 34 Màn hình trang chủ 44](#_Toc167104476)

[Hình 2. 35 Màn hình đặt lịch khám 45](#_Toc167104477)

[Hình 2. 36 Màn hình liên hệ 45](#_Toc167104478)

[Hình 2. 37 Màn hình đăng nhập 46](#_Toc167104479)

[Hình 2. 38 Màn hình trang Admin 46](#_Toc167104480)

[Hình 2. 39 Màn hình quản lý nhân viên 47](#_Toc167104481)

[Hình 2. 40 Màn hình quản lý nha sĩ 47](#_Toc167104482)

[Hình 2. 41 Màn hình quản lý ca khám 48](#_Toc167104483)

[Hình 2. 42 Màn hình quản lý khách hàng 49](#_Toc167104484)

[Hình 3. 1 Giao diện trang chủ…….…...…………………………………….50](#_Toc167104492)

[Hình 3. 2 Giao diện đặt lịch 50](#_Toc167104493)

[Hình 3. 3 Giao diện biểu đồ 51](#_Toc167104494)

[Hình 3. 4 Giao diện lịch khám người dùng 51](#_Toc167104495)

[Hình 3. 5 Giao diện đăng nhập 51](#_Toc167104496)

[Hình 3. 6 Bảng kiểm thử chức năng phía người dùng 51](#_Toc167104497)

[Hình 3. 7 Bảng kiểm thử chức năng phía người dùng 52](#_Toc167104498)

# **LỜI CẢM ƠN**

Trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp, Em đã nhận được sự hướng dẫn, giúp đỡ nhiệt tình của thầy giáo **TS.** **Nguyễn Văn Tỉnh**. Với lòng kính trọng và sự biết ơn sâu sắc. Em xin được bày tỏ lời cảm ơn chân thành tới thầy, người đã hết lòng giúp đỡ, dạy bảo, động viên và tạo mọi điều kiện thuận lợi cho em trong suốt quá trình học tập và hoàn thành báo cáo.

Đó là kinh nghiệm và bài học quý báu giúp cho em sẽ có những kinh nghiệm quý báu sau khi tốt nghiệp.

# **MỞ ĐẦU**

## **1. Lý do chọn đề tài.**

Trong thời đại công nghệ số phát triển như hiện nay, các hoạt động trực tuyến đang ngày càng trở nên phổ biến. Do đó, xây dựng một website đặt lịch khám bệnh online đã trở thành một lựa chọn tuyệt vời để mở rộng hoạt động của các phòng khám, bệnh viện.

Một trong những lý do quan trọng nhất khi chọn đề tài xây dựng website đặt lịch khám bệnh online là sự tiện lợi. Với một website đặt lịch khám bệnh online, khách hàng có thể đặt lịch mọi lúc, mọi nơi thông qua các thiết bị kết nối internet như máy tính, điện thoại thông minh hoặc máy tính bảng. Điều này không chỉ tiết kiệm thời gian mà còn mang lại cho khách hàng sự linh hoạt trong việc lựa chọn dịch vụ mà họ cần.

Trong bối cảnh thị trường ngày càng cạnh tranh gay gắt, việc sở hữu một website đặt lịch khám bệnh online chính là yếu tố quan trọng giúp phòng khám, bệnh viện phát triển và đạt được thành công. Do đó, Em quyết định chọn đề tài “ Xây dựng website đặt lịch khám cho phòng khám nha khoa Thủ Đô ” với mong muốn tạo ra một giải pháp tiên tiến, hiệu quả và tiện lợi cho khách hàng.

## **2. Đối tượng và phạm vi nghiên cứu.**

- Công cụ phát triển phần mềm: ASP.NET Framework.

- Mô hình MVC.

- Ngôn ngữ: C#.

- Công cụ: Visual Studio, Case Studio 2, Rational Rose, SQL Server, Draw.io.

## **3. Kết quả mong muốn.**

- Dự án đạt tiêu chuẩn đầu ra.

- Giao diện thân thiện, đẹp mắt.

- Chức năng thông minh, dễ dàng sử dụng.

## **4. Cấu trúc đồ án.**

Nội dung đồ án chia thành 3 chương:

- Chương 1: Tổng quan về ASP.NET Framework và mô hình MVC.

- Chương 2: Xây dựng website đặt lịch khám bệnh online.

- Chương 3: Kết quả đạt được.

# **CHƯƠNG 1. TỔNG QUAN VỀ** **ASP.NET FRAMEWORK VÀ MÔ HÌNH MVC**

## **1.1. Tìm hiểu về ASP.NET Framework.**

### **1.1.1. ASP.NET Framework**

ASP.NET Framework là một framework phần mềm được phát triển bởi Microsoft, dùng để xây dựng ứng dụng web và dịch vụ web. Nó cung cấp một mô hình lập trình mạnh mẽ và linh hoạt cho việc phát triển các ứng dụng web động và tương tác. ASP.NET Framework hỗ trợ nhiều ngôn ngữ lập trình như C# và VB.NET, và sử dụng mô hình lập trình dựa trên sự kiện (event-driven programming model).

ASP.NET Framework cung cấp một loạt các tính năng và dịch vụ như quản lý trạng thái (state management), xác thực và ủy quyền, kiểm soát dữ liệu và giao diện người dùng, cũng như khả năng tương tác với cơ sở dữ liệu. Nó là một phần quan trọng của công nghệ phát triển ứng dụng web của Microsoft và đã trải qua nhiều phiên bản và cải tiến qua thời gian.

### **1.1.2. Một số tính năng của ASP.NET Framework.**

**Web Forms và MVC:** ASP.NET hỗ trợ cả hai mô hình lập trình là Web Forms và MVC (Model-View-Controller), cho phép lựa chọn theo sở thích và yêu cầu cụ thể của ứng dụng.

**Quản lý Trạng thái (State Management):**ASP.NET cung cấp các cơ chế quản lý trạng thái như ViewState và Session để theo dõi trạng thái của ứng dụng trên nhiều yêu cầu từ người dùng.

**Xác thực và Ủy quyền:** Bảo vệ ứng dụng bằng cách xác thực người dùng và quản lý quyền truy cập thông qua ủy quyền.

**Lập trình Sự kiện (Event-Driven Programming):** Sử dụng mô hình lập trình dựa trên sự kiện, giúp xử lý sự kiện người dùng và tương tác với các yếu tố trên trang web.

**Giao tiếp với Cơ sở dữ liệu:** Hỗ trợ các công nghệ kết nối cơ sở dữ liệu như ADO.NET để tương tác với cơ sở dữ liệu như SQL Server, MySQL, Oracle, và nhiều hệ quản trị cơ sở dữ liệu khác.

**ASP.NET Web API:** Cho phép xây dựng và triển khai các dịch vụ web RESTful, giúp tương tác với các ứng dụng và thiết bị khác nhau.

**ASP.NET SignalR:** Cung cấp mô hình lập trình thời gian thực (real-time) thông qua kết nối hai chiều giữa máy chủ và trình duyệt.

**ASP.NET Core:**Phiên bản tiếp theo của ASP.NET Framework, ASP.NET Core, mang lại tính di động và đa nền tảng, hỗ trợ Linux và macOS cũng như Windows.

**Hỗ trợ Ngôn ngữ:** Hỗ trợ nhiều ngôn ngữ lập trình như C#, VB.NET, và F#.

**Kiểm thử và Gỡ lỗi:** Cung cấp công cụ tích hợp để kiểm thử và gỡ lỗi ứng dụng, bao gồm cả Visual Studio Debugger.

**Đa ngôn ngữ và Vùng:** Hỗ trợ phát triển ứng dụng có khả năng đa ngôn ngữ và vùng (globalization và localization).

**Bảo mật:**Cung cấp các tính năng bảo mật như SSL, quản lý phiên, kiểm soát truy cập và chống tấn công.

### **1.1.3. Lý do nên sử dụng ASP.NET Framework**

**Tích Hợp Mạnh Mẽ với Microsoft:** ASP.NET Framework được phát triển và hỗ trợ chính thức bởi Microsoft, điều này mang lại sự tích hợp mạnh mẽ với các sản phẩm và dịch vụ khác của Microsoft như Windows Server, SQL Server, Azure, Visual Studio, và nhiều hơn nữa.

**Ngôn Ngữ Lập Trình Phổ Biến:** Hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C#, VB.NET, và F#, giúp nhà phát triển lựa chọn ngôn ngữ mà họ thoải mái và hiểu biết.

**Mô Hình Lập Trình Dựa Trên Sự Kiện:** ASP.NET Framework sử dụng mô hình lập trình dựa trên sự kiện, giúp quản lý tốt các sự kiện người dùng trên trang web và cung cấp một quy trình làm việc rõ ràng.

**Quản Lý Trạng Thái Thuận Tiện:** Cung cấp các cơ chế quản lý trạng thái như ViewState và Session, giúp theo dõi trạng thái ứng dụng trên nhiều yêu cầu từ người dùng.

**Bảo Mật Mạnh Mẽ:** ASP.NET Framework tích hợp nhiều tính năng bảo mật như xác thực và ủy quyền, kiểm soát truy cập, SSL, và bảo vệ chống tấn công.

**Phát Triển Nhanh Chóng:** Sử dụng Visual Studio IDE giúp nhà phát triển tận dụng được nhiều tính năng như gỡ lỗi, tự động hoá, và kiểm thử, giúp tiết kiệm thời gian và tăng hiệu suất phát triển.

**Đa Nền Tảng (ASP.NET Core):** Phiên bản ASP.NET Core mang lại tính di động và đa nền tảng, hỗ trợ Linux và macOS cũng như Windows, giúp ứng dụng có khả năng chạy trên nhiều môi trường.

**Hỗ Trợ Cộng Đồng Lớn:** ASP.NET có một cộng đồng lớn và tích cực, với nhiều tài nguyên, tài liệu, và diễn đàn hỗ trợ trực tuyến.

**Dễ Dàng Tích Hợp Với Cơ Sở Dữ Liệu:** Hỗ trợ tốt cho kết nối và tương tác với các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, và Oracle.

**Hiệu Năng Cao:** ASP.NET Framework được tối ưu hóa để đạt được hiệu suất cao và xử lý một lượng lớn người dùng đồng thời.

### **1.1.4. Ưu, nhược điểm của ASP.NET Framework.**

**\*Ưu Điểm:**

- Tích Hợp Mạnh Mẽ với Microsoft:Tích hợp tốt với các sản phẩm và dịch vụ của Microsoft, giúp dễ dàng triển khai và quản lý ứng dụng trên nền tảng Microsoft.

- Ngôn Ngữ Lập Trình Phổ Biến :Hỗ trợ nhiều ngôn ngữ lập trình, cho phép nhà phát triển lựa chọn ngôn ngữ phù hợp với nhu cầu và sở thích.

- Mô Hình Lập Trình Dựa Trên Sự Kiện: Mô hình lập trình dựa trên sự kiện giúp quản lý sự kiện người dùng một cách hiệu quả.

- Bảo Mật Mạnh Mẽ:Cung cấp các tính năng bảo mật mạnh mẽ như xác thực và ủy quyền, giúp bảo vệ ứng dụng khỏi rủi ro bảo mật.

- Phát Triển Nhanh Chóng: Visual Studio IDE hỗ trợ mạnh mẽ, giúp tăng tốc độ phát triển và cung cấp các công cụ hữu ích.

- Hỗ Trợ Cộng Đồng Lớn: Có một cộng đồng lớn và tích cực, với nhiều tài nguyên và hỗ trợ trực tuyến.

- Dễ Dàng Tích Hợp Với Cơ Sở Dữ Liệu: Hỗ trợ tốt cho kết nối và tương tác với các hệ quản trị cơ sở dữ liệu phổ biến.

- Hiệu Năng Cao: Tối ưu hóa để đạt được hiệu suất cao và xử lý một lượng lớn người dùng đồng thời.

**\*Nhược Điểm:**

- Thể Hiện: Một số người phát triển có thể cảm thấy thể hiện của ASP.NET Framework nặng hơn so với một số công nghệ khác.

- Tính Năng "Overhead":Một số tính năng mạnh mẽ cũng đi kèm với sự phức tạp và chi phí tính toán, gây "overhead" cho ứng dụng.

- Phiên Bản Không Tương Thích: Các phiên bản mới có thể không hoàn toàn tương thích với phiên bản cũ, đòi hỏi công việc cập nhật và thử nghiệm.

- Giới Hạn Đa Nền Tảng (ASP.NET Framework): Phiên bản ASP.NET Framework truyền thống giới hạn đa nền tảng, không hỗ trợ nền tảng Linux và macOS một cách tốt như ASP.NET Core.

- Khả Năng Mở Rộng: Một số lập trình viên cho rằng khả năng mở rộng của ASP.NET Framework không linh hoạt bằng một số công nghệ khác.

- Chi Phí Tổ Chức và Học Tập: Đối với một số doanh nghiệp, chi phí tổ chức và học tập có thể cao đối với ASP.NET Framework.

- Khả Năng Tùy Biến Cao: Một số nhà phát triển cho rằng, do có nhiều tính năng và quy tắc, việc tùy biến có thể trở nên phức tạp.

## **1.2. Tìm hiểu về mô hình MVC.**

### **1.2.1. Mô hình MVC**

MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Các thành phần bên trong MVC:

+ Model (M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…

+ View (V): Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.

+ Controller (C): Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

### **1.2.2. Luồng đi trong mô hình MVC.**

Khi một yêu cầu từ máy client gửi tới server, Controller sẽ thực hiện nhiệm vụ của mình đó là tiếp nhận và xử lý yêu cầu. Trong trường hợp cần thiết, nó có thể liên hệ Model – bộ phận làm việc với database để hỗ trợ.

Khi xử lý xong yêu cầu, kết quả sẽ được trả về View. Tại View sẽ tạo mã HTML thành giao diện và trả về hiển thị trên trình duyệt.

### **1.2.3. Ưu nhược điểm của mô hình MVC.**

**\*Ưu Điểm:**

- Nhẹ, tiết kiệm băng thông: việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn

- Kiểm tra dễ dàng: Với MVC, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.

- Chức năng control: Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, Javascript có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.

- View và size: View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.

- Chức năng Soc (Separation of Concern): Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.

- Tính kết hợp: Việc tích hợp ở mô hình MVC cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.

- Đơn giản: Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.

**\*Nhược Điểm:**

- MVC thường được sử dụng vào những dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

### **1.2.4. Ứng dụng mô hình MVC vào lập trình.**

Mô hình MVC được ứng dụng trong nhiều ngôn ngữ lập trình khác nhau, nhưng phổ biến nhất là ứng dụng ASP.NET MVC.

MVC đang là mô hình được ứng dụng rất nhiều trong lập trình:

+ Hệ thống MVC phát triển tốt sẽ cho phép phát triển front – end, back – end cùng trên hệ thống mà không có sự can thiệp, chia sẻ, chỉnh sửa các tập tin trong khi một hoặc hai bên vẫn đang làm việc.

+ Việc sử dụng mô hình tương đối đơn giản. Chỉ cần hiểu rõ quy trình vận hành, nắm được các chức năng của từng bộ phận thì việc triển khai mô hình MVC tương đối dễ dàng.

# **CHƯƠNG 2. XÂY DỰNG WEBSITE ĐẶT LỊCH KHÁM BỆNH ONLINE**

## **2.1. Tổng quan về bài toán.**

Bài toán xây dựng một trang website đặt lịch khám bệnh trực tuyến là việc thiết kế, phát triển cho phép người dùng truy cập vào trang web để xem thông tin lịch khám, đặt lịch khám.

Một trang web đặt lịch khám bệnh trực tuyến có thể cung cấp cho khách hàng một số tính năng như:

+ Cho phép người dùng đặt, sửa và xóa lịch khám bệnh.

+ Có chức năng xem lịch khám cho phép khách hàng xem các lịch khám của mình.

+ Cho phép người dùng phản hồi thông tin, sửa thông tin cá nhân.

+ Cho phép người dùng quản lý lịch sử khám bệnh.

Để xây dựng một trang web đặt lịch khám bệnh trực tuyến thành công, cần phải đảm bảo có đủ tài nguyên và kỹ năng để triển khai các tính năng, bảo đảm an toàn thông tin khách hàng và đảm bảo hệ thống hoạt động ổn định và hiệu quả.

### **2.1.1. Quy trình xây dựng sản phẩm.**

Quy trình xây dựng một trang web đặt lịch khám bệnh trực tuyến có thể được phân chia thành các bước chính sau:

* Phân tích nhu cầu và đối tượng khách hàng:
* Xác định đối tượng khách hàng cần nhắm tới, họ có nhu cầu đặt lịch khám bệnh trực tuyến hay không?
* Thiết kế giao diện và trang web:
* Thiết kế giao diện và trang web để đảm bảo trải nghiệm người dùng tốt nhất.
* Thiết kế hệ thống cơ sở dữ liệu để quản lý thông tin nha sĩ, nhân viên.
* Phát triển tính năng:
* Phát triển tính năng cho trang web bao gồm quản lý nhân viên, nha sĩ.

### **2.1.2. Kết quả đạt được.**

**-** Website hoạt động ổn định.

- Giao diện thân thiện, dễ sử dụng.

- Tính chính xác và bảo mật cao.

## **2.2. Khảo sát hệ thống.**

### **2.2.1. Khảo sát sơ bộ.**

Khi xây dựng một website đặt lịch khám bệnh online, cần phải đưa ra một kế hoạch chi tiết và khảo sát bài toán kỹ càng để đảm bảo rằng trang web được phát triển đầy đủ chức năng và đáp ứng được nhu cầu của khách hàng.

Đầu tiên, cần phải xác định mục tiêu và đối tượng khách hàng mà trang web hướng đến. Phải hiểu rõ về nhu cầu của khách hàng.

Thứ hai, cần thiết kế giao diện trang web sao cho thân thiện với người dùng, dễ sử dụng và thân thiện với ngành y tế.

Thứ ba, cần phải xác định các tính năng cần thiết cho trang web, bao gồm các tính năng đặt lịch khám, quản lý lịch khám.

Ngoài ra, cần tích hợp các tính năng bảo mật và giao dịch an toàn để đảm bảo thông tin khách hàng được bảo vệ tốt nhất.

Tóm lại, việc xây dựng một website đặt lịch khám bệnh online đòi hỏi sự kỹ lưỡng và quan tâm đến chi tiết để đảm bảo trang web hoạt động tốt và đáp ứng được nhu cầu của khách hàng.

#### **2.2.1.1. Yêu cầu chức năng.**

**Đăng nhập:**

- Để đăng nhập trước tiên người dùng cần đăng ký lịch khám trước.

- Nhập Email người dùng và nhấn “Đăng nhập”. Hệ thống sẽ kiểm tra email đã đăng ký , nếu có lỗi sẽ thông báo nếu email đã đăng ký thì sẽ gửi một mã OTP về Email nếu nhập đúng OTP sẽ cho phép xem lịch khám của khách hàng.

**Đăng xuất:**

- Người dùng cần đăng nhập để sử dụng tính năng này.

- Cho phép người dùng đăng xuất khỏi hệ thống.

**Quản lý lịch khám**

**-Đặt lịch khám:**

- Ở trang chủ người dùng nhập đủ thông tin họ tên, email, nha sĩ, ngày khám, ca khám sau đó nhấn “Đặt lịch” hệ thống sẽ hiển thị màn hình thông báo đặt lịch thành công.

**-Sửa lịch khám:**

- Người dùng phải đăng nhập để sử dụng tính năng này.

- Sau khi đăng nhập người dùng sẽ thấy lịch khám của mình nhấn vào biểu tượng “Sửa” ở lịch khám. Sau đó thay đổi các thông tin cần thiết rồi nhấn “Save”.

**-Xóa lịch khám:**

- Người dùng phải đăng nhập để sử dụng tính năng này.

- Sau khi đăng nhập người dùng sẽ thấy lịch khám của mình nhấn vào biểu tượng “Xóa” ở lịch khám. Sau đó màn hình sẽ nhận thêm thông báo xác nhận xóa, người dùng nhấn “Xác nhận”.

**Xem lịch sử khám bệnh:**

- Người dùng phải đăng nhập để sử dụng tính năng này.

- Sau khi đăng nhập người dùng sẽ chọn “Lịch sử khám bệnh”. Ở đây hệ thống sẽ hiển thị lịch sử khám bệnh của người dùng.

**Xem lịch khám:**

- Người dùng phải đăng nhập để sử dụng tính năng này.

- Sau khi đăng nhập màn hình sẽ hiển thị lên các lịch khám của người dùng.

**Thống kê lịch khám:**

- Người quản trị phải đăng nhập để sử dụng tính năng này.

- Sau khi đăng nhập người quản trị sẽ thấy mà hình hiển thị biểu đồ thống kê lịch khám theo năm.

**Xuất báo cáo:**

- Người quản trị phải đăng nhập để sử dụng tính năng này.

- Sau khi đăng nhập người quản trị vào phần “Quản lý lịch khám”. Ở đây hệ thống sẽ hiển thị màn hình xem các lịch khám của khách hàng và người quản trị chọn “Xuất file Excel” hoặc “Xuất file PDF”.

**Quản lý nha sĩ:**

- Người quản trị phải đăng nhập để sử dụng tính năng này.

- Chức năng cho phép người quản trị thêm, sửa xóa nha sĩ.

**Quản lý nhân viên:**

- Người quản trị phải đăng nhập để sử dụng tính năng này.

- Chức năng cho phép người quản trị thêm, sửa, xóa nhân viên.

#### **2.2.1.2. Yêu cầu phi chức năng.**

**-** Trang web hoạt động ổn định 24/24, giao diện dễ nhìn, dễ thao tác.

- Hỗ trợ hoạt động trên nhiều trình duyệt khác nhau.

- Hiện thông báo khi xảy ra lỗi trong quá trình xử lý.

- An toàn, bảo mật: Đảm bảo an toàn thông tin của người dùng, không để bị lộ. Các thao tác nghiệp vụ của người quản trị chỉ được thực hiện khi có quyền.

### **2.2.2. Tài liệu đặc tả yêu cầu.**

- Mục đích của website: Website được thiết kế để cho phép người dùng đặt lịch khám bệnh trực tuyến.

- Mô tả chung:

+ Trang web có giao diện đơn giản và dễ sử dụng.

- Các tính năng của trang web:

+ Quản lý lịch khám: Người dùng có thể đặt, sửa, hủy lịch khám và xem thông tin chi tiết của lịch khám gồm ngày khám, ca khám, nha sĩ khám, email người dùng, tên người dùng.

- Yêu cầu kỹ thuật:

+ Trang web phải hoạt động tốt trên các trình duyệt web phổ biến như Chrome, Firefox, Safari và Edge.

+ Trang web phải đảm bảo tính bảo mật của dữ liệu khách hàng và thông tin thanh toán.

+ Trang web phải đảm bảo tính đáp ứng và tối ưu hóa trải nghiệm người dùng trên các thiết bị khác nhau, từ máy tính để bàn đến thiết bị di động.

+ Các chức năng phải hoạt động đúng và chính xác.

+ Thời gian tải trang web phải nhanh để đảm bảo trải nghiệm người dùng tốt.

## **2.3. Phân tích hệ thống.**

### **2.3.1. Mô hình Use case.**

#### **2.3.1.1. Biểu đồ Use case.**



Hình 2. 1 Biểu đồ Use Case

#### **2.3.1.2. Mô tả chi tiết Use case.**

1. **Use case đặt lịch khám**

Hình 2. 2 Mô tả chi tiết Use case Đặt lịch

|  |  |
| --- | --- |
| Mô tả | Use case cho phép khách hàng đặt lịch khám bệnh. |
| Người thực hiện | Khách hàng |
| Tiền điều kiện | Không |
| Luồng sự kiện chính | Use case này bắt đầu khi khách hàng click vào “Đặt lịch”.  Khách hàng nhập các thông tin theo yêu cầu của hệ thống và click vào nút “Đặt lịch”. Hệ thống lưu thông tin lịch khám vào bảng “AppointmentSchule” và hiển thị lên màn hình thông báo đăng ký thành công. Use case kết thúc. |
| Luồng rẽ nhánh | Nếu nhập không hợp lệ một trong các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, khách hàng có thể nhập lại trường dữ liệu đó hoặc thoát khỏi màn hình đặt lịch.  Không kết nối được cơ sở dữ liệu: 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. |
| Hậu điều kiện | Không |

1. **Use case sửa lịch khám**

Hình 2. 3 Mô tả chi tiết Use case Sửa lịch

|  |  |
| --- | --- |
| Mô tả | Use case cho phép khách hàng sửa lịch khám bệnh. |
| Người thực hiện | Khách hàng |
| Tiền điều kiện | Khách hàng đã đặt lịch |
| Luồng sự kiện chính | Use case này bắt đầu khi khách hàng click vào “Sửa”.  Khách hàng nhập các thông tin cần sửa và click vào nút “Đặt lịch”. Hệ thống lưu thông tin lịch khám vào bảng “AppointmentSchule” và hiển thị lên màn hình thông báo sửa thành công. Use case kết thúc. |
| Luồng rẽ nhánh | Nếu nhập không hợp lệ một trong các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, khách hàng có thể nhập lại trường dữ liệu đó hoặc thoát khỏi màn hình đặt lịch.  Không kết nối được cơ sở dữ liệu: 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. |
| Hậu điều kiện | Không |

1. **Use case hủy lịch khám**

Hình 2. 4 Mô tả chi tiết Use case Hủy lịch

|  |  |
| --- | --- |
| Mô tả | Use case cho phép khách hàng hủy lịch khám bệnh. |
| Người thực hiện | Khách hàng |
| Tiền điều kiện | Khách hàng đã đặt lịch |
| Luồng sự kiện chính | Use case này bắt đầu khi khách hàng click vào “Hủy”.  Khách hàng click vào nút “Xác nhận”. Hệ thống sẽ xóa lịch khám khỏi bảng “AppointmentSchule” và hiển thị lên màn hình thông báo xóa thành công. Use case kết thúc. |
| Luồng rẽ nhánh | Không kết nối được cơ sở dữ liệu: 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. |
| Hậu điều kiện | Không |

1. **Use case quản lý nhân viên**

Hình 2. 5 Mô tả chi tiết Use case Quản lý nhân viên

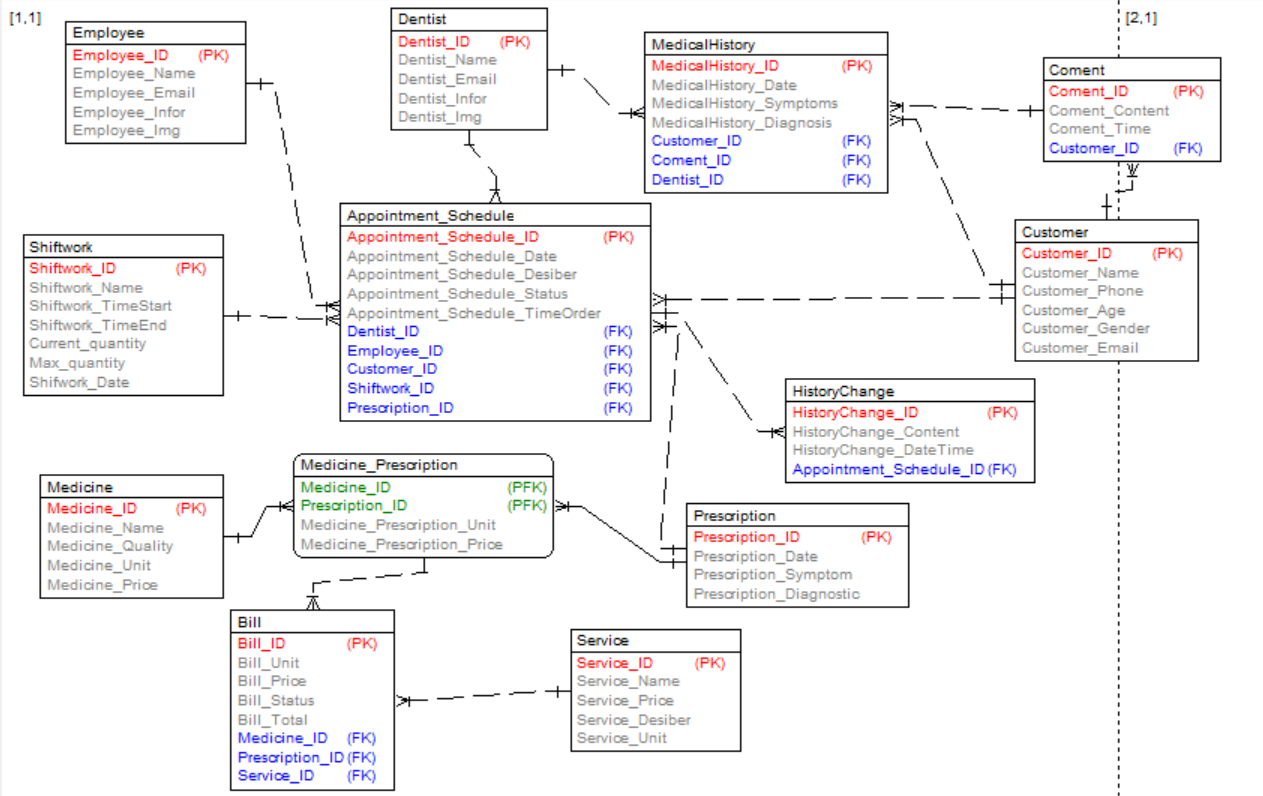
|  |  |
| --- | --- |
| Mô tả | Use case cho phép người quản trị thêm, sửa, xóa thông tin nhân viên |
| Người thực hiện | Người quản trị |
| Tiền điều kiện | Đăng nhập với tư cách người quản trị |
| Luồng sự kiện chính | Use case này bắt đầu khi người quản trị kích vào nút “Quản lý nhân viên” trên thanh menu quản trị. Hệ thống sẽ lấy danh sách thông tin nhân viên và hiển thị lên màn hình.  **Thêm nhân viên**  a. Người quản trị kích vào nút “Thêm Mới” trên cửa sổ danh sách nhân viên . Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho nhân viên.  b. Người quản trị nhập thông tin cho nhân viên mới và kích vào nút “Thêm Mới”. Hệ thống sẽ tạo một bản ghi mới trong bảng Employee và hiển thị danh sách các nhân viên được cập nhật ra màn hình.  **Sửa nhân viên**  a. Người quản trị kích vào nút “Sửa” trên một dòng nhân viên bất kỳ. Hệ thống lấy thông tin cũ của nhân viên và hiển thị lên màn hình trong các Textbox.  b. Người quản trị nhập thông tin mới của nhân viên và kích vào nút “Lưu”. Hệ thống sẽ lưu thông tin mới vào bảng Employee và hiển thị danh sách đã cập nhật ra màn hình.  **Xóa nhân viên**  a. Người quản trị ấn vào nút “Xóa” trên một dòng nhân viên, hệ thống sẽ hiển thị một yêu cầu xác nhận xóa.  b. Người quản trị ấn vào nút “Đồng Ý”. Hệ thống xóa nhân viên ra khỏi bảng Employee và hiện danh sách nhân viên đã cập nhật.  Use case kết thúc. |
| Luồng rẽ nhánh | Nếu nhập không hợp lệ một trong các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, người quản trị có thể nhập lại trường dữ liệu đó.  Không kết nối được cơ sở dữ liệu: 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. |
| Hậu điều kiện | Nếu use case kết thúc thành công thì thông tin về sản phẩm sẽ được cập nhật trong cơ sở dữ liệu. |

1. **Use case quản lý nha sĩ**

Hình 2. 6 Mô tả chi tiết Use case Quản lý nha sĩ

|  |  |
| --- | --- |
| Mô tả | Use case cho phép người quản trị thêm, sửa, xóa thông tin nha sĩ |
| Người thực hiện | Người quản trị |
| Tiền điều kiện | Đăng nhập với tư cách người quản trị |
| Luồng sự kiện chính | Use case này bắt đầu khi người quản trị kích vào nút “Quản lý nha sĩ” trên thanh menu quản trị. Hệ thống sẽ lấy danh sách thông tin nha sĩ và hiển thị lên màn hình.  **Thêm nha sĩ**  a. Người quản trị kích vào nút “Thêm Mới” trên cửa sổ danh sách nha sĩ . Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho nha sĩ.  b. Người quản trị nhập thông tin cho nha sĩ mới và kích vào nút “Thêm Mới”. Hệ thống sẽ tạo một bản ghi mới trong bảng Dentist và hiển thị danh sách các nha sĩ được cập nhật ra màn hình.  **Sửa nha sĩ**  a. Người quản trị kích vào nút “Sửa” trên một dòng nha sĩ bất kỳ. Hệ thống lấy thông tin cũ của nha sĩ và hiển thị lên màn hình trong các Textbox.  b. Người quản trị nhập thông tin mới của nha sĩ và kích vào nút “Lưu”. Hệ thống sẽ lưu thông tin mới vào bảng Dentist và hiển thị danh sách đã cập nhật ra màn hình.  **Xóa nhân viên**  a. Người quản trị ấn vào nút “Xóa” trên một dòng nha sĩ, hệ thống sẽ hiển thị một yêu cầu xác nhận xóa.  b. Người quản trị ấn vào nút “Đồng Ý”. Hệ thống xóa sản nha sĩ ra khỏi bảng Dentist và hiện danh sách nha sĩ đã cập nhật.  Use case kết thúc. |
| Luồng rẽ nhánh | Nếu nhập không hợp lệ một trong các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, người quản trị có thể nhập lại trường dữ liệu đó.  Không kết nối được cơ sở dữ liệu: 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. |
| Hậu điều kiện | Nếu use case kết thúc thành công thì thông tin về sản phẩm sẽ được cập nhật trong cơ sở dữ liệu. |

### **2.3.2 Biểu đồ lớp của hệ thống**

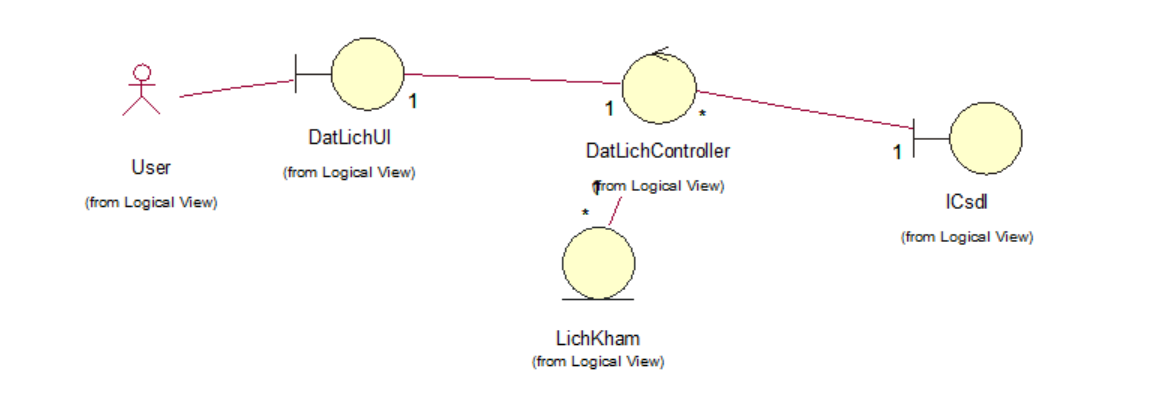
****

Hình 2. 7 Biểu đồ lớp

### **2.3.3 Hiện thực hóa các Use case**

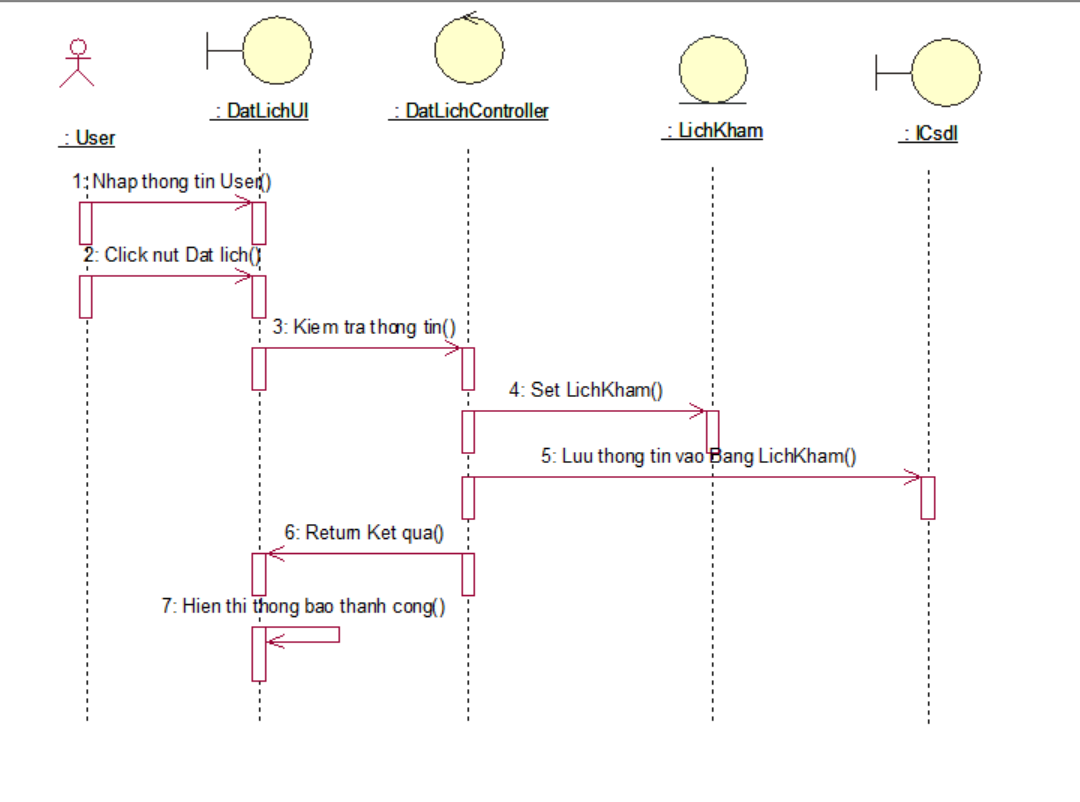
**2.3.3.1 Use case đặt lịch**

* **Biểu đồ lớp**



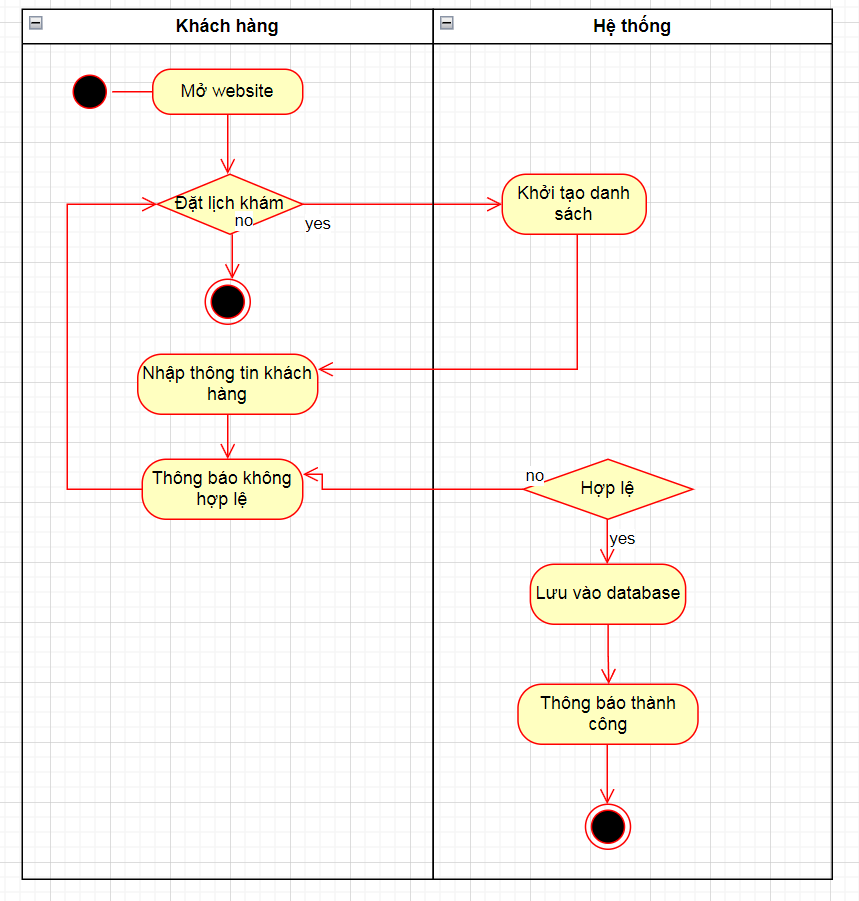
Hình 2. 8 Biểu đồ lớp Use case Đặt lịch

* **Biểu đồ trình tự**



Hình 2. 9 Biểu đồ trình tự Use case Đặt lịch

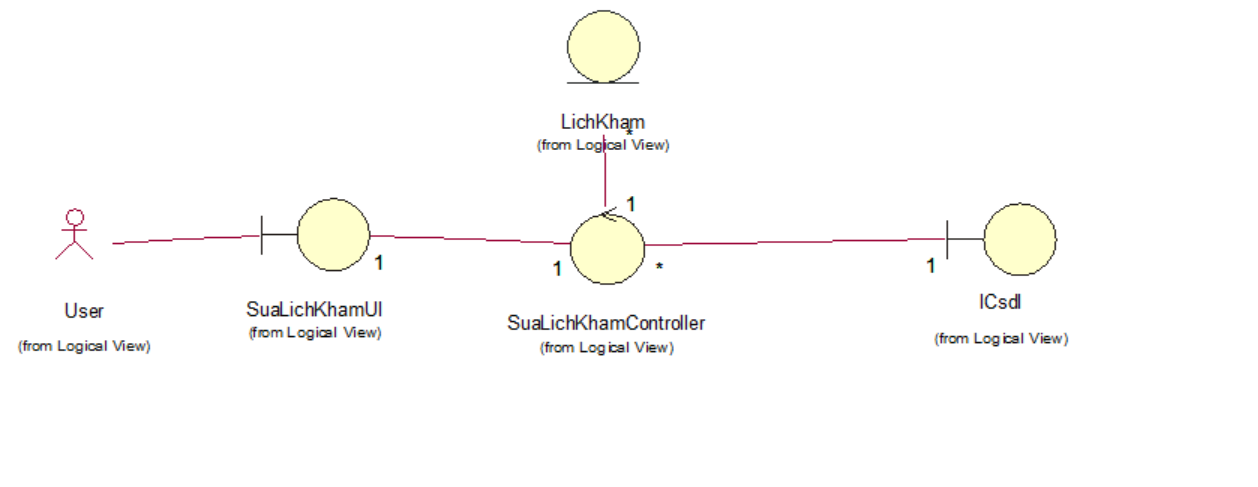
* **Biểu đồ hoạt động**

****

Hình 2. 10 Biểu đồ hoạt động Use case Đặt lịch

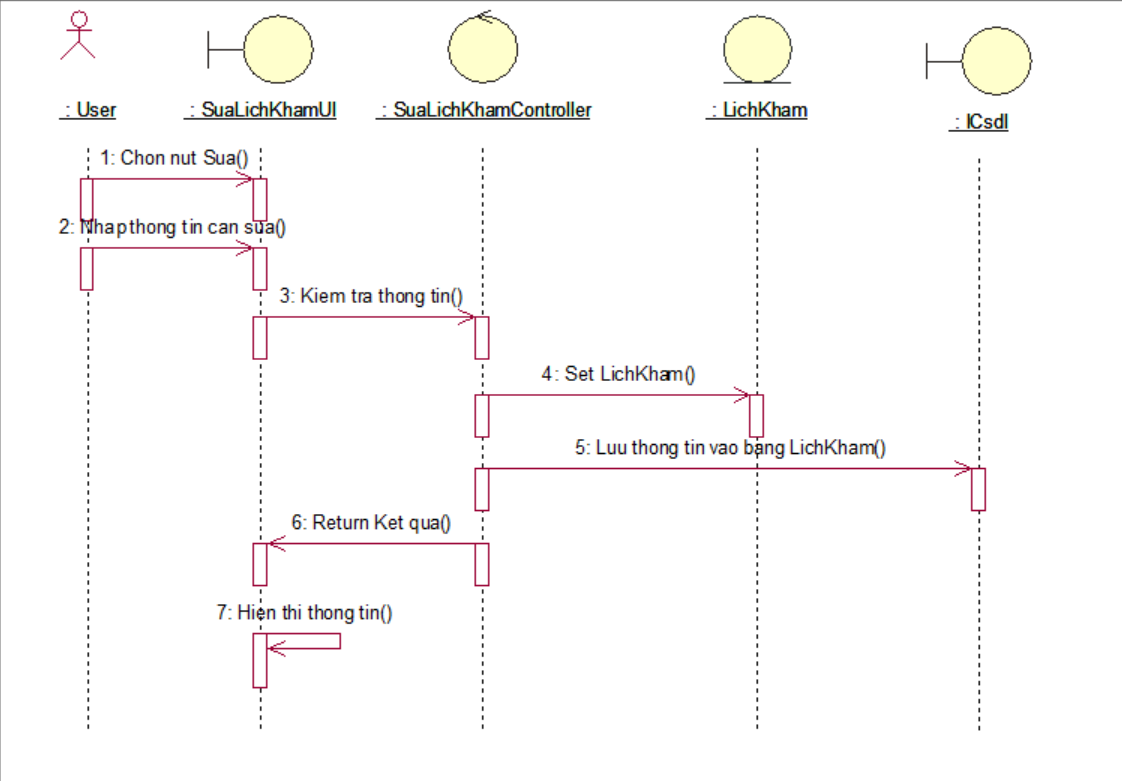
**2.3.3.2 Use case sửa lịch**

* **Biểu đồ lớp**



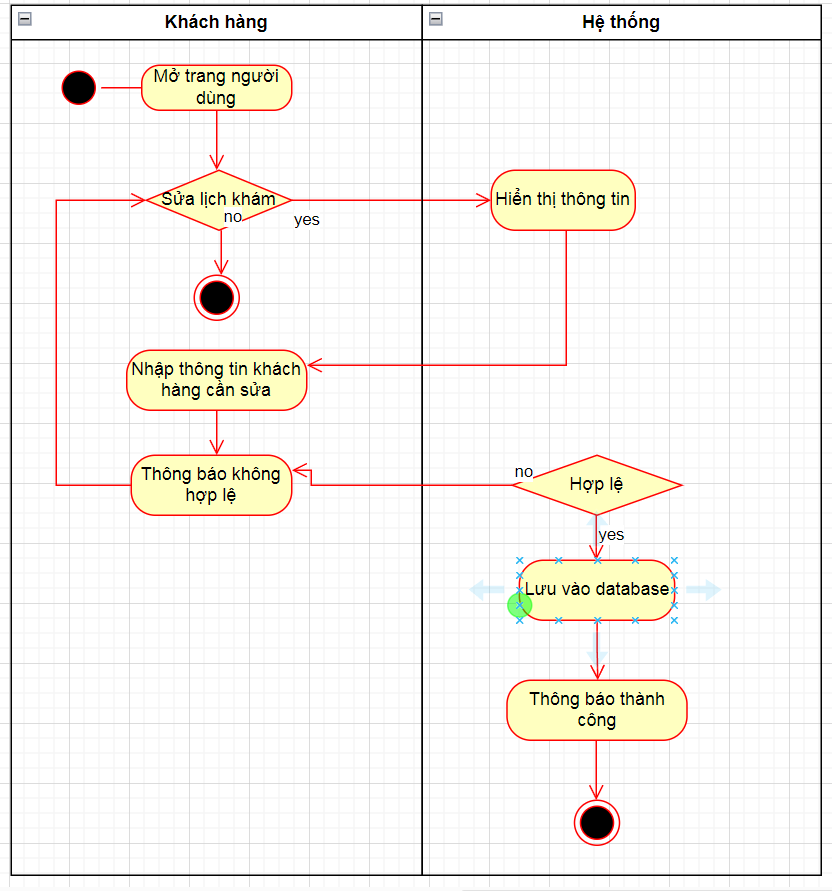
Hình 2. 11 Biểu đồ lớp Use case Sửa lịch

* **Biểu đồ trình tự**



Hình 2. 12 Biểu đồ trình tự Use case Sửa lịch

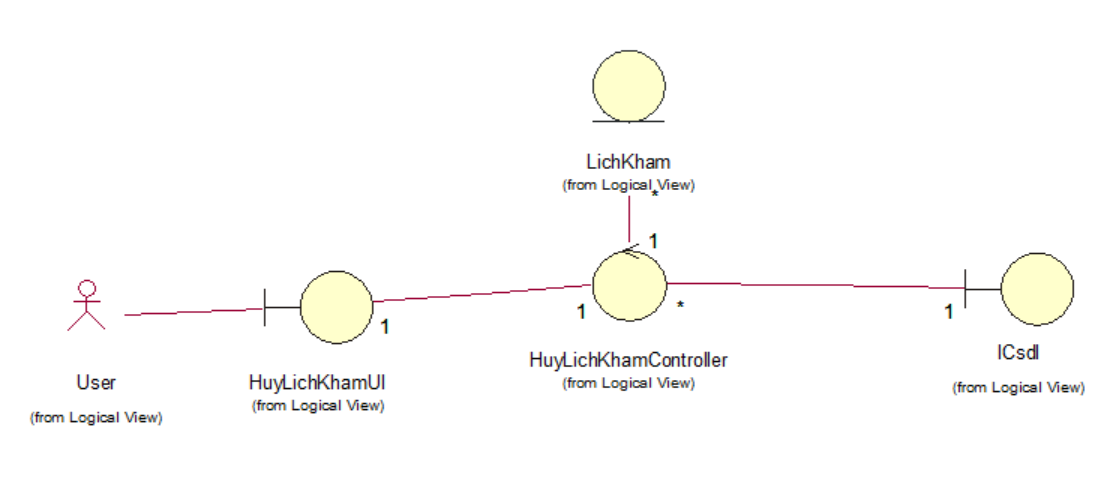
* **Biểu đồ hoạt động**

****

Hình 2. 13 Biểu đồ hoạt động Use case Sửa lịch

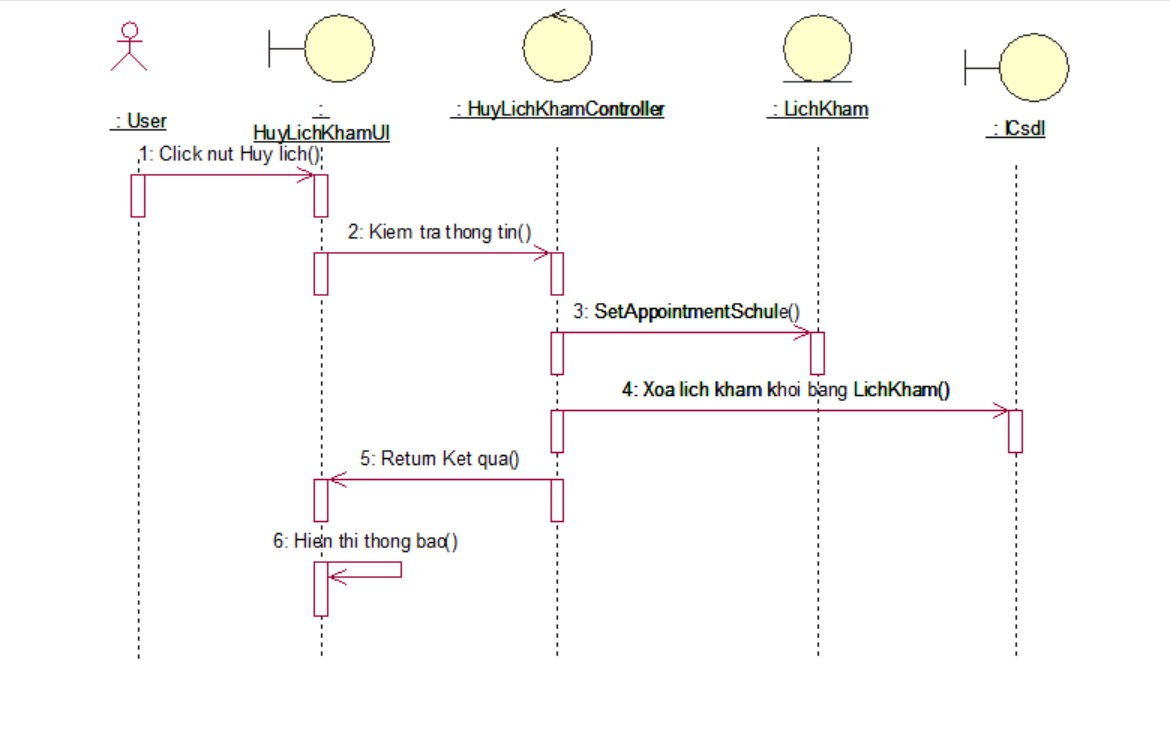
**2.3.3.3 Use case hủy lịch**

* **Biểu đồ lớp**



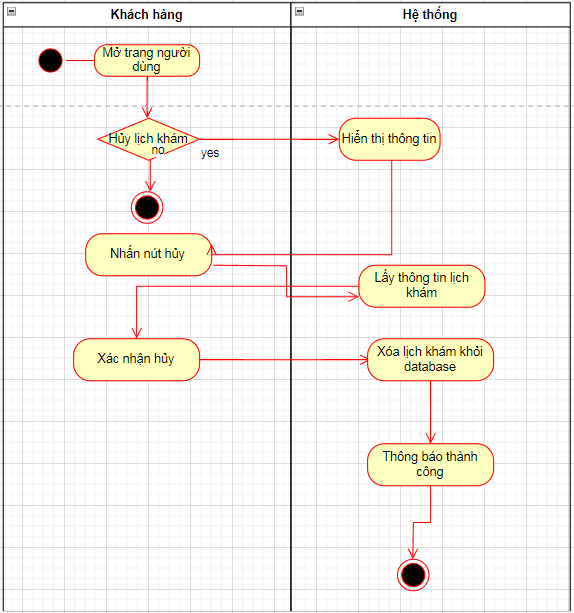
Hình 2. 14 Biểu đồ lớp Use case Hủy lịch

* **Biểu đồ trình tự**



Hình 2. 15 Biểu đồ Use case Hủy lịch

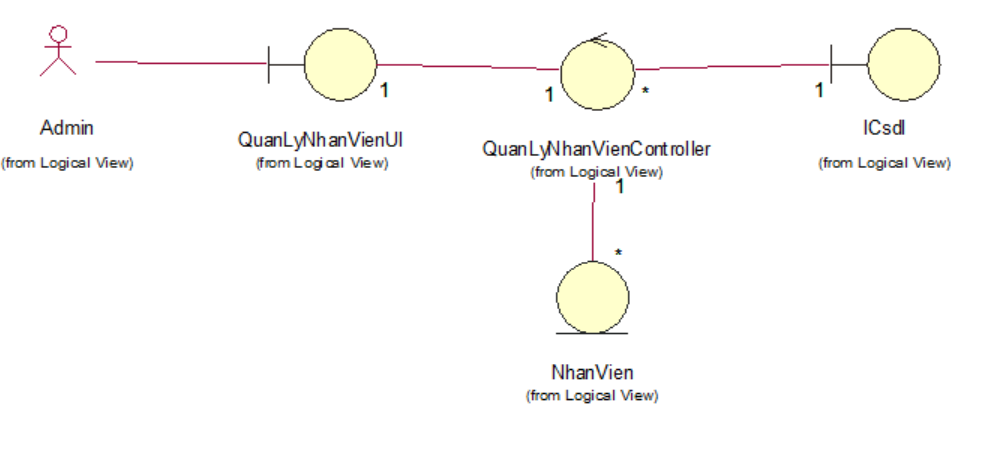
* **Biểu đồ hoạt động**

****

Hình 2. 16 Biểu đồ hoạt động Use case Hủy lịch

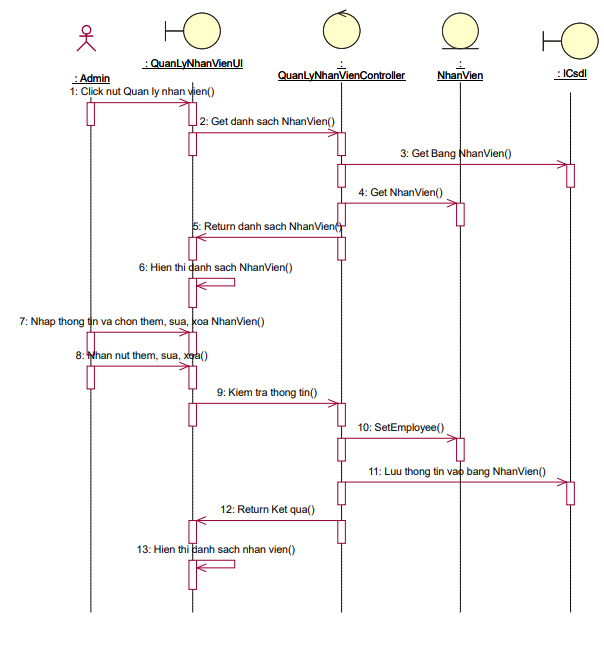
**2.3.3.4 Use case quản lý nhân viên**

* **Biểu đồ lớp**



Hình 2. 17 Biểu đồ lớp Use case Quản lý nhân viên

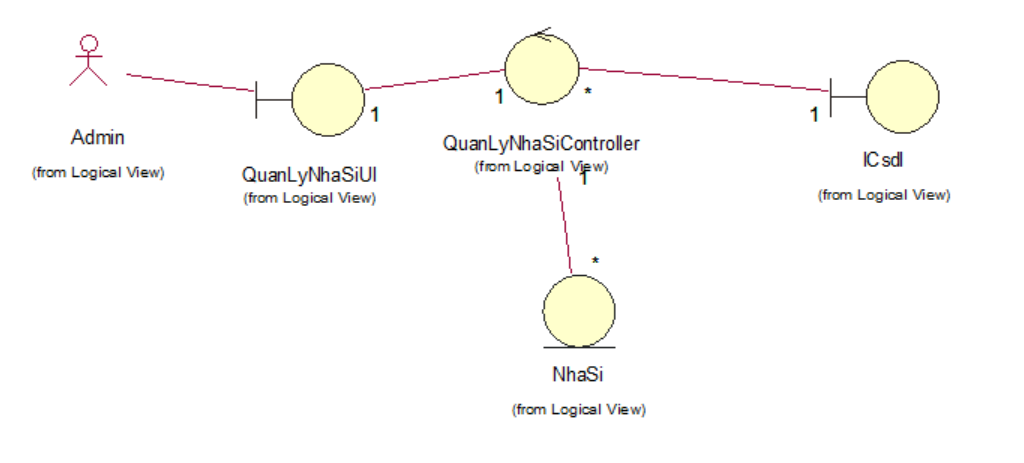
* **Biểu đồ trình tự**



Hình 2. 18 Biểu đồ trình tự Use case Quản lý nhân viên

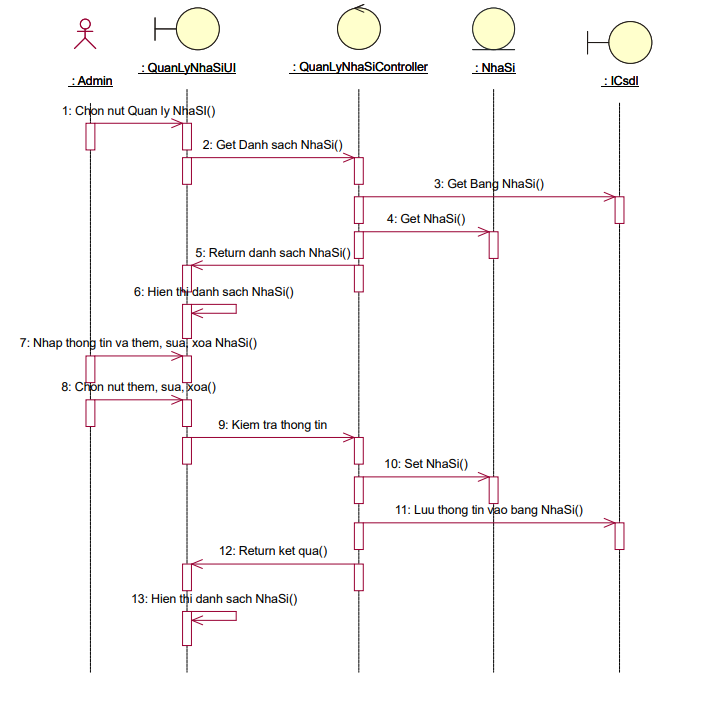
**2.3.3.5 Use case quản lý nha sĩ**

* **Biểu đồ lớp**



Hình 2. 19 Biểu đồ lớp Use case Quản lý nha sĩ

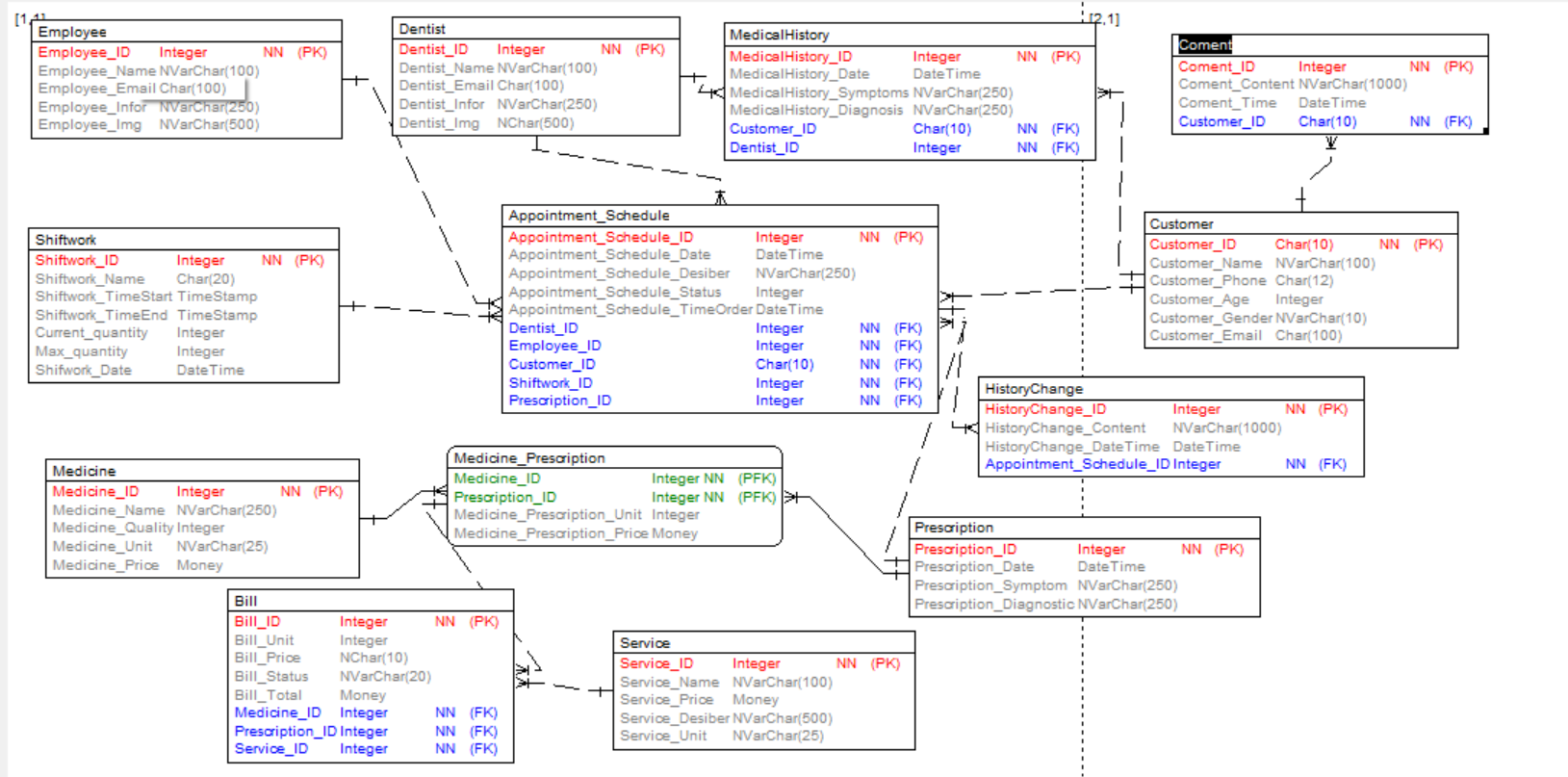
* **Biểu đồ trình tự**



Hình 2. 20 Biểu đồ Use case Quản lý nha sĩ

### **2.3.4 Mô hình hóa dữ liệu**

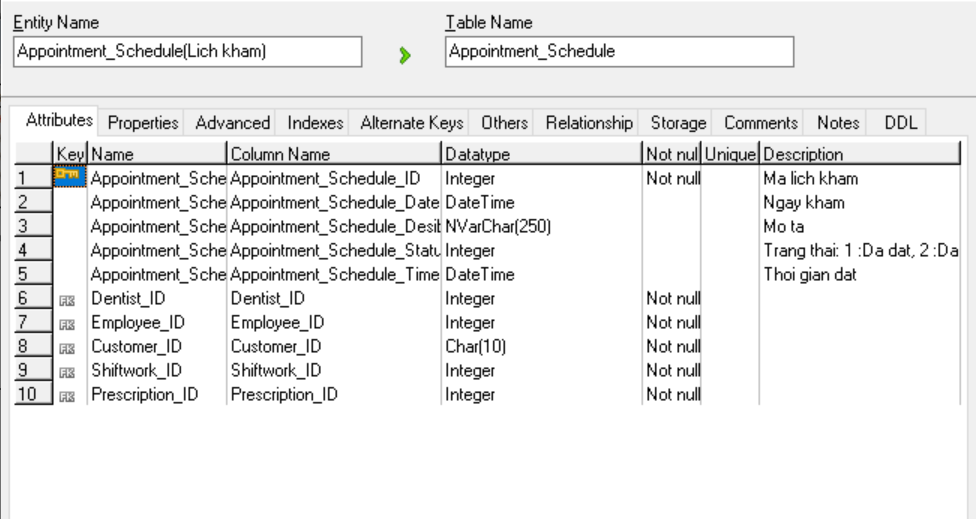
**2.3.4.1 Mô hình cơ sở dữ liệu hệ thống**



Hình 2. 21 Biểu đồ thực thể liên kết

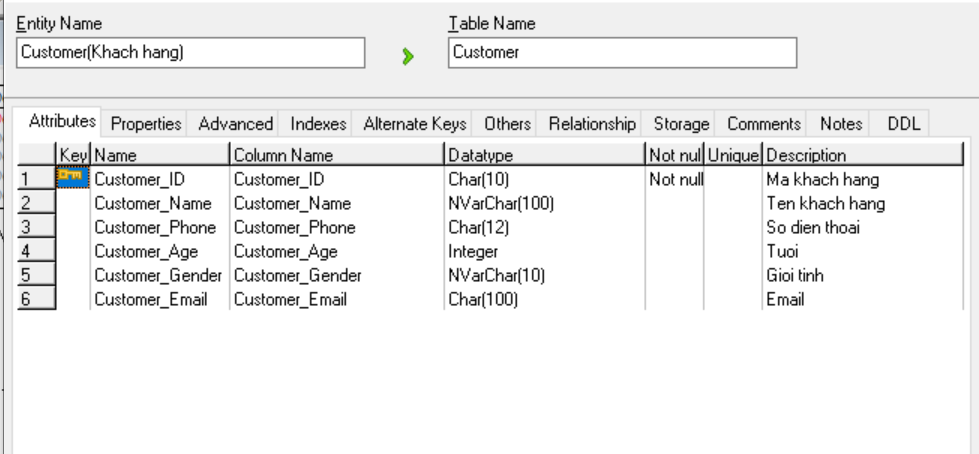
### **2.3.5 Thiết kế các bảng dữ liệu**

**2.3.5.1 Bảng lịch khám**

****

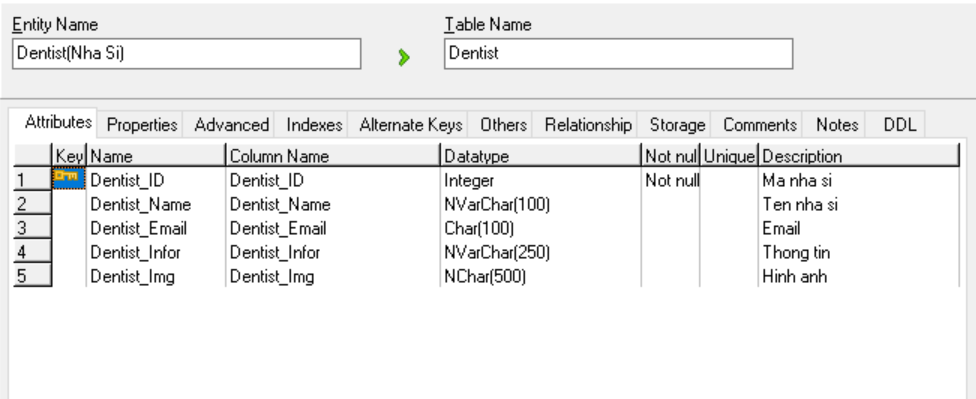
Hình 2. 22 Thiết kế Bảng LichKham

**2.3.5.2 Bảng khách hàng**

****

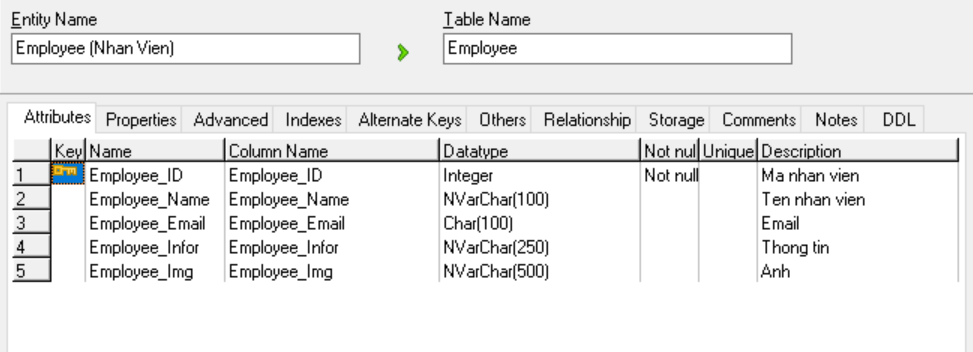
Hình 2. 23 Thiết kế bảng KhachHang

**2.3.5.3 Bảng nha sĩ**

****

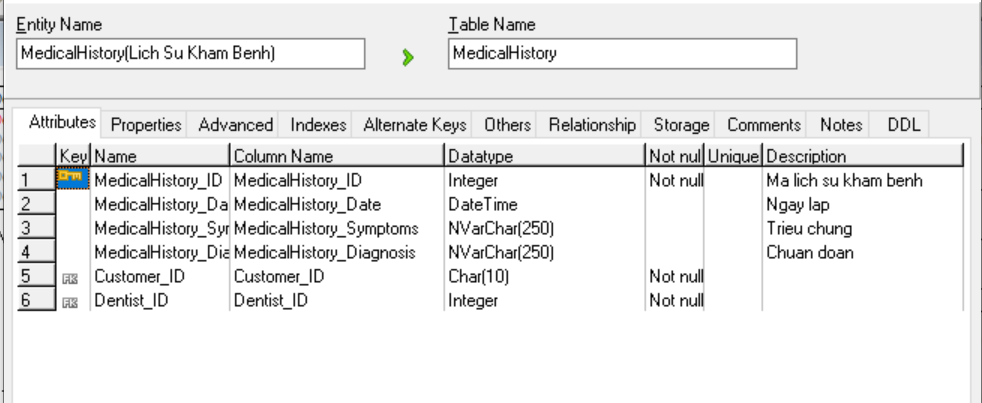
Hình 2. 24 Thiết kế bảng NhaSi

**2.3.5.4 Bảng nhân viên**

****

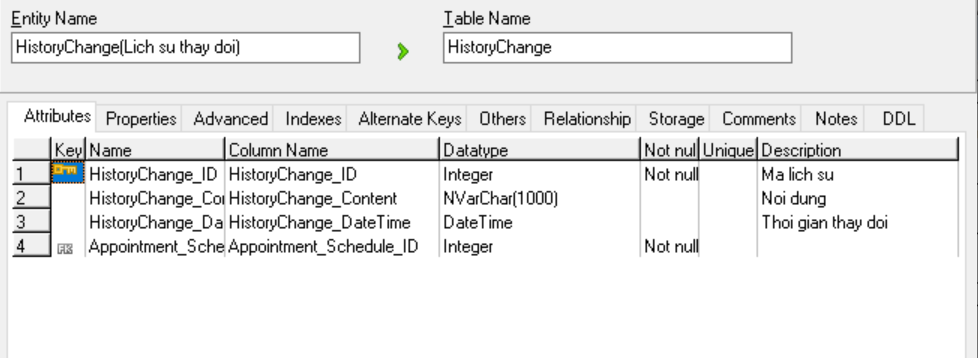
Hình 2. 25 Thiết kế bảng NhanVien

**2.3.5.5 Bảng lịch sử khám bệnh**

****

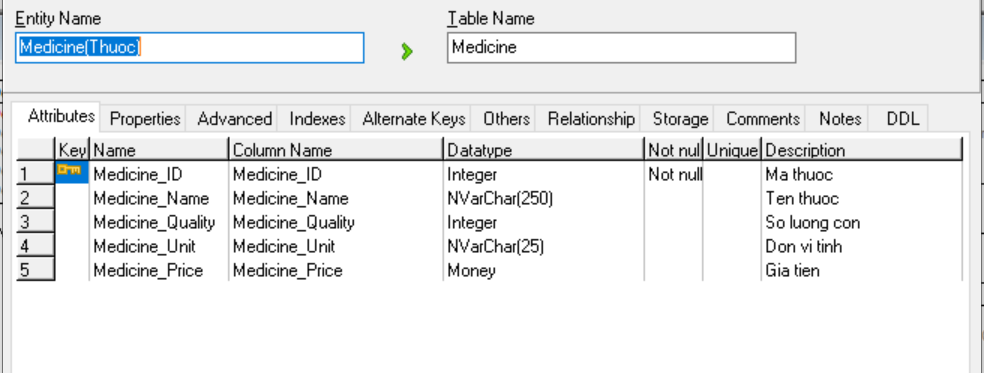
Hình 2. 26 Thiết kế bảng LichSuKhamBenh

**2.3.5.6 Bảng lịch sử thay đổi**

****

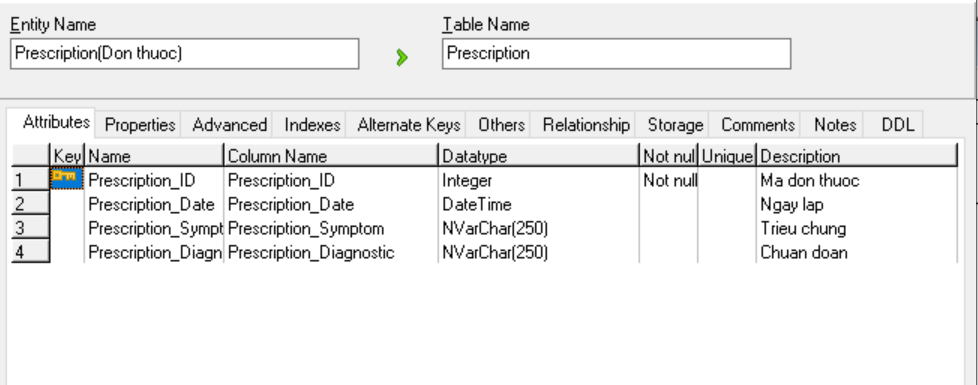
Hình 2. 27 Thiết kế bảng LichSuThayDoi

**2.3.5.7 Bảng thuốc**

****

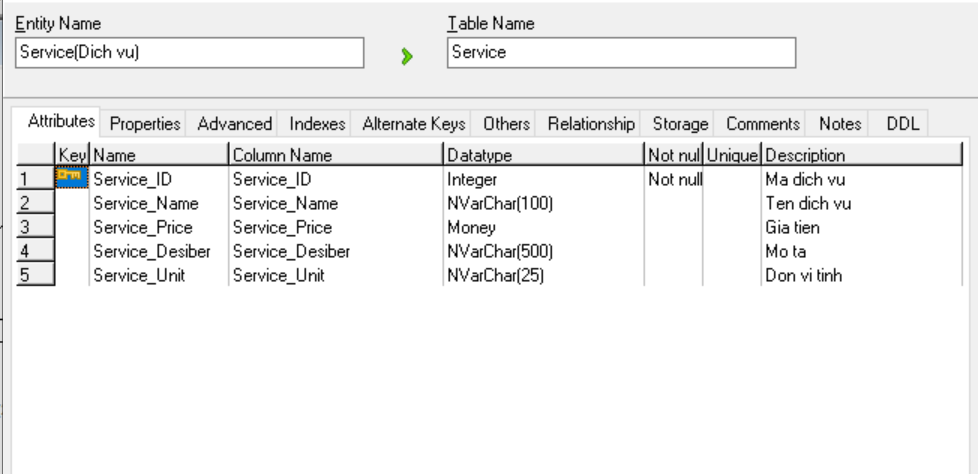
Hình 2. 28 Thiết kế bảng Thuoc

**2.3.5.8 Bảng đơn thuốc**

****

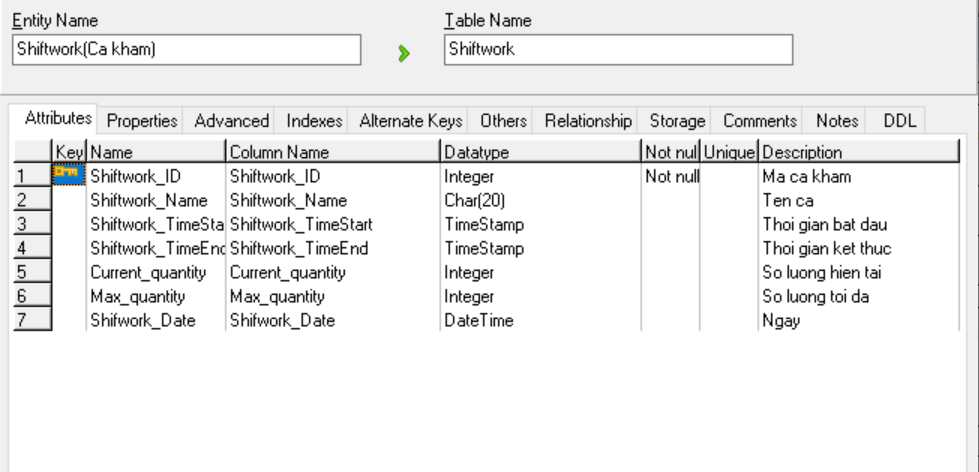
Hình 2. 29 Thiết kế bảng DonThuoc

**2.3.5.9 Bảng dịch vụ**

****

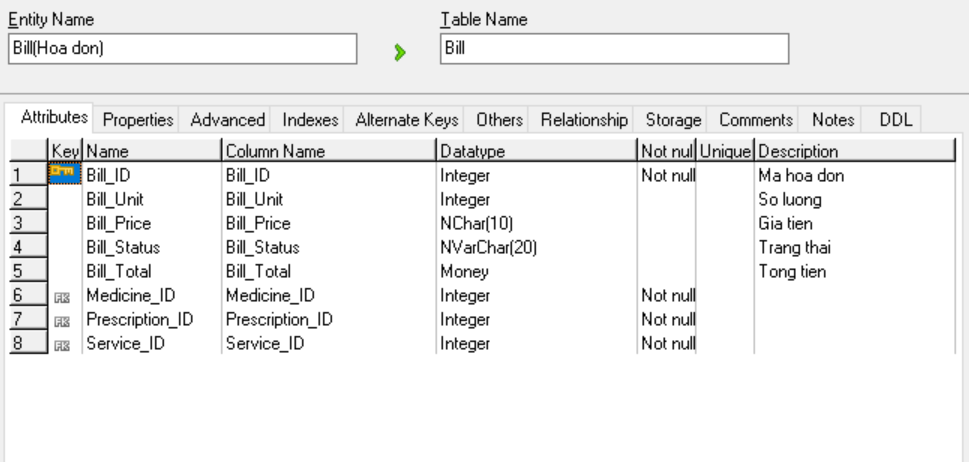
Hình 2. 30 Thiết kế bảng DichVu

**2.3.5.10 Bảng ca khám**

****

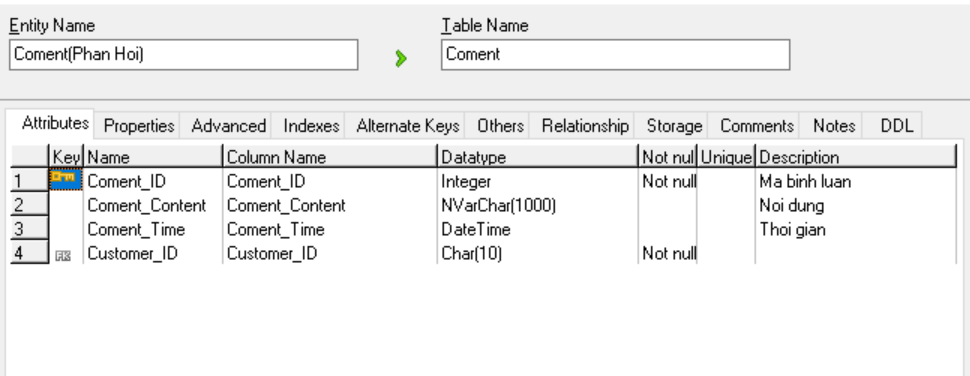
Hình 2. 31 Thiết kế bảng CaKham

**2.3.5.11 Bảng hóa đơn**

****

Hình 2. 32 Thiết kế bảng HoaDon

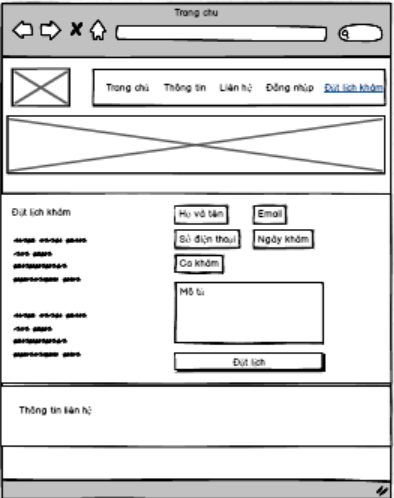
**2.3.5.12 Bảng phản hồi**

****

Hình 2. 33 Thiết kế bảng PhanHoi

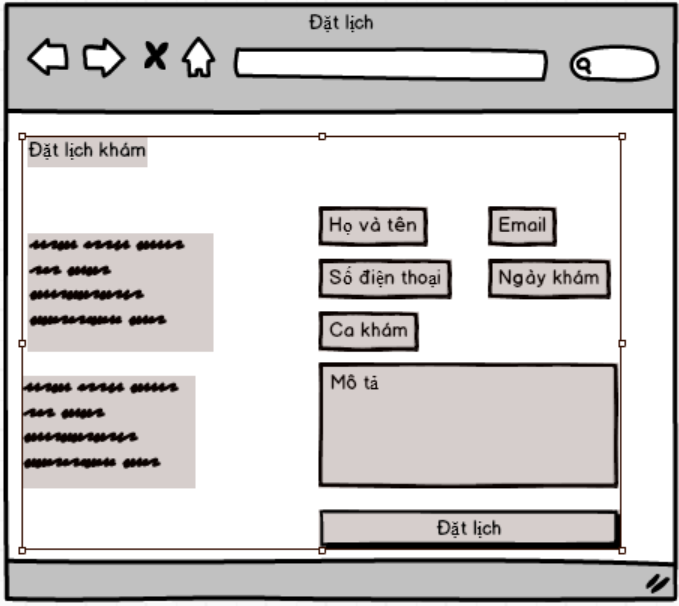
### **2.3.6 Thiết kế giao diện**

**2.3.6.1 Màn hình trang chủ**

****

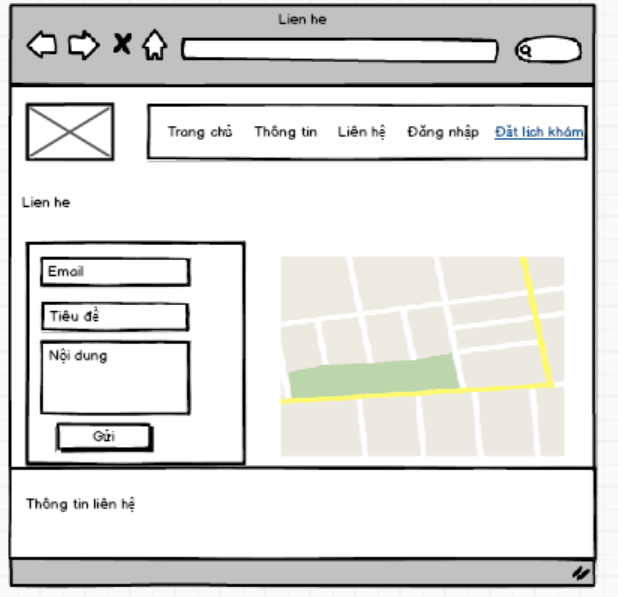
Hình 2. 34 Màn hình trang chủ

**2.3.6.2 Màn hình đặt lịch khám**

****

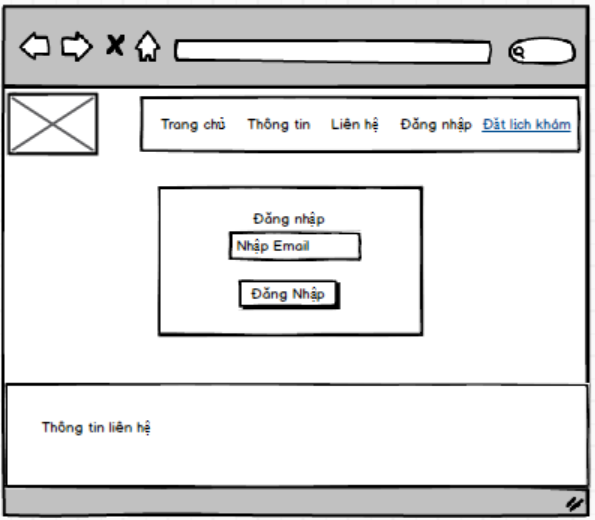
Hình 2. 35 Màn hình đặt lịch khám

**2.3.6.3 Màn hình liên hệ**

****

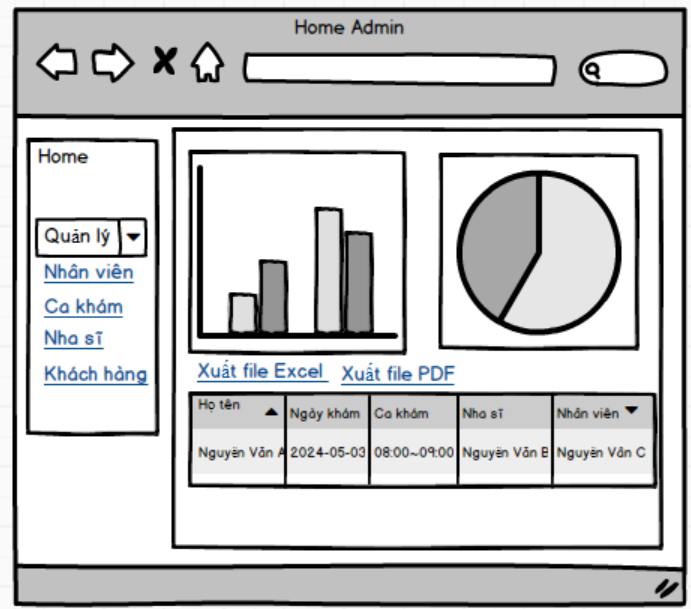
Hình 2. 36 Màn hình liên hệ

**2.3.6.4 Màn hình đăng nhập**

****

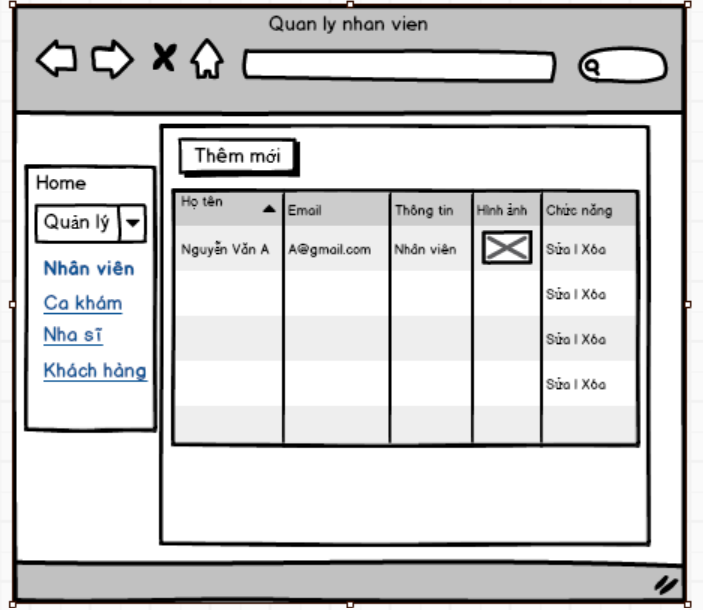
Hình 2. 37 Màn hình đăng nhập

**2.3.6.5 Màn hình trang Admin**

****

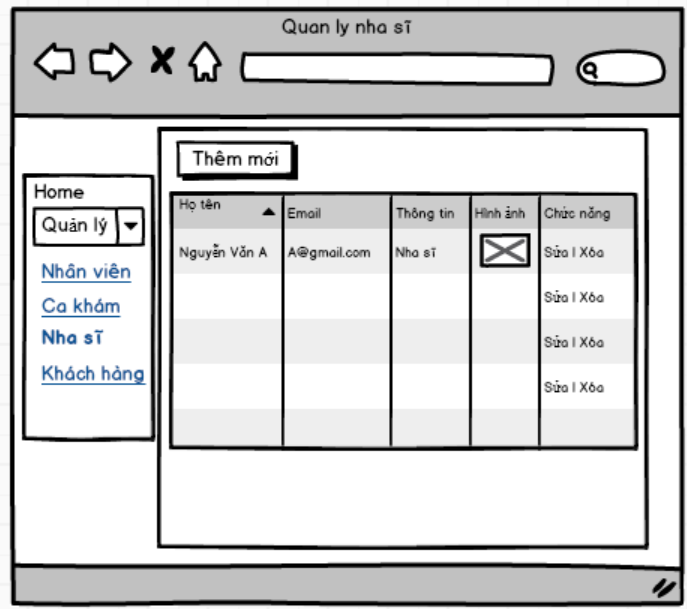
Hình 2. 38 Màn hình trang Admin

**2.3.6.6 Màn hình quản lý nhân viên**

****

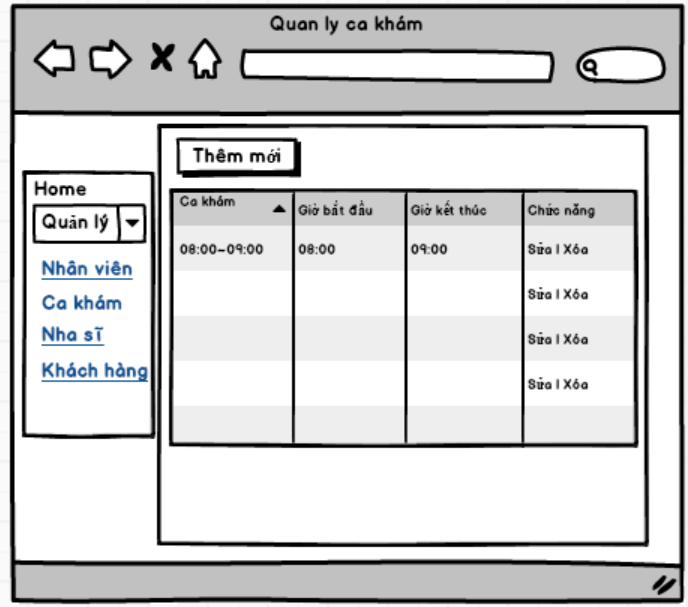
Hình 2. 39 Màn hình quản lý nhân viên

**2.3.6.7 Màn hình quản lý nha sĩ**

****

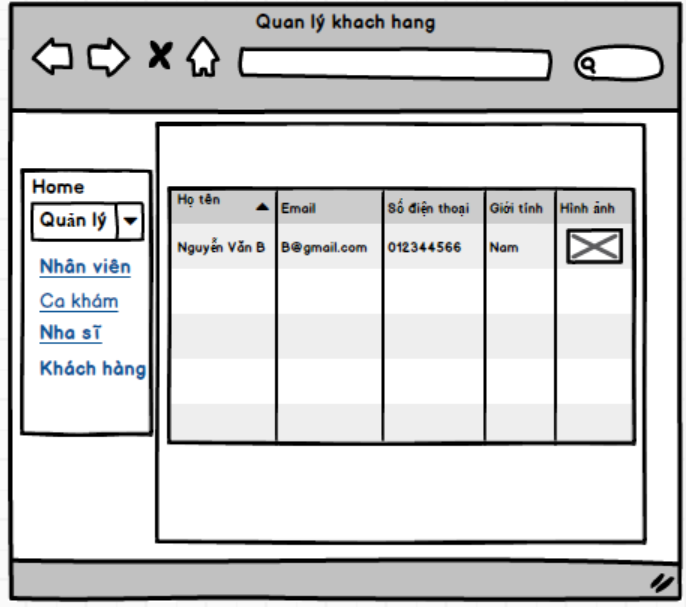
Hình 2. 40 Màn hình quản lý nha sĩ

**2.3.6.8 Màn hình quản lý ca khám**

****

Hình 2. 41 Màn hình quản lý ca khám

**2.3.6.9 Màn hình quản lý khách hàng**

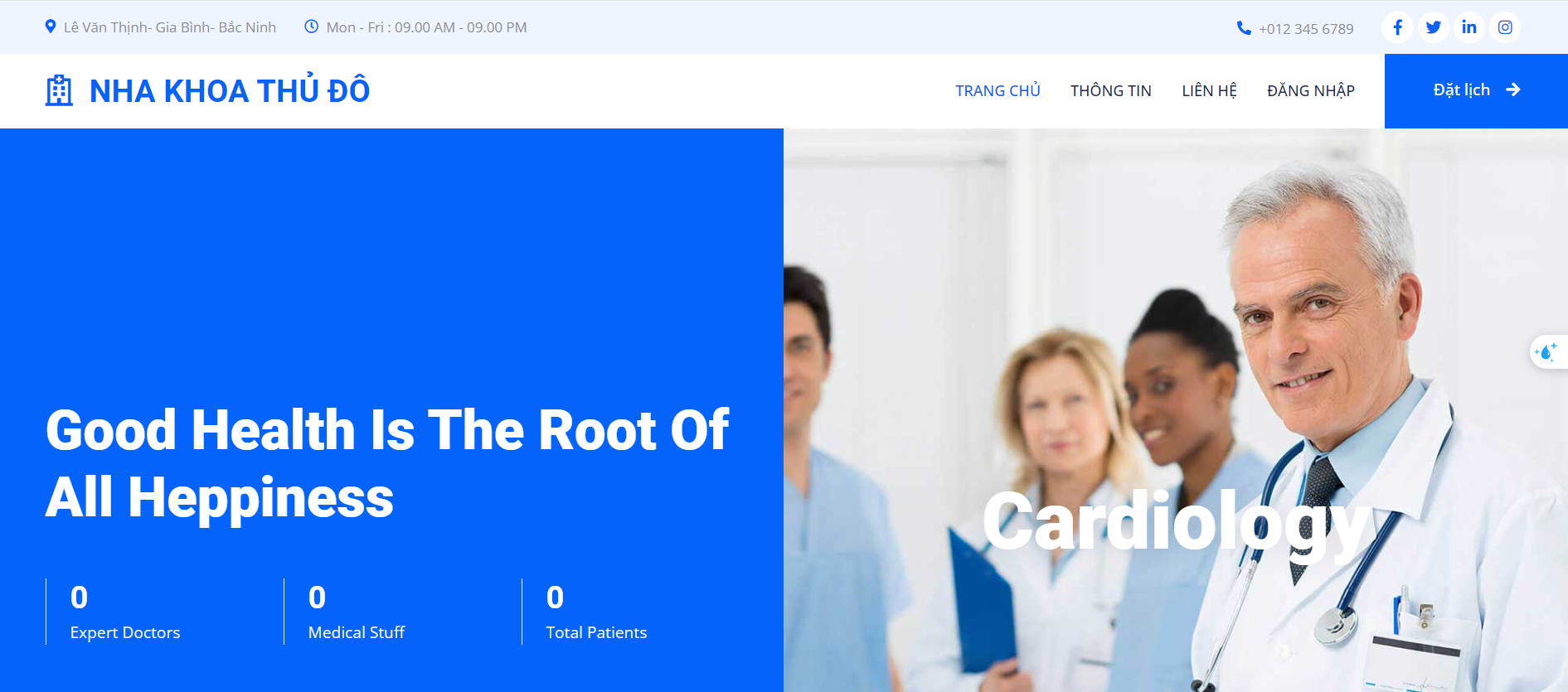


Hình 2. 42 Màn hình quản lý khách hàng

# **CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC.**

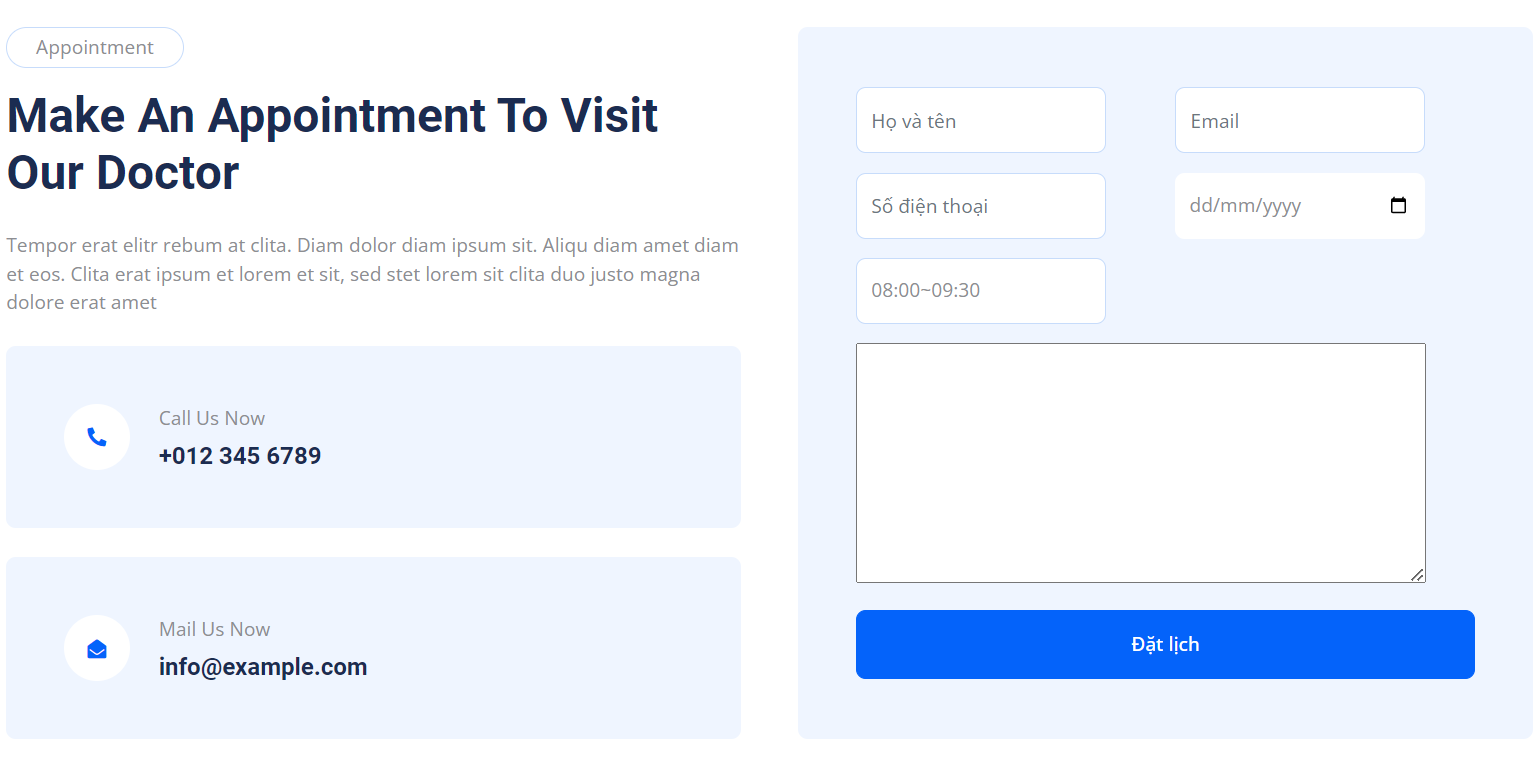
## **3.1. Kết quả đạt được.**

* **Trang chủ**

****

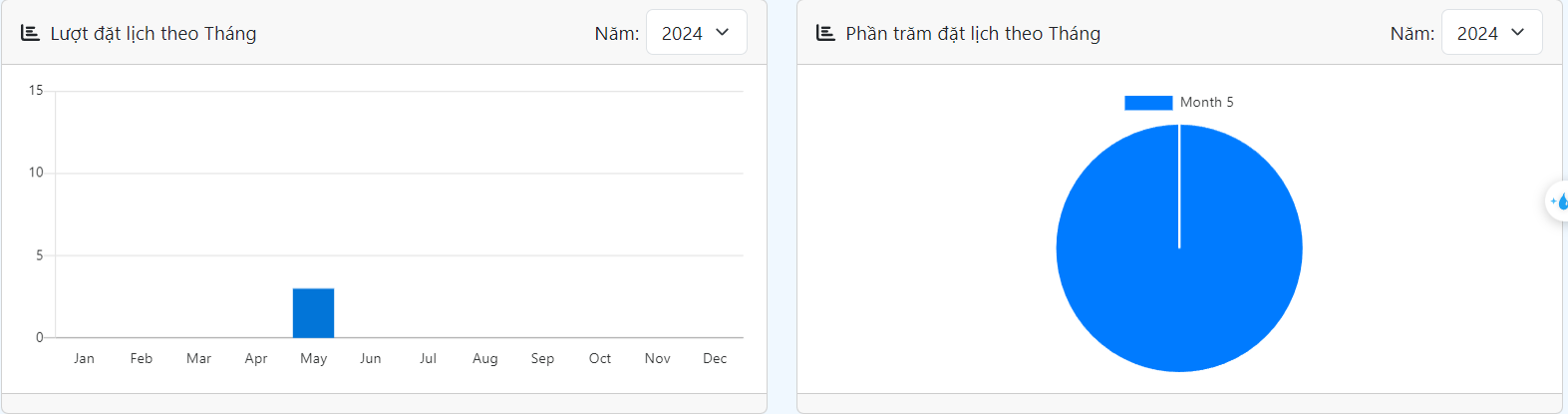
Hình 3. 1 Giao diện trang chủ

* **Đặt lịch**

****

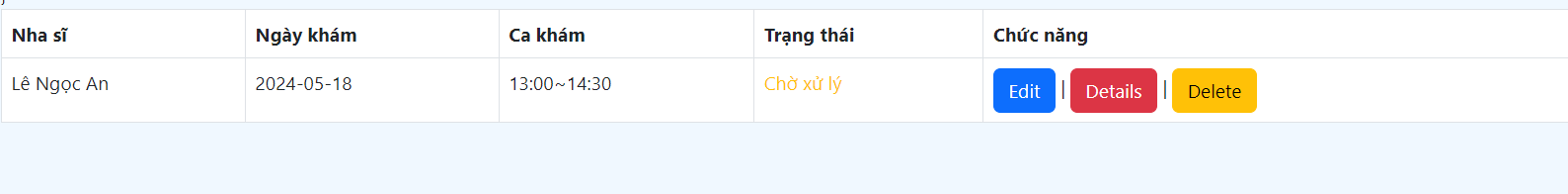
Hình 3. 2 Giao diện đặt lịch

* **Biểu đồ**

****

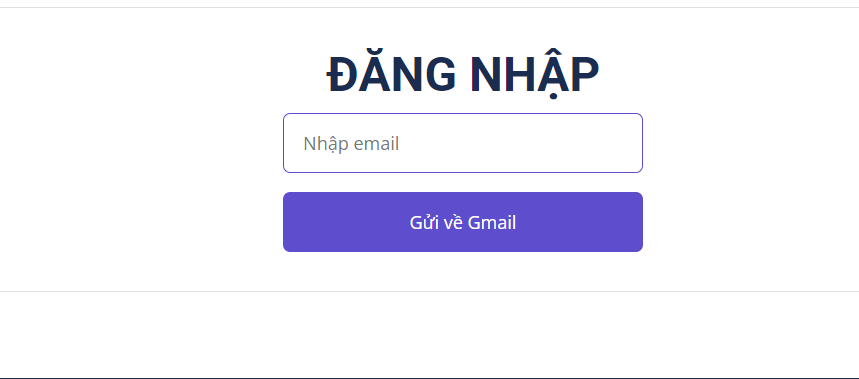
Hình 3. 3 Giao diện biểu đồ

* **Lịch khám người dùng**

****

Hình 3. 4 Giao diện lịch khám người dùng

* **Đăng nhập**

****

Hình 3. 5 Giao diện đăng nhập

## **3.2. Kiểm thử chức năng hệ thống.**

### **3.2.1 Kiểm thử chức năng phía người dùng.**

Hình 3. 6 Bảng kiểm thử chức năng phía người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Case** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| 1 | Kiểm tra chức năng đăng nhập | Nhập Email & mã OTP hợp lệ | Đăng nhập thành công, chuyển sang trang Người dùng | Pass |
| Nhập Email & mã OTP không hợp lệ | Đăng nhập không thành công | Pass |
| 2 | Kiểm tra chức năng đặt lịch khám | Nhập thông tin hợp lệ | Đặt lịch thành công, thông báo ra màn hình | Pass |
| Nhập thông tin không hợp lệ | Đặt lịch không thành công | Pass |
| 3 | Kiểm tra chức năng sửa lịch khám | Nhập thông tin hợp lệ | Sửa lịch khám thành công | Pass |
| Nhập thông tin không hợp lệ | Sửa lịch khám không thành công | Pass |
| 4 | Kiểm tra chức năng xóa lịch khám | Đồng ý xóa | Xóa lịch khám thành công | Pass |
| Hủy xóa | Xóa lịch khám không thành công | Pass |
| 5 | Kiểm tra chức năng sửa thông tin cá nhân | Nhập thông tin hợp lệ | Sửa thông tin cá nhân thành công | Pass |
| Nhập thông tin không hợp lệ | Sửa thông tin các nhân không thành công | Pass |

### **3.2.2 Kiểm thử chức năng phía người quản lý.**

Hình 3. 7 Bảng kiểm thử chức năng phía người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Case** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| 1 | Kiểm tra chức năng thêm nha sĩ | Nhập thông tin hợp lệ | Thêm nha sĩ thành công, chuyển đến danh sách nha sĩ | Pass |
| Nhập thông tin không hợp lệ | Thêm nha sĩ không thành công | Pass |
| 2 | Kiểm tra chức năng sửa thông tin nha sĩ | Nhập thông tin hợp lệ | Sửa thông tin thành công, chuyển đến danh sách nha sĩ | Pass |
| Nhập thông tin không hợp lệ | Sửa thông tin không thành công | Pass |
| 3 | Kiểm tra chức năng xóa nha sĩ | Đồng ý xóa | Xóa nha sĩ thành công | Pass |
| Hủy xóa | Xóa nha sĩ không thành công | Pass |
| 4 | Kiểm tra chức năng thêm nhân viên | Nhập thông tin hợp lệ | Thêm nhân viên thành công, chuyển đến danh sách nhân viên | Pass |
| Nhập thông tin không hợp lệ | Thêm nhân viên không thành công | Pass |
| 5 | Kiểm tra chức năng sửa thông tin nhân viên | Nhập thông tin hợp lệ | Sửa thông tin thành công, chuyển đến danh sách nhân viên | Pass |
| Nhập thông tin không hợp lệ | Sửa thông tin không thành công | Pass |
| 6 | Kiểm tra chức năng xóa nhân viên | Đồng ý xóa | Xóa nhân viên thành công | Pass |
| Hủy xóa | Xóa nhân viên không thành công | Pass |

# **KẾT LUẬN**

- Những kết quả đạt được

+ Về công nghệ:

\* Hiểu được quá trình thiết kế một ứng dụng thực tế, đi từ những bước cơ bản: khảo sát, phân tích, thiết kế, thực hiện, kiểm thử,…

\* Xây dựng thành công website đặt lịch khám bệnh đáp ứng nhu cầu đặt lịch khám bệnh trực tuyến của đa số khách hàng.

\* Nắm được các kiến thức xây dựng phần mềm.

\* Sử dụng thành thạo các công cụ hỗ trợ: Rational Rose, Draw.io,…

+ Về chương trình:

\* Cho phép khách hàng có thể đặt lịch trực tiếp.

\* Thống kê, cập nhật số liệu chính xác.

- Hướng phát triển

+ Hỗ trợ đa nền tảng.

+ Phát triển bản thương mại.

+ Phần báo cáo hoàn thiện chuyên sâu hơn.

# **TÀI LIỆU THAM KHẢO**

[1] Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga(2015) – *Giáo trình hệ quản trị cơ sở dữ liệu SQL,* NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Kim Phượng(2011) – *Giáo trình phân tích thiết kế hệ thống,* NXB Giáo dục Việt Nam.

[3] Nhà xuất bản Thông tin và truyền thông, Ths. Lê Văn Hùng – *Cơ sở dữ liệu quan hệ và công nghệ phân tích – thiết kế*.

[4] Tìm hiểu về .NET: <https://dotnet.microsoft.com/> ,truy cập cuối cùng ngày 10/05/2024.

[5] Tìm hiểu về mô hình MVC: <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/>, truy cập cuối cùng ngày 09/05/2024.

[6] Tìm hiểu về quy trình thiết kế website: <https://arena.fpt.edu.vn/quy-trinh-thiet-ke-website/>, truy cập cuối cùng ngày 10/05/2024.