

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*\*\*

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

**QUẢN LÝ PHÒNG KHÁM**

Mã đồ án: **XXXX**

Sinh viên thực hiện**: Đặng Quang Huy**

Mã sinh viên: **94365**

Lớp: **65PM4**

Giảng viên hướng dẫn: **Ths.** **Nguyễn Hồng Hạnh**

HÀ NỘI 04/2024

# LỜI CẢM ƠN

Để đạt được kết quả như ngày hôm nay, bên cạnh việc bản thân nỗ lực phấn đấu trau dồi kiến thức thì không thể thiếu được sự quan tâm, chỉ bảo tận tình và những lời động viên chân thành và quý báu từ phía Thầy Cô, gia đình, bạn bè.

Em xin chân thành cảm ơn Cô Nguyễn Hồng Hạnh, là người đã tận tình hướng dẫn em, giúp em giải quyết các vấn đề, chỉ bảo em trong suốt thời gian thực hiện đề tài.

Em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Bộ môn Công nghệ phần mềm - Khoa Công nghệ Thông tin Trường Đại học Xây dựng Hà Nội đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu trong những năm học vừa qua đồng thời đã tạo điều kiện tốt cho em thực hiện đề tài này.

Mặc dù đã rất cố gắng hoàn thành ĐATN này trong phạm vi và khả năng cho phép, nhưng chắc chắn em sẽ không thể tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô.

*Hà Nội ,Ngày…Tháng…Năm…*

Sinh viên

*Đặng Quang Huy*

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Đặng Quang Huy

Điện thoại liên lạc: 0835317153 Email: [huy94365@huce.edu.vn](mailto:trinhminhduc2310@gmail.com) Lớp: 65PM4 Hệ đào tạo: Chính quy Năm tốt nghiệp: 2024 Đồ án tốt nghiệp được thực hiện tại: Đại học Xây dựng Hà Nội

Thời gian làm ĐATN: Từ ngày 01/10/2023 đến 31/12/2023

1. Mục đích nội dung của ĐATN

Xây dựng Hệ thống Quản lý Phòng Khám cho Phòng khám tư nhân – MyClinic.

1. Các nhiệm vụ cụ thể của ĐATN
   1. Tiếp nhận bài toán nghiệp vụ
   2. Nghiên cứu quy trình nghiệp vụ
   3. Tìm hiểu kiến trúc hệ thống tổng thể
   4. Phân tích thiết kế hệ thống
   5. Xây dựng hệ thống:

Hệ thống bao gồm các chức năng chính sau:

* + - Quản lý Đợt đồ án, Giáo viên, Sinh viên
    - Quản lý việc phân Giáo viên hướng dẫn cho Sinh viên
    - Sinh viên đăng kí nguyện vọng chọn Giáo viên hướng dẫn, Giáo viên chọn sinh viên đăng kí
    - Cập nhật đề tài của sinh viên, đánh giá quá trình thực hiện đồ án của sinh viên
    - Quản lý phân Giáo viên phản biện cho Sinh viên, đánh giá quá trình phản biện đồ án của sinh viên
    - Quản lý việc phân Hội đồng bảo vệ cho sinh viên, chấm điểm Hội đồng bảo vệ

1. Lời cam đoan của sinh viên:

Tôi – *Đặng Quang Huy* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của ThS. Nguyễn Hồng Hạnh

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

*Hà Nội, ngày tháng năm*

Tác giả ĐATN

*Đặng Quang Huy*

1. Xác nhận của giảng viên hướng dẫn về mức độ hoàn thành ĐATN và cho phép bảo vệ

………………………………………………………………………………………………

…………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………

……………………………………………………………………………………………

*Hà Nội, ngày tháng năm*

Giảng viên hướng dẫn

*ThS. Nguyễn Hồng Hạnh*

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc164161017)

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 3](#_Toc164161018)

[MỤC LỤC 5](#_Toc164161019)

[DANH MỤC HÌNH ẢNH 7](#_Toc164161020)

[GIẢI THÍCH THUẬT NGỮ, TỪ VIẾT TẮT 10](#_Toc164161021)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 12](#_Toc164161022)

[1. Cơ sở thực tiễn 12](#_Toc164161023)

[2. Mục tiêu đồ án tốt nghiệp 12](#_Toc164161024)

[3. Đề xuất giải pháp 12](#_Toc164161025)

[4. Công nghệ sử dụng 15](#_Toc164161026)

[4.1. MongoDB 15](#_Toc164161027)

[4.2. Node.js 16](#_Toc164161028)

[4.3. Express.js 17](#_Toc164161029)

[4.4. HTML 17](#_Toc164161030)

[4.5. CSS 18](#_Toc164161031)

[4.6. Bootstrap 19](#_Toc164161032)

[4.7. Các thư viện hỗ trợ khác 19](#_Toc164161033)

[4.8. GIT 19](#_Toc164161034)

[4.9. Visual Studio Code 20](#_Toc164161035)

[CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG 21](#_Toc164161036)

[1. Khái quát bài toán 21](#_Toc164161037)

[1.1. Mô tả sơ lược 21](#_Toc164161038)

[1.2. Nhu cầu quản lý 22](#_Toc164161039)

[1.3. Phạm vi 22](#_Toc164161040)

[1.4. Đối tượng 22](#_Toc164161041)

[2. Xác định yêu cầu nghiệp vụ 22](#_Toc164161042)

[2.1. Quản lý Thông Tin Bệnh Nhân 22](#_Toc164161043)

[2.2. Quản lý Thông Tin Bác Sĩ 22](#_Toc164161044)

[2.3. Quản lý Thông Tin Thuốc và Dịch Vụ 23](#_Toc164161045)

[2.4. Quản lý Lịch Hẹn Khám Bệnh 23](#_Toc164161046)

[2.5. Quản lý Hồ Sơ Bệnh Án 23](#_Toc164161047)

[2.6. Quản lý Tài Chính 23](#_Toc164161048)

[2.7. Quản lý Báo Cáo 23](#_Toc164161049)

[3. Phân tích yêu cầu hệ thống 23](#_Toc164161050)

[3.1. Yêu cầu chức năng 23](#_Toc164161051)

[3.2. Yêu cầu phi chức năng 23](#_Toc164161052)

[4. Phân tích thiết kế 24](#_Toc164161053)

[4.1. Biểu đồ UseCase 24](#_Toc164161054)

[4.2. Biểu đồ Sequence 30](#_Toc164161055)

[4.3. Biểu đồ Lớp 31](#_Toc164161056)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 33](#_Toc164161057)

[1. Mô hình kiến trúc hệ thống 33](#_Toc164161058)

[2. Thiết kế Cơ sở dữ liệu Danh sách các Bảng 34](#_Toc164161059)

[3. Thiết kế Giao diện hệ thống 34](#_Toc164161060)

[3.1. Giao diện dành cho Bác sĩ 34](#_Toc164161061)

[3.2. Giao diện dành cho Bệnh nhân 34](#_Toc164161062)

[CHƯƠNG 4: XÂY DỰNG VÀ CÀI ĐẶT HỆ THỐNG 35](#_Toc164161063)

[CHƯƠNG 5: TỔNG KẾT 37](#_Toc164161064)

[1. Kết quả đạt được 37](#_Toc164161065)

[2. Những hạn chế tồn tại 37](#_Toc164161066)

[3. Hướng phát triển hoàn thiện hệ thống 37](#_Toc164161067)

[TÀI LIỆU THAM KHẢO 38](#_Toc164161068)

# DANH MỤC HÌNH ẢNH

[Hình 1 Biểu đồ UseCase Tổng quát 34](#_bookmark44)

# GIẢI THÍCH THUẬT NGỮ, TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Ý nghĩa** |
| 1 | GVHD | Giáo viên hướng dẫn |
| 2 | GVPB | Giáo viên phản biện |
| 3 | SV | Sinh viên |
| 4 | GV | Giáo viên |
| 5 | DATN | Đồ án tốt nghiệp |
| 6 | HD | Hội đồng |
| 7 | HDTN | Hội đồng tốt nghiệp |

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Xây dựng Hệ thống Quản lý quy trình giao nhận đồ án tốt nghiệp cho Bộ môn Công nghệ Phần mềm bao gồm các nội dung chính:

1. Tiếp nhận bài toán nghiệp vụ
2. Nghiên cứu quy trình nghiệp vụ
3. Tìm hiểu kiến trúc hệ thống tổng thể
4. Phân tích thiết kế hệ thống
5. Xây dựng hệ thống: bao gồm các chức năng chính:
   * Quản lý Thuốc, Dịch vụ, Lịch làm việc.
   * Quản lý Lịch hẹn khám của bệnh nhân.
   * Bệnh nhân đặt lịch hẹn khám với bác sĩ mình chọn, nếu có thay đổi thì có thể huỷ lịch hẹn.
   * Bác sĩ có thể chấp nhận hoặc huỷ lịch hẹn của bệnh nhân.
   * Cập nhật danh sách bệnh nhân trong ngày.
   * Quản lý các hồ sơ bệnh án

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## Cơ sở thực tiễn

Hiện nay xã hội ngày càng phát triển, trong khi môi trường sống ngày càng ô nhiễm, sức khỏe con người cũng đối mặt với nhiều nguy cơ bệnh tật hơn. Minh chứng là những dịch bệnh trong các năm gần đây như Covid-19, bệnh đậu mùa khỉ, … Vì nhu cầu khám chữa bệnh hiện nay rất lớn, nhưng số lượng bệnh viện lại không đủ đáp ứng nhu cầu đó, nên ngày càng nhiều phòng khám tư được mở ra. Trong lĩnh vực quản lý phòng khám, Công nghệ thông tin cũng đóng vai trò quan trọng trong việc tối ưu hóa quy trình làm việc và cải thiện hiệu suất. Quy trình tiếp đón bệnh nhân, đặc biệt là tại phòng khám, có thể được tối ưu hóa thông qua việc áp dụng các giải pháp tin học công nghệ.Từ thực tế đó, em đã thực hiện phân tích và thiết kế một chương trình quản lý quy trình giao nhận đồ án tốt nghiệp

## Mục tiêu đồ án tốt nghiệp

Xây dựng phần mềm quản lý Quy trình giao nhận đồ án tốt nghiệp cho Bộ môn Công nghệ Phần mềm nhằm năng cao hiệu quả quản lý, giảm thời gian công sức, tập trung hoá các nguồn thông tin bằng việc cho sinh viên chủ động đăng kí nguyện vọng chọn GVHD; hỗ trợ bộ môn phân chia GVHD, GVPB và Hội đồng, thống kê tổng hợp,…

Xây dựng phần mềm quản lý phòng khám cho bệnh viện tư nhân nhằm nâng cao hiệu quả quản lý, giảm tải thời gian công sức, tập trung vào việc khám chữa bệnh để có thể nâng cao hiệu suất khám bệnh cho các bệnh nhân bằng việc hỗ trợ bệnh nhân đăng ký lịch hẹn khám; hỗ trợ bác sĩ việc lập các phiếu khám bệnh, quản lý được lịch làm việc của mình, quản lý thuốc, quản lý dịch vụ,…

## Đề xuất giải pháp

Để giải quyết bài toán nêu trên, sinh viên thực hiện đồ án tốt nghiệp đề xuất xây dựng giải pháp phần mềm dựa trên nền web sử dụng Node.js, Express.js để xây dựng hệ thống, sử dụng hệ quản trị cơ sở dữ liệu MongoDB để lưu trữ dữ liệu. Đây đều là giải pháp công nghệ có tốc độ thực thi nhanh, dễ dài cài đặt triển khai.

Hệ thống phục vụ cho công tác quản lý phòng khám cho phòng khám tư nhân  
có các chức năng chính sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên yêu cầu chức năng** | **Mô tả** | **Đối tượng sử dụng** |
| **1** | **Quản lý**  **Thuốc** | Xem, sửa, xoá thông tin | Bác sĩ |
| **2** | **Quản lý**  **Dịch vụ** | Xem, sửa, xoá thông tin | Bác sĩ |
| **3** | **Xem**  **Hồ sơ bệnh án** | Xem, export thông tin | Bác sĩ / Bệnh nhân |
| **4** | **Quản lý Lịch làm việc cá nhân** | Có thể thay đổi lịch làm việc của mình để bệnh nhân có thể xem và đặt lịch hẹn khám | Bác sĩ |
| **5** | **Quản lý thông tin bệnh nhân** | Xem danh sách bệnh nhân, export thông tin bệnh nhân | Bác sĩ |

|  |  |  |  |
| --- | --- | --- | --- |
| **6** | **Quản lý Lịch hẹn khám** | Xem danh sách lịch hẹn khám của bệnh nhân, có thể chỉnh sửa chấp nhận, hủy lịch hẹn khám,… | Bác sĩ |
| **7** | **Lập Phiếu khám bệnh** | Lập phiếu khám bệnh cho bệnh nhân lưu các thông tin như: Họ tên, Ngày sinh, Chẩn đoán,… và kê đơn thuốc | Bác sĩ |
| **8** | **Lập danh sách khám bệnh** | Lập danh sách các bệnh nhân khám bệnh trong ngày để dễ dàng tính toán để lập báo cáo, danh sách sẽ reset vào 0:00 hằng ngày | Bác sĩ |
| 1**9**0  **9** | **Lập báo cáo doanh thu** | Lập báo cáo doanh thu hàng tháng, tính tỉ lệ từng ngày so với các ngày khác trong tháng | Bác sĩ |
| **10** | **Tạo phiếu hẹn** | Tạo phiếu hẹn khám bệnh với bác sĩ  mình muốn hẹn | Bệnh nhân |
| **11** | **Hủy**  **phiếu hẹn** | Hủy lịch hẹn mà mình mình muốn | Bệnh nhân |
| **12** | **Chỉnh sửa thông tin cá nhân** | Có thể chỉnh sửa được các thông tin cá nhân của bản thân như: Họ tên, Ngày sinh, Giới tính,… | Bệnh nhân |

## Công nghệ sử dụng

Các công nghệ lựa chọn trong triển khai thiết kế phần mềm bao gồm: MongoDB, ExpressJS, các ngôn ngữ lập trình web như HTML, CSS,…

Lý do lựa chọn các công nghệ này để xây dựng phần mềm:

* **Tính linh hoạt và mở rộng**: Cả MongoDB và ExpressJS đều được biết đến với tính linh hoạt và khả năng mở rộng, giúp phát triển ứng dụng có thể mở rộng được dễ dàng theo thời gian và nhu cầu của người dùng.
* **Hiệu suất**: MongoDB và ExpressJS đều được tối ưu hóa để cung cấp hiệu suất cao, giúp ứng dụng hoạt động mượt mà và đáp ứng nhanh chóng với các yêu cầu từ phía người dùng.
* **Cộng đồng hỗ trợ lớn**: Cả hai công nghệ này đều có cộng đồng người dùng và hỗ trợ lớn, điều này có nghĩa là có nhiều tài liệu, ví dụ và giải pháp sẵn có để giúp trong quá trình phát triển và triển khai.
* **Tương thích và tích hợp tốt**: MongoDB và ExpressJS đều có khả năng tương thích và tích hợp tốt với các công nghệ và dịch vụ khác, giúp việc phát triển và mở rộng ứng dụng trở nên thuận lợi hơn.

### MongoDB

MongoDB là một trong những cơ sở dữ liệu NoSQL phổ biến nhất, là một cơ sở dữ liệu hướng tài liệu có mã nguồn mở. Thuộc loại cơ sở dữ liệu không quan hệ (NoSQL), MongoDB không dựa trên cấu trúc cơ sở dữ liệu quan hệ giống như bảng mà sẽ lưu trữ và truy xuất dữ liệu ở dạng document. Định dạng dữ liệu mới mà MongoDB sử dụng là BSON (khá gần giống với định dạng JSON).

MongoDB được phát hành vào năm 2009 được viết bằng ngôn ngữ C++, C++ là ngôn ngữ gần với ngôn ngữ máy nên dễ dàng hiểu rằng MongoDB có thể tính toán ở tốc độ cao hơn hẳn các hệ quản trị cơ sở dữ liệu khác. Đó cũng là lý do mà MongoDB luôn được các nhà phát triển đánh giá rất cao.

MongoDB hỗ trợ đa nền tảng theo hướng đối tượng dùng lưu trữ dữ liệu có cấu trúc phức tạp. Không như các hệ quản trị cơ sở dữ liệu khác lưu trữ ở dạng bảng, MongoDB lưu dữ liệu vào collection theo hướng tài liệu JSON. Đây là kiểu dữ liệu dạng Key-Value truy xuất nhanh và khả năng mở rộng không bị ràng buộc mới tạo khóa ngoại hay khóa chính.

### NodeJS

Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event- driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

Node.js là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng Node.js được viết bằng Javascript và có thể chạy trong môi trường Node.js trên hệ điều hành Window, Linux,...

Node.js cũng cung cấp cho chúng ta các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node.js với các phần mở rộng.

Node.js hoạt động với một luồng duy nhất và có khả năng asynchronous (bất đồng bộ). Không giống như server được viết bằng PHP thì mỗi ông request đến server thì server sẽ tạo ra một thread để xử lý trong khi đó server node xử lý mọi hành động trong một thread duy nhất. Với cách thiết kế như vậy Node.js sẽ hỗ trợ trang web một số điều như:

* Tối ưu hóa thời gian thực hiện tiến trình: Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.
* Không đồng bộ và phát sinh sự kiện: Tất cả các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking. Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dữ liệu. Server chuyển sang một API sau khi gọi nó và có cơ chế thông báo về sự kiện của Node.js giúp server nhận được phản hồi từ các API gọi trước đó.
* Có khả năng mở rộng trong các ứng dụng web với nhiều hoạt động I/O liên tục.
* Phù hợp để xây dựng các ứng dụng web stream hay các game chơi trên nền web đảm bảo việc độ trễ thời gian xử lý hành động là nhỏ nhất.
* Dễ dàng để xây dựng các ứng dụng real-time.
* Cách viết ứng dụng với Node đó là các ứng dụng được cấu tạo từ các module nhỏ sau đó được kết hợp lại với nhau điều này đảm bảo cho việc sửa đổi bảo trì một cách nhanh chóng.
* Hiệu năng cao: Node.js sử dụng một mô hình luồng đơn (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request. Node.js sử dụng các chương trình đơn luồngvà các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.

### ExpressJS

ExpressJS là một framework web được xây dựng trên nền tảng của Node.js, được sử dụng phổ biến trong việc phát triển các ứng dụng web và API. Với sự linh hoạt và tiện ích mạnh mẽ, ExpressJS đã trở thành một lựa chọn phổ biến cho các nhà phát triển web.

**Tính linh hoạt và dễ sử dụng:** ExpressJS cung cấp một cách tiếp cận linh hoạt cho việc xử lý yêu cầu và phản hồi HTTP. Bằng cách sử dụng các middleware, bạn có thể tạo ra các ứng dụng web phức tạp với một cách tiếp cận module hóa và tái sử dụng mã.

**Xử lý yêu cầu và phản hồi:** ExpressJS giúp dễ dàng xác định các định tuyến (routes) cho ứng dụng của bạn. Bạn có thể xác định các hàm xử lý cho mỗi tuyến đường, cho phép bạn điều hướng yêu cầu đến các trang hoặc dịch vụ khác nhau dựa trên các điều kiện như phương thức HTTP, URL, và tham số.

**Middleware mạnh mẽ:** Middleware cho phép bạn thực hiện các chức năng trung gian trước hoặc sau khi yêu cầu được xử lý. Điều này cho phép bạn thực hiện các nhiệm vụ như xác thực, xử lý lỗi, ghi nhật ký (logging), nén dữ liệu và nhiều hơn nữa một cách dễ dàng.

**Cộng đồng mạnh mẽ và hỗ trợ đa dạng:** ExpressJS có một cộng đồng lớn và sôi nổi của các nhà phát triển, cung cấp các module middleware và gói mở rộng (extensions) đa dạng. Điều này giúp bạn dễ dàng tích hợp các tính năng và chức năng mở rộng vào ứng dụng của mình.

### HTML

HTML (viết tắt của từ Hypertext Markup Language, hay là “Ngôn ngữ Đánh dấu Siêu văn bản”) là một [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) được thiết kế ra để tạo nên các [trang](https://vi.wikipedia.org/wiki/Website) [web](https://vi.wikipedia.org/wiki/Website) trên [World Wide Web](https://vi.wikipedia.org/wiki/World_Wide_Web). Cùng với [CSS](https://vi.wikipedia.org/wiki/CSS) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), HTML là một trong những ngôn ngữ quan trọng trong lĩnh vực thiết kế website. HTML được định nghĩa như là một ứng dụng đơn giản của [SGML](https://vi.wikipedia.org/wiki/SGML) và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn mực của [Internet](https://vi.wikipedia.org/wiki/Internet) do tổ chức [World Wide Web Consortium](https://vi.wikipedia.org/wiki/W3C) (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng [XHTML](https://vi.wikipedia.org/wiki/XHTML). Hiện nay, phiên bản mới nhất của ngôn ngữ này là [HTML5](https://vi.wikipedia.org/wiki/HTML5).

Lưu ý: HTML không phải là ngôn ngữ lập trình.

Bằng cách dùng [HTML động](https://vi.wikipedia.org/wiki/HTML_%C4%91%E1%BB%99ng) hoặc [Ajax,](https://vi.wikipedia.org/wiki/Ajax_(l%E1%BA%ADp_tr%C3%ACnh)) lập trình viên có thể được tạo ra và xử lý bởi số lượng lớn các công cụ, từ một chương trình soạn thảo văn bản đơn giản – có thể gõ vào ngay từ những dòng đầu tiên – cho đến những công cụ xuất bản [WYSIWYG](https://vi.wikipedia.org/wiki/WYSIWYG) phức tạp. Hypertext là cách mà các trang Web (được thiết kế bằng HTML) được kết nối với nhau. Và như thế, đường link có trên trang Web được gọi là Hypertext. Như tên gọi đã nói, HTML là ngôn ngữ đánh dấu bằng thẻ (Markup Language), nghĩa là bạn sử dụng HTML để đánh dấu một tài liệu text bằng các thẻ (tag) để nói cho trình duyệt Web cách để cấu trúc nó để hiển thị

### CSS

CSS là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là [HTML](https://www.hostinger.vn/huong-dan/html-la-gi/). Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

CSS được phát triển bởi W3C ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

Những tag như <font> được ra mắt trong HTML phiên bản 3.2, nó gây rất nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ không chỉ là một trang chứa văn bản mà không có gì khác.

### Bootstrap

Bootstrap là một front-end framework miễn phí giúp quá trình phát triển web được nhanh và dễ dàng hơn. Bootstrap bao gồm các mẫu thiết kế dựa trên HTML và CSS như typography, forms, buttons, tables, navigation, modals, image carousels… cũng như các plugin JavaScript tùy chọn. Bootstrap cũng cung cấp khả năng tạo ra các responsive designs một cách dễ dàng.

### GIT

Git (/ɡɪt/) là [phần mềm quản lý mã nguồn](https://vi.wikipedia.org/w/index.php?title=Ph%E1%BA%A7n_m%E1%BB%81m_qu%E1%BA%A3n_l%C3%BD_m%C3%A3_ngu%E1%BB%93n&action=edit&redlink=1) [phân tán](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_t%C3%A1n&action=edit&redlink=1) được phát triển bởi [Linus](https://vi.wikipedia.org/wiki/Linus_Torvalds) [Torvalds](https://vi.wikipedia.org/wiki/Linus_Torvalds) vào năm [2005,](https://vi.wikipedia.org/wiki/2005) ban đầu dành cho việc phát triển [nhân Linux](https://vi.wikipedia.org/wiki/H%E1%BA%A1t_nh%C3%A2n_Linux). Hiện nay, Git trở thành một trong các phần mềm quản lý mã nguồn phổ biến nhất. Git là phần mềm mã nguồn mở được phân phối theo giấy phép công cộng [GPL2](https://vi.wikipedia.org/wiki/Gi%E1%BA%A5y_ph%C3%A9p_C%C3%B4ng_c%E1%BB%99ng_GNU)

Git có khả năng chạy trên nhiều hệ điều hành khác nhau như [Linux](https://vi.wikipedia.org/wiki/Linux),[Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Mac OSX](https://vi.wikipedia.org/wiki/OS_X) v.v..

**Khái niệm cơ bản trong GIT:**

* **Repository:**
* Repository (nhà kho) hay được gọi tắt là Repo đơn giản là nơi chứa/cơ sở dữ liệu (database) tất cả những thông tin cần thiết để duy trì và quản lý các sửa đổi và lịch sử của hợp đồng.
* *Trong Repo có 2 cấu trúc dữ liệu chính là Object Store và Index.* Tất cả dữ liệu của Repo đều được chứa trong thư mục bạn đang làm việc dưới dạng folder ẩn có tên là.git (không có phần tên trước dấu chấm).
* **Object store** là trái tim của Git, nó chứa dữ liệu nguyên gốc (original data files), các file log ghi chép quá trình sửa đổi, tên người tạo file, ngày tháng và các thông tin khác. Git có bốn loại object là: Blobs, Trees, Commits, Tags
* **Blobs:** là file nhị phân có thể chứa được mọi loại dữ liệu bất kể là dữ liệu của chương trình gì.
* **Trees:** lớp đại diện cho thông tin thư mục như thông tin định danh của blob, đường dẫn, chứa một ít metadata chứa thông tin cấu trúc và các thư mục nhỏ có trong thư mục đó.
* **Commits:** Chứa metadata có thông tin về mọi thứ như tên tác giả, người tải lên (commiter), ngày tải lên, thông tin log...
* **Tags:** đánh dấu cho dễ đọc. Thay vì một cái tên dài như là 9da581d910c9c4ac93557ca4859e767f5caf5169, chúng ta có thể tên tag là Ver-1.0- Alpha. Dễ nhớ và dễ sử dụng hơn.
* **Index:**
* Index là file nhị phân động và tạm thời miêu tả cấu trúc thư mục của toàn bộ Repo và trạng thái của hợp đồng được thể hiện thông qua commit và tree tại một thời điểm nào đó trong lịch sử phát triển. Git là một hệ thống truy tìm nội dung (content tracking system).
* Index không chứa nội dung file mà chỉ dùng để truy tìm (track) những thứ mà bạn muốn commit.

### Visual Studio Code

Visual Studio Code là một IDE mã nguồn mở, nhẹ nhàng và mạnh mẽ được phát triển bởi Microsoft. Nó được thiết kế để hỗ trợ nhiều ngôn ngữ lập trình và công nghệ khác nhau, từ JavaScript và Python đến C# và Java.

**Tính nhẹ nhàng và nhanh chóng:** VS Code được thiết kế để có hiệu suất cao và tương thích trên nhiều hệ điều hành khác nhau. Với giao diện sáng sủa và thân thiện, nó mang lại trải nghiệm làm việc dễ dàng và thoải mái cho người dùng.

**Hỗ trợ mạnh mẽ cho lập trình web:** VS Code cung cấp các công cụ mạnh mẽ để phát triển ứng dụng web. Tích hợp sẵn hỗ trợ cho HTML, CSS, và JavaScript, cùng với các extension phổ biến như Live Server và Emmet, giúp tăng tốc độ phát triển và nâng cao hiệu suất.

**Hỗ trợ đa ngôn ngữ và đa nền tảng:** VS Code hỗ trợ nhiều ngôn ngữ lập trình và khung làm việc, từ Node.js và React đến Python và Django. Điều này giúp cho việc phát triển ứng dụng đa năng và đa nền tảng trở nên dễ dàng hơn bao giờ hết.

**Cộng đồng lớn và mở rộng:** VS Code có một cộng đồng lớn và sôi động của các nhà phát triển, cung cấp hàng trăm extension và theme khác nhau để tùy chỉnh và mở rộng chức năng của IDE.

**Hỗ trợ tích hợp và công cụ phát triển:** VS Code tích hợp tốt với các công cụ phát triển khác như Git, Docker và Terminal, giúp tối ưu hóa quy trình làm việc và tăng cường hiệu suất phát triển.

**Khả năng mở rộng và tuỳ chỉnh:** VS Code cho phép bạn tùy chỉnh giao diện, cài đặt và extension để phù hợp với nhu cầu cụ thể của bạn trong quá trình phát triển.

# CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

## Khái quát bài toán

Hiện này xã hội ngày càng phát triển, trong khi môi trường sống ngày càng ô nhiễm, sức khỏe con người cũng đối mặt với nhiều nguy cơ bệnh tật hơn. Minh chứng là những dịch bệnh trong các năm gần đây như Covid-19, bệnh đậu mùa khỉ, … Vì nhu cầu khám chữa bệnh hiện nay rất lớn, nhưng số lượng bệnh viện lại không đủ đáp ứng nhu cầu đó, nên ngày càng nhiều phòng khám tư được mở ra. Trong lĩnh vực quản lý phòng khám, Công nghệ thông tin cũng đóng vai trò quan trọng trong việc tối ưu hóa quy trình làm việc và cải thiện hiệu suất. Quy trình tiếp đón bệnh nhân, đặc biệt là tại phòng khám, có thể được tối ưu hóa thông qua việc áp dụng các giải pháp tin học công nghệ.

### Mô tả sơ lược

Phần mềm quản lý phòng khám là một hệ thống phần mềm fđược thiết kế để giúp phòng khám quản lý thông tin và hoạt động của mình một cách hiệu quả. Ở đây người bệnh có thể vào web để xem thông tin các loại thuốc, xem được thông tin bác sĩ, lịch làm và có thể đặt lịch cho bác sĩ mà mình muốn hẹn lịch, có thể thay đổi thông tin. Bác sĩ có thể quản lý được thông tin của mình, quản lý được các loại thuốc có trong kho, quản lý các dịch vụ, quản lý hồ sơ bệnh án

Phần mềm quản lý phòng khám mang lại nhiều lợi ích cho phòng khám, bao gồm:

* Tăng cường quản lý thông tin: Phần mềm giúp phòng khám quản lý thông tin về bệnh nhân, bác sĩ, thuốc, dịch vụ,... một cách hiệu quả, giúp phòng khám tiết kiệm thời gian và công sức.
* Nâng cao chất lượng dịch vụ: Phần mềm giúp phòng khám nâng cao chất lượng dịch vụ khám chữa bệnh, giúp bệnh nhân hài lòng hơn với dịch vụ của phòng khám.
* Tăng cường hiệu quả hoạt động: Phần mềm giúp phòng khám tăng cường hiệu quả hoạt động, giúp phòng khám tiết kiệm chi phí và nâng cao lợi nhuận.
* Tóm lại, phần mềm quản lý phòng khám là một công cụ hữu ích giúp phòng khám quản lý thông tin và hoạt động của mình một cách hiệu quả, giúp phòng khám nâng cao chất lượng dịch vụ và tăng cường hiệu quả hoạt động.

### Nhu cầu quản lý

* Quản lý thông tin bệnh nhân: Phần mềm cho phép phòng khám lưu trữ và quản lý thông tin về bệnh nhân, bao gồm tên, tuổi, địa chỉ, số điện thoại, lịch sử khám bệnh,...
* Quản lý thông tin bác sĩ: Phần mềm cho phép phòng khám lưu trữ và quản lý thông tin về bác sĩ, bao gồm tên, chuyên khoa, lịch làm việc,...
* Quản lý thông tin thuốc và dịch vụ: Phần mềm cho phép phòng khám lưu trữ và quản lý thông tin về các loại thuốc và dịch vụ đang được cung cấp tại phòng khám.
* Quản lý lịch hẹn khám bệnh: Phần mềm cho phép bệnh nhân đặt lịch hẹn khám bệnh với bác sĩ, và phòng khám quản lý danh sách lịch hẹn khám bệnh.
* Quản lý hồ sơ bệnh án: Phần mềm cho phép phòng khám lưu trữ và quản lý hồ sơ bệnh án của bệnh nhân, bao gồm kết quả khám, kê đơn thuốc,...
* Quản lý tài chính: Phần mềm cho phép phòng khám quản lý thu chi, doanh thu, chi phí,...
* Quản lý báo cáo: Phần mềm cho phép phòng khám tạo và xem các báo cáo về hoạt động của phòng khám.

### Phạm vi

Phòng khám tư nhân MyClinic

### Đối tượng

* ***Bác sĩ****:* có thể sử dụng phần mềm để quản lý hồ sơ bệnh án của bệnh nhân, kê đơn thuốc, và xem các thông tin khác về phòng khám
* ***Bệnh nhân****:* có quyền xem thông tin về phòng khám; tạo, hủy phiếu hẹn, chỉnh sửa thông tin cá nhân,…

## Xác định yêu cầu nghiệp vụ

* 1. **Quản lý Thông Tin Bệnh Nhân**
* Lưu trữ Thông Tin Bệnh Nhân: Cho phép nhập, sửa đổi và xóa thông tin bệnh nhân, bao gồm tên, tuổi, địa chỉ, số điện thoại, và lịch sử khám bệnh.
* Tìm Kiếm Nhanh: Cung cấp khả năng tìm kiếm nhanh chóng thông tin bệnh nhân theo các tiêu chí khác nhau.

* 1. **Quản lý Thông Tin Bác Sĩ**
* Lưu trữ Thông Tin Bác Sĩ: Cho phép nhập, sửa đổi và xóa thông tin bác sĩ, bao gồm tên, chuyên khoa, và lịch làm việc.
* Lập Lịch Làm Việc: Cho phép bác sĩ cập nhật và quản lý lịch làm việc của mình.
  1. **Quản lý Thông Tin Thuốc và Dịch Vụ**
* Lưu trữ Thông Tin Chi Tiết: Cho phép nhập, sửa đổi và xóa thông tin chi tiết về thuốc và dịch vụ, bao gồm tên, mô tả, và giá.
* Danh Mục Thuốc và Dịch Vụ: Tạo danh mục để phân loại thuốc và dịch vụ theo các nhóm khác nhau.
  1. **Quản lý Lịch Hẹn Khám Bệnh**
* Đặt Lịch Hẹn: Cho phép bệnh nhân đặt lịch hẹn trực tuyến với bác sĩ theo sự thoải mái và linh hoạt.
* Quản lý Lịch Hẹn: Hiển thị và quản lý danh sách lịch hẹn khám bệnh, cung cấp thông tin chi tiết và khả năng chỉnh sửa.
  1. **Quản lý Hồ Sơ Bệnh Án**
* Lưu Trữ Hồ Sơ Bệnh Án: Cho phép nhập, sửa đổi và xóa thông tin hồ sơ bệnh án, bao gồm kết quả khám, kê đơn thuốc, và các thông tin liên quan.
* Tổ chức theo Bệnh Nhân: Tổ chức hồ sơ bệnh án theo từng bệnh nhân để dễ dàng tra cứu và theo dõi.
  1. **Quản lý Tài Chính**
* Theo dõi Thu Chi: Ghi chép và theo dõi các giao dịch tài chính, bao gồm thu, chi, doanh thu, và chi phí.
* Báo Cáo Tài Chính: Cung cấp báo cáo chi tiết về tình hình tài chính của phòng khám.
  1. **Quản lý Báo Cáo**
* Tạo và Xem Báo Cáo: Cho phép tạo và xem các báo cáo về hoạt động của phòng khám, bao gồm doanh thu, số lượng bệnh nhân, và các thống kê khác.
* Tùy Chọn Xuất Báo Cáo: Hỗ trợ xuất báo cáo dưới dạng file Excel, PDF để dễ dàng chia sẻ và lưu trữ.

## Phân tích yêu cầu hệ thống

* 1. **Yêu cầu chức năng**
     1. ***Bệnh nhân có thể đặt được lịch hẹn và có thể huỷ khi cần thiết***
     2. ***Bệnh nhân có thể tìm kiếm các thông tin liên quan đến phòng khám***
     3. ***Bác sĩ có thể quản lý được danh sách lịch hẹn khám***
     4. ***Bác sĩ có thể thay đổi lịch làm việc hàng tuần***
     5. ***Bác sĩ có thể tạo hồ sơ bệnh án và lưu trữ vào CSDL***
     6. ***Bác sĩ có thể quản lý được thuốc, dịch vụ***

### Yêu cầu phi chức năng

Giao diện thân thiện, dễ sử dụng với người dùng

Ổn định, xử lí hiệu quả

Khả năng lưu trữ dữ liệu và phục hồi khi gặp sự cố

Yêu cầu phân quyền người dùng

## Phân tích thiết kế

* 1. **Biểu đồ Usecase**
     1. **Usecase tổng quát**

A diagram of a person's relationship

Description automatically generated

*Hình 1. Biểu đồ UseCase Tổng quát*

* + 1. **Usecase quản lý tài khoản cá nhân**

A blue rectangular object with text

Description automatically generated

*Hình 2. Biểu đồ UseCase Quản lý tài khoản cá nhân*

|  |  |  |
| --- | --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Xem lịch sử đặt hẹn | Hủy lịch hẹn |
| UseCase ID | UC02.UC01 | UC02.UC02 |
| UseCase Name | Xem lịch sử đặt hẹn | Hủy lịch hẹn |
| Description | Xem lại danh sách các lịch hẹn mình đã đặt trên website | Hủy lịch hẹn mà mình muốn hủy |
| Actor | Bệnh nhân | Bệnh nhân |
| Trigger | Bệnh nhân muốn xem danh sách các lịch hẹn đã đặt | Bệnh nhân muốn hủy lịch hẹn đã đặt |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nốI Internet khi thực hiện thao tác | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nốI Internet khi thực hiện thao tác |
| Post-Condition | Xem được danh sách lịch sử đặt hẹn đã từng đặt | Hủy được các lịch hẹn chưa được phản hồi từ bác sĩ |
| Basic Flow | 1. Người dùng truy cập vào màn hình thông tin cá nhân   Hệ thống hiện thị danh sách lịch sử đặt hẹn của mình | 1. Người dùng truy cập vào màn hình thông tin cá nhân 2. Hệ thống hiện thị danh sách lịch sử đặt hẹn của mình 3. Người dùng có thể chọn lịch hẹn mình muốn hủy bằng cách bấm nút Hủy   Người dùng xác nhận bằng cách bấm nút Có ở Modal hiển thị lên |
| Alternative Flow | Không có | Người dùng chọn Không để huỷ bỏ việc xóa |

|  |  |  |
| --- | --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Xem lịch sử khám bệnh | Thay đổi thông tin tài khoản |
| UseCase ID | UC02.UC03 | UC02.UC04 |
| UseCase Name | Xem lịch sử khám bệnh | Thay đổi thông tin tài khoản |
| Description | Xem lại danh sách các lần khám bệnh khám ở phòng khám | Thay đổi thông tin tài khoản cá nhân của mình |
| Actor | Bệnh nhân | Bệnh nhân |
| Trigger | Bệnh nhân muốn xem danh sách các lần khám ở phòng khám | Bệnh nhân muốn thay đổi thông tin tài khoản |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nối Internet khi thực hiện thao tác | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nối Internet khi thực hiện thao tác |
| Post-Condition | Xem được danh sách lịch sử đặt hẹn đã từng đặt | Thay đổi được thông tin các nhân |
| Basic Flow | 1. Người dùng truy cập vào màn hình thông tin cá nhân   Hệ thống hiện thị danh sách lịch sử khám bệnh ở phòng khám | 1. Người dùng truy cập vào màn hình thông tin cá nhân  2. Hệ thống hiện thị các thông tin cá nhân  3. Người dùng có thể tùy chọn thay đổi thông tin cá nhân mình muốn  Người dùng xác nhận bằng cách bấm nút Lưu |
| Alternative Flow | Không có | Người dùng chọn Không để huỷ bỏ việc xóa |

|  |  |
| --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Đổi mật khẩu |
| UseCase ID | UC02.UC05 |
| UseCase Name | Đổi mật khẩu cho tài khoản |
| Description | Có thể đổi mật khẩu tài khoản |
| Actor | Bệnh nhân |
| Trigger | Bệnh nhân muốn đổi mật khẩu của tài khoản cá nhân mình |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nối Internet khi thực hiện thao tác |
| Post-Condition | Thay đổi được mật khẩu tài khoản cá nhân mình |
| Basic Flow | * + - 1. Người dùng truy cập vào màn hình thông tin cá nhân       2. Hệ thống hiển thị các ô nhập mật khẩu   Người dùng xác nhận bằng cách bấm nút Đổi mật khẩu |
| Alternative Flow | Không có |

* + 1. **Usecase Tra cứu**

A diagram of a person's relationship

Description automatically generated

*Hình 3: Biểu đồ UseCase Tra cứu*

|  |  |  |
| --- | --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Tra cứu thuốc | Tra cứu dịch vụ |
| UseCase ID | UC03.UC01 | UC03.UC02 |
| UseCase Name | Tra cứu thuốc | Tra cứu dịch vụ |
| Description | Tra cứu thông tin thuốc có trên hệ thống | Tra cứu thông tin dịch vụ có trên hệ thống |
| Actor | Bệnh nhân/Bác sĩ | Bệnh nhân/Bác sĩ |
| Trigger | Bệnh nhân hoặc bác sĩ cần tìm kiếm thông tin thuốc | Bệnh nhân hoặc bác sĩ cần tìm thông tin dịch vụ |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nốI Internet khi thực hiện thao tác | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nốI Internet khi thực hiện thao tác |
| Post-Condition | Xem danh sách các loại thuốc có trên hệ thống | Xem danh sách các loại dịch vụ có trên hệ thống |
| Basic Flow | 1. Người dùng truy cập vào màn hình Tìm kiếm/Thuốc   Hệ thống hiện thị danh sách các loại thuốc có trên hệ thống | 1. Người dùng truy cập vào màn hình Tìm kiếm/Dịch vụ   Hệ thống hiện thị danh sách các loại dịch vụ có trên hệ thống |
| Alternative Flow | Không có | Không có |

|  |  |  |
| --- | --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Tra cứu bác sĩ | Tra cứu hồ sơ bệnh án |
| UseCase ID | UC03.UC03 | UC03.UC04 |
| UseCase Name | Tra cứu bác sĩ | Tra cứ hồ sơ bệnh án |
| Description | Tra cứu thông tin bác sĩ có trong hệ thống | Tra cứu các hồ sơ bệnh án có trên hệ thống |
| Actor | Bệnh nhân/Bác sĩ | Bác sĩ |
| Trigger | Bệnh nhân hoặc bác sĩ cần tìm kiếm thông tin bác sĩ | Bác sĩ cần tìm thông tin hồ sơ bệnh án |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nốI Internet khi thực hiện thao tác | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nốI Internet khi thực hiện thao tác |
| Post-Condition | Xem danh sách các loại thuốc có trên hệ thống | Xem danh sách các loại dịch vụ có trên hệ thống |
| Basic Flow | 1. Bệnh nhân hoặc bác sĩ truy cập vào màn hình Tìm kiếm/Bác sĩ   Hệ thống hiện thị danh sách các loại bác sĩ có trên hệ thống | 1. Bác sĩ truy cập vào màn hình Tìm kiếm/Hồ sơ bệnh án   Hệ thống hiện thị danh sách các hồ sơ bệnh án có trên hệ thống |
| Alternative Flow | Không có | Không có |

|  |  |
| --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Tra cứu bệnh nhân |
| UseCase ID | UC03.UC05 |
| UseCase Name | Tra cứu bệnh nhân |
| Description | Tra cứu thông tin bệnh nhân có trong hệ thống |
| Actor | Bác sĩ |
| Trigger | Bác sĩ cần tìm kiếm thông tin bệnh nhân đăng ký trong hệ thống |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nối Internet khi thực hiện thao tác |
| Post-Condition | Xem danh sách các bệnh nhân đăng ký tài khoản có trên hệ thống |
| Basic Flow | 1. Bệnh nhân hoặc bác sĩ truy cập vào màn hình Tìm kiếm/Bệnh nhân   Hệ thống hiện thị danh sách các bệnh nhân đăng ký tài khoản có trên hệ thống |
| Alternative Flow | Không có |

* + 1. **Usecase Quản lý thuốc**

**A person standing next to a blue square

Description automatically generated**

*Hình 4: Biểu đồ UseCase Quản lý thuốc*

|  |  |  |
| --- | --- | --- |
| Trường hợp  sử dụng  Thuộc tính | Xem thông tin thuốc | Sửa thông tin thuốc |
| UseCase ID | UC04.UC01 | UC04.UC02 |
| UseCase Name | Xem thông tin chi tiết thuốc | Sửa thông tin thuốc |
| Description | Xem lại danh sách các lần khám bệnh khám ở phòng khám | Thay đổi thông tin tài khoản cá nhân của mình |
| Actor | Bệnh nhân | Bệnh nhân |
| Trigger | Bệnh nhân muốn xem danh sách các lần khám ở phòng khám | Bệnh nhân muốn thay đổi thông tin tài khoản |
| Pre-Condition | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nối Internet khi thực hiện thao tác | * Người dùng đã đăng nhập vào hệ thống   Thiết bị của người dùng được kết nối Internet khi thực hiện thao tác |
| Post-Condition | Xem được danh sách lịch sử đặt hẹn đã từng đặt | Thay đổi được thông tin các nhân |
| Basic Flow | 1. Người dùng truy cập vào màn hình thông tin cá nhân   Hệ thống hiện thị danh sách lịch sử khám bệnh ở phòng khám | 1. Người dùng truy cập vào màn hình thông tin cá nhân  2. Hệ thống hiện thị các thông tin cá nhân  3. Người dùng có thể tùy chọn thay đổi thông tin cá nhân mình muốn  Người dùng xác nhận bằng cách bấm nút Lưu |

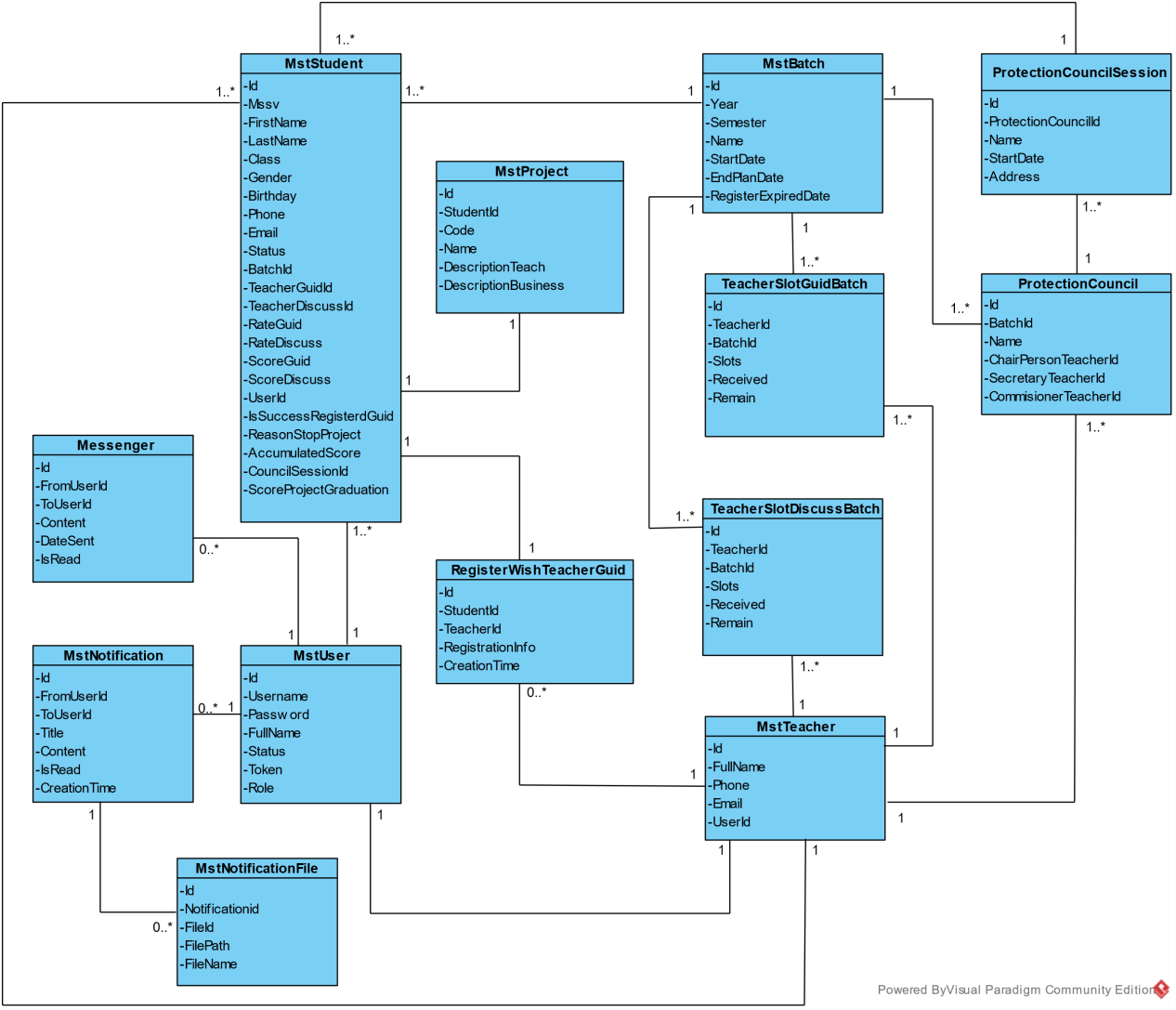
**4.1.5. Usecase Tra cứu**

* + 1. **Usecase Tra cứu**
    2. **Usecase Tra cứu**
    3. **Usecase Tra cứu**

### Biểu đồ Sequence

### Biểu đồ Lớp

##### Biểu đồ Lớp Thực thể



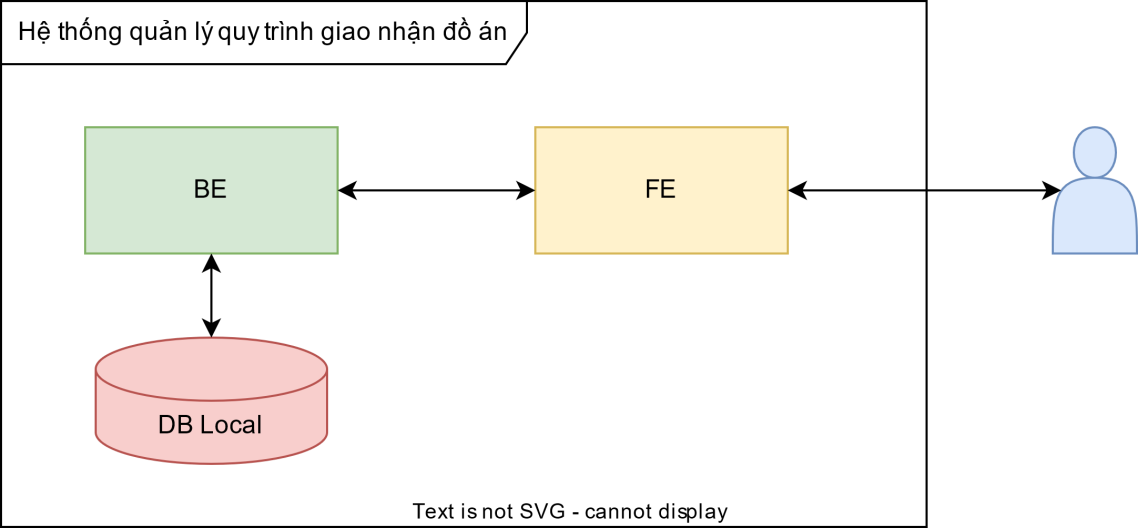
*Hình 43 Biểu đồ lớp Thực thể*

##### Biểu đồ lớp Giao diện

*Hình 44 Biểu đồ lớp Giao diện*

**CHƯƠNG 3: THIẾT KẾ HỆ THỐNG**

## Mô hình kiến trúc hệ thống



*Hình 45 Mô hình kiến trúc hệ thống*

Mô hình kiến trúc tổng thể hệ thống được mô tả như sau:

* Hệ thống được thiết kế để chạy trên hệ điều hành Windows, có khả năng mở rộng thêm nếu được deploy
* Hệ thống sử dụng Database riêng biệt, đọc dữ liệu từ file import thông tin Sinh viên đầu tiên và những thông tin có sẵn trong hệ thống (hiện tại sử dụng Db local)
* Người sử dụng ( Bộ môn, Giáo viên, Sinh viên ) truy cập và sử dụng hệ thống qua môi trường mạng LAN hoặc Internet (hiện tại hệ thống đang ở môi trường local)

## Thiết kế Cơ sở dữ liệu Danh sách các Bảng

## Thiết kế Giao diện hệ thống

* 1. **Giao diện dành cho Bác sĩ**
  2. **Giao diện dành cho Bệnh nhân**

**CHƯƠNG 4: XÂY DỰNG VÀ CÀI ĐẶT HỆ THỐNG**

#### Quy trình xây dựng hệ thống

**Bước 1:**

Tiếp nhận bài toán nghiệp vụ của hệ thống, phân rã chức năng, xác định các trường dữ liệu cần thiết trên từng chức năng

**Bước 2:**

Xây dựng giao diện cho từng chức năng. Chốt giao diện với khách hàng tới khi có thống nhất cuối cùng đáp ứng đầy đủ yêu cầu của khách hàng

**Bước 3:**

Phân tích Database của khách hàng, tìm cách kết nối và đọc dữ liệu sẵn có của khách hàng (chỉ đọc)

**Bước 4:**

Thiết kế Database phù hợp vừa giải quyết được yêu cầu bài toán nghiệp vụ, vừa đảm bảo được hiệu năng của hệ thống giảm tải cho quá trình xử lý của hệ thống

**Bước 5:**

Tạo dự án ExpressJS, cài đặt các dependency như Express, Mongoose (để kết nối với MongoDB), và các middleware cần thiết, tạo kết nối tới cơ sở dữ liệu MongoDB, thiết kế và triển khai các route và controller để xử lý logic nghiệp vụ, kiểm thử các API endpoints.

**Bước 6:**

Trên FE (Frontend) code giao diện đã được thống nhất và các hàm để hiển thị các giao diện

**Bước 7:**

Phía BE (Backend) tạo kết nối LinQ từ Database Internal và Database của khác hàng đến project

**Bước 8:**

Trên project BE code xử lý nghiệp vụ của bài toán cần thực hiện, trả ra các Controller cho phía FE sử dụng

**Bước 9:**

Trên project FE tạo Web Service đến các Controller của BE

**Bước 10:**

Trên project FE tạo các component, các hàm thực hiện việc nhận dữ liệu và call các service FE tương ứng controller được cung cấp từ phía BE

**Bước 11:**

Từ giao diện lấy các dữ liệu, sử dụng các hàm trong B10; nhận lại dữ liệu và check rồi hiển thị thông báo hoặc dữ liệu trên giao diện

#### Cài đặt Hệ thống

* 1. **Yêu cầu phần cứng**
     + Ổ cứng dung lượng tối thiểu 25GB
     + RAM tối thiểu 2GB

#### Yêu cầu phần mềm

* + - Chạy hệ điều hành Windows 7 trở lên
    - Hệ điều hành phải được cài đặt các gói cần thiết để chạy được ứng dụng C# như:
      * .Net Framework 5
      * Internet Information Service (IIS)
      * SQL Server 2014 trở lên

#### Các bước cài đặt chương trình

**Bước 1:** Thay đổi cấi hình kết nối Cơ sở dữ liệu trong file .env để kết nối vào MongoDB

**Bước 2:** Chạy câu lệnh Migrate Database để tạo các bảng

**Bước 3:** Insert dữ liệu mẫu vào trong Database

**Bước 4:** Chạy dự án Backend

**Bước 5:** Chạy dự án Frontend

**Bước 6:** Truy cập địa chỉ [http://localhost:](http://localhost/)[port]

# CHƯƠNG 5: TỔNG KẾT

## Kết quả đạt được

Hệ thống đã xây dựng xong bao gồm các chức năng nghiệp vụ đáp ứng được cơ bản các yêu cầu của quy trình giao nhận đồ án trong bộ môn Công nghệ Phần mềm Đại học Xây dựng Hà Nội

Trong quá trình nghiên cứu và thực hiện đồ án tốt nghiệp, sinh viên đã thu hoạch thêm các kiến thức sau:

* + Nắm được quy trình cơ bản của hoạt động Quản lý phòng khám
  + Tìm hiểu về ExpressJS để xây dựng giao diện
  + Phân tích thiết kế hệ thống
  + Sử dụng ExpressJS kết hợp các ngôn ngữ lập trình web như HTML, CSS, JS để phát triển hệ thống
  + Áp dụng Entity Framework để truy vấn đến đối tượng trong .NET

## Những hạn chế tồn tại

* Do thời gian trong khuôn khổ đồ án tốt nghiệp có hạn, nên việc phân tích yêu cầu nghiệp vụ có thể có hạn chế
* Về chức năng: một số chức năng như Nhắn tin, Thông báo còn hiển thị số lượng chưa đọc theo thời gian thực, chức năng Mượn – trả đồ án còn chưa có,…
* Về giao diện, thiết kế: UI – UX còn nhiều hạn chế

## Hướng phát triển hoàn thiện hệ thống

* Tìm hiểu thêm về nghiệp vụ, từ đó có thể phát triển ứng dụng đáp ứng được các yêu cầu của hệ thống
* Phát triển thêm các chức năng Nhắn tin và Thông báo, kiểm soát dữ liệu chặt chẽ hơn; Phát triển thêm tính năng Mượn trả đồ án
* Hoàn thiện, cải tiến UI – UX người dùng

# TÀI LIỆU THAM KHẢO

1. <https://angular.io/docs>
2. <https://adminlte.io/docs/3.2/>
3. <https://chat.openai.com/>
4. <https://echarts.apache.org/examples/en/>
5. <https://www.w3schools.com/>
6. https://code-maze.com/net-core-series/