

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

Tel. (+84.0236) 3736949, Fax. (84-511) 3842771
Website: <http://dut.udn.vn/khoacntt>, E-mail: cntt@dut.udn.vn



BÁO CÁO MÔN HỌC
DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

ĐỀ TÀI:
Chương trình ứng dụng quản lý đăng ký khám bệnh

| Họ tên sinh viên | Mã sinh viên | Nhóm HP |
|------------------|--------------|---------|
| Nguyễn Tấn Lộc | 102220155 | 22Nh11 |
| Nguyễn Quốc Lam | 102220154 | 22Nh11 |

CBHD : PGS.TS. NGUYỄN TẤN KHÔI

Đà Nẵng, 12/2024

MỤC LỤC

| | |
|---|-----------|
| LỜI NÓI ĐẦU | II |
| CHƯƠNG 1: CƠ SỞ LÝ THUYẾT | 2 |
| 1.1. KIẾN TRÚC HỆ THỐNG | 2 |
| 1.2. JAVASCRIPT | 2 |
| 1.3. NODEJS | 2 |
| 1.4. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG | 3 |
| 1.5. MANG MÁY TÍNH..... | 3 |
| 1.6. KẾT CHƯƠNG | 3 |
| CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG | 4 |
| 2.1. PHÁT BIỂU BÀI TOÁN | 4 |
| 2.2. PHÂN TÍCH HIỆN TRẠNG | 4 |
| 2.3. PHÂN TÍCH CHỨC NĂNG..... | 5 |
| 2.3.1. Đối tượng sử dụng | 5 |
| 2.3.2. Đăng nhập..... | 5 |
| 2.3.3. Quản lý người dùng, chuyên khoa, cơ sở y tế, phòng khám trong hệ thống | 5 |
| 2.3.4. Quản lý lịch khám bệnh, bệnh nhân | 5 |
| 2.3.5. Quản lý thời gian, lịch khám bệnh của bác sĩ | 5 |
| 2.3.6. Gửi email cho người dùng | 5 |
| 2.3.7. Đăng kí khám bệnh..... | 5 |
| 2.3.8. Công nghệ sử dụng | 6 |
| 2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU | 7 |
| 2.4.1. Sơ đồ lớp..... | 7 |
| 2.4.2. Sơ đồ cơ sở dữ liệu | 8 |
| 2.4.3. Các bảng | 9 |
| 2.5. XÂY DỰNG CHƯƠNG TRÌNH | 15 |
| 2.5.1. Tổ chức thư mục Backend..... | 15 |
| 2.5.2. Tổ chức thư mục Frontend | 16 |
| 2.6. KẾT CHƯƠNG | 17 |
| CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ | 18 |
| 3.1. MÔ HÌNH TRIỂN KHAI..... | 18 |
| 3.1.1. Môi trường triển khai..... | 18 |
| 3.1.2. Các công cụ sử dụng..... | 18 |
| 3.2. KẾT QUẢ THỰC NGHIỆM | 21 |
| 3.2.1. Kịch bản 1: Đăng nhập hệ thống | 21 |
| 3.2.2. Kịch bản 2: Trang chủ người dùng..... | 23 |
| 3.2.3. Kịch bản 3: Xem thông tin bác sĩ, phòng khám, chuyên khoa..... | 24 |
| 3.2.4. Kịch bản 4: User thực hiện các chức năng | 25 |
| 3.2.5. Kịch bản 5: Doctor thực hiện các chức năng..... | 28 |
| 3.2.6. Kịch bản 6: Admin thực hiện các chức năng | 33 |
| 3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ | 38 |
| 3.4. KẾT CHƯƠNG | 39 |

LỜI NÓI ĐẦU

Những năm gần đây, vai trò của hệ thống thông tin trong doanh nghiệp đang càng ngày lớn mạnh. Từ chỗ chỉ sử dụng để hỗ trợ một số hoạt động trong văn phòng, hệ thống thông tin đã trở nên có vai trò chiến lược trong doanh nghiệp. Đặc biệt những thành tựu trong công nghệ thông tin đã khiến doanh nghiệp ngày càng chú ý hơn tới việc áp dụng nó để gia tăng ưu thế cạnh tranh và tạo cơ hội cho mình. Hiện nay trào lưu ứng dụng thành tựu công nghệ thông tin không chỉ giới hạn trong chỉ doanh nghiệp lớn tầm cỡ đa quốc gia, mà còn lan rộng trong tất cả doanh nghiệp, kể cả những doanh nghiệp vừa và nhỏ ở những nước đang phát triển.

Tuy nhiên việc ứng dụng thành tựu công nghệ thông tin không phải đơn giản, ngay với cả những doanh nghiệp lớn, đòi hỏi nguồn tài lực, nhân lực và kinh nghiệm. Một ứng dụng thành công trong doanh nghiệp này cũng chưa chắc thành công trong doanh nghiệp khác. Trong thời đại công nghệ số hiện nay, việc tìm kiếm một nơi ở phù hợp đã trở nên dễ dàng và tiện lợi hơn bao giờ hết. Phần mềm của chúng em ra đời với mục tiêu cung cấp cho người dùng một công cụ hiệu quả, nhanh chóng và đáng tin cậy trong việc tìm kiếm và đăng ký khám bệnh. Với giao diện thân thiện, tính năng tìm kiếm và đăng ký thông minh chúng em hy vọng sẽ giúp người dùng tiết kiệm thời gian và công sức, đồng thời mang lại trải nghiệm tốt nhất. Vì thời gian thực hiện không nhiều, nên chắc chắn còn nhiều thiếu sót, mong được nhận được sự góp ý từ thầy/cô.

Chúng em xin trân trọng cảm ơn!

DANH SÁCH HÌNH ẢNH

- Hình 1. Kiến trúc hệ thống
- Hình 2. Sơ đồ lớp
- Hình 3. Sơ đồ cơ sở dữ liệu
- Hình 4. Cấu trúc thư mục code của Backend
- Hình 5. Cấu trúc thư mục code của Frontend
- Hình 6. Chạy chương trình BackEnd
- Hình 7. Test dữ liệu với Postman
- Hình 8. Quản lý cơ sở dữ liệu với XamPP
- Hình 9. Quản lý mã nguồn với GitHub
- Hình 10. Danh sách các API của phần mềm
- Hình 11. Giao diện đăng nhập vào hệ thống
- Hình 12. Giao diện Admin
- Hình 13. Giao diện Doctor
- Hình 14. Giao diện đăng nhập thất bại
- Hình 15. Giao diện trang chủ
- Hình 16. Giao diện bác sĩ
- Hình 17. Giao diện của chuyên khoa
- Hình 18. Giao diện của từng cơ sở y tế
- Hình 19. Người dùng chọn lịch để đăng ký khám
- Hình 20. Người dùng nhập đầy đủ thông tin khám bệnh
- Hình 21. Người dùng đặt lịch thành công
- Hình 22. Email từ chương trình
- Hình 23. Người dùng xác nhận khám bệnh
- Hình 24. Người dùng đặt lịch thất bại
- Hình 25. Giao diện quản lý lịch khám của bác sĩ
- Hình 26. Chọn lịch khám phù hợp
- Hình 27. Đăng ký lịch khám thành công
- Hình 28. Giao diện quản lý lịch khám bệnh
- Hình 29. Danh sách bệnh nhân đã đăng ký
- Hình 30. Xác nhận gửi thông báo hoàn thành khám
- Hình 31. Gửi thông báo xác nhận cho người dùng
- Hình 32. Email xác nhận hoàn thành khám bệnh
- Hình 33. Gửi thông báo huỷ lịch cho người dùng
- Hình 34. Email thông báo huỷ lịch khám
- Hình 35. Giao diện quản lý User
- Hình 36. Thêm người dùng thành công
- Hình 37. Xoá User thành công
- Hình 38. Cập nhật User thành công
- Hình 39. Giao diện quản lý phòng khám
- Hình 40. Thêm thành công phòng khám
- Hình 41. Giao diện quản lý chuyên khoa
- Hình 42. Thêm thành công chuyên khoa
- Hình 43. Giao diện quản lý lịch khám
- Hình 44. Chọn bác sĩ và chọn lịch
- Hình 45. Đăng kí lịch thành công

BẢNG PHÂN CÔNG CÔNG VIỆC

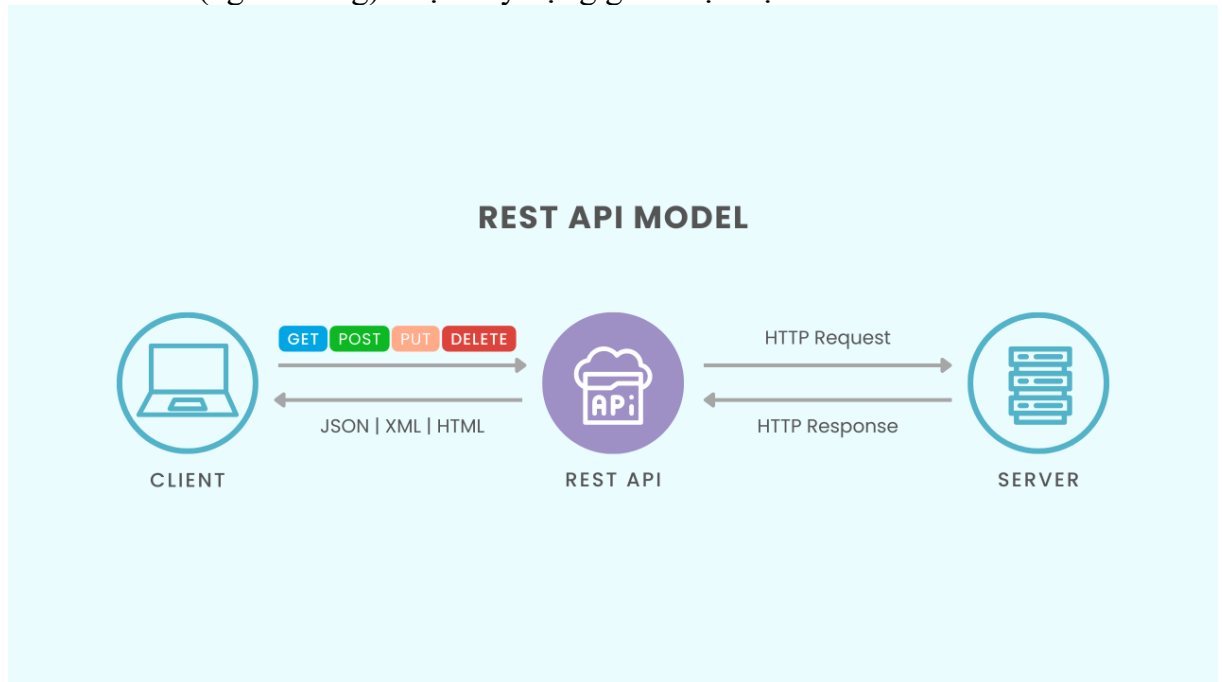
| STT | Công việc | Người thực hiện |
|-----|-----------------------------|-----------------|
| 1 | Chọn đề tài | Cả nhóm |
| 2 | Phân tích thiết kế hệ thống | Cả nhóm |
| 3 | Thiết kế cơ sở dữ liệu | Lộc |
| 4 | Thiết kế giao diện | Lam |
| 5 | Thiết kế máy chủ | Cả nhóm |
| 6 | Cài đặt | Cả nhóm |
| 7 | Kiểm tra chương trình | Cả nhóm |
| 8 | Viết báo cáo | Cả nhóm |

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. KIẾN TRÚC HỆ THỐNG

Chương trình được thiết kế theo mô hình MVC và RestfulAPI: Sự kết hợp giữa mô hình **MVC (Model-View-Controller)** và **RESTful API** tạo ra một kiến trúc mạnh mẽ để phát triển ứng dụng web hoặc hệ thống backend, giúp tối ưu hóa sự phân tách giữa logic nghiệp vụ, giao diện người dùng và giao tiếp qua mạng.

- Server (máy chủ) được xây dựng dựa trên NodeJS.
- Client (người dùng) được xây dựng giao diện dựa trên WPF.



Hình 1. Kiến trúc hệ thống

1.2. JAVASCRIPT

JavaScript được biết đến lần đầu tiên là Mocha và chỉ ngay sau đó một thời gian nó lại được đổi tên thành LiveScript và cuối cùng Netscape đã đổi tên thành JavaScript, bởi vì sự phổ biến của Java như là một hiện tượng lúc bấy giờ. Và sau khi ngôn ngữ mới được tạo ra, nhóm marketing của Netscape đã yêu cầu Sun cho phép họ đặt tên ngôn ngữ là JavaScript để truyền thông và quảng bá. Và đó cũng chính là lý do mà tại sao hầu hết người dùng chưa sử dụng JavaScript bao giờ, đã nghĩ rằng JavaScript có liên quan đến Java.

1.3. NODEJS

Node.js là một môi trường runtime mã nguồn mở được xây dựng trên nền tảng JavaScript, giúp thực thi JavaScript ngoài trình duyệt. Trước khi Node.js ra đời, JavaScript chủ yếu được sử dụng để phát triển phía client (frontend) trên các trình duyệt web. Node.js đã mở rộng khả năng này, cho phép sử dụng JavaScript để phát

triển các ứng dụng phía server (backend), từ đó tạo ra các ứng dụng full-stack bằng một ngôn ngữ duy nhất.

1.4. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Lập trình hướng đối tượng (Object-Oriented Programming - OOP) là một phương pháp lập trình dựa trên việc tổ chức mã nguồn thành các đối tượng (objects). Mỗi đối tượng là một thực thể trong chương trình, đại diện cho một phần cụ thể của vấn đề cần giải quyết.

1.5. MẠNG MÁY TÍNH

Trong mô hình MVC khi xây dựng một ứng dụng có API giao tiếp giữa client và server, **mạng máy tính** sẽ chủ yếu được xử lý ở phần **Controller** và một phần ở tầng **Service** (nếu bạn tách Service riêng ra). Dưới đây là cách mạng máy tính tích hợp vào các thành phần của MVC:

- Controller đóng vai trò làm cầu nối giữa client và backend. Khi client gửi request (qua HTTP hoặc WebSocket), Controller nhận request này, xử lý ban đầu, và gọi tới các tầng khác (như Service hoặc Model) để xử lý logic. Các hoạt động liên quan đến mạng tại Controller bao gồm: Xử lý request từ client, Xử lý response, Middleware (để kiểm tra xác thực hoặc các chính sách mạng như CORS.)
- Service là nơi xử lý logic nghiệp vụ và tương tác với các tầng khác. Nếu ứng dụng của bạn cần kết nối đến các API của bên thứ ba, phần này có thể bao gồm các logic gọi HTTP hoặc các giao thức mạng khác.
- View trong mô hình MVC truyền thống thường ở phía server và chịu trách nhiệm render giao diện. Tuy nhiên, nếu bạn đang xây dựng client tách biệt (Single Page Application hoặc ứng dụng React, Angular), thì phần View nằm ở phía client và giao tiếp với server qua API (HTTP request, WebSocket, ...).

1.6. KẾT CHƯƠng

Chương 1 – Cơ sở lý thuyết đã mô tả đầy đủ kiến trúc hệ thống, các ngôn ngữ lập trình, Framework đang được sử dụng trong hệ thống.

Trong chương này, chúng ta đã trình bày và phân tích các khái niệm cơ bản làm nền tảng cho toàn bộ nghiên cứu. Việc hiểu rõ cơ sở lý thuyết không chỉ giúp làm sáng tỏ vấn đề nghiên cứu mà còn định hướng cho các bước thực nghiệm và phát triển tiếp theo. Từ việc xây dựng các định nghĩa, mô hình lý thuyết đến phân tích các phương pháp và công cụ liên quan, chương này đã đặt nền móng vững chắc để giải quyết bài toán nghiên cứu một cách có hệ thống.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. PHÁT BIỂU BÀI TOÁN

Phần mềm quản lý đăng ký khám bệnh được phát triển với mục đích chính là mang đến cho người dùng một giải pháp hiệu quả và tiện lợi trong việc đăng ký khám bệnh tại các phòng khám hoặc trung tâm y tế. Với nhu cầu người dùng ngày càng cao, phần mềm sẽ giúp người dùng tìm được các dịch vụ, trung tâm y tế phù hợp trực tuyến thay vì phải đi khảo sát bên ngoài. Do đó, phần mềm này có thể giúp người dùng tiết kiệm được thời gian, công sức và có thể nhận được thông tin chính xác, tạo ra kết nối tốt từ người dùng tới các trung tâm y tế.

2.2. PHÂN TÍCH HIỆN TRẠNG

Hiện trạng: Hiện nay, nhu cầu y tế ngày càng gia tăng, đặc biệt là ở các thành phố lớn và các khu vực gần trường học, khu công nghiệp, và trung tâm thương mại. Tuy nhiên, quá trình này vẫn gặp nhiều khó khăn và thách thức:

- Thông tin phân tán và không đầy đủ: Người dân thường phải thu thập thông tin từ nhiều nguồn khác nhau như mạng xã hội, trang web, tờ rơi, và môi giới. Điều này không chỉ tốn thời gian mà còn dễ dẫn đến tình trạng thông tin không đầy đủ hoặc không chính xác.
- Khó khăn trong việc xác thực thông tin: Nhiều người dân gặp phải các tình huống thông tin về trung tâm y tế không đúng sự thật, dẫn đến mất thời gian và chi phí đi lại để kiểm tra. Bên cạnh đó, nhiều người dân gặp khó khăn trong việc di chuyển giữa các thành phố, tỉnh khác nhau rất khó nắm bắt được thông tin chính xác.
- Thiếu công cụ hỗ trợ tìm kiếm hiệu quả: Phần lớn các công cụ hiện có chưa đáp ứng được yêu cầu tìm kiếm linh hoạt, khó lọc theo các tiêu chí cụ thể như giá cả, vị trí, tiện ích kèm theo.

Yêu cầu:

- Tích hợp thông tin phong phú và đáng tin cậy: Phần mềm cần tổng hợp và cung cấp thông tin chi tiết từ nhiều nguồn uy tín, đảm bảo tính chính xác và cập nhật liên tục.
- Tính năng tìm kiếm và lọc thông minh: Cung cấp các công cụ tìm kiếm nâng cao cho phép người dùng dễ dàng lọc kết quả theo các tiêu chí như vị trí, giá thuê, tiện ích đi kèm, và các thông tin khác.
- Giao diện thân thiện và dễ sử dụng: Thiết kế giao diện đơn giản, trực quan giúp người dùng dễ dàng thao tác và tìm kiếm thông tin một cách nhanh chóng.
- Chức năng xác thực thông tin: Phần mềm nên có các biện pháp kiểm duyệt và xác thực thông tin từ các trung tâm y tế hay phòng khám, đảm bảo người dân nhận được thông tin chính xác và đáng tin cậy.
- Tính năng kết nối trực tiếp: Cho phép người thuê liên hệ trực tiếp với trung tâm y tế hoặc phòng khám qua ứng dụng, hỗ trợ trao đổi và thương lượng nhanh chóng.

2.3. PHÂN TÍCH CHỨC NĂNG

2.3.1. Đối tượng sử dụng

Hầu hết tất cả mọi người đều có thể sử dụng ứng dụng này một cách đơn giản mà không cần đăng nhập hay đăng kí. Khi đăng kí lịch khám bệnh, người dùng chỉ cần cung cấp đầy đủ thông tin mà hệ thống yêu cầu là được.

Trong đó, người dùng chính là những người muốn đăng kí lịch khám bệnh, nhân viên (bác sĩ) thuộc các cơ sở y tế, phòng khám có trong hệ thống.

2.3.2. Đăng nhập

Để quản lí các thông tin người dùng, chuyên khoa, cơ sở y tế,... Admin cần đăng nhập vào hệ thống để thực hiện các chức năng xem, thêm, sửa hay xoá khi cần thiết. Những bác sĩ thuộc các cơ sở y tế và phòng khám cũng sẽ được cấp đầy đủ các tài khoản để quản lí lịch khám, bệnh nhân của mình.

Đối với bệnh nhân, họ không cần phải đăng kí tài khoản mà chỉ cần cung cấp đầy đủ thông tin cần thiết, khi đó sẽ có email gửi về tài khoản email của bệnh nhân.

2.3.3. Quản lí người dùng, chuyên khoa, cơ sở y tế, phòng khám trong hệ thống

Trước tiên, quản trị viên hay bác sĩ cần đăng nhập vào hệ thống để thực hiện các chức năng xem, thêm, sửa hay xoá các thực thể này. Với giao diện người dùng dễ sử dụng thì rất dễ dàng để thực hiện các chức năng này trong hệ thống.

2.3.4. Quản lí lịch khám bệnh, bệnh nhân

Sau khi bệnh nhân đăng kí và xác nhận lịch khám bệnh xong, thông tin bệnh nhân sẽ được chuyển trực tiếp đến người bác sĩ mà bệnh nhân đã đăng kí kèm theo thời gian cụ thể. Sau khi khám xong, bác sĩ có thể xác nhận việc khám xong tại giao diện chương trình. Bên cạnh đó, bác sĩ cũng có thể huỷ lịch khám nếu có các vấn đề xảy ra và mọi thông tin sẽ được lưu vào cơ sở dữ liệu của hệ thống.

2.3.5. Quản lí thời gian, lịch khám bệnh của bác sĩ

Với người dùng là bác sĩ hoặc quản trị viên đều có thể chọn ngày, giờ phù hợp cho từng bác sĩ để đăng lên hệ thống cho bệnh nhân đăng kí khám bệnh.

2.3.6. Gửi email cho người dùng

Sau khi người dùng đăng kí thành công khám bệnh hoặc được bác sĩ xác nhận đã hoàn thành khám bệnh hoặc bác sĩ đã huỷ lịch khám bệnh, sẽ có một email được gửi về tài khoản email mà người dùng đã cung cấp cho hệ thống. Nếu email sai sót hoặc có các vấn đề xảy ra khi gửi email thì sẽ có thông báo lỗi cho người dùng.

2.3.7. Đăng kí khám bệnh

Đối với người dùng là bệnh nhân thì không cần đăng kí hay đăng nhập để sử dụng hệ thống. Bệnh nhân có thể sử dụng trực tiếp các chức năng tìm kiếm bác sĩ, cơ sở y tế, chuyên khoa hay phòng khám phù hợp với bản thân mình. Sau đó, người dùng chọn một lịch theo ngày, giờ sao cho phù hợp với bản thân mình. Tiếp theo đó là điền đầy đủ thông tin một cách chính xác và bệnh nhân sẽ nhận được một email xác nhận được gửi từ hệ thống.

2.3.8. Công nghệ sử dụng

Visual Studio Code: Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và MacOS. Nó hỗ trợ chức năng debug, đi kèm với git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt và các tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở, mặc dù gói tải xuống chính là có giấy phép.

Visual Studio: Visual Studio là một môi trường phát triển tích hợp (IDE) được phát triển bởi Microsoft, nổi tiếng trong giới lập trình viên và phát triển phần mềm.

Visual Studio cung cấp nhiều công cụ mạnh mẽ hỗ trợ phát triển các ứng dụng đa nền tảng, từ ứng dụng Windows, Web, đến ứng dụng di động. Visual Studio hỗ trợ các ngôn ngữ phổ biến như C#, C++, Python, JavaScript, TypeScript, F#, và nhiều ngôn ngữ khác, giúp lập trình viên dễ dàng phát triển các dự án đa dạng. Hỗ trợ thiết kế giao diện bằng cách kéo thả, giúp tăng tốc độ phát triển giao diện người dùng. Hỗ trợ Git và tích hợp với Azure DevOps.

WPF: WPF là một framework do Microsoft phát triển, dùng để xây dựng ứng dụng giao diện người dùng (UI) trên nền tảng Windows. WPF thuộc bộ .NET Framework và thường được sử dụng cùng với ngôn ngữ C#. Sử dụng XAML là ngôn ngữ đánh dấu dùng để định nghĩa giao diện trong WPF.

Framework:

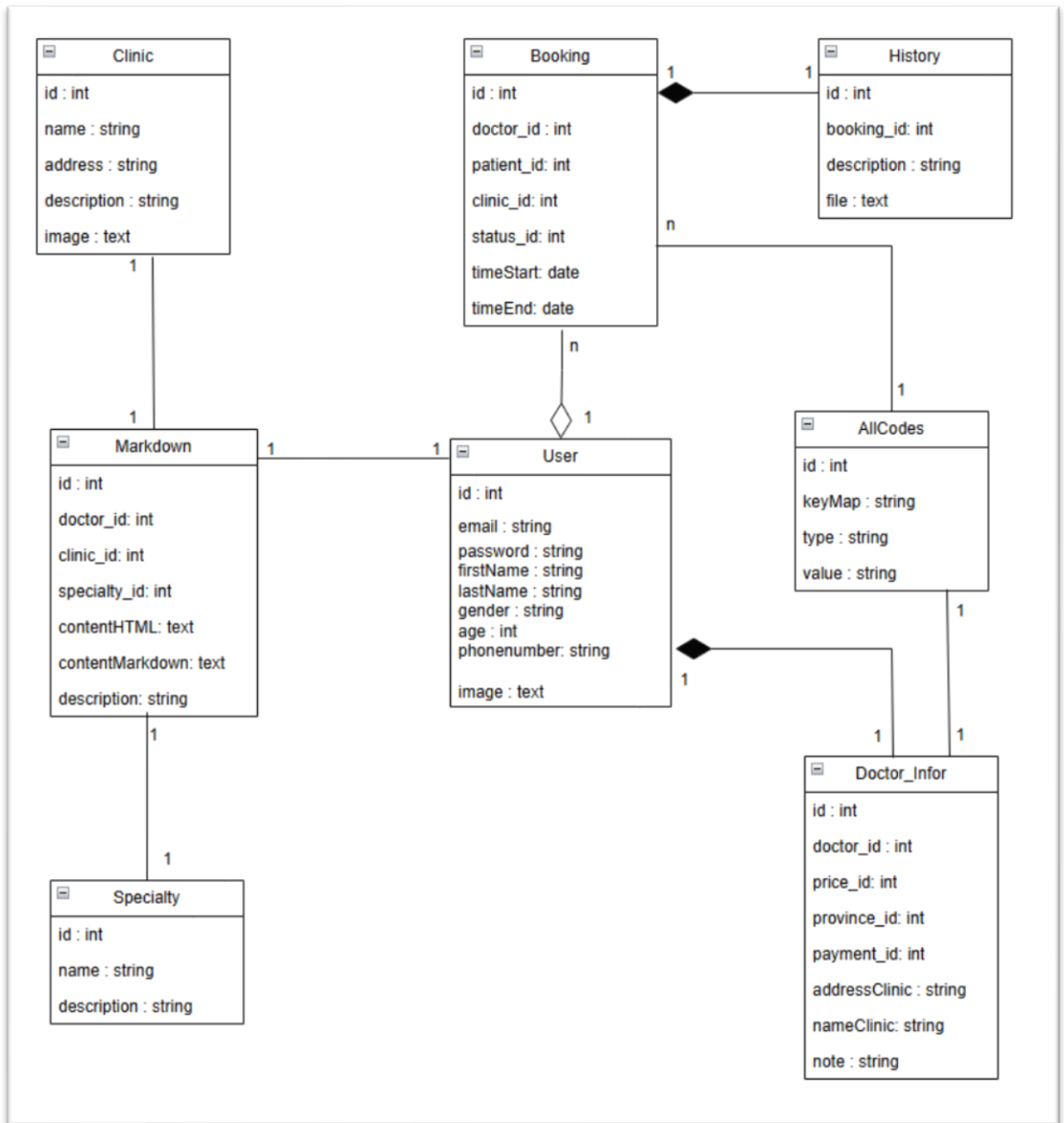
- Express.js: Express.js là một framework web đơn giản và linh hoạt dành cho Node.js, giúp việc xây dựng ứng dụng web và API dễ dàng hơn. Được xây dựng trên nền tảng Node.js, Express.js giúp tổ chức mã nguồn tốt hơn, giảm bớt những đoạn mã phức tạp trong quá trình phát triển backend và tăng hiệu suất làm việc.
- Sequelize: Sequelize là một ORM (Object-Relational Mapping) mạnh mẽ và phổ biến dành cho Node.js, giúp lập trình viên làm việc với cơ sở dữ liệu một cách dễ dàng và hiệu quả. Thay vì viết truy vấn SQL thủ công, Sequelize cho phép bạn tương tác với cơ sở dữ liệu bằng cách sử dụng các đối tượng và phương thức trong JavaScript.

Cơ sở dữ liệu: MySQL

Các công cụ hỗ trợ khác: Git, GitHub, Postman, Swagger.

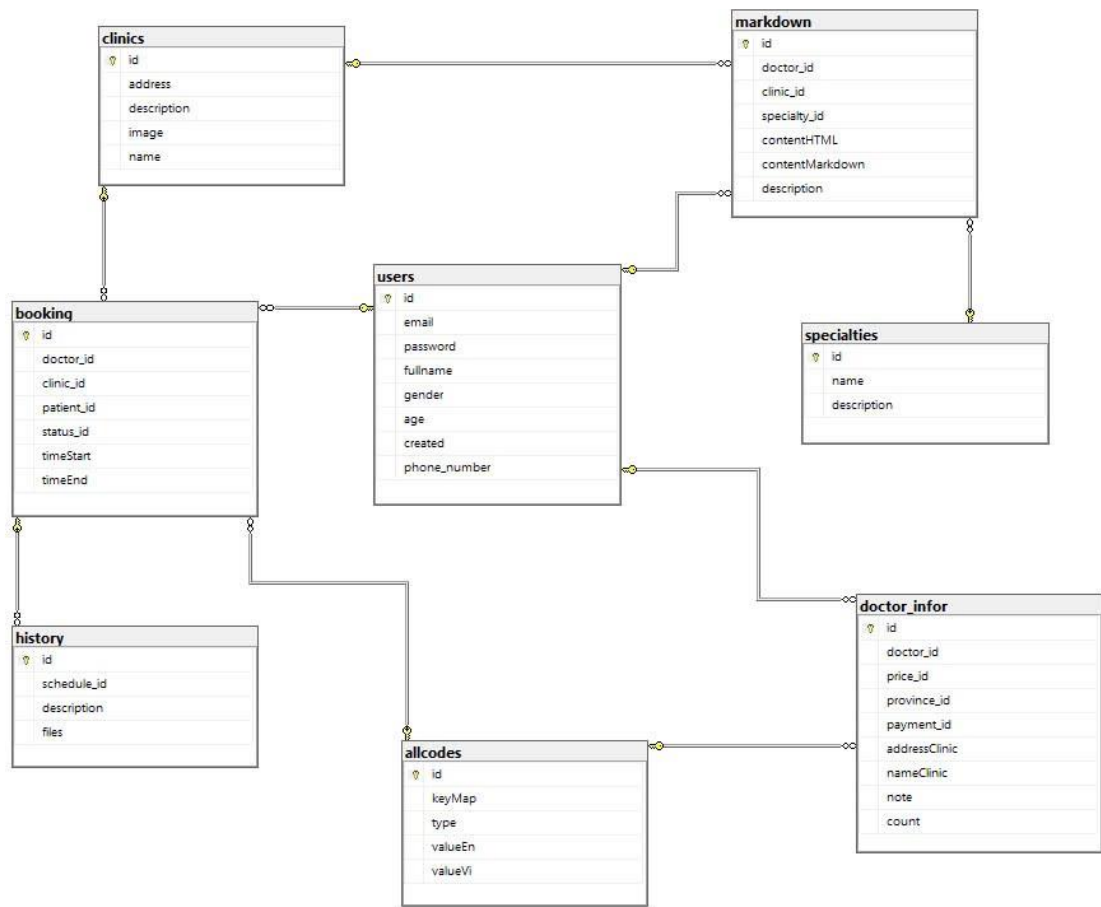
2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU

2.4.1. Sơ đồ lớp



Hình 2. Sơ đồ lớp

2.4.2. Sơ đồ cơ sở dữ liệu



Hình 3. Sơ đồ cơ sở dữ liệu

Users: lưu thông tin người dùng

Allcodes: lưu một số thông tin căn bản về giờ giấc, trạng thái,...

Clinics: lưu thông tin các phòng khám, cơ sở y tế

Specialties: lưu thông tin chuyên khoa y tế

Markdown: lưu các thông tin mô tả về phòng khám, cơ sở y tế, chuyên khoa...

Bookings: lưu các thông tin lịch khám khi người dùng đặt lịch

Histories: lưu lại lịch sử đặt lịch sau khi người dùng hoàn thành lịch khám

Doctor_Infor: lưu thông tin chi tiết của bác sĩ(gồm giá khám, cơ sở khám,...)

2.4.3. Các bảng

Bảng Users:

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|--------------|--------------|---|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Có giá trị tự động tăng. |
| email | string | Địa chỉ email của người dùng. Nó phải là duy nhất trong hệ thống và có thể được sử dụng để xác thực người dùng. |
| password | string | Mật khẩu của người dùng. Đây là thông tin bảo mật quan trọng và nên được mã hóa trước khi lưu trữ trong cơ sở dữ liệu. |
| fullname | string | Họ và tên đầy đủ của người dùng. Dùng để hiển thị tên người dùng trong hệ thống. |
| gender | string | Giới tính của người dùng. Có thể là "Nam", "Nữ", hoặc "Khác". |
| phone_number | string | Số điện thoại của người dùng. Có thể được sử dụng để xác thực và liên hệ với người dùng trong các tình huống cần thiết. |
| age | int | Tuổi của người dùng. Thường được tính toán từ ngày sinh để phân loại đối tượng người dùng. |
| avatar | blob | Được lưu dưới dạng file blob, để dàng xử lý khi đọc ghi dữ liệu. |

Bảng Doctor_infor:

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|------------|--------------|--|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Thường có giá trị tự động tăng. |
| doctor_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng User. Đây là mã số của bác sĩ liên kết với phòng khám. |

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

| | | |
|---------------|--------|--|
| price_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Allcodes. Đây là mã số của mức giá dịch vụ tại phòng khám. |
| province_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Allcodes. Đây là mã số tỉnh thành nơi phòng khám tọa lạc. |
| payment_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Allcodes. Đây là mã số phương thức thanh toán được chấp nhận tại phòng khám. |
| addressClinic | string | Địa chỉ của phòng khám. Cung cấp thông tin cụ thể về vị trí của phòng khám để người bệnh dễ dàng tìm thấy. |
| nameClinic | string | Tên của phòng khám. Đây là tên gọi chính thức của phòng khám dùng để nhận diện trong hệ thống. |
| note | string | Ghi chú về phòng khám. Đây là thông tin bổ sung về phòng khám, như giờ mở cửa, dịch vụ đặc biệt, v.v. |

Bảng Clinics:

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------------|---------------------|---|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Thường có giá trị tự động tăng và dùng để phân biệt các phòng khám trong hệ thống. |
| address | string | Địa chỉ của phòng khám. Đây là thông tin về vị trí cụ thể của phòng khám, giúp bệnh nhân hoặc người dùng dễ dàng tìm kiếm và di chuyển đến phòng khám.. |
| description | string | Mô tả về phòng khám. Đây là thông tin bổ sung về phòng khám, như các dịch vụ chính, chuyên môn, hoặc các đặc điểm nổi bật khác mà phòng khám cung cấp. |
| image | string | Đường dẫn đến hình ảnh của phòng khám. Thuộc tính này lưu trữ liên kết đến hình ảnh của phòng khám, giúp hiển thị hình |

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

| | | |
|--|--|--|
| | | ảnh trực quan về phòng khám trên giao diện người dùng. |
|--|--|--|

Bảng Specialties:

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------|--------------|---|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi chuyên khoa trong cơ sở dữ liệu. Thường có giá trị tự động tăng. |
| name | string | Tên của chuyên khoa. Đây là tên gọi chính thức của chuyên khoa, dùng để phân biệt các chuyên khoa khác nhau. |
| description | string | Mô tả chi tiết về chuyên khoa. Thông tin này giúp giải thích các dịch vụ, nhiệm vụ hoặc lĩnh vực chuyên môn mà chuyên khoa đó đảm nhận. |

Bảng Allcodes:

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|------------|--------------|--|
| id | string | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Thường là mã hoặc chuỗi ký tự đại diện cho một bản ghi cụ thể trong bảng. |
| keyMap | string | Là khóa (key) trong hệ thống mã hóa, dùng để phân loại các giá trị trong bảng. Thường dùng để nhóm các giá trị liên quan đến một loại dữ liệu hoặc chức năng cụ thể. |
| type | string | Loại của mã hoặc dữ liệu. Thuộc tính này cho biết loại thông tin mà bản ghi đại diện, ví dụ như "loại thanh toán", "tình trạng", "phương thức liên lạc", v.v. |
| valueEn | string | Giá trị của bản ghi trong ngôn ngữ tiếng Anh. Đây là giá trị mô tả hoặc thông tin liên quan đến bản ghi trong bảng Allcodes khi sử dụng ngôn ngữ tiếng Anh. |
| valueVi | string | Giá trị của bản ghi trong ngôn ngữ tiếng Việt. Đây là giá trị mô tả hoặc thông tin |

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

| | | |
|--|--|--|
| | | liên quan đến bản ghi trong bảng Allcodes khi sử dụng ngôn ngữ tiếng Việt. |
|--|--|--|

Bảng History

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------------|---------------------|--|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Thường có giá trị tự động tăng. |
| schedule_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Schedule. Đây là mã số của lịch hẹn hoặc kế hoạch liên quan đến lịch sử, giúp kết nối bản ghi này với một lịch trình cụ thể. |
| description | string | Mô tả chi tiết về lịch sử, có thể là thông tin về sự kiện, quá trình đã xảy ra hoặc ghi chú về một hành động cụ thể trong quá trình thực hiện kế hoạch. |
| files | string | Đường dẫn đến các tập tin đính kèm liên quan đến lịch sử. Đây có thể là các tài liệu, hình ảnh, hoặc tập tin khác có liên quan đến sự kiện lịch sử này. |

Bảng Booking

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------------|---------------------|--|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Thường có giá trị tự động tăng. |
| doctor_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng User. Đây là mã số của bác sĩ được chỉ định cho cuộc hẹn. |
| clinic_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Clinic. Đây là mã số phòng khám nơi cuộc hẹn sẽ diễn ra. |
| patient_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng User. Đây là mã số của bệnh nhân tham gia cuộc hẹn. |

| | | |
|-----------|----------|---|
| status_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Allcodes. Đây là trạng thái của cuộc hẹn, ví dụ: đã xác nhận, đang chờ, đã hủy, v.v. |
| timeStart | datetime | Thời gian bắt đầu của cuộc hẹn. Đây là thời điểm mà cuộc hẹn sẽ bắt đầu diễn ra. |
| timeEnd | datetime | Thời gian kết thúc của cuộc hẹn. Đây là thời điểm mà cuộc hẹn sẽ kết thúc, giúp xác định khoảng thời gian mà bác sĩ và bệnh nhân tham gia cuộc hẹn. |

Bảng Markdown

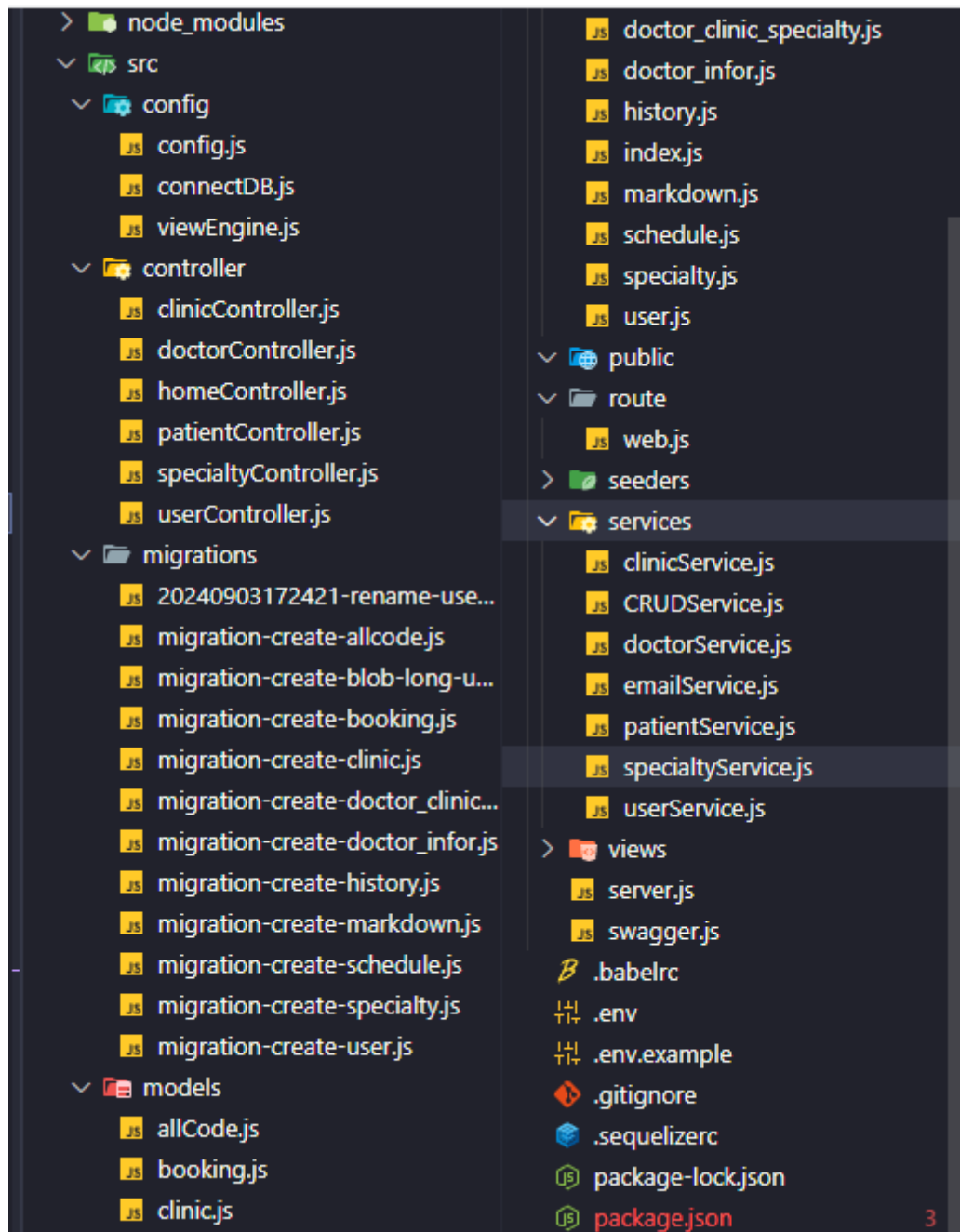
| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------------|---------------------|--|
| id | int | Là khóa chính (Primary Key) của bảng, dùng để xác định duy nhất mỗi bản ghi trong cơ sở dữ liệu. Thường có giá trị tự động tăng. |
| doctor_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng User. Đây là mã số của bác sĩ liên quan đến nội dung markdown, giúp xác định bác sĩ tạo ra hoặc chịu trách nhiệm về nội dung đó. |
| clinic_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Clinic. Đây là mã số của phòng khám nơi nội dung markdown này có liên quan, giúp xác định phòng khám liên quan. |
| specialty_id | int | Là khóa ngoại (Foreign Key) tham chiếu đến bảng Specialty. Đây là mã số chuyên khoa liên quan đến nội dung markdown, giúp phân loại nội dung theo chuyên môn. |
| contentHTML | longtext | Nội dung markdown được chuyển đổi sang định dạng HTML. Đây là bản hiển thị nội dung trong trình duyệt, có thể bao gồm các thẻ HTML để hiển thị văn bản, hình ảnh, liên kết, v.v. |

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

| | | |
|-----------------|----------|--|
| contentMarkdown | longtext | Nội dung gốc của bản ghi, được lưu trữ dưới định dạng Markdown. Đây là bản nội dung chưa được chuyển đổi sang HTML và có thể bao gồm các cú pháp Markdown như tiêu đề, danh sách, in đậm, in nghiêng, v.v. |
| description | string | Mô tả ngắn gọn về nội dung của markdown. Đây có thể là tóm tắt hoặc giới thiệu về nội dung mà bác sĩ hoặc phòng khám muốn chia sẻ |

2.5. XÂY DỰNG CHƯƠNG TRÌNH

2.5.1. Tổ chức thư mục Backend



Hình 4. Cấu trúc thư mục code của Backend

node_modules: là thư mục chứa các gói thư viện hỗ trợ chạy NodeJS.

config: là thư mục cấu hình

controllers: là nơi điều hướng xử lý sau khi API được gọi.

services: sau khi controllers được gọi tới thì dữ liệu sẽ được chuyển tiếp từ controllers đến services để có thể xử lý logic, tương tác với cơ sở dữ liệu và trả lại phản hồi cho người dùng.

models: là thư mục lưu trữ các model (đối tượng) được sử dụng trong phần mềm.

migrations: là nơi lưu trữ các file để có thể tạo ra các model ở trong hệ quản trị cơ sở dữ liệu.

route: là thư mục lưu trữ file chứa các API. Đây cũng chính là nơi chứa các URL backend kèm theo các giao thức mạng để giúp liên lạc giữa client và servers.

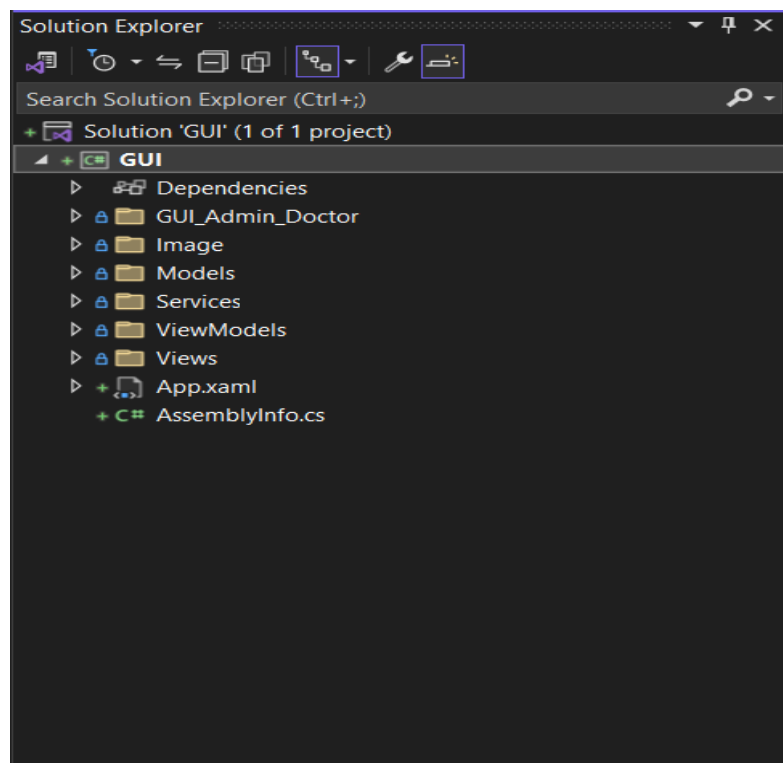
server.js: là file cấu hình được chạy đầu tiên khi chương trình backend(server) được chạy.

package.json: là file chứa các phiên bản của thư viện được cài đặt và sử dụng trong hệ thống.

.sequelizerc: là file được sử dụng để cấu hình các đường dẫn cho Sequelize CLI, giúp CLI xác định vị trí của các tệp cấu hình, models, migrations, và seeders trong một dự án Node.js sử dụng Sequelize.

.env: là file chứa các cấu hình trong môi trường bao gồm cổng chạy server (Port), tên cơ sở dữ liệu (db_name) hay cổng kết nối cơ sở dữ liệu(db_port),...

2.5.2. Tổ chức thư mục Frontend



Hình 5. Cấu trúc thư mục code của Frontend

Dependencies: Chứa các gói thư viện bên ngoài (các dependencies) được sử dụng trong dự án. Các thư viện này được quản lý thông qua công cụ như NuGet.

GUI_Admin_Doctor: Đây có thể là một thư mục chứa các logic, view và code liên quan đến tính năng quản trị cho admin và bác sĩ.

Image: Thư mục chứa các tài nguyên hình ảnh được sử dụng trong giao diện của ứng dụng.

Models: Chứa các lớp đại diện cho dữ liệu hoặc đối tượng trong ứng dụng. Nó tương tự như thư mục models trong Node.js, dùng để định nghĩa các đối tượng và các quy tắc liên quan.

Services: Nơi thực hiện các logic xử lý nghiệp vụ (business logic), như gọi API, xử lý dữ liệu, hoặc giao tiếp với các tầng khác.

ViewModels: Thư mục này có chức năng làm cầu nối giữa Views (giao diện người dùng) và Models (dữ liệu). Nó thường được sử dụng trong mô hình MVVM (Model-View-ViewModel) để cung cấp dữ liệu cho Views.

Views: Chứa các file giao diện người dùng (UI). Đây là nơi bạn định nghĩa cách ứng dụng hiển thị dữ liệu và tương tác với người dùng.

App.xaml: File này xác định các tài nguyên toàn cục (global resources) và các cài đặt khởi tạo cho ứng dụng WPF của bạn.

AssemblyInfo.cs: File chứa thông tin metadata của ứng dụng, chẳng hạn như phiên bản ứng dụng, bản quyền, và các thông tin khác liên quan đến dự án.

2.6. KẾT CHƯƠNG

Trong chương 2, các bước phân tích và thiết kế hệ thống đã được trình bày chi tiết nhằm xây dựng một nền tảng vững chắc cho quá trình phát triển phần mềm. Từ việc thu thập yêu cầu, phân tích nghiệp vụ, đến thiết kế tổng quan và chi tiết, toàn bộ quy trình đều hướng đến việc đảm bảo hệ thống đáp ứng đúng nhu cầu và hoạt động hiệu quả.

Các mô hình và sơ đồ được sử dụng, như sơ đồ lớp và sơ đồ cơ sở dữ liệu, không chỉ giúp minh họa cấu trúc và chức năng của hệ thống một cách trực quan, mà còn hỗ trợ phát hiện và giải quyết những vấn đề tiềm ẩn ngay từ giai đoạn thiết kế.

Với cơ sở phân tích và thiết kế được thực hiện kỹ lưỡng, hệ thống được định hình rõ ràng cả về kiến trúc lẫn chức năng. Điều này tạo tiền đề để chuyển sang giai đoạn hiện thực hóa hệ thống, đảm bảo tính khả thi và hiệu suất cao trong quá trình triển khai thực tế.

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. MÔ HÌNH TRIỂN KHAI

3.1.1. Môi trường triển khai

Chương trình ứng dụng được triển khai tại máy tính cá nhân của người dùng trên môi trường localhost tại cổng 8080 đối với Backend (Nodejs). Và Nodejs có phiên bản v14.17.0. Express.js là framework được sử dụng để tạo các API phục vụ cho việc xử lý yêu cầu từ phía người dùng và kết nối với cơ sở dữ liệu.

Frontend (WPF) được chạy tại Visual Studio và có thể kết nối với server thông qua các API do người lập trình viên định nghĩa. WPF được sử dụng nhằm mang lại một trải nghiệm người dùng mượt mà và tương tác tốt trên các hệ điều hành Windows.

Cơ sở dữ liệu: Sử dụng MySQL 8.0 được chạy tại cổng 3306 ở môi trường localhost. Các kết nối đến MySQL được thực hiện thông qua thư viện Sequelize, sử dụng để tương tác với cơ sở dữ liệu từ backend Node.js.

Hệ điều hành được sử dụng trong môi trường phát triển là Windows 10 hoặc 11, nhằm đảm bảo tính tương thích với cả frontend (WPF) và backend (Node.js).

Môi trường localhost sẽ được sử dụng cho các phiên bản thử nghiệm và kiểm tra, giúp phát triển nhanh chóng và hiệu quả. Các dữ liệu và yêu cầu từ người dùng được xử lý ngay trong máy tính cá nhân mà không cần phải triển khai trên máy chủ bên ngoài trong giai đoạn thử nghiệm.

3.1.2. Các công cụ sử dụng

Sử dụng Visual Studio Code để chạy dự án BackEnd và Visual Studio 2022 để chạy dự án FrontEnd và cả BackEnd và FrontEnd đều được chạy ở localhost.

```
PS D:\Booking Care\nodejs> npm start

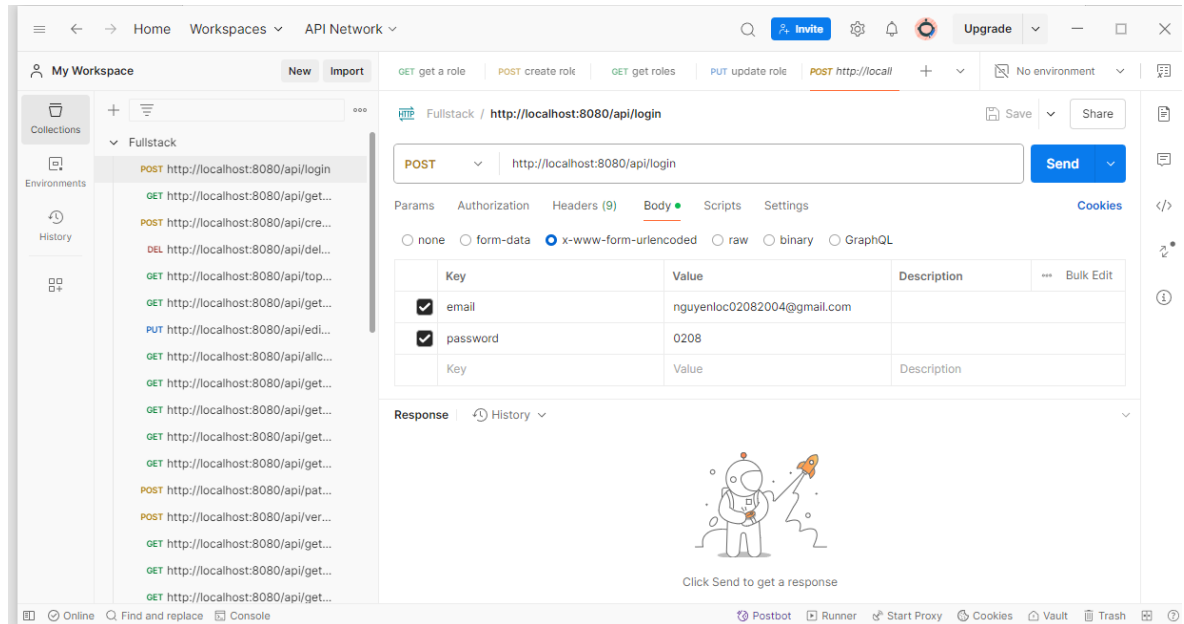
> booking-care@1.0.0 start
> nodemon --exec ./node_modules/.bin/babel-node src/server.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `./node_modules/.bin/babel-node src/server.js`
[nodemon] restarting due to changes...
[nodemon] starting `./node_modules/.bin/babel-node src/server.js`
backend node js is running on the port: 8080
Connection has been established successfully.
█
```

Hình 6. Chạy chương trình BackEnd

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

Sau khi kiểm tra kết nối đến cơ sở dữ liệu và server đã chạy ổn định, ta có thể kiểm tra (test) dữ liệu trực tiếp với PostMan.



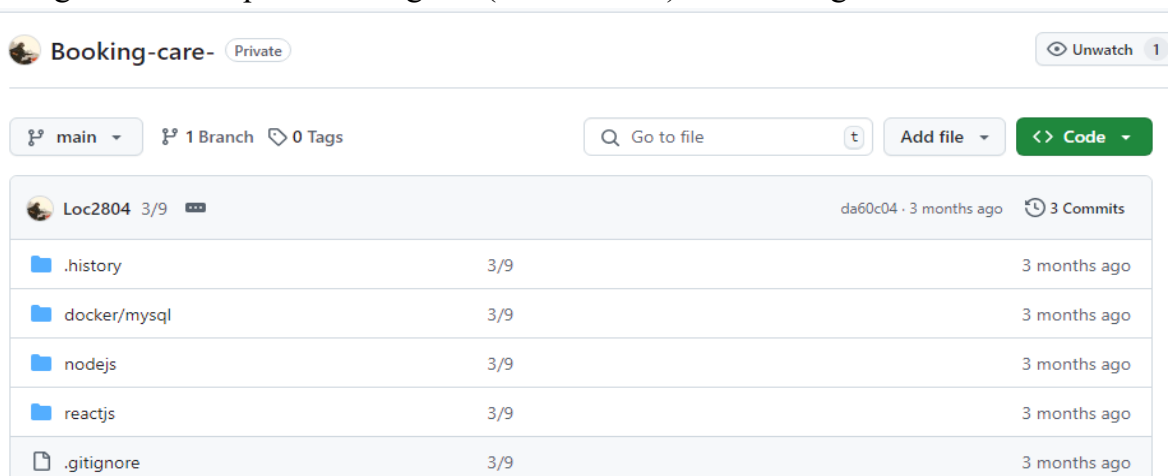
Hình 7. Test dữ liệu với Postman

Sử dụng Xampp với MySQL để quản lý cơ sở dữ liệu có trong hệ thống.

| Bảng | Hành động | Hàng | Kiểu | Bảng mã đối chiếu | Kích thước | Tổng chỉ phí |
|--|---|------|--------|--------------------|------------|--------------|
| <input type="checkbox"/> allcodes | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 43 | InnoDB | utf8mb4_general_ci | 16,0 KiB | - |
| <input type="checkbox"/> bookings | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 10 | InnoDB | utf8mb4_general_ci | 16,0 KiB | - |
| <input type="checkbox"/> clinics | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 7 | InnoDB | utf8mb4_general_ci | 1,5 MiB | - |
| <input type="checkbox"/> doctor_clinic_specialty | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 0 | InnoDB | utf8mb4_general_ci | 16,0 KiB | - |
| <input type="checkbox"/> doctor_infor | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 6 | InnoDB | utf8mb4_general_ci | 16,0 KiB | - |
| <input type="checkbox"/> histories | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 5 | InnoDB | utf8mb4_general_ci | 432,0 KiB | - |
| <input type="checkbox"/> markdowns | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 6 | InnoDB | utf8mb4_general_ci | 64,0 KiB | - |
| <input type="checkbox"/> schedules | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 3 | InnoDB | utf8mb4_general_ci | 16,0 KiB | - |
| <input type="checkbox"/> sequelizemeta | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 13 | InnoDB | utf8_unicode_ci | 32,0 KiB | - |
| <input type="checkbox"/> specialties | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 11 | InnoDB | utf8mb4_general_ci | 464,0 KiB | - |
| <input type="checkbox"/> users | ★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa | 10 | InnoDB | utf8mb4_general_ci | 2,0 MiB | - |
| 11 bảng | Tổng | 114 | InnoDB | utf8mb4_general_ci | 4,6 MiB | 0 B |

Hình 8. Quản lý cơ sở dữ liệu với Xampp

Dùng GitHub để quản lý mã nguồn (Source code) của chương trình.



Hình 9. Quản lý mã nguồn với GitHub

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

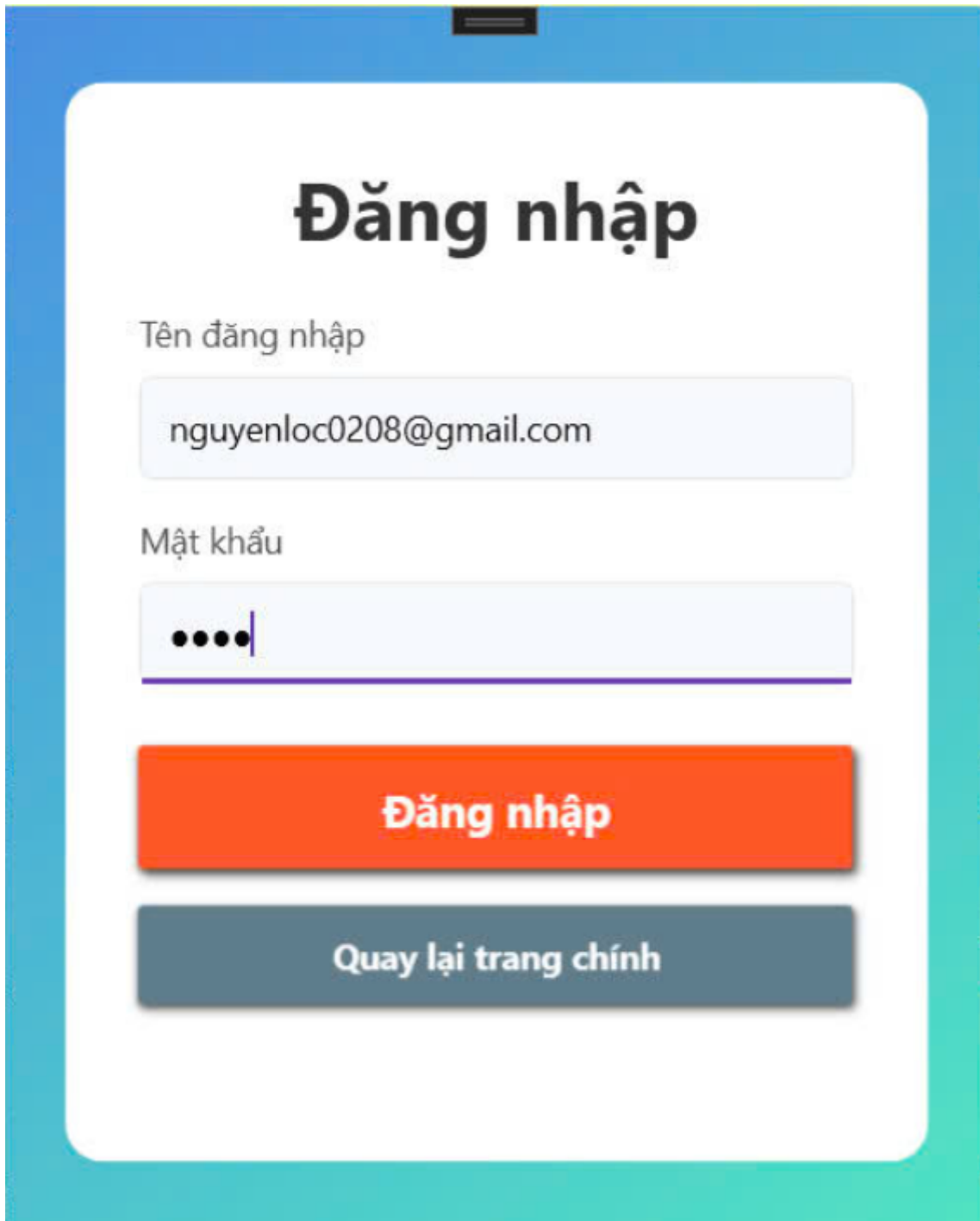
Bên cạnh đó, ta có thể sử dụng Swagger được cung cấp từ các thư viện để quản lý và test các API trực tiếp một cách dễ dàng.

| | | | |
|--------|----------------------------------|---|---|
| POST | /api/login | Users login | ▼ |
| GET | /api/get-all-users | Get all users | ▼ |
| POST | /api/create-new-user | Create new user | ▼ |
| PUT | /api/edit-user | Edit user | ▼ |
| DELETE | /api/delete-user | Delete user | ▼ |
| PUT | /api/change-password | Change password | ▼ |
| GET | /api/allcode | Get all code service | ▼ |
| GET | /api/top-doctor-home | Get top doctors for the homepage | ▼ |
| GET | /api/get-all-doctors | Get all doctors | ▼ |
| POST | /api/save-info-doctor | Save doctor information | ▼ |
| GET | /api/get-detail-doctor-by-id | Get details of a doctor by ID | ▼ |
| POST | /api/bulk-create-schedule | Bulk create doctor schedules | ▼ |
| GET | /api/get-schedule-doctor-by-date | Get doctor's schedule by date | ▼ |
| GET | /api/get-extra-info-doctor-by-id | Get extra information of a doctor by ID | ▼ |
| GET | /api/get-profile-doctor-by-id | Get profile information of a doctor by ID | ▼ |
| GET | /api/get-list-patient | Get list of patients | ▼ |
| POST | /api/send-remedy | Send remedy to a patient | ▼ |
| POST | /api/patient-book-appointment | Book an appointment for a patient | ▼ |
| POST | /api/verify-book-appointment | Verify the booking of an appointment | ▼ |
| POST | /api/create-new-specialty | Create a new specialty | ▼ |
| GET | /api/get-all-specialty | Get all specialties | ▼ |
| GET | /api/get-detail-specialty-by-id | Get details of a specialty by ID | ▼ |
| POST | /api/create-new-clinic | Create a new clinic | ▼ |
| GET | /api/get-all-clinic | Get all clinics | ▼ |
| GET | /api/get-detail-clinic-by-id | Get details of a clinic by ID | ▼ |

Hình 10. Danh sách các API của phần mềm

3.2. KẾT QUẢ THỰC NGHIỆM

3.2.1. Kịch bản 1: Đăng nhập hệ thống



Đăng nhập

Tên đăng nhập

nguyenloc0208@gmail.com

Mật khẩu

•••••

Đăng nhập

Quay lại trang chính

Hình 11. Giao diện đăng nhập vào hệ thống
Khi đăng nhập thành công: kiểm tra role(vai trò) người dùng và điều hướng sang các giao diện Admin và Doctor sao cho phù hợp với tài khoản.

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

- Đăng nhập thành công với Admin

The Admin Window interface features a top navigation bar with tabs: "Quản lí người dùng", "Quản lí chuyên khoa", "Quản lí phòng khám", and "Quản lí lịch khám". The current view is "Quản lí người dùng". The main area contains a form for adding or editing a user, with fields for Email, Mật khẩu, Họ và tên đệm, Tên, Số điện thoại, Địa chỉ, Giới tính, Ảnh đại diện, Vai trò, and Vị trí. There are buttons for "Lưu sự thay đổi", "Clear", and "Xoá". Below the form is a table of users.

| Id | Email | FirstName | LastName | PhoneNumber | Address | Gender |
|----|-------------------------------|-----------------|------------|-------------|------------------------|--------|
| 1 | nguyenloc02082004@gmail.com | Nguyễn Tấn | Lộc | 0914324759 | Lac Long Quan, Da Nang | Nam |
| 2 | kimvi@gmail.com | Nguyễn Thị | Kim Vi | 0343426416 | Cao Xuan Huy, Da Nang | Nữ |
| 15 | nguyenloc02082003@gmail.com | Nguyen | Loc | 01239124 | Da Nang | Nam |
| 16 | nguyenloc02082002@gmail.com | Hoc | Hoi | 014232414 | Viet Nam | Nam |
| 17 | huyh@gmail.com | Pham | Ngoc Huynh | 03124676 | Quang Nam | Nam |
| 18 | lam@gmail.com | Quoc | Lam | 0572367 | Dai Loc | Nam |
| 19 | holdanit@gmail.com | Nguyễn Tấn | Lộc | 03746863214 | Kon Tum | Khác |
| 20 | doctor6@gmail.com | John | Smith | 02783567 | UK | Nữ |
| 24 | nguyenquoclam210801@gmail.com | Nguyễn Quốc Lam | | 0174687146 | Quảng Nam | Nữ |

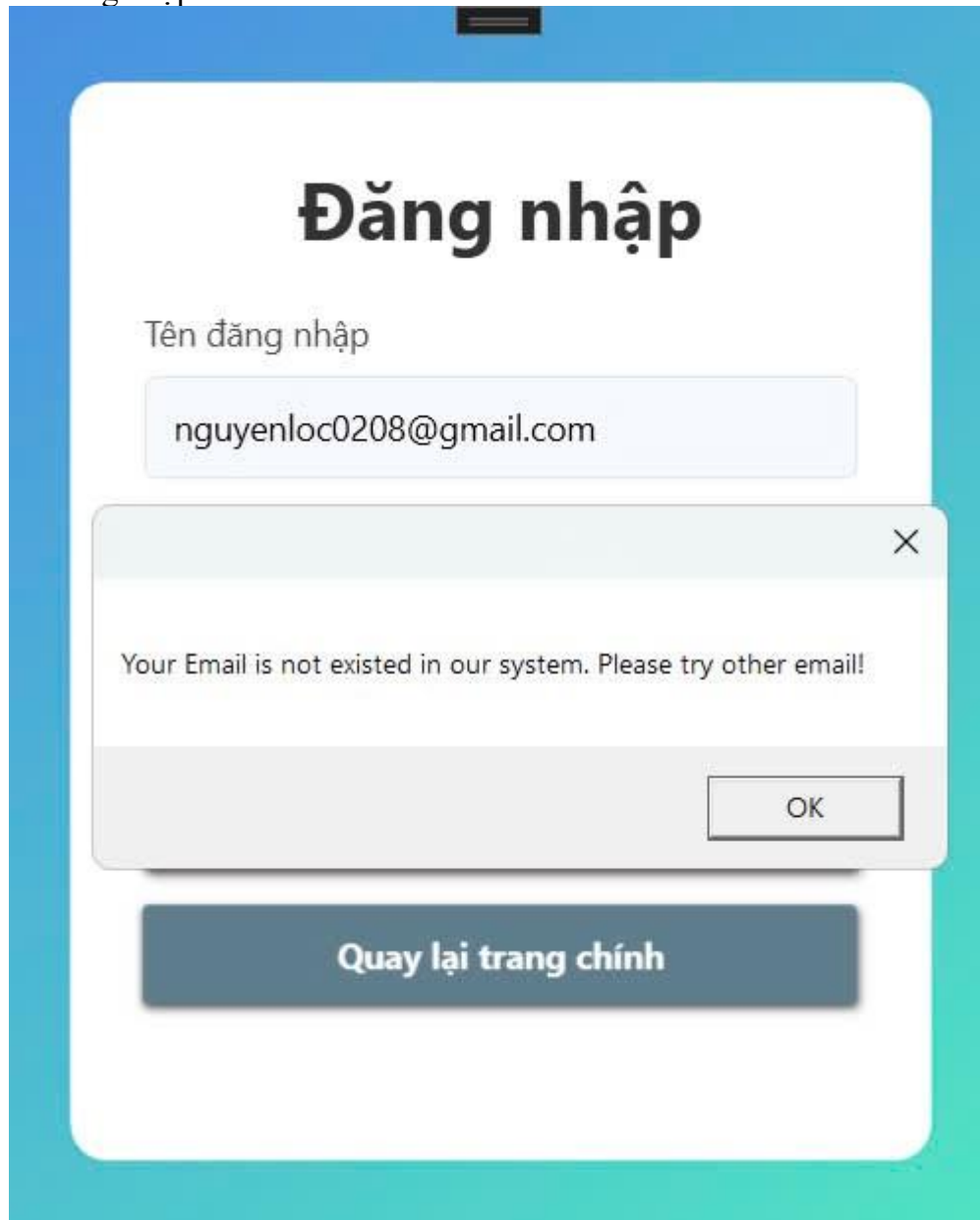
Hình 12. Giao diện Admin

- Khi đăng nhập thành công với Doctor

The Doctor Window interface has a top navigation bar with tabs: "Quản lí bệnh nhân" and "Quản lí lịch khám". The current view is "Quản lí bệnh nhân". The main area is titled "Quản lí bệnh nhân khám bệnh". It includes a "Chọn ngày khám" section with a calendar icon, and two buttons: "Xác nhận" and "Hủy bỏ".

Hình 13. Giao diện Doctor

Khi đăng nhập thất bại: thông báo lỗi cho người dùng và cho người dùng nhập lại thông tin đăng nhập.



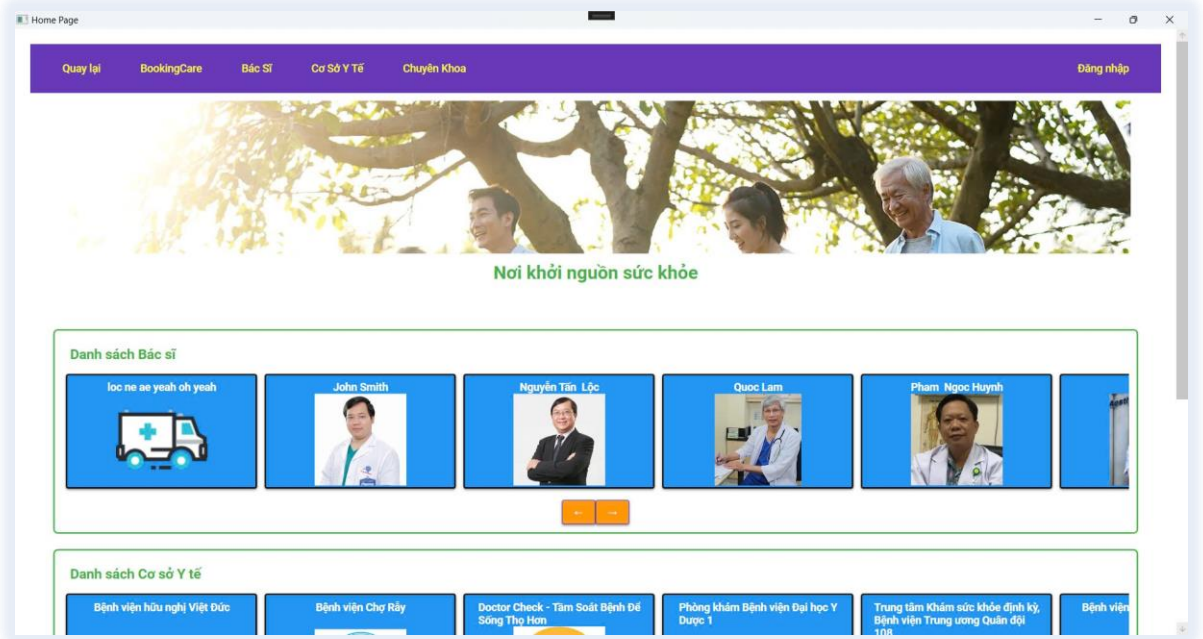
Hình 14. Giao diện đăng nhập thất bại

Sau khi sử dụng các chức năng trong hệ thống, người dùng có thể nhấn vào nút đăng xuất để trở lại trang chủ.

3.2.2. Kịch bản 2: Trang chủ người dùng

Người dùng không cần đăng nhập để sử dụng các chức năng nhưng cần nhập đầy đủ và chính xác thông tin trong đó có email là bắt buộc chính xác khi thực hiện chức năng đăng kí lịch khám bệnh.

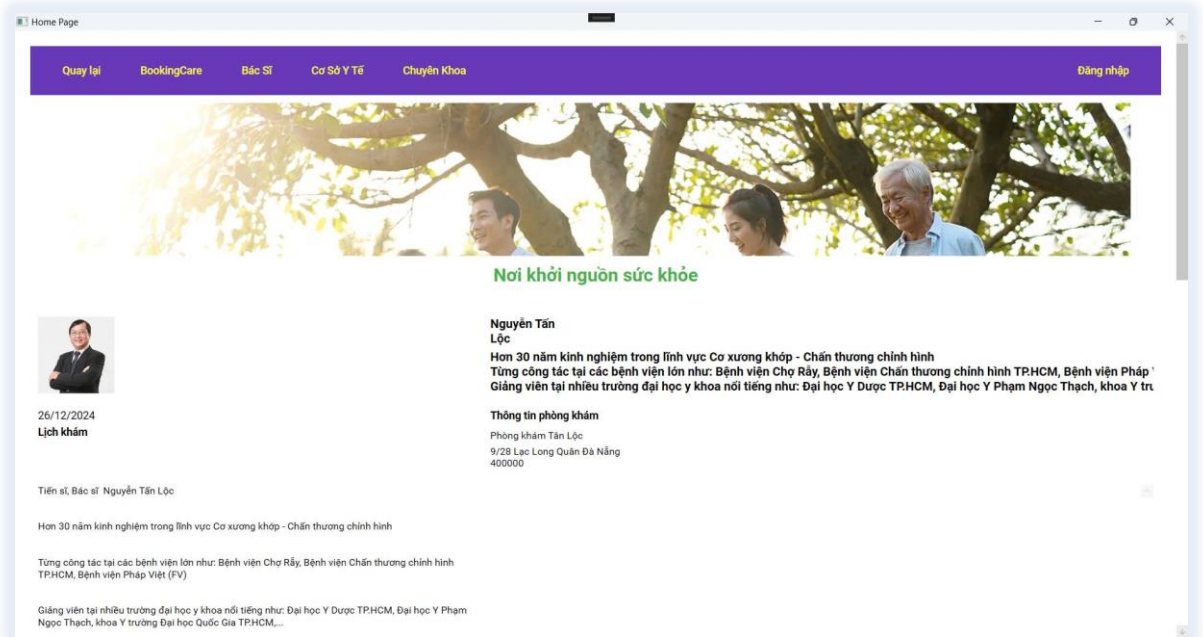
PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MANG MÁY TÍNH



Hình 15. Giao diện trang chủ

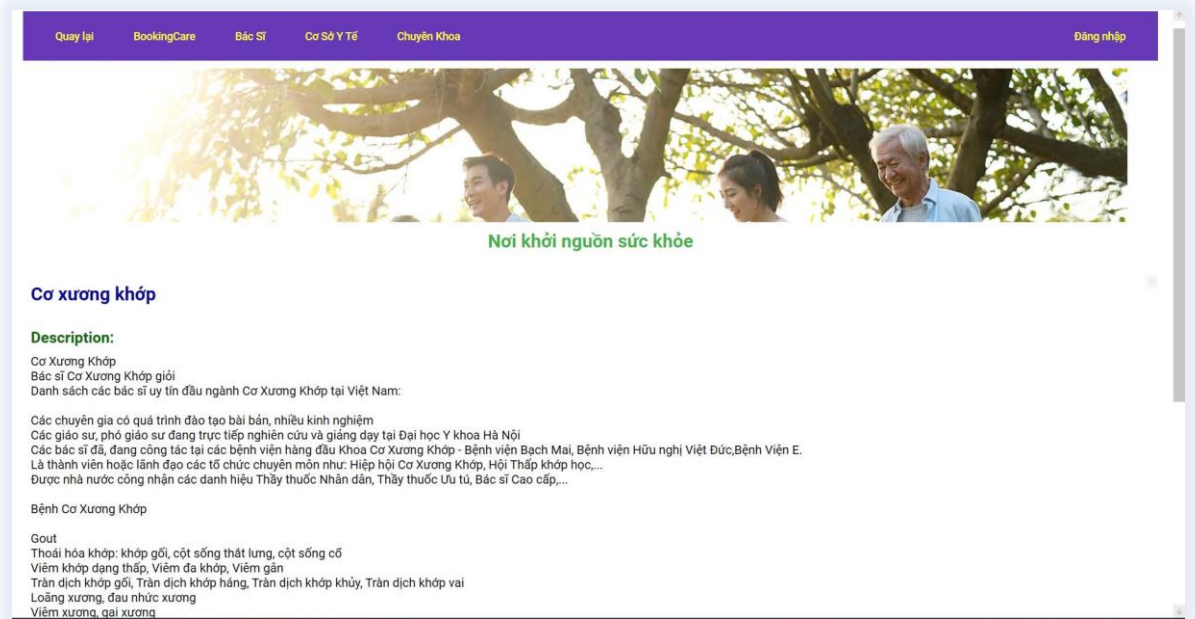
3.2.3. Kịch bản 3: Xem thông tin bác sĩ, phòng khám, chuyên khoa...

Giao diện của từng bác sĩ:



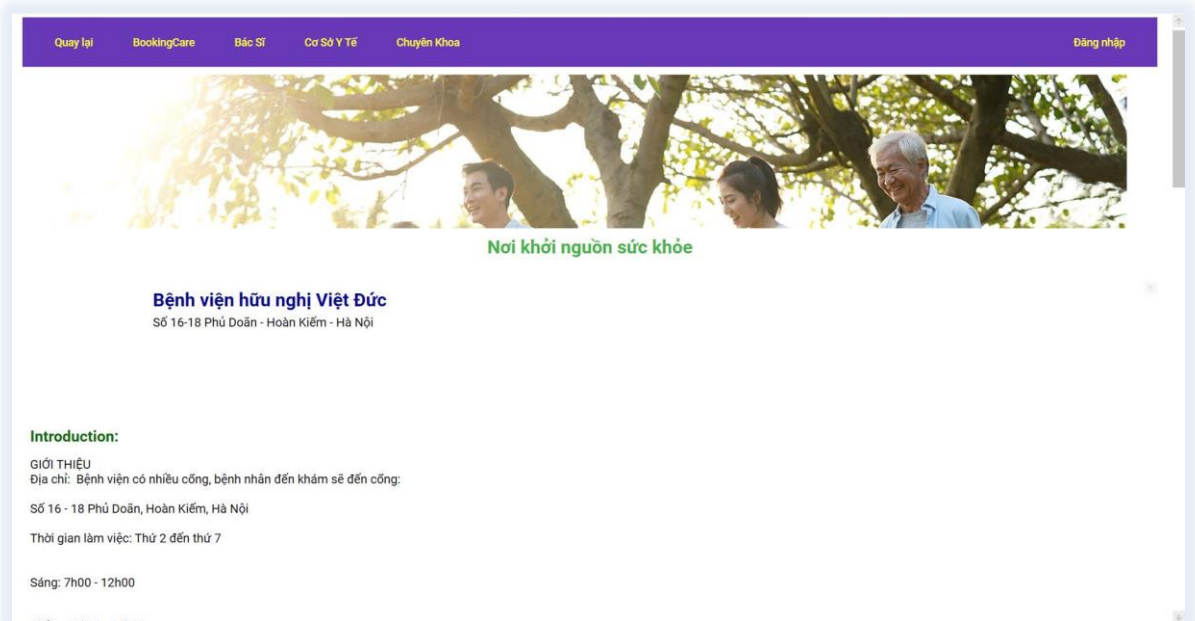
Hình 16. Giao diện bác sĩ

Giao diện của từng chuyên khoa:



Hình 17. Giao diện của chuyên khoa

Giao diện của từng cơ sở y tế:

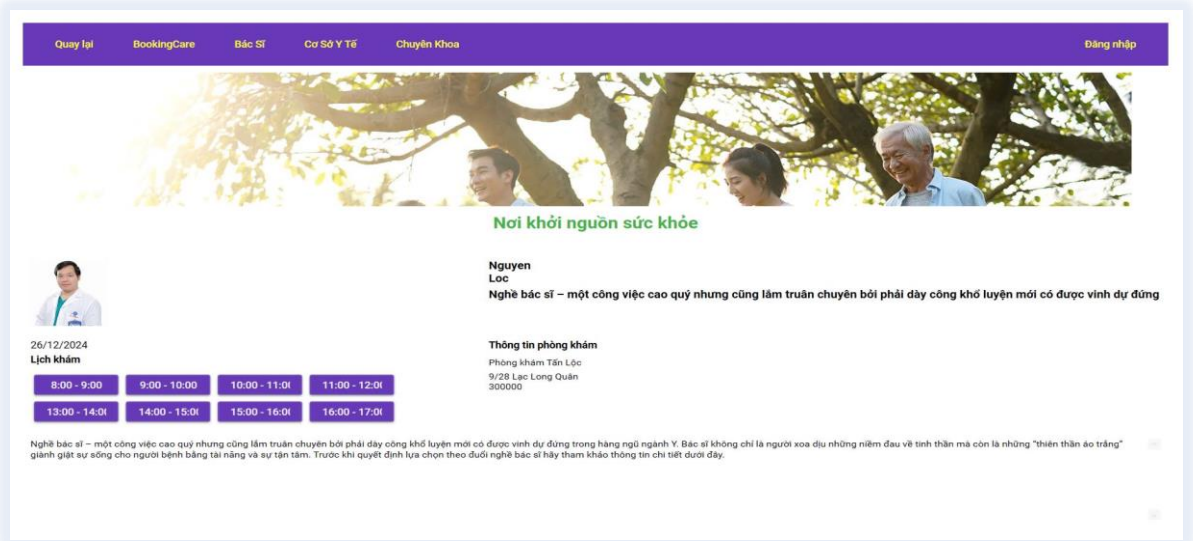


Hình 18. Giao diện của từng cơ sở y tế

3.2.4. Kịch bản 4: User thực hiện các chức năng

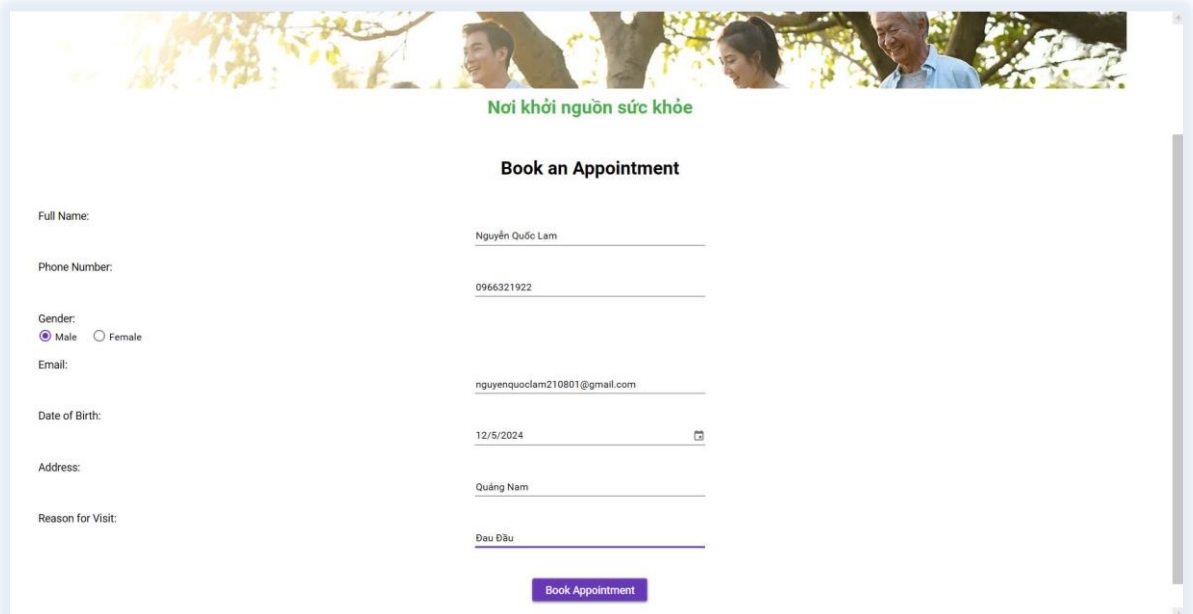
PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

Người dùng thực hiện chức năng đặt lịch khám:



The screenshot shows a doctor's profile page. At the top, there is a navigation bar with links: Quay lại, BookingCare, Bác Sĩ, Cơ Sở Y Tế, Chuyên Khoa, and Đăng nhập. Below the navigation bar is a banner image of a family. The main content area features a doctor's profile for "Nguyễn Lộc". The profile includes a photo of the doctor, a bio stating "Nghề bác sĩ - một công việc cao quý nhưng cũng lắm truân chuyên bởi phải dày công khổ luyện mới có được vinh dự đứng", and a section for "Thông tin phòng khám" (Clinic Information) listing "Phòng khám Tấn Lộc", "9/28 Lạc Long Quân", and "300000". A "Lịch khám" (Appointment Schedule) for "26/12/2024" is displayed with a grid of time slots: 8:00 - 9:00, 9:00 - 10:00, 10:00 - 11:00, 11:00 - 12:00, 13:00 - 14:00, 14:00 - 15:00, 15:00 - 16:00, and 16:00 - 17:00. A disclaimer at the bottom states: "Nghề bác sĩ - một công việc cao quý nhưng cũng lắm truân chuyên bởi phải dày công khổ luyện mới có được vinh dự đứng trong hàng ngũ ngành Y. Bác sĩ không chỉ là người xoa dịu những niềm đau về tinh thần mà còn là những 'thiên thần áo trắng' giành giật sự sống cho người bệnh bằng tài năng và sự tận tâm. Trước khi quyết định lựa chọn theo đuổi nghề bác sĩ hãy tham khảo thông tin chi tiết dưới đây."

Hình 19. Người dùng chọn lịch để đăng ký khám
Sau đó người dùng điền đầy đủ thông tin và nhấn vào nút đặt lịch khám:



The screenshot shows the "Book an Appointment" form. The form has a title "Book an Appointment" and a "Book Appointment" button. The form fields are: Full Name (Nguyễn Quốc Lam), Phone Number (0966321922), Gender (Male selected, Female unselected), Email (nguyenquoclam210801@gmail.com), Date of Birth (12/5/2024), Address (Quảng Nam), and Reason for Visit (Đau Đầu). The form is styled with a clean, modern design and a purple accent color.

Hình 20. Người dùng nhập đầy đủ thông tin khám bệnh

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

Nếu thành công người dùng sẽ có được thông báo và nhận được 1 email được gửi từ chương trình:

Nơi khởi nguồn sức khỏe

Book an Appointment

Full Name: Nguyễn Quốc Lam

Phone Number: 0966321922

Gender: ☒ Male ☐ Female

Email: nguyenguoclam.h.vu@nguyenguoclam.h.vu

Date of Birth: 12/5/2024

Address: Quảng Nam

Reason for Visit: Đau Đầu

[Book Appointment](#)

Bạn đã đặt lịch thành công!

OK

Hình 21. Người dùng đặt lịch thành công
Chương trình gửi 1 email thông báo về cho người dùng:



Hình 22. Email từ chương trình
Người dùng xác nhận đặt lịch sau khi nhận được email bằng cách nhấn vào nút xác nhận bên dưới:



Hình 23. Người dùng xác nhận khám bệnh

Người dùng đặt lịch thất bại:

The screenshot shows a web form titled "Book an Appointment" under the heading "Nơi khởi nguồn sức khỏe". The form fields are: Full Name (Nguyễn Quốc Lam), Phone Number, Gender (Male selected), Email, Date of Birth (12/5/2024), Address (Quảng Nam), and Reason for Visit (Đau Đầu). A purple "Book Appointment" button is at the bottom. A modal error message is displayed in the center, stating: "Thời gian này đã có người đặt trước, vui lòng chọn thời gian khác!" (This time has already been booked by someone, please choose a different time!).

Hình 24. Người dùng đặt lịch thất bại

3.2.5. Kịch bản 5: Doctor thực hiện các chức năng

- Quản lý lịch khám bệnh:

The screenshot shows a web interface for a doctor's appointment management. The top navigation bar has "Quản lý bệnh nhân" and "Quản lý lịch khám" (selected). The user is logged in as "Xin chào, Nguyen Loc". The main heading is "Quản lý lịch hẹn bác sĩ". Below this, there are input fields for "Tên bác sĩ" (filled with "Xin chào, Nguyen Loc") and "Ngày lên lịch" (12/26/2024). A blue "Lưu" (Save) button is on the right. At the bottom, there is a row of eight yellow buttons representing time slots: "8:00 - 9:00", "9:00 - 10:00", "10:00 - 11:00", "11:00 - 12:00", "13:00 - 14:00", "14:00 - 15:00", "15:00 - 16:00", and "16:00 - 17:00".

Hình 25. Giao diện quản lý lịch khám của bác sĩ

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

Bác sĩ chọn vào những lịch khám bệnh phù hợp và chọn vào ô ngày để đăng ký lịch phù hợp sau đó bấm lưu:

The screenshot shows a web application interface for managing doctor appointments. The header is blue with two tabs: 'Quản lý bệnh nhân' and 'Quản lý lịch khám'. The user is logged in as 'Xin chào, Nguyen Loc'. The main title is 'Quản lý lịch hẹn bác sĩ'. There are two input fields: 'Tên bác sĩ' with the value 'Xin chào, Nguyen Loc' and 'Ngày lên lịch' with the value '12/26/2024'. A calendar pop-up is displayed over the date field, showing 'Thu, Dec 26' for December 2024. Below the inputs is a row of time slots: 8:00 - 9:00, 9:00 - 10:00, 10:00 - 11:00, 11:00 - 12:00, 12:00 - 13:00, 13:00 - 14:00, 14:00 - 15:00, 15:00 - 16:00, and 16:00 - 17:00. A blue 'Lưu' button is on the right.

Hình 26. Chọn lịch khám phù hợp
Sau khi nhấn lưu sẽ có thông báo bác sĩ đặt lịch thành công:

The screenshot shows the same web application interface as Figure 26, but with a success message. The date field now shows '12/27/2024'. A modal dialog box is displayed in the center with the text 'Successful' and an 'OK' button. The time slots are the same as in Figure 26, but the '12:00 - 13:00' slot is now highlighted in yellow.

Hình 27. Đăng ký lịch khám thành công

- Quản lí lịch khám bệnh của bệnh nhân:

Quản lí bệnh nhân | Quản lí lịch khám Xin chào, Nguyen Loc

Quản lí bệnh nhân khám bệnh

Chọn ngày khám

Chọn 1 bản ghi để có thể nhấn nút xác nhận **Xác nhận**

Chọn 1 bản ghi để có thể nhấn nút huỷ lịch **Hủy bỏ**

Hình 28. Giao diện quản lí lịch khám bệnh
Bác sĩ chọn các ngày để hiển thị các lịch khám của bệnh nhân đã được xác nhận:

Quản lí bệnh nhân | Quản lí lịch khám Xin chào, Nguyen Loc

Quản lí bệnh nhân khám bệnh

Chọn ngày khám

Chọn 1 bản ghi để có thể nhấn nút xác nhận **Xác nhận**

Chọn 1 bản ghi để có thể nhấn nút huỷ lịch **Hủy bỏ**

| email | patientId | fullName | phone | address | gender | time | timeType |
|-------------------------------|-----------|-----------------|------------|-----------|--------|-------------|----------|
| nguyenquoclam210801@gmail.com | 24 | Nguyễn Quốc Lam | 0174687146 | Quảng Nam | Nữ | 8:00 - 9:00 | T1 |

Hình 29. Danh sách bệnh nhân đã đăng ký

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MANG MÁY TÍNH

Bác sĩ chọn một bản ghi và nhấn xác nhận để xác nhận việc hoàn thành khám bệnh cho bệnh nhân này và hệ thống gửi mail về cho bệnh nhân để thông báo:

Quản lý bệnh nhân | Quản lý lịch khám | Xin chào, Nguyễn Lạc

Quản lý bệnh nhân khám bệnh

Chọn ngày khám: 12/26/2024

| email | patientid | fullName |
|-------------------------------|-----------|-----------------|
| nguyenquoclam210801@gmail.com | 24 | Nguyễn Quốc Lam |

RemedyForm

Hình ảnh hoá đơn, đơn thuốc

Chọn ảnh

Mô tả

Uống 3 lần 1 ngày, sau khi ăn!

Gửi

Hình 30. Xác nhận gửi thông báo hoàn thành khám
Xác nhận hoàn thành khám bệnh thành công:

Quản lý bệnh nhân | Quản lý lịch khám | Xin chào, Nguyễn Lạc

Quản lý bệnh nhân khám bệnh

Chọn ngày khám: 12/26/2024

| email | patientid | fullName |
|-------------------------------|-----------|-----------------|
| nguyenquoclam210801@gmail.com | 24 | Nguyễn Quốc Lam |

RemedyForm

Hình ảnh hoá đơn, đơn thuốc

Chọn ảnh

Mô tả

Uống 3 lần 1 ngày, sau khi ăn!

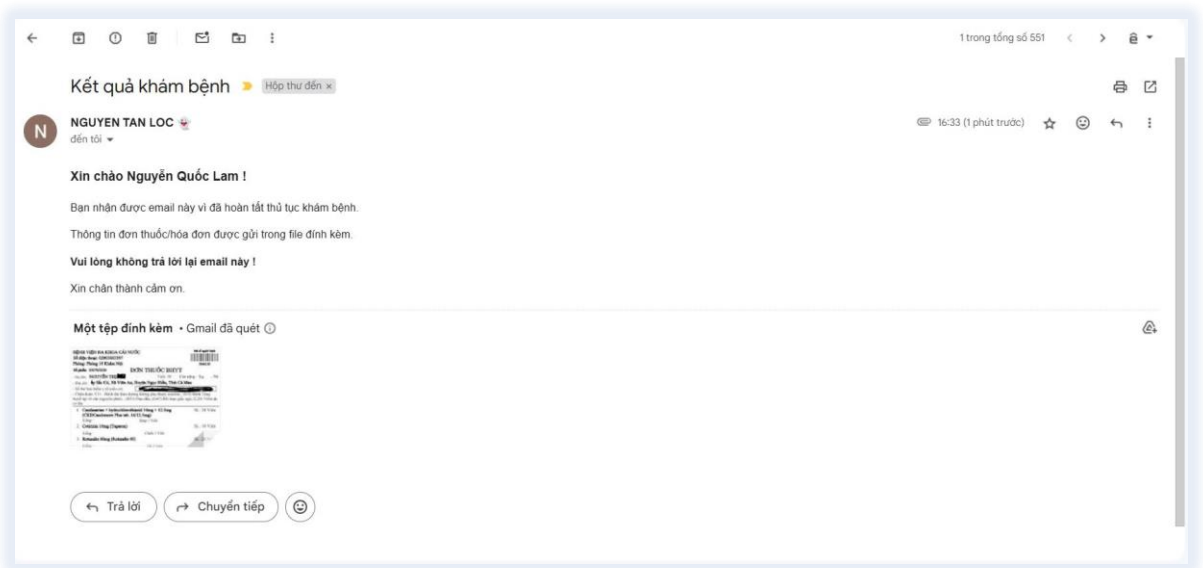
Gửi

Send email successful

OK

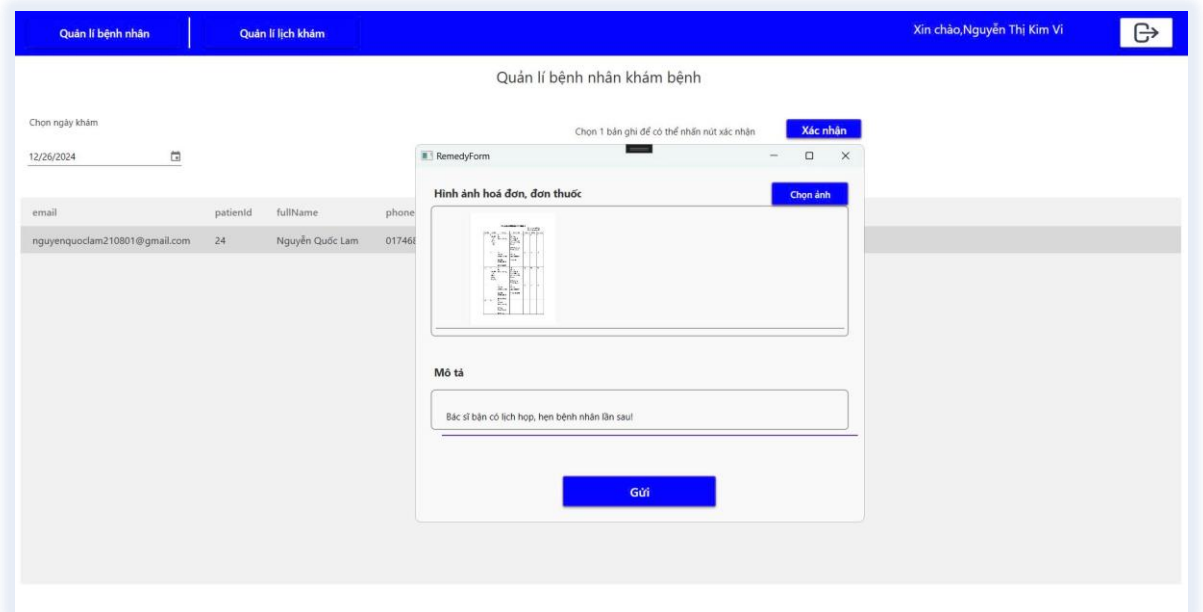
Hình 31. Gửi thông báo xác nhận cho người dùng

Email xác nhận được gửi về từ chương trình:



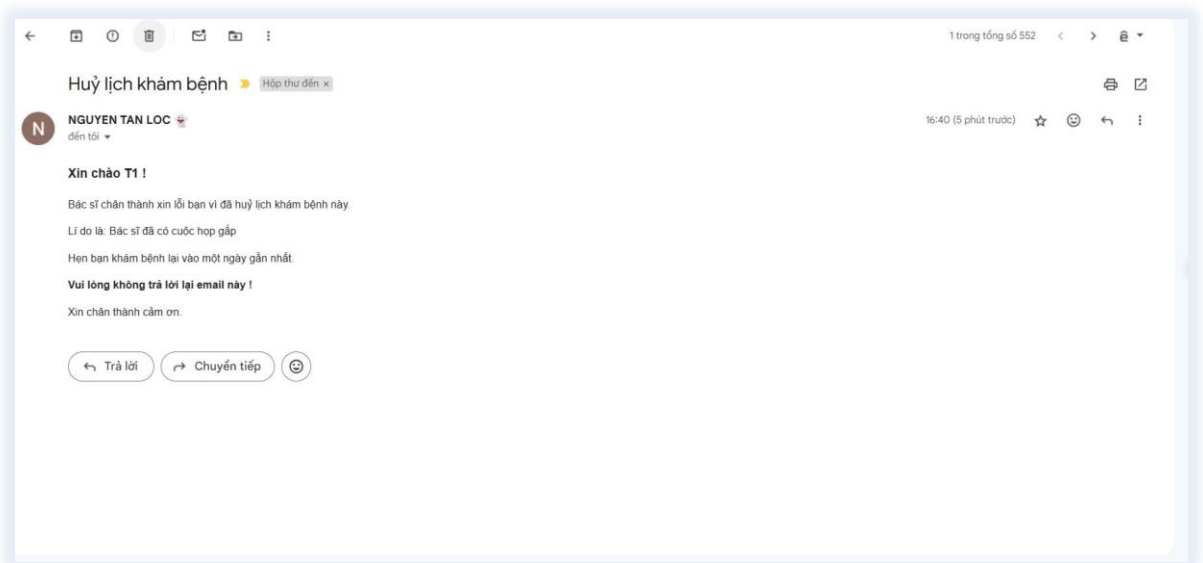
Hình 32. Email xác nhận hoàn thành khám bệnh

Bác sĩ chọn vào một bản ghi để hủy bỏ lịch khám nếu có các vấn đề xảy ra:



Hình 33. Gửi thông báo hủy lịch cho người dùng

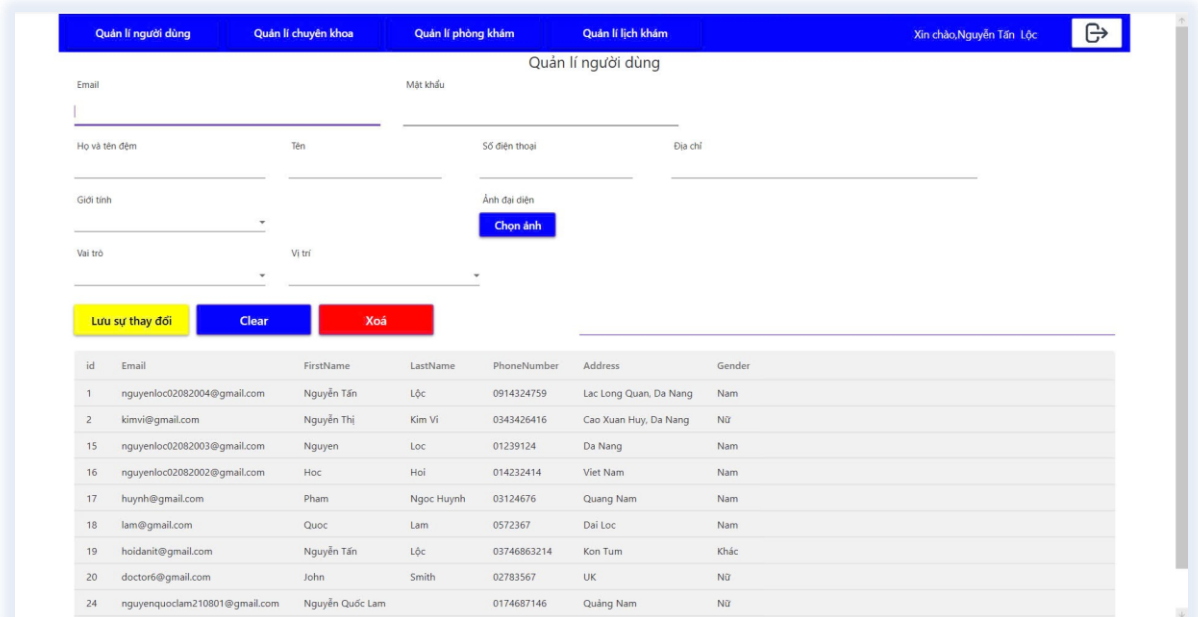
Hủy lịch thành công và gửi email thông báo về cho người dùng:



Hình 34. Email thông báo hủy lịch khám

3.2.6. Kịch bản 6: Admin thực hiện các chức năng

- Quản lý User:



Hình 35. Giao diện quản lý User

Admin có thể thêm User bằng cách điền đầy đủ thông tin và nhấn vào lưu sự thay đổi để thêm mới User:

The screenshot shows the 'Quản lý người dùng' (User Management) interface. At the top, there are navigation tabs: 'Quản lý người dùng', 'Quản lý chuyên khoa', 'Quản lý phòng khám', and 'Quản lý lịch khám'. The 'Quản lý người dùng' tab is active. Below the tabs, there is a form to add a new user. The form includes fields for Email, Mật khẩu (Password), and a 'Chọn ảnh' (Choose image) button. Below these, there are fields for 'Họ và tên đệm' (Middle name), 'Tên' (First name), 'Số điện thoại' (Phone number), and 'Địa chỉ' (Address). There are also dropdown menus for 'Giới tính' (Gender) and 'Vai trò' (Role). A 'Lưu sự thay đổi' (Save changes) button is visible. A modal window shows a success message: 'Create a new user successful!'. Below the form, there is a table listing existing users.

| id | Email | FirstName | LastName | PhoneNu | | Gender |
|----|-------------------------------|-----------------|------------|-------------|------------------------|--------|
| 1 | nguyenloc02082004@gmail.com | Nguyễn Tấn | Lộc | 0914324759 | Lac Long Quan, Da Nang | Nam |
| 2 | kimvi@gmail.com | Nguyễn Thị | Kim Vi | 0343426416 | Cao Xuan Huy, Da Nang | Nữ |
| 15 | nguyenloc02082003@gmail.com | Nguyen | Loc | 01239124 | Da Nang | Nam |
| 16 | nguyenloc02082002@gmail.com | Hoc | Hoi | 014232414 | Viet Nam | Nam |
| 17 | huyh@gmail.com | Pham | Ngoc Huynh | 03124676 | Quang Nam | Nam |
| 18 | lam@gmail.com | Quoc | Lam | 0572367 | Dai Loc | Nam |
| 19 | hoidanit@gmail.com | Nguyễn Tấn | Lộc | 03746863214 | Kon Tum | Khác |
| 20 | doctor6@gmail.com | John | Smith | 02783567 | UK | Nữ |
| 24 | nguyenquoclam210801@gmail.com | Nguyễn Quốc Lam | | 0174687146 | Quảng Nam | Nữ |

Hình 36. Thêm người dùng thành công

Admin chọn một bản ghi sau đó nhấn nút xóa thì sẽ xóa được một User thành công:

The screenshot shows the 'Quản lý người dùng' (User Management) interface. The 'Quản lý người dùng' tab is active. The form for adding a new user is visible. A modal window shows a success message: 'The user is deleted'. Below the form, there is a table listing existing users.

| id | Email | FirstName | LastName | PhoneNu | | Gender |
|----|-------------------------------|-----------------|--------------|-------------|-----------------------|--------|
| 2 | kimvi@gmail.com | Nguyễn Thị | Kim Vi | 0343426416 | Cao Xuan Huy, Da Nang | Nữ |
| 15 | nguyenloc02082003@gmail.com | Nguyen | Loc | 01239124 | Da Nang | Nam |
| 16 | nguyenloc02082002@gmail.com | Hoc | Hoi | 014232414 | Viet Nam | Nam |
| 17 | huyh@gmail.com | Pham | Ngoc Huynh | 03124676 | Quang Nam | Nam |
| 18 | lam@gmail.com | Quoc | Lam | 0572367 | Dai Loc | Nam |
| 19 | hoidanit@gmail.com | Nguyễn Tấn | Lộc | 03746863214 | Kon Tum | Khác |
| 20 | doctor6@gmail.com | John | Smith | 02783567 | UK | Nữ |
| 24 | nguyenquoclam210801@gmail.com | Nguyễn Quốc Lam | | 0174687146 | Quảng Nam | Nữ |
| 36 | 15dfhdfh | loc ne ae | yeah oh yeah | qwqr | ewqwtqthq | Nữ |

Hình 37. Xóa User thành công

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

Admin nhấn đúp vào 1 bản ghi để hiển thị thông tin người dùng lên giao diện và nhập các thông tin cần sửa sau đó nhấn lưu sự thay đổi sẽ lưu lại các thông tin mới được cập nhật của người dùng đó:

Quản lý người dùng

Email: lam@gmail.com | Mật khẩu: ****

Họ và tên đệm: Nguyễn | Tên: Lam | Số điện thoại: 0966321922 | Địa chỉ: Quảng Nam

Giới tính: Nam | Ảnh đại diện: [Chọn ảnh]

Vai trò: Bác sĩ | Vị trí: Bác sĩ

[Lưu sự thay đổi] [Clear] [Xoá]

update successfull [OK]

| id | Email | FirstName | LastName | PhoneNumber | Address | Gender |
|----|-----------------------------|------------|------------|-------------|------------------------|--------|
| 1 | nguyenloc02082004@gmail.com | Nguyễn Tấn | Lộc | 0914324759 | Lac Long Quan, Da Nang | Nam |
| 2 | kimvi@gmail.com | Nguyễn Thị | Kim Vi | 0343426416 | Cao Xuan Huy, Da Nang | Nữ |
| 15 | nguyenloc02082003@gmail.com | Nguyen | Loc | 01239124 | Da Nang | Nam |
| 16 | nguyenloc02082002@gmail.com | Hoc | Hoi | 014232414 | Viet Nam | Nam |
| 17 | huynh@gmail.com | Pham | Ngoc Huynh | 03124676 | Quang Nam | Nam |
| 18 | lam@gmail.com | Quoc | Lam | 0572367 | Dai Loc | Nam |
| 19 | hoidanit@gmail.com | Nguyễn Tấn | Lộc | 03746863214 | Kon Tum | Khác |
| 20 | doctor6@gmail.com | John | Smith | 02783567 | UK | Nữ |

Hình 38. Cập nhật User thành công

- Quản lý Phòng khám:

Quản lý phòng khám

Tên phòng khám: | Địa chỉ phòng khám: | Thông tin chi tiết:

Ảnh mô tả: [Chọn ảnh]

[Lưu thông tin]

| ID | Name | Address |
|----|--|--|
| 1 | Bệnh viện hữu nghị Việt Đức | Số 16-18 Phủ Doãn - Hoàn Kiếm - Hà Nội |
| 2 | Bệnh viện Chợ Rẫy | 201B Nguyễn Chí Thanh, Phường 12, Quận 5, Hồ Chí Minh |
| 3 | Doctor Check - Tầm Soát Bệnh Để Sống Thọ Hơn | 429 Tô Hiến Thành, Phường 14, Quận 10, Tp. Hồ Chí Minh |
| 4 | Phòng khám Bệnh viện Đại học Y Dược 1 | 20-22 Dương Quang Trung, Phường 12, Quận 10, Tp. HCM |
| 5 | Trung tâm Khám sức khỏe định kỳ, Bệnh viện Trung ương Quân đội 108 | Số 18 Trần Hưng Đạo, Quận Hai Bà Trưng, Hà Nội |

Hình 39. Giao diện quản lý phòng khám

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH

Admin nhập đầy đủ thông vào các trường thông tin và nhấn lưu thông tin để thêm Phòng khám vào hệ thống:

Quản lý phòng khám

Tên phòng khám:

Địa chỉ phòng khám:

Ảnh mô tả:

Thông tin chi tiết:

****GIỚI THIỆU****

****Địa chỉ**** Bệnh viện có nhiều cổng, bệnh nhân đến khám sẽ đến cổng:

* 201 Nguyễn Hữu Thọ, Đà Nẵng

****Thời gian làm việc****: Thứ 2 đến thứ 7

* Sáng: 7h00 - 12h00

* Chiều: 13h30 - 16h30

Bệnh viện Gia Đình là một trong 5 bệnh viện Việt Nam gần gũi với những thành tựu

Create clinic successfully

OK

Lưu thông tin

| ID | Name | Address |
|----|--|--|
| 1 | Bệnh viện hữu nghị Việt Đức | Số 16-18 Phủ Doãn - Hoàn Kiếm - Hà Nội |
| 2 | Bệnh viện Chợ Rẫy | 201B Nguyễn Chí Thanh, Phường 12, Quận 5, Hồ Chí Minh |
| 3 | Doctor Check - Tâm Soát Bệnh Để Sống Thọ Hơn | 429 Tô Hiến Thành, Phường 14, Quận 10, Tp. Hồ Chí Minh |
| 4 | Phòng khám Bệnh viện Đại học Y Dược 1 | 20-22 Dương Quang Trung, Phường 12, Quận 10, Tp. HCM |
| 5 | Trung tâm Khám sức khỏe định kỳ, Bệnh viện Trung ương Quân đội 108 | Số 18 Trần Hưng Đạo, Quận Hai Bà Trưng, Hà Nội |

Hình 40. Thêm thành công phòng khám

- Quản lý Chuyên khoa:

Quản lý chuyên khoa

Tên Chuyên khoa:

Ảnh mô tả:

Thông tin chi tiết:

Lưu thông tin

| ID | NameVi | NameEn |
|----|---------------|------------------|
| 1 | Cơ xương khớp | Muscle - Bone |
| 2 | Thần kinh | Mental |
| 3 | Tiểu hóa | Digestion |
| 4 | Tim mạch | Heart |
| 5 | Tai mũi họng | Ears-Eyes-Throat |

Hình 41. Giao diện quản lý chuyên khoa

PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH VÀ MANG MÁY TÍNH

Admin nhập đầy đủ thông vào các trường thông tin và nhấn lưu thông tin để thêm Chuyên khoa vào hệ thống:

| ID | NameVi | NameEn |
|----|---------------|------------------|
| 1 | Cơ xương khớp | Muscle - Bone |
| 2 | Thần kinh | Mental |
| 3 | Tiêu hóa | Digestion |
| 4 | Tim mạch | Heart |
| 5 | Tai mũi họng | Ears-Eyes-Throat |

Hình 42. Thêm thành công chuyên khoa

- Quản lý Lịch khám: chức năng này tương tự với Doctor nhưng người dùng được chọn Bác sĩ trước khi đăng kí lịch khám

Chọn bác sĩ: _____

Chọn lịch khám: _____

Lưu

8:00 - 9:00 9:00 - 10:00 10:00 - 11:00 11:00 - 12:00 13:00 - 14:00 14:00 - 15:00 15:00 - 16:00 16:00 - 17:00

Hình 43. Giao diện quản lý lịch khám

Admin chọn bác sĩ và chọn lịch, sau đó lưu lịch đăng kí:

Quản lý lịch khám bệnh

Chọn bác sĩ: Nguyễn Lộc

Chọn lịch khám: 12/27/2024

Lưu

8:00 - 10:00 10:00 - 11:00 11:00 - 12:00 13:00 - 14:00 14:00 - 15:00 15:00 - 16:00 16:00 - 17:00

Hình 44. Chọn bác sĩ và chọn lịch

Đăng kí lịch thành công:

Quản lý lịch khám bệnh

Chọn bác sĩ: Nguyễn Lộc

Chọn lịch khám: 12/27/2024

Lưu

Successful

OK

8:00 - 9:00 9:00 - 10:00 10:00 - 11:00 11:00 - 12:00 13:00 - 14:00 14:00 - 15:00 15:00 - 16:00 16:00 - 17:00

Hình 45. Đăng kí lịch thành công

3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ

Qua kết quả thực nghiệm, tác giả có những nhận xét như sau:

- Vấn đề Giao diện: giao diện cũng đã được xây dựng cơ bản hoàn thành nhưng cần làm đẹp và phức tạp hơn để đem lại cảm giác tốt nhất cho người dùng.

- Vấn đề Chức năng: về cơ bản đã đầy đủ các chức năng cần có trong hệ thống. Hệ thống còn đơn giản, cần thời gian phát triển hoàn thiện hơn từ cơ sở dữ liệu đến các chức năng khác. Các nghiệp vụ xử lý ở phần server đều được xử lý cụ thể, chi tiết đối với từng đối tượng riêng biệt. Xử lý lỗi, bảo mật ở mức cơ bản.
- Vấn đề Xử lý lỗi: các lỗi khi người dùng sử dụng ứng dụng đề được thông báo đến cho người dùng, từ việc đăng nhập, đăng kí lịch và cả những chức năng quản lí các đối tượng.

3.4. KẾT CHƯƠng

Trong chương 3 này, chi tiết quá trình triển khai hệ thống ứng dụng từ việc cài đặt các môi trường phát triển cho đến cấu hình hệ thống đã được trình bày rõ ràng. Quá trình triển khai đã được thực hiện qua nhiều bước, bao gồm cài đặt phần mềm, cấu hình máy chủ, kết nối giữa các thành phần hệ thống như backend, frontend và cơ sở dữ liệu, cũng như kiểm tra và đánh giá hiệu quả của hệ thống.

Bên cạnh đó, quá trình đánh giá kết quả sau triển khai được thực hiện thông qua việc kiểm tra các tính năng của hệ thống, hiệu suất hoạt động và khả năng tương tác giữa các thành phần. Các công cụ và phương pháp đánh giá hiệu quả hệ thống đã giúp nhận diện và khắc phục các vấn đề phát sinh trong quá trình triển khai.

Kết quả của quá trình triển khai cho thấy hệ thống hoạt động ổn định, đáp ứng đầy đủ các yêu cầu chức năng đã đề ra. Các yếu tố về hiệu suất và khả năng mở rộng cũng được đảm bảo, mang lại trải nghiệm người dùng mượt mà và hiệu quả.

Cuối cùng, các kết quả đánh giá cũng giúp định hướng cho các bước tiếp theo trong quá trình phát triển và hoàn thiện hệ thống. Dự án đã đạt được mục tiêu và sẽ tiếp tục được hoàn thiện qua các giai đoạn tiếp theo, với những cải tiến và tối ưu hóa dựa trên phản hồi từ người dùng và các kết quả đánh giá.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đồ án đã đạt được những kết quả sau:

Về mặt lý thuyết:

- Hiểu rõ được các mô hình như client-server hay restful API.
- Biết được cách xây dựng, tạo API để tạo sợi dây kết nối request và response.
- Biết cách sử dụng các công cụ hỗ trợ test dữ liệu và quản lý mã nguồn rõ hơn.
- Phân biệt, tìm hiểu được các môi trường triển khai và phát triển ứng dụng.
- Nắm rõ hơn các kiến thức về lập trình hướng đối tượng và hiểu được cách ánh xạ dữ liệu giữa các bảng và các đối tượng trong hệ thống.

Đồ án đã đạt được: Sự phát triển của công nghệ thông tin đã giúp việc tìm kiếm và đăng ký khám bệnh không còn khó khăn như trước nữa. Vấn đề về thời gian, khoảng cách hay các điều kiện không tốt dẫn đến việc bạn không tìm được các trung tâm y tế hay phòng khám phù hợp với bản thân mình đã được ứng dụng này giải quyết.

Ưu điểm:

- Đầy đủ các chức năng cơ bản nhất theo yêu cầu của người dùng.
- Hoạt động tốt và xử lý nhanh.
- Giao diện thân thiện và dễ sử dụng.

Khuyết điểm:

- Cơ sở dữ liệu chưa được hoàn chỉnh và chính xác hoàn toàn.
- Cần cải thiện thêm việc tự động làm mới cơ sở dữ liệu. Vì đây là chương trình ứng dụng app desktop nên cần có PC để sử dụng, còn chạy ở localhost, cơ sở dữ liệu vẫn còn nằm ở máy cá nhân nên chưa tự động cập nhật liên tục gây ra sự bất ổn định cho người dùng.

2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Thiết kế giao diện: Nâng cấp giao diện người dùng của ứng dụng để đảm bảo tính thân thiện, dễ sử dụng và hấp dẫn hơn. Sử dụng các thư viện đồ họa và framework mới nhất để tạo ra một giao diện đẹp và tương tác tốt hơn cho người dùng.

Phát triển hoàn chỉnh mô hình Client-Server, sử dụng các công nghệ web hiện nay để phát triển ứng dụng Hỗ trợ tìm kiếm và đăng ký khám bệnh nhằm mục đích phục vụ cho người dùng một cách tốt nhất. Khi phát triển, triển khai thành một ứng dụng web hoàn chỉnh thì mọi thứ sẽ được cập nhật tự động, các yêu cầu từ người dùng sẽ được phản hồi nhanh hơn. Tích hợp được nhiều tính năng hiện nay như google map hay có thể hỗ trợ đa ngôn ngữ.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Đặng Văn Đức (2001), Hệ thống thông tin địa lý, Nhà xuất bản Khoa học và Kỹ Thuật Hà Nội.
- [2] Phạm Hữu Đức (2005), Cơ sở dữ liệu và hệ thống thông tin địa lý GIS, Nhà xuất bản Xây dựng.

Tiếng Anh

- [3] Senthil Shanmugan (2004). “Digital urban management programme - Evolution of Bangalore GIS model”, Proceedings of the third International Conference on Environment and Health, India.
- [4] Teemu Nuortio, Harri Niska (2003), “Improved route planning and scheduling of waste collection and transport”, Department of Environmental Sciences, University of Kuopio, Finland.

Internet

- [5] [http:// www.mapreduce.org](http://www.mapreduce.org)
- [6] [http:// www.vre.cse.hcmut.edu.vn](http://www.vre.cse.hcmut.edu.vn)