**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**NGUYỄN TUẤN KIỆT**

**WEBSITE ĐẶT LỊCH KHÁM BỆNH SỬ DỤNG ASP.NET VÀ REACTJS**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2024BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**NGUYỄN TUẤN KIỆT**

**WEBSITE ĐẶT LỊCH KHÁM BỆNH SỬ DỤNG ASP.NET VÀ REACTJS**

**Mã số sinh viên: 2151013046**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: LƯU QUANG PHƯƠNG**

**TP. HỒ CHÍ MINH, 2024**

LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn tình cảm mà các quý Thầy Cô khoa Công Nghệ Thông Tin trường Đại học Mở TP.HCM đã giảng dạy tận tình và truyền đạt những kiến thức cốt lõi, cần thiết và ý nghĩa để em có một nền tảng vững chắc và kinh nghiệm quý báu để có thể thực hiện bài đồ án ngành này. Đặc biệt là Thầy Ths. Lưu Quang Phương đã tận tình hướng dẫn và có những đánh giá trong suốt quá trình thực hiện đồ án này.

Tuy nhiên thời gian thì cũng có hạn, em cũng đã không thể phát huy được toàn bộ ý tưởng, những hỗ trợ của ngôn ngữ lập trình, khả năng của công nghệ và kỹ thuật lập trình áp dụng vào đề tài, nên trong quá trình xây dựng và phát triển hệ thống em đã không thể tránh khỏi những sai sót, em mong nhận được những lời nhận xét, đóng góp của quý thầy cô để có thể cải thiện được bản thân và có thêm kinh nghiệm trong tương lai.

Em xin chân thành cảm ơn.

TP. Hồ Chí Minh, ngày 12/10/2024

Sinh viên thực hiện

Nguyễn Tuấn Kiệt

Lớp DH21CS02 – Trường đại học Mở TP.HCM

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

Đồ án có nội dung xây dựng một ứng dụng đặt lịch khám bệnh dựa trên cơ sở dữ liệu trực tuyến và mô hình đặt lịch khám bệnh của các bệnh viện tư.

Đồ án chia hệ thống thành 4 đối tượng khác nhau bao gồm: bệnh nhân, bác sĩ, y tá và người quản trị.

Trong đó vai trò của người quản trị là quan trọng nhất, bao gồm xây dựng, thống kê, tạo mới, quản lý tất cả tài khoản, hóa đơn, thuốc, lịch khám, lịch trực, thực hiện được mọi chức năng của những vai trò khác.

Vai trò của bệnh nhân là đặt lịch và thanh toán hóa đơn.

Vai trò của y tá là xác nhận lịch khám và xuất hóa đơn cho bệnh nhân.

Vai trò của bác sĩ là nhận lịch khám, chẩn đoán bệnh, tìm thuốc và ra toa thuốc cho bệnh nhân.

Đồ án mang tính xã hội cao, gần gũi với những hoạt động thường trực của bệnh nhân, y tá, bác sĩ và người quản trị. Góp phần giúp các bệnh viện tư áp dụng được công nghệ hiện đại trong việc quản lý lịch khám, quản lý thuốc để hỗ trợ phòng khám trong việc khám, chữa bệnh. Đồng thời cũng góp phần giải quyết những khó khăn trong việc đặt lịch khám, giúp bệnh nhân có thể đặt lịch khám để có thể khám bệnh một các dễ dàng.

Khi áp dụng mô hình đặt lịch khám và chữa bệnh thì có thể giúp các bệnh viện tư nhân có thể dễ dàng quản lý những thông tin cần thiết và tiết kiệm thời gian và công sức cho các bác sĩ và y tá, mang đến những lợi ích mà trước kia không có.

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc179655935)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc179655936)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc179655937)

[ABSTRACT 3](#_Toc179655938)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc179655939)

[DANH MỤC HÌNH VẼ 8](#_Toc179655940)

[DANH MỤC BẢNG 9](#_Toc179655941)

[MỞ ĐẦU 10](#_Toc179655942)

[Chương 1. TỔNG QUAN ĐỀ TÀI 11](#_Toc179655943)

[1.1. Giới thiệu đề tài 11](#_Toc179655944)

[1.2. Lý do chọn đề tài 11](#_Toc179655945)

[1.3. Mục tiêu nghiên cứu 12](#_Toc179655946)

[1.4. Bố cục báo cáo 12](#_Toc179655947)

[Chương 2. CƠ SỞ LÝ THUYẾT 13](#_Toc179655948)

[2.1. Giới thiệu về hệ thống quản lý phòng khám 13](#_Toc179655949)

[2.1.1. Khái niệm về hệ thống quản lý phòng khám 13](#_Toc179655950)

[2.1.2. Nhu cầu hiện nay 13](#_Toc179655951)

[2.1.3. Lợi ích của hệ thống quản lý phòng khám 14](#_Toc179655952)

[2.1.4. Hạn chế của hệ thống quản lý phòng khám 14](#_Toc179655953)

[2.2. Công nghệ áp dụng trong đề tài 14](#_Toc179655954)

[2.2.1. PostgreSQL 14](#_Toc179655955)

[2.2.2. Ngôn ngữ lập trình C# 15](#_Toc179655956)

[2.2.3. Ngôn ngữ lập trình JavaScript 17](#_Toc179655957)

[Chương 3. NỘI DUNG THỰC HIỆN 18](#_Toc179655958)

[3.1. Các bước thực hiện đề tài 18](#_Toc179655959)

[3.2. Mô tả chức năng nghiệp vụ chính trong đề tài 20](#_Toc179655960)

[3.3. Mô tả chức năng của đề tài 21](#_Toc179655961)

[3.3.1. Lược đồ Use Case 21](#_Toc179655962)

[3.4. Mô tả dữ liệu 31](#_Toc179655963)

[3.4.1. Sơ đồ lớp 31](#_Toc179655964)

[3.4.2. Lược đồ cơ sở dữ liệu quan hệ 32](#_Toc179655965)

[3.4.3. Chi tiết các thuộc tính trong từng bảng trong cơ sở dữ liệu 34](#_Toc179655966)

[3.5. Sơ đồ hoạt động 40](#_Toc179655967)

[3.6. Sơ đồ tuần tự 44](#_Toc179655968)

[3.7. Các giao diện theo vai trò 48](#_Toc179655969)

[3.7.1. Giao diện đăng nhập và đăng ký 48](#_Toc179655970)

[3.7.2. Giao diện cho Patient 49](#_Toc179655971)

[3.7.3. Giao diện cho Nurse 51](#_Toc179655972)

[3.7.4. Giao diện cho Doctor 53](#_Toc179655973)

[Chương 4. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 54](#_Toc179655974)

[4.1. Kết quả đạt được 54](#_Toc179655975)

[4.1.1. Đối với cá nhân 54](#_Toc179655976)

[4.1.2. Đối với sản phẩm 54](#_Toc179655977)

[4.2. Hạn chế 54](#_Toc179655978)

[4.2.1. Đối với hệ thống 54](#_Toc179655979)

[4.2.2. Đối với bệnh nhân 55](#_Toc179655980)

[4.2.3. Đối với y tá 55](#_Toc179655981)

[4.2.4. Đối với bác sĩ 55](#_Toc179655982)

[4.3. Hướng phát triển 55](#_Toc179655983)

[TÀI LIỆU THAM KHẢO 56](#_Toc179655984)

[PHỤ LỤC 56](#_Toc179655985)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| API | Application Prgramming Interface |
| ASP | Active Server Pages |
| SQL | Structre Query Language |
| ORM | Object Relational Mapping |
| RDBMS | Relational Database Management System |
| DOM | Document Object Model |
| HTML | HyperText Markup Language |
| JSON | JavaScript Object Notation |
| XML | Extensible Markup Language |
| ADO | ActiveX Data Object |

DANH MỤC HÌNH VẼ

[Hình 2.1: Mô hình ASP.NET Core Web API 16](#_Toc179655986)

[Hình 2.2: Mô hình DOM ảo trong React 18](#_Toc179655987)

[Hình 3.1: Lược đồ Use Case. 21](#_Toc179655988)

[Hình 3.2: Sơ đồ lớp 31](#_Toc179655989)

[Hình 3.3: Lược đồ sơ sở dữ liệu quan hệ. 32](#_Toc179655990)

[Hình 3.4: Sơ đồ hoạt động của chức năng đổi mật khẩu. 40](#_Toc179655991)

[Hình 3.5: Sơ đồ hoạt động của chức năng đặt lịch khám. 41](#_Toc179655992)

[Hình 3.6: Sơ đồ hoạt động của chức năng quản lý thuốc. 42](#_Toc179655993)

[Hình 3.7: Sơ đồ hoạt động của chức năng thanh toán. 43](#_Toc179655994)

[Hình 3.8: Sơ đồ tuần tự của chức năng cập nhật tài khoản 44](#_Toc179655995)

[Hình 3.9: Sơ đồ tuần tự chức năng đặt lịch khám. 45](#_Toc179655996)

[Hình 3.10: Sơ đồ tuần tự chức năng kê toa thuốc. 46](#_Toc179655997)

[Hình 3.11: Sơ đồ tuần tự cho chức năng thanh toán. 47](#_Toc179655998)

[Hình 3.12: Giao diện đăng nhập người dùng 48](#_Toc179655999)

[Hình 3.13: Giao diện đăng ký người dùng với vai trò là bệnh nhân 48](#_Toc179656000)

[Hình 3.14: Giao diện màn hình chính cho Patient 49](#_Toc179656001)

[Hình 3.15: Giao diện đặt lịch cho Patient 49](#_Toc179656002)

[Hình 3.16: Giao diện hóa đơn người dùng 50](#_Toc179656003)

[Hình 3.17: Giao diện thông tin Patient 50](#_Toc179656004)

[Hình 3.18: Giao diện chính cho Nurse 51](#_Toc179656005)

[Hình 3.19: Giao diện hóa đơn cho Nurse 51](#_Toc179656006)

[Hình 3.20: Giao diện hồ sơ cho Nurse 52](#_Toc179656007)

[Hình 3.21: Giao diện chính cho Doctor 53](#_Toc179656008)

[Hình 3.22: Giao diện hồ sơ cho Doctor 53](#_Toc179656009)

DANH MỤC BẢNG

[Bảng 3.1: Bảng đặc tả use case chức năng đăng nhập. 22](#_Toc179656010)

[Bảng 3.2: Bảng đặc tả use case chức năng đăng xuất. 23](#_Toc179656011)

[Bảng 3.3: Bảng đặc tả use case chức năng quản lý người dùng. 24](#_Toc179656012)

[Bảng 3.4: Bảng đặc tả use case chức năng kê toa thuốc. 25](#_Toc179656013)

[Bảng 3.5: Bảng đặc tả use case chức năng tìm kiếm thuốc. 26](#_Toc179656014)

[Bảng 3.6: Bảng đặc tả use case chức năng tìm kiếm lịch khám. 27](#_Toc179656015)

[Bảng 3.7: Bảng đặc tả use case chức năng tạo lịch khám. 28](#_Toc179656016)

[Bảng 3.8: Bảng đặc tả use case chức năng xác nhận lịch khám. 29](#_Toc179656017)

[Bảng 3.9: Bảng đặc tả use case chức năng xuất hóa đơn. 30](#_Toc179656018)

MỞ ĐẦU

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

Đề tài xây dựng hệ thống quản lý phòng khám dựa trên cơ sở dữ liệu trực tuyến và mô hình quản lý, khám và chữa bệnh của các bệnh viện tư hiện nay.

Trong bối cảnh khi mà công nghệ đang phát triển hiện nay, mọi thứ đề đã có thể dễ dàng thực hiện thông qua các thiết bị công nghệ thông minh và có kết nối Internet. “Hệ thống quản lý phòng khám” được sinh ra để có thể giúp cho nhiều người có thể dễ dàng đặt lịch và khám bệnh một cách dễ dàng ngay tại nhà.

Đề tài “Hệ thống quản lý phòng khám” nhằm xây dựng một hệ thống thông tin đáng tin cậy và hiệu quả để quản lý các hoạt động trong phòng khám y tế. Góp phần giúp cho phòng khám có thể tiết kiệm thời gian và công sức cho các y tá, bác sĩ trong việc phải vừa khám chữa bệnh và vừa phải quản lý sổ sách giấy tờ về ngày khám mà trước đây phải thực hiện thủ công sau một ngày làm việc.

## Lý do chọn đề tài

Trước đây, khi mà trước khi đi khám bệnh, mọi người thường phải xếp hàng và chờ lấy số thứ tự để có thể tới lượt của mình được khám bệnh, việc này có thể gây khó khăn đến cho nhiều người, ví dụ như có những người họ sống một mình, việc phải vừa lên bệnh viện, sau đó xếp hàng và chờ đến số thứ tự khiến cho việc đi khám bệnh trở nên khó khăn hơn cho họ và làm cho họ, hoặc có những người già hay trẻ nhỏ chỉ có người trong coi thì việc lấy số càng trở nên khó khăn hơn khi vừa phải trông coi người bệnh mà còn vừa phải xếp hàng lấy số thứ tự. Nhưng với sự phát triển công nghệ hiện nay, đặc biệt là trong ngành công nghệ thông tin, thì việc mà khám chữa bệnh đã không còn trở nên khó khăn hơn khi mà đã có “Hệ thống quản lý phòng khám”, nơi mà mọi người có thể đặt lịch khám ngay tại nhà, đồng thời cũng giúp người dùng và bệnh nhân dễ dàng quản lý lịch khám, hóa đơn khám bệnh.

Ngoài ra y tá cũng có thể dễ dàng quản lý lịch khám bằng cách kiểm tra lịch khám của bệnh nhân trước một ngày, đồng thời y tá cũng sẽ xuất hóa đơn khám bệnh cho người nhà hoặc người giám sát bệnh nhân sau khi bệnh nhân đã được bác sĩ khám chữa bệnh xong.

Bác sĩ cũng sẽ là người khám chữa bệnh sau khi lấy thông tin về lịch khám và lý do khám bệnh, chẩn đoán triệu chứng và kết luận bệnh, đồng thời cũng sẽ là người kê toa thuốc uống phù hợp cho bệnh nhân.

Quản trị viên có thể dễ dàng xuất ra báo cáo linh hoạt theo tháng hoặc theo quý, đồng thời cũng dễ dàng để quản lý thông tin người dùng

## Mục tiêu nghiên cứu

* Nghiên cứu về hệ thống quản lý phòng khám.
* Xây dựng cơ sở dữ liệu với cơ sở dữ liệu qua hệ PostgreSQL.
* Xây dựng website với ASP.NET Core Web API và JavaScript ReactJS.

## Bố cục báo cáo

Báo cáo được chia thành các chương sau:

* Chương 1: Tổng quan đề tài.
* Chương 2: Cơ sở lý thuyết.
* Chương 3: Nội dung thực hiện.

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về hệ thống quản lý phòng khám

### Khái niệm về hệ thống quản lý phòng khám

Quản lý phòng khám là việc người dùng hay bệnh nhân có thể tiến hành đặt lịch hẹn khám bệnh trên website, ứng dụng đã được thiết kế phù hợp cho các phòng khám dựa trên cơ sở dữ liệu đã được cung cấp.

Hệ thống quản lý phòng khám bao gồm hai dịch vụ chính là khám bệnh và quản lý. Dịch vụ khám bệnh là các tác vụ phục vụ cho phía người dùng, bệnh nhân, giúp mang lại trải nghiệm khám bệnh ổn định và dễ tiếp cận. Dịch vụ quản lý là bên phía y tá, bác sĩ và quản trị viên, có các chức năng tiêu biểu như xác nhận, cập nhật, tạo mới và chỉnh sửa.

### Nhu cầu hiện nay

Trong thời đại 4.0 hiện nay, các công nghệ hiện đại đã được áp dụng vào trong cuộc sống để hỗ trợ các công việc truyền thống trở nên dễ dàng hơn. Việc áp dụng các công nghệ hiện đại để xây dựng hệ thống website, ứng dụng nhằm mục đích phục vụ cho công việc quản lý và khám chữa bệnh ở Việt Nam hiện nay cũng mới chỉ vừa mới nổi trong những năm trở lại đây.

Đặc biệt là trong thời đại hiện nay, khi mà mọi người quá bận rộn, người bệnh thì không thể tự đến phòng khám và đăng ký khám bệnh, hay những người giám sát, người chăm sóc thì họ không thể vừa phải đưa người bệnh đến khám, vừa phải chờ để đến lượt thì “Hệ thống quản lý phòng khám” như một giải pháp, một cứu cánh cho nhiều bệnh nhân, những người giám sát, khi mà đã không còn khó khăn cho việc đi khám bệnh.

Tuy vậy, hệ thống quản lý phòng khám dành cho những người bệnh và người giám sát vẫn chưa được phổ biến, nhiều bệnh viện tư vẫn còn áp dụng phương pháp cũ là chờ lấy phiếu và đợi gọi tên đến lượt khám. Chưa theo kịp sự thay đổi về công nghệ trong ngành y tế ngày càng tiến bộ của Việt Nam. Nắm bắt được những xu thế, nhu cầu đó, em quyết định xây dựng đồ án với đề tài xây dựng hệ thống quản lý phòng khám dành cho các phòng khám tư.

### Lợi ích của hệ thống quản lý phòng khám

Hệ thống giải quyết được những khuyết điểm của hình thức khám bệnh truyền thống như phải chờ đợi lấy số, phải chờ đến lượt được khám, thanh toán chi phí khám bằng giấy tờ phức tạp, kiểm tra thuốc được kê toa.

Hạn chế được nhiều sai sót như khi thanh toán hóa đơn bằng giấy viết tay, không bị xáo trộn thứ tự khám khi có nhiều người đến đăng ký.

Mang lại giải pháp đặt lịch nhanh chóng, tự động, dễ dàng xác nhận và có thể dễ dàng hủy nếu có việc đột xuất xảy ra.

Tiết kiệm chi phí cho việc in ấn hóa đơn, giấy tờ khám bệnh, tiền quản lý,…

Dễ dàng quản lý được lịch khám, hóa đơn, thuốc,…

### Hạn chế của hệ thống quản lý phòng khám

Dễ phát sinh những biến cố như lỗi hệ thống khi mà số lượng bệnh nhân đăng ký lịch khám bỗng dưng tăng vọt, hay khi bệnh nhân hay người giám sát đặt lịch mà đến ngày lại không xuất hiện thì có thể ảnh hưởng đến người sau vì số thứ tự đã được đánh trên mỗi phiếu đặt lịch.

## Công nghệ áp dụng trong đề tài

### PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu về quan hệ và đối tượng (Object-Relational Database Management System) miễn phí mã nguồn mở (RDBMS) tiên tiến nhất hiện nay, với khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Hiện nay, PostgreSQL đã trở thành một lựa chọn quan trọng và đáng tin cậy cho nhiều cá nhân và doanh nghiệp trong việc quản lý dữ liệu cho các ứng dụng, đặc biệt là hiệu năng cao khi sử dụng trên các dịch vụ của Amazon Web Services (AWS).

Với nhiều tính năng hỗ trợ thì PostgreSQL đã trở thành giải pháp để quản lý dữ liệu cho nhiều cá nhân và doanh nghiệp nhỏ khi mà PostgreSQL là một hệ quản trị mã nguồn mở miễn phí. Đồng thời thì tính bảo mật cao khi mà PostgreSQL hỗ trợ các cơ chế bảo mật tiên tiến như Role-based Access Control (RBAC) hay SSL (Secure Sockets Layer) /TLS (Transport Layer Security). Giúp cho dữ liệu không bị các bên thứ ba truy cập, sử dụng và tấn công trái phép.

### Ngôn ngữ lập trình C#

#### Giới thiệu

C# là một ngôn ngữ lập trình được phát triển bởi Microsoft và ra mắt lần đầu tiên vào năm 2000 như một phần của .NET Framework. C# là một ngôn ngữ cấp cao, được thiết kế để phát triển nhiều loại ứng dụng như ứng dụng Desktop, Mobile, Website, cho đến các dịch vụ đám mây và trò chơi điện tử.

Các đặc điểm chính của C#:

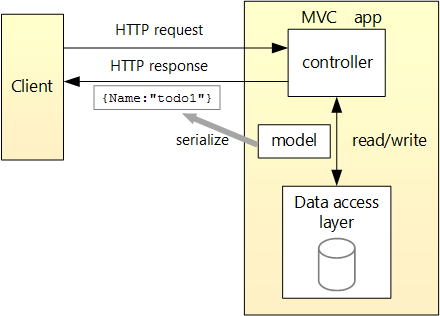
* Hướng đối tượng (Object-Oriented Programming): C# hỗ trợ các nguyên tắc lập trình hướng đối tượng như trừu tượng, kế thừa, đa hình và đóng gói tương tự như ngôn ngữ Java. Điều này giúp mã nguồn dễ dàng bảo trì và tái sử dụng.
* Tính bảo mật và an toàn: C# hỗ trợ tính năng kiểm soát dữ liệu an toàn (Type-Safety) giúp tránh được các lỗi liên quan đến kiểu dữ liệu. Đồng thời nhờ có sự hỗ trợ quản lý tự động (Gabage Collector) trong C# giúp giảm thiểu việc bị rò rỉ bộ nhớ.
* Hỗ trợ lập trình bất đồng bộ (Asynchronous Programming): với hai từ khóa là “async” và “await”, C# cho phép phát triển ứng dụng bất đồng bộ dữ liệu hiệu quả, đặc biệt là trong xử lý các tác vụ Input/Output hoặc kết nối mạng, giúp cải thiện hiệu suất của ứng dụng.
* Hỗ trợ đa nền tảng (Cross-platform): Nhờ vào .NET Core, C# hiện nay có thể được sử dụng để phát triển ứng dụng đa nền tảng chạy trên Windows, Linux và macOS. Điều này giúp C# trở thành ngôn ngữ linh hoạt và thuận tiện cho các nhà phát triển trong việc phát triển ứng dụng, hệ thống.

#### Một số ứng dụng của C#

* Ứng dụng Desktop: Xây dựng những ứng dụng trên hệ điều hành Windows với Windows Forms hoặc WPF (Windows Presentation Foundation).
* Ứng dụng Web: Phát triển các Website và các dịch vụ Web với ASP.NET Core Web API và ASP.NET Core MVC.
* Ứng dụng di động: Xây dựng các ứng dụng di động với Xamarin và Multi-Platform App UI (MAUI).
* Trò chơi điện tử: Phát triển trò chơi điện tử với Unity.
* Dịch vụ đám mây: Xây dựng và triển khai dịch vụ với Microsoft Azure.

#### ASP.NET Core Web API

ASP.NET Core Web API là một framework mạnh mẽ được phát triển bởi Microsoft để xây dựng các RESTful API trên nền tảng ASP.NET Core. Framework được sử dụng để tạo các dịch vụ web mà các ứng dụng khác như ứng dụng web, ứng dụng mobile hay những hệ thống bên thứ ba có thể tương tác thông qua giao thức HTTP. API Cung cấp khả năng gửi và nhận dữ liệu giữa Server và Client, dưới dạng JSON hoặc XML.



Hình .: Mô hình ASP.NET Core Web API

#### Entity Framework Core

Trong ASP.NET, việc tương tác với cơ sở dữ liệu sẽ cần phải thông qua kỹ thuật giúp ánh xạ cơ sở dữ liệu sang các đối tượng trong ngôn ngữ lập trình. Và Object Relational Mapping (ORM) là kỹ thuật cực kỳ phổ biến mà các ngôn ngữ lập trình hiện nay như Java, C# đang sử dụng.

Trong ngôn ngữ lập trình C# hiện nay thì ORM đang được sử dụng phổ biến nhất hiện nay là Entity Framework Core.

Entity Framework Core là một ORM cho phép truy xuất dễ dàng và lưu trữ dữ liệu trong cơ sở dữ liệu thông qua việc ánh xạ cơ sở dữ liệu thành các đối tượng tương ứng trong ngôn ngữ lập trình. là một công nghệ mới của C# nhằm thay thế cho công nghệ cũ trước kia của Microsoft là ADO.NET (ActiveX Data Object), một thư viện cho phép truy vấn dữ liệu sử dụng truy vấn cơ sở dữ liệu trực tiếp mà không phải thông qua tầng trung gian. Tuy vậy ADO.NET khiến cho việc truy vấn dữ liệu trở nên khó sử dụng cho người mới do phải có kiến thức trước đó về truy vấn cơ sở dữ liệu, gây ra trở ngại cho nhiều người, đặc biệt là những người dùng mới chưa thành thạo về truy vấn trong cơ sở dữ liệu.

Với Entity Framework, đây như là một công cụ đã giúp cho nhiều người mới có thể dễ dàng tiếp cận được với việc tương tác với cơ sở dữ liệu, những nhược điểm mà ADO.NET đã khắc phục. Với Entity Framework, các cú pháp rất dễ để sử dụng, các đối tượng từ cơ sở dữ liệu sang ngôn ngữ lập trình C# theo hướng đối tượng (OOP), mặc dù nhược điểm sẽ là tốc độ truy vấn chậm hơn ADO.NET do phải thông qua tầng trung gian để chuyển đổi từ cơ sở dữ liệu sang đối tượng với ORM.

### Ngôn ngữ lập trình JavaScript

#### Giới thiệu

JavaScript là một ngôn ngữ lập trình đa năng và mạnh mẽ, được sử dụng chủ yếu để phát triển Website. Ban đầu, JavaScript được phát triển để làm cho các trang Web có thể có logic và có thể tương tác. Nhưng ngày nay thì với sự ra đời của Node.js, JavaScript đã trở thành một ngôn ngữ không chỉ cho phía front-end (Phía người dùng) mà còn cho cả phía back-end (Phía máy chủ)

#### Một số đặc điểm của ngôn ngữ JavaScript

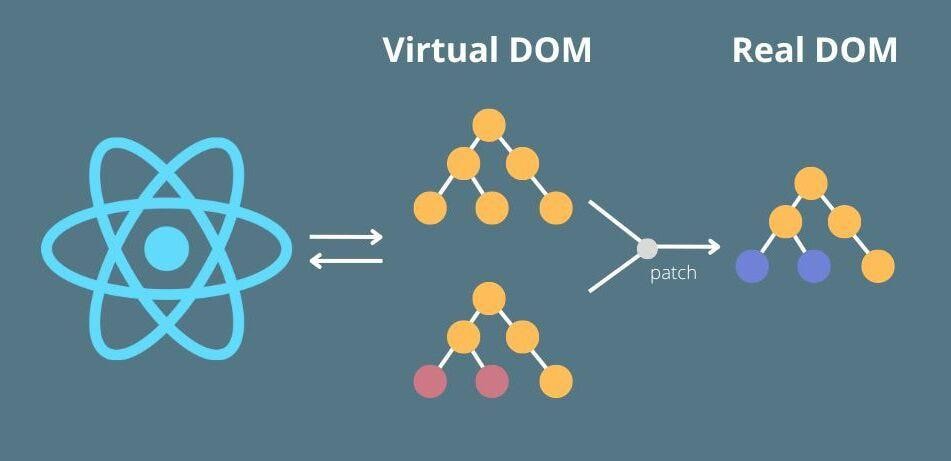
* Ngôn ngữ cho cả hai phía Server và Client: Với client, JavaScript có thể được dùng để tạo ra các trải nghiệm tương tác với người dùng mà không cần tải lại trang. Với sự ra đời của Node.js, JavaScript đã có thể chạy ở cả phía Server, giúp xây dựng các ứng dụng một cách toàn diện mà không phải sử dụng ngôn ngữ lập trình khác cho phía Server.
* Kiểu dữ liệu động: JavaScript sử dụng kiểu dữ liệu động (dynamic typing) cho phép lập trình viên có thể lập trình mà không cần khai báo biến một cách rõ ràng và cụ thể kiểu dữ liệu cho các biến.
* Tương tác với DOM (Document Object Model): Một trong những khả năng mạnh mẽ của JavaScript là khả năng tương tác với DOM của trang web. DOM là đại diện của cấu trúc HTML và JavaScript có thể sử dụng DOM để thay đổi nội dung, kiểu dáng của trang web một cách động và linh hoạt.
* Bất đồng bộ (Asynchronous): Với các tính năng callbacks, promises và async/await giúp cho JavaScript có thể xử lý các tác vụ lâu dài như yêu cầu HTTP, đọc file hay truy xuất dữ liệu từ cơ sở dữ liệu mà không làm chậm web.

#### Một số ứng dụng của JavaScript

* Phát triển giao diện người dùng (front-end) cho những hệ thống website như: ReactJS, Angular, Vue.js,…
* Phát triển giao diện ứng dụng với React Native cho cả Android và iOS.
* Phát triển hệ thống phía máy chủ (back-end) với Node.js, Express.js, Next.js…

#### ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng website. ReactJS tập trung và việc phát triển các thành phần giao diện (Components), giúp tạo ra các UI có thể dễ dàng tái sử dụng và quản lý. React nổi bật với cách tiếp cận hướng thành phần và sử dụng DOM ảo (Virtual DOM) để tối ưu hóa hiệu năng.



Hình .: Mô hình DOM ảo trong React

# NỘI DUNG THỰC HIỆN

## Các bước thực hiện đề tài

Bước 1: Xác định yêu cầu của đề tài

* Tiến hành nghiên cứu và phân tích yêu cầu của hệ thống từ góc nhìn của bệnh nhân, bác sĩ, y tá và người quản trị.
* Xây dựng mô hình Use Case để hiểu rõ hơn về chức năng cần thiết và luồng làm việc giữa người dùng và hệ thống.

Bước 2: Thiết kế hệ thống

* Thiết kế cơ sở dữ liệu: Xác định cấu trúc của dữ liệu cần thiết để lưu trữ thông tin của bệnh nhân, bác sĩ, y tá, hóa đơn, thuốc, toa thuốc, lịch hẹn và báo cáo.
* Thiết kế giao diện người dùng: Tạo giao diện người dùng dễ sử dụng và thân thiện để có thể thực hiện các chức năng như đặt lịch khám, tạo toa thuốc và thanh toán hóa đơn.
* Thiết kế xử lý: Xây dựng các thành phần logic để xử lý kinh doanh và tương tác với cơ sở dữ liệu.

Bước 3: Hiện thực phần mềm

* Sử dụng framework ASP.NET Core Web API để triển khai hệ thống ở phía máy chủ và ReactJS để hiển thị giao diện ở phía người dùng.
* Triển khai ứng dụng web trên một máy chủ.
* Thử nghiệm và kiểm tra ứng dụng để đảm bảo tính ổn định và hoạt động một cách đúng đắn.

Bước 4: Đánh giá, bảo trì và cải tiến

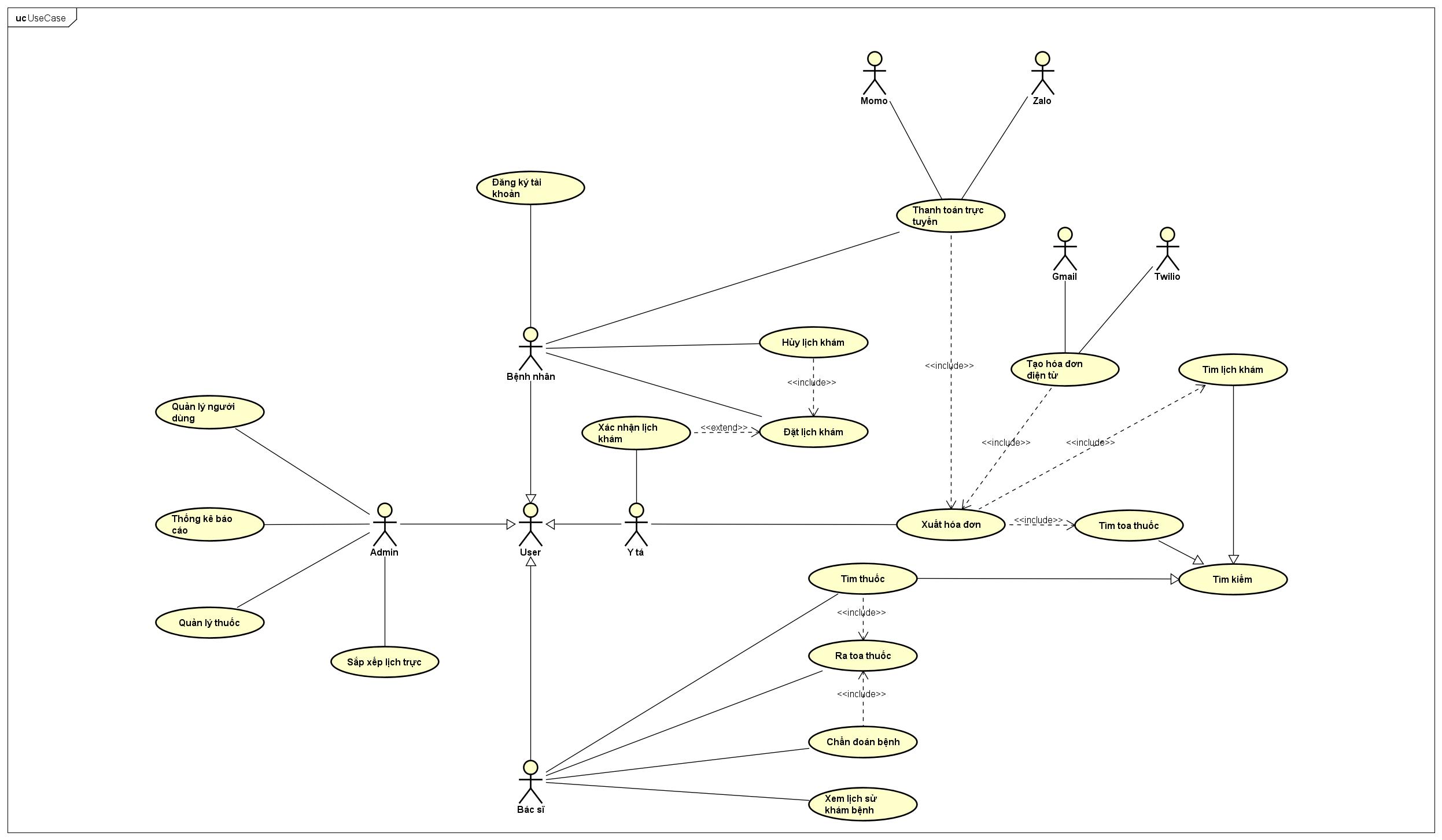
* Thu thập phản hồi từ người dùng và đánh giá hiệu suất của hệ thống.
* Cải thiện những lỗi và bảo trì hệ thống một cách thường xuyên mỗi khi có chỉnh sửa, cập nhật hoặc cải tiến.
* Cải thiện phần mềm dựa trên phản hồi của người dùng để nâng cao hiệu suất và chất lượng của hệ thống.

## Mô tả chức năng nghiệp vụ chính trong đề tài

* Chức năng đăng nhập/đăng ký: Người dùng sẽ đăng ký để có tài khoản sau đó đăng nhập để có thể truy cập vào hệ thống. Với người dùng là bệnh nhân, sẽ được đăng ký thông tin để có thể đăng nhập vào hệ thống, với người dùng là bác sĩ và y tá thì thông tin sẽ được liên hệ đến quản trị viên để có thể tạo tài khoản và sử dụng ứng dụng.
* Chức năng đặt lịch khám: Người dùng với vai trò là bệnh nhân sẽ đăng ký lịch khám cho phòng khám, khi này thì lịch khám sẽ được đưa vào hàng chờ để y tá có thể duyệt lịch khám cho bệnh nhân.
* Chức năng chẩn đoán bệnh: Người dùng với vai trò là bác sĩ sẽ lấy những lịch khám đã được y tá xác nhận để khám cho bệnh nhân, đảm bảo rằng những lịch khám này đã được lấy đúng ngày và đúng bệnh nhân để khám bệnh.
* Chức năng kê toa thuốc: Người dùng với vai trò là bác sĩ sau khi khám bệnh cho bệnh nhân sẽ tiến hành kê toa thuốc cho bệnh nhân để có thể giúp bệnh nhân xác định được toa thuốc để sử dụng.
* Chức năng xuất hóa đơn: Người dùng với vai trò là y tá sau khi bác sĩ đã khám bệnh xong cho bệnh nhân sẽ dựa vào chẩn đoán để có thể xuất hóa đơn cho bệnh nhân, các hóa đơn sẽ không chỉ là hóa đơn khám bệnh mà sẽ còn bao gồm là hóa đơn dịch vụ và hóa đơn thuốc.

## Mô tả chức năng của đề tài

### Lược đồ Use Case



Hình .: Lược đồ Use Case.

|  |  |
| --- | --- |
| Use-case id | UC-1 |
| Tên use-case | Đăng nhập |
| Mô tả | Người dùng nhập thông tin tài khoản vào hệ thống để đăng nhập. |
| Actor | Người dùng. |
| Điều kiện tiên quyết | Người dùng đã đăng ký tài khoản. |
| Luồng sự kiện chính | Người dùng nhập email và mật khẩu vào ô đăng nhập.  Hệ thống xác thực thông tin đăng nhập.  Nếu thông tin đăng nhập là đúng, người dùng sẽ được đưa đến giao diện chính của hệ thống.  Nếu thông tin email hoặc mật khẩu sai, hệ thống sẽ báo lỗi. |
| Kết quả | Người dùng đăng nhập vào hệ thống thành công. |

Bảng .: Bảng đặc tả use case chức năng đăng nhập.

|  |  |
| --- | --- |
| Use-case id | UC-2 |
| Tên use-case | Đăng xuất |
| Mô tả | Người dùng thoát khỏi hệ thống. |
| Actor | Người dùng |
| Điều kiện tiên quyết | Người dùng đã đăng nhập thành công vào hệ thống. |
| Luồng sự kiện chính | 1. Người dùng chọn chức năng đăng xuất. 2. Hệ thống xác nhận, kết thúc phiên làm việc hiện tại. 3. Người dùng được chuyển về trang đăng nhập của hệ thống. |
| Kết quả | Người dùng đăng xuất thành công khỏi hệ thống. |

Bảng .: Bảng đặc tả use case chức năng đăng xuất.

|  |  |
| --- | --- |
| Use-case id | UC-3 |
| Tên use-case | Quản lý người dùng |
| Mô tả | Quản trị viên có thể xem, thêm, sửa, xóa thông tin của người dùng. |
| Actor | Quản trị viên (Admin) |
| Điều kiện tiên quyết | Quản trị viên đã đăng nhập vào hệ thống. |
| Luồng sự kiện chính | 1. Quản trị viên chọn chức năng quản lý thông tin người dùng. 2. Hệ thống hiển thị danh sách người dùng theo vai trò. 3. Quản trị viên có thể:  * Thêm người dùng mới. * Chỉnh sửa thông tin người dùng. * Xem chi tiết thông tin của người dùng. * Xóa thông tin người dùng. |
| Kết quả | Thông tin người dùng được quản trị viên quản lý một cách hiệu quả và chặt chẽ. |

Bảng .: Bảng đặc tả use case chức năng quản lý người dùng.

|  |  |
| --- | --- |
| Use-case id | UC-4 |
| Tên use-case | Kê toa thuốc |
| Mô tả | Bác sĩ kê toa thuốc cho bệnh nhân. |
| Actor | Bác sĩ |
| Điều kiện tiên quyết | Bác sĩ đã đăng nhập. |
| Luồng sự kiện chính | 1. Bác sĩ chọn chức năng kê toa thuốc. 2. Bác sĩ chọn lịch khám cần kê toa thuốc. 3. Bác sĩ tìm kiếm và chọn thuốc (kết hợp với tìm kiếm thuốc). 4. Bác sĩ xác nhận và lưu toa thuốc. |
| Kết quả | Toa thuốc được tạo và lưu trong hệ thống. |

Bảng .: Bảng đặc tả use case chức năng kê toa thuốc.

|  |  |
| --- | --- |
| Use-case id | UC-5 |
| Tên use-case | Tìm kiếm thuốc |
| Mô tả | Bác sĩ tìm kiếm thông tin về các loại thuốc có sẵn trong hệ thống. |
| Actor | Bác sĩ |
| Điều kiện tiên quyết | Bác sĩ đã đăng nhập. |
| Luồng sự kiện chính | 1. Bác sĩ chọn chức năng tìm kiếm thuốc. 2. Hệ thống hiển thị giao diện tìm kiếm thuốc. 3. Bác sĩ nhập thông tin tìm kiếm thuốc theo tên thuốc. 4. Hệ thống trả về kết quả tìm kiếm. |
| Kết quả | Bác sĩ tìm thấy thông tin thuốc cần thiết. |

Bảng .: Bảng đặc tả use case chức năng tìm kiếm thuốc.

|  |  |
| --- | --- |
| Use-case id | UC-6 |
| Tên use-case | Tìm kiếm lịch khám |
| Mô tả | Y tá tìm kiếm lịch khám của bệnh nhân. |
| Actor | Y tá |
| Điều kiện tiên quyết | Y tá đã đăng nhập vào hệ thống. |
| Luồng sự kiện chính | 1. Y tá chọn chức năng tìm kiếm lịch khám. 2. Hệ thống hiển thị giao diện lịch khám. 3. Y tá nhập thông tin bệnh nhân hoặc ngày khám. 4. Hệ thống trả về thông tin lịch khám phù hợp |
| Kết quả | Y tá tìm thấy thông tin lịch khám cần thiết. |

Bảng .: Bảng đặc tả use case chức năng tìm kiếm lịch khám.

|  |  |
| --- | --- |
| Use-case id | UC-7 |
| Tên use-case | Tạo lịch khám |
| Mô tả | Bệnh nhân đăng ký lịch khám cho phòng khám tư. |
| Actor | Bệnh nhân |
| Điều kiện tiên quyết | Bệnh nhân đã có tài khoản và đăng nhập thành công. |
| Luồng sự kiện chính | 1. Bệnh nhân chọn chức năng đăng ký lịch khám. 2. Hệ thống hiển thị giao diện đăng ký lịch khám. 3. Bệnh nhân nhập ngày khám và thông tin khám bệnh. 4. Hệ thống kiểm tra thông tin và lưu lịch khám. |
| Kết quả | Lịch khám được tạo thành công và bệnh nhân có thể chờ xác nhận hoặc hủy lịch khám. |

Bảng .: Bảng đặc tả use case chức năng tạo lịch khám.

|  |  |
| --- | --- |
| Use-case id | UC-8 |
| Tên use-case | Xác nhận lịch khám |
| Mô tả | Y tá xác nhận lịch khám sau khi bệnh nhân đã đăng ký khám bệnh. |
| Actor | Y tá |
| Điều kiện tiên quyết | Y tá đã đăng nhập. |
| Luồng sự kiện chính | 1. Y tá chọn chức năng tìm kiếm lịch khám đang chờ 2. Y tá nhấn chọn xác nhận lịch khám. 3. Hệ thống ghi nhận thông tin xác nhận của y tá. 4. Hệ thống thông báo cho bệnh nhân để có thể đến khám vào ngày đã đăng ký. |
| Kết quả | Lịch khám bệnh nhân đã đăng ký được y tá xác nhận để có thể đến khám bệnh. |

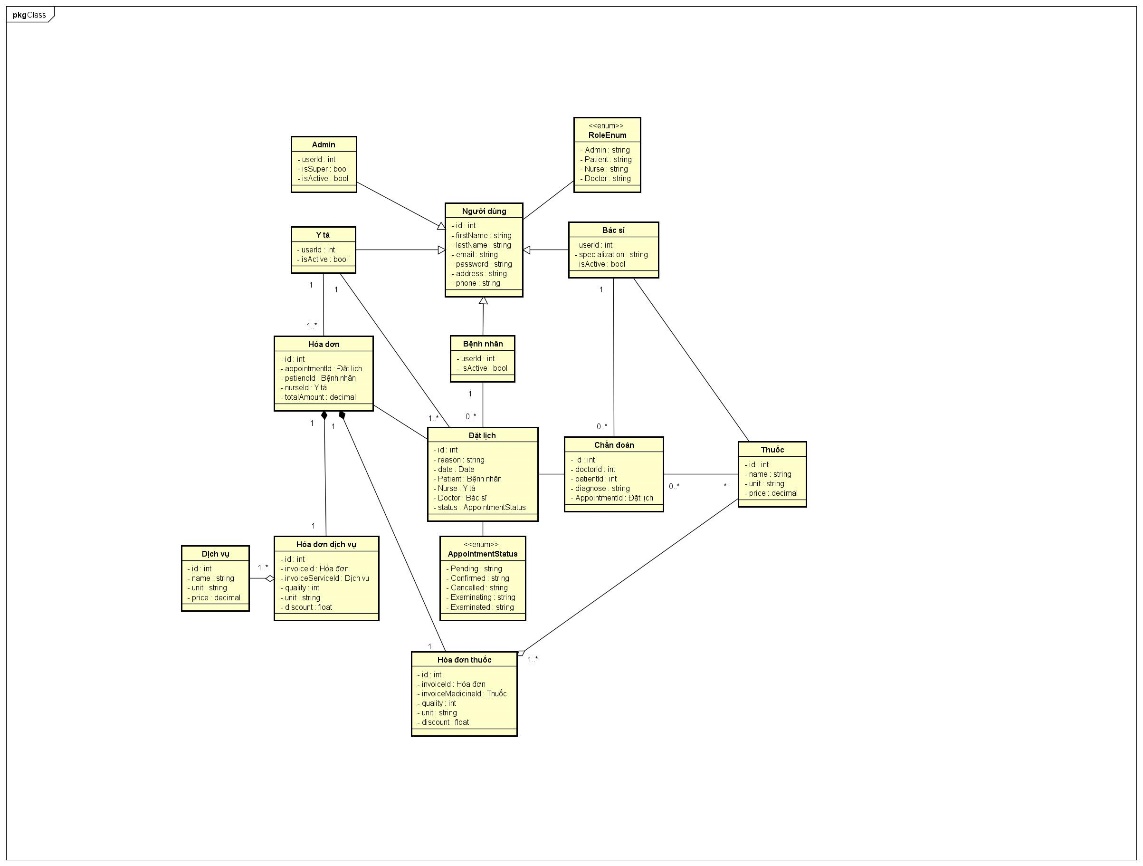
Bảng .: Bảng đặc tả use case chức năng xác nhận lịch khám.

|  |  |
| --- | --- |
| Use-case id | UC-9 |
| Tên use-case | Xuất hóa đơn |
| Mô tả | Y tá xuất hóa đơn cho bệnh nhân sau khi hoàn tất dịch vụ khám chữa bệnh. |
| Actor | Y tá |
| Điều kiện tiên quyết | Y tá đã đăng nhập và bệnh nhân đã được bác sĩ khám bệnh. |
| Luồng sự kiện chính | 1. Y tá chọn chức năng xuất hóa đơn. 2. Hệ thống yêu cầu nhập thông tin bệnh nhân và dịch vụ đã đăng ký. 3. Hệ thống tính toán và hiển thị hóa đơn (Bao gồm việc tìm kiếm toa thuốc). 4. Y tá xác nhận và in ra hóa đơn. |
| Kết quả | Hóa đơn đã được xuất và in ra thành công. |

Bảng .: Bảng đặc tả use case chức năng xuất hóa đơn.

## Mô tả dữ liệu

### Sơ đồ lớp

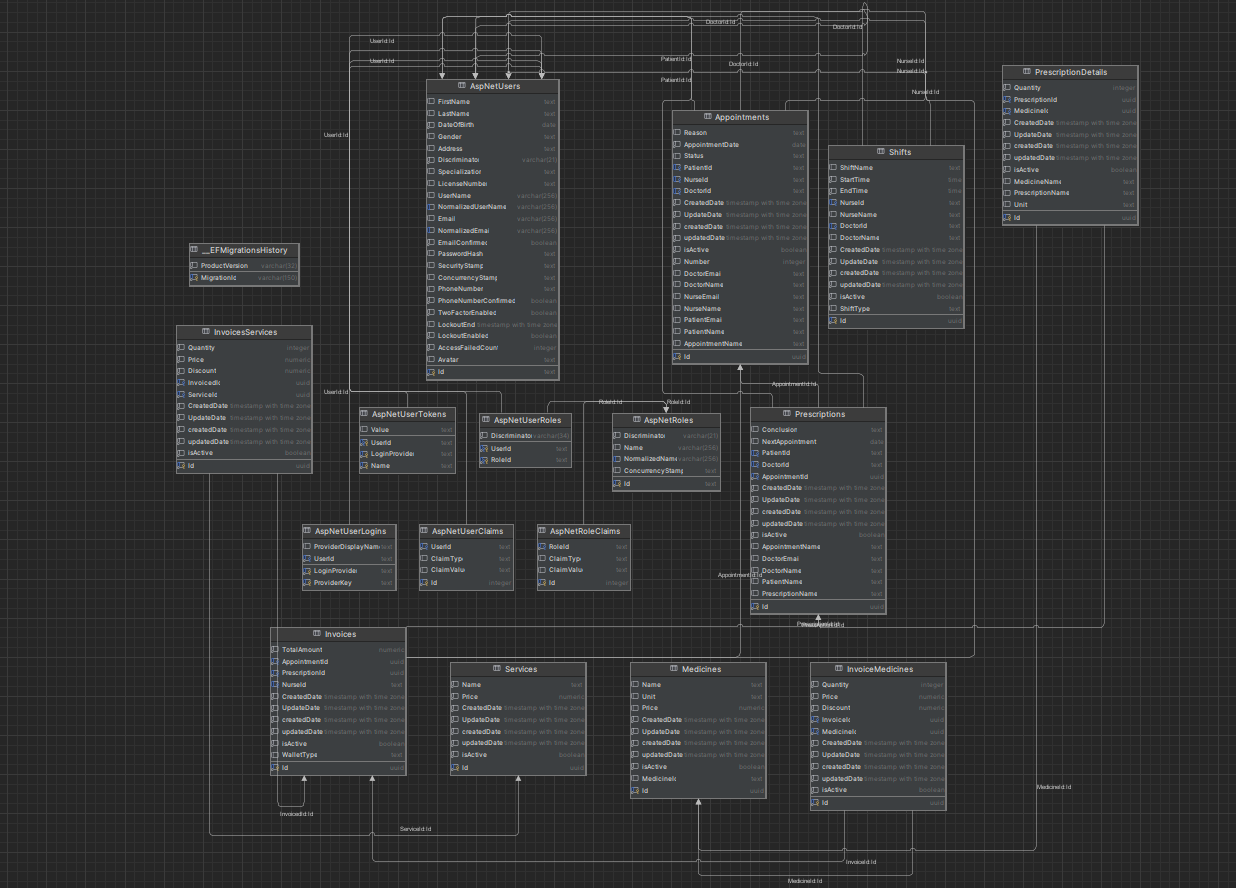


Hình .: Sơ đồ lớp

Giải thích về sơ đồ lớp:

* Bảng Người Dùng là bảng chung cho các bảng Admin, Bệnh Nhân, Y Tá và Bác Sĩ kế thừa, bảng Admin sẽ có thêm thuộc tính isSuper để trở thành Người Dùng có quyền cao nhất, bảng Bác Sĩ sẽ có thêm thuộc tính là Specialization để hiển thị thông tin về chứng chỉ.
* Bảng Đặt Lịch được kết nối với Bệnh Nhân quan hệ 1 – \* cho phép bệnh nhân có thể có từ 0 đến nhiều lịch khám và 1 lịch khám chỉ thuộc về 1 bệnh nhân. Y tá có mối quan hệ 1 – \* với đặt lịch để chỉ khi có lịch thì y tá mới xác nhận được lịch khám.
* Bảng chẩn đoán có mối quan hệ 1 – 1 với bảng đặt lịch để chỉ có 1 chẩn đoán kết nối với 1 đặt lịch. Bảng thuốc có quan hệ \* – 1 với bảng chẩn đoán để có thể có nhiều thuốc trong 1 chẩn đoán.
* Bảng Hóa Đơn có mối quan hệ 1 – \* với bảng Y tá để Y tá có thể tạo nhiều hóa đơn và một hóa đơn chỉ được tạo bởi 1 y tá.
* Các bảng như dịch vụ, bảng thuốc có mối quan hệ aggregation với bảng hóa đơn để khi bảng hóa đơn có bị xóa thì thuốc và dịch vụ sẽ vẫn tồn tại độc lập.

### Lược đồ cơ sở dữ liệu quan hệ



Hình .: Lược đồ sơ sở dữ liệu quan hệ.

Giải thích về lược đồ cơ sở dữ liệu quan hệ:

* Các bảng về AspNet là các bảng liên quan trực tiếp đến người dùng trong hệ thống bao gồm các thông tin cơ bản, thông tin về vai trò người dùng, thông tin về các token đã được sử dụng để đăng nhập.
* Bảng Appointment có mối quan hệ n – 1 với bảng User, tương ứng với một người dùng, cụ thể là người dùng có vai trò bệnh nhân, một bệnh nhân có nhiều lịch khám và một lịch khám chỉ thuộc về một bệnh nhân. Với vai trò là bác sĩ và y tá thì cũng sẽ có quan hệ là 1 – n tương ứng với một y tá có thể xác nhận nhiều lịch khám, một bác sĩ có thể nhận nhiều lịch khám và một lịch khám chỉ thuộc về một y tá đã xác nhận và một bác sĩ đã nhận lịch.
* Bảng Prescription có mối quan hệ n – 1 với bảng User, cụ thể là User có vai trò là bác sĩ, một bác sĩ có thể chẩn đoán cho nhiều bệnh nhân đã xác nhận lịch và một chẩn đoán chỉ thuộc về một bệnh nhân.
* Bảng PrescriptionDetails có mối quan hệ 1 – 1 với bảng Prescription, ở bảng này sẽ chứa thông tin chi tiết của một Prescription, bao gồm cả thông tin về thuốc được kê toa cho bệnh nhân.
* Bảng Medicines lưu thông tin về thuốc trong hệ thống. Bảng sẽ có mối quan hệ 1 – n với bảng PrescriptionDetails, một PrescriptionDetails sẽ có thể thêm được nhiều Medicines, một Medicine chỉ thuộc về một PrescriptionDetails.
* Với bảng Invoice, bảng này sẽ có mối quan hệ n – 1 với bảng User, với hai vai trò là y tá và bệnh nhân, một hóa đơn sẽ được tạo bởi một y tá và thanh toán bởi một bệnh nhân; có mối quan hệ 1 – 1 với bảng Prescription và Appointment, một Invoice sẽ chỉ thuộc về một Prescription và Appointment và ngược lại.
* Hai bảng là InvoiceService và InvoiceMedicine sẽ có mối quan hệ 1 – 1 với bảng Invoice, để mở rộng cho bảng Invoice, các bảng này sẽ chứa thông tin chi tiết về giá thuốc và giá dịch vụ trong hệ thống, giúp cho bảng Invoice không bị chứa quá nhiều thuộc tính. Các bảng này sẽ có mối quan hệ 1 – n với bảng Medicine và Service, cho phép các Invoice này có thể chứa nhiều Medicines và Service.
* Với bảng Shift thì sẽ là bảng dành cho User có vai trò là bác sĩ và y tá, có mối quan hệ là n – 1 tương đương với một bác sĩ hay một y tá có thể có nhiều lịch khám và một lịch khám chỉ thuộc về một bác sĩ.
* Trong ASP.NET Core Web API, đặc biệt là với Entity Framework sử dụng mô hình Code First, thì bảng EFMigration cho phép lưu các thay đổi từ trên code xuống database.

### Chi tiết các thuộc tính trong từng bảng trong cơ sở dữ liệu

* Các bảng liên quan đến AspNet bao gồm những thuộc tính chính và cần thiết được nhóm lại với nhau:

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| FirstName | String(50) | Tên |
| LastName | String(50) | Họ |
| Email | String(50) | Email |
| PhoneNumber | String(10) | Số điện thoại |
| Password | String(20) | Mật khẩu |
| DateOfBirth | Date | Ngày sinh |
| Gender | String(10) | Giới tính |
| Address | Text | Địa chỉ |
| Avatar | String | Ảnh đại diện |

* Bảng Appointments

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| AppointmentName | String(50) | Tên Appointment |
| Reason | Text | Lý do khám bệnh |
| Status | String(50) | Trạng thái |
| PatientId | Uuid | Id Bệnh nhân |
| PatientName | String(50) | Tên bệnh nhân |
| NurseId | Uuid | Id Y tá |
| NurseName | String(50) | Tên y tá |
| DoctorId | Uuid | Id Bác sĩ |
| DoctorName | String(50) | Tên bác sĩ |
| Number | Int | Số thứ tự |

* Bảng Prescriptions

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| PrescriptionName | String(50) | Tên Prescription |
| Conclusion | Text | Kết luận triệu chứng |
| NextAppointment | Date | Lịch hẹn tiếp theo |
| PatientId | Uuid | Id Bệnh nhân |
| PatientName | String(50) | Tên Bệnh nhân |
| DoctorId | Uuid | Id Bác sĩ |
| DoctorName | String(50) | Tên bác sĩ |
| AppointmentId | Uuid | Id Appointment |
| AppointmentName | String(50) | Tên Appointment |

* Bảng PrescriptionDetails

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| PrescriptionId | Uuid | Id Prescription |
| PrescriptionName | String(50) | Tên Prescription |
| MedicineId | Uuid | Id Medicine |
| MedicineName | String(50) | Tên Medicine |
| Unit | String(50) | Đơn vị tính |
| Quantity | Int | Số lượng |

* Bảng Medicine

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| Name | String(50) | Tên thuốc |
| Unit | String(50) | Đơn vị tính |
| Price | Decimal | Giá tiền |
| MedicineId | String(50) | Id riêng của thuốc |

* Bảng Service

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| Name | String(50) | Tên dịch vụ |
| Price | Decimail | Giá tiền |

* Bảng Invoice

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| InvoiceName | String(50) | Tên Invoice |
| TotalAmount | Decimal | Tổng giá tiền |
| AppointmentId | Uuid | Id Appointment |
| PrescriptionId | Uuid | Id Prescription |
| NurseId | Uuid | Id Y tá |
| PatientId | Uuid | Id Bệnh nhân |
| WalletType | String | Loại ví thanh toán |

* Bảng InvoiceService

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| InvoiceId | Uuid | Id Invoice |
| ServiceId | Uuid | Id Service |
| Quantity | Int | Số lượng |
| Price | Decimal | Đơn giá |
| Discount | Float | Giảm giá |

* Bảng InvoiceMedicine

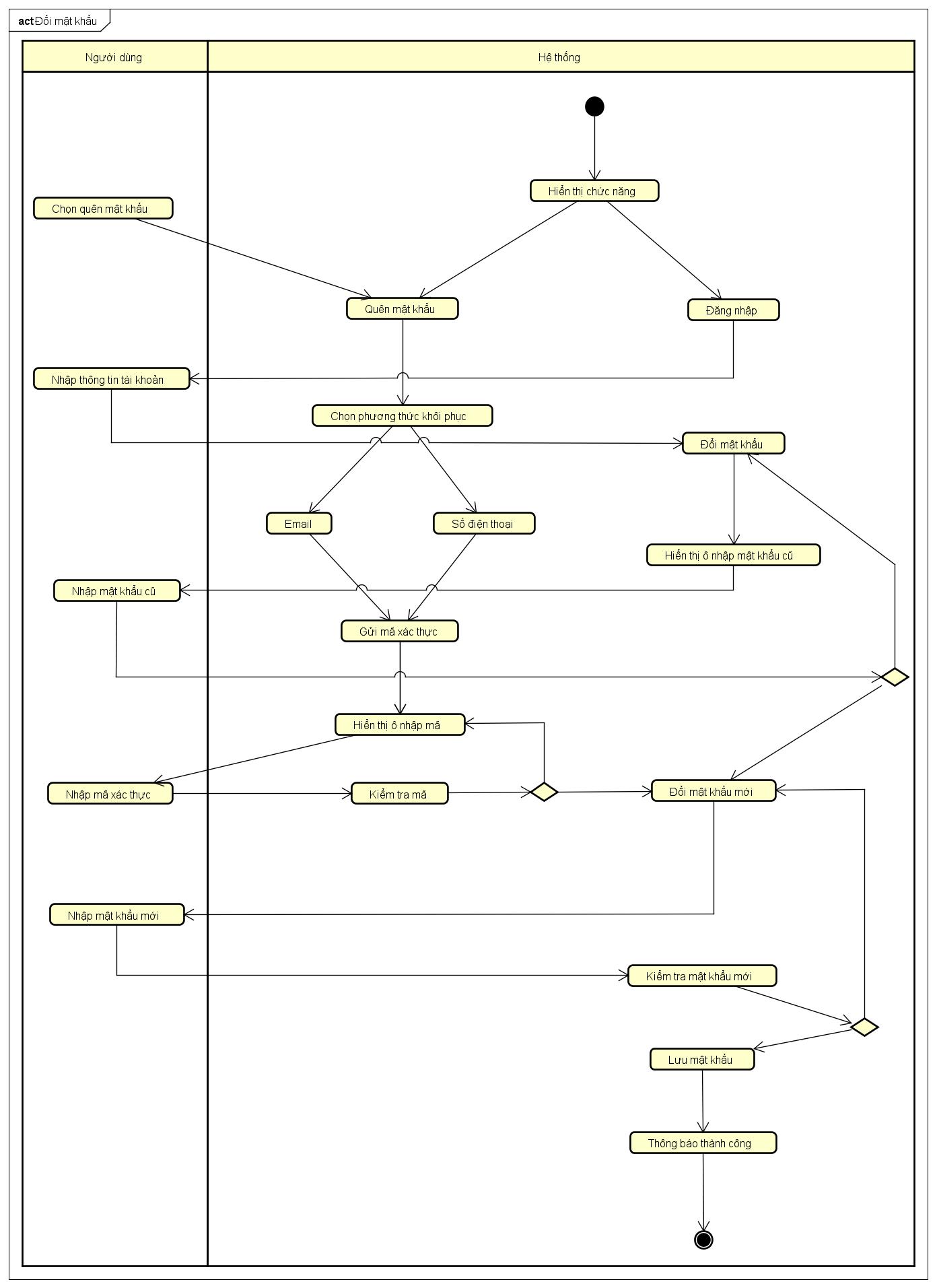
|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| InvoiceId | Uuid | Id Invoice |
| MedicineId | Uuid | Id Medicine |
| Quantity | Int | Số lượng |
| Price | Decimal | Giá tiền |
| Discount | Float | Giảm giá |

* Bảng Shift

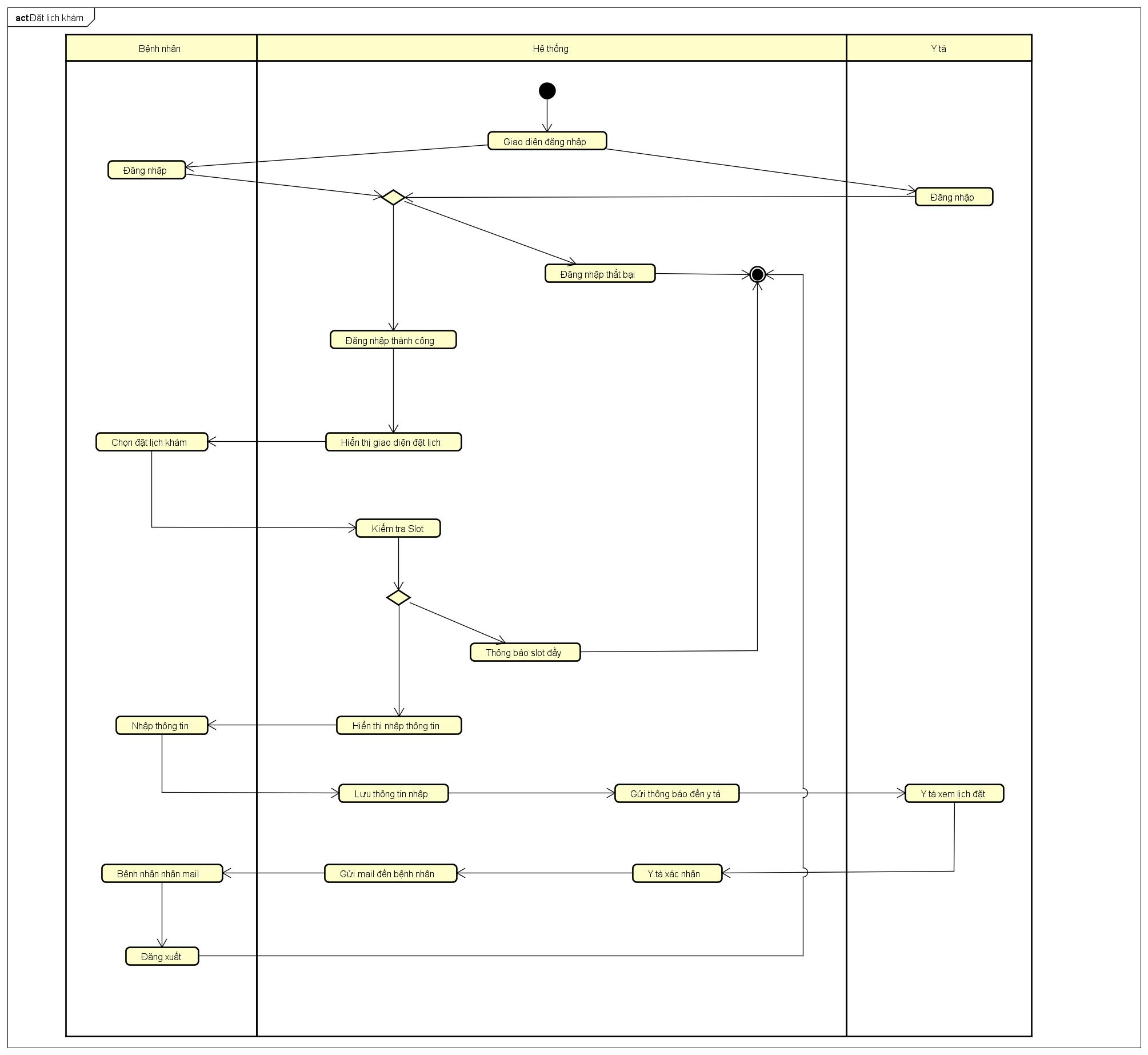
|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Uuid | Id |
| ShiftName | String(50) | Tên ca |
| StartTime | Time | Giờ bắt đầu |
| EndTime | Time | Giờ kết thúc |
| ShiftType | String(50) | Loại ca làm |
| NurseId | Uuid | Id Y tá |
| NurseName | String(50) | Tên Y tá |
| DoctorId | Uuid | Id Bác sĩ |
| DoctorName | String(50) | Tên Bác sĩ |

## Sơ đồ hoạt động

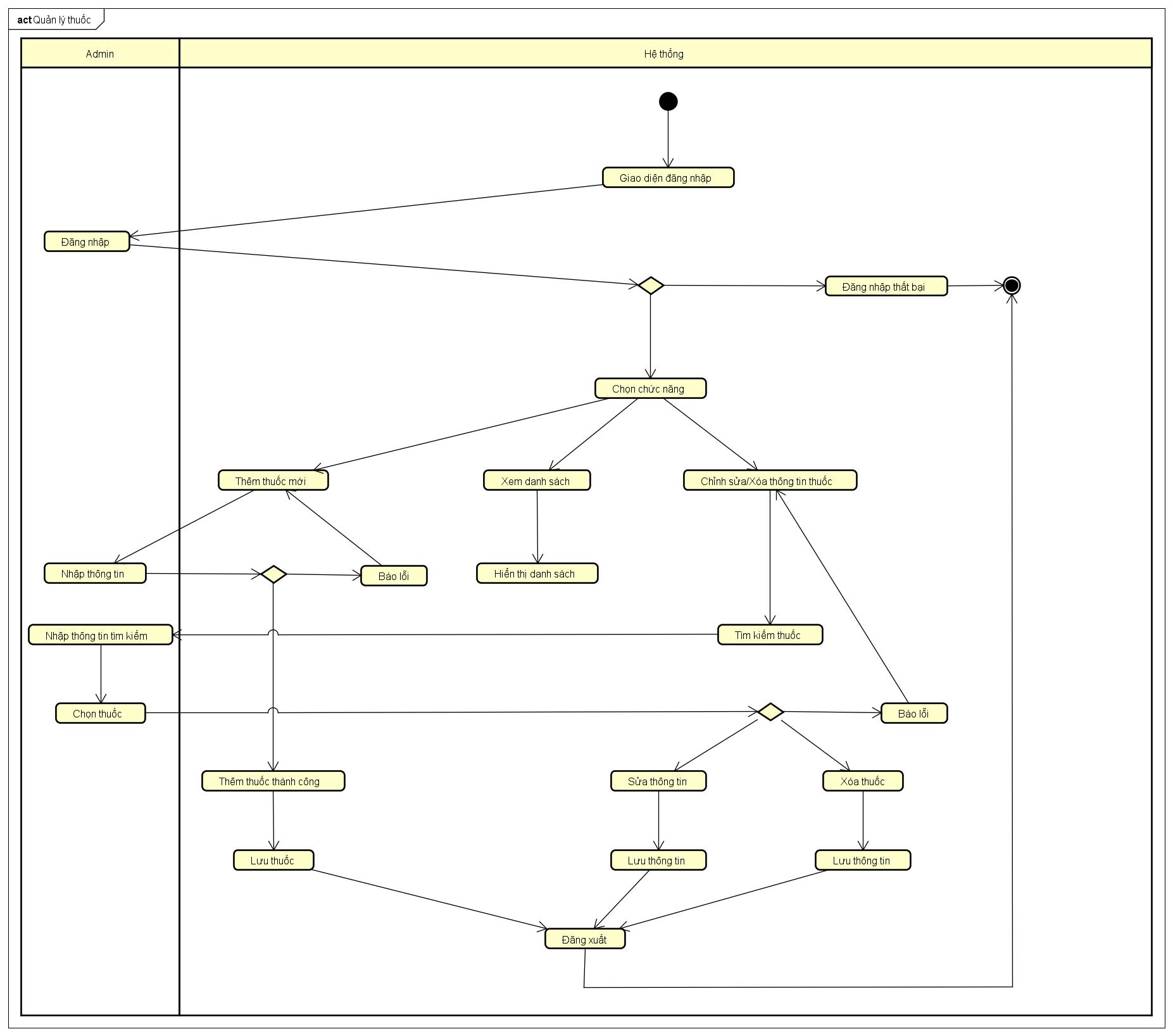
Các chức năng được sử dụng trong hệ thống bao gồm:



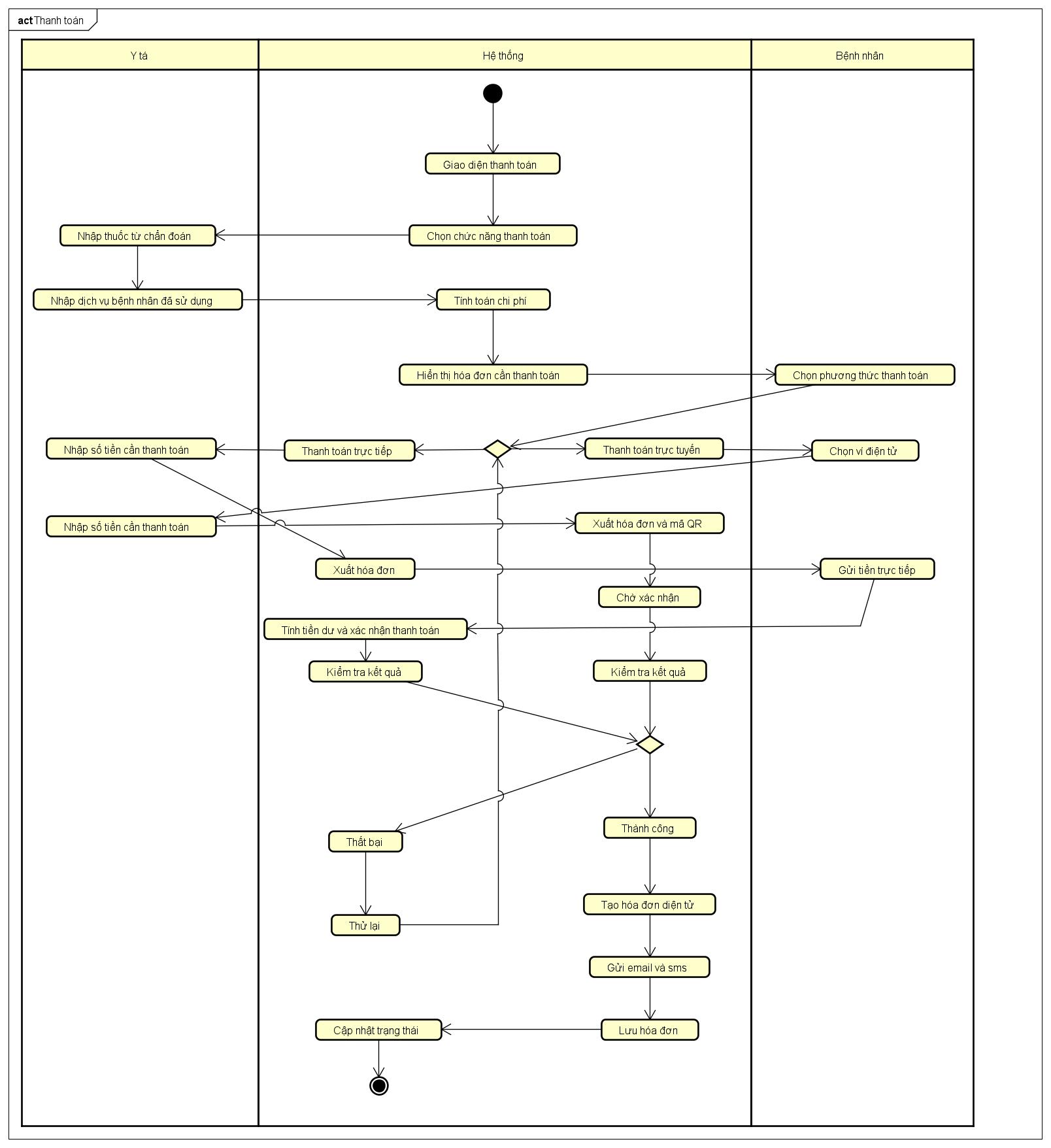
Hình .: Sơ đồ hoạt động của chức năng đổi mật khẩu.



Hình .: Sơ đồ hoạt động của chức năng đặt lịch khám.

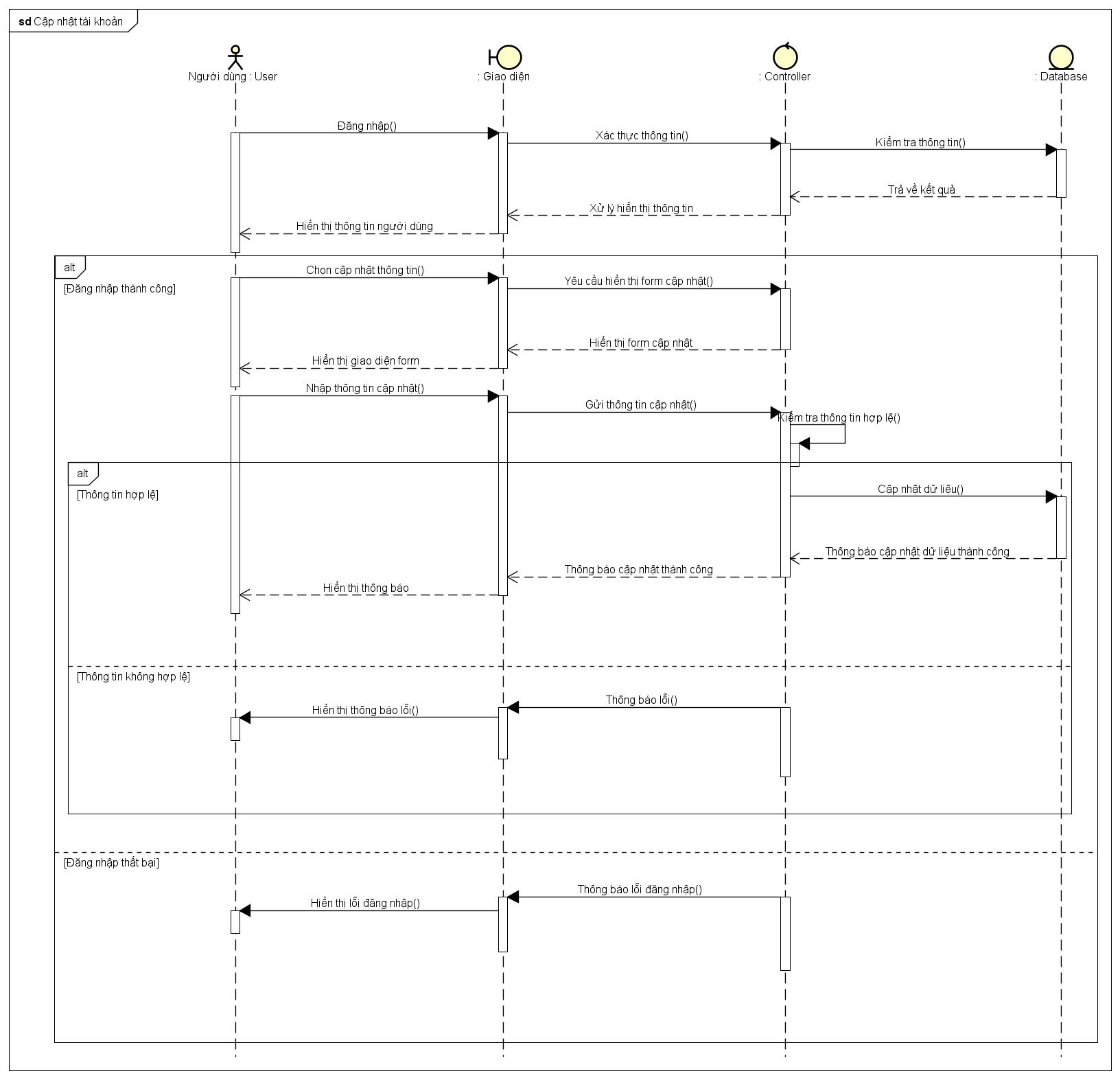


Hình .: Sơ đồ hoạt động của chức năng quản lý thuốc.

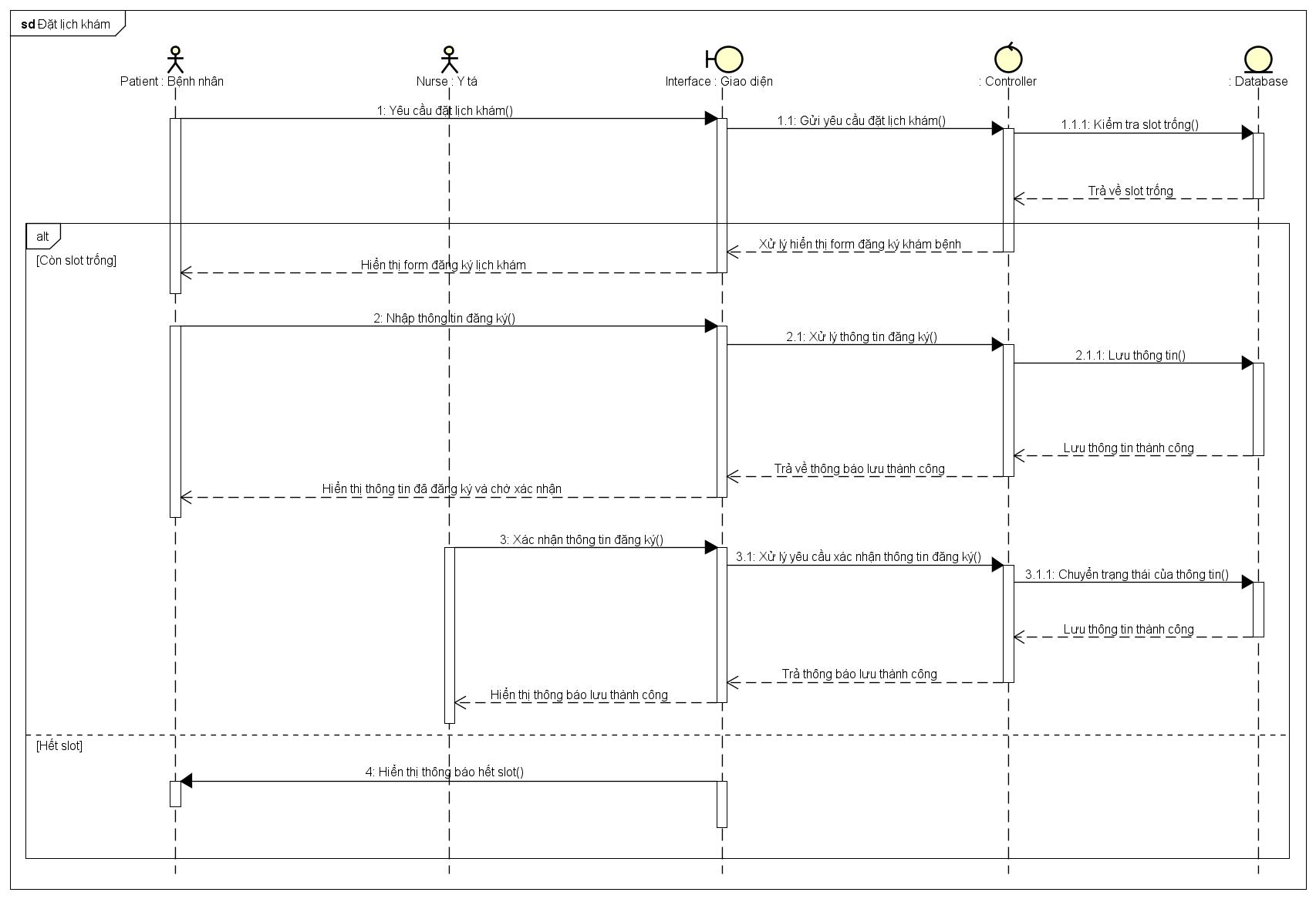


Hình .: Sơ đồ hoạt động của chức năng thanh toán.

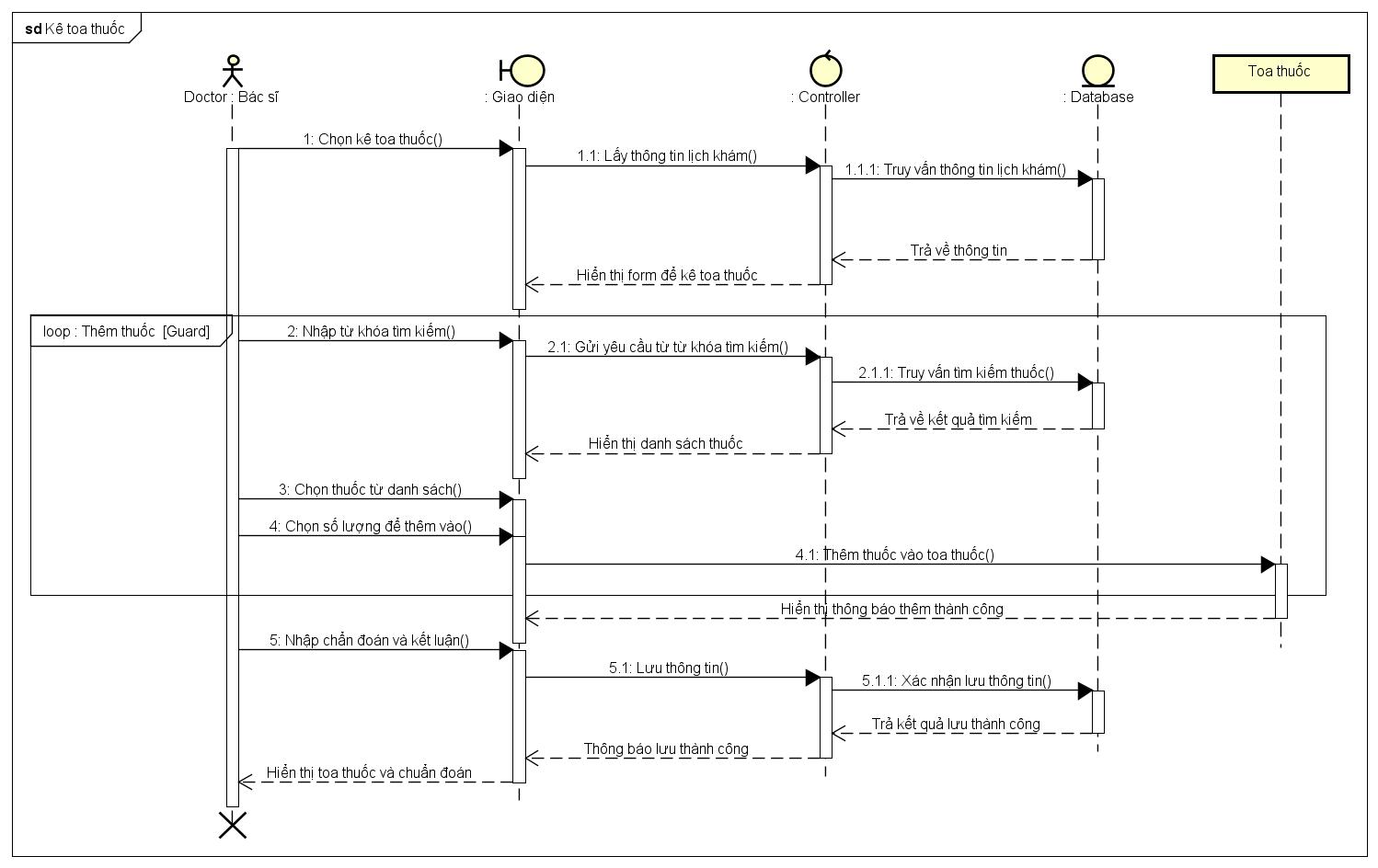
## Sơ đồ tuần tự



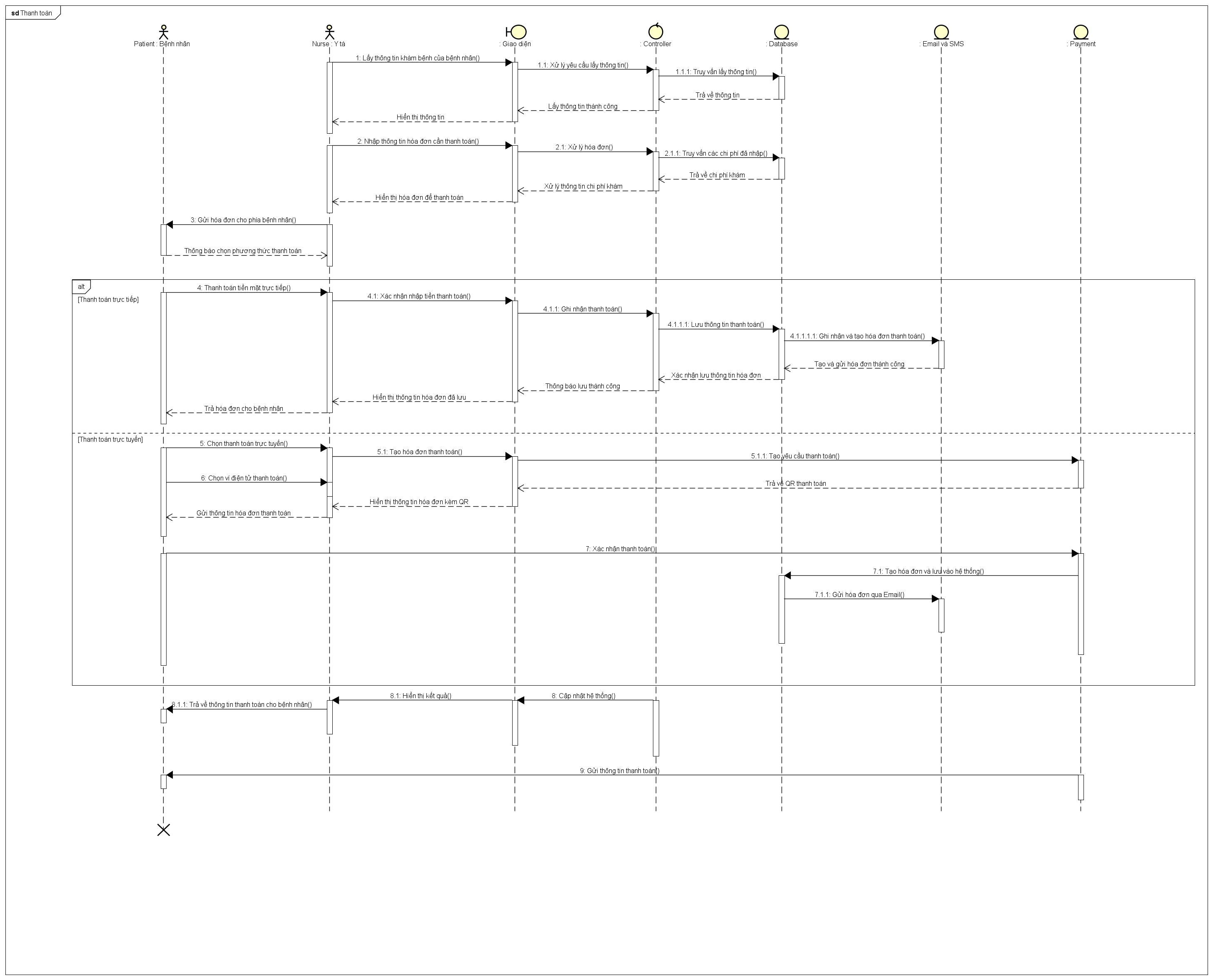
Hình .: Sơ đồ tuần tự của chức năng cập nhật tài khoản



Hình .: Sơ đồ tuần tự chức năng đặt lịch khám.



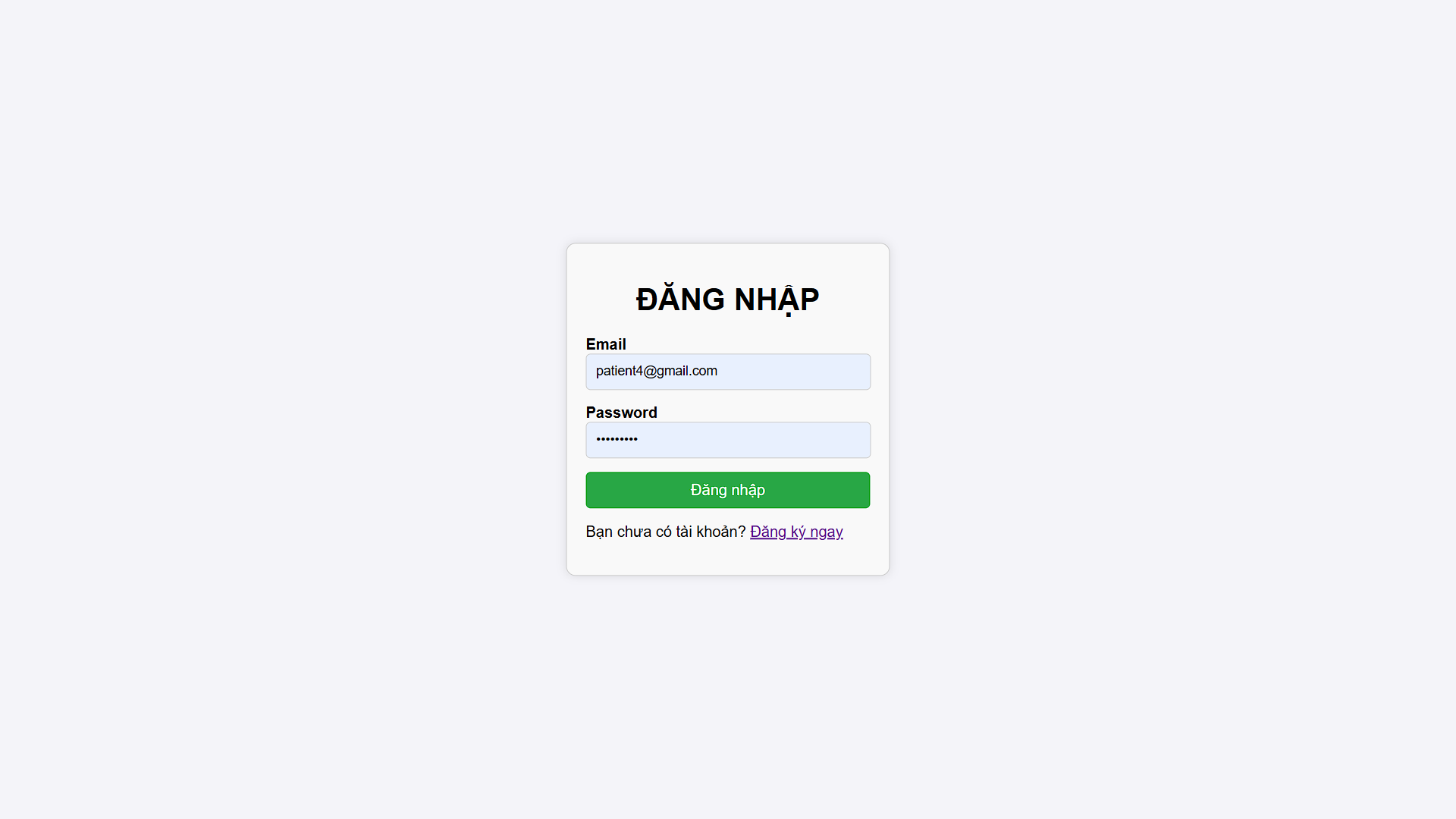
Hình .: Sơ đồ tuần tự chức năng kê toa thuốc.



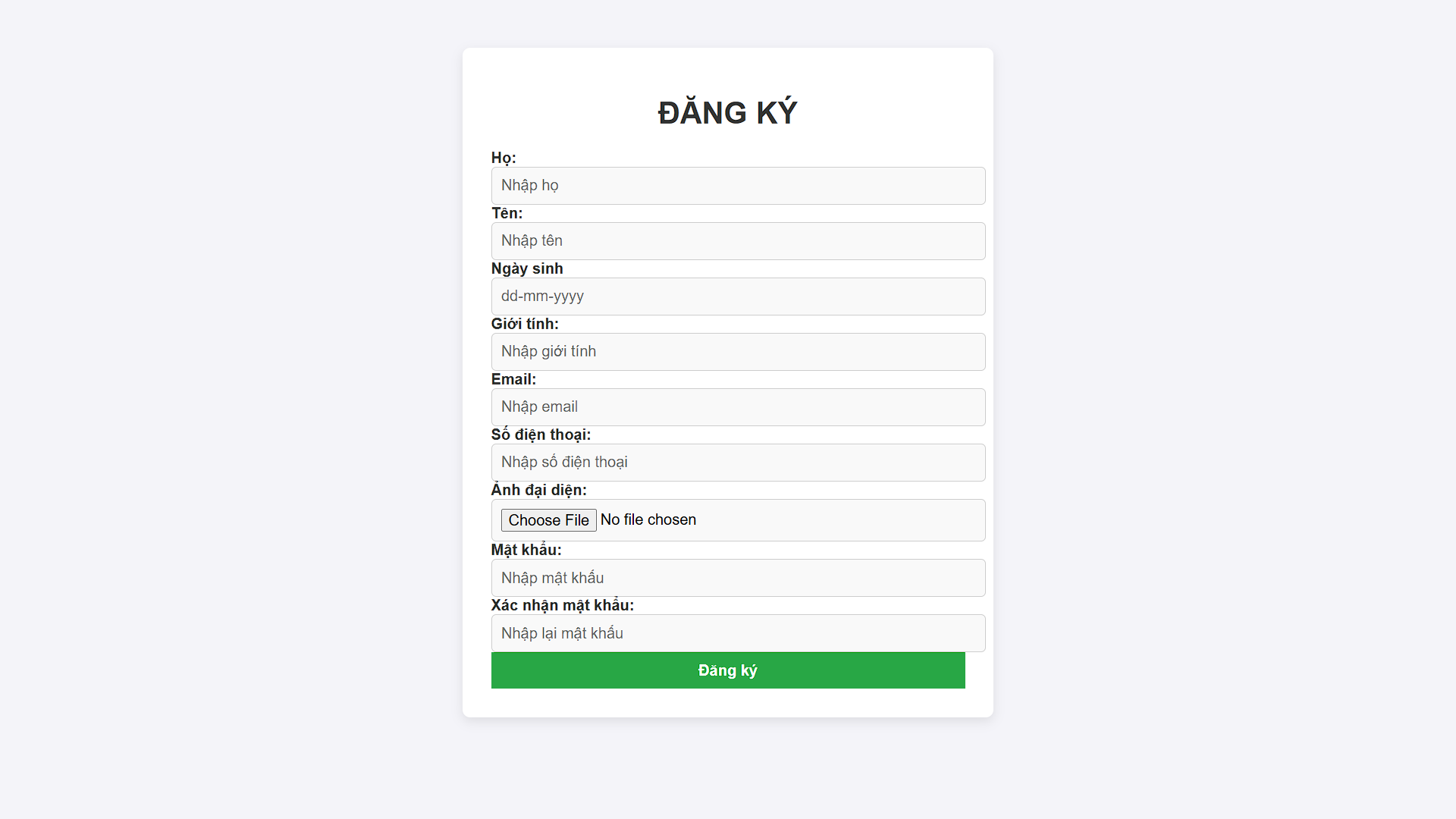
Hình .: Sơ đồ tuần tự cho chức năng thanh toán.

## Các giao diện theo vai trò

### Giao diện đăng nhập và đăng ký

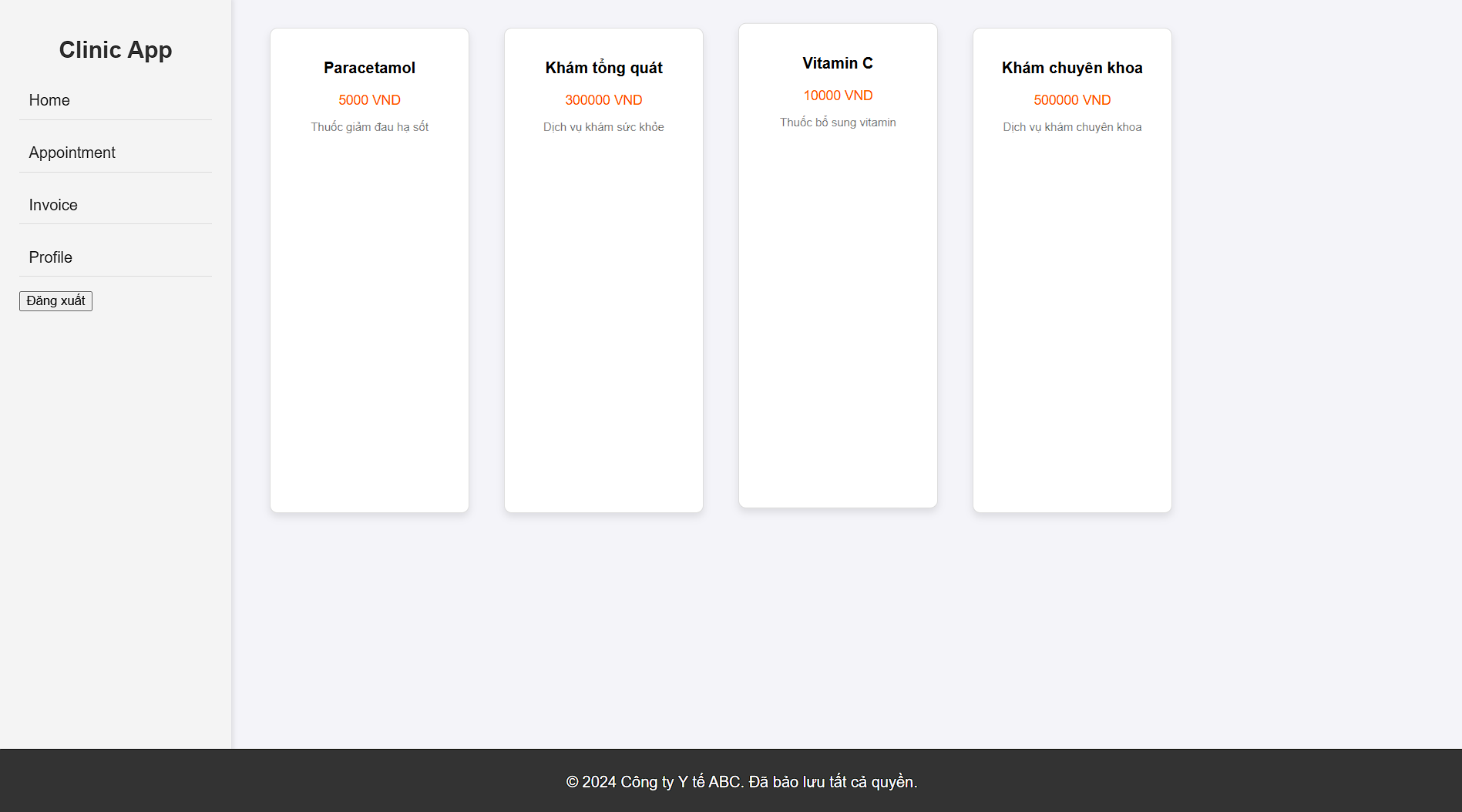


Hình .: Giao diện đăng nhập người dùng

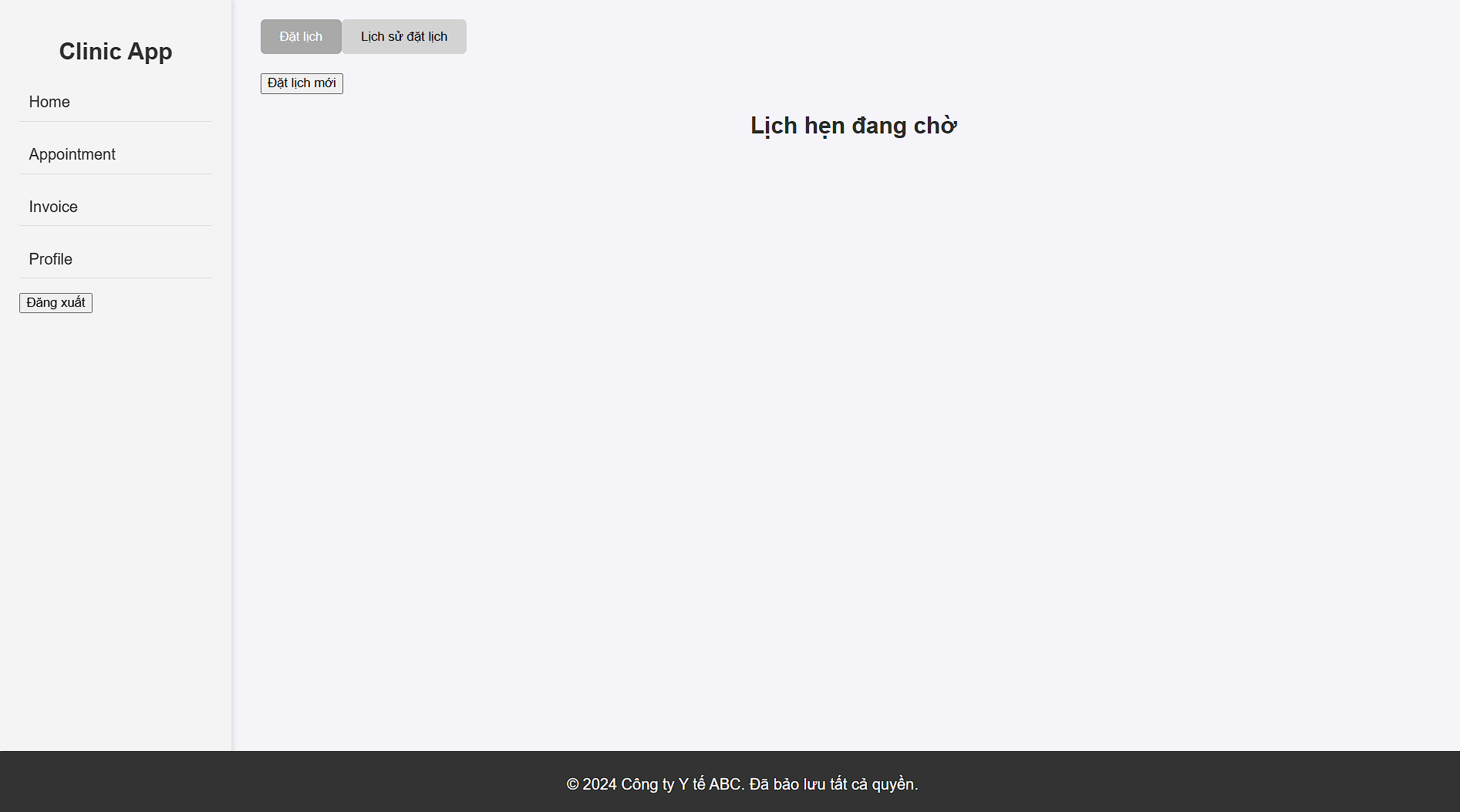


Hình .: Giao diện đăng ký người dùng với vai trò là bệnh nhân

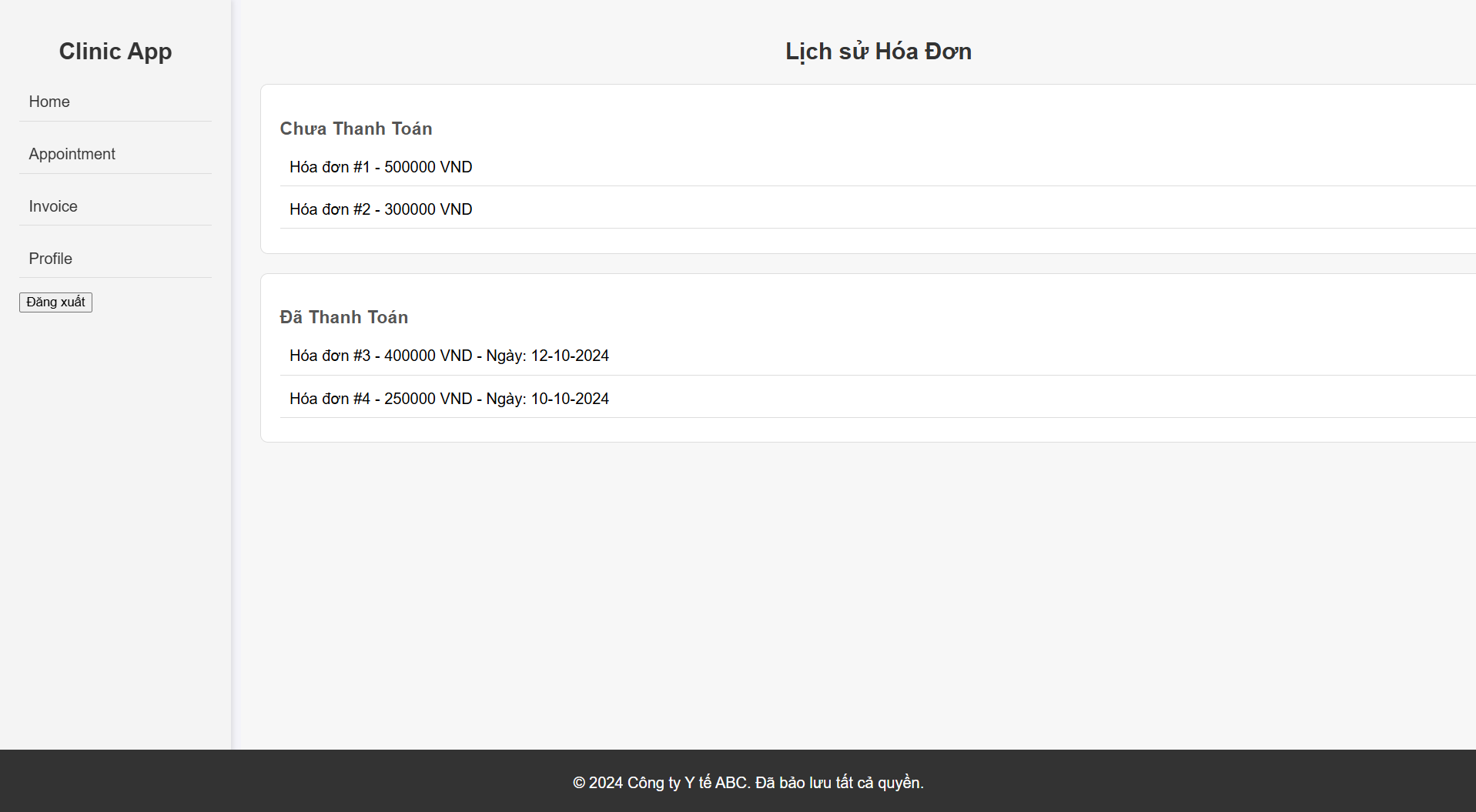
### Giao diện cho Patient



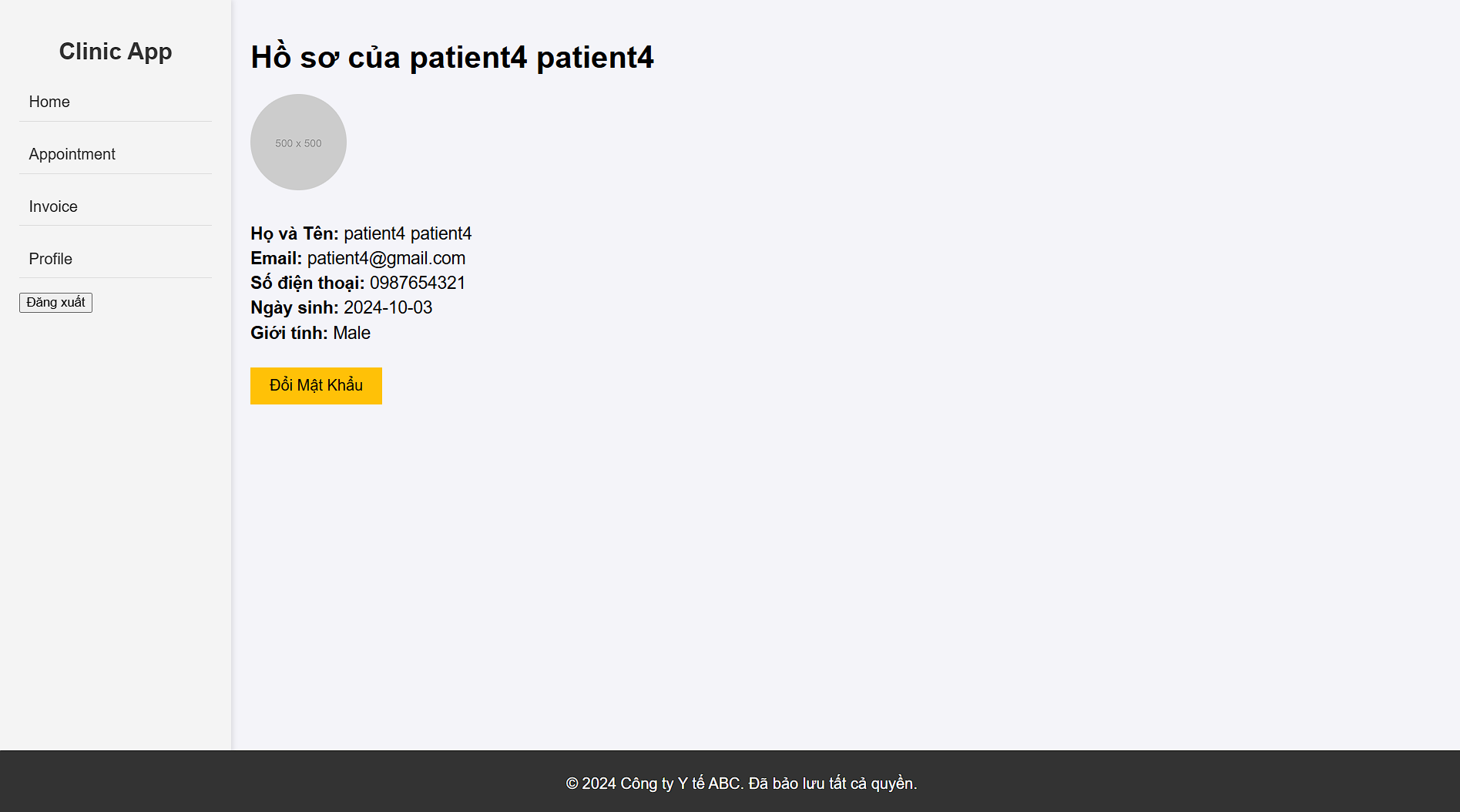
Hình .: Giao diện màn hình chính cho Patient



Hình .: Giao diện đặt lịch cho Patient

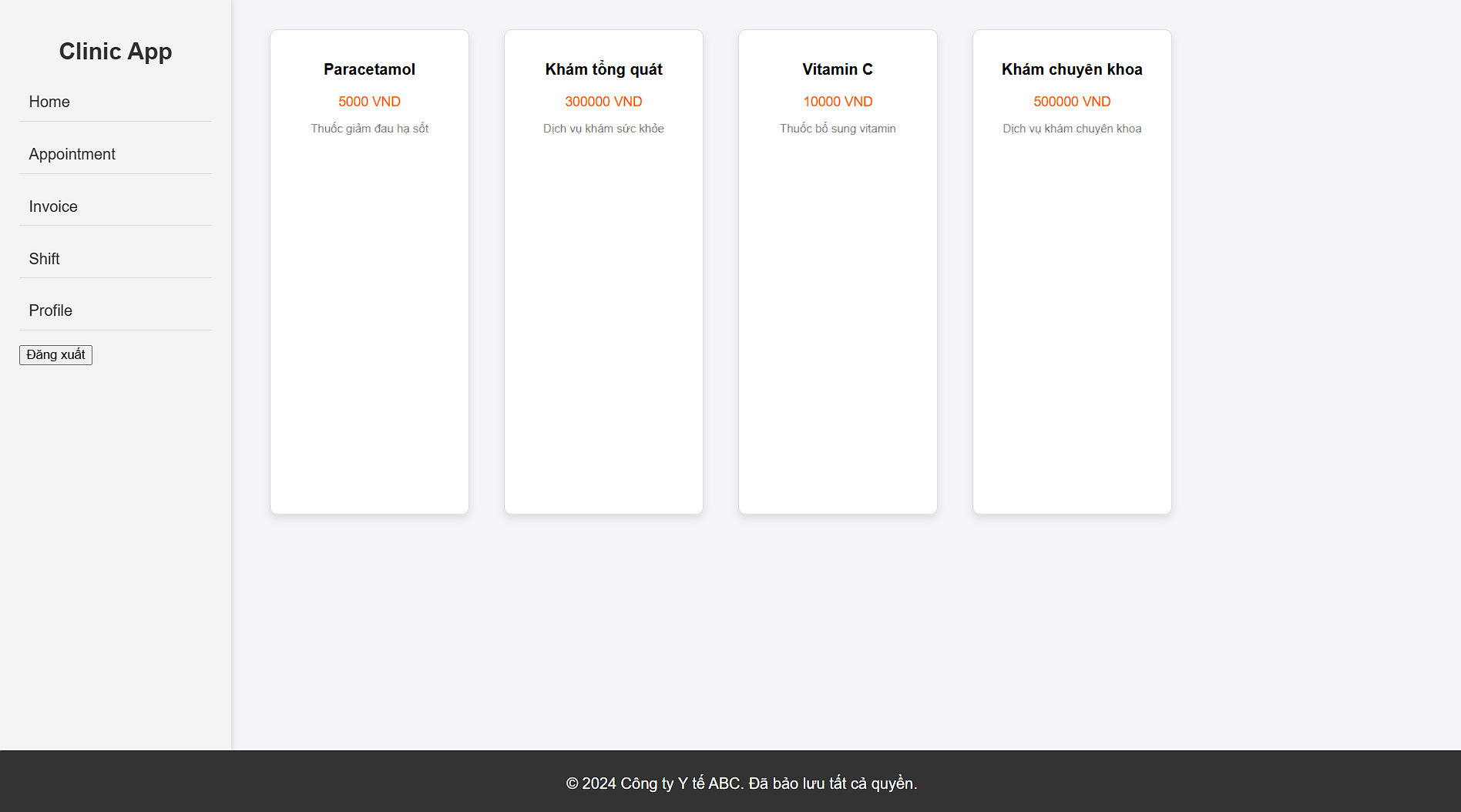


Hình .: Giao diện hóa đơn người dùng

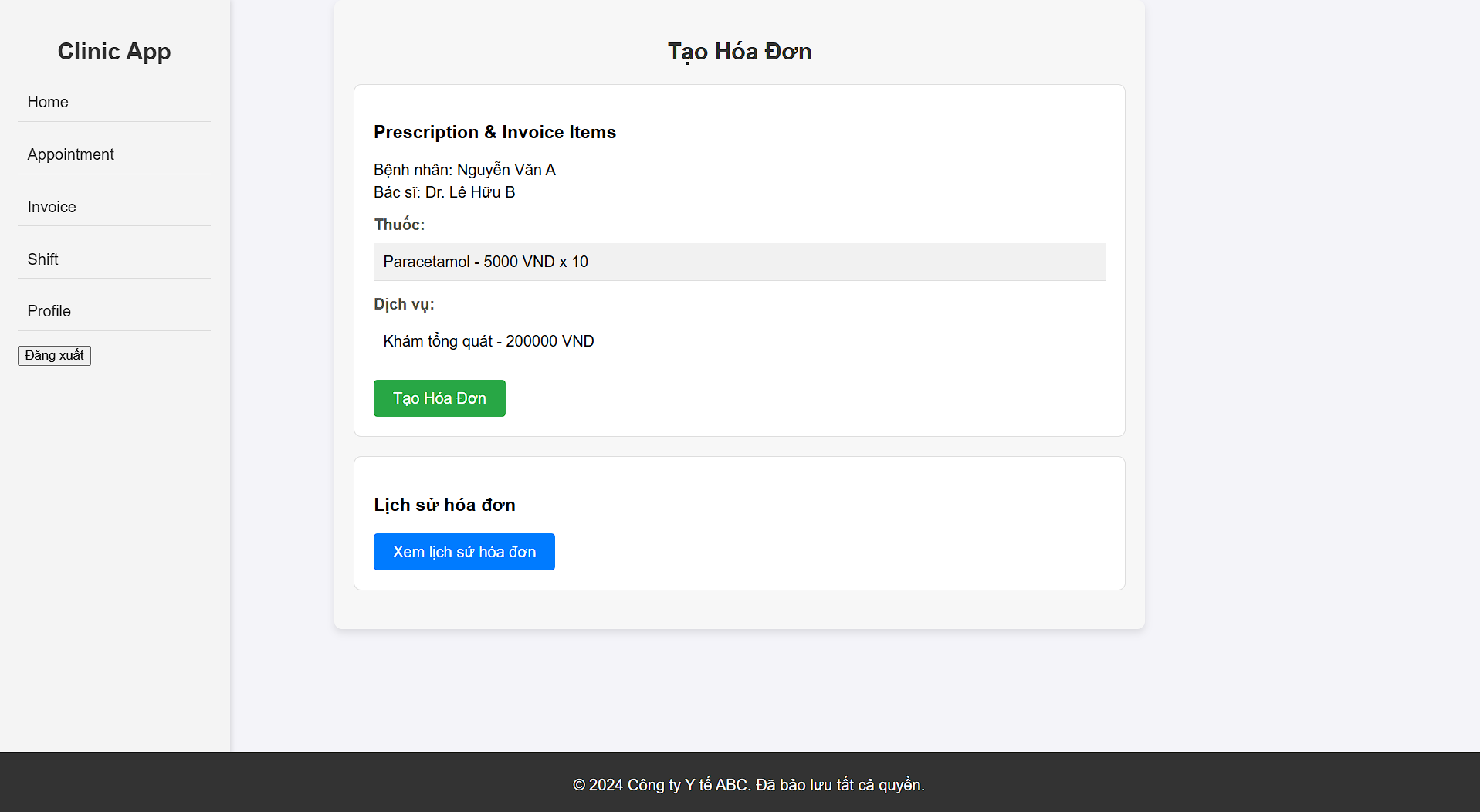


Hình .: Giao diện thông tin Patient

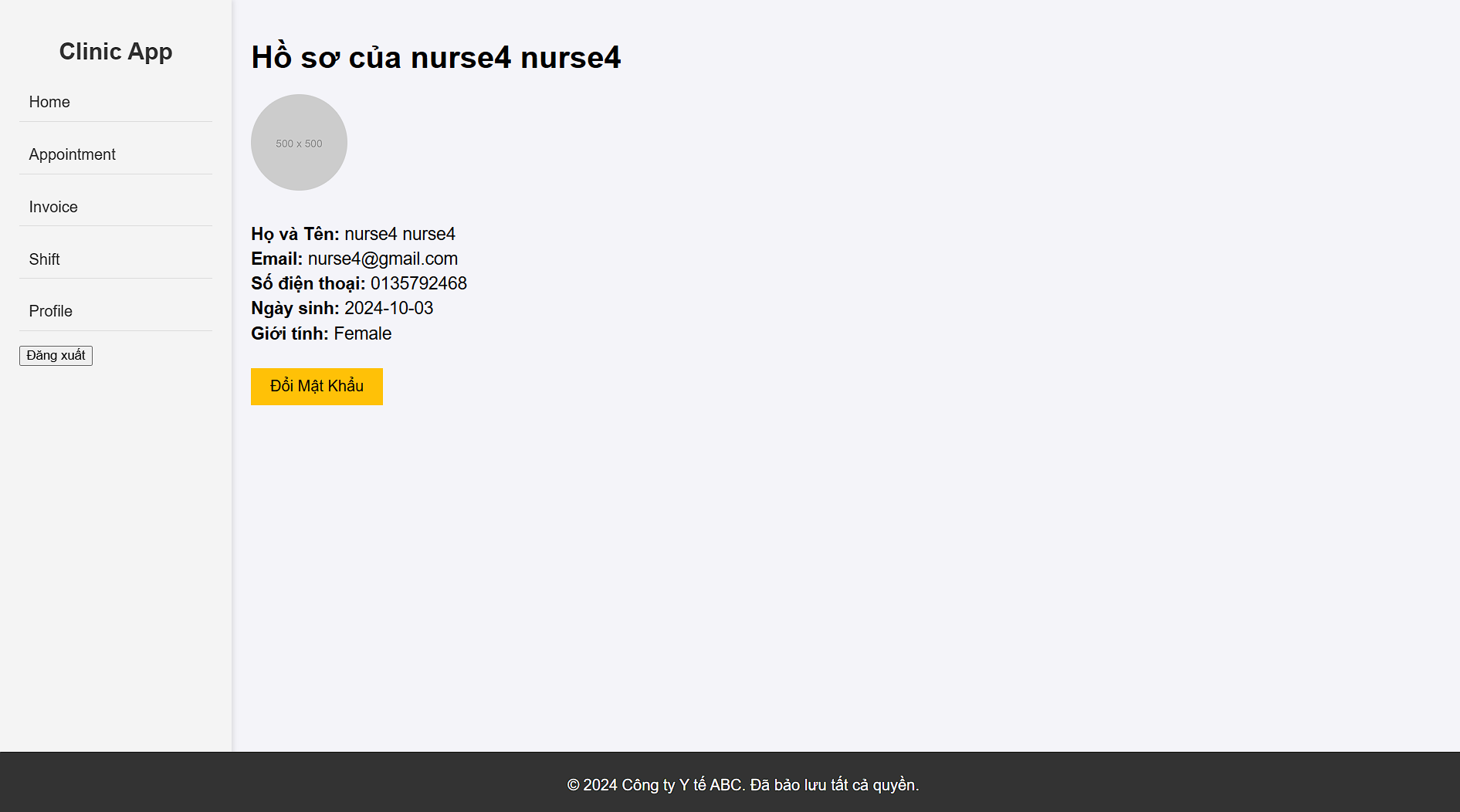
### Giao diện cho Nurse



Hình .: Giao diện chính cho Nurse

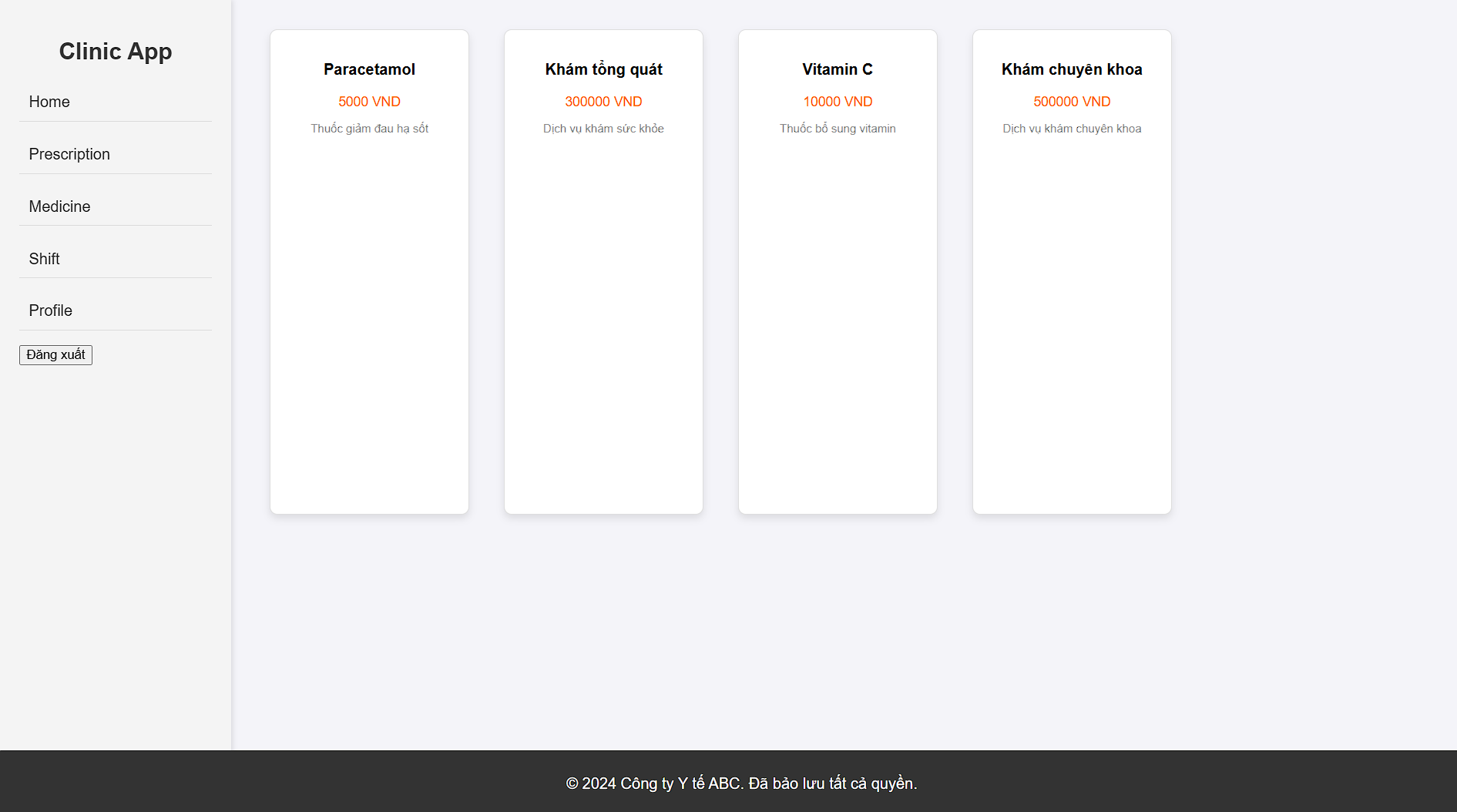


Hình .: Giao diện hóa đơn cho Nurse



Hình .: Giao diện hồ sơ cho Nurse

### Giao diện cho Doctor



Hình .: Giao diện chính cho Doctor



Hình .: Giao diện hồ sơ cho Doctor

# TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

### Đối với cá nhân

* Kiến thức và trải nghiệm về quá trình nghiên cứu và ứng dụng vào đề tài.
* Khả năng phân tích, vận dụng những kiến thức đã học vào đồ án.
* Hiểu rõ hơn về một website phục vụ cho ngành y tế.
* Phát triển được khả năng nghiên cứu, tìm tòi, học hỏi kết hợp và vận dụng vào đồ án

### Đối với sản phẩm

* Bệnh nhân:
* Có thể tạo được tài khoản để có thể sử dụng hệ thống.
* Đặt lịch khám để có thể khám bệnh.
* Y tá:
* Có thể xác nhận lịch khám cho bệnh nhân.
* Kiểm tra lịch làm việc.
* Bác sĩ:
* Có thể nhận lịch khám sau khi y tá đã xác nhận cho bệnh nhân để tiến hành khám bệnh.
* Tìm kiếm thuốc và kê toa thuốc cho bệnh nhân khám bệnh.
* Kiểm tra lịch làm việc.

## Hạn chế

### Đối với hệ thống

* Vai trò quản trị viên (Admin) chưa được phát triển để có thể thống kê cho hệ thống.
* Chức năng avatar chưa được tích hợp Cloudinary để có thể giúp cho người dùng đăng tải avatar.
* Chưa phát triển được chức năng đăng nhập bên thứ ba (Facebook, Google), chưa tích hợp được thanh toán như Momo hoặc ZaloPay.

### Đối với bệnh nhân

* Chưa có chức năng để có thể thanh toán hóa đơn khám bệnh.
* Mỗi lần chỉ có thể đặt lịch một lần duy nhất trong một ngày vì không thể có hai đặt lịch của cùng một bệnh nhân được.

### Đối với y tá

* Chưa có chức năng tạo hóa đơn cho bệnh nhân ngay sau khi khám bệnh xong.
* Chức năng xác nhận lịch khám cũng chưa có thể gửi thông báo xác nhận thông qua Email hoặc SMS.
* Chức năng kiểm tra lịch làm việc (đang phát triển) mới chỉ dừng lại ở việc thêm ngày làm chứ chưa thể cho y tá có thể nhận lịch làm việc được.

### Đối với bác sĩ

* Chức năng chẩn đoán bệnh mặc dù hoạt động nhưng chưa thể thêm hay kê toa thuốc để xác nhận cho bệnh nhân.
* Cũng như y tá thì bác sĩ với chức năng nhận lịch khám cũng chỉ dừng lại ở việc thêm ngày làm và chưa thể cho bác sĩ có thể nhận lịch làm việc.

## Hướng phát triển

* Phát triển thêm các chức năng cho từng vai trò còn thiếu.
* Phát triển thêm chức năng quản trị viên.
* Tối ưu hiệu suất hoạt động của hệ thống.
* Phát triển thêm chức năng gửi nhận thông báo cho các người dùng.
* Ràng buộc các chức năng chặt chẽ hơn.
* Tích hợp các bên thứ ba vào chức năng đăng nhập, thanh toán.
* Tích hợp lọc dữ liệu vào trong các chức năng để có thể dễ dàng sử dụng hơn.
* Cải thiện và hoàn tất các nghiệp vụ còn nhiều hạn chế.

TÀI LIỆU THAM KHẢO

[1] “C Sharp (Ngôn ngữ lập trình)”, 08/09/2024. [Trực tuyến]. Địa chỉ: <https://vi.wikipedia.org/wiki/C_Sharp_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)>

[2] “Tutorial: Create a Web API with ASP.NET Core”, Microsoft, 23/08/2024. [Trực tuyến]. Địa chỉ: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-8.0&tabs=visual-studio>

[3] “JavaScript”, 20/08/2024 [Trực tuyến]. Địa chỉ: <https://vi.wikipedia.org/wiki/JavaScript>

[4] “ReactJS là gì? Những điều có thể bạn chưa biết về ReactJS”, Softwate Engineer Training, 25/06/2021. [Trực tuyến]. Địa chỉ: <https://viblo.asia/p/reactjs-la-gi-nhung-dieu-co-the-ban-chua-biet-ve-reactjs-eW65G4RRKDO>

[5] “Tìm hiểu về hệ quản trị cơ sở dữ liệu PostgreSQL”, Nguyễn Văn Thịnh, 21/08/2021/ [Trực tuyến]. Địa chỉ: <https://viblo.asia/p/tim-hieu-he-quan-tri-co-so-du-lieu-postgresql-m68Z0eLdlkG>

[6] “[[Phần 1] Tổng quan về Entity framework core]”, LongNguyen, 03/08/2024. [Trực tuyến]. Địa chỉ: <https://viblo.asia/p/phan-1-tong-quan-ve-entity-framework-core-4dbZNQNaKYM>

PHỤ LỤC