**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------🕮 🙞🙜---------**

****

**ĐỒ ÁN MÔN: PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG**

**Đề tài: Nghiên cứu các phương pháp xử lý phân loại ảnh (Classification) y khoa và xây dụng ứng dụng hỗ trợ chuẩn đoán bệnh minh họa**

**Giảng viên hướng dẫn: Ths. Phạm Trọng Huynh**

**Sinh viên thực hiện: Bùi Thanh Tuyền**

***TP. Hồ Chí Minh, tháng 11 năm 2014***

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------🕮 🙞🙜---------**

****

**ĐỒ ÁN MÔN: PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG**

**Đề tài: Nghiên cứu các phương pháp xử lý phân loại ảnh (Classification) y khoa và xây dụng ứng dụng hỗ trợ chuẩn đoán bệnh minh họa**

**Giảng viên hướng dẫn: Ths. Phạm Trọng Huynh**

**Sinh viên thực hiện: Bùi Thanh Tuyền**

***TP. Hồ Chí Minh, tháng 11 năm 2014***

# Lời mở đầu

"Trong những năm gần đây, sự phát triển vượt bậc của trí tuệ nhân tạo và học máy đã mang đến những đột phá đáng kể trong lĩnh vực y tế. Đặc biệt, việc ứng dụng các thuật toán học sâu vào xử lý ảnh y khoa đã mở ra triển vọng mới trong việc hỗ trợ chẩn đoán bệnh chính xác và nhanh chóng. Đồ án này tập trung vào việc nghiên cứu các phương pháp xử lý phân loại ảnh (classification) y khoa và xây dựng một ứng dụng hỗ trợ chẩn đoán bệnh minh họa, nhằm tận dụng sức mạnh của OOP để xây dựng một hệ thống phần mềm linh hoạt, mở rộng và hiệu quả."

"Chẩn đoán sớm các bệnh lý là yếu tố quan trọng quyết định hiệu quả điều trị. Tuy nhiên, việc phân tích và đánh giá một lượng lớn hình ảnh y khoa đòi hỏi thời gian và công sức lớn từ phía các bác sĩ."

"Các phương pháp xử lý ảnh y khoa như phân đoạn, trích xuất đặc trưng và phân loại đã được nghiên cứu và ứng dụng rộng rãi. Tuy nhiên, việc lựa chọn và kết hợp các phương pháp này một cách hiệu quả để xây dựng một hệ thống hỗ trợ chẩn đoán bệnh là một thách thức lớn."

"Phương pháp OOP cung cấp một khuôn khổ mạnh mẽ để mô hình hóa các đối tượng trong thế giới thực, từ đó giúp xây dựng các hệ thống phần mềm phức tạp một cách dễ dàng và hiệu quả. Trong đồ án này, OOP sẽ được áp dụng để xây dựng các lớp, đối tượng đại diện cho các loại hình ảnh y khoa, các thuật toán xử lý và các thành phần của ứng dụng."

**Lời cảm ơn**

Để hoàn thành đồ án môn học “Phương pháp phát triển phần mềm hướng đối tượng”.

Đầu tiên, em xin gửi lời cảm ơn chân thành đến thầyThs.Phạm Trọng Huynh, giảng viên hướng dẫn đồ án, đã tận tình chỉ bảo, tạo điều kiện và truyền đạt những kiến thức bổ ích giúp em hoàn thành tốt đồ án này. Nhờ sự hướng dẫn tận tâm của thầy cùng với những kiến thức chuyên môn sâu rộng mà thầy đã truyền đạt, em đã có những kiến thức sâu sắc hơn về “Phương pháp phát triển phần mềm hướng đối tượng” và tự tin hơn trong việc áp dụng vào thực tế.

Chân thành cảm ơn thầy đã quan tâm và hỗ trợ em trong suốt thời gian qua. Những điều này thực sự có ý nghĩa lớn với em và động viên em tiếp tục phấn đấu hơn nữa trong công việc học tập của mình.

Em xin bày tỏ lòng biết ơn sâu sắc đến thầy , ủng hộ em trong mọi bước đi. Sự tin tưởng và lòng quý mến của thầy là nguồn động viên to lớn giúp em không ngừng hoàn thiện và phát triển.

Tuy đã có nhiều cố gắng nhưng do kiến thức và thời gian nghiên cứu có hạn nên đồ án của em không tránh khỏi những sai sót. Do đó, một lần nữa em rất mong nhận được sự góp ý của các thầy để em có điều kiện hoàn thiện hơn kiến thức của mình.

Em xin chân thành cảm ơn thầy!

**Lời cam đoan**

Em cam đoan rằng môn học “Phương pháp phát triển phần mềm hướng đối tượng” không chỉ là một môn học cần thiết trong chương trình đào tạo tôi em còn là một trải nghiệm học tập đầy ý nghĩa và hữu ích. Em hiểu rõ rằng môn học này không chỉ đơn thuần là việc học lý thuyết mà còn là sự kết hợp tuyệt vời giữa kiến thức và thực tiễn

Báo cáo cáo đồ án này mang tên "Nghiên cứu các phương pháp xử lý phân loại ảnh (Classification) y khoa và xây dựng ứng dụng hỗ trợ chuẩn đoán bệnh minh họa" là kết quả của quá trình nghiên cứu độc lập của em. Cam đoan là sản phẩm sẽ tuân theo nguyên tắc của môn học, cá nhân em sẽ cố gắng để sản phẩm của mình có kết quả tốt nhất

Em sẽ cố gắng tích cực sửa lỗi và tiếp thu khi được thầy và các bạn kiểm tra và đánh giá sản phẩm của em, em sẽ cải thiện và phát triển bản thân cũng như phát triển hơn trong đề tài lần sau.

**Nhận xét của giảng viên**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

ngày….tháng…. năm 2023

Giảng viên

Lê Quang Thiện

MỤC LỤC

[Lời mở đầu 3](#_Toc183054348)

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 10](#_Toc183054349)

[1.1 Khảo sát hiện trạng 10](#_Toc183054350)

[1.1.1 Đặt vấn đề 10](#_Toc183054351)

[1.1.2 Hiện trạng 10](#_Toc183054352)

[1.1.3 Mục tiêu nghiên cứu 12](#_Toc183054353)

[1.1.4 Hướng Đi Nghiên Cứu 12](#_Toc183054354)

[1.1.5 Phạm vi nghiên cứu 13](#_Toc183054355)

[1.1.6 Đóng góp của nghiên cứu 13](#_Toc183054356)

[1.1.7 Yêu cầu hệ thống 13](#_Toc183054357)

[1.1.8 Mục tiêu của hệ thống 15](#_Toc183054358)

[1.1.9 Ứng dụng thực tế 15](#_Toc183054359)

[1.2 Yêu cầu hệ thống 16](#_Toc183054360)

[1.2.1 Sơ đồ tổ chức hệ thống 16](#_Toc183054361)

[1.2.2 Chức năng nhiệm vụ của từng bộ phận 17](#_Toc183054362)

[1.2.3 Danh sách các yêu cầu 19](#_Toc183054363)

[1.2.4 Danh sách các biểu mẫu 21](#_Toc183054364)

[CHƯƠNG Ⅱ: MÔ HÌNH USECASE 25](#_Toc183054365)

[2.1. Mô hình UseCase 25](#_Toc183054366)

[2.2. Danh sách các Actor 28](#_Toc183054367)

[2.3 Danh Sách Các Use Case 28](#_Toc183054368)

[2.4 Đặt tả UseCase 29](#_Toc183054369)

[2.4.1. Đặt tả UseCase “Đăng Nhập” 29](#_Toc183054370)

[2.4.2. Đặt tả UseCase “Đăng Xuất” 30](#_Toc183054371)

[2.4.3. Đặt tả UseCase “Đổi Mật Khẩu” 30](#_Toc183054372)

[2.4.4. Đặt tả UseCase “Cập Nhật Thông Tin Bệnh Nhân” 31](#_Toc183054373)

[2.4.5. Đặt tả UseCase “Chẩn đoán và lưu trữ hồ sơ bệnh” 32](#_Toc183054374)

[2.4.6. Đặt tả UseCase “Quản Lý Hình Ảnh Y Khoa” 33](#_Toc183054375)

[2.4.7. Đặt tả UseCase “Truy Xuất Dữ Liệu Bệnh Nhân” 34](#_Toc183054376)

[2.4.8. Đặt tả UseCase “Tư Vấn Và Hỗ Trợ Điều Trị” 34](#_Toc183054377)

[2.4.9. Đặt tả UseCase “Lập Báo Cáo Tiến Trình Điều Trị” 35](#_Toc183054378)

[2.4.10. Đặt tả UseCase “Tra Cứu Thông Tin Bệnh” 35](#_Toc183054379)

[2.4.11. Đặt tả UseCase “Nhận Tư Vấn Và Hỗ Trợ Điều Trị” 37](#_Toc183054380)

[2.4.12. Đặt tả UseCase “Tra Cứu Lịch Sử Khám Chữa Bệnh” 37](#_Toc183054381)

[2.4.13. Đặt tả UseCase “Quản Lý Thông Tin Hệ Thống” 38](#_Toc183054382)

[2.4.14. Đặc tả UseCase “Quản Lý Người Dùng” 38](#_Toc183054383)

[2.4.15. Đặc tả UsesCase “Cập Nhật Thông Tin Cá Nhân” 39](#_Toc183054384)

[CHƯƠNG III: PHÂN TÍCH 40](#_Toc183054385)

[3.1 Sơ đồ lớp 40](#_Toc183054386)

[3.1.1 Sơ đồ lớp ( mức phân tích ) 40](#_Toc183054387)

[3.1.2 Danh sách các lớp đối tượng và quan hệ 44](#_Toc183054388)

[3.1.3 Mô tả chi tiết đối tượng 48](#_Toc183054389)

[3.2 Sơ đồ trạng thái 51](#_Toc183054390)

[3.2.1 Sơ đồ trạng thái tổng quát 51](#_Toc183054391)

[3.2.2 Sơ đồ trạng thái đăng nhập 52](#_Toc183054392)

[3.2.3 Sơ đồ trạng thái đăng xuất 53](#_Toc183054393)

[3.2.4 Sơ đồ trạng thái đổi mật khẩu 54](#_Toc183054394)

[3.2.5 Sơ đồ trạng thái cập nhập thông tin 55](#_Toc183054395)

[3.2.6 Sơ đồ trạng thái chuẩn đoán và lưu trữ hồ sơ 56](#_Toc183054396)

[3.2.7 Sơ đồ trạng thái quản lý hình ảnh y khoa 57](#_Toc183054397)

[3.2.8 Sơ đồ trạng thái truy xuất dữ liệu bệnh nhân 58](#_Toc183054398)

[3.2.9 Sơ đồ trạng thái nhận tư vấn và hỗ trợ điều trị 59](#_Toc183054399)

[3.2.10 Sơ đồ trạng thái lập báo cáo tiến trình điều trị 60](#_Toc183054400)

[3.2.11 Sơ đồ trạng thái tra cứu thông tin bệnh 61](#_Toc183054401)

[3.2.12 Sơ đồ trạng thái tư vấn và hỗ trợ điều trị 62](#_Toc183054402)

[3.2.13 Sơ đồ trạng thái tra cứu lịch sử khám chữa bệnh 63](#_Toc183054403)

[3.2.14 Sơ đồ trạng thái quản lý thông tin hệ thống 64](#_Toc183054404)

[3.2.15 Sơ đồ trạng thái quản lý người dùng 65](#_Toc183054405)

[3.2.16 Sơ đồ trạng thái cập nhập thông tin cá nhân 66](#_Toc183054406)

[3.3 Sơ đồ tuần tự 67](#_Toc183054407)

[3.3.1 Đăng nhập 67](#_Toc183054408)

[3.3.2 Đăng xuất 68](#_Toc183054409)

[3.3.3 Đăng kí 71](#_Toc183054410)

[CHƯƠNG IV: THIẾT KẾ DỮ LIỆU LƯU TRỮ 73](#_Toc183054411)

[4.1 Sơ đồ logic 73](#_Toc183054412)

[4.2 Mô tả chi tiết các thành phần trong sơ đồ logic 74](#_Toc183054413)

[4.2.1 Đối tượng người dùng (User) 74](#_Toc183054414)

[4.2.2 Đối tượng: Bệnh (Disease) 78](#_Toc183054415)

[4.2.3 Đối tượng: Hồ sơ bệnh án (Medical Record) 80](#_Toc183054416)

[4.2.4 Đối tượng: Báo cáo (Report) 83](#_Toc183054417)

[4.2.5 Đối tượng: Hình ảnh y khoa (medical Image) 86](#_Toc183054418)

[4.2.6 Đối tượng: Lịch sử khám chữa bệnh (Medical History) 89](#_Toc183054419)

[CHƯƠNG V: THIẾT KẾ GIAO DIỆN 92](#_Toc183054420)

[5.1 Kiến trúc hệ thống 92](#_Toc183054421)

[**5.1.1 Sơ đồ kiến trúc tổng quan** 92](#_Toc183054422)

[5.1.2 Sơ đồ lớp hệ thống chăm sóc sức khỏe trực tiếp 93](#_Toc183054423)

[5.1.3 Sơ đồ phát triển hệ thống chăm sóc sức khỏe trực tiếp 96](#_Toc183054424)

[5.1.4 Sơ đồ quan sát vật lý của hệ thống chăm sóc sức khỏe trực tiếp 100](#_Toc183054425)

[5.1.5 Biểu đồ trường hợp sử dụng hệ thống chăm sóc sức khỏe trực tiếp 101](#_Toc183054426)

[5.1.6 Lựa chọn Django Framework 103](#_Toc183054427)

[**5.1.7 Mô hình kiến trúc hệ thống** 104](#_Toc183054428)

[5.2 Chi tiết kỹ thuật 105](#_Toc183054429)

[5.2.1 Cơ sở dữ liệu 105](#_Toc183054430)

[5.2.2 API 106](#_Toc183054431)

[5.2.3 Bảo mật 107](#_Toc183054432)

[5.2.4 Tích hợp 107](#_Toc183054433)

[5.3 Mô tả chi tiết thành phần trong hệ thống 107](#_Toc183054434)

[CHƯƠNG 6: GIAO DIỆN HOÀN THÀNH 109](#_Toc183054435)

[CHƯƠNG 7: KẾT LUẬN 110](#_Toc183054436)

[7.1 Môi trường phát triển và môi trường triển khai 110](#_Toc183054437)

[7.2 Kết quả đạt được 111](#_Toc183054438)

[**7.2.1 Kết quả tổng quan:** 111](#_Toc183054439)

[7.2.2 Phân tích chi tiết các kết quả 111](#_Toc183054440)

[7.2.3 Các yếu tố góp phần thành công 112](#_Toc183054441)

[7.3 Hướng phát triển 113](#_Toc183054442)

[TÀI LIỆU THAM KHẢO 115](#_Toc183054443)

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

## Khảo sát hiện trạng

* + 1. **Đặt vấn đề**

**Tầm quan trọng của chẩn đoán hình ảnh y tế:** Nhấn mạnh vai trò không thể thiếu của hình ảnh y tế (X-quang, MRI, CT...) trong việc chẩn đoán và điều trị bệnh.

**Thách thức trong phân loại hình ảnh y tế:** Đề cập đến các khó khăn như:

* Độ phức tạp của hình ảnh: Nhiều loại bệnh có biểu hiện tương tự, nhiễu, biến đổi về kích thước, góc độ.
* Số lượng lớn dữ liệu: Cần xử lý và phân tích một lượng lớn hình ảnh y tế.
* Yêu cầu cao về độ chính xác: Sai sót trong chẩn đoán có thể ảnh hưởng đến tính mạng bệnh nhân.

**Vai trò của trí tuệ nhân tạo (AI) và học sâu (deep learning) trong y tế:** Giới thiệu về sự phát triển của AI và deep learning, đặc biệt là trong lĩnh vực y tế. Nhấn mạnh khả năng của các mô hình học sâu trong việc phân tích hình ảnh với độ chính xác cao.

### 1.1.2 Hiện trạng

* **Tình trạng phổ biến**: Các bệnh ngoài da như viêm da, mụn trứng cá, chàm, nấm da, v.v., rất phổ biến ở mọi lứa tuổi, từ trẻ em đến người lớn.
* **Ảnh hưởng**:
  + Bệnh tuy không nguy hiểm đến tính mạng nhưng gây mất thẩm mỹ, ảnh hưởng đến ngoại hình và khiến người bệnh thiếu tự tin trong giao tiếp.
  + Tình trạng bệnh có thể gây khó chịu, ngứa ngáy, ảnh hưởng đến sinh hoạt hàng ngày của người bệnh.
* **Hạn chế trong việc tiếp cận thông tin**: Nhiều người gặp khó khăn trong việc tiếp cận thông tin về các bệnh ngoài da hoặc không biết cách điều trị đúng đắn.
* **Thiếu hệ thống hỗ trợ**: Hiện nay chưa có nhiều hệ thống trực tuyến tập trung vào việc cung cấp thông tin và hỗ trợ điều trị cho các bệnh ngoài da.
  + 1. **Mục tiêu nghiên cứu**

**Mục tiêu chung:** Xây dựng một hệ thống hỗ trợ chẩn đoán bệnh dựa trên phân loại hình ảnh y tế, nhằm nâng cao hiệu quả và độ chính xác trong quá trình chẩn đoán.

**Các mục tiêu cụ thể:**

* Nghiên cứu các phương pháp phân loại hình ảnh y tế hiện đại (CNN, Transformer,...) và đánh giá hiệu quả của chúng.
* Xây dựng một mô hình phân loại hiệu quả cho loại hình ảnh y tế cụ thể (ví dụ: X-quang phổi, MRI não).
* Phát triển một ứng dụng hỗ trợ chẩn đoán bệnh dựa trên mô hình đã xây dựng.

### 1.1.4 Hướng Đi Nghiên Cứu

**- Phát triển các thuật toán phân đoạn hiệu quả hơn**: Nghiên cứu tập trung vào việc cải thiện độ chính xác, tốc độ và khả năng tổng quát của các thuật toán phân đoạn. Điều này có thể bao gồm việc cải tiến các kiến trúc mạng nơ-ron, áp dụng các kỹ thuật học sâu mới và phát triển các phương pháp lai ghép giữa các mô hình khác nhau để đạt được kết quả tốt hơn trong phân đoạn ảnh y tế.

**- Xây dựng các cơ sở dữ liệu lớn và chất lượng cao**: Việc thu thập và chú thích một lượng lớn dữ liệu y tế có chất lượng cao là rất cần thiết để huấn luyện các mô hình học sâu hiệu quả. Các cơ sở dữ liệu này cần phải được xây dựng với sự hợp tác của các chuyên gia y tế để đảm bảo tính chính xác của các nhãn. Điều này không chỉ giúp cải thiện độ chính xác của các mô hình mà còn tạo ra các tài nguyên mở để cộng đồng nghiên cứu có thể sử dụng.

**- Phát triển các phương pháp tăng cường dữ liệu**: Tăng cường số lượng và đa dạng của dữ liệu huấn luyện thông qua các kỹ thuật như xoay, lật, thay đổi độ sáng hoặc thêm nhiễu là rất quan trọng để cải thiện khả năng tổng quát của mô hình. Những phương pháp này giúp tạo ra các mẫu dữ liệu mới từ các hình ảnh hiện có, từ đó làm cho mô hình trở nên mạnh mẽ hơn trong việc phân đoạn các cấu trúc trong các tình huống khác nhau.

**- Nghiên cứu các phương pháp giải thích**: Phát triển các công cụ và phương pháp giúp giải thích các quyết định của mô hình là rất quan trọng trong việc xây dựng sự tin tưởng giữa các bác sĩ và các công cụ phân đoạn tự động. Các phương pháp như Grad-CAM (Gradient-weighted Class Activation Mapping) hoặc LIME (Local Interpretable Model-agnostic Explanations) có thể được áp dụng để trực quan hóa các khu vực trong ảnh mà mô hình dựa vào để đưa ra quyết định, từ đó giúp bác sĩ hiểu rõ hơn về lý do đằng sau các kết quả phân đoạn.

### 1.1.5 Phạm vi nghiên cứu

* **Loại hình ảnh y tế:** Xác định rõ loại hình ảnh mà bạn sẽ tập trung nghiên cứu (ví dụ: X-quang, MRI, CT,...) và các loại bệnh liên quan.
* **Các phương pháp phân loại:** Liệt kê các phương pháp phân loại sẽ được nghiên cứu và so sánh.
* **Các chỉ số đánh giá:** Xác định các chỉ số đánh giá hiệu suất của mô hình (ví dụ: độ chính xác, độ nhạy, độ đặc hiệu).

### 1.1.6 Đóng góp của nghiên cứu

* **Đóng góp khoa học:** Đưa ra những đóng góp mới vào lĩnh vực phân loại hình ảnh y tế, đặc biệt là trong việc ứng dụng các kỹ thuật học sâu.
* **Đóng góp thực tiễn:** Cung cấp một công cụ hỗ trợ hữu ích cho các bác sĩ trong quá trình chẩn đoán bệnh, góp phần nâng cao chất lượng chăm sóc sức khỏe.

**1.1.7 Yêu cầu hệ thống**

* **Cung cấp thông tin**:
  + Hệ thống cần cung cấp thông tin chi tiết về các loại bệnh ngoài da, nguyên nhân, triệu chứng, và các phương pháp phòng ngừa.
  + Cập nhật các phương pháp điều trị và những lưu ý về chăm sóc da trong quá trình điều trị bệnh.
* **Tư vấn và chẩn đoán ban đầu**:
  + Cần có một chức năng hỗ trợ tư vấn và chẩn đoán ban đầu để người dùng có thể tự đánh giá tình trạng của mình dựa trên các triệu chứng mô tả.
* **Hỗ trợ hình ảnh**:
  + Hệ thống có thể bao gồm chức năng tải ảnh để so sánh hoặc phân tích các triệu chứng ngoài da, giúp người dùng dễ dàng xác định bệnh.
* **Đề xuất điều trị**:
  + Cung cấp thông tin về các phương pháp điều trị khác nhau (thuốc bôi ngoài da, thuốc uống, liệu pháp tự nhiên) phù hợp với từng loại bệnh.

**1.1.8 Mục tiêu của hệ thống**

* Tăng cường kiến thức về các bệnh ngoài da cho người dùng, giúp họ hiểu rõ hơn về tình trạng của mình.
* Hỗ trợ người bệnh có thể tự tin hơn trong giao tiếp và cuộc sống hàng ngày thông qua sự chăm sóc và điều trị đúng cách.

**1.1.9 Ứng dụng thực tế**

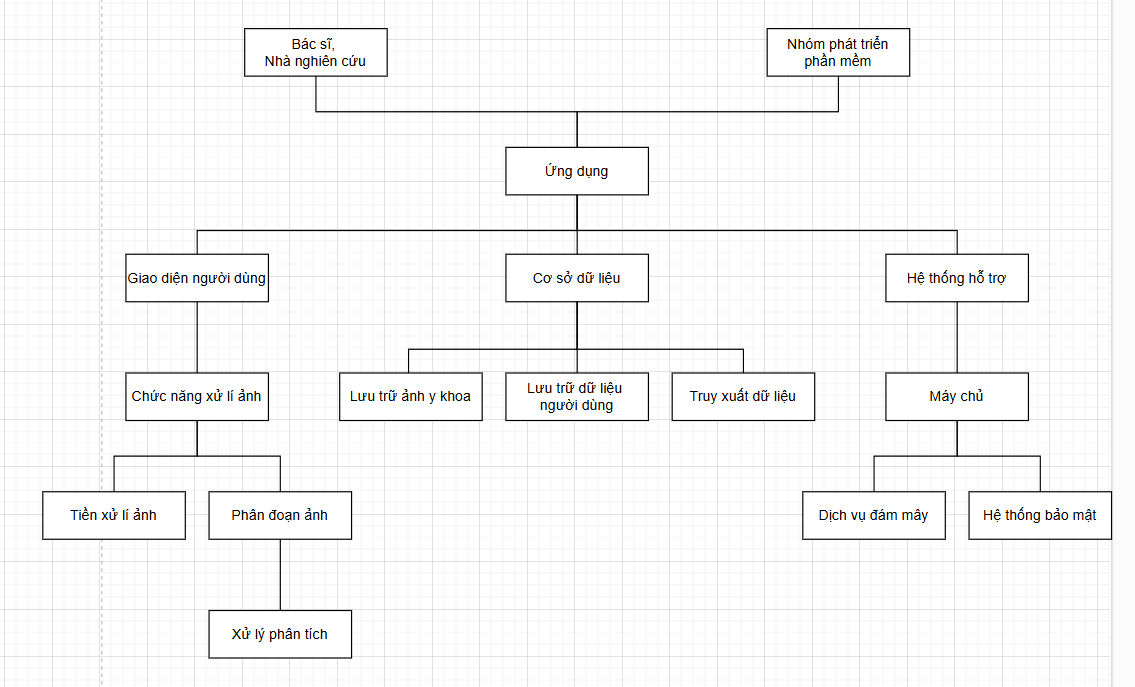
- **Hỗ trợ chẩn đoán**: Các hệ thống phân đoạn ảnh tự động có vai trò quan trọng trong việc hỗ trợ bác sĩ đưa ra chẩn đoán chính xác và nhanh chóng hơn. Bằng cách sử dụng các thuật toán học sâu để phân tích hình ảnh y tế, hệ thống này có thể phát hiện và phân đoạn các cấu trúc bất thường, chẳng hạn như khối u, tổn thương hoặc các bất thường về hình dạng. Việc này giúp giảm thiểu sai sót do con người và cung cấp các phát hiện sớm hơn, từ đó tăng cường khả năng can thiệp kịp thời. Các kết quả phân đoạn có thể được hiển thị trực quan, giúp bác sĩ dễ dàng nhận diện và đánh giá các vấn đề sức khỏe.

- **Lập kế hoạch điều trị**: Thông qua việc cung cấp thông tin chi tiết về kích thước, hình dạng và vị trí của các khối u hoặc tổn thương, phân đoạn ảnh hỗ trợ bác sĩ trong việc lập kế hoạch điều trị hiệu quả. Đối với các ca phẫu thuật, thông tin này giúp các bác sĩ xác định phạm vi can thiệp cần thiết, từ đó tối ưu hóa quy trình phẫu thuật và giảm thiểu rủi ro. Trong trường hợp điều trị xạ trị, việc phân đoạn các vùng mô cần điều trị và mô khỏe mạnh xung quanh là rất quan trọng để tối ưu hóa liều lượng bức xạ, giảm thiểu tác dụng phụ cho bệnh nhân và tăng hiệu quả điều trị.

- **Theo dõi tiến trình điều trị**: Phân đoạn ảnh cũng rất hữu ích trong việc theo dõi hiệu quả của quá trình điều trị. Bằng cách so sánh các hình ảnh được phân đoạn ở các thời điểm khác nhau, bác sĩ có thể đánh giá sự thay đổi về kích thước và hình dạng của khối u hoặc tổn thương, từ đó xác định xem liệu điều trị có hiệu quả hay không. Việc theo dõi này giúp phát hiện kịp thời sự tái phát của bệnh hoặc sự phát triển của các tổn thương mới, từ đó điều chỉnh kế hoạch điều trị phù hợp. Các hệ thống phân đoạn tự động có thể tự động hóa quá trình này, giúp tiết kiệm thời gian và nguồn lực cho bác sĩ, đồng thời nâng cao độ chính xác trong việc theo dõi tiến triển bệnh lý.

## Yêu cầu hệ thống

### 1.2.1 Sơ đồ tổ chức hệ thống



### 1.2.2 Chức năng nhiệm vụ của từng bộ phận

* **Bác sĩ, Nhà nghiên cứu**:
  + **Tham gia vào quá trình xây dựng mô hình:** Không chỉ cung cấp dữ liệu, các chuyên gia y tế còn cần tham gia vào quá trình thiết kế và đánh giá các mô hình học máy, đảm bảo rằng các kết quả dự đoán phù hợp với kiến thức y khoa.
  + **Đánh giá và cải tiến liên tục:** Thực hiện việc đánh giá thường xuyên các kết quả dự đoán của hệ thống, so sánh với kết quả chẩn đoán thực tế và đưa ra các đề xuất cải tiến.
  + **Phát triển các hướng dẫn sử dụng:** Tạo ra các hướng dẫn chi tiết về cách sử dụng ứng dụng cho các bác sĩ và người dùng khác.
* **Nhóm phát triển phần mềm**:
  + **Xây** **dựng API:** Phát triển một giao diện lập trình ứng dụng (API) để cho phép các ứng dụng bên thứ ba tích hợp với hệ thống, mở rộng khả năng ứng dụng.
  + **Đảm bảo tính tương thích:** Đảm bảo ứng dụng tương thích với nhiều nền tảng khác nhau (web, di động) và các thiết bị y tế.
  + **Quản lý phiên bản:** Áp dụng các công cụ và quy trình quản lý phiên bản để theo dõi các thay đổi và cập nhật của ứng dụng.
* **Giao diện người dùng**:
  + **Thiết kế trực quan và dễ sử dụng:** Giao diện cần được thiết kế đơn giản, trực quan, giúp người dùng dễ dàng sử dụng các chức năng của ứng dụng, ngay cả khi không có kiến thức chuyên môn về công nghệ.
  + **Cung cấp các công cụ hỗ trợ:** Cung cấp các công cụ hỗ trợ người dùng trong việc chụp ảnh, chỉnh sửa ảnh và lựa chọn các vùng quan tâm trên hình ảnh.
  + **Hiển thị kết quả rõ ràng:** Hiển thị kết quả dự đoán của mô hình một cách rõ ràng, kèm theo các thông tin giải thích chi tiết để người dùng dễ hiểu.
* **Cơ sở dữ liệu**:
  + **Bảo mật dữ liệu**: Áp dụng các biện pháp bảo mật nghiêm ngặt để bảo vệ dữ liệu người dùng, tuân thủ các quy định về bảo mật thông tin y tế.
  + **Sao lưu và phục hồi**: Thực hiện việc sao lưu dữ liệu định kỳ và xây dựng các quy trình phục hồi dữ liệu khi xảy ra sự cố.
  + **Truy vấn dữ liệu linh hoạt**: Cung cấp các công cụ truy vấn dữ liệu linh hoạt để hỗ trợ các hoạt động phân tích và báo cáo.
* **Hệ thống hỗ trợ**:
  + **Tăng cường khả năng mở rộng:** Thiết kế hệ thống để có thể dễ dàng mở rộng quy mô khi số lượng người dùng tăng lên và lượng dữ liệu lớn hơn.
  + **Đảm bảo tính sẵn sàng:** Đảm bảo hệ thống hoạt động ổn định 24/7, giảm thiểu thời gian gián đoạn.
  + **Theo dõi và cảnh báo:** Xây dựng hệ thống giám sát để theo dõi tình trạng hoạt động của hệ thống và gửi cảnh báo khi phát hiện sự cố.
* **Các chức năng cần phát triển thêm:**
  + **Phân tích thời gian thực:** Phân tích các xu hướng thay đổi của bệnh nhân theo thời gian dựa trên lịch sử bệnh án và hình ảnh.
  + **Tích hợp với các hệ thống khác:** Tích hợp với các hệ thống quản lý bệnh viện, hệ thống lưu trữ hình ảnh y tế (PACS) để tạo thành một hệ sinh thái y tế hoàn chỉnh.
  + **Học máy liên tục:** Áp dụng các kỹ thuật học máy liên tục để cải thiện độ chính xác của mô hình theo thời gian.

### 1.2.3 Danh sách các yêu cầu

* Cung cấp thông tin về bệnh ngoài da một cách chi tiết và dễ hiểu.
* Cho phép người dùng chẩn đoán ban đầu qua các chức năng xử lý ảnh.
* Lưu trữ và bảo mật dữ liệu y tế của người dùng.
* Hỗ trợ bác sĩ và nhà nghiên cứu truy cập và phân tích dữ liệu nhanh chóng.
* Đảm bảo giao diện thân thiện, dễ sử dụng cho mọi lứa tuổi.

**Yêu cầu về nội dung:**

* **Cơ sở dữ liệu bệnh:**
  + Bao gồm thông tin chi tiết về các loại bệnh ngoài da phổ biến:
    - Định nghĩa, nguyên nhân, triệu chứng.
    - Hình ảnh minh họa đa dạng (góc độ, mức độ nghiêm trọng).
    - Phương pháp điều trị, phòng ngừa.
  + Cập nhật liên tục thông tin dựa trên các nghiên cứu y khoa mới nhất.
* **Thông tin về thuốc:**
  + Cung cấp thông tin về các loại thuốc điều trị bệnh ngoài da:
    - Công dụng, liều dùng, tác dụng phụ.
    - Tương tác thuốc.

**Yêu cầu về chức năng:**

* **Chức năng chẩn đoán ban đầu:**
  + **Xử lý ảnh:**
    - Nhận diện các vùng bị tổn thương.
    - Phân tích các đặc điểm hình thái của tổn thương (màu sắc, kích thước, hình dạng).
    - So sánh với cơ sở dữ liệu hình ảnh bệnh để đưa ra gợi ý chẩn đoán.
  + **Trợ lý ảo:**
    - Hỏi đáp trực tuyến về các triệu chứng, giúp người dùng xác định bệnh.
    - Cung cấp các câu hỏi gợi ý để thu thập thông tin đầy đủ.
* **Lưu trữ dữ liệu:**
  + **Lịch sử bệnh án:**
    - Lưu trữ các thông tin về các lần khám bệnh, kết quả chẩn đoán, phương pháp điều trị.
    - Cho phép người dùng theo dõi tiến trình điều trị.
  + **Ảnh bệnh nhân:**
    - Lưu trữ hình ảnh của bệnh nhân tại các thời điểm khác nhau để so sánh và đánh giá hiệu quả điều trị.
* **Hỗ trợ bác sĩ:**
  + **Báo cáo:**
    - Tạo các báo cáo thống kê về các loại bệnh thường gặp, xu hướng phát triển của bệnh.
  + **Công cụ phân tích:**
    - Cung cấp các công cụ để phân tích dữ liệu lớn, tìm kiếm các mối liên hệ giữa các yếu tố nguy cơ và bệnh.
* **Giao diện thân thiện:**
  + **Đơn giản, trực quan:**
    - Sử dụng ngôn ngữ dễ hiểu, tránh thuật ngữ chuyên ngành.
    - Thiết kế giao diện đẹp mắt, dễ sử dụng.
  + **Tính năng tìm kiếm:**
    - Cho phép người dùng tìm kiếm thông tin về bệnh, thuốc một cách nhanh chóng.
  + **Tính năng cá nhân hóa:**
    - Cho phép người dùng tạo hồ sơ cá nhân, lưu trữ các thông tin liên quan đến sức khỏe của mình.

**Yêu cầu về bảo mật:**

* **Bảo mật dữ liệu:**
  + Mã hóa dữ liệu người dùng.
  + Quản lý quyền truy cập chặt chẽ.
  + Tuân thủ các quy định về bảo mật thông tin y tế.
* **Bảo mật hệ thống:**
  + Xây dựng hệ thống tường lửa, phát hiện và ngăn chặn các cuộc tấn công.
  + Thường xuyên cập nhật phần mềm, vá lỗ hổng bảo mật.

### 1.2.4 Danh sách các biểu mẫu

**Biểu mẫu đăng ký thông tin người dùng**: Thu thập thông tin cá nhân, lịch sử bệnh án của người dùng.

* **Thông tin cá nhân:** Tên, ngày sinh, giới tính, số điện thoại, địa chỉ email, địa chỉ.
* **Lịch sử bệnh án:** Các bệnh đã mắc phải, dị ứng thuốc, các cuộc phẫu thuật trước đây.
* **Thông tin liên hệ người thân:** Tên, số điện thoại của người thân để liên hệ trong trường hợp cần thiết.
* **Quyền riêng tư:** Một phần để người dùng đồng ý với các điều khoản sử dụng và chính sách bảo mật của ứng dụng.

**Biểu mẫu tải ảnh y khoa**: Người dùng tải lên hình ảnh các triệu chứng để hệ thống xử lý và phân tích.

* **Hướng dẫn chụp ảnh:** Các hướng dẫn chi tiết về cách chụp ảnh vùng da bị tổn thương (góc độ, ánh sáng, khoảng cách).
* **Chọn loại hình ảnh:** Cho phép người dùng chọn loại hình ảnh cần tải lên (ảnh chụp mới, ảnh cũ).
* **Mô tả triệu chứng:** Một ô văn bản để người dùng mô tả chi tiết về các triệu chứng mà họ đang gặp phải.

**Biểu mẫu nhập liệu chẩn đoán**: Được sử dụng bởi bác sĩ để nhập kết quả chẩn đoán và thông tin điều trị.

* **Thông tin bệnh nhân:** Hiển thị lại thông tin bệnh nhân đã đăng ký.
* **Kết quả chẩn đoán:** Bác sĩ chọn hoặc nhập trực tiếp tên bệnh.
* **Phương pháp điều trị:** Mô tả chi tiết các phương pháp điều trị được đề xuất.
* **Thuốc kê đơn:** Liệt kê các loại thuốc, liều dùng, cách dùng.
* **Các xét nghiệm bổ sung:** Nếu cần, bác sĩ có thể yêu cầu người bệnh thực hiện các xét nghiệm bổ sung.
* **Lịch hẹn tái khám:** Đặt lịch hẹn cho bệnh nhân.

**Biểu mẫu phản hồi người dùng**: Cho phép người dùng cung cấp phản hồi về trải nghiệm sử dụng hệ thống và chất lượng chẩn đoán.

* **Đánh giá chung:** Người dùng đánh giá chung về ứng dụng trên thang điểm 5 sao.
* **Đánh giá từng chức năng:** Đánh giá cụ thể từng chức năng của ứng dụng (chức năng chẩn đoán, giao diện, tốc độ phản hồi).
* **Gợi ý cải tiến:** Ô trống để người dùng đưa ra các ý kiến, góp ý nhằm cải thiện ứng dụng.
* **Báo cáo lỗi:** Cho phép người dùng báo cáo các lỗi kỹ thuật gặp phải trong quá trình sử dụng.

**Các lưu ý khi thiết kế biểu mẫu:**

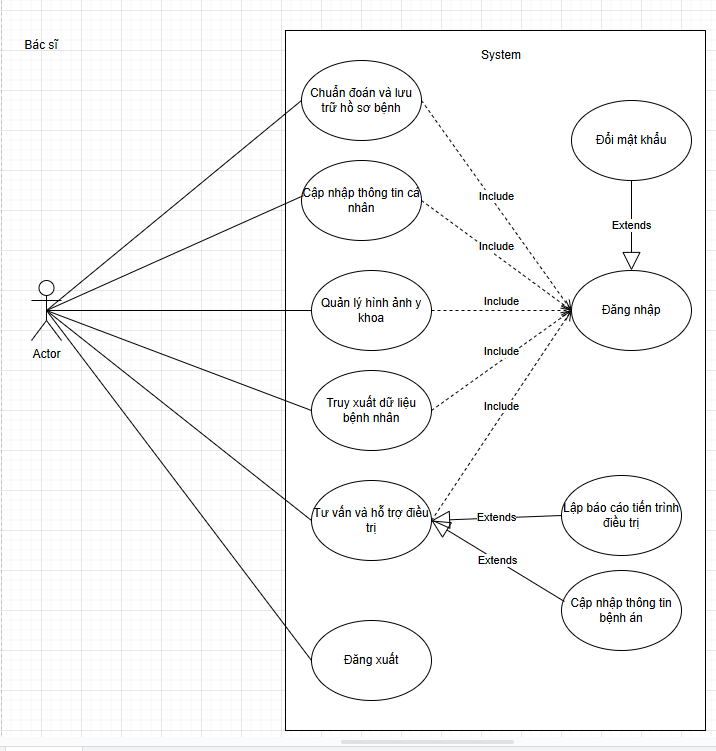
* **Giao diện trực quan:** Sử dụng các biểu tượng, màu sắc rõ ràng để hướng dẫn người dùng.
* **Ngôn ngữ đơn giản:** Tránh sử dụng thuật ngữ chuyên ngành, sử dụng ngôn ngữ dễ hiểu.
* **Kiểm tra tính hợp lệ dữ liệu:** Đảm bảo dữ liệu nhập vào hợp lệ trước khi lưu trữ.
* **Bảo mật dữ liệu:** Mã hóa dữ liệu người dùng để đảm bảo an toàn thông tin.

**Các tính năng nâng cao cho biểu mẫu:**

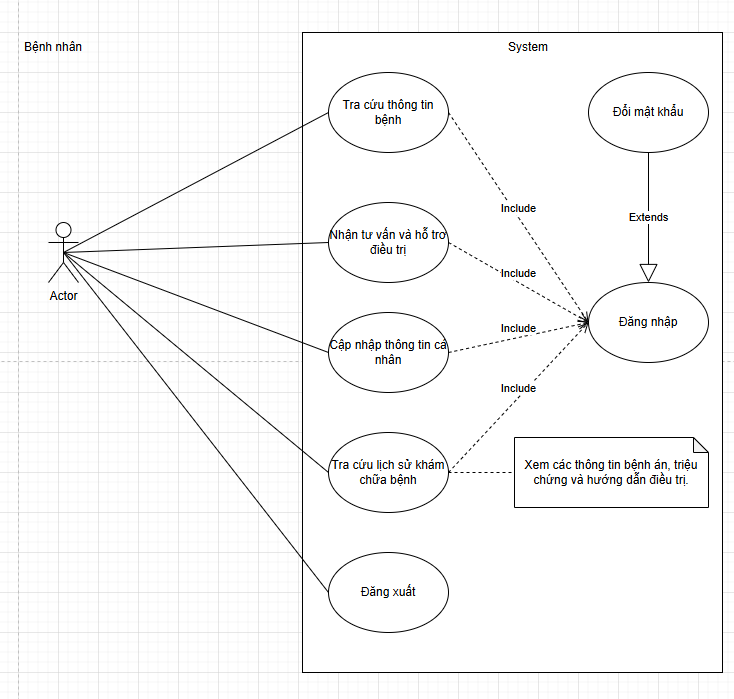
* **Nhắc nhở:** Hệ thống gửi thông báo nhắc nhở người dùng đến khám bệnh định kỳ.
* **Tích hợp lịch:** Tích hợp lịch để người dùng quản lý lịch hẹn khám bệnh.
* **Chia sẻ kết quả:** Cho phép người dùng chia sẻ kết quả chẩn đoán với bác sĩ gia đình hoặc người thân.

**CHƯƠNG Ⅱ: MÔ HÌNH USECASE**

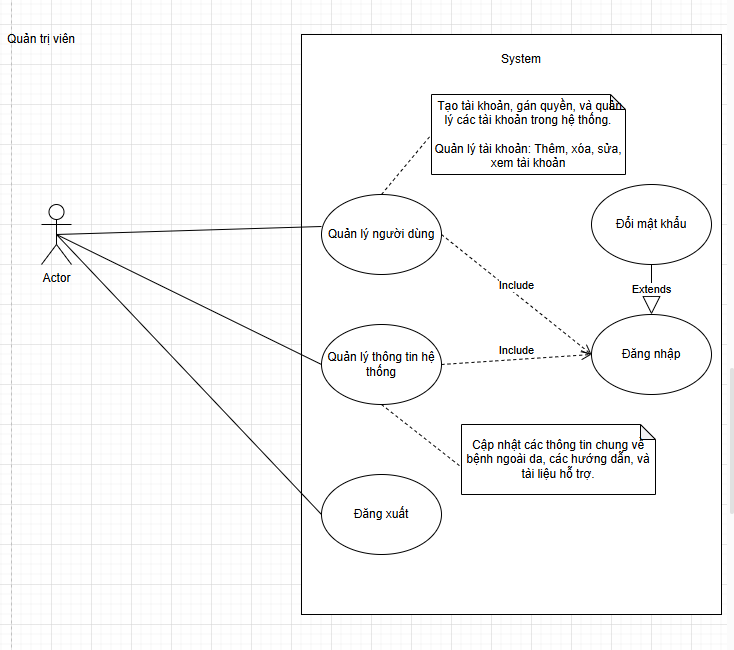
* 1. **Mô hình UseCase**
* **Bác sĩ**



* **Bệnh nhân**



* **Quản trị viên**



* 1. **Danh sách các Actor**
* **Bác sĩ**: Người sử dụng hệ thống để chẩn đoán, lưu trữ hồ sơ và theo dõi tiến trình điều trị cho bệnh nhân.
* **Bệnh nhân**: Người sử dụng hệ thống để theo dõi thông tin bệnh án, triệu chứng và nhận tư vấn từ bác sĩ.
* Quản trị viên**: Người quản lý hệ thống, bảo trì dữ liệu và hỗ trợ các tác nhân khác khi cần thiết.**

## 2.3 Danh Sách Các Use Case

* **UC01**: Đăng nhập
* **UC02**: Đăng xuất
* **UC03**: Đổi mật khẩu
* **UC04**: Cập nhập thông tin bệnh nhân
* **UC05**: Chẩn đoán và lưu trữ hồ sơ bệnh
* **UC06**: Quản lý hình ảnh y khoa
* **UC07**: Truy xuất dữ liệu bệnh nhân
* **UC08**: Tư vấn và hỗ trợ điều trị
* **UC09**: Lập báo cáo tiến trình điều trị
* **UC10**: Tra cứu thông tin bệnh
* **UC11**: Nhận tư vấn và hỗ trợ điều trị
* **UC12**: Tra cứu lịch sử khám chữa bệnh
* **UC13**: Quản lý thông tin hệ thống
* **UC14**: Quản lý người dùng
* **UC15**: Cập nhập thông tin cá nhân

## 2.4 Đặt tả UseCase

### 2.4.1. Đặt tả UseCase “Đăng Nhập”

#### Use Case UC01: Đăng nhập

* **Tên Use Case**: Đăng nhập
* **Mô tả**: Cho phép người dùng đăng nhập vào hệ thống.
* **Actor**: Bác sĩ, Bệnh nhân, Quản trị viên
* **Ưu tiên**: Cao
* **Kích hoạt**: Người dùng mở trang đăng nhập và nhập thông tin.
* **Tiền điều kiện**: Người dùng đã đăng ký tài khoản trong hệ thống.
* **Hậu điều kiện**: Người dùng truy cập thành công vào hệ thống và có quyền sử dụng các chức năng theo vai trò.
* **Luồng cơ bản**:
  + Người dùng truy cập trang đăng nhập.
  + Hệ thống hiển thị form đăng nhập.
  + Người dùng nhập tên đăng nhập và mật khẩu.
  + Hệ thống xác thực thông tin.
  + Nếu thông tin chính xác, người dùng được chuyển đến trang chính.
* **Luồng thay thế**:
  + Người dùng có thể yêu cầu đặt lại mật khẩu nếu quên mật khẩu.
* **Luồng ngoại lệ**:
  + Nếu thông tin không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.

### 2.4.2. Đặt tả UseCase “Đăng Xuất”

#### Use Case UC02: Đăng xuất

* **Tên Use Case**: Đăng xuất
* **Mô tả**: Cho phép người dùng đăng xuất khỏi hệ thống.
* **Actor**: Bác sĩ, Bệnh nhân, Quản trị viên
* **Ưu tiên**: Trung bình
* **Kích hoạt**: Người dùng chọn chức năng đăng xuất.
* **Tiền điều kiện**: Người dùng đã đăng nhập vào hệ thống.
* **Hậu điều kiện**: Phiên làm việc của người dùng kết thúc và họ quay lại trang đăng nhập.
* **Luồng cơ bản**:
  1. Người dùng chọn tùy chọn đăng xuất từ giao diện chính.
  2. Hệ thống kết thúc phiên đăng nhập và chuyển hướng về trang đăng nhập.

### 2.4.3. Đặt tả UseCase “Đổi Mật Khẩu”

#### Use Case UC03: Đổi mật khẩu

* **Tên Use Case**: Đổi mật khẩu
* **Mô tả**: Cho phép người dùng thay đổi mật khẩu của họ.
* **Actor**: Bác sĩ, Bệnh nhân, Quản trị viên
* **Ưu tiên**: Trung bình
* **Kích hoạt**: Người dùng chọn chức năng đổi mật khẩu.
* **Tiền điều kiện**: Người dùng đã đăng nhập vào hệ thống.
* **Hậu điều kiện**: Mật khẩu mới của người dùng được lưu trong hệ thống.
* **Luồng cơ bản**:
  + Người dùng truy cập chức năng đổi mật khẩu.
  + Hệ thống yêu cầu người dùng nhập mật khẩu cũ và mật khẩu mới.
  + Người dùng nhập thông tin và xác nhận.
  + Hệ thống kiểm tra tính hợp lệ của mật khẩu mới và cập nhật mật khẩu.
* **Luồng ngoại lệ**:
  + Nếu mật khẩu cũ không chính xác, hệ thống sẽ hiển thị thông báo lỗi.

### 2.4.4. Đặt tả UseCase “Cập Nhật Thông Tin Bệnh Nhân”

#### Use Case UC04: Cập nhật thông tin bệnh nhân

* **Tên Use Case**: Cập nhật thông tin bệnh nhân
* **Mô tả**: Cho phép Bác sĩ hoặc Quản trị viên cập nhật thông tin của bệnh nhân trong hệ thống.
* **Actor**: Bác sĩ, Quản trị viên
* **Ưu tiên**: Cao
* **Kích hoạt**: Người dùng chọn chức năng cập nhật thông tin bệnh nhân.
* **Tiền điều kiện**: Bệnh nhân đã có hồ sơ trong hệ thống.
* **Hậu điều kiện**: Thông tin bệnh nhân được cập nhật và lưu trong hệ thống.
* **Luồng cơ bản**:
  + Bác sĩ hoặc Quản trị viên truy cập hồ sơ bệnh nhân.
  + Hệ thống hiển thị thông tin hiện tại của bệnh nhân.
  + Người dùng cập nhật các thông tin cần thiết.
  + Hệ thống lưu các thay đổi.
* **Luồng ngoại lệ**:
  + Nếu thông tin nhập không hợp lệ, hệ thống hiển thị thông báo lỗi.

### 2.4.5. Đặt tả UseCase “Chẩn đoán và lưu trữ hồ sơ bệnh”

#### Use Case UC05: Chẩn đoán và lưu trữ hồ sơ bệnh

* **Tên Use Case**: Chẩn đoán và lưu trữ hồ sơ bệnh
* **Mô tả**: Cho phép Bác sĩ chẩn đoán bệnh và lưu trữ hồ sơ của bệnh nhân.
* **Actor**: Bác sĩ
* **Ưu tiên**: Cao
* **Kích hoạt**: Bác sĩ muốn lưu lại chẩn đoán của bệnh nhân.
* **Tiền điều kiện**:
  + Bác sĩ đã đăng nhập.
  + Bệnh nhân có hồ sơ trong hệ thống.
* **Hậu điều kiện**: Hồ sơ chẩn đoán được lưu trong hệ thống và sẵn sàng để truy xuất.
* **Luồng cơ bản**:
  + Bác sĩ truy cập hồ sơ của bệnh nhân.
  + Bác sĩ nhập các thông tin chẩn đoán và điều trị.
  + Hệ thống lưu trữ thông tin chẩn đoán.
* **Luồng thay thế**:
  + Bác sĩ có thể cập nhật chẩn đoán nếu có thông tin mới.

### 2.4.6. Đặt tả UseCase ****“Quản Lý Hình Ảnh Y Khoa”****

#### Use Case: UC06 - Quản lý hình ảnh y khoa

* **Use Case Name**: Quản lý hình ảnh y khoa
* **Description**: Cho phép Bác sĩ quản lý (thêm, xem, xóa) hình ảnh y khoa của bệnh nhân.
* **Actor(s)**: Bác sĩ
* **Priority**: Trung bình
* **Trigger**: Bác sĩ cần lưu trữ hoặc xem hình ảnh y khoa của bệnh nhân.
* **Pre-Condition(s)**: Bệnh nhân có hồ sơ trong hệ thống.
* **Post-Condition(s)**: Hình ảnh y khoa của bệnh nhân được lưu trữ và sẵn sàng để truy xuất.
* **Basic Flow**:
  + Bác sĩ truy cập phần hình ảnh y khoa của bệnh nhân.
  + Bác sĩ tải lên hình ảnh hoặc xem hình ảnh đã có.
  + Hệ thống lưu trữ hình ảnh mới hoặc hiển thị hình ảnh đã có.
* **Alternative Flow**:
  + Bác sĩ có thể xóa hình ảnh không còn cần thiết.

### 

### 2.4.7. Đặt tả UseCase “Truy Xuất Dữ Liệu Bệnh Nhân”

#### Use Case: UC07 - Truy xuất dữ liệu bệnh nhân

* **Use Case Name**: Truy xuất dữ liệu bệnh nhân
* **Description**: Cho phép Bác sĩ hoặc Quản trị viên truy cập và xem dữ liệu của bệnh nhân.
* **Actor(s)**: Bác sĩ, Quản trị viên
* **Priority**: Cao
* **Trigger**: Bác sĩ hoặc Quản trị viên cần xem thông tin bệnh nhân.
* **Pre-Condition(s)**: Bệnh nhân có hồ sơ trong hệ thống.
* **Post-Condition(s)**: Thông tin bệnh nhân được hiển thị.
* **Basic Flow**:
  1. Người dùng truy cập hồ sơ bệnh nhân.
  2. Hệ thống hiển thị thông tin chi tiết của bệnh nhân.

### 2.4.8. Đặt tả UseCase “Tư Vấn Và Hỗ Trợ Điều Trị”

#### Use Case: UC08 - Tư vấn và hỗ trợ điều trị

* **Use Case Name**: Tư vấn và hỗ trợ điều trị
* **Description**: Cho phép Bác sĩ tư vấn và hỗ trợ điều trị cho bệnh nhân.
* **Actor(s)**: Bác sĩ
* **Priority**: Cao
* **Trigger**: Bác sĩ nhận được yêu cầu tư vấn từ bệnh nhân.
* **Pre-Condition(s)**:
  + Bác sĩ và bệnh nhân đã đăng nhập.
  + Bệnh nhân đã gửi yêu cầu tư vấn.
* **Post-Condition(s)**: Bệnh nhân nhận được tư vấn từ bác sĩ.
* **Basic Flow**:
  + Bác sĩ xem yêu cầu tư vấn của bệnh nhân.
  + Bác sĩ trả lời câu hỏi hoặc đưa ra lời khuyên.
  + Hệ thống gửi thông tin tư vấn đến bệnh nhân.
* **Alternative Flow**:
  + Nếu có câu hỏi bổ sung, bệnh nhân có thể phản hồi trong cùng phiên tư vấn.

### 2.4.9. Đặt tả UseCase “Lập Báo Cáo Tiến Trình Điều Trị”

#### Use Case: UC09 - Lập báo cáo tiến trình điều trị

* **Use Case Name**: Lập báo cáo tiến trình điều trị
* **Description**: Cho phép Bác sĩ tạo báo cáo về tiến trình điều trị của bệnh nhân.
* **Actor(s)**: Bác sĩ
* **Priority**: Cao
* **Trigger**: Bác sĩ muốn lưu lại tiến trình điều trị của bệnh nhân.
* **Pre-Condition(s)**: Bệnh nhân có hồ sơ trong hệ thống.
* **Post-Condition(s)**: Báo cáo tiến trình điều trị được lưu trong hồ sơ bệnh nhân.
* **Basic Flow**:
  1. Bác sĩ truy cập hồ sơ bệnh nhân.
  2. Bác sĩ nhập chi tiết tiến trình điều trị.
  3. Hệ thống lưu thông tin vào hồ sơ bệnh nhân.

### 2.4.10. Đặt tả UseCase “Tra Cứu Thông Tin Bệnh”

#### Use Case: UC10 - Tra cứu thông tin bệnh

* **Use Case Name**: Tra cứu thông tin bệnh
* **Description**: Cho phép người dùng tra cứu thông tin về các loại bệnh.
* **Actor(s)**: Bác sĩ, Bệnh nhân
* **Priority**: Trung bình
* **Trigger**: Người dùng muốn tìm hiểu về một loại bệnh cụ thể.
* **Pre-Condition(s)**: Người dùng đã đăng nhập.
* **Post-Condition(s)**: Thông tin bệnh được hiển thị cho người dùng.
* **Basic Flow**:
  + Người dùng nhập từ khóa tìm kiếm liên quan đến bệnh.
  + Hệ thống tìm kiếm và hiển thị các kết quả phù hợp.
* **Exception Flow**:
  + Nếu không có kết quả phù hợp, hệ thống thông báo cho người dùng.

### 2.4.11. Đặt tả UseCase “Nhận Tư Vấn Và Hỗ Trợ Điều Trị”

**Use Case Name**: Nhận tư vấn và hỗ trợ điều trị

* **Description**: Cho phép Bệnh nhân yêu cầu tư vấn và nhận hỗ trợ điều trị từ Bác sĩ.
* **Actor(s)**: Bệnh nhân
* **Priority**: Cao
* **Trigger**: Bệnh nhân cần tư vấn hoặc hỗ trợ điều trị.
* **Pre-Condition(s)**: Bệnh nhân đã đăng nhập và có hồ sơ trong hệ thống.
* **Post-Condition(s)**: Yêu cầu tư vấn của bệnh nhân được gửi cho Bác sĩ.
* **Basic Flow**:
  1. Bệnh nhân truy cập phần tư vấn và hỗ trợ.
  2. Bệnh nhân gửi yêu cầu tư vấn.
  3. Hệ thống thông báo yêu cầu tư vấn đã được gửi đi.
  4. Hiển thị kết quả

### 2.4.12. Đặt tả UseCase “Tra Cứu Lịch Sử Khám Chữa Bệnh”

#### Use Case: UC12 - Tra cứu lịch sử khám chữa bệnh

* **Use Case Name**: Tra cứu lịch sử khám chữa bệnh
* **Description**: Cho phép Bệnh nhân tra cứu lịch sử khám chữa bệnh của mình.
* **Actor(s)**: Bệnh nhân
* **Priority**: Trung bình
* **Trigger**: Bệnh nhân muốn xem lại lịch sử khám chữa bệnh.
* **Pre-Condition(s)**: Bệnh nhân đã đăng nhập và có lịch sử khám chữa bệnh trong hệ thống.
* **Post-Condition(s)**: Hệ thống hiển thị lịch sử khám chữa bệnh của bệnh nhân.
* **Basic Flow**:
  1. Bệnh nhân truy cập phần lịch sử khám chữa bệnh.
  2. Hệ thống hiển thị lịch sử khám chữa bệnh cho bệnh nhân.

### 2.4.13. Đặt tả UseCase “Quản Lý Thông Tin Hệ Thống”

#### Use Case: UC13 - Quản lý thông tin hệ thống

* **Use Case Name**: Quản lý thông tin hệ thống
* **Description**: Cho phép Quản trị viên cập nhật và quản lý thông tin của hệ thống.
* **Actor(s)**: Quản trị viên
* **Priority**: Cao
* **Trigger**: Quản trị viên cần cập nhật thông tin hệ thống.
* **Pre-Condition(s)**: Quản trị viên đã đăng nhập.
* **Post-Condition(s)**: Thông tin hệ thống được cập nhật.
* **Basic Flow**:
  1. Quản trị viên truy cập chức năng quản lý thông tin hệ thống.
  2. Quản trị viên thực hiện thay đổi và cập nhật thông tin.
  3. Hệ thống lưu các thay đổi.

### 2.4.14. Đặc tả UseCase “Quản Lý Người Dùng”

**Use Case: UC14 - Quản lý người dùng**

* **Use Case Name**: Quản lý người dùng
* **Description**: Cho phép Quản trị viên quản lý (thêm, xóa, sửa) thông tin người dùng.
* **Actor(s)**: Quản trị viên
* **Priority**: Cao
* **Trigger**: Quản trị viên cần quản lý người dùng.
* **Pre-Condition(s)**: Quản trị viên đã đăng nhập.
* **Post-Condition(s)**: Thông tin người dùng được cập nhật.
* **Basic Flow**:
  1. Quản trị viên truy cập chức năng quản lý người dùng.
  2. Quản trị viên thực hiện các thao tác quản lý cần thiết.
  3. Hệ thống lưu các thay đổi.

### 2.4.15. Đặc tả UsesCase “Cập Nhật Thông Tin Cá Nhân”

#### Use Case: UC15 - Cập nhật thông tin cá nhân

* **Use Case Name**: Cập nhật thông tin cá nhân
* **Description**: Cho phép Bác sĩ, Bệnh nhân và Quản trị viên cập nhật thông tin cá nhân của mình.
* **Actor(s)**: Bác sĩ, Bệnh nhân, Quản trị viên
* **Priority**: Thấp
* **Trigger**: Người dùng cần cập nhật thông tin cá nhân.
* **Pre-Condition(s)**: Người dùng đã đăng nhập.
* **Post-Condition(s)**: Thông tin cá nhân được cập nhật.
* **Basic Flow**:
  1. Người dùng truy cập phần thông tin cá nhân.
  2. Người dùng thực hiện các thay đổi cần thiết.
  3. Hệ thống lưu các thay đổi.

**CHƯƠNG III: PHÂN TÍCH**

Bottom of Form

## 3.1 Sơ đồ lớp

### 3.1.1 Sơ đồ lớp ( mức phân tích )

Sơ đồ lớp mô tả các thành phần chính trong hệ thống quản lý bệnh ngoài da, bao gồm các lớp chính như Người dùng, Bệnh, Hồ sơ bệnh án, và Báo cáo. Các lớp này thể hiện các thuộc tính và phương thức cần thiết để hệ thống hoạt động đúng theo yêu cầu.

**Các lớp chính và thuộc tính**

* **Người dùng**: Định nghĩa các loại người dùng của hệ thống, bao gồm Bác sĩ, Nhà nghiên cứu, và Quản trị viên.
* **Thuộc tính:** ID, họ tên, ngày sinh, giới tính, vai trò (bác sĩ, nhà nghiên cứu, quản trị viên), thông tin liên hệ, mật khẩu.
* **Phương thức:** đăng nhập, đăng ký, cập nhật thông tin cá nhân.
* **Bệnh**: Lớp chứa thông tin về các loại bệnh, triệu chứng, nguyên nhân, và phương pháp điều trị.
* **Hồ sơ bệnh án**: Quản lý chẩn đoán và các thông tin liên quan đến bệnh của bệnh nhân.

Báo cáo: Lập và lưu trữ các báo cáo tiến trình điều trị của bệnh nhân, các thống kê, và thông tin hỗ trợ.

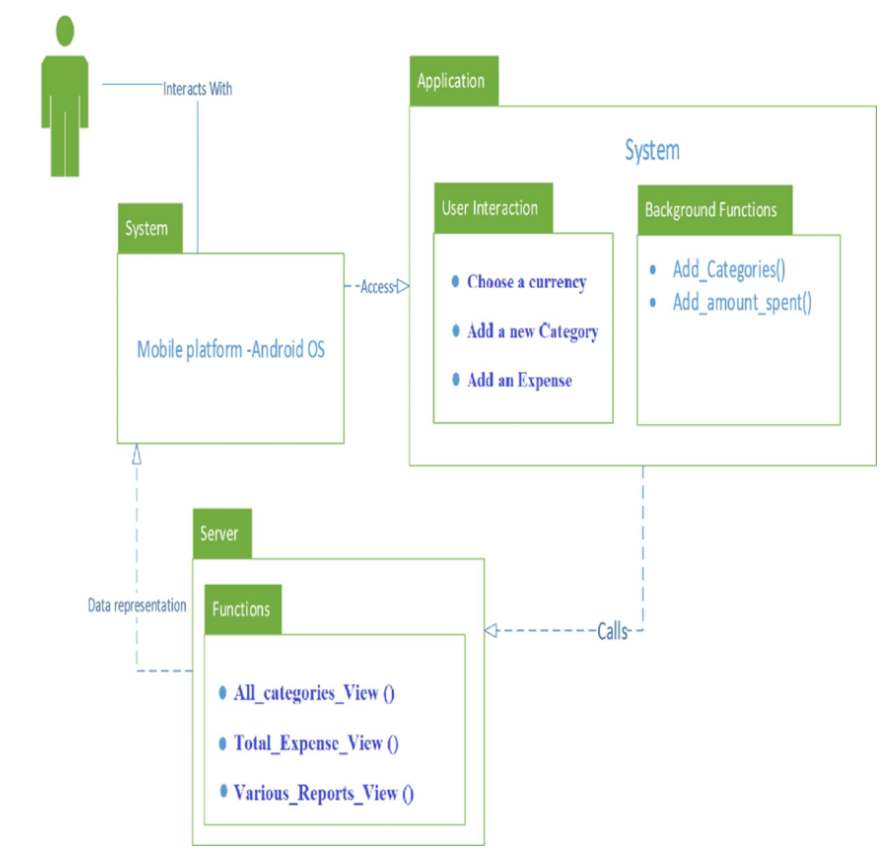
**Các lớp phụ và mối quan hệ**

* **Hình ảnh:**
* **Thuộc tính:** ID hình ảnh, đường dẫn, mô tả (hình ảnh bệnh nhân, hình ảnh minh họa bệnh).
* **Mối quan hệ:** Một bệnh có thể có nhiều hình ảnh minh họa. Một hồ sơ bệnh án có thể có nhiều hình ảnh của bệnh nhân.
* **Thuốc:**
* **Thuộc tính:** ID thuốc, tên thuốc, công dụng, liều dùng, tác dụng phụ.
* **Mối quan hệ:** Một bệnh có thể được điều trị bằng nhiều loại thuốc. Một hồ sơ bệnh án có thể kê nhiều loại thuốc.
* **Xét nghiệm:**
* **Thuộc tính:** ID xét nghiệm, tên xét nghiệm, kết quả.
* **Mối quan hệ:** Một hồ sơ bệnh án có thể bao gồm nhiều kết quả xét nghiệm.
* **Lịch:**
* **Thuộc tính:** ID lịch, ngày giờ khám, lý do khám.
* **Mối quan hệ:** Một hồ sơ bệnh án có thể có nhiều lịch khám.

**Mối quan hệ giữa các lớp**

* **Một-nhiều:** Một bệnh nhân có thể có nhiều hồ sơ bệnh án. Một bác sĩ có thể khám cho nhiều bệnh nhân. Một bệnh có thể có nhiều phương pháp điều trị.
* **Nhiều-nhiều:** Một hồ sơ bệnh án có thể liên quan đến nhiều bệnh, nhiều thuốc, nhiều xét nghiệm.

**Sơ đồ lớp sơ bộ**



**Mở Rộng và Cải Tiến**

* **Phân quyền:** Thêm lớp Quyền để quản lý quyền truy cập của từng loại người dùng vào các chức năng của hệ thống.
* **Thông báo:** Thêm lớp Thông báo để gửi các thông báo cho người dùng (ví dụ: nhắc nhở khám bệnh, kết quả xét nghiệm).
* **Hệ thống chuyên gia:** Tích hợp hệ thống chuyên gia để hỗ trợ chẩn đoán bệnh.
* **Học máy:** Áp dụng các thuật toán học máy để phân tích dữ liệu, dự đoán bệnh và đề xuất phương pháp điều trị.

### 3.1.2 Danh sách các lớp đối tượng và quan hệ

|  |  |  |
| --- | --- | --- |
| Lớp | Quan hệ | Mô tả |
| Người dùng | Tạo báo cáo | Người dùng tạo và quản lý báo cáo về tiến trình điều trị của bệnh nhân. |
| Người dùng | Chẩn đoán bệnh | Người dùng chẩn đoán bệnh dựa trên triệu chứng của bệnh nhân. |
| Bệnh | Gắn với hồ sơ bệnh án | Bệnh được gắn với hồ sơ bệnh án của bệnh nhân để lưu trữ thông tin chẩn đoán. |
| Hồ sơ bệnh án | Chứa bệnh | Hồ sơ bệnh án chứa các thông tin về bệnh đã được chẩn đoán cho bệnh nhân. |
| Hồ sơ bệnh án | Liên kết báo cáo | Hồ sơ bệnh án được liên kết với các báo cáo tiến trình điều trị của bệnh nhân. |
| Hình ảnh y khoa | Gắn với hồ sơ bệnh án | Hình ảnh y khoa được gắn với hồ sơ bệnh án của bệnh nhân để lưu trữ thông tin bệnh |
| Lịch sử khám chữa bệnh | Liên kết với người dùng | Lịch sử khám chữa bệnh được liên kết với người dùng lưu lại các lần khám trước đó |

**Mở rộng các lớp đối tượng:**

* **Thuốc:** Mở rộng thông tin về thuốc, bao gồm tên thuốc, công dụng, liều dùng, nhà sản xuất.
* **Xét nghiệm:** Thêm lớp Xét nghiệm để quản lý các loại xét nghiệm, kết quả xét nghiệm.
* **Lịch khám:** Thêm lớp Lịch khám để quản lý lịch hẹn khám bệnh.
* **Thông báo:** Thêm lớp Thông báo để quản lý các thông báo gửi đến người dùng (nhắc nhở khám bệnh, kết quả xét nghiệm).
* **Quyền truy cập:** Thêm lớp Quyền để quản lý quyền truy cập của từng loại người dùng vào các chức năng của hệ thống.

**Chi tiết hóa các mối quan hệ:**

* **Một bệnh nhân có thể có nhiều hồ sơ bệnh án:** Nếu bệnh nhân khám nhiều lần.
* **Một hồ sơ bệnh án có thể liên kết với nhiều bệnh:** Nếu bệnh nhân mắc nhiều bệnh.
* **Một hồ sơ bệnh án có thể liên kết với nhiều hình ảnh y khoa, nhiều loại thuốc, nhiều kết quả xét nghiệm.**
* **Một bác sĩ có thể chẩn đoán cho nhiều bệnh nhân.**
* **Một thuốc có thể được kê đơn cho nhiều bệnh.**

**Đề xuất các lớp và quan hệ mới:**

* **Lớp Hệ thống chuyên gia:** Để hỗ trợ chẩn đoán bệnh dựa trên các quy tắc và kiến thức chuyên môn.
* **Lớp Mô hình máy học:** Để lưu trữ các mô hình máy học được sử dụng để phân tích dữ liệu hình ảnh và dự đoán bệnh.
* **Lớp Báo cáo thống kê:** Để tạo các báo cáo thống kê về tình hình bệnh tật, hiệu quả điều trị.

**Bảng tổng hợp các lớp và quan hệ chi tiết hơn:**

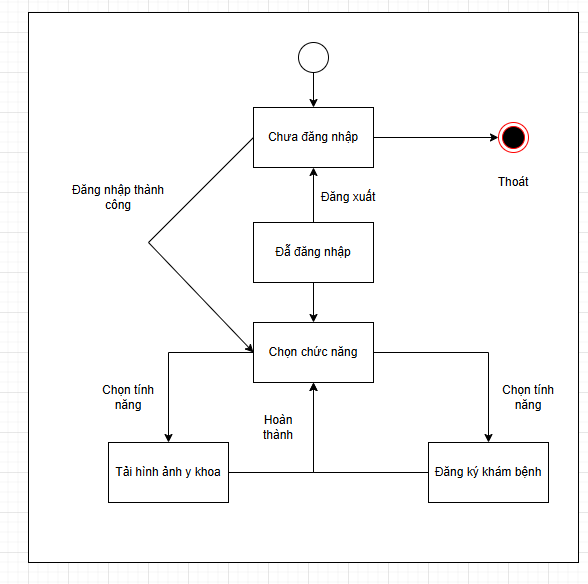
|  |  |  |  |
| --- | --- | --- | --- |
| Lớp | Thuộc tính chính | Quan hệ với các lớp khác | Mô tả |
| Người dùng | ID, tên, vai trò, thông tin liên hệ | Tạo báo cáo, chẩn đoán bệnh, có nhiều hồ sơ bệnh án | Đại diện cho người sử dụng hệ thống |
| Bệnh | ID, tên bệnh, triệu chứng, nguyên nhân, phương pháp điều trị, hình ảnh | Gắn với hồ sơ bệnh án, liên kết với thuốc | Mô tả chi tiết về một loại bệnh |
| Hồ sơ bệnh án | ID, bệnh nhân, bác sĩ, ngày khám, chẩn đoán, kết quả xét nghiệm, đơn thuốc | Chứa bệnh, liên kết báo cáo, gắn với hình ảnh, liên kết lịch khám | Lưu trữ thông tin chi tiết về quá trình điều trị của một bệnh nhân |
| Hình ảnh | ID, đường dẫn, mô tả | Gắn với hồ sơ bệnh án | Lưu trữ hình ảnh y khoa của bệnh nhân |
| Thuốc | ID, tên thuốc, công dụng, liều dùng | Liên kết với hồ sơ bệnh án | Mô tả thông tin về thuốc |
| Xét nghiệm | ID, tên thuốc, công dụng, liều dùng | Liên kết với hồ sơ bệnh án | Lưu trữ kết quả xét nghiệm của bệnh nhân |
| Lịch khám | ID, ngày giờ, lý do | Liên kết với hồ sơ bệnh án | Quản lý lịch hẹn khám bệnh |
| Báo cáo | ID, loại báo cáo, nội dung | Được tạo bởi người dùng, liên kết với hồ sơ bệnh án | Lưu trữ các báo cáo về tiến trình điều trị, thống kê |
| Hệ thống chuyên gia | Quy tắc, kiến thức | Hỗ trợ chẩn đoán bệnh | Cung cấp kiến thức chuyên môn để hỗ trợ chẩn đoán |
| Mô hình máy học | Mô hình | Dự đoán bệnh | Sử dụng các thuật toán học máy để phân tích dữ liệu và dự đoán bệnh |
| Quyền truy cập | Tên quyền, danh sách quyền hạn | Liên kết với người dùng | Quản lý quyền truy cập của từng loại người dùng |

### 3.1.3 Mô tả chi tiết đối tượng

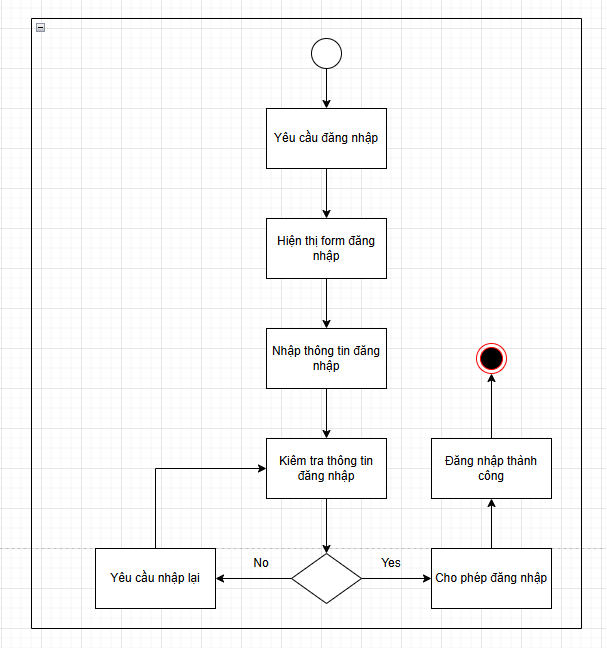
1. **Người dùng**
   * **Thuộc tính**:
     + id: Mã định danh duy nhất cho từng người dùng.
     + name: Tên người dùng
     + date\_of\_birth: Ngày sinh
     + gender: Giới tính
     + userName: Tên tài khoản đăng nhập vào hệ thống.
     + contact\_info: Thông tin liên hệ
     + passwork: Mật khẩu bảo mật của người dùng.
     + Role: Phân loại người dùng (Bác sĩ, Nhà nghiên cứu, Quản trị viên).
     + medical\_history\_id: Lịch sử y khoa (nếu có)
     + created\_at: Ngày tạo
     + updated\_at: Ngày cập nhật
   * **Phương thức**:
     + diagnosisOfDisease: Xác định bệnh dựa trên triệu chứng.
     + createRepo: Lập báo cáo về tiến trình điều trị và thông tin bệnh nhân.
2. **Bệnh**
   * **Thuộc tính**:
     + id: Mã định danh của bệnh.
     + name: Tên của bệnh (ví dụ: Eczema, Psoriasis).
     + symptoms: Các triệu chứng điển hình của bệnh.
     + reasons: Nguyên nhân hoặc các yếu tố gây bệnh.
     + treatments: Các cách thức điều trị bệnh.
     + created\_at: Ngày tạo
     + updated\_at: Ngày cập nhật
   * **Phương thức**:
     + updateIllness: Thêm hoặc chỉnh sửa thông tin về bệnh.
     + viewIllness: Tra cứu chi tiết về bệnh và các triệu chứng.
3. **Hồ sơ bệnh án**
   * **Thuộc tính**:
     + id: Mã định danh cho hồ sơ bệnh án của từng bệnh nhân.
     + patient\_id: ID bệnh nhân
     + disease\_id: ID bệnh
     + doctor\_id: ID bác sĩ phụ trách
     + diagnosis: Kết quả chẩn đoán về bệnh.
     + treatment\_plan: Các biện pháp điều trị áp dụng cho bệnh nhân.
     + note: Các ghi chú bổ sung từ bác sĩ.
     + created\_at: Ngày tạo
     + updated\_at: Ngày cập nhật
   * **Phương thức**:
     + update: Thêm thông tin mới hoặc chỉnh sửa thông tin trong hồ sơ bệnh án.
     + view: Lấy ra hồ sơ bệnh án của bệnh nhân khi cần thiết.
4. **Báo cáo**
   * **Thuộc tính**:
     + id: ID báo cáo (Primary Key)
     + medical\_record\_id: ID hồ sơ bệnh án (liên kết đến bảng medical\_records)
     + report\_date: Ngày lập báo cáo
     + content: Nội dung báo cáo
     + doctor\_id: ID bác sĩ lập báo cáo (liên kết đến bảng users)
     + created\_at: Ngày tạo
     + updated\_at: Ngày cập nhật
   * **Phương thức**:
     + create: Lập báo cáo mới về tiến trình điều trị.
     + view: Truy xuất và xem nội dung báo cáo.
     + update: Cập nhật thông tin trong báo cáo.
5. **Hình ảnh y khoa**
   * **Thuộc tính**:
     + id: ID hình ảnh y khoa
     + medical\_record\_id: ID hồ sơ bệnh án
     + image\_path: Đường dẫn đến file hình ảnh
     + description: Mô tả hình ảnh
     + uploaded\_at: Ngày tải lên
6. **Lịch sử khám chữa bệnh**
   * **Thuộc tính**:
     + id: ID lịch sử khám
     + patient\_id: ID bệnh nhân
     + visit\_date: Ngày khám
     + doctor\_id: ID bác sĩ khám
     + notes: Ghi chú về lần khám chữa bệnh
     + created\_at: Ngày tạo
     + updated\_at: Ngày cập nhật
   * **Phương thức**:
     + update: Cập nhật thông tin

## 3.2 Sơ đồ trạng thái

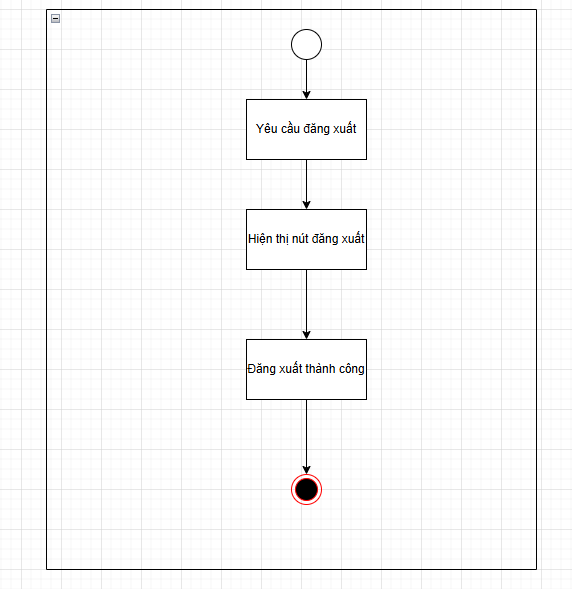
### 3.2.1 Sơ đồ trạng thái tổng quát



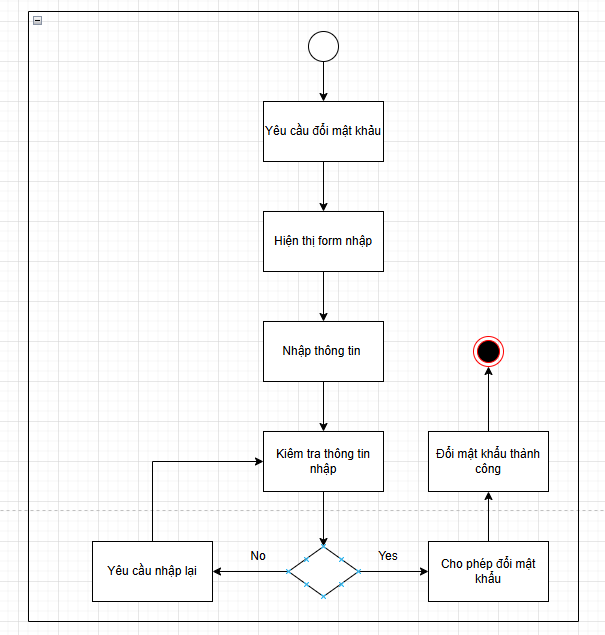
### 3.2.2 Sơ đồ trạng thái đăng nhập



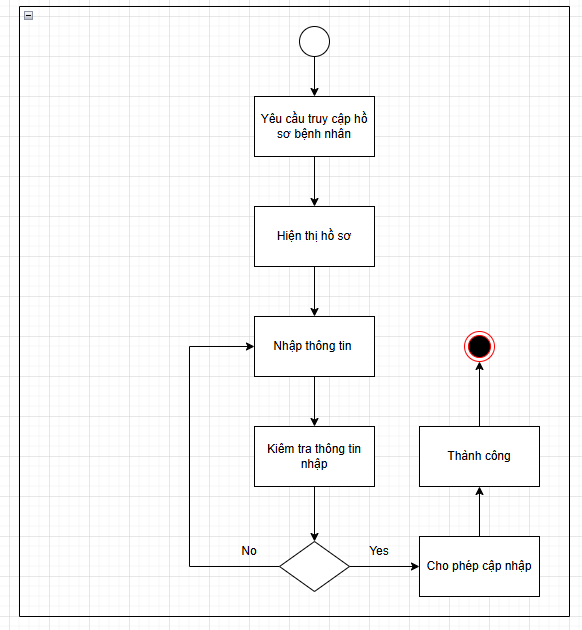
### 3.2.3 Sơ đồ trạng thái đăng xuất



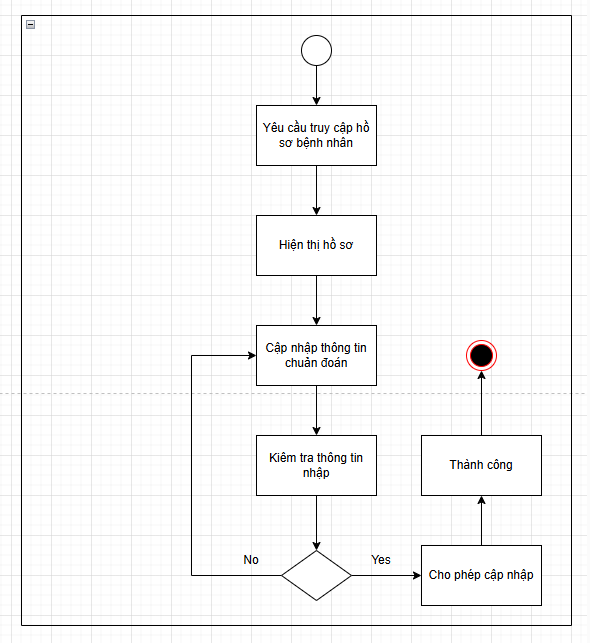
### 3.2.4 Sơ đồ trạng thái đổi mật khẩu



### 3.2.5 Sơ đồ trạng thái cập nhập thông tin



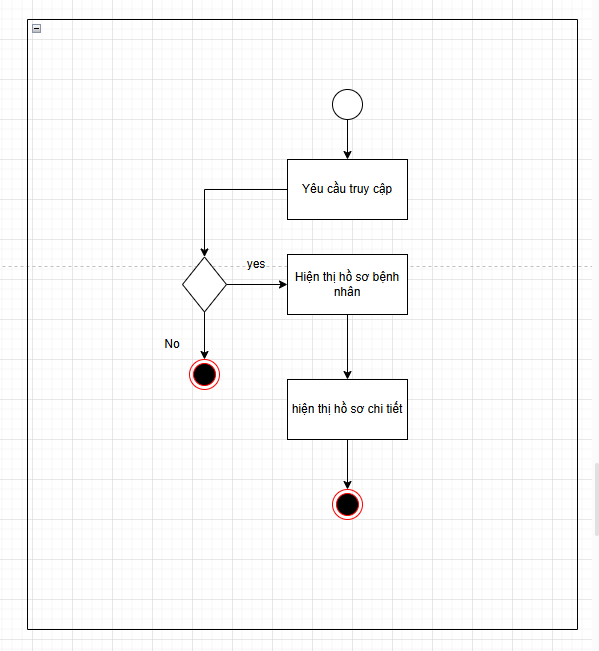
### 3.2.6 Sơ đồ trạng thái chuẩn đoán và lưu trữ hồ sơ



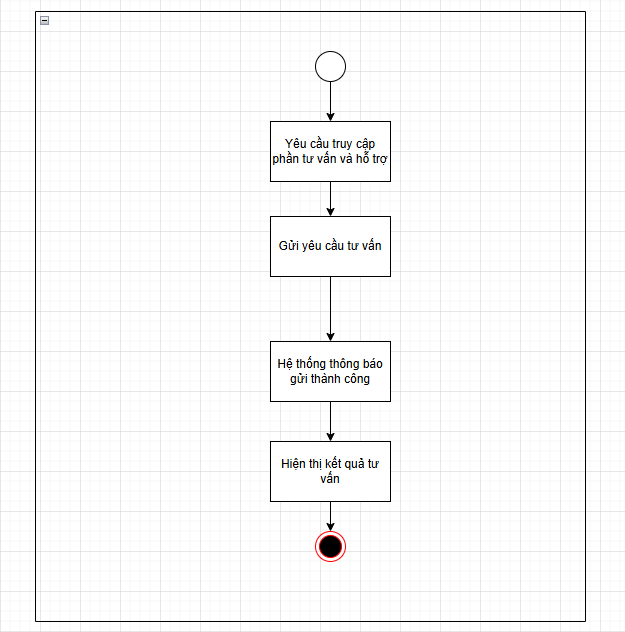
### 3.2.7 Sơ đồ trạng thái quản lý hình ảnh y khoa

#### **https://lh7-rt.googleusercontent.com/docsz/AD_4nXd3QSLTA3sY_aiCghnR0UFPCgmpbUHT7fMikmnCoZfbgad-9bQVe79dYCERP6OaQA0Z-gS0WveIMg6TrbrOSL2TgHsBJNTPwFkCjC5in7EB-pCw4xzWIDTyAlMQBXZLJ6wp-wt_Yw?key=N9cDrIp_0vdv2MD101azs0SG**

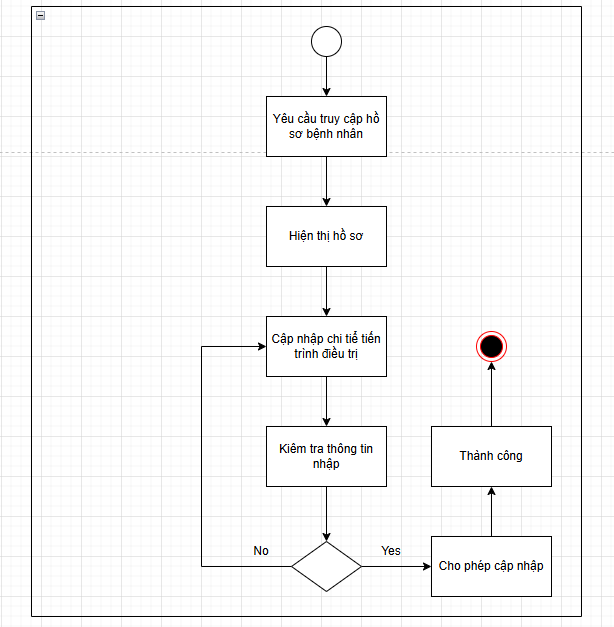
### 3.2.8 Sơ đồ trạng thái truy xuất dữ liệu bệnh nhân



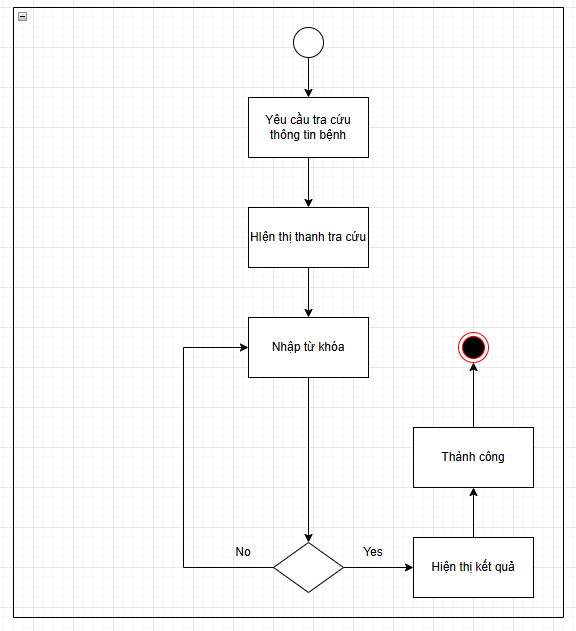
### 3.2.9 Sơ đồ trạng thái nhận tư vấn và hỗ trợ điều trị



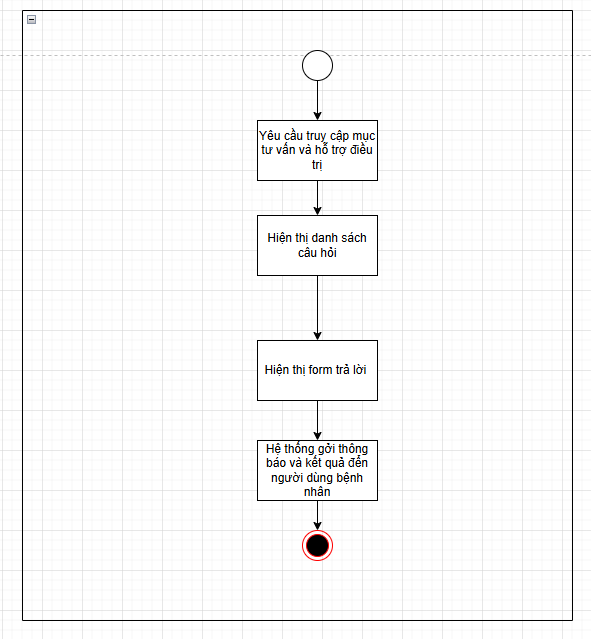
### 3.2.10 Sơ đồ trạng thái lập báo cáo tiến trình điều trị



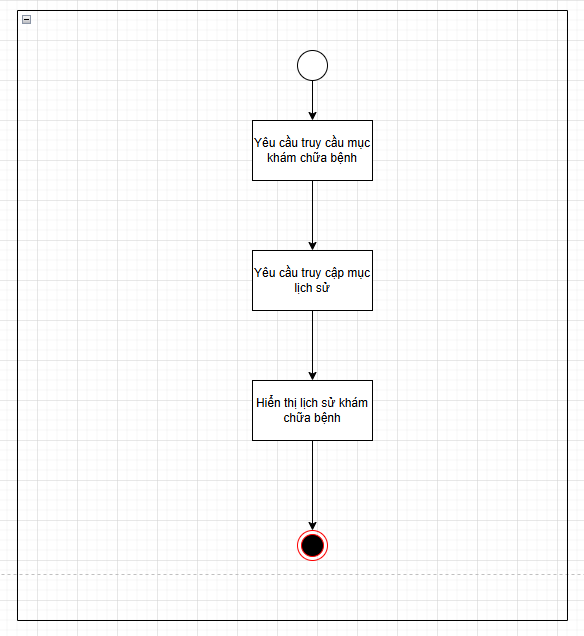
### 3.2.11 Sơ đồ trạng thái tra cứu thông tin bệnh



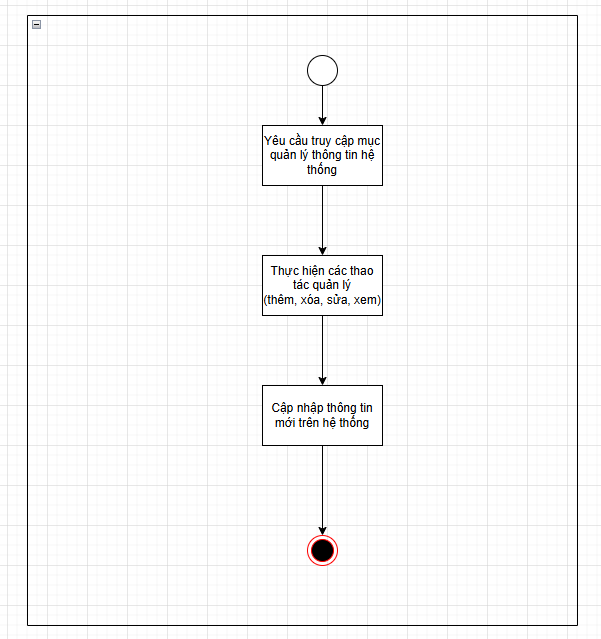
### 3.2.12 Sơ đồ trạng thái tư vấn và hỗ trợ điều trị



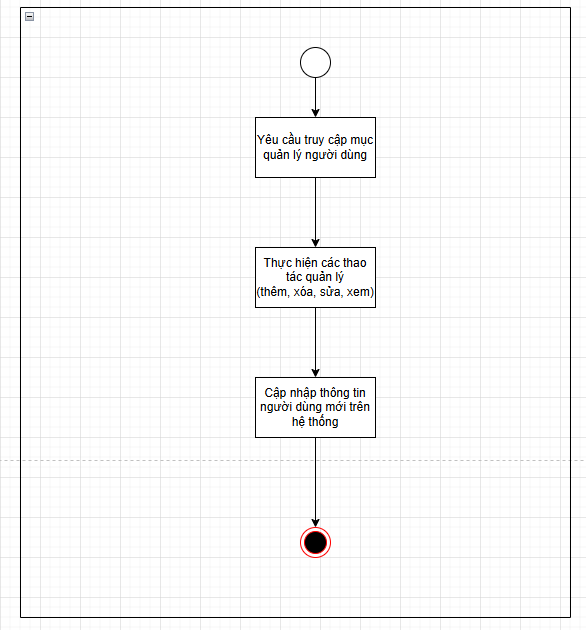
### 3.2.13 Sơ đồ trạng thái tra cứu lịch sử khám chữa bệnh



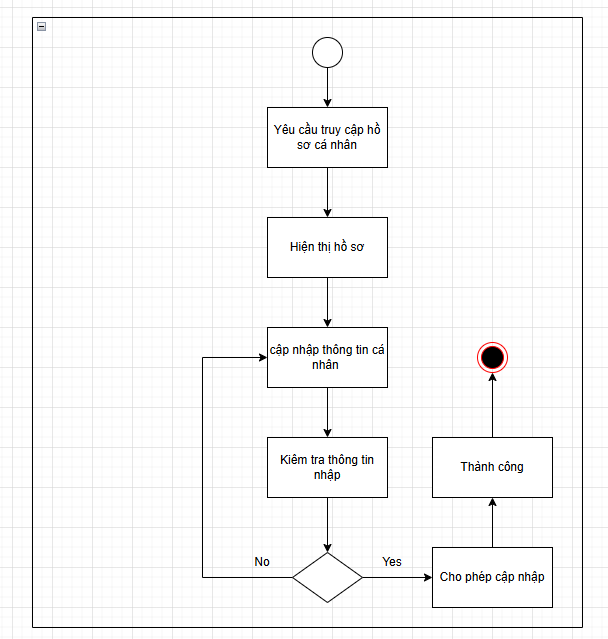
### 3.2.14 Sơ đồ trạng thái quản lý thông tin hệ thống



### 3.2.15 Sơ đồ trạng thái quản lý người dùng

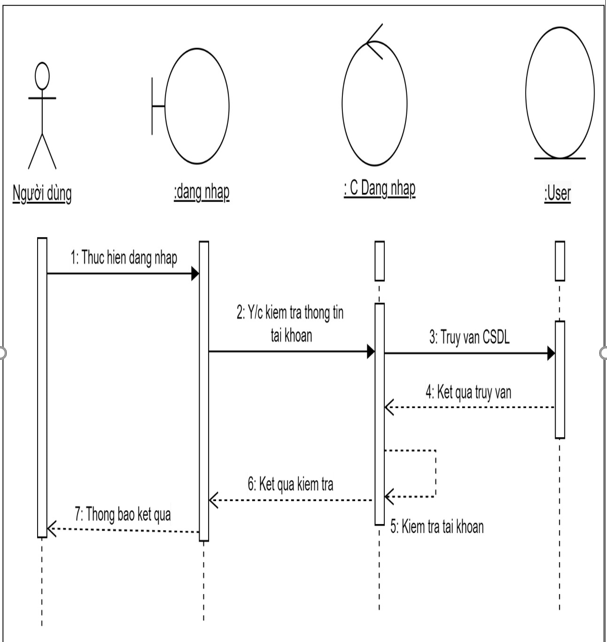


### 3.2.16 Sơ đồ trạng thái cập nhập thông tin cá nhân



## 3.3 Sơ đồ tuần tự

### 3.3.1 Đăng nhập



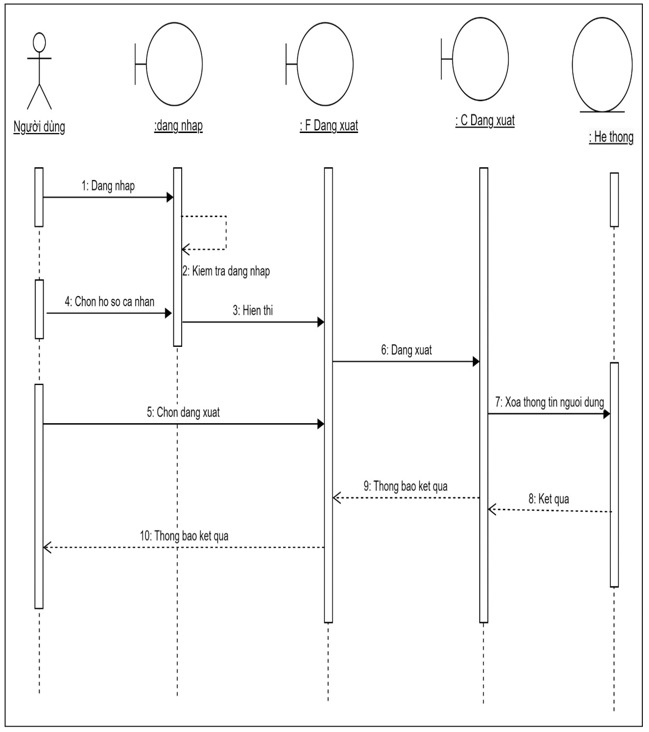
**Các đối tượng tham gia:**

* **Người dùng (User):** Là người trực tiếp sử dụng hệ thống và thực hiện yêu cầu đăng nhập.
* **Thành phần xử lý yêu cầu đăng nhập:** Đảm nhiệm nhận yêu cầu đăng nhập từ người dùng và điều phối các bước kiểm tra tiếp theo.
* **Thành phần C Đăng nhập tài khoản:** Chịu trách nhiệm kiểm tra thông tin đăng nhập trong cơ sở dữ liệu (CSDL) của hệ thống.
* **Cơ sở dữ liệu (CSDL):** Là nơi lưu trữ thông tin tài khoản, được sử dụng để đối chiếu và xác minh thông tin đăng nhập.

**Luồng hoạt động:**

1. **Người dùng** bắt đầu quá trình đăng nhập bằng cách gửi yêu cầu đăng nhập đến hệ thống.
2. **Hệ thống** nhận được yêu cầu và chuyển đến **Thành phần xử lý yêu cầu đăng nhập**.
3. **Thành phần xử lý yêu cầu đăng nhập** tiếp tục gửi yêu cầu kiểm tra thông tin đăng nhập đến **Thành phần C Đăng nhập tài khoản**.
4. **Thành phần C Đăng nhập tài khoản** tiến hành truy vấn **Cơ sở dữ liệu** để kiểm tra xem thông tin đăng nhập (tên đăng nhập, mật khẩu) mà người dùng cung cấp có khớp với thông tin đã lưu trong cơ sở dữ liệu hay không.
5. Sau khi nhận được kết quả từ cơ sở dữ liệu, **Thành phần C Đăng nhập tài khoản** tiến hành so sánh và kiểm tra xem thông tin có chính xác và hợp lệ hay không.
6. Nếu thông tin đăng nhập được xác nhận là đúng, **Thành phần C Đăng nhập tài khoản** sẽ gửi kết quả xác thực về cho hệ thống.
7. Cuối cùng, hệ thống thông báo kết quả cho **người dùng**, cho biết quá trình đăng nhập thành công hoặc thất bại.

### 3.3.2 Đăng xuất



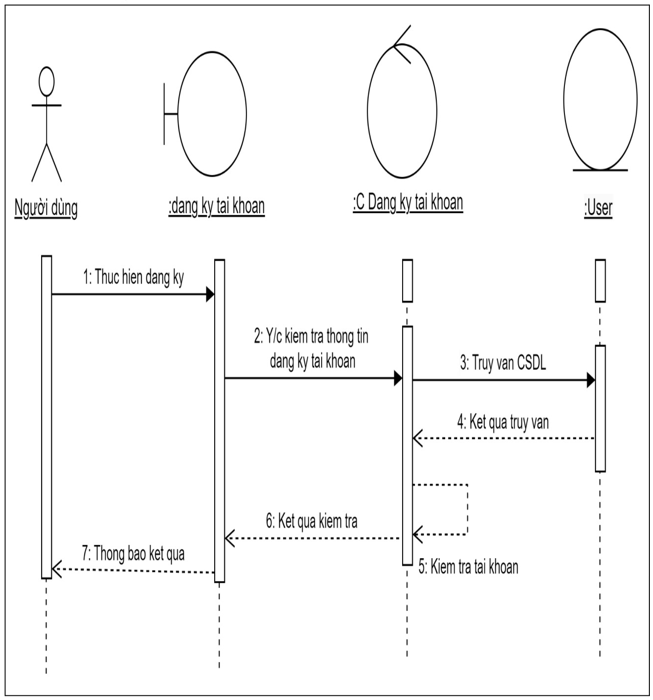
**Các đối tượng tham gia:**

* **Người dùng:** Người trực tiếp sử dụng hệ thống, thực hiện các yêu cầu như đăng nhập, đăng xuất.
* **Thành phần đăng nhập:** Phụ trách xử lý yêu cầu đăng nhập, kiểm tra thông tin đăng nhập của người dùng.
* **Thành phần F Đăng xuất:** Nhận yêu cầu đăng xuất từ người dùng và tiến hành các bước đăng xuất.
* **Thành phần C Đăng xuất:** Chịu trách nhiệm xóa thông tin người dùng khỏi hệ thống sau khi đăng xuất.
* **Hệ thống:** Là cơ sở dữ liệu hoặc hệ thống lưu trữ thông tin người dùng.

**Luồng hoạt động:**

1. **Người dùng** gửi yêu cầu đăng nhập vào hệ thống.
2. **Thành phần đăng nhập** kiểm tra thông tin đăng nhập (tên đăng nhập, mật khẩu) mà người dùng cung cấp.
3. **Nếu thông tin hợp lệ:** Hệ thống cho phép người dùng truy cập vào tài khoản của mình.
4. **Nếu thông tin không hợp lệ:** Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại.
5. **Sau khi đăng nhập thành công:** Người dùng có thể thực hiện các thao tác trên hệ thống.
6. **Khi người dùng muốn đăng xuất:** Họ gửi yêu cầu đến **Thành phần F Đăng xuất**.
7. **Thành phần F Đăng xuất** chuyển yêu cầu đến **Thành phần C Đăng xuất**.
8. **Thành phần C Đăng xuất** tiến hành xóa các thông tin liên quan đến người dùng đó khỏi hệ thống.
9. **Hệ thống** xác nhận việc xóa thông tin và thông báo lại cho **Thành phần C Đăng xuất**.
10. **Thành phần C Đăng xuất** thông báo kết quả đăng xuất thành công đến **Thành phần F Đăng xuất**, sau đó **Thành phần F Đăng xuất** thông báo lại cho **người dùng**.

### 3.3.3 Đăng kí



**Các đối tượng tham gia:**

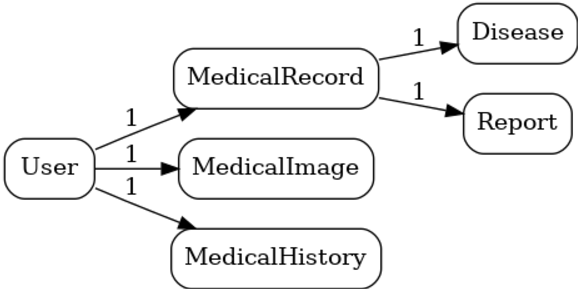
* **Người dùng (User):** Người trực tiếp sử dụng hệ thống, thực hiện yêu cầu đăng ký.
* **Thành phần xử lý yêu cầu đăng ký:** Đảm nhiệm nhận yêu cầu đăng ký từ người dùng và điều phối các bước kiểm tra tiếp theo.
* **Thành phần C Đăng ký tài khoản:** Chịu trách nhiệm kiểm tra thông tin đăng ký trong cơ sở dữ liệu (CSDL) của hệ thống.
* **Cơ sở dữ liệu (CSDL):** Là nơi lưu trữ thông tin tài khoản, được sử dụng để đối chiếu và xác minh thông tin đăng ký.

**Luồng hoạt động:**

1. **Người dùng** bắt đầu quá trình đăng ký bằng cách gửi yêu cầu đăng ký đến hệ thống.
2. **Hệ thống** nhận được yêu cầu và chuyển đến **Thành phần xử lý yêu cầu đăng ký**.
3. **Thành phần xử lý yêu cầu đăng ký** tiếp tục gửi yêu cầu kiểm tra thông tin đăng ký đến **Thành phần C Đăng ký tài khoản**.
4. **Thành phần C Đăng ký tài khoản** tiến hành truy vấn **Cơ sở dữ liệu** để kiểm tra xem thông tin đăng ký (tên đăng nhập, mật khẩu, email,...) mà người dùng cung cấp có trùng lặp với thông tin đã lưu trong cơ sở dữ liệu hay không.
5. Sau khi nhận được kết quả từ cơ sở dữ liệu, **Thành phần C Đăng ký tài khoản** tiến hành so sánh và kiểm tra xem thông tin có hợp lệ và tuân thủ các quy định của hệ thống hay không.
6. Nếu thông tin đăng ký được xác nhận là hợp lệ, **Thành phần C Đăng ký tài khoản** sẽ gửi kết quả xác thực về cho hệ thống.
7. Cuối cùng, hệ thống thông báo kết quả cho **người dùng**, cho biết quá trình đăng ký thành công hoặc thất bại.

# CHƯƠNG IV: THIẾT KẾ DỮ LIỆU LƯU TRỮ

## 4.1 Sơ đồ logic



**Các thực thể và mối quan hệ:**

* **User (Người dùng):** Đại diện cho một bệnh nhân hoặc một người được quản lý thông tin sức khỏe.
* **MedicalRecord (Hồ sơ bệnh án):** Là tập hợp các thông tin liên quan đến sức khỏe của một người dùng, bao gồm bệnh sử, kết quả xét nghiệm, chẩn đoán,...
* **MedicalImage (Hình ảnh y tế):** Là các hình ảnh y tế được sử dụng để chẩn đoán và theo dõi bệnh tình (như X-quang, MRI, CT scan,...).
* **MedicalHistory (Bệnh sử):** Là lịch sử bệnh tật của người dùng, bao gồm các bệnh đã mắc phải, các lần nhập viện, phẫu thuật,...
* **Disease (Bệnh):** Đại diện cho các loại bệnh mà người dùng có thể mắc phải.
* **Report (Báo cáo):** Là các báo cáo y tế được tạo ra từ việc phân tích dữ liệu y tế, chẳng hạn như báo cáo kết quả xét nghiệm, báo cáo sinh thiết,...

**Các mối quan hệ:**

* **User - MedicalRecord (1-1):** Mỗi người dùng chỉ có một hồ sơ bệnh án duy nhất, và mỗi hồ sơ bệnh án chỉ thuộc về một người dùng.
* **MedicalRecord - Disease (1-N):** Một hồ sơ bệnh án có thể liên quan đến nhiều bệnh khác nhau, nhưng một bệnh có thể liên quan đến nhiều hồ sơ bệnh án.
* **MedicalRecord - Report (1-N):** Một hồ sơ bệnh án có thể có nhiều báo cáo liên quan, nhưng một báo cáo chỉ liên quan đến một hồ sơ bệnh án.
* **User - MedicalImage (1-N):** Một người dùng có thể có nhiều hình ảnh y tế liên quan, nhưng một hình ảnh y tế chỉ thuộc về một người dùng.
* **User - MedicalHistory (1-1):** Mỗi người dùng chỉ có một bệnh sử duy nhất, và mỗi bệnh sử chỉ thuộc về một người dùng.

**Ý nghĩa của sơ đồ:**

* Sơ đồ này cho thấy một người dùng có thể có nhiều loại dữ liệu y tế khác nhau, bao gồm hồ sơ bệnh án, hình ảnh y tế và bệnh sử.
* Mỗi hồ sơ bệnh án có thể liên quan đến nhiều bệnh khác nhau và tạo ra nhiều báo cáo khác nhau.
* Các mối quan hệ này cho phép chúng ta truy xuất và quản lý thông tin y tế của người dùng một cách hiệu quả.

## 4.2 Mô tả chi tiết các thành phần trong sơ đồ logic

### 4.2.1 Đối tượng người dùng (User)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Giải thích |
| 1 | Id | ObjectId | NOT NULL, UNIQUE | Đây là một mã định danh duy nhất, tự động sinh ra cho mỗi người dùng, đảm bảo mỗi người dùng chỉ có một ID duy nhất. |
| 2 | name | String | NOT NULL | Lưu trữ tên đầy đủ của người dùng. |
| 3 | date\_of\_birth | Date | NOT NULL | Lưu trữ ngày sinh của người dùng. |
| 4 | gende | String | NOT NULL | Lưu trữ giới tính của người dùng. |
| 5 | userName | String | NOT NULL, UNIQUE | Đây là tên người dùng sẽ sử dụng để đăng nhập vào hệ thống. Tên này phải duy nhất để tránh trùng lặp. |
| 6 | contact\_info | String |  | Lưu trữ thông tin liên hệ của người dùng như số điện thoại, email, địa chỉ,... |
| 7 | password | String | NOT NULL | Lưu trữ mật khẩu của người dùng, thường được mã hóa để đảm bảo bảo mật. |
| 8 | role | String |  | Xác định vai trò, phân loại người dùng trong hệ thống (ví dụ: admin, user, guest). |
| 9 | medical\_history\_id | ObjectId |  | Nếu hệ thống quản lý thông tin y tế, trường này sẽ tham chiếu đến ID của hồ sơ bệnh án của người dùng. |
| 10 | created\_at | Timestamp | NOT NULL | Lưu trữ thời điểm tài khoản được tạo. |
| 11 | updated\_at | Timestamp |  | Lưu trữ thời điểm thông tin tài khoản được cập nhật lần cuối. |

**Ý nghĩa tổng quan:**

Bảng thông tin này cung cấp một cấu trúc cơ bản để lưu trữ thông tin về người dùng trong một hệ thống. Các thông tin này có thể được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như:

* **Xác thực người dùng:** Kiểm tra tên đăng nhập và mật khẩu khi người dùng đăng nhập.
* **Quản lý thông tin cá nhân:** Cập nhật, sửa đổi thông tin cá nhân của người dùng.
* **Phân quyền:** Cấp quyền truy cập khác nhau cho các nhóm người dùng khác nhau dựa trên vai trò của họ.
* **Phân tích dữ liệu:** Sử dụng thông tin người dùng để phân tích hành vi, sở thích của người dùng.

### 4.2.2 Đối tượng: Bệnh (Disease)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Giải thích |
| 1 | Id | ObjectId | NOT NULL, UNIQUE | Đây là một mã định danh duy nhất, tự động sinh ra cho mỗi bệnh, đảm bảo mỗi bệnh chỉ có một ID duy nhất. |
| 2 | Name | String | NOT NULL | Lưu trữ tên của bệnh, ví dụ: "Eczema", "Psoriasis". |
| 3 | Symptoms | Array[String] |  | Lưu trữ một danh sách các triệu chứng điển hình của bệnh, ví dụ: "Ngứa", "Đỏ da", "Phát ban". |
| 4 | Reasons | String |  | Lưu trữ thông tin về nguyên nhân hoặc các yếu tố gây ra bệnh. |
| 5 | Treatments | Array[String] |  | Lưu trữ một danh sách các phương pháp điều trị cho bệnh. |
| 6 | Created\_at | Timestamp | NOT NULL | Lưu trữ thời điểm thông tin về bệnh được tạo. |
| 7 | Updated\_at | Timestamp |  | Lưu trữ thời điểm thông tin về bệnh được cập nhật lần cuối. |

**Ý nghĩa tổng quan:**

Bảng thông tin này cung cấp một cấu trúc cơ bản để lưu trữ thông tin về các loại bệnh. Các thông tin này có thể được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như:

* **Tìm kiếm bệnh:** Người dùng có thể tìm kiếm bệnh dựa trên tên bệnh, triệu chứng hoặc nguyên nhân.
* **Quản lý thông tin bệnh:** Cập nhật, sửa đổi thông tin về các bệnh.
* **Hỗ trợ chẩn đoán:** Các bác sĩ có thể tham khảo thông tin về bệnh để đưa ra chẩn đoán chính xác.
* **Phát triển ứng dụng y tế:** Dữ liệu về bệnh có thể được sử dụng để phát triển các ứng dụng hỗ trợ chẩn đoán, điều trị bệnh.

**Ứng dụng thực tế:**

* **Hệ thống quản lý bệnh viện:** Bảng thông tin này có thể được sử dụng để quản lý danh mục các bệnh mà bệnh viện điều trị.
* **Ứng dụng y tế trên di động:** Người dùng có thể tra cứu thông tin về các bệnh thông qua ứng dụng.
* **Công cụ hỗ trợ chẩn đoán:** Các hệ thống hỗ trợ chẩn đoán có thể sử dụng thông tin về bệnh để đưa ra gợi ý chẩn đoán.

### 4.2.3 Đối tượng: Hồ sơ bệnh án (Medical Record)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Giải thích |
| 1 | Id | ObjectId | NOT NULL, UNIQUE | Đây là một mã định danh duy nhất, tự động sinh ra cho mỗi hồ sơ bệnh án, đảm bảo mỗi hồ sơ bệnh án chỉ có một ID duy nhất. |
| 2 | Patient\_id | ObjectId | NOT NULL | Đây là ID của bệnh nhân, liên kết hồ sơ bệnh án này với thông tin của bệnh nhân trong một bảng khác (bảng người dùng). |
| 3 | Disease\_id | ObjectId | NOT NULL | Đây là ID của bệnh mà bệnh nhân đang mắc phải, liên kết hồ sơ bệnh án này với thông tin về bệnh trong một bảng khác (bảng bệnh). |
| 4 | Doctor\_id | ObjectId |  | Đây là ID của bác sĩ phụ trách điều trị cho bệnh nhân, liên kết hồ sơ bệnh án này với thông tin của bác sĩ trong một bảng khác (bảng người dùng, giả sử bác sĩ cũng là một người dùng trong hệ thống). Trường này có thể để trống nếu chưa có bác sĩ nào phụ trách. |
| 5 | Diagnosis | String |  | Lưu trữ kết quả chẩn đoán của bác sĩ về bệnh tình của bệnh nhân. |
| 6 | Treatment\_plan | Array[String] |  | Lưu trữ một danh sách các phương pháp điều trị mà bác sĩ đưa ra cho bệnh nhân. |
| 7 | Note | String |  | Lưu trữ các ghi chú bổ sung từ bác sĩ, ví dụ như các thông tin quan trọng về tình trạng bệnh nhân, các câu hỏi cần làm rõ,... |
| 8 | Created\_at | Timestamp | NOT NULL | Lưu trữ thời điểm hồ sơ bệnh án được tạo. |
| 9 | Updated\_at | Timestamp |  | Lưu trữ thời điểm hồ sơ bệnh án được cập nhật lần cuối (ví dụ: khi có kết quả xét nghiệm mới, khi thay đổi phương án điều trị). |

**Ý nghĩa tổng quan:**

Bảng thông tin này đóng vai trò trung tâm trong một hệ thống quản lý hồ sơ bệnh án. Nó liên kết các thông tin về bệnh nhân, bệnh, bác sĩ và quá trình điều trị lại với nhau. Thông qua bảng này, chúng ta có thể:

* **Theo dõi lịch sử bệnh án của từng bệnh nhân:** Bằng cách truy vấn dựa trên patient\_id.
* **Quản lý thông tin về các bệnh:** Bằng cách truy vấn dựa trên disease\_id.
* **Đánh giá hiệu quả điều trị:** Bằng cách so sánh kết quả chẩn đoán ban đầu với kết quả điều trị.
* **Phân tích dữ liệu:** Sử dụng thông tin trong bảng này để phân tích các xu hướng bệnh tật, hiệu quả của các phương pháp điều trị, v.v.

**Liên kết với các bảng khác:**

* **Bảng người dùng:** Liên kết qua trường patient\_id và doctor\_id để lấy thông tin chi tiết về bệnh nhân và bác sĩ.
* **Bảng bệnh:** Liên kết qua trường disease\_id để lấy thông tin chi tiết về bệnh.

### 4.2.4 Đối tượng: Báo cáo (Report)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Giải thích |
| 1 | Id | ObjectId | NOT NULL, UNIQUE | Đây là một mã định danh duy nhất, tự động sinh ra cho mỗi báo cáo, đảm bảo mỗi báo cáo chỉ có một ID duy nhất. |
| 2 | Medical\_record\_id | ObjectId | NOT NULL | Đây là ID của hồ sơ bệnh án mà báo cáo này liên quan đến. Nó tạo ra một mối liên kết giữa báo cáo và hồ sơ bệnh án của bệnh nhân. |
| 3 | Report\_date | Date | NOT NULL | Đây là ngày mà báo cáo được lập. |
| 4 | Content | String | NOT NULL | Trường này chứa nội dung chính của báo cáo, bao gồm kết quả xét nghiệm, chẩn đoán, phác đồ điều trị, nhận xét của bác sĩ, v.v. |
| 5 | Doctor\_id | ObjectId, | NOT NULL | Đây là ID của bác sĩ đã lập báo cáo. Nó tạo ra một mối liên kết giữa báo cáo và bác sĩ phụ trách. |
| 6 | Created\_at | Timestamp | NOT NULL | Lưu trữ thời điểm báo cáo được tạo. |
| 7 | Updated\_at | Timestamp |  | Lưu trữ thời điểm thông tin trong báo cáo được cập nhật lần cuối. |

**Ý nghĩa tổng quan:**

Bảng thông tin này đóng vai trò quan trọng trong việc lưu trữ và quản lý các báo cáo y tế. Nó cho phép chúng ta theo dõi quá trình điều trị của bệnh nhân, đánh giá hiệu quả của các phương pháp điều trị, và cung cấp thông tin cần thiết cho việc ra quyết định y tế.

**Liên kết với các bảng khác:**

* **Bảng hồ sơ bệnh án:** Liên kết qua trường medical\_record\_id để truy xuất thông tin chi tiết về bệnh nhân và bệnh.
* **Bảng người dùng:** Liên kết qua trường doctor\_id để truy xuất thông tin chi tiết về bác sĩ.

**Ứng dụng thực tế:**

* **Theo dõi tiến trình điều trị:** Bằng cách xem các báo cáo được tạo trong quá trình điều trị, bác sĩ có thể đánh giá tiến triển của bệnh nhân và điều chỉnh phác đồ điều trị nếu cần.
* **Phân tích dữ liệu:** Dữ liệu từ các báo cáo có thể được sử dụng để phân tích hiệu quả của các phương pháp điều trị khác nhau, xác định các yếu tố nguy cơ, và cải thiện chất lượng chăm sóc sức khỏe.
* **Lưu trữ hồ sơ bệnh án điện tử:** Các báo cáo được lưu trữ trong hệ thống điện tử, giúp dễ dàng truy xuất và chia sẻ thông tin với các bên liên quan.

### 4.2.5 Đối tượng: Hình ảnh y khoa (medical Image)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Giải thích |
| 1 | Id | ObjectId | NOT NULL, UNIQUE | Đây là một mã định danh duy nhất, tự động sinh ra cho mỗi hình ảnh y khoa, đảm bảo mỗi hình ảnh chỉ có một ID duy nhất. |
| 2 | Medical\_record\_id | ObjectId | NOT NULL | Đây là ID của hồ sơ bệnh án mà hình ảnh này liên quan đến. Nó tạo ra một mối liên kết giữa hình ảnh và hồ sơ bệnh án của bệnh nhân. |
| 3 | Image\_path | String | NOT NULL | Đây là đường dẫn đến file chứa hình ảnh y khoa trên hệ thống lưu trữ. Đường dẫn này cho phép hệ thống tìm và hiển thị hình ảnh một cách chính xác. |
| 4 | Description | String | NOT NULL | Trường này chứa mô tả ngắn gọn về hình ảnh, ví dụ như loại hình ảnh (X-quang, MRI, CT scan), vùng cơ thể được chụp, v.v. |
| 5 | Uploaded\_at | Timestamp | NOT NULL | Lưu trữ thời điểm hình ảnh được tải lên hệ thống. |

**Ý nghĩa tổng quan:**

Bảng thông tin này đóng vai trò quan trọng trong việc lưu trữ và quản lý các hình ảnh y tế. Nó cho phép chúng ta liên kết hình ảnh với hồ sơ bệnh án của bệnh nhân, giúp bác sĩ dễ dàng xem xét và chẩn đoán bệnh.

**Liên kết với các bảng khác:**

* **Bảng hồ sơ bệnh án:** Liên kết qua trường medical\_record\_id để truy xuất thông tin chi tiết về bệnh nhân và bệnh.

**Ứng dụng thực tế:**

* **Hỗ trợ chẩn đoán:** Các bác sĩ có thể xem trực tiếp hình ảnh y tế để đánh giá tình trạng bệnh của bệnh nhân và đưa ra chẩn đoán chính xác.
* **Theo dõi tiến trình điều trị:** Bằng cách so sánh các hình ảnh được chụp ở các thời điểm khác nhau, bác sĩ có thể theo dõi sự tiến triển của bệnh và hiệu quả của phương pháp điều trị.
* **Lưu trữ hồ sơ bệnh án điện tử:** Hình ảnh y tế được lưu trữ cùng với hồ sơ bệnh án, giúp hoàn thiện hồ sơ bệnh án của bệnh nhân.

**Các trường hợp sử dụng khác:**

* **Telehealth:** Hình ảnh y tế có thể được chia sẻ giữa các bác sĩ để tham khảo ý kiến hoặc thực hiện hội chẩn từ xa.
* **Nghiên cứu y học:** Hình ảnh y tế có thể được sử dụng để nghiên cứu các bệnh mới, phát triển các phương pháp điều trị mới.

### 4.2.6 Đối tượng: Lịch sử khám chữa bệnh (Medical History)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Giải thích |
| 1 | Id | ObjectId | NOT NULL, UNIQUE | Đây là một mã định danh duy nhất, tự động sinh ra cho mỗi lần khám bệnh, đảm bảo mỗi lần khám chỉ có một ID duy nhất. |
| 2 | Ipatient\_idd | ObjectId | NOT NULL | Đây là ID của bệnh nhân, liên kết lần khám bệnh này với thông tin của bệnh nhân trong một bảng khác (bảng người dùng). |
| 3 | Visit\_date | Date | NOT NULL | Đây là ngày mà bệnh nhân đến khám bệnh. |
| 4 | Doctor\_id | ObjectId |  | Đây là ID của bác sĩ khám bệnh cho bệnh nhân, liên kết lần khám bệnh này với thông tin của bác sĩ trong một bảng khác (bảng người dùng). Trường này có thể để trống nếu chưa có bác sĩ nào khám. |
| 5 | Notes | String |  | Lưu trữ các ghi chú về lần khám bệnh, ví dụ như triệu chứng bệnh nhân, kết quả khám, chẩn đoán, phương pháp điều trị, lời khuyên của bác sĩ,... |
| 6 | Created\_at | Timestamp | NOT NULL | Lưu trữ thời điểm tạo lịch sử khám bệnh. |
| 7 | Updated\_at | Timestamp |  | Lưu trữ thời điểm cập nhật thông tin của lần khám bệnh. |

**Ý nghĩa tổng quan:**

Bảng thông tin này cung cấp một lịch sử chi tiết về các lần khám bệnh của mỗi bệnh nhân. Nó giúp theo dõi sự tiến triển của bệnh, đánh giá hiệu quả điều trị và cung cấp thông tin cần thiết cho các lần khám sau.

**Liên kết với các bảng khác:**

Bảng người dùng: Liên kết qua trường patient\_id và doctor\_id để truy xuất thông tin chi tiết về bệnh nhân và bác sĩ.

**Ứng dụng thực tế**:

* Theo dõi sức khỏe: Bệnh nhân có thể xem lại lịch sử khám bệnh của mình để hiểu rõ hơn về tình trạng sức khỏe và quá trình điều trị.
* Quản lý bệnh nhân: Bác sĩ có thể dễ dàng truy cập vào lịch sử khám bệnh của bệnh nhân để đưa ra các quyết định điều trị phù hợp.
* Nghiên cứu y học: Dữ liệu từ bảng lịch sử khám bệnh có thể được sử dụng để nghiên cứu các xu hướng bệnh tật, hiệu quả của các phương pháp điều trị, v.v.

**Các trường hợp sử dụng khác:**

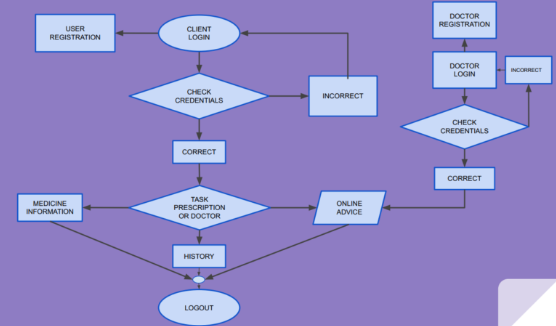
* Đánh giá chất lượng dịch vụ: Bằng cách phân tích dữ liệu về thời gian chờ đợi, thời gian khám bệnh, sự hài lòng của bệnh nhân, chúng ta có thể đánh giá chất lượng dịch vụ của cơ sở y tế.
* Quản lý chi phí: Dữ liệu về các dịch vụ y tế được cung cấp trong mỗi lần khám bệnh có thể được sử dụng để tính toán chi phí điều trị.

# CHƯƠNG V: THIẾT KẾ GIAO DIỆN

## 5.1 Kiến trúc hệ thống

Hệ thống quản lý thông tin y tế được phát triển với mục tiêu cung cấp công cụ quản lý và hỗ trợ trong quá trình khám và điều trị bệnh. Để đáp ứng yêu cầu về bảo mật, tính linh hoạt và khả năng mở rộng, chúng tôi sử dụng **Django Framework** - một framework mạnh mẽ và phổ biến được phát triển trên nền tảng Python. Django giúp giảm thiểu thời gian và công sức phát triển bằng cách cung cấp nhiều tính năng tích hợp sẵn và một mô hình lập trình rõ ràng, phù hợp cho các hệ thống lớn cần khả năng bảo trì cao.

**5.1.1 Sơ đồ kiến trúc tổng quan**

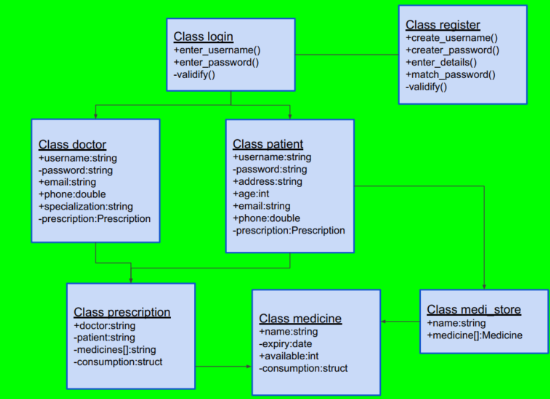
****

**Giải thích chi tiết:**

Sơ đồ này mô tả cách các thành phần khác nhau của hệ thống tương tác với nhau.

* **Client:** Đây là giao diện mà người dùng trực tiếp tương tác. Nó có thể là một ứng dụng web truy cập thông qua trình duyệt hoặc một ứng dụng di động. Client gửi các yêu cầu đến server để thực hiện các tác vụ như đăng nhập, đặt lịch hẹn, xem hồ sơ bệnh án, v.v.
* **Server:** Máy chủ chạy ứng dụng Django. Khi nhận được yêu cầu từ client, server sẽ xử lý yêu cầu đó, truy xuất hoặc cập nhật dữ liệu trong cơ sở dữ liệu, và gửi trả kết quả về cho client.
* **Database:** Cơ sở dữ liệu lưu trữ tất cả dữ liệu của hệ thống. Các dữ liệu này có thể bao gồm:
  + Thông tin cá nhân của bệnh nhân (tên, tuổi, giới tính, địa chỉ,...)
  + Thông tin về bác sĩ (chuyên khoa, lịch làm việc,...)
  + Hồ sơ bệnh án (chẩn đoán, kê đơn, kết quả xét nghiệm,...)
  + Lịch hẹn khám bệnh
  + Dữ liệu hình ảnh y tế (X-quang, MRI,...)
* **API:** Đây là một giao diện lập trình ứng dụng cho phép các ứng dụng bên thứ ba tương tác với hệ thống. Ví dụ, một ứng dụng di động có thể sử dụng API để truy xuất thông tin về lịch hẹn của bệnh nhân.

### 5.1.2 Sơ đồ lớp hệ thống chăm sóc sức khỏe trực tiếp

****

**Các lớp và ý nghĩa**

 **Class Login:**

* Đại diện cho quá trình đăng nhập vào hệ thống.
* **Các phương thức:**
  + **create\_username()**: Tạo tên người dùng mới.
  + **enter\_username()**: Nhập tên người dùng để đăng nhập.
  + **enter\_password()**: Nhập mật khẩu để đăng nhập.
  + **validify()**: Kiểm tra tính hợp lệ của tên đăng nhập và mật khẩu.

 **Class Register:**

* Đại diện cho quá trình đăng ký tài khoản mới.
* **Các phương thức:**
  + **create\_username()**: Tạo tên người dùng mới.
  + **enter\_password()**: Nhập mật khẩu.
  + **enter\_details()**: Nhập các thông tin chi tiết khác (ví dụ: email, địa chỉ).
  + **match\_password()**: Kiểm tra xem hai mật khẩu nhập vào có trùng nhau hay không.
  + **validify()**: Kiểm tra tính hợp lệ của thông tin đăng ký.

 **Class Doctor:**

* Đại diện cho một bác sĩ.
* **Các thuộc tính:**
  + **username**: Tên đăng nhập.
  + **password**: Mật khẩu.
  + **email**: Địa chỉ email.
  + **phone**: Số điện thoại.
  + **specialization**: Chuyên khoa.
  + **prescription**: Đơn thuốc (liên kết đến lớp Prescription).

 **Class Patient:**

* Đại diện cho một bệnh nhân.
* **Các thuộc tính:**
  + **username**: Tên đăng nhập.
  + **password**: Mật khẩu.
  + **address**: Địa chỉ.
  + **age**: Tuổi.
  + **email**: Địa chỉ email.
  + **phone**: Số điện thoại.
  + **prescription**: Đơn thuốc (liên kết đến lớp Prescription).

 **Class Prescription:**

* Đại diện cho một đơn thuốc.
* **Các thuộc tính:**
  + **doctor**: Bác sĩ kê đơn (liên kết đến lớp Doctor).
  + **patient**: Bệnh nhân (liên kết đến lớp Patient).
  + **medicines**: Danh sách các loại thuốc trong đơn.
  + **consumption**: Liều lượng và cách dùng thuốc.

 **Class Medicine:**

* Đại diện cho một loại thuốc.
* **Các thuộc tính:**
  + **name**: Tên thuốc.
  + **expiry\_date**: Hạn sử dụng.
  + **available**: Số lượng còn lại.
  + **consumption**: Cách dùng thuốc.

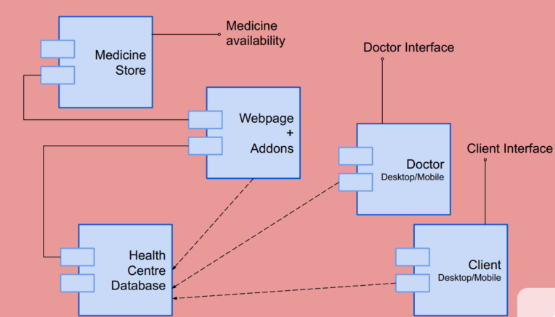
 **Class MediStore:**

* Đại diện cho một nhà thuốc.
* **Các thuộc tính:**
  + **name**: Tên nhà thuốc.
  + **medicines**: Danh sách các loại thuốc có sẵn trong nhà thuốc.

**Quan hệ giữa các lớp**

* **Login** và **Register** có quan hệ chặt chẽ với các lớp **Doctor** và **Patient** vì quá trình đăng nhập và đăng ký là để tạo tài khoản cho bác sĩ và bệnh nhân.
* **Doctor** và **Patient** có quan hệ một-nhiều với **Prescription** vì một bác sĩ có thể kê nhiều đơn thuốc cho nhiều bệnh nhân và một bệnh nhân có thể nhận nhiều đơn thuốc từ các bác sĩ khác nhau.
* **Prescription** có quan hệ một-nhiều với **Medicine** vì một đơn thuốc có thể bao gồm nhiều loại thuốc khác nhau.
* **Medicine** và **MediStore** có quan hệ nhiều-nhiều vì một loại thuốc có thể có ở nhiều nhà thuốc khác nhau và một nhà thuốc có thể bán nhiều loại thuốc.

### 5.1.3 Sơ đồ phát triển hệ thống chăm sóc sức khỏe trực tiếp

****

**Các thành phần chính:**

Health Centre Database:

* Đây là trung tâm lưu trữ tất cả dữ liệu của hệ thống.
* Bao gồm thông tin về bệnh nhân, bác sĩ, lịch hẹn, đơn thuốc, thuốc, và các thông tin y tế khác.

Webpage + Addons:

* Đây là giao diện chính mà người dùng tương tác với hệ thống.
* Cho phép người dùng thực hiện các tác vụ như:
* Đăng nhập
* Đặt lịch hẹn
* Xem hồ sơ bệnh án
* Tìm kiếm thông tin về thuốc
* Và các chức năng khác tùy theo thiết kế của hệ thống.

Doctor Interface:

* Giao diện dành riêng cho bác sĩ.
* Cho phép bác sĩ:
* Xem thông tin bệnh nhân
* Kê đơn thuốc
* Theo dõi lịch khám
* Và các tác vụ liên quan đến khám chữa bệnh.

Client Interface:

* Giao diện dành cho bệnh nhân.
* Cho phép bệnh nhân:
* Đặt lịch hẹn
* Xem hồ sơ bệnh án
* Thanh toán
* Và các tác vụ liên quan đến việc quản lý sức khỏe cá nhân.

Medicine Store:

* Đại diện cho kho thuốc.
* Lưu trữ thông tin về các loại thuốc, số lượng tồn kho.
* Cung cấp thông tin về tình trạng sẵn có của thuốc cho các thành phần khác trong hệ thống.

**Quan hệ giữa các thành phần:**

* Health Centre Database: Là trung tâm của hệ thống, cung cấp dữ liệu cho tất cả các thành phần khác.
* Webpage + Addons: Là cầu nối giữa người dùng và hệ thống. Nó truy xuất dữ liệu từ cơ sở dữ liệu và hiển thị cho người dùng.
* Doctor Interface và Client Interface: Là các giao diện chuyên biệt cho từng nhóm người dùng, truy xuất dữ liệu từ cơ sở dữ liệu thông qua webpage.
* Medicine Store: Cung cấp thông tin về thuốc cho webpage và các giao diện khác.

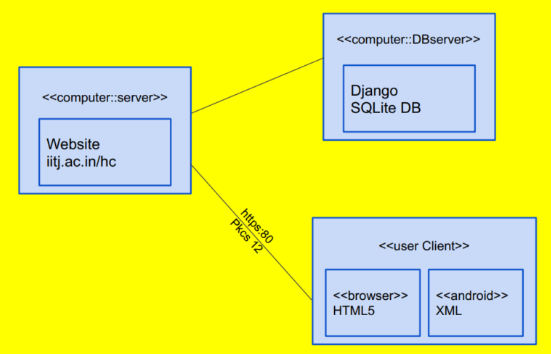
**Các dòng liên kết:**

* Dòng liền nét: Thể hiện mối quan hệ trực tiếp, ví dụ như webpage truy xuất dữ liệu từ cơ sở dữ liệu.
* Dòng đứt nét: Thể hiện mối quan hệ gián tiếp, ví dụ như doctor interface truy xuất dữ liệu về thuốc thông qua webpage.

**Ý nghĩa của sơ đồ:**

Sơ đồ này cho thấy một hệ thống y tế được số hóa, nơi mà thông tin của bệnh nhân được quản lý tập trung. Bác sĩ và bệnh nhân có thể truy cập và cập nhật thông tin một cách dễ dàng và thuận tiện. Hệ thống cũng giúp quản lý thuốc và các tài nguyên y tế hiệu quả hơn.

### 5.1.4 Sơ đồ quan sát vật lý của hệ thống chăm sóc sức khỏe trực tiếp

****

**Tổng quan**

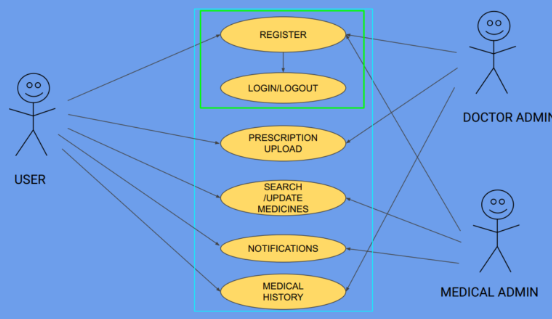
Sơ đồ này mô tả một hệ thống web đơn giản, có thể là một phần của một hệ thống quản lý thông tin y tế lớn hơn. Hệ thống này bao gồm các thành phần chính sau:

* **Client:** Là nơi người dùng tương tác trực tiếp với hệ thống, thông qua trình duyệt web hoặc ứng dụng di động (Android).
* **Server:** Là máy chủ chạy ứng dụng web, trong trường hợp này là Django. Django sẽ xử lý các yêu cầu từ client và trả về kết quả.
* **Database:** Là nơi lưu trữ dữ liệu của hệ thống, trong trường hợp này là SQLite.

**Chi tiết các thành phần và mối quan hệ**

1. **Client:**
   * **Browser (HTML5):** Người dùng truy cập vào hệ thống thông qua trình duyệt web, giao diện được xây dựng bằng HTML5.
   * **Android (XML):** Nếu có, người dùng cũng có thể truy cập hệ thống thông qua ứng dụng di động trên nền tảng Android, giao diện được xây dựng bằng XML.
2. **Server:**
   * **Django:** Đây là một framework web của Python, được sử dụng để xây dựng ứng dụng web. Django xử lý các yêu cầu từ client, tương tác với cơ sở dữ liệu và trả về kết quả dưới dạng HTML.
   * **SQLite:** Đây là một hệ quản trị cơ sở dữ liệu nhẹ, được sử dụng để lưu trữ dữ liệu của hệ thống.
3. **Mối quan hệ:**
   * **Client -> Server:** Khi người dùng thực hiện một hành động trên giao diện (ví dụ: đăng nhập, tìm kiếm thông tin), một yêu cầu HTTP sẽ được gửi từ client đến server.
   * **Server -> Database:** Django sẽ truy xuất hoặc cập nhật dữ liệu trong cơ sở dữ liệu SQLite để đáp ứng yêu cầu của client.
   * **Server -> Client:** Sau khi xử lý xong yêu cầu, server sẽ gửi kết quả (thường là một trang web) trở lại client để hiển thị.

### 5.1.5 Biểu đồ trường hợp sử dụng hệ thống chăm sóc sức khỏe trực tiếp

****

**Các thành phần chính trong sơ đồ:**

* **Actor:**
  + **User:** Đại diện cho người dùng chung, có thể là bệnh nhân, khách hàng.
  + **Doctor Admin:** Đại diện cho người quản lý, bác sĩ có quyền quản trị hệ thống.
  + **Medical Admin:** Đại diện cho người quản lý y tế, có quyền quản lý thông tin y tế.
* **Use Case:**
  + **Register:** Chức năng đăng ký tài khoản mới.
  + **Login/Logout:** Chức năng đăng nhập và đăng xuất khỏi hệ thống.
  + **Prescription Upload:** Chức năng tải lên đơn thuốc.
  + **Search/Update Medicines:** Chức năng tìm kiếm và cập nhật thông tin về thuốc.
  + **Notifications:** Chức năng thông báo cho người dùng.
  + **Medical History:** Chức năng xem lịch sử bệnh án.

**Quan hệ giữa các thành phần:**

* **User:** Người dùng có thể thực hiện hầu hết các chức năng của hệ thống, bao gồm đăng ký, đăng nhập, tìm kiếm thông tin, xem lịch sử bệnh án.
* **Doctor Admin:** Có quyền truy cập vào các chức năng quản lý như tải lên đơn thuốc, cập nhật thông tin thuốc.
* **Medical Admin:** Có quyền truy cập vào các chức năng quản lý cấp cao hơn, có thể liên quan đến việc quản lý toàn bộ hệ thống, dữ liệu bệnh nhân.

**Ý nghĩa của sơ đồ:**

* **Sơ đồ use case** giúp chúng ta hiểu rõ hơn về các chức năng mà hệ thống cần cung cấp.
* **Giúp xác định các yêu cầu của người dùng** và lên kế hoạch phát triển phần mềm một cách hiệu quả.
* **Dùng để giao tiếp giữa các thành viên trong nhóm phát triển** và khách hàng.

### 5.1.6 Lựa chọn Django Framework

Django được chọn làm nền tảng phát triển vì các lý do sau:

* **Nhanh chóng và hiệu quả**
* **Thư viện phong phú:** Django cung cấp một kho tàng các thư viện sẵn có, bao gồm các thư viện xử lý form, xác thực, phiên làm việc, ... Điều này giúp các nhà phát triển tiết kiệm rất nhiều thời gian trong việc viết lại các chức năng cơ bản.
* **ORM mạnh mẽ:** Django ORM (Object-Relational Mapper) cung cấp một cách tiếp cận trực quan để tương tác với cơ sở dữ liệu, cho phép bạn viết các truy vấn phức tạp bằng ngôn ngữ Python tự nhiên.
* **Tích hợp sẵn nhiều tính năng:** Django đã tích hợp sẵn nhiều tính năng hữu ích như hệ thống quản trị nội dung (CMS), hệ thống xác thực người dùng, hệ thống email, ... giúp bạn xây dựng các ứng dụng đầy đủ tính năng một cách nhanh chóng.
* **Cấu trúc MTV rõ ràng**: Django sử dụng kiến trúc MTV (Model-Template-View), tương tự với mô hình MVC truyền thống nhưng có sự điều chỉnh phù hợp hơn cho ứng dụng web. Trong đó:
  + **Model**: Đảm nhiệm việc quản lý và tương tác với dữ liệu, cung cấp một lớp ORM giúp thao tác với cơ sở dữ liệu dễ dàng.
  + **Template**: Xử lý giao diện và hiển thị nội dung dữ liệu đến người dùng.
  + **View**: Xử lý các yêu cầu từ người dùng, thực thi logic nghiệp vụ và trả về kết quả.
* **Tính bảo mật cao**:
* **Các tính năng bảo mật tích hợp:** Django đã tích hợp sẵn nhiều tính năng bảo mật giúp bảo vệ ứng dụng của bạn khỏi các cuộc tấn công phổ biến như:
* **XSS:** Tự động thoát các ký tự đặc biệt để ngăn chặn các cuộc tấn công XSS.
* **CSRF:** Sử dụng token CSRF để ngăn chặn các cuộc tấn công CSRF.
* **SQL Injection:** Django ORM tự động trích xuất các tham số và ngăn chặn các cuộc tấn công SQL Injection.
* **Cập nhật bảo mật thường xuyên:** Django có một cộng đồng lớn và năng động, luôn cập nhật các lỗ hổng bảo mật mới và phát hành các bản vá lỗi một cách nhanh chóng.

* **Cộng đồng lớn và tài liệu phong phú**:
* **Cộng đồng sôi động:** Django có một cộng đồng người dùng rất lớn, luôn sẵn sàng giúp đỡ và chia sẻ kinh nghiệm.
* **Tài liệu chi tiết:** Tài liệu chính thức của Django rất đầy đủ và chi tiết, bao gồm các ví dụ minh họa và hướng dẫn từng bước.
* **Thư viện thứ ba:** Có rất nhiều thư viện thứ ba được phát triển cho Django, giúp bạn mở rộng chức năng của ứng dụng một cách dễ dàng.

**Ưu điểm khác của Django**

* **Linh hoạt:** Django có thể được sử dụng để xây dựng các loại ứng dụng web khác nhau, từ các trang web đơn giản đến các ứng dụng doanh nghiệp phức tạp.
* **Mở rộng:** Django có một hệ thống plugin mạnh mẽ, cho phép bạn tùy chỉnh và mở rộng chức năng của ứng dụng.

**5.1.7 Mô hình kiến trúc hệ thống**

Hệ thống được thiết kế theo mô hình phân lớp với các lớp chức năng chính, đảm bảo khả năng mở rộng và dễ dàng bảo trì. Các lớp chức năng bao gồm:

1. **Lớp Presentation (Giao diện)**:
   * Lớp này chịu trách nhiệm hiển thị dữ liệu và tương tác với người dùng cuối.
   * Django sử dụng hệ thống template mạnh mẽ giúp hiển thị dữ liệu từ backend ra giao diện người dùng một cách trực quan.
   * Các công nghệ HTML, CSS, và JavaScript được sử dụng để xây dựng giao diện, tạo trải nghiệm người dùng mượt mà và thân thiện.
2. **Lớp Business Logic (Xử lý nghiệp vụ)**:
   * Đây là lớp nơi các quy tắc nghiệp vụ của hệ thống được thực hiện. Django View sẽ chứa các logic xử lý chính.
   * Các thao tác như xử lý thông tin bệnh nhân, lưu trữ hồ sơ bệnh án, và quản lý hình ảnh y khoa đều được thực hiện ở lớp này.
   * Các hàm xử lý đảm bảo rằng dữ liệu từ người dùng được kiểm tra và xử lý đúng cách trước khi đưa vào cơ sở dữ liệu.
3. **Lớp Data Access (Truy cập dữ liệu)**:
   * Django sử dụng ORM (Object-Relational Mapping) giúp tương tác với cơ sở dữ liệu qua các model mà không cần viết câu lệnh SQL phức tạp.
   * ORM giúp truy xuất, thêm, sửa, xóa dữ liệu dễ dàng và nhất quán, nâng cao hiệu suất phát triển.
   * Điều này giúp các nhà phát triển dễ dàng chuyển đổi giữa các hệ quản trị cơ sở dữ liệu khác nhau nếu cần thiết.
4. **Lớp Database (Cơ sở dữ liệu)**:
   * Hệ quản trị cơ sở dữ liệu MySQL được lựa chọn để lưu trữ dữ liệu của hệ thống.
   * Các bảng trong cơ sở dữ liệu lưu trữ các thông tin quan trọng như hồ sơ bệnh nhân, chẩn đoán, hình ảnh y khoa và lịch sử điều trị.

## 5.2 Chi tiết kỹ thuật

### 5.2.1 Cơ sở dữ liệu

* **Thiết kế cơ sở dữ liệu:**
  + Các bảng: bệnh nhân, bác sĩ, hồ sơ bệnh án, lịch hẹn, kết quả xét nghiệm, v.v.
  + Quan hệ giữa các bảng: sử dụng khóa ngoại để liên kết các bảng với nhau.
  + Chỉ mục: Tạo chỉ mục cho các trường được truy vấn thường xuyên để tăng tốc độ truy vấn.
* **Normal hóa dữ liệu:** Đảm bảo dữ liệu được lưu trữ một cách hợp lý và tránh trùng lặp.
* **Bảo mật dữ liệu:** Mã hóa dữ liệu nhạy cảm như mật khẩu, thông tin bệnh nhân.

### 5.2.2 API

* **REST API:** Sử dụng REST API để cho phép các ứng dụng bên thứ ba tương tác với hệ thống.
* **Authentication:** Sử dụng token hoặc OAuth để xác thực các yêu cầu API.
* **Rate limiting:** Hạn chế số lượng yêu cầu API để bảo vệ hệ thống.

### 5.2.3 Bảo mật

* **Xác thực và ủy quyền:** Sử dụng các cơ chế xác thực mạnh mẽ như mật khẩu hash, hai yếu tố xác thực.
* **Quyền truy cập:** Cấp quyền truy cập khác nhau cho từng loại người dùng (bệnh nhân, bác sĩ, quản trị viên).
* **Mã hóa dữ liệu:** Mã hóa dữ liệu nhạy cảm khi truyền và lưu trữ.
* **Bảo vệ chống tấn công:** Áp dụng các biện pháp bảo vệ chống lại các loại tấn công như SQL injection, XSS, CSRF.

### 5.2.4 Tích hợp

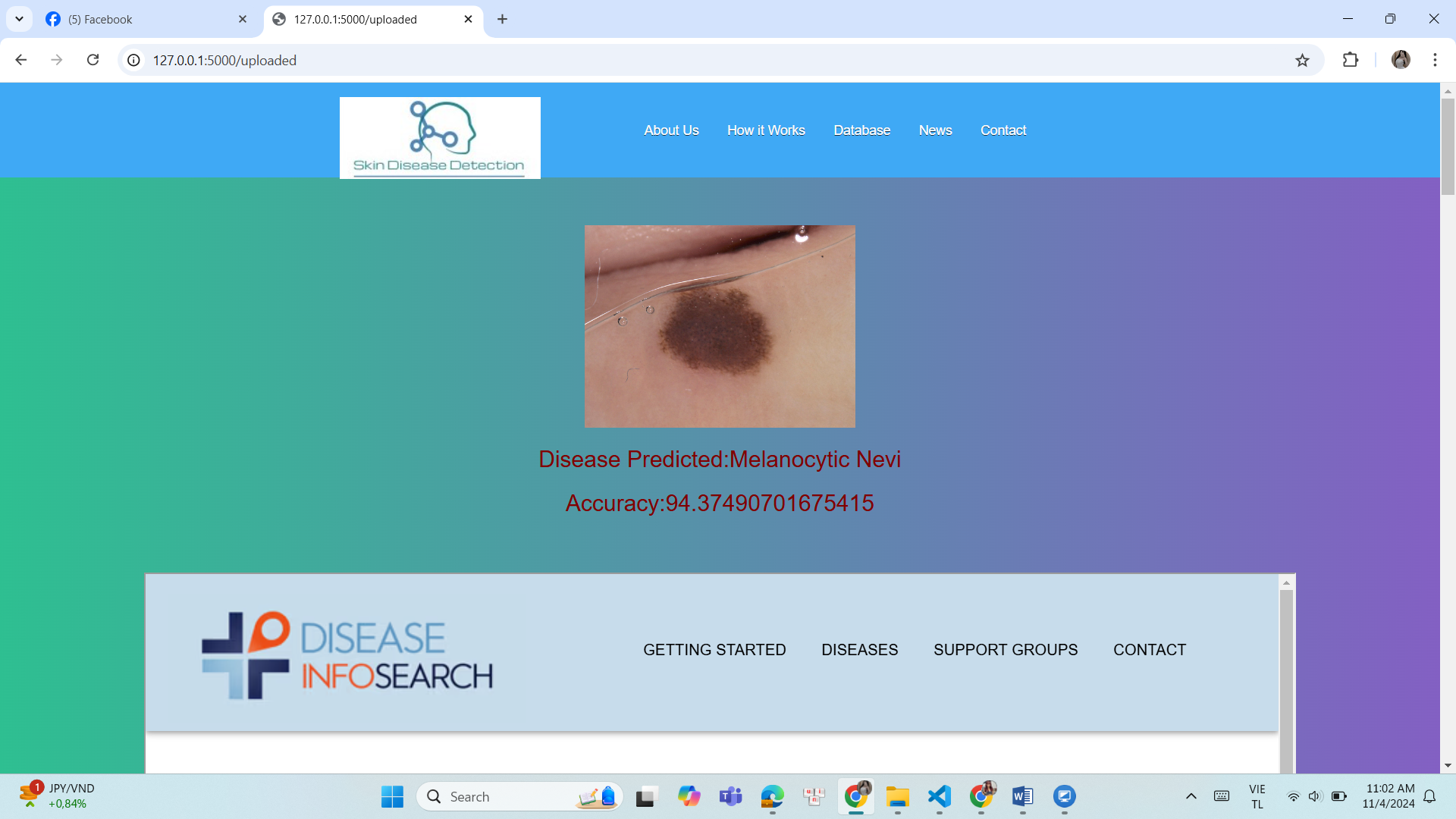
* **Tích hợp với các hệ thống khác:** Sử dụng các giao thức như HL7, FHIR để tích hợp với các hệ thống y tế khác.
* **Tích hợp với các thiết bị y tế:** Kết nối với các thiết bị y tế để thu thập dữ liệu.

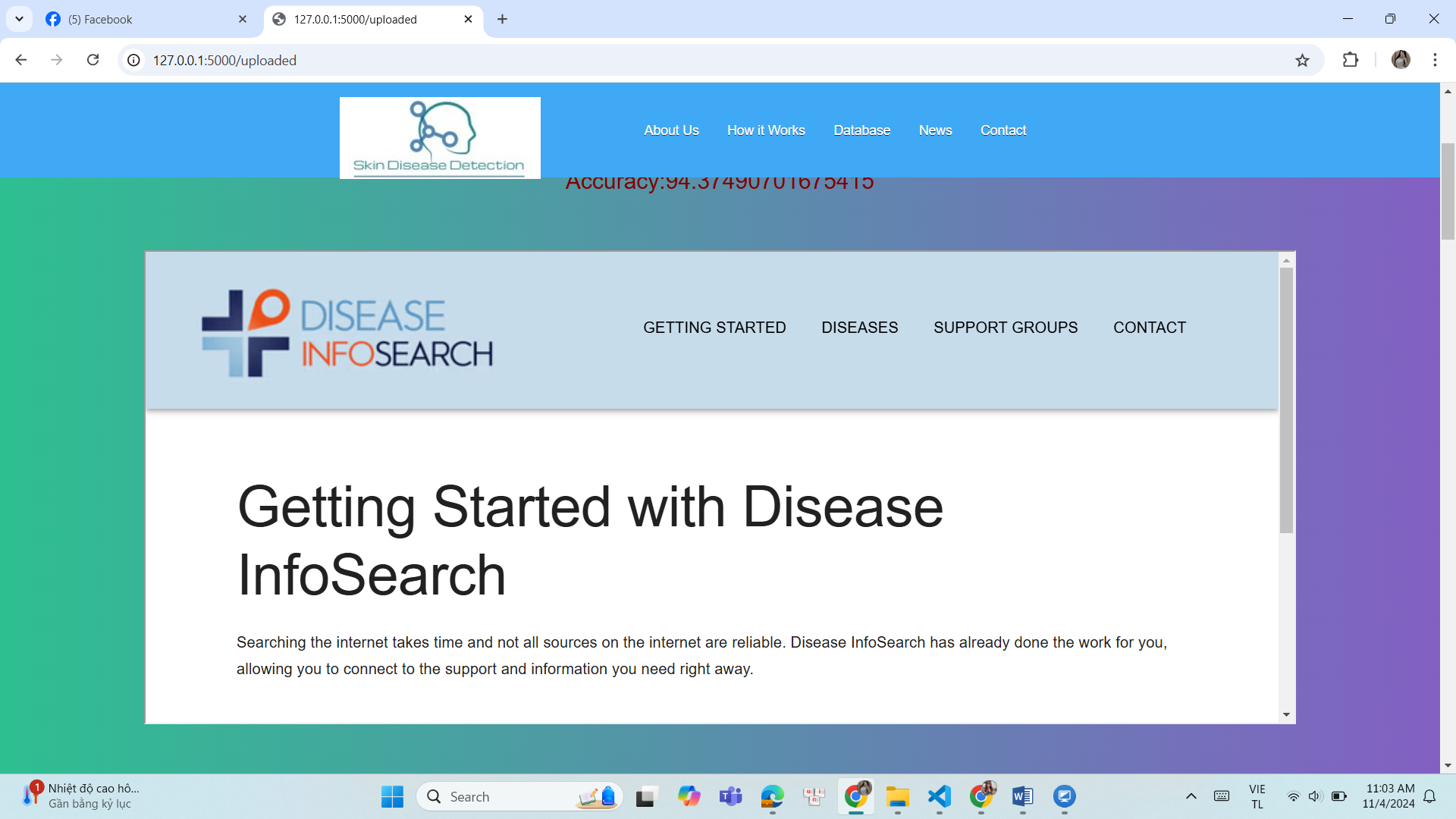
## 5.3 Mô tả chi tiết thành phần trong hệ thống

Để đáp ứng các yêu cầu của hệ thống quản lý thông tin y tế, tôi chia thành các thành phần chính sau:

1. **Authentication Module (Mô-đun Xác thực)**:
   * Mô-đun này quản lý việc đăng nhập, đăng xuất và xác thực người dùng.
   * Django cung cấp các lớp và công cụ hỗ trợ xác thực, quản lý phiên làm việc của người dùng, đảm bảo rằng chỉ người dùng đã xác thực mới có thể truy cập vào các tính năng bảo mật.
2. **Patient Management (Quản lý người dùng)**:
   * Thành phần này quản lý thông tin và hồ sơ người dùng, bao gồm các thao tác thêm, sửa, xóa và tra cứu thông tin.
   * Các thông tin về bệnh nhân sẽ được lưu trữ trong cơ sở dữ liệu và được truy cập qua ORM của Django, đảm bảo tính nhất quán và bảo mật.
3. **Medical Record Management (Quản lý hồ sơ y khoa)**:
   * Thành phần này lưu trữ và truy xuất hồ sơ bệnh án của bệnh nhân, bao gồm các chẩn đoán, phương pháp điều trị và lịch sử khám.
   * Hồ sơ bệnh án sẽ giúp bác sĩ và nhân viên y tế theo dõi quá trình điều trị và cung cấp tư vấn y khoa chính xác.
4. **Image Management (Quản lý hình ảnh y khoa)**:
   * Mô-đun này chịu trách nhiệm quản lý và lưu trữ hình ảnh y khoa của bệnh nhân như X-quang, CT scan.
   * Django kết hợp với các dịch vụ lưu trữ hình ảnh để quản lý tập tin ảnh và cho phép truy xuất khi cần.
5. **Consultation & Treatment Support (Hỗ trợ tư vấn và điều trị)**:
   * Cho phép bệnh nhân nhận tư vấn và hỗ trợ từ bác sĩ trực tuyến, thông qua các tính năng như chat hoặc gửi yêu cầu tư vấn.
   * Thành phần này giúp nâng cao trải nghiệm của bệnh nhân bằng cách cung cấp dịch vụ tư vấn từ xa, đặc biệt hữu ích trong các tình huống cần điều trị khẩn cấp.
6. **Report Generation (Tạo báo cáo)**:
   * Chức năng này tạo các báo cáo tiến trình điều trị cho từng bệnh nhân.
   * Báo cáo được xuất theo định kỳ hoặc theo yêu cầu của bác sĩ và bệnh nhân, giúp theo dõi kết quả điều trị và hỗ trợ quyết định y tế.

# CHƯƠNG 6: GIAO DIỆN HOÀN THÀNH





# CHƯƠNG 7: KẾT LUẬN

## 7.1 Môi trường phát triển và môi trường triển khai

Trong quá trình phát triển và triển khai hệ thống quản lý hồ sơ bệnh án cho các bệnh viện hoặc phòng khám da liễu, chúng tôi đã sử dụng các công cụ và công nghệ tiên tiến để đảm bảo chất lượng và hiệu quả.

* **Môi trường phát triển**:
  + Ngôn ngữ lập trình chính được sử dụng là **Python** với framework **Django** để xây dựng các tính năng chính, đảm bảo tính bảo mật và khả năng mở rộng.
  + **JavaScript** (React) được dùng để phát triển giao diện người dùng, giúp hệ thống có khả năng phản hồi nhanh và giao diện thân thiện.
  + **MySQL** được chọn làm hệ quản trị cơ sở dữ liệu vì khả năng xử lý dữ liệu lớn và khả năng tích hợp dễ dàng với các công cụ khác.
  + Quản lý phiên bản và hợp tác nhóm được thực hiện thông qua **Git** và các nền tảng như **GitHub**, giúp theo dõi và kiểm soát các thay đổi mã nguồn.
  + Ngoài ra, chúng tôi sử dụng **Docker** để tạo môi trường phát triển nhất quán, đảm bảo tính tương thích giữa các môi trường khác nhau. Công cụ IDE chủ yếu là **Visual Studio Code** hoặc **PhpStorm** để hỗ trợ lập trình hiệu quả hơn.

**Các lợi ích cụ thể**

* **Phát triển nhanh:**
  + Django cung cấp nhiều tính năng sẵn có, giúp giảm thiểu thời gian viết code.
  + React giúp xây dựng giao diện nhanh chóng.
* **Bảo mật:**
  + Django có các tính năng bảo mật tích hợp như phòng chống XSS, CSRF, SQL Injection.
  + MySQL cung cấp các cơ chế bảo mật để bảo vệ dữ liệu.
* **Khả năng mở rộng:**
  + Hệ thống có thể dễ dàng mở rộng để đáp ứng nhu cầu tăng trưởng.
  + Django và React có cộng đồng lớn, nhiều thư viện hỗ trợ.
* **Dễ bảo trì:**
  + Cấu trúc MVC của Django giúp tách biệt các thành phần, dễ dàng bảo trì.
  + React giúp quản lý giao diện một cách hiệu quả.
* **Hợp tác nhóm:**
  + Git và GitHub tạo điều kiện cho các thành viên trong nhóm làm việc cùng nhau.
* **Môi trường làm việc ổn định:**
  + Docker đảm bảo ứng dụng chạy ổn định trên các môi trường khác nhau.

## Kết quả đạt được

**7.2.1 Kết quả tổng quan:**

**Hoàn thiện chức năng:** Hệ thống đã cung cấp đầy đủ các tính năng cần thiết cho việc quản lý hồ sơ bệnh án.

**Tăng cường trải nghiệm người dùng:** Giao diện thân thiện, dễ sử dụng, tăng hiệu quả công việc.

**Bảo mật:** Đảm bảo an toàn cho dữ liệu nhạy cảm của bệnh nhân.

**Khả năng mở rộng:** Hệ thống có thể phát triển và thích ứng với nhu cầu thay đổi.

### Phân tích chi tiết các kết quả

#### 1. Hoàn thiện các chức năng chính

* **Đăng nhập, đăng xuất:** Quản lý quyền truy cập của từng người dùng, đảm bảo tính bảo mật.
* **Quản lý hồ sơ bệnh án:** Lưu trữ đầy đủ thông tin bệnh nhân, lịch sử khám chữa bệnh, kết quả xét nghiệm, ...
* **Tra cứu thông tin bệnh:** Cho phép nhanh chóng tìm kiếm thông tin bệnh nhân, lịch sử bệnh án, đơn thuốc, ...
* **Lưu trữ hình ảnh y khoa:** Lưu trữ các hình ảnh X-quang, MRI, ... giúp hỗ trợ chẩn đoán.
* **Báo cáo tiến trình điều trị:** Cung cấp các báo cáo thống kê về tình trạng bệnh nhân, hiệu quả điều trị, ...

#### 2. Tăng cường trải nghiệm người dùng

* **Giao diện thân thiện:** Thiết kế trực quan, dễ sử dụng, giảm thiểu lỗi thao tác.
* **Tốc độ phản hồi nhanh:** Hệ thống hoạt động ổn định, không xảy ra tình trạng giật lag.
* **Tìm kiếm thông tin dễ dàng:** Cung cấp các công cụ tìm kiếm mạnh mẽ, giúp người dùng nhanh chóng tìm thấy thông tin cần thiết.

#### 3. Bảo mật và bảo vệ dữ liệu

* **Mã hóa dữ liệu:** Bảo vệ thông tin nhạy cảm của bệnh nhân khỏi bị truy cập trái phép.
* **Quản lý quyền truy cập:** Chỉ cấp quyền truy cập cho những người có thẩm quyền.
* **Tuân thủ quy định:** Đảm bảo tuân thủ các quy định về bảo mật dữ liệu y tế.

#### 4. Khả năng mở rộng và bảo trì

* **Cấu trúc module:** Hệ thống được chia thành các module nhỏ, dễ dàng thay đổi và cập nhật.
* **Mã nguồn rõ ràng:** Dễ dàng đọc hiểu và sửa lỗi.
* **Kiểm thử tự động:** Giảm thiểu lỗi và đảm bảo chất lượng.

### ****7.2.3 Các yếu tố góp phần thành công****

* **Lựa chọn công nghệ phù hợp:** Sử dụng các công nghệ hiện đại, ổn định và có cộng đồng lớn.
* **Thiết kế kiến trúc hệ thống hợp lý:** Tách biệt các thành phần, dễ dàng bảo trì và mở rộng.
* **Đội ngũ phát triển có kinh nghiệm:** Đội ngũ có kiến thức chuyên sâu về phát triển phần mềm và lĩnh vực y tế.
* **Quá trình thử nghiệm kỹ lưỡng:** Đảm bảo hệ thống hoạt động ổn định và không có lỗi.

## 7.3 Hướng phát triển

Để đáp ứng nhu cầu ngày càng cao của lĩnh vực y tế, hệ thống cần được nâng cấp và phát triển thêm trong tương lai nhằm cung cấp các dịch vụ và tính năng tiên tiến hơn:

* **Nâng cao bảo mật**: Trong tương lai, chúng tôi sẽ tiếp tục tối ưu hóa hệ thống bảo mật, tích hợp thêm các công nghệ như xác thực hai yếu tố (2FA), bảo mật nâng cao thông qua AI để phát hiện các hành vi truy cập bất thường, đảm bảo an toàn tối đa cho dữ liệu bệnh nhân.
* **Tích hợp trí tuệ nhân tạo**: Để hỗ trợ tốt hơn cho các bác sĩ trong chẩn đoán bệnh, hệ thống có thể tích hợp các mô hình trí tuệ nhân tạo, giúp phân tích hình ảnh và các chỉ số y khoa để dự đoán và phát hiện sớm các bệnh da liễu.
* **Tăng cường khả năng tương tác và tư vấn**: Phát triển thêm các công cụ tư vấn từ xa, cho phép bệnh nhân có thể nhận tư vấn qua video, kết nối với các bác sĩ mà không cần đến tận nơi. Điều này đặc biệt hữu ích trong tình hình dịch bệnh hoặc với các bệnh nhân ở xa.
* **Mở rộng tính năng báo cáo và thống kê**: Hệ thống có thể được bổ sung tính năng báo cáo chi tiết về tiến trình điều trị, thống kê các loại bệnh và tỷ lệ thành công trong điều trị, giúp các bác sĩ và bệnh viện có cái nhìn tổng quan về hiệu quả của dịch vụ y tế.
* **Tối ưu hiệu suất và khả năng chịu tải**: Để đáp ứng lượng người dùng ngày càng tăng, hệ thống sẽ được tối ưu về mặt hiệu suất, cải thiện tốc độ xử lý và khả năng chịu tải của server. Điều này đảm bảo hệ thống hoạt động mượt mà ngay cả khi có nhiều người dùng truy cập cùng lúc.

### TÀI LIỆU THAM KHẢO

1. **Hồ sơ bệnh án điện tử trong y tế**Nguyễn, T. H. (2022). *Ứng dụng hồ sơ bệnh án điện tử trong quản lý bệnh viện*. Nhà xuất bản Y học.
2. **Phát triển hệ thống quản lý bệnh án**Lee, S. H., & Kim, J. (2020). "Development of an Electronic Health Record System for Dermatology Clinics." *Journal of Medical Informatics*, 15(4), 112-118. doi:10.1002/jmi.2020.0154.
3. **Tính bảo mật trong hệ thống y tế**Chen, L., & Zhang, X. (2021). "Security Challenges and Solutions in Healthcare Data Management Systems." *International Journal of Healthcare Information Systems and Informatics*, 8(2), 35-45. doi:10.4018/IJHISI.20210401.
4. **Ứng dụng AI trong chẩn đoán bệnh da liễu**Sharma, P., & Kumar, R. (2023). "Artificial Intelligence in Dermatology: A Step Towards Better Diagnosis." *Journal of Dermatology Research*, 17(3), 77-82. doi:10.1093/jder.2023.0176.
5. **Tạo dựng hệ thống web với Django**Williams, R. (2019). *Django for Web Development*. 3rd edition. O'Reilly Media.
6. **ReactJS và tối ưu hóa giao diện người dùng**Brown, A. (2020). *React.js Essentials: The Complete Guide for Modern Web Development*. Packt Publishing.
7. **Docker trong phát triển phần mềm**Jones, B., & Patel, K. (2018). *Docker: Up and Running*. O'Reilly Media.
8. **Quản lý cơ sở dữ liệu MySQL**MySQL Documentation Team. (2021). *MySQL 8.0 Reference Manual*. Oracle Corporation. Retrieved from<https://dev.mysql.com/doc/>
9. **Công nghệ bảo mật trong ứng dụng y tế**Chang, S. & Williams, M. (2020). "Healthcare Data Security: Techniques and Tools." *Cybersecurity for Healthcare Systems*, 5(1), 23-30.
10. **Tối ưu hóa hiệu suất hệ thống web**Johnson, M. (2017). "Optimizing Web Application Performance." *Web Performance Optimization*, 12(2), 47-56. Springer.