

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

**Hệ thống quản lý và phân tích chuyên sâu dữ liệu y
sinh của bệnh ung thư phổi**

VƯƠNG KIM HOÀNG

hoang.vk215584@sis.hust.edu.vn

Chương trình đào tạo: Kỹ thuật máy tính

Giảng viên hướng dẫn: TS. Nguyễn Hồng Quang

Chữ ký GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2025

LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành và sự tri ân sâu sắc đến tất cả các thầy cô trường Công nghệ thông tin và Truyền thông – trường Đại học Bách Khoa Hà Nội, đã giảng dạy và truyền đạt cho em những kiến thức vô cùng quý giá, giúp em có được một nền tảng vững chắc để thực hiện đồ án tốt nghiệp này cũng như bước tiếp trên con đường sự nghiệp trong tương lai. Đặc biệt, em xin cảm ơn thầy hướng dẫn Nguyễn Hồng Quang đã luôn đồng hành, chỉ bảo em trong suốt quá trình thực hiện đồ án. Em xin được cảm ơn gia đình, người thân và bạn bè đã luôn động viên, ủng hộ em trong suốt thời gian vừa qua. Lời cuối cùng, em muốn cảm ơn chính bản thân mình vì đã không ngừng cố gắng, giữ vững sự kiên trì và quyết tâm để hoàn thành đồ án này một cách trọn vẹn nhất.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Đề tài “Xây dựng hệ thống quản lý và phân tích dữ liệu y sinh của bệnh ung thư phổi” được thực hiện nhằm giải quyết nhu cầu cấp thiết trong việc tích hợp quản lý và phân tích dữ liệu y sinh liên quan đến căn bệnh ung thư phổi – một trong những căn bệnh có tỷ lệ tử vong cao hiện nay. Mặc dù đã có nhiều hướng tiếp cận nhằm xử lý và quản lý dữ liệu lâm sàng, các giải pháp hiện hành vẫn gặp phải những hạn chế nghiêm trọng về khả năng kết nối và thống nhất thông tin giữa các bộ phận chuyên môn, từ đó ảnh hưởng tiêu cực đến quá trình đưa ra quyết định chẩn đoán và điều trị chính xác. Trước thực trạng đó, đề tài lựa chọn hướng tích hợp dựa trên việc kết hợp công nghệ thông tin hiện đại và các phương pháp phân tích dữ liệu tiên tiến để khắc phục những bất cập đang tồn tại. Giải pháp được đề xuất trong đồ án bao gồm xây dựng một website có giao diện trực quan, cho phép thu thập, lưu trữ và xử lý dữ liệu từ nhiều nguồn đa dạng. Hệ thống còn ứng dụng các thuật toán học máy để khai thác những thông tin ẩn trong dữ liệu y sinh, từ đó tạo ra các báo cáo phân tích hỗ trợ hiệu quả cho công tác chẩn đoán và điều trị. Bên cạnh đó, các chức năng trực quan hóa dữ liệu được tích hợp nhằm giúp người sử dụng theo dõi và đánh giá mức độ tiến triển của bệnh một cách toàn diện, góp phần nâng cao hiệu quả trong việc quản lý và điều trị bệnh ung thư phổi. Đóng góp chính của đồ án nằm ở việc đề xuất một giải pháp toàn diện, không chỉ nâng cao hiệu quả quản lý dữ liệu mà còn cải thiện chất lượng phân tích y sinh, từ đó tạo tiền đề cho việc phát triển các phương pháp chẩn đoán và điều trị sớm. Kết quả đạt được là một hệ thống linh hoạt, đáng tin cậy và có khả năng mở rộng, góp phần tạo bước đột phá trong ứng dụng công nghệ thông tin vào lĩnh vực y tế hiện đại.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	5
2.1 Khảo sát hiện trạng	5
2.2 Tổng quan chức năng	9
2.2.1 Biểu đồ use case tổng quát	9
2.2.2 Biểu đồ use case phân rã "Quản lý bài báo khoa học"	10
2.2.3 Biểu đồ use case phân rã "Quản lý hồ sơ bệnh phẩm"	10
2.2.4 Biểu đồ use case phân rã "Quản lý file dữ liệu"	11
2.2.5 Biểu đồ use case phân rã "Xem thống kê tổng quan"	11
2.3 Đặc tả chức năng	12
2.3.1 Đặc tả use case "Xem thống kê tổng quan"	12
2.3.2 Đặc tả use case "Thêm bài báo khoa học".....	12
2.3.3 Đặc tả use case "Xem danh sách mẫu bệnh phẩm"	13
2.3.4 Đặc tả use case "Thêm mới file dữ liệu"	14
2.3.5 Đặc tả use case "Hỏi đáp với chatbot AI"	15
2.4 Yêu cầu phi chức năng	15
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	17
3.1 React	17
3.2 Tailwind CSS và Shadcn/UI.....	18
3.3 Express.js	19

3.4 MongoDB	20
3.5 Recharts.....	21
3.6 Supabase Storage	22
3.7 React Hook Form	23
3.8 React-window	24
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	26
4.1 Thiết kế kiến trúc.....	26
4.1.1 Lựa chọn kiến trúc phần mềm	26
4.1.2 Thiết kế tổng quan.....	27
4.1.3 Thiết kế chi tiết.....	27
4.2 Thiết kế chi tiết.....	30
4.2.1 Thiết kế giao diện	30
4.2.2 Thiết kế lớp	32
4.2.3 Thiết kế cơ sở dữ liệu	36
4.3 Xây dựng ứng dụng.....	38
4.3.1 Thư viện và công cụ sử dụng	38
4.3.2 Kết quả đạt được	39
4.3.3 Minh họa các chức năng chính	42
4.4 Kiểm thử.....	46
4.4.1 Kiểm thử chức năng "Thêm bài báo nghiên cứu"	46
4.4.2 Kiểm thử chức năng "Thêm dữ liệu bệnh phẩm"	47
4.4.3 Kiểm thử chức năng "Tải lên file dữ liệu"	47
4.5 Triển khai	48
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	49
5.1 Phương pháp Kaplan-Meier	49
5.2 Biểu đồ Kaplan-Meier.....	50

5.3 Chatbot Ollama	52
5.4 Nền tảng Streamlit	53
5.5 Bản đồ bộ gen ung thư TCGA	54
5.6 Docker.....	55
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	57
6.1 Kết luận	57
6.2 Hướng phát triển.....	58
TÀI LIỆU THAM KHẢO.....	60

DANH MỤC HÌNH VẼ

Hình 2.1 Giao diện tổng quan dữ liệu ung thư biểu mô tuyến phổi (TCGA, GDC) trên cBioPortal [1]	6
Hình 2.2 Giao diện tổng quan dữ liệu ung thư biểu mô tuyến phổi (TCGA - LUAD) trên GDC Data Portal [2]	7
Hình 2.3 Biểu đồ usecase tổng quát	9
Hình 2.4 Biểu đồ usecase phân rã "Quản lý bài báo khoa học"	10
Hình 2.5 Biểu đồ usecase phân rã "Quản lý hồ sơ bệnh phẩm"	10
Hình 2.6 Biểu đồ usecase phân rã "Quản lý file dữ liệu"	11
Hình 2.7 Biểu đồ usecase phân rã "Xem thống kê tổng quan"	11
Hình 4.1 Kiến trúc tổng quan	27
Hình 4.2 Kiến trúc thành phần User Service	28
Hình 4.3 Kiến trúc thành phần Research Service	28
Hình 4.4 Kiến trúc thành phần Record Service	29
Hình 4.5 Thiết kế giao diện "Xem thống kê"	31
Hình 4.6 Thiết kế giao diện "Xem danh sách bài báo"	31
Hình 4.7 Thiết kế giao diện "Thêm mới bài báo"	31
Hình 4.8 Thiết kế giao diện "Xem danh sách mẫu bệnh phẩm"	32
Hình 4.9 Thiết kế giao diện "Sửa dữ liệu mẫu bệnh phẩm"	32
Hình 4.10 Thiết kế giao diện "Quản lý file dữ liệu"	32
Hình 4.11 Thiết kế lớp UserController	33
Hình 4.12 Thiết kế lớp ResearchController	33
Hình 4.13 Thiết kế lớp RecordController	34
Hình 4.14 Thiết kế lớp FileController	35
Hình 4.15 Biểu đồ trình tự usecase "Thêm mới dữ liệu bệnh phẩm"	36
Hình 4.16 Giao diện trang đăng nhập	42
Hình 4.17 Giao diện trang thống kê	42
Hình 4.18 Giao diện biểu đồ Kaplan-Meier khi phóng to	43
Hình 4.19 Giao diện trang nghiên cứu	43
Hình 4.20 Giao diện trang danh sách mẫu bệnh phẩm	44
Hình 4.21 Giao diện trang chi tiết mẫu bệnh phẩm	44
Hình 4.22 Giao diện trang quản lý tệp dữ liệu	45
Hình 4.23 Giao diện trang xem file dữ liệu thô	45
Hình 4.24 Giao diện trang xem file ảnh xét nghiệm	46
Hình 4.25 Giao diện hỏi đáp với chatbot	46

Hình 5.1 Biểu đồ phân tích sống sót Kaplan-Meier theo nhóm giới tính và nhóm tuổi	51
Hình 5.2 Bảng so sánh mô hình Llama 3.1 405B với các mô hình khác [8]	53
Hình 5.3 Giao diện chatbot Ollama chạy trên Streamlit	54

DANH MỤC BẢNG BIỂU

Bảng 2.1	Bảng đặc tả usecase "Xem thông kê tổng quan"	12
Bảng 2.2	Bảng đặc tả usecase "Thêm bài báo nghiên cứu"	12
Bảng 2.3	Bảng đặc tả usecase "Xem danh sách mẫu bệnh phẩm"	13
Bảng 2.4	Bảng đặc tả usecase "Thêm mới file dữ liệu"	14
Bảng 2.5	Bảng đặc tả usecase "Hỏi đáp với chatbot AI"	15
Bảng 4.1	Mô tả các phương thức lớp UserController	33
Bảng 4.2	Mô tả các phương thức lớp ResearchController	34
Bảng 4.3	Mô tả các phương thức lớp RecordController	34
Bảng 4.4	Mô tả các phương thức lớp FileController	35
Bảng 4.8	Danh sách thư viện và công cụ sử dụng	38
Bảng 4.9	Thông kê thông tin của ứng dụng	40
Bảng 4.10	Thông kê thông tin thư mục backend	40
Bảng 4.11	Thông kê thông tin thư mục frontend	41
Bảng 4.12	Thông kê thông tin thư mục ollama3.1	41
Bảng 4.13	Mô tả các trường thông tin form thêm bài nghiên cứu	46
Bảng 4.14	Mô tả các trường thông tin form thêm dữ liệu bệnh phẩm	47
Bảng 4.15	Kết quả kiểm thử chức năng "Tải lên file dữ liệu"	47

DANH MỤC THUẬT NGỮ VÀ TỪ VIỆT TẮT

Thuật ngữ	Ý nghĩa
AI	Trí tuệ nhân tạo (Artificial Intelligence)
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CI/CD	Tích hợp, chuyển giao và triển khai liên tục (Continuous Integration/Continuous Delivery - Deployment)
CORS	Chia sẻ tài nguyên giữa các nguồn khác nhau (Cross-Origin Resource Sharing)
CSS	Ngôn ngữ dùng để định dạng giao diện và kiểu mẫu (Cascading Style Sheets)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
JSX	Phần mở rộng cú pháp của JavaScript (JavaScript XML)
LUAD	Ung thư biểu mô tuyến phổi (Lung Adenocarcinoma)
ORM	Ánh xạ quan hệ - đối tượng (Object-Relational Mapping)

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Ung thư phổi hiện đang là một trong những nguyên nhân hàng đầu gây tử vong do ung thư trên thế giới, với hơn 2 triệu ca mắc mới và gần 1,8 triệu ca tử vong mỗi năm. Tại Việt Nam, theo số liệu năm 2020, tỉ lệ mắc ung thư phổi xếp thứ 2 với 26.262 ca mắc mới chiếm 14.4% và 23.797 ca tử vong vì căn bệnh này. Quá trình chẩn đoán sớm, theo dõi tiến triển và đánh giá hiệu quả điều trị ung thư phổi đòi hỏi phải thu thập, tổng hợp và phân tích rất nhiều loại dữ liệu y sinh đa dạng—từ hình ảnh học, sinh hóa, xét nghiệm đột biến gen cho tới thông tin lâm sàng. Hiện nay, dữ liệu này thường được lưu trữ phân tán tại nhiều bệnh viện, phòng xét nghiệm và cơ sở y tế khác nhau, chưa có sự đồng bộ và thiếu công cụ chuyên sâu để tổng hợp, dẫn đến nguy cơ bỏ sót thông tin quan trọng, cũng như khó khăn trong việc so sánh kết quả giữa các giai đoạn điều trị hoặc nhóm bệnh nhân khác nhau.

Trong bối cảnh đó, việc xây dựng một hệ thống quản lý và phân tích chuyên sâu dữ liệu y sinh cho bệnh ung thư phổi trở nên vô cùng quan trọng. Thời gian xử lý và tổng hợp thông tin kéo dài không chỉ làm chậm trễ quyết định điều trị mà còn có thể khiến bác sĩ bỏ lỡ những dấu hiệu cảnh báo sớm của biến chứng hoặc tái phát. Đồng thời, thiếu khả năng phân tích dữ liệu lịch sử và dự báo tiến triển bệnh bằng các mô hình thống kê hay trí tuệ nhân tạo cũng khiến hiệu quả điều trị chưa được tối ưu, ảnh hưởng trực tiếp đến cơ hội sống sót của người bệnh.

Nếu bài toán quản lý và phân tích dữ liệu y sinh được giải quyết, lợi ích đem lại cho cộng đồng y tế và người bệnh sẽ rất lớn. Bác sĩ có thể truy cập nhanh chóng và đầy đủ thông tin xét nghiệm, hình ảnh và hồ sơ lâm sàng để đưa ra phác đồ điều trị kịp thời, chính xác hơn. Chuyên gia nghiên cứu y sinh cũng sẽ có nguồn dữ liệu lớn, đồng nhất để khai thác trong việc phát hiện chỉ dấu sinh học mới, xây dựng và hiệu chỉnh các mô hình dự báo tiến triển bệnh. Người bệnh sẽ được chẩn đoán và theo dõi sát hơn, nhờ đó giảm thiểu tối đa biến chứng và cải thiện cơ hội sống sót.

Với thiết kế mở, hệ thống không những phục vụ cho ung thư phổi mà còn có thể nhân rộng cho các loại ung thư khác như ung thư vú, ung thư đại tràng hoặc thậm chí các bệnh mạn tính như tim mạch, thận. Từ quy mô một bệnh viện, mô hình còn có khả năng mở rộng lên toàn bộ mạng lưới y tế, tạo tiền đề cho y tế thông minh và nghiên cứu đa trung tâm. Như vậy, việc tập trung giải quyết bài toán này không chỉ giúp nâng cao hiệu quả điều trị và chất lượng chăm sóc cho bệnh nhân ung thư phổi, mà còn đóng góp thiết thực vào việc phát triển khoa học y sinh và nền y tế hiện đại.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay trên thế giới có nhiều giải pháp hỗ trợ quản lý và phân tích dữ liệu y sinh phục vụ chẩn đoán và điều trị ung thư như hệ thống PACS (Picture Archiving and Communication System) chuyên lưu trữ và chia sẻ hình ảnh y khoa, các phần mềm LIMS (Laboratory Information Management System) quản lý kết quả xét nghiệm, cũng như một số nền tảng EMR (Electronic Medical Record) tích hợp thông tin lâm sàng. Bên cạnh đó, một số nghiên cứu ứng dụng trí tuệ nhân tạo cho ung thư phổi đã phát triển các mô-đun phân tích di truyền, nhận diện tổn thương trên ảnh CT hoặc xây dựng mô hình tiên lượng dựa vào dữ liệu lớn. Tuy nhiên, mỗi sản phẩm trên đều tập trung riêng lẻ vào một khía cạnh, chưa có giải pháp nào thực sự đồng bộ hóa toàn bộ chuỗi dữ liệu y sinh—từ hình ảnh, xét nghiệm sinh hóa và sinh học phân tử cho đến thông tin lâm sàng—và cung cấp công cụ phân tích chuyên sâu, thân thiện với bác sĩ lẫn nhà nghiên cứu.

Khi so sánh tổng quan, hệ thống PACS dù mạnh về luồng quản lý hình ảnh nhưng thiếu khả năng tích hợp với các chỉ số sinh hóa hay dữ liệu di truyền; LIMS quản lý xét nghiệm tốt nhưng phân tích đa chiều còn hạn chế; các nền tảng EMR thường hướng đến quản lý hồ sơ lâm sàng chung chung mà chưa tối ưu cho phân tích chuyên sâu ung thư phổi. Trong khi đó, các nghiên cứu AI mang tính thử nghiệm cao, chưa thực sự đóng gói thành sản phẩm dễ triển khai trong môi trường bệnh viện. Những công cụ này ít hỗ trợ trực quan hóa tiến triển bệnh theo thời gian, thiếu cơ chế cảnh báo sớm dựa trên tổng hợp đa nguồn dữ liệu và chưa cho phép người dùng (bác sĩ, nhà nghiên cứu) tùy biến bộ lọc, tham số phân tích theo nhu cầu thực tế.

Từ đánh giá trên, có thể thấy rõ hạn chế chính là sự phân tán dữ liệu và thiếu liên kết chặt chẽ giữa các loại dữ liệu y sinh, cũng như thiếu nền tảng phân tích tổng thể, trực quan và mở rộng cho nghiên cứu lẩn ứng dụng lâm sàng. Việc không có một hệ thống tập trung cho phép truy xuất, chuẩn hóa và phân tích đa chiều theo các tiêu chí chuyên sâu khiến quyết định lâm sàng đôi khi thiếu cơ sở dữ liệu toàn diện, đồng thời làm chậm trễ tiến trình nghiên cứu khoa học và triển khai phác đồ điều trị cá thể hóa.

Trên cơ sở đó, đề tài hướng đến mục tiêu chính là thiết lập một nền tảng tích hợp toàn diện, đồng bộ hóa và chuẩn hóa dữ liệu từ nhiều nguồn; cung cấp các chức năng chính bao gồm thu thập tự động, lưu trữ có cấu trúc, trực quan hóa tiến triển bệnh, phân tích thống kê và cảnh báo sớm bằng mô hình AI. Đột phá của đề tài là việc kết hợp linh hoạt giữa lưu trữ hình ảnh, dữ liệu lâm sàng và phân tử, đồng thời đưa vào các công cụ phân tích tùy chỉnh cho phép bác sĩ và nhà nghiên cứu khai

thác sâu theo từng kịch bản lâm sàng hoặc nghiên cứu khoa học. Phạm vi nghiên cứu tập trung vào dữ liệu y sinh của bệnh nhân ung thư phổi, song kiến trúc hệ thống được thiết kế mở để trong tương lai có thể mở rộng sang các loại ung thư khác hoặc các bệnh mạn tính đòi hỏi phân tích dữ liệu đa nguồn.

1.3 Định hướng giải pháp

Giải pháp tổng quan của hệ thống bao gồm hai tầng chính: tầng dữ liệu và tầng ứng dụng. Tầng dữ liệu được triển khai trên MongoDB với cấu trúc bảng chuẩn hóa lưu trữ thông tin bệnh nhân và các chỉ số phân tích. Ngoài ra, các file dữ liệu như ảnh khối u, kết quả xét nghiệm hay dữ liệu đa-omics được lưu trên nền tảng mã nguồn mở Supabase. Tầng ứng dụng cung cấp giao diện dashboard tích hợp sẵn các module vẽ biểu đồ Kaplan–Meier cho phân tích sống sót, biểu đồ histogram, biểu đồ tròn để thống kê các chỉ số lâm sàng theo tuần, tháng. Quy trình xử lý đưa từ nhập liệu, xử lý dữ liệu thô đến hiển thị kết quả trên giao diện được tự động hóa, giúp bác sĩ và nhà nghiên cứu dễ dàng tương tác, lọc, so sánh dữ liệu theo nhu cầu.

Đóng góp chính của đồ án nằm ở việc kết hợp đồng bộ cả luồng thu thập, lưu trữ, phân tích và hiển thị dữ liệu y sinh về ung thư phổi. Hệ thống không chỉ cho phép vẽ và hiển thị biểu đồ ngay trên giao diện mà còn tích hợp linh hoạt các biểu đồ thống kê đa chiều, giúp rút ngắn thời gian phân tích và ra quyết định lâm sàng. Việc tích hợp Chatbot Ollama giúp người dùng nhận được thông tin chính xác và nhanh chóng liên quan đến ung thư phổi. Cơ chế bảo mật OAuth2 kết hợp JSON Web Token (JWT) cho cơ chế xác thực và phân quyền để người dùng cơ bản chỉ có quyền xem dữ liệu, trong khi quản trị viên có toàn quyền quản lý và thao tác với toàn bộ hệ thống. Kết quả đạt được hiện tại là một phần mềm hoàn chỉnh với đầy đủ chức năng nhập liệu, quản lý người dùng, tạo báo cáo phân tích và xuất dữ liệu, sẵn sàng cho giai đoạn thử nghiệm lâm sàng và đánh giá hiệu quả.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau:

Chương 2 thực hiện khảo sát và phân tích yêu cầu hệ thống. Trong chương này, em tiến hành khảo sát các hệ thống quản lý và phân tích dữ liệu y sinh hiện nay, đặc biệt là các nền tảng hỗ trợ nghiên cứu ung thư phổi. Em lập bảng so sánh, phân tích ưu nhược điểm của các hệ thống tương tự, từ đó xác định các yêu cầu chức năng cần có trong hệ thống đề xuất. Ngoài ra, chương này cũng đặc tả một số chức năng chính và làm rõ các yêu cầu phi chức năng, đảm bảo hệ thống đáp ứng tốt nhu cầu thực tế.

Chương 3 trình bày chi tiết về các công nghệ được áp dụng trong đồ án, bao gồm framework frontend và backend, cơ sở dữ liệu, các thư viện hỗ trợ trực quan

hóa dữ liệu và những công cụ tối ưu hóa hiệu suất. Việc lựa chọn công nghệ được phân tích dựa trên tính phù hợp với đặc thù của hệ thống, nhằm đảm bảo tính ổn định, khả năng mở rộng và hiệu quả xử lý thông tin.

Chương 4 trình bày về thiết kế, triển khai và đánh giá hệ thống. Chương này bao gồm các bước thiết kế hệ thống như mô hình hóa cấu trúc, mô hình hóa hành vi, thiết kế kiến trúc tổng quan, thiết kế lớp, thiết kế cơ sở dữ liệu và giao diện người dùng. Sau khi thiết kế, hệ thống được triển khai thực tế và tiến hành kiểm thử nhằm đánh giá mức độ đáp ứng yêu cầu. Kết quả triển khai được phân tích để đưa ra nhận xét về tính khả thi và hiệu suất của hệ thống.

Chương 5 trình bày những giải pháp tối ưu được áp dụng trong hệ thống, bao gồm cải tiến hiệu suất, nâng cao khả năng xử lý dữ liệu và mở rộng tính năng hỗ trợ nghiên cứu y sinh. Đồng thời, những đóng góp nổi bật trong quá trình phát triển hệ thống cũng được tổng hợp nhằm làm rõ giá trị mà hệ thống mang lại cho cộng đồng nghiên cứu ung thư phổi.

Chương 6 tổng kết lại những nội dung đã thực hiện trong đồ án, đánh giá kết quả đạt được và đề xuất những hướng phát triển tiếp theo cho hệ thống. Dựa trên những phân tích và đánh giá từ quá trình triển khai, chương này đưa ra định hướng cải tiến trong tương lai nhằm nâng cao khả năng ứng dụng của hệ thống trong chẩn đoán sớm và điều trị ung thư phổi.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Chương 1 đã giới thiệu bối cảnh, mục tiêu và phạm vi nghiên cứu hệ thống quản lý và phân tích dữ liệu y sinh ung thư phổi, Chương 2 xuất hiện để làm rõ nhu cầu thực tế và yêu cầu chức năng, phi chức năng của hệ thống. Chương này bắt đầu bằng khảo sát các giải pháp quản lý và phân tích dữ liệu y sinh hiện có, tập trung vào những nền tảng hỗ trợ nghiên cứu ung thư phổi, đồng thời so sánh ưu, nhược điểm của từng hệ thống. Trên cơ sở đó tiến hành trình bày bảng tổng hợp và phân tích kết quả khảo sát, từ đó xác định tập hợp các yêu cầu chức năng cần có cho hệ thống đề xuất. Cuối cùng, chương này mô tả chi tiết một số chức năng chính và yêu cầu phi chức năng.

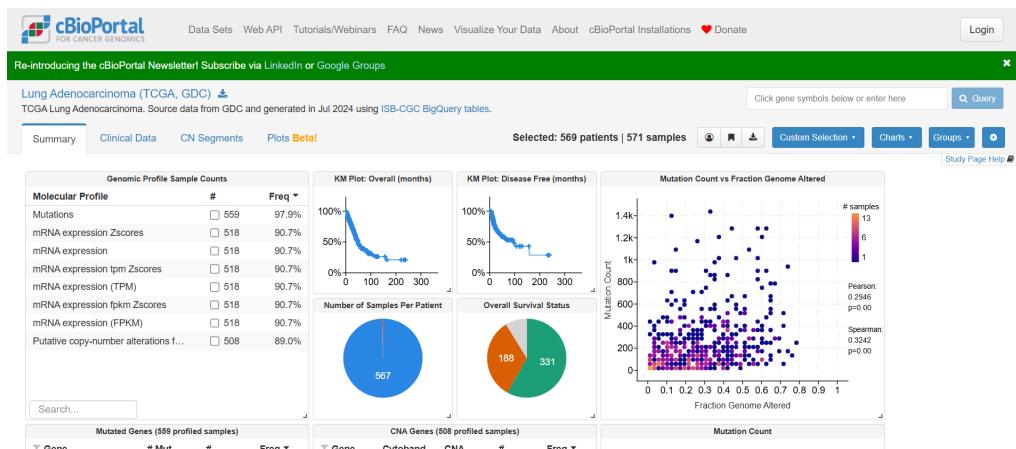
2.1 Khảo sát hiện trạng

Trước hết, ta cần làm rõ khái niệm về hệ thống quản lý và phân tích dữ liệu ung thư phổi là gì? Hệ thống này được định nghĩa là một giải pháp phần mềm tích hợp các công nghệ hiện đại, có nhiệm vụ thu thập, lưu trữ, phân tích cũng như trực quan hóa dữ liệu y sinh liên quan đến bệnh ung thư phổi. Các dữ liệu này bao gồm thông tin lâm sàng, kết quả xét nghiệm, hình ảnh y khoa và dữ liệu di truyền, giúp hỗ trợ cho quá trình chẩn đoán, theo dõi và nghiên cứu. Ở cấp độ của các tổ chức y tế và trung tâm nghiên cứu, hệ thống này không chỉ là công cụ xử lý dữ liệu mà còn đóng vai trò hỗ trợ quyết định trong điều trị và phát triển các chiến lược phòng ngừa hiệu quả.

Để đáp ứng mục tiêu đã nêu trong chương trước về việc thiết kế và xây dựng một hệ thống quản lý và phân tích dữ liệu ung thư phổi, cần phải tiến hành khảo sát các giải pháp phần mềm hiện nay nhằm tìm ra những đặc điểm và tính năng cốt lõi cần có. Quá trình khảo sát hiện trạng được thực hiện dựa trên ba nguồn chính: ý kiến đóng góp từ người dùng và các chuyên gia y sinh, phân tích các hệ thống đã có trên thị trường, cũng như so sánh với các ứng dụng tương tự được triển khai ở các quốc gia khác. Theo khảo sát từ phía người dùng, đặc biệt là các bác sĩ, chuyên gia nghiên cứu và quản trị viên y tế, họ mong đợi một giao diện người dùng trực quan, dễ thao tác và có khả năng phân tích dữ liệu đa chiều, nhằm hỗ trợ việc ra quyết định dựa trên các thông số phân tích chính xác và cập nhật. Trong khuôn khổ đồ án, em xin được khảo sát hai hệ thống cBioPortal for Cancer Genomics và GDC Data Portal.

cBioPortal là một nền tảng phần mềm mã nguồn mở được thiết kế đặc thù nhằm quản lý và phân tích dữ liệu ung thư, trong đó bao gồm dữ liệu di truyền, lâm sàng cũng như các thông tin liên quan đến bệnh nhân ung thư phổi. cBioPortal cung cấp

giao diện trực quan, cho phép người dùng—đặc biệt là các nhà nghiên cứu và bác sĩ chuyên môn—có thể nhanh chóng truy cập, khám phá và trực quan hóa dữ liệu từ các dự án nghiên cứu toàn cầu như TCGA. Nền tảng này được ứng dụng rộng rãi tại các trung tâm nghiên cứu và bệnh viện trên thế giới nhờ khả năng chuyển hóa các dữ liệu phức tạp thành thông tin hình ảnh dễ hiểu, hỗ trợ việc ra quyết định trong điều trị và nghiên cứu ung thư.



Hình 2.1: Giao diện tổng quan dữ liệu ung thư biểu mô tuyến phổi (TCGA, GDC) trên cBioPortal [1]

Nền tảng này có nhiều ưu điểm nổi bật, trước hết là (i) trực quan hóa dữ liệu mạnh mẽ, cung cấp một loạt các công cụ biểu diễn dữ liệu bao gồm các biểu đồ đường, biểu đồ phân tán, sơ đồ gene-centric và bảng điều khiển tương tác, giúp người dùng dễ dàng phát hiện các xu hướng cũng như mối liên hệ giữa các yếu tố y sinh quan trọng. Thứ hai là (ii) phân tích đa chiều, cụ thể hệ thống tích hợp các chức năng như so sánh nhóm, phân tích sống sót và hiển thị chi tiết đường thời gian điều trị của từng bệnh nhân, hỗ trợ hiệu quả cho việc nghiên cứu phân tích sâu. Thứ ba là (iii) quyền truy cập vào một kho dữ liệu khổng lồ từ các dự án gen ung thư hàng đầu như The Cancer Genome Atlas (TCGA) và The International Cancer Genome Consortium (ICGC). Thứ tư là (iv) mã nguồn mở và khả năng tùy chỉnh, điều này cũng thúc đẩy sự phát triển chung của cộng đồng và sự ra đời của các công cụ hỗ trợ như pyBioPortal trong Python để truy vấn dữ liệu một cách linh hoạt.

Tuy nhiên trong quá trình vận hành, hệ thống vẫn tồn tại một số hạn chế, đầu tiên là (i) sự không đồng nhất của dữ liệu do được tổng hợp từ nhiều nghiên cứu khác nhau, và quy trình xử lý dữ liệu có thể không hoàn toàn đồng nhất giữa các nghiên cứu. Điều này đòi hỏi người dùng phải cẩn trọng khi so sánh trực tiếp kết quả từ các bộ dữ liệu khác nhau. Thứ hai là (ii) do có đa dạng biểu đồ và công cụ phân tích mạnh mẽ nên giao diện nhìn chung khá phức tạp và không thân thiện đối với những người dùng không quen với các hệ thống phân tích dữ liệu chuyên sâu. Thứ ba là (iii) việc cài đặt, tùy chỉnh và cấu hình đòi hỏi kiến thức chuyên môn cao

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

về công nghệ thông tin cũng như hiểu biết sâu về kiến thức chuyên môn như cấu tạo gen và sinh học phân tử, điều này có thể là rào cản đối với người dùng không chuyên. Thứ tư là (iv) khi thực hiện các truy vấn phức tạp trên nhiều bộ dữ liệu lớn (ví dụ như ảnh chụp cắt lớp), hiệu suất của nền tảng có thể bị chậm lại.

GDC Data Portal là một nền tảng quan trọng do NCI Genomic Data Commons phát triển, cung cấp một môi trường truy cập, phân tích và trực quan hóa dữ liệu di truyền và lâm sàng từ các dự án nghiên cứu ung thư quy mô lớn như TCGA. Nền tảng này tập trung vào việc hợp nhất dữ liệu từ nhiều nguồn khác nhau, cho phép người dùng xây dựng các nhóm bệnh nhân tùy chỉnh (cohort) dựa trên các tiêu chí lâm sàng, sinh học và mẫu sinh, từ đó cung cấp cơ sở dữ liệu toàn diện để so sánh và phân tích xu hướng của các bệnh ung thư.

The screenshot shows the GDC Data Portal interface for the TCGA-LUAD project. At the top, there is a navigation bar with links for Video Guides, Send Feedback, Browse Annotations, Manage Sets, Cart, Login, and GDC Apps. A search bar contains the query "e.g. BRAF, Breast, TCGA-BLCA, TCGA-A5-ADG2". Below the navigation bar, the project name "PROJECT • TCGA-LUAD" is displayed, along with buttons for Save New Cohort, Biospecimen, Clinical, and Manifest. To the right, it shows a total of 585 cases, 36,215 files, and 3,944 annotations. The main content area is titled "SUMMARY" and includes a table with project details:

Project ID	TCGA-LUAD	Disease Type	• Acinar Cell Neoplasms • Adenomas and Adenocarcinomas • Cystic, Mucinous and Serous Neoplasms • Ductal and Lobular Neoplasms
dbGaP Study Accession	phs000178	Primary Site	Bronchus and lung
Project Name	Lung Adenocarcinoma	Program	TCGA

Below the summary, there are two sections: "CASES AND FILE COUNTS BY DATA CATEGORY" and "CASES AND FILE COUNTS BY EXPERIMENTAL STRATEGY".

Hình 2.2: Giao diện tổng quan dữ liệu ung thư biểu mô tụy phổi (TCGA - LUAD) trên GDC Data Portal [2]

Nền tảng này có nhiều ưu điểm nổi bật, trước hết là (i) dữ liệu được chuẩn hóa và có chất lượng cao, không chỉ lưu trữ dữ liệu mà còn xử lý lại các dữ liệu thô từ nhiều dự án khác nhau thông qua các quy trình tin sinh học chuẩn hóa. Thứ hai là (ii) tích hợp dữ liệu từ các chương trình gen ung thư lớn nhất, cung cấp một nguồn tài nguyên vô giá bao gồm dữ liệu gen, phiên mã, biểu sinh và lâm sàng của hàng chục nghìn bệnh nhân. Thứ ba là (iii) khả năng truy cập dữ liệu linh hoạt, người dùng có thể tìm kiếm, lọc và xây dựng các nhóm bệnh nhân (cohort) tùy chỉnh dựa trên các tiêu chí lâm sàng, sinh học hoặc dữ liệu phân tử. Dữ liệu có thể được tải xuống trực tiếp từ trình duyệt hoặc thông qua công cụ chuyên dụng. Thứ tư là (iv) hỗ trợ phân tích và trực quan hóa tích hợp. Cụ thể, nền tảng tích hợp các công cụ phân tích trực quan như biểu đồ tần suất đột biến, cụm phân tích biểu hiện gen, phân tích sống sót và các module trực quan hóa dữ liệu khác, giúp người dùng khai thác thông tin một cách hiệu quả.

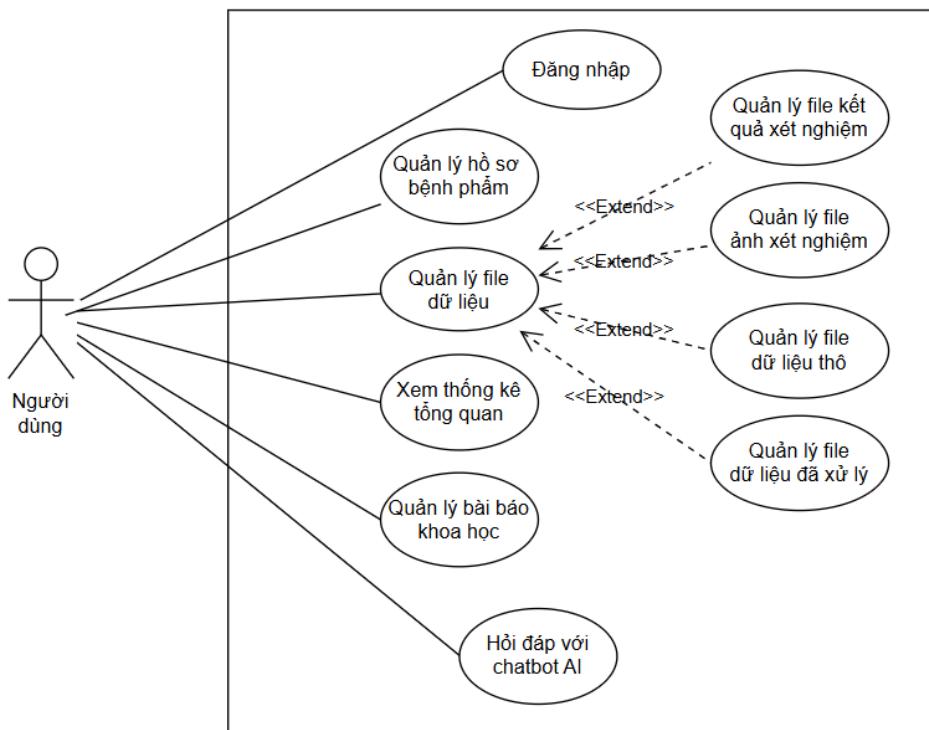
Bên cạnh những ưu điểm nêu trên, hệ thống cũng bộc lộ một số hạn chế trong quá trình sử dụng, đầu tiên là (i) việc xử lý dữ liệu tải về có thể gây khó khăn cho

người dùng mới. Dữ liệu tải về thường có cấu trúc thư mục và tên tệp phức tạp (sử dụng UUID - mã định danh duy nhất), đòi hỏi người dùng phải sử dụng tệp "Sample Sheet" đi kèm để ánh xạ tệp với các trường hợp bệnh nhân cụ thể. Thứ hai là (ii) không phải tất cả dữ liệu đều được truy cập công khai. Một số dữ liệu nhạy cảm và các bộ dữ liệu chi tiết đòi hỏi phải có quyền truy cập hoặc các bước kiểm duyệt, điều này làm hạn chế khả năng khai thác dữ liệu đối với người dùng ngoài cộng đồng nghiên cứu. Thứ ba là (iii) giới hạn về phân tích trực tuyến, các phân tích sâu và phức tạp hơn yêu cầu người dùng phải tải dữ liệu về và xử lý trên hệ thống của riêng mình.

Sau khi khảo sát hai hệ thống cBioPortal và GDC Data Portal, em nhận thấy rằng để xây dựng một hệ thống quản lý và phân tích dữ liệu y sinh về ung thư phổi, cần trang bị một số tính năng thiết yếu và quan trọng. Trước hết, hệ thống phải có khả năng tích hợp dữ liệu từ nhiều nguồn khác nhau, bao gồm dữ liệu di truyền, lâm sàng và hình ảnh y khoa, nhằm tạo ra một cơ sở dữ liệu toàn diện phục vụ cho quá trình phân tích chuyên sâu. Ngoài ra, khả năng trực quan hóa dữ liệu một cách rõ ràng và tương tác là yếu tố không thể thiếu; các công cụ biểu diễn như đồ thị, sơ đồ phân tán, các biểu đồ phân tích sống sót Kaplan-Meier, ... sẽ giúp người dùng nhanh chóng nắm bắt xu hướng, mối liên hệ giữa các chỉ số và đưa ra những nhận định chính xác về tình trạng ung thư. Hệ thống cũng cần tích hợp các công cụ phân tích nâng cao, cho phép so sánh cohort bệnh nhân, đánh giá các đột biến cụ thể và rút ra các kết luận định hướng cho phương pháp điều trị. Song song đó, việc đảm bảo bảo mật thông tin và quản lý quyền truy cập một cách chặt chẽ là điều cần thiết, nhằm bảo vệ tính nhạy cảm của dữ liệu y sinh. Hơn nữa, có thể tích hợp chatbot AI hỗ trợ giải đáp nhanh chóng và chính xác các vấn đề chuyên sâu. Cuối cùng, hệ thống cần có tính linh hoạt và khả năng mở rộng cao để dễ dàng cập nhật và tùy chỉnh theo những công trình khoa học mới cũng như nhu cầu nghiên cứu ngày càng đa dạng của người dùng.

2.2 Tổng quan chức năng

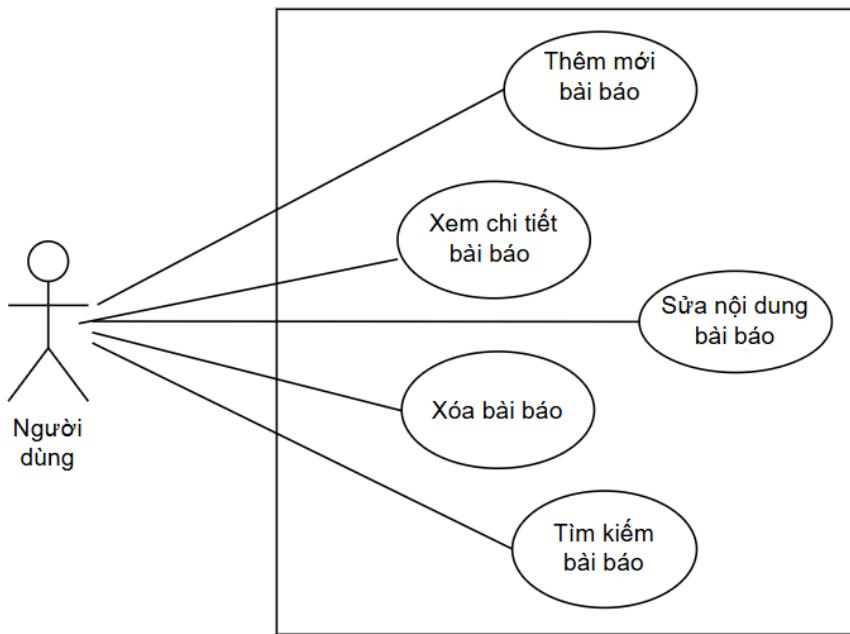
2.2.1 Biểu đồ use case tổng quát



Hình 2.3: Biểu đồ usecase tổng quát

Trong biểu đồ use case tổng quát, người dùng sau khi đăng nhập thành công, hệ thống tự động điều hướng đến trang thống kê tổng quan gồm các biểu đồ giúp trực quan hóa chỉ số y sinh như giai đoạn bệnh, thời gian sống sót tổng thể. Khi cần theo dõi hoặc cập nhật thông tin mẫu bệnh phẩm, người dùng chuyển sang trang quản lý hồ sơ bệnh phẩm. Quản lý tệp dữ liệu cho phép truy cập bốn loại tài liệu gồm kết quả xét nghiệm, hình ảnh chẩn đoán, dữ liệu thô và dữ liệu đã xử lý khi nhấn xem chi tiết một mẫu bệnh phẩm. Chức năng quản lý bài báo khoa học giúp người dùng quản lý và tìm kiếm các bài báo liên quan đến ung thư phổi, bao gồm tiêu đề, tác giả và tóm tắt, đồng thời liên kết đến nguồn tài liệu gốc. Chức năng hỏi đáp với chatbot AI hỗ trợ trả lời thắc mắc về ý nghĩa chỉ số y sinh và các thông tin liên quan.

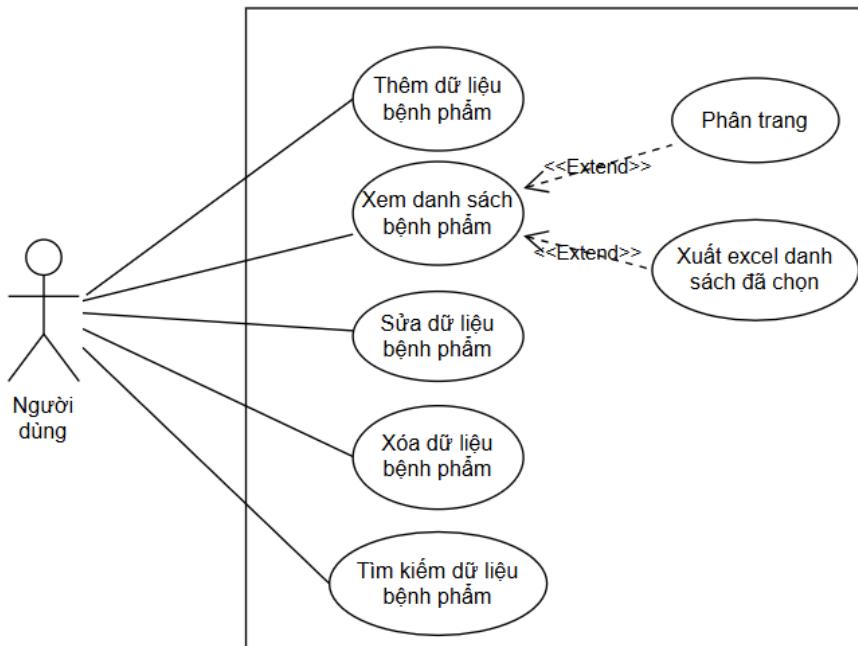
2.2.2 Biểu đồ use case phân rã "Quản lý bài báo khoa học"



Hình 2.4: Biểu đồ usecase phân rã "Quản lý bài báo khoa học"

Trong biểu đồ use case "Quản lý bài báo khoa học", người dùng có thể thực hiện đầy đủ các thao tác quản lý bài báo khoa học: thêm mới, xem chi tiết, sửa nội dung, xóa và tìm kiếm bài báo. Mỗi chức năng tương tác trực tiếp với hệ thống, cho phép người dùng cập nhật, tra cứu thông tin một cách thuận tiện và hiệu quả.

2.2.3 Biểu đồ use case phân rã "Quản lý hồ sơ bệnh phẩm"

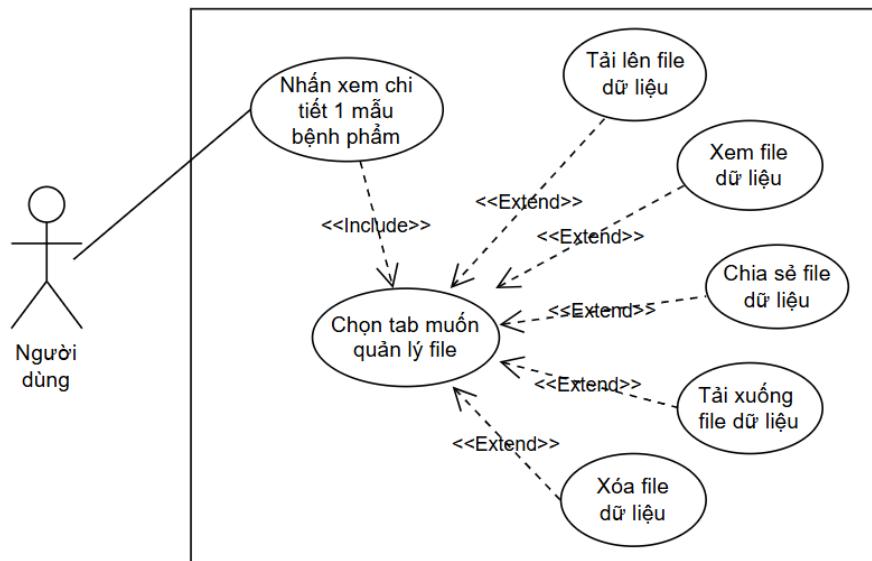


Hình 2.5: Biểu đồ usecase phân rã "Quản lý hồ sơ bệnh phẩm"

Trong biểu đồ use case "Quản lý hồ sơ bệnh phẩm", người dùng thực hiện các chức năng chính gồm thêm mới, xem danh sách dạng bảng, chỉnh sửa, xóa và tìm

kiểm dữ liệu bệnh phẩm. Ngoài ra, người dùng có thể lựa chọn số dòng muốn hiển thị (phân trang) và xuất excel danh sách đã chọn giúp trích xuất dữ liệu ra file báo cáo.

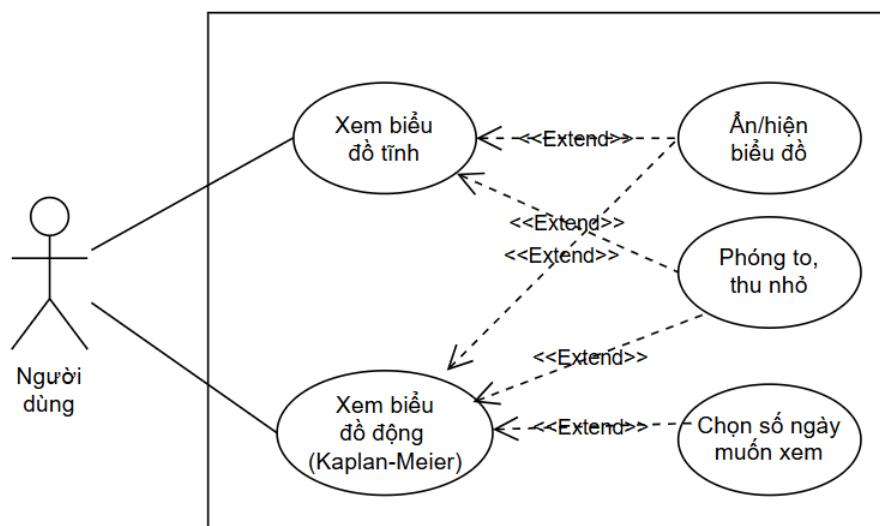
2.2.4 Biểu đồ use case phân rã "Quản lý file dữ liệu"



Hình 2.6: Biểu đồ usecase phân rã "Quản lý file dữ liệu"

Trong biểu đồ use case "Quản lý file dữ liệu", người dùng sau khi nhấn xem chi tiết một mẫu bệnh phẩm, sẽ chọn tab muốn quản lý file tương ứng. Khi chọn tab, hệ thống cho phép thực hiện các nghiệp vụ: tải lên, xem trực tiếp trên giao diện, chia sẻ, tải xuống và xóa file dữ liệu.

2.2.5 Biểu đồ use case phân rã "Xem thống kê tổng quan"



Hình 2.7: Biểu đồ usecase phân rã "Xem thống kê tổng quan"

Trong biểu đồ use case "Xem thống kê tổng quan", người dùng có thể xem biểu đồ tĩnh cũng như biểu đồ động với các tùy chọn mở rộng: ẩn/hiện biểu đồ để linh

hoạt hiển thị, và phóng to/thu nhỏ nhằm quan sát chi tiết các vùng dữ liệu. Ngoài ra, người dùng có thể xem biểu đồ động (Kaplan-Meier) với chức năng mở rộng cho phép chọn số ngày muốn xem để điều chỉnh khoảng thời gian phân tích.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case "Xem thống kê tổng quan"

Bảng 2.1: Bảng đặc tả usecase "Xem thống kê tổng quan"

Mã usecase	UC01	Tên usecase	Xem thống kê tổng quan
Tác nhân	Người dùng		
Mô tả	Công dụng cho phép người dùng xem biểu đồ thống kê dữ liệu trên hệ thống		
Tiền điều kiện	Người dùng đăng nhập thành công vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn tab "Thống kê"
	2	Hệ thống	Gửi truy vấn lên cơ sở dữ liệu để lấy dữ liệu tổng quan
	3	Hệ thống	Xử lý kết quả (tổng hợp số liệu, tính toán tỉ lệ, sắp xếp)
	4	Hệ thống	Hiển thị biểu đồ thống kê, bảng số liệu, biểu đồ Kaplan-Meier
	5	Người dùng	Quan sát, tùy ý lựa chọn ẩn/hiện hoặc phóng to biểu đồ
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Nếu không lấy được dữ liệu từ database, màn hình hiển thị "Không nhận được dữ liệu"
	5a	Người dùng	Nếu hệ thống không hiển thị biểu đồ, người dùng nhấn tải lại trang trên trình duyệt
Hậu điều kiện	Không		

2.3.2 Đặc tả use case "Thêm bài báo khoa học"

Bảng 2.2: Bảng đặc tả usecase "Thêm bài báo nghiên cứu"

Mã usecase	UC02	Tên usecase	Thêm bài báo nghiên cứu
Tác nhân	Người dùng		

Mô tả	Ca sử dụng cho phép người dùng thêm mới bài báo khoa học liên quan đến bệnh ung thư phổi		
Tiền điều kiện	Người dùng đăng nhập thành công vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn tab Nghiên cứu
	2	Người dùng	Nhấn nút "Thêm bài báo"
	3	Hệ thống	Hiển thị hộp thoại chứa form để nhập thông tin
	4	Người dùng	Nhấn nút "Thêm" sau khi điền đầy đủ thông tin
	5	Hệ thống	Kiểm tra tính hợp lệ của dữ liệu
Luồng sự kiện thay thế	6	Hệ thống	Thông báo thêm bài báo thành công nếu dữ liệu được thêm vào CSDL
	STT	Thực hiện bởi	Hành động
Luồng sự kiện thay thế	5a	Hệ thống	Thông báo "Vui lòng điền vào trường này" khi người dùng nhập thiếu thông tin
Hậu điều kiện	Không		

2.3.3 Đặc tả use case "Xem danh sách mẫu bệnh phẩm"

Bảng 2.3: Bảng đặc tả usecase "Xem danh sách mẫu bệnh phẩm"

Mã usecase	UC03	Tên usecase	Xem danh sách mẫu bệnh phẩm
Tác nhân	Người dùng		
Mô tả	Ca sử dụng cho phép người dùng xem bảng danh sách mẫu bệnh phẩm trong hệ thống		
Tiền điều kiện	Người dùng đăng nhập thành công vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn tab "Hồ sơ"
	2	Hệ thống	Gửi truy vấn lên cơ sở dữ liệu để lấy toàn bộ dữ liệu
	3	Hệ thống	Trả về danh sách dạng bảng kèm phân trang và các thao tác chỉnh sửa dữ liệu (xem, sửa, xóa)
	4	Người dùng	Tìm kiếm dữ liệu nhanh hơn theo mã mẫu bệnh phẩm
	STT	Thực hiện bởi	Hành động
Luồng sự kiện thay thế			

	3a	Hệ thống	Nếu không có dữ liệu hoặc lấy dữ liệu thất bại, hệ thống trả về thông báo "Không có dữ liệu"
	3b	Người dùng	Tùy ý chọn số dòng muốn hiển thị trong một trang
	3c	Người dùng	Có thể chọn nhiều dòng để xuất dữ liệu ra file Excel
Hậu điều kiện	Không		

2.3.4 Đặc tả use case "Thêm mới file dữ liệu"

Bảng 2.4: Bảng đặc tả usecase "Thêm mới file dữ liệu"

Mã usecase	UC04	Tên usecase	Thêm mới file dữ liệu
Tác nhân	Người dùng		
Mô tả	Ca sử dụng cho phép người dùng thêm các loại file dữ liệu (ảnh, báo cáo cho từng mẫu bệnh phẩm)		
Tiền điều kiện	Người dùng đang xem chi tiết 1 mẫu bệnh phẩm		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn 1 trong 4 tab Báo cáo bệnh lý, Ảnh khôi u, Dữ liệu thô và Dữ liệu đã xử lý
	2	Hệ thống	Trả về danh sách file nếu có
	3	Người dùng	Kéo thả hoặc nhập vào khung để chọn file muốn thêm
	4	Hệ thống	Tự động đẩy file kèm mã mẫu bệnh phẩm lên bucket tương ứng (đã tạo trước) của nền tảng Supabase
	5	Hệ thống	Hiển thị file đã thêm trong bảng bên dưới kèm theo các thao tác với file (xem, tải xuống, chia sẻ, xóa)
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Trả về lỗi "Kích thước file quá lớn" đối với file có dung lượng lớn hơn 50MB
	4b	Hệ thống	Trả về lỗi "Không thể tải tệp lên" đối với file có tên chứa dấu cách hoặc không đúng định dạng

Hậu điều kiện	Không
----------------------	-------

2.3.5 Đặc tả use case "Hỏi đáp với chatbot AI"

Bảng 2.5: Bảng đặc tả usecase "Hỏi đáp với chatbot AI"

Mã usecase	UC05	Tên usecase	Hỏi đáp với chatbot AI
Tác nhân	Người dùng		
Mô tả	Ca sử dụng cho phép người dùng đặt câu hỏi và nhận câu trả lời tự động từ chatbot AI được tích hợp trong hệ thống		
Tiền điều kiện	Người dùng đăng nhập thành công vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn tab "Chatbot AI"
	2	Hệ thống	Kết nối tới URL chatbot và hiển thị trên giao diện
	3	Người dùng	Nhập câu hỏi vào khung chat và nhấn "Gửi"
	4	Hệ thống	Gửi truy vấn tới model chatbot AI
	5	Chatbot AI	Xử lý truy vấn, tìm câu trả lời từ mô hình ngôn ngữ
	6	Chatbot AI	Trả về kết quả cho hệ thống
	7	Hệ thống	Hiển thị câu trả lời lên giao diện chat
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Nếu không thể kết nối tới URL chatbot, màn hình hiển thị "Lỗi tải chatbot. Kiểm tra lại kết nối"
	7a	Hệ thống	Nếu đột ngột mất kết nối hoặc hỏi sai chủ đề, hiển thị thông báo lỗi tương ứng
Hậu điều kiện	Không		

2.4 Yêu cầu phi chức năng

Hệ thống cần hỗ trợ xác thực sử dụng JWT, đồng thời áp dụng phân quyền chi tiết theo vai trò cho người dùng và quản trị viên. Ngoài ra phần mềm cần tuân thủ các quy định về bảo vệ dữ liệu y tế và hỗ trợ cơ chế xóa hoặc ẩn danh dữ liệu cá nhân theo yêu cầu bệnh nhân.

Để đảm bảo trải nghiệm mượt mà, giao diện chính phải hiển thị trong 2 giây và thời gian phản hồi của hệ thống tối đa 2s với tập dữ liệu dưới 10.000 bản ghi.

Thiết kế giao diện tuân thủ nguyên tắc material hoặc phương pháp thiết kế nguyên tử, với bộ màu sắc nhất quán và độ tương phản phù hợp. Các form nhập liệu, bảng dữ liệu và biểu đồ tương tác phải hỗ trợ tính năng lọc, sắp xếp, thu phóng một cách trực quan. Giao diện phải phản hồi nhanh và hiển thị tốt trên máy tính lẫn tablet. Ngoài ra, hệ thống cần trang bị chú thích và hướng dẫn tại chỗ cho những trường và biểu đồ phức tạp.

Sử dụng kiến trúc microservices giúp tách biệt module quản lý người dùng, nhập liệu và phân tích thành các dịch vụ độc lập, dễ dàng phân bổ tài nguyên và phát triển song song. Để đảm bảo chất lượng và ổn định lâu dài, mã nguồn cần tuân thủ chuẩn coding style (ESLint/Prettier). Hệ thống vẫn đang trong giai đoạn phát triển và mở rộng nên cần thiết kế để dễ dàng sửa đổi và nâng cấp sao cho phù hợp mỗi khi có yêu cầu mới.

Chương 2 đã khảo sát thực trạng các hệ thống quản lý và phân tích dữ liệu y sinh, đặc biệt là trong nghiên cứu ung thư phổi, đồng thời đã xác định rõ các yêu cầu chức năng yếu và yêu cầu phi chức năng. Qua phân tích so sánh, những tính năng như tìm kiếm dữ liệu, quản lý thông tin xét nghiệm, bảo mật dữ liệu và khả năng mở rộng hệ thống đóng vai trò không thể thiếu. Trên cơ sở này, kiến trúc về yêu cầu nghiệp vụ và phi nghiệp vụ đã chuẩn bị để làm đầu vào cho việc lựa chọn công nghệ và nền tảng, nội dung sẽ được trình bày chi tiết trong Chương 3.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Dựa trên các yêu cầu đã xác định trong Chương 2, Chương 3 trình bày lý do lựa chọn và phân tích các công nghệ, nền tảng quan trọng để hiện thực hóa hệ thống. Chương này sẽ đi sâu vào việc giới thiệu framework và nền tảng lưu trữ dữ liệu, thư viện trực quan hóa và các công cụ tối ưu hóa hiệu suất, giải thích rõ mỗi công nghệ giải quyết vấn đề cụ thể nào của hệ thống hiện tại. Ngoài ra còn so sánh với những hướng tiếp cận thay thế và đưa ra lý do lựa chọn công nghệ đó.

3.1 React

React là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook vào năm 2013 và hiện được duy trì bởi một cộng đồng các nhà phát triển trên thế giới. Thư viện này được thiết kế để xây dựng các giao diện người dùng động, đặc biệt phù hợp với các ứng dụng đơn trang (Single-Page Applications). React nổi bật với kiến trúc dựa trên thành phần, cho phép các nhà phát triển chia nhỏ giao diện thành các thành phần độc lập, tái sử dụng, giúp quản lý mã nguồn dễ dàng và hiệu quả. Một tính năng quan trọng của React là Virtual DOM, một biểu diễn ảo của DOM thực tế, giúp tối ưu hóa hiệu suất bằng cách chỉ cập nhật những phần giao diện thay đổi thay vì tải lại toàn bộ trang. Điều này đặc biệt hữu ích cho các ứng dụng yêu cầu cập nhật thường xuyên, chẳng hạn như hiển thị dữ liệu thời gian thực hoặc xử lý tương tác người dùng. React cũng hỗ trợ JSX, một cú pháp mở rộng của JavaScript, cho phép viết mã HTML ngay trong JavaScript, giúp mã nguồn trực quan và dễ bảo trì hơn.

Trong hệ thống này, React được sử dụng để đáp ứng nhiều yêu cầu quan trọng về giao diện và chức năng. Về giao diện động và tương tác, React tận dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện, chỉ render lại những phần thay đổi, mang lại trải nghiệm người dùng mượt mà. Điều này đặc biệt quan trọng cho các tính năng như cập nhật dữ liệu thời gian thực hoặc các yếu tố tương tác như thanh tìm kiếm và bảng dữ liệu, vốn cần phản hồi nhanh chóng mà không phải tải lại toàn bộ trang. Về quản lý trạng thái, React cung cấp các công cụ như Context API hoặc Redux, cho phép quản lý trạng thái tập trung, giúp xử lý logic trạng thái phức tạp và đảm bảo sự nhất quán giữa các thành phần, đặc biệt khi nhiều thành phần cần chia sẻ hoặc phản ứng với cùng một trạng thái. Về tái sử dụng mã nguồn, kiến trúc dựa trên thành phần của React cho phép tạo các thành phần có thể tái sử dụng, như nút bấm, thanh điều hướng hoặc biểu mẫu, giúp giảm thiểu lặp lại mã và tăng tính dễ bảo trì. Cuối cùng, về tích hợp với API backend, React hoạt động tốt với các thư viện HTTP như Axios hoặc Fetch API, hỗ trợ giao tiếp dễ dàng với các dịch vụ

backend để lấy hoặc gửi dữ liệu, đáp ứng các tính năng như xác thực người dùng, lấy dữ liệu hiển thị và gửi biểu mẫu.

Ngoài React, các công nghệ frontend phổ biến khác bao gồm Angular và Vue.js. Angular là một framework toàn diện do Google phát triển, cung cấp nhiều tính năng tích hợp sẵn như routing, form validation và dependency injection. Tuy nhiên, Angular có cấu trúc phức tạp hơn, điều này có thể làm tăng thời gian phát triển ban đầu. Vue.js là một framework cải tiến, dễ học và linh hoạt, thích hợp cho các dự án nhỏ đến trung bình. Tuy nhiên, Vue có cộng đồng nhỏ hơn so với React và Angular, điều này có thể ảnh hưởng đến việc tìm kiếm tài liệu và hỗ trợ.

Với những ưu điểm trên, React là công nghệ cốt lõi để xây dựng giao diện người dùng, đáp ứng các yêu cầu về giao diện động, quản lý trạng thái và tích hợp API.

3.2 Tailwind CSS và Shadcn/UI

Tailwind CSS là một framework CSS mã nguồn mở, áp dụng triết lý "utility-first" — tức là cung cấp các lớp tiện ích nhỏ gọn, mỗi lớp đảm nhận một chức năng cụ thể như căn lề, định dạng chữ, màu sắc, kích thước. Thay vì sử dụng các thành phần được định dạng sẵn như trong Bootstrap, Tailwind cho phép nhà phát triển xây dựng giao diện bằng cách kết hợp các lớp tiện ích trực tiếp trong HTML, mang lại sự linh hoạt và khả năng tùy biến cao. Một số tính năng nổi bật của Tailwind CSS bao gồm khả năng tùy chỉnh cao thông qua tệp cấu hình tailwind.config.js, hỗ trợ thiết kế responsive với các breakpoint linh hoạt, và chế độ Just-In-Time (JIT) giúp tạo ra CSS tối ưu bằng cách chỉ sinh ra các lớp được sử dụng trong dự án. Những đặc điểm này giúp giảm kích thước tệp CSS cuối cùng và cải thiện hiệu suất tải trang.

Song song với Tailwind CSS, thư viện shadcn/ui được áp dụng để cung cấp các thành phần giao diện được thiết kế sẵn với phong cách tối giản, hiện đại và dễ tùy biến. Shadcn/ui được xây dựng dựa trên nền tảng của Tailwind CSS, cho phép tích hợp sát sao các lớp tiện ích và đảm bảo giao diện của ứng dụng không chỉ đẹp mắt mà còn duy trì được sự đồng bộ trong cách trình bày. Sự kết hợp của hai công nghệ này mang đến hiệu quả vượt trội, giúp hệ thống hiển thị các dữ liệu phân tích y sinh—bao gồm bảng số liệu, biểu đồ và các thành phần tương tác khác—một cách mượt mà, rõ ràng và hiện đại, hỗ trợ người dùng trong việc tiếp cận và phân tích thông tin. [3]

Nhằm tìm kiếm giải pháp tối ưu cho việc xây dựng giao diện, các lựa chọn thay thế như Bootstrap, Material-UI hay Ant Design cũng được cân nhắc. Bootstrap nổi tiếng với việc cung cấp một hệ thống lưới và giao diện mặc định phong phú, nhưng lại gặp hạn chế về tính linh hoạt do các kiểu dáng mặc định khó tùy chỉnh để phù

hợp với giao diện chuyên biệt của ứng dụng y sinh. Material-UI mang đến giao diện theo phong cách thiết kế của Google với khả năng trình bày trực quan và thu hút, tuy nhiên, cấu hình và tùy biến của nó đôi khi khá phức tạp khi cần phải điều chỉnh cho phù hợp với các yêu cầu phân tích dữ liệu cụ thể. Ant Design cung cấp một bộ thành phần mạnh mẽ với đầy đủ tính năng, nhưng lại đòi hỏi việc tuỳ chỉnh chi tiết khá tốn thời gian để có thể đáp ứng được tiêu chí hiện đại và chuyên sâu của một hệ thống y sinh. Trong khi đó, sự phối hợp giữa Tailwind CSS và shadcn/ui không chỉ giúp rút ngắn thời gian phát triển mà còn tạo ra một giao diện "headless", tức là giao diện chỉ tập trung vào việc trình bày dữ liệu theo cách mà người dùng cần, mà không bị ràng buộc bởi các kiểu dáng mặc định cứng nhắc, từ đó đem lại trải nghiệm mượt mà, dễ dàng tùy biến và phù hợp với toàn bộ yêu cầu của đề tài nghiên cứu ung thư phổi.

Với những ưu điểm vượt trội về sự linh hoạt, dễ dàng tùy chỉnh và tích hợp, Tailwind CSS kết hợp với shadcn/ui đã khẳng định vai trò quan trọng trong việc xây dựng giao diện của hệ thống. Giải pháp này không chỉ giúp tối ưu hóa tốc độ phát triển giao diện mà còn đảm bảo hệ thống có một thiết kế hiện đại, trực quan và phù hợp với yêu cầu phân tích chuyên sâu dữ liệu y sinh, từ đó tạo điều kiện thuận lợi cho quá trình nghiên cứu và ra quyết định trong lĩnh vực ung thư phổi.

3.3 Express.js

Express.js là một framework website dành cho Node.js, được sử dụng như thành phần cốt lõi trong xây dựng backend. Trong hệ thống này, Express.js giúp giải quyết bài toán thiết kế và tổ chức các API phục vụ cho việc quản lý mẫu bệnh phẩm, bài báo khoa học và thông tin đăng nhập của người dùng. Cụ thể, hệ thống yêu cầu xây dựng một tập hợp các endpoint RESTful hỗ trợ đầy đủ các chức năng như thêm, sửa và tìm kiếm dữ liệu. Express.js với hệ thống định tuyến linh hoạt cho phép định nghĩa rõ ràng từng tuyến đường và phương thức HTTP phù hợp cho từng chức năng. Nhờ đó, các thao tác như thêm mới mẫu bệnh phẩm, cập nhật bài báo khoa học hay tìm kiếm kết quả xét nghiệm đều được tách biệt về mặt logic, giúp mã nguồn trở nên trong sáng, dễ đọc và dễ bảo trì.

Bên cạnh đó, Express.js còn đóng vai trò quan trọng trong việc xử lý các lớp middleware cho xác thực và phân quyền người dùng. Dữ liệu y sinh trong hệ thống là dữ liệu mật, do đó cần có các lớp kiểm tra bảo mật để đảm bảo chỉ những người dùng hợp lệ mới có quyền truy cập và chỉnh sửa thông tin. Express.js hỗ trợ cơ chế middleware theo chuỗi, giúp dễ dàng triển khai các lớp kiểm tra như xác thực token JWT, xác định danh tính người dùng và kiểm tra quyền hạn tương ứng của từng tài khoản. Việc triển khai các middleware bảo mật này đảm bảo rằng mọi yêu cầu gửi

đến hệ thống đều được kiểm tra kỹ lưỡng trước khi được phép xử lý tiếp theo, từ đó duy trì tính toàn vẹn và bảo mật của dữ liệu. Hệ thống phải thường xuyên giao tiếp với cơ sở dữ liệu để truy xuất và cập nhật thông tin, do vậy với nền tảng Node.js dựa trên cơ chế event loop và mô hình non-blocking I/O, Express.js giúp hệ thống có thể xử lý đồng thời nhiều yêu cầu mà không bị nghẽn hoặc chậm trễ. Điều này đặc biệt quan trọng khi hệ thống phải phục vụ nhiều người dùng cùng truy cập dữ liệu y sinh hoặc cập nhật hàng loạt kết quả nghiên cứu một cách liên tục, nhưng vẫn duy trì được hiệu suất phản hồi nhanh và ổn định.

Để thay thế Express.js, có một số lựa chọn khác như Fastify, Koa hoặc NestJS. Fastify là một framework hiệu suất cao, phù hợp với các ứng dụng cần xử lý lượng lớn yêu cầu một cách tối ưu nhất. Koa, do nhóm phát triển Express.js tạo ra, mang đến một thiết kế tối giản hơn với khả năng xử lý middleware theo phong cách `async/await`, giúp cải thiện tính bảo trì và dễ đọc của mã nguồn. NestJS là một lựa chọn mạnh mẽ hơn với thiết kế dựa trên mô hình modular, phù hợp cho các dự án lớn cần tính tổ chức cao. Tuy nhiên, Express.js vẫn được ưu tiên trong đồ án này vì sự phổ biến, tài liệu phong phú và cộng đồng lớn, giúp giảm thời gian phát triển và đảm bảo hệ thống có sự hỗ trợ tốt từ nguồn tài nguyên mở.

Như vậy việc ứng dụng Express.js đã giải quyết đồng thời nhiều yêu cầu kỹ thuật quan trọng như tổ chức API một cách rõ ràng, đảm bảo bảo mật thông qua middleware, tối ưu hiệu năng bất đồng bộ, chuẩn hóa xử lý lỗi và tổ chức mã nguồn có cấu trúc khoa học. Điều này tạo nên một nền tảng backend hỗ trợ hiệu quả cho việc phát triển và vận hành hệ thống hiện tại.

3.4 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến, được thiết kế để xử lý dữ liệu linh hoạt và hiệu quả trong các ứng dụng hiện đại. Với khả năng lưu trữ dữ liệu dưới dạng tài liệu JSON linh hoạt, MongoDB giúp hệ thống dễ dàng quản lý và truy vấn các thông tin liên quan đến bệnh nhân, kết quả xét nghiệm, hình ảnh y khoa và các chỉ số phân tích gen mà không bị ràng buộc bởi cấu trúc dữ liệu cố định như các hệ cơ sở dữ liệu quan hệ. Điều này đặc biệt quan trọng trong nghiên cứu ung thư phổi, nơi dữ liệu thường có sự biến đổi lớn tùy theo từng bệnh nhân, phương pháp điều trị và quá trình tiến triển của bệnh.

MongoDB hỗ trợ khả năng mở rộng theo chiều ngang, cho phép hệ thống xử lý khối lượng dữ liệu y sinh lớn và phức tạp một cách hiệu quả. Đặc biệt, với các nghiên cứu chuyên sâu liên quan đến phân tích gen, phân tích hình ảnh tế bào, và dự đoán tiến triển bệnh dựa trên dữ liệu lịch sử, MongoDB giúp tối ưu quá trình lưu trữ và truy vấn, đồng thời cung cấp cơ chế index thông minh để tăng tốc độ tìm

kiểm thông tin quan trọng.

Một số công nghệ thay thế có thể được xem xét cho đề tài này bao gồm PostgreSQL, Neo4j, và Firebase Firestore. PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, phù hợp với các mô hình dữ liệu có tính toàn vẹn cao và cần xử lý các truy vấn phức tạp. Neo4j là một hệ cơ sở dữ liệu đồ thị, tối ưu cho việc phân tích mối quan hệ giữa các yếu tố y sinh như sự tương tác của các loại gen và tác động của môi trường lên sự phát triển ung thư. Firebase Firestore lại là một giải pháp NoSQL đám mây, phù hợp với các hệ thống thời gian thực, nơi dữ liệu bệnh nhân và kết quả xét nghiệm cần được cập nhật liên tục. Tuy nhiên, MongoDB được lựa chọn trong đồ án này vì sự linh hoạt, khả năng mở rộng, cũng như sự hỗ trợ tốt cho các mô hình dữ liệu động và phức tạp trong lĩnh vực y sinh học.

Nhờ những ưu điểm trên, MongoDB đóng vai trò quan trọng trong việc xây dựng nền tảng dữ liệu của hệ thống, hỗ trợ phân tích chuyên sâu để đưa ra các thông tin hữu ích trong việc nghiên cứu và điều trị ung thư phổi.

3.5 Recharts

Recharts là một thư viện đồ thị mạnh mẽ dựa trên React, giúp tạo ra các biểu đồ tương tác một cách linh hoạt và hiệu quả, đóng vai trò quan trọng trong việc hiển thị dữ liệu thống kê, kết quả xét nghiệm, và sự tiến triển của bệnh ung thư phổi. Thư viện này cung cấp một bộ thành phần React có thể tái sử dụng, cho phép các nhà phát triển tạo ra các biểu đồ tương tác một cách dễ dàng và hiệu quả. Recharts được sử dụng để trực quan hóa dữ liệu y sinh, giúp các nhà nghiên cứu và bác sĩ dễ dàng phân tích xu hướng, so sánh kết quả và rút ra các kết luận quan trọng từ dữ liệu bệnh nhân. [4]

Với khả năng hỗ trợ nhiều loại biểu đồ như biểu đồ đường, cột, tròn và khu vực, Recharts giúp hệ thống dễ dàng minh họa các mối quan hệ giữa các chỉ số y sinh. Ví dụ, thông qua biểu đồ đường, hệ thống có thể trực quan hóa sự thay đổi của một chỉ số sinh học theo thời gian, hoặc sử dụng biểu đồ cột để so sánh các phương pháp điều trị khác nhau trên tập dữ liệu bệnh nhân. Điều này không chỉ giúp các nhà nghiên cứu nắm bắt thông tin nhanh hơn mà còn hỗ trợ quá trình ra quyết định dựa trên dữ liệu chính xác.

Một số thư viện thay thế có thể được xem xét gồm Chart.js, D3.js, và Victory. Chart.js là một thư viện đơn giản và dễ sử dụng, phù hợp cho các biểu đồ thông kê với dữ liệu không quá phức tạp. D3.js là một công cụ mạnh mẽ hơn, cho phép tạo ra các biểu đồ tùy chỉnh với độ linh hoạt cao, nhưng cần nhiều công sức phát triển hơn. Victory là một thư viện React chuyên biệt, dễ tích hợp với các ứng dụng React hiện đại. Tuy nhiên, Recharts được chọn trong đồ án này vì tính tiện dụng,

khả năng tích hợp tốt với hệ sinh thái React và hiệu suất cao khi hiển thị dữ liệu y sinh phức tạp.

Với những ưu điểm như khả năng tùy chỉnh cao, hiệu suất tốt và cộng đồng hỗ trợ rộng rãi, Recharts đóng vai trò quan trọng trong hệ thống, giúp các nhà nghiên cứu và bác sĩ có một công cụ trực quan hóa dữ liệu mạnh mẽ để hỗ trợ phân tích chuyên sâu về ung thư phổi.

3.6 Supabase Storage

Supabase là một nền tảng mã nguồn mở cung cấp các dịch vụ backend hiện đại, trong đó Supabase Storage đóng vai trò quan trọng trong việc lưu trữ và phục vụ các tệp tin như hình ảnh, video, tài liệu và các loại dữ liệu số khác. Trong khi MongoDB đảm nhiệm vai trò lưu trữ dữ liệu có cấu trúc và xử lý các thông tin y sinh như mẫu bệnh phẩm và các chỉ số phân tích, Supabase đảm bảo khâu lưu trữ các file dữ liệu như hình ảnh y khoa, báo cáo xét nghiệm, dữ liệu y sinh chuyên sâu dạng bảng một cách riêng biệt và hiệu quả. [5]

Trong quá trình tải lên các hình ảnh chẩn đoán hay tài liệu xét nghiệm, Supabase Storage hỗ trợ cơ chế truyền tải an toàn, có thể tự động khôi phục khi gián đoạn kết nối mạng, nhờ đó đảm bảo rằng việc cập nhật tài liệu y tế diễn ra ổn định ngay cả trong điều kiện hạ tầng mạng không ổn định. Đồng thời, Supabase cũng cung cấp giao diện quản lý trực quan, cho phép quản trị viên hệ thống dễ dàng kiểm soát, phân quyền truy cập theo từng nhóm người dùng như bác sĩ điều trị hay chuyên gia nghiên cứu. Việc kiểm soát truy cập nghiêm ngặt này khá quan trọng để bảo vệ an toàn thông tin bệnh nhân và tuân thủ các quy định bảo mật y tế.

Một số giải pháp thay thế có thể được cân nhắc cho khâu lưu trữ file gồm Firebase Storage, AWS S3 hoặc Cloudinary. Firebase Storage mặc dù tích hợp chặt chẽ với hệ sinh thái Google và hỗ trợ các ứng dụng thời gian thực, nhưng thường gặp hạn chế về khả năng tùy biến cơ chế bảo mật cũng như chi phí vận hành có thể tăng cao theo quy mô dữ liệu. Trong khi đó, AWS S3 được đánh giá cao về tính ổn định và khả năng mở rộng nhưng đòi hỏi phải cấu hình phức tạp và chi phí quản lý thường khó kiểm soát đối với các dự án không có quy mô quá lớn, điều này có thể gây khó khăn cho việc tích hợp liền mạch với hệ thống backend hiện có. Cloudinary, mặc dù chuyên môn hóa trong việc xử lý và tối ưu hóa hình ảnh, lại hạn chế về khả năng quản lý và lưu trữ tổng thể các file văn bản, khiến cho việc đồng bộ hóa với dữ liệu y sinh trở nên không liên mạch. Chính những hạn chế này đã làm nổi bật lợi thế của Supabase, khi cung cấp một giải pháp toàn diện vừa dễ tích hợp, vừa đáp ứng được yêu cầu lưu trữ file dữ liệu hiệu quả.

Với khả năng lưu trữ file mạnh mẽ, kiểm soát truy cập linh hoạt và tích hợp sâu

với các dịch vụ backend khác, Supabase Storage là một giải pháp lý tưởng cho các ứng dụng cần quản lý và phục vụ nội dung số, đặc biệt trong các lĩnh vực như y tế, nơi việc lưu trữ và bảo mật dữ liệu là vô cùng quan trọng. Bên cạnh đó, chi phí hợp lý, dễ vận hành, dễ bảo trì và có tài liệu hướng dẫn đầy đủ giúp nền tảng này trở thành lựa chọn phù hợp nhất cho hệ thống lưu trữ tài liệu y sinh có quy mô vừa và lớn trong nghiên cứu và phân tích chuyên sâu về ung thư phổi.

3.7 React Hook Form

React Hook Form là thư viện quản lý form trên React, thiết kế hiệu năng cao và dễ sử dụng, do vậy được sử dụng để quản lý các form nhập liệu khi thêm hoặc chỉnh sửa mẫu bệnh phẩm, bài báo khoa học, thông tin bệnh nhân và kết quả xét nghiệm. Các biểu mẫu trong hệ thống thường chứa nhiều trường dữ liệu, yêu cầu tính chính xác và nhất quán cao, do đó việc sử dụng một thư viện hỗ trợ tốt về hiệu năng và kiểm tra dữ liệu là cần thiết. [6]

Thư viện này cung cấp hook useForm, hỗ trợ các chức năng giúp đơn giản hóa việc khởi tạo, kiểm soát trạng thái và xử lý logic khi submit. Với những form phức tạp trong lĩnh vực y sinh, cần kiểm tra đầu vào các trường thông tin hoặc xác thực dữ liệu y tế — React Hook Form giúp dễ dàng áp dụng các quy tắc xác thực dữ liệu. Việc hiển thị lỗi cũng được tối ưu, đảm bảo người dùng nhanh chóng nhận biết và sửa lỗi nhập liệu ngay trong quá trình thao tác. Một điểm mạnh khác là khả năng tích hợp với các thư viện xác thực như Yup hoặc Zod. Các quy tắc kiểm tra phức tạp có thể được định nghĩa rõ ràng, tách biệt khỏi giao diện người dùng, giúp mã nguồn dễ đọc, dễ bảo trì và giảm thiểu sai sót logic. Trong các hệ thống y sinh, nơi dữ liệu cần có độ chính xác cao và đôi khi phải đáp ứng những quy tắc nghiệp vụ khắt khe, việc tách biệt rõ ràng giữa giao diện nhập liệu và logic xác thực đóng vai trò then chốt trong việc duy trì chất lượng phần mềm. Về mặt giao diện, React Hook Form hỗ trợ tốt việc tích hợp với các thư viện giao diện người dùng phổ biến như Material UI hay Shadcn/ui. Điều này giúp hệ thống dễ dàng xây dựng các giao diện đẹp mắt, chuẩn hoá và nhất quán, đồng thời đảm bảo khả năng tiếp cận cho các bác sĩ, chuyên viên xét nghiệm hay nhà nghiên cứu khi thao tác nhập liệu trên hệ thống.

Một số thư viện thay thế khác cũng được cân nhắc gồm Formik, React Final và Redux Form. Formik là thư viện quản lý form khá phổ biến trước đây nhờ API rõ ràng, dễ dùng, hỗ trợ tốt với các controlled component. Tuy nhiên, trong các form lớn, phức tạp như nhập liệu y sinh, Formik thường gây ra nhiều lần render không cần thiết, ảnh hưởng đến hiệu năng và tốc độ phản hồi của giao diện. React Final Form cũng là một lựa chọn nhẹ hơn Formik, nhưng việc sử dụng controlled

components khiến việc tối ưu hiệu năng trở nên khó khăn khi làm việc với các form động có nhiều trường thông tin. Redux Form là giải pháp đồng bộ trạng thái form lên Redux Store, phù hợp với các ứng dụng yêu cầu chia sẻ trạng thái form giữa nhiều component phức tạp, tuy nhiên trong bối cảnh hệ thống quản lý dữ liệu y sinh, việc đưa toàn bộ form state lên global store là không cần thiết, làm tăng thêm độ phức tạp và chi phí bảo trì. Xét về hiệu năng, React Hook Form cho phép tối ưu re-render nhờ cơ chế chỉ cập nhật những trường có sự thay đổi. Điều này đặc biệt hữu ích khi làm việc với các form lớn, có nhiều trường động như danh sách chỉ số xét nghiệm hoặc các tác giả trong bài báo khoa học. So với các thư viện quản lý form sử dụng controlled component truyền thống, thư viện này duy trì được trải nghiệm người dùng mượt mà, ổn định ngay cả khi số lượng trường dữ liệu tăng lên đáng kể.

Nhờ vào sự cân bằng giữa hiệu năng, khả năng mở rộng, dễ tích hợp, dễ bảo trì và tính tương thích cao với hệ sinh thái React cùng với khối lượng dữ liệu lớn, nhiều trường thông tin, nhiều logic kiểm tra phức tạp và yêu cầu hiệu suất vận hành ổn định lâu dài, React Hook Form được lựa chọn để làm form nhập liệu đáp ứng đầy đủ các tiêu chí cho hệ thống này.

3.8 React-window

React-window là một thư viện mã nguồn mở được thiết kế để tối ưu hóa hiển thị các danh sách hoặc bảng dữ liệu lớn trong ứng dụng sử dụng React. Trong hệ thống hiện tại, thư viện trên giải quyết vấn đề hiệu suất bằng cách giảm số lượng phần tử DOM, chỉ render dữ liệu trong tầm nhìn khi hiển thị các file dữ liệu multi-omics (tiêu biểu gồm dữ liệu biểu hiện gen, mức độ methyl hóa DNA, miRNA, số lượng copy bản sao) với hàng chục nghìn hoặc hàng triệu dòng, đảm bảo ứng dụng chạy mượt mà và tương tác hiệu quả.

React-window cung cấp các thành phần như FixedSizeList, VariableSizeList, FixedSizeGrid, và VariableSizeGrid để hiển thị danh sách hoặc bảng. Trong hệ thống, nó được sử dụng để hiển thị bảng dữ liệu khi người dùng xem chi tiết file, hỗ trợ cả dữ liệu cố định và thay đổi kích thước, phù hợp với các dòng có nội dung phức tạp. Kỹ thuật windowing đảm bảo hiệu suất cao, cuộn mượt mà, và tích hợp dễ dàng với giao diện, giúp người dùng xem trước nội dung file và tìm kiếm dữ liệu mà không gặp trễ ngại.

Các thư viện thay thế như react-virtualized, react-table, và react-virtuoso có thể được cân nhắc. React-virtualized hỗ trợ nhiều tính năng nhưng có bundle lớn và phức tạp. React-table nhẹ nhưng không tự hỗ trợ virtualization. React-virtuoso hiện đại nhưng khó thiết lập. React-window được chọn nhờ kích thước nhỏ, API

đơn giản, hỗ trợ kích thước thay đổi, và cân bằng giữa hiệu suất và tính năng, phù hợp với dữ liệu dung lượng lớn trong hệ thống.

Tóm lại, việc áp dụng React-window mang lại hiệu suất ổn định, cho phép hiển thị hàng triệu dòng dữ liệu mà không bị chậm, nâng cao trải nghiệm người dùng cho nhóm nghiên cứu và bác sĩ. Nó hỗ trợ khả năng mở rộng, xử lý dữ liệu ngày càng lớn, và tối ưu hóa tài nguyên bằng cách giảm tải bộ nhớ và CPU, đảm bảo ứng dụng chạy hiệu quả trên nhiều thiết bị.

Chương 3 đã làm rõ cơ sở lý thuyết và thực tiễn cho việc chọn lựa các công nghệ. Mỗi giải pháp đều gắn liền với yêu cầu đã đề ra: từ quản lý API, lưu trữ, truy vấn dữ liệu đến trực quan hóa kết quả. Trên cơ sở những phân tích này, em đã quyết định sử dụng bộ công nghệ này để đảm bảo yêu cầu về hiệu năng, bảo mật và mở rộng, làm tiền đề cho bước thiết kế và triển khai hệ thống trong Chương 4.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

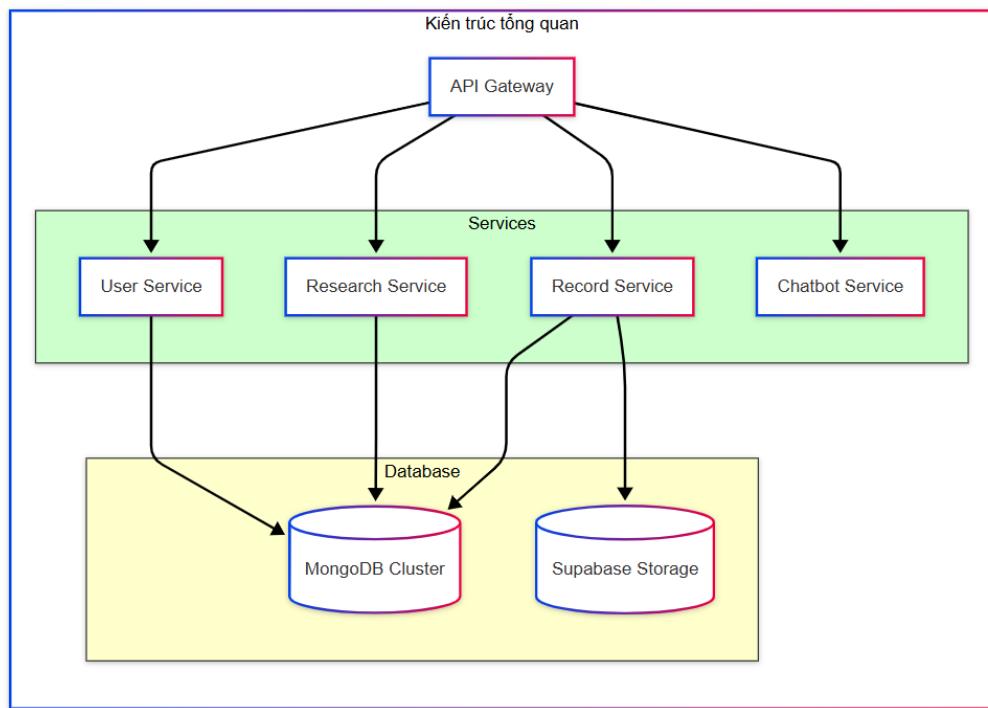
Để đáp ứng yêu cầu về khả năng mở rộng, hiệu năng xử lý cao và dễ dàng bảo trì trong bối cảnh phân tích dữ liệu ung thư phổi – một lĩnh vực đòi hỏi xử lý lượng lớn dữ liệu phức tạp từ nhiều nguồn khác nhau – kiến trúc microservices được lựa chọn làm nền tảng cho hệ thống. Kiến trúc này chia ứng dụng thành các dịch vụ nhỏ, độc lập, mỗi dịch vụ đảm nhiệm một chức năng cụ thể của hệ thống. Nhờ vậy, các dịch vụ này có thể được phát triển, triển khai và duy trì một cách riêng biệt mà không ảnh hưởng đến toàn bộ hệ thống, tạo điều kiện thuận lợi cho việc nâng cấp và tích hợp các chức năng mới theo thời gian.

Trong đề tài này, việc áp dụng kiến trúc microservices giúp tối ưu hóa quá trình xử lý và truy xuất dữ liệu y sinh, từ đó hỗ trợ các hoạt động phân tích chuyên sâu về ung thư phổi. Hơn nữa, để kết nối mượt mà các microservice này với nhau, hệ thống triển khai một API Gateway. Vai trò của API Gateway là điều phối các yêu cầu từ phía client, phân phối chúng đến các dịch vụ tương ứng và sau đó tổng hợp kết quả trả về, tạo nên một giao diện nhất quán và đơn giản cho người dùng. Cơ chế này không chỉ giúp giảm thiểu mức độ phức tạp khi làm việc với các dịch vụ độc lập mà còn tăng cường bảo mật và hiệu quả trong việc quản lý lưu lượng truy cập.

Ngoài ra, việc triển khai các microservice theo mô hình container (sử dụng Docker) giúp hệ thống dễ dàng mở rộng độc lập từng thành phần, xây dựng quy trình CI/CD chặt chẽ, và giảm thiểu rủi ro trong quá trình nâng cấp hoặc sửa lỗi. Điều này cho phép ứng dụng có thể nhanh chóng thích nghi với sự gia tăng của khối lượng dữ liệu ung thư phổi cũng như các yêu cầu phân tích ngày càng phức tạp trong tương lai.

Nhờ áp dụng kiến trúc Microservices kết hợp với hệ cơ sở dữ liệu NoSQL MongoDB và nền tảng lưu trữ file Supabase, hệ thống đảm bảo khả năng xử lý dữ liệu nhanh chóng, dễ dàng mở rộng và tối ưu hóa cho các nhu cầu nghiên cứu y sinh chuyên sâu. Việc tổ chức các collection theo từng nhóm chức năng riêng biệt cũng giúp nâng cao hiệu quả trong truy xuất và phân tích dữ liệu, từ đó hỗ trợ việc nghiên cứu và điều trị ung thư phổi một cách chính xác và khoa học.

4.1.2 Thiết kế tổng quan

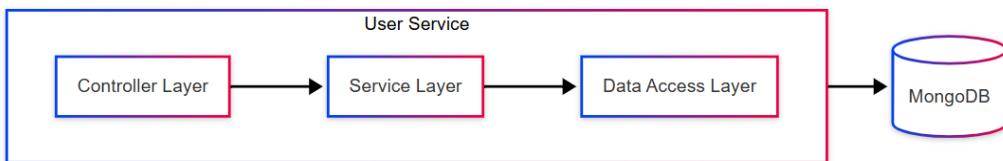


Hình 4.1: Kiến trúc tổng quan

Trong ảnh trên, hệ thống sử dụng cơ sở dữ liệu NoSQL MongoDB để quản lý dữ liệu thông qua ba collection chính: records, researches và users. Ngoài ra, nền tảng Supabase Storage hỗ trợ lưu trữ dữ liệu dạng file. Mỗi collection này được quản lý bởi một microservice tương ứng. Cụ thể, Record Service chịu trách nhiệm quản lý thông tin bệnh nhân, kết quả xét nghiệm, hình ảnh y khoa, dữ liệu multi-omics và các chỉ số lâm sàng. Dịch vụ này đảm bảo dữ liệu được thu thập, lưu trữ và truy xuất một cách nhanh chóng và chính xác, phục vụ cho quá trình phân tích xu hướng và phát hiện các mối liên hệ giữa các chỉ số y sinh. Research Service quản lý các dữ liệu nghiên cứu về ung thư phổi, bao gồm các bài báo khoa học thường thức và khoa học chuyên sâu. Dịch vụ này cung cấp các phương thức để tạo mới, cập nhật, xóa và truy vấn bài báo, hỗ trợ tìm kiếm theo tiêu đề, tác giả, cho phép phân loại theo chuyên ngành và liên kết đến các bài báo trên Internet nhằm mở rộng nguồn tham khảo. User Service đảm nhiệm quản lý thông tin người dùng của hệ thống từ người dùng cơ bản như bác sĩ, nhà nghiên cứu đến quản trị viên cùng các chức năng phân quyền và bảo mật thông tin. Đây là thành phần quan trọng giúp đảm bảo an toàn dữ liệu nhạy cảm và tạo điều kiện cho việc kiểm soát truy cập một cách chặt chẽ. Hệ thống còn kết nối tới Chatbot Service độc lập, chịu trách nhiệm tương tác với người dùng thông qua giao diện hội thoại.

4.1.3 Thiết kế chi tiết

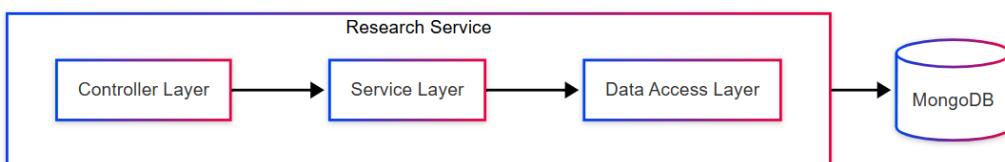
Kiến trúc thành phần User Service:



Hình 4.2: Kiến trúc thành phần User Service

Trong kiến trúc của User Service, tầng điều khiển đóng vai trò là lớp giao tiếp đầu tiên với các request từ phía client. Mỗi controller tương ứng với một endpoint cụ thể, chịu trách nhiệm tiếp nhận và phân tích yêu cầu đầu vào, sau đó chuyển dữ liệu cần thiết đến tầng nghiệp vụ để xử lý và đảm nhiệm toàn bộ logic nghiệp vụ cốt lõi của thông tin người dùng. Cuối cùng, tầng truy cập dữ liệu đảm nhiệm kết nối và tương tác trực tiếp với cơ sở dữ liệu MongoDB. Tầng này có thể chứa thành phần UserRepository để thực hiện các thao tác như tạo mới, cập nhật và truy vấn người dùng trong collection tương ứng, đảm bảo dữ liệu luôn được đồng bộ và bảo mật một cách hiệu quả.

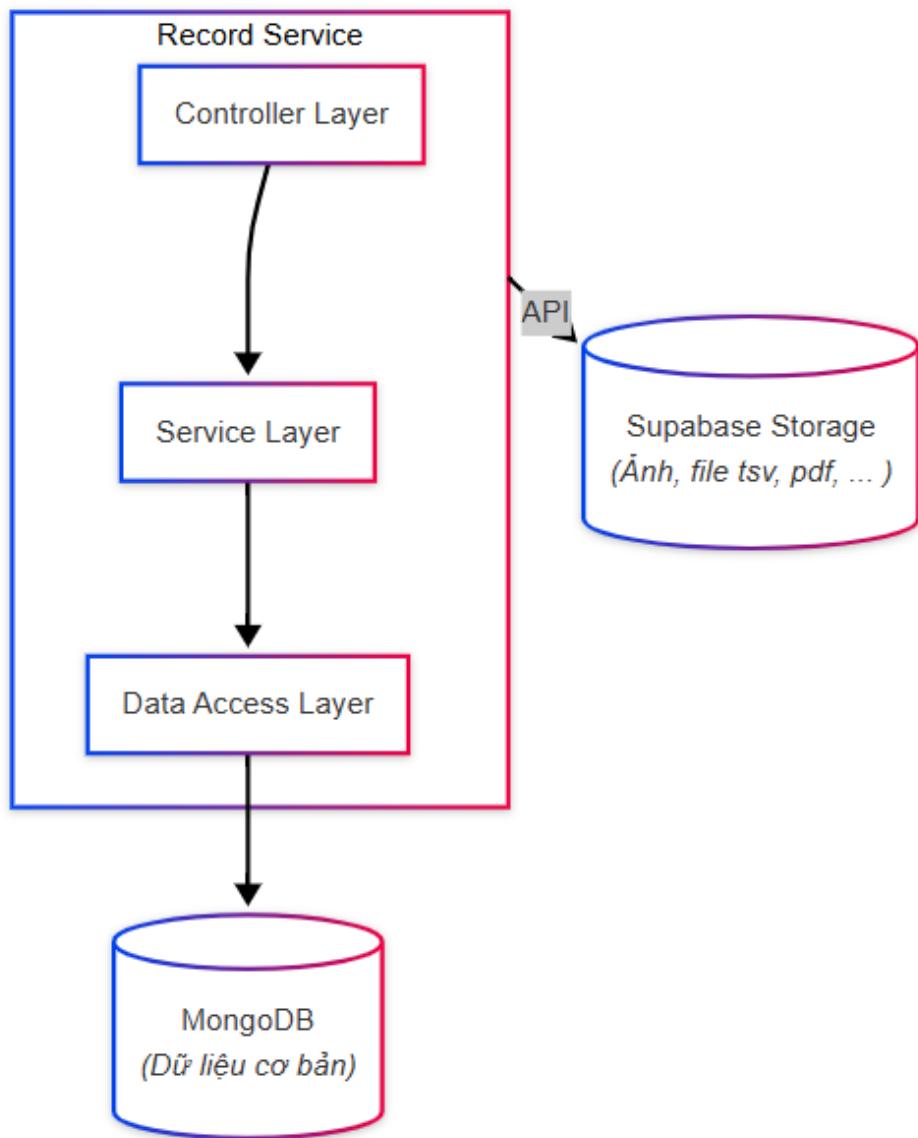
Kiến trúc thành phần Research Service:



Hình 4.3: Kiến trúc thành phần Research Service

Trong kiến trúc của Research Service, tầng điều khiển đóng vai trò là lớp giao tiếp đầu tiên với các request từ phía client. Mỗi controller tương ứng với một endpoint cụ thể, chịu trách nhiệm tiếp nhận và phân tích yêu cầu đầu vào, sau đó chuyển dữ liệu cần thiết đến tầng nghiệp vụ để xử lý và đảm nhiệm toàn bộ logic nghiệp vụ cốt lõi liên quan đến thông tin các bài báo khoa học. Cuối cùng, tầng truy cập dữ liệu đảm nhiệm kết nối và tương tác trực tiếp với cơ sở dữ liệu MongoDB. Tầng này có thể chứa thành phần ResearchRepository để thực hiện các thao tác như tạo mới, cập nhật, truy vấn và xóa các bài báo khoa học trong collection tương ứng, đảm bảo dữ liệu luôn được đồng bộ và quản lý một cách hiệu quả.

Kiến trúc thành phần Record Service:

**Hình 4.4:** Kiến trúc thành phần Record Service

Trong kiến trúc của Record Service, tầng điều khiển đóng vai trò là lớp giao tiếp đầu tiên với các yêu cầu từ phía client. Mỗi controller tương ứng với một endpoint cụ thể, chịu trách nhiệm tiếp nhận và phân tích yêu cầu đầu vào, sau đó chuyển dữ liệu cần thiết đến tầng nghiệp vụ để xử lý và đảm nhiệm toàn bộ logic nghiệp vụ cốt lõi liên quan đến việc quản lý bản ghi, xử lý các thao tác cần thiết dựa trên dữ liệu nhận được. Cuối cùng, tầng truy cập dữ liệu đảm nhiệm kết nối và tương tác trực tiếp với cơ sở dữ liệu MongoDB, nơi lưu trữ dữ liệu dạng bảng, với các thành phần như repository có thể thực hiện các thao tác như tạo mới, cập nhật, truy vấn và xóa bản ghi trong collection tương ứng, đảm bảo dữ liệu được đồng bộ và quản lý hiệu quả. Ngoài ra, tầng điều khiển cũng có luồng kết nối gián tiếp qua API đến Supabase Storage, nơi lưu trữ các file như ảnh, văn bản và các định dạng khác, với việc quản lý file được thực hiện thông qua API từ phía frontend, không trực tiếp qua backend.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

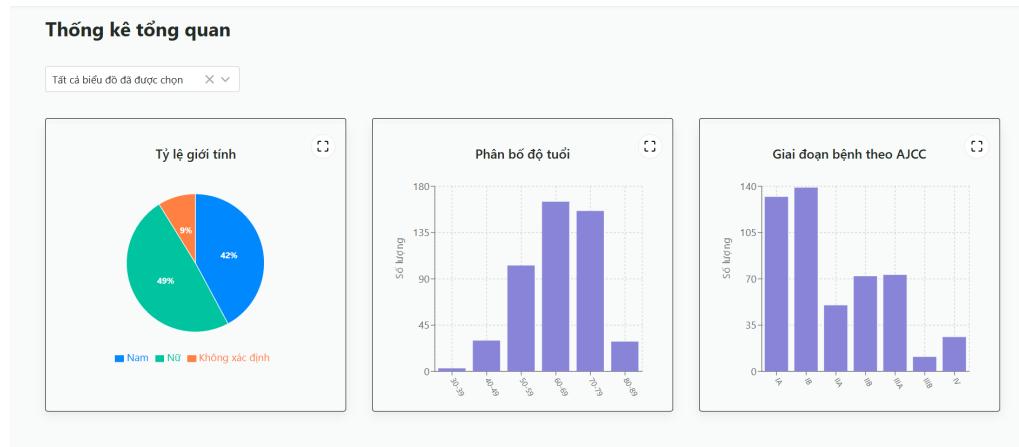
Trong đề tài này, giao diện được thiết kế theo tiêu chuẩn hiện đại, đảm bảo khả năng tương thích trên nhiều loại thiết bị với độ phân giải linh hoạt từ màn hình máy tính để bàn (ví dụ: 1920×1080, 2560×1440) đến các laptop và thiết bị di động với độ phân giải tối thiểu khoảng 1024×768. Cấu trúc trang đảm bảo khoảng cách hợp lý giữa các thành phần và tạo không gian thoáng đãng giúp người dùng dễ dàng tập trung vào thông tin quan trọng.

Việc sử dụng bộ công cụ Shadcn/ui kết hợp với Tailwind CSS cho phép xây dựng các component theo phong cách “headless”, tức chỉ tập trung vào chức năng và khả năng tùy biến. Điều này giúp các button, form và các thành phần hiển thị thông báo phản hồi được thiết kế với kích thước, khoảng cách và vị trí ổn định, đồng thời cho phép tùy chỉnh theo nhu cầu cụ thể của hệ thống. Ví dụ, nút bấm được chuẩn hóa với kích thước đủ lớn để thao tác trên các màn hình cảm ứng hoặc chuột, vị trí các thông báo sau thao tác được đặt sát các form nhập liệu để đảm bảo người dùng luôn nhận được phản hồi kịp thời.

Hơn nữa, những tiêu chuẩn phối màu và giao diện như khai báo các màu sắc chính, phụ, màu hover, màu thông báo đã được thống nhất toàn bộ trong dự án, đảm bảo tính nhất quán và chuyên nghiệp cho giao diện. Việc này không chỉ giúp người dùng cảm thấy quen thuộc khi sử dụng các chức năng của hệ thống mà còn làm nổi bật các thông tin phân tích quan trọng liên quan đến dữ liệu ung thư phổi.

Cuối cùng, để minh họa cho các chức năng quan trọng, em cung cấp một số hình ảnh bản vẽ thiết kế giao diện của các màn hình như: trang dashboard chính hiển thị biểu đồ và bảng số liệu, màn hình quản lý thông tin bệnh nhân và giao diện nhập liệu, cũng như các thành phần thông báo và phản hồi khi thao tác. Các hình ảnh này chỉ mang tính chất thiết kế ban đầu (prototype) và chưa phản ánh giao diện sản phẩm cuối cùng, nhằm cung cấp một cái nhìn tổng quan và làm cơ sở cho các bước triển khai sau này.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.5: Thiết kế giao diện "Xem thống kê"



Hình 4.6: Thiết kế giao diện "Xem danh sách bài báo"

This is a form for adding a new research article. It includes fields for 'Mã nghiên cứu' (Research ID), 'Loại' (Type), 'Tiêu đề' (Title), 'Ngày đăng' (Publication Date), and 'Nhóm tác giả' (Author Group). Each field has a placeholder text and a green border indicating it is a required field.

Hình 4.7: Thiết kế giao diện "Thêm mới bài báo"

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Bảng dữ liệu bệnh phẩm											<input type="button" value="Nhập mã mẫu bệnh phẩm"/>	<input type="button" value="Thêm bản ghi"/>
Kết quả tìm kiếm (567)											<input type="button" value="Xuất Excel"/>	
Hành động	STT	Mã bệnh nhân	Loại ung thư	Mã mẫu bệnh phẩm	Tuổi chẩn đoán	Giới tính	Tình trạng	Giai đoạn AJCC	Vị trí sinh thiết	Chẩn đoán chính	Năm ch.	
<input type="checkbox"/>	1	TCGA-05-4244	Lung Adenocarci...	TCGA-05-4244-01	70	Male	Alive	Stage IV	Lower lobe, lung	Adenocarcinoma...	2009	
<input type="checkbox"/>	2	TCGA-05-4384	Lung Adenocarci...	TCGA-05-4384-01	66	Male	Alive	Stage IIIA	Lower lobe, lung	Adenocarcinoma...	2009	
<input type="checkbox"/>	3	TCGA-05-4389	Lung Adenocarci...	TCGA-05-4389-01	70	Male	Alive	Stage IA	Upper lobe, lung	Adenocarcinoma...	2005	
<input type="checkbox"/>	4	TCGA-05-4402	Lung Adenocarci...	TCGA-05-4402-01	56	Female	Dead	Stage IV	Lower lobe, lung	Adenocarcinoma...	2007	
<input type="checkbox"/>	5	TCGA-05-4417	Lung Adenocarci...	TCGA-05-4417-01	51	Female	Alive	Stage IB	Upper lobe, lung	Adenocarcinoma...	2008	

Hình 4.8: Thiết kế giao diện "Xem danh sách mẫu bệnh phẩm"

Chỉnh sửa bản ghi y tế

Mã bệnh nhân	Mô tả di căn xa (M) theo AJCC
TCGA-05-4244	Stage M1
Mã mẫu bệnh phẩm	Mô tả hạch (N) theo AJCC
TCGA-05-4244-01	Stage N2
Tuổi lúc chẩn đoán	Giai đoạn bệnh lý theo AJCC
70	Stage IV
Vị trí sinh thiết	Mô tả khối u (T) theo AJCC
Lower lobe, lung	Stage T2
Loại ung thư	Chẩn đoán chính
Lung Adenocarcinoma	Adenocarcinoma, NOS
Số tháng không bệnh (sau điều trị)	Vị trí khối u chính
NA	Bronchus and lung
Tình trạng không bệnh	Có tiền sử ung thư ác tính trước đó

Hình 4.9: Thiết kế giao diện "Sửa dữ liệu mẫu bệnh phẩm"

Tổng quan **Báo cáo bệnh lý** **Ánh khối u** **Dữ liệu đa omics**

Thông tin chuyên ngành

Quản lý file kết quả xét nghiệm

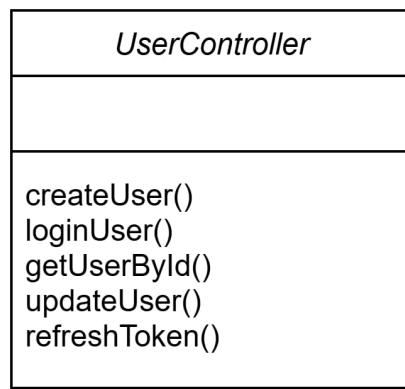
Kéo thả hoặc nhập để chọn tệp (tối đa 50MB)

[TCGA-55-6987.345eb1ff-9520-4463-8a84-07aeb26ce559.pdf](#)

Hình 4.10: Thiết kế giao diện "Quản lý file dữ liệu"

4.2.2 Thiết kế lớp

Lớp UserController:



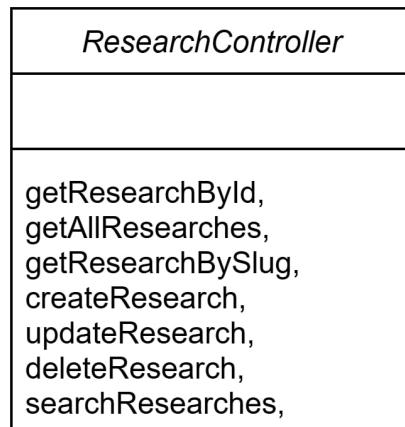
Hình 4.11: Thiết kế lớp UserController

Các phương thức của lớp UserController được thiết kế chi tiết và mô tả trong bảng 4.1

Bảng 4.1: Mô tả các phương thức lớp UserController

STT	Tên phương thức	Chi tiết
1	createUser()	Tạo tài khoản mới
2	loginUser()	Đăng nhập vào hệ thống
3	getUserById()	Lấy thông tin người dùng với ID tương ứng
4	updateUser()	Cập nhật thông tin người dùng với ID tương ứng
5	refreshToken()	Tạo token mới khi token cũ hết hạn

Lớp ResearchController:



Hình 4.12: Thiết kế lớp ResearchController

Các phương thức của lớp ResearchController được thiết kế chi tiết và mô tả trong

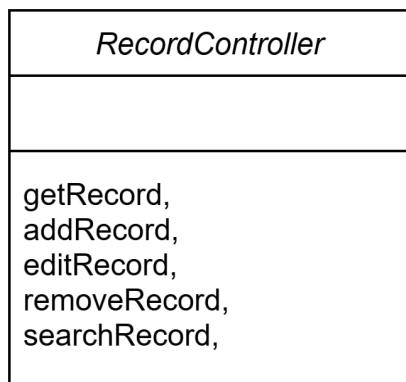
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

bảng 4.2

Bảng 4.2: Mô tả các phương thức lớp ResearchController

STT	Tên phương thức	Chi tiết
1	getResearchById()	Lấy thông tin bài báo dựa trên mã định danh duy nhất (ID)
2	getAllResearches()	Truy xuất toàn bộ danh sách các bài báo hiện có trong hệ thống.
3	getResearch-BySlug()	Tìm kiếm bài báo thông qua định danh dạng chuỗi
4	createResearch()	Tạo mới một bài báo trong cơ sở dữ liệu.
5	updateResearch()	Chỉnh sửa và cập nhật nội dung của một bài báo đã tồn tại.
6	deleteResearch()	Xoá một bài báo khỏi hệ thống dựa trên ID.
7	searchResearches()	Tìm kiếm bài báo theo từ khoá hoặc điều kiện lọc cụ thể.

Lớp RecordController:



Hình 4.13: Thiết kế lớp RecordController

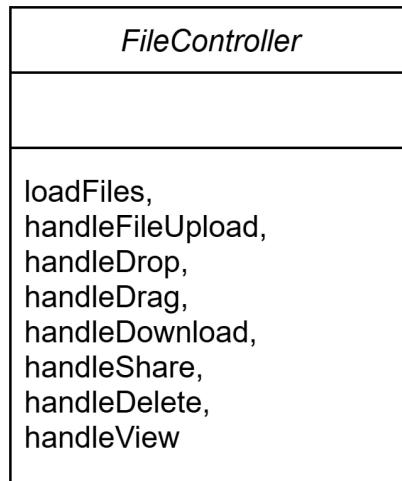
Các phương thức của lớp RecordController được thiết kế chi tiết và mô tả trong bảng 4.3

Bảng 4.3: Mô tả các phương thức lớp RecordController

STT	Tên phương thức	Chi tiết
1	getRecord()	Lấy thông tin bản ghi
2	addRecord()	Thêm bản ghi mới vào hệ thống

3	editRecord()	Chỉnh sửa thông tin bản ghi hiện có
4	removeRecord()	Xoá bản ghi khỏi hệ thống
5	searchRecord()	Tìm kiếm bản ghi theo mã bệnh phẩm

Lớp FileController:



Hình 4.14: Thiết kế lớp FileController

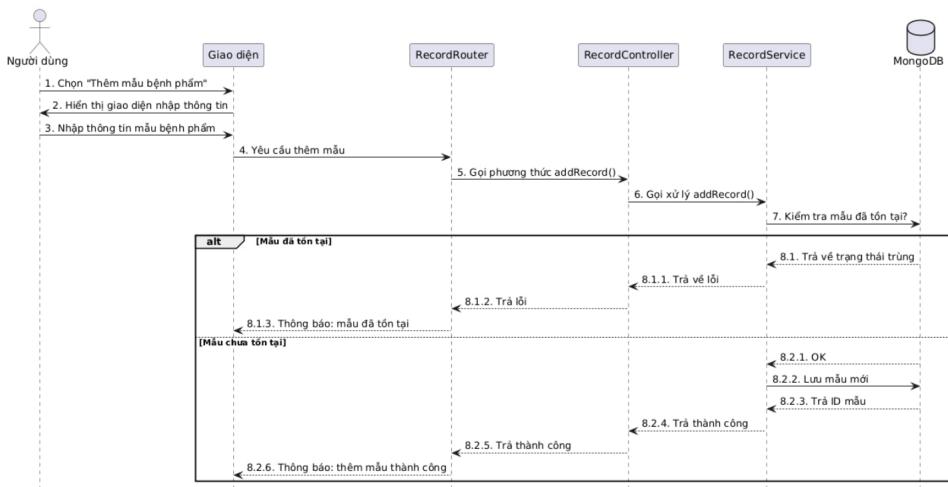
Các phương thức của lớp UserController được thiết kế chi tiết và mô tả trong bảng 4.4

Bảng 4.4: Mô tả các phương thức lớp FileController

STT	Tên phương thức	Chi tiết
1	loadFiles()	Tải danh sách các tệp từ nền tảng Supabase Storage
2	handleFileUpload()	Xử lý việc tải lên tệp từ phía người dùng.
3	handleDrop()	Xử lý hành động thả tệp vào giao diện.
4	handleDrag()	Xử lý sự kiện kéo tệp trong giao diện.
5	handleDownload()	Xử lý việc tải tệp xuống từ hệ thống.
6	handleShare()	Xử lý việc chia sẻ tệp với người dùng khác hoặc hệ thống ngoài.
7	handleDelete()	Xóa tệp khỏi hệ thống.
8	handleView()	Hiển thị nội dung tệp hoặc thông tin liên quan đến tệp.

Để trình bày rõ hơn miêu tả chi tiết phương thức của một lớp, ảnh 4.15 dưới đây minh họa luồng hoạt động của usecase thêm mới dữ liệu bệnh phẩm:

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.15: Biểu đồ trình tự usecase "Thêm mới dữ liệu bệnh phẩm"

4.2.3 Thiết kế cơ sở dữ liệu

Collection "users":

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	email	string	Email
2	password	string	Mật khẩu
3	name	string	Tên người dùng
4	role	string	Vai trò

Collection "records":

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	patient_id	string	Mã bệnh nhân
2	sample_id	string	Mã mẫu bệnh phẩm
3	diagnosis_age	int	Tuổi lúc chẩn đoán
4	biopsy_site	string	Vị trí sinh thiết
5	cancer_type	string	Loại ung thư
6	disease_free_months	double	Số tháng không bệnh (sau điều trị)
7	disease_free_status	string	Tình trạng không bệnh
8	disease_type	string	Loại bệnh
9	ethnicity_category	string	Dân tộc
10	fraction_genome_altered	double	Tỷ lệ bộ gen bị biến đổi
11	icd_10_classification	string	Phân loại theo mã ICD-10

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

STT	Tên trường	Kiểu dữ liệu	Mô tả
12	is_ffpe	string	Có phải mẫu FFPE không?
13	morphology	string	Mô học
14	mutation_count	int	Số lượng đột biến
15	overall_survival_-months	double	Số tháng sống sót tổng thể
16	ajcc_pathologic_m	string	Mô tả di căn xa (M)
17	ajcc_pathologic_n	string	Mô tả hạch (N)
18	ajcc_pathologic_-stage	string	Giai đoạn bệnh lý
19	ajcc_pathologic_t	string	Mô tả khối u (T)
20	primary_diagnosis	string	Chẩn đoán chính
21	primary_tumor_site	string	Vị trí khối u chính
22	prior_malignancy	string	Có tiền sử ung thư trước đó không?
23	prior_treatment	string	Đã từng điều trị trước đó không?
24	sample_type	string	Loại mẫu
25	sex	string	Giới tính
26	years_smoked	int	Số năm hút thuốc
27	cigarette_smoking_-history_pack_year	double	Số gói thuốc hút trong 1 năm
28	vital_status	string	Tình trạng sống
29	year_of_death	int	Năm mất
30	year_of_diagnosis	int	Năm được chẩn đoán

Collection "researches":

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	research_id	string	Mã bài báo
2	type	string	Loại bài báo
3	title	string	Tiêu đề
4	date	date	Ngày đăng
5	author	string	Nhóm tác giả
6	link	string	Đường dẫn
7	description	string	Mô tả

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

STT	Tên trường	Kiểu dữ liệu	Mô tả
8	image	string	Ảnh minh họa
9	detail	string	Nội dung chi tiết

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.8: Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Phiên bản	Địa chỉ URL
IDE lập trình	Visual Studio Code	1.100	https://code.visualstudio.com/
Công cụ thử nghiệm API	Postman	11.48	https://www.postman.com/
Môi trường chạy	Node.js	20.18.0	https://nodejs.org/
Trình quản lý gói	npm	11.2.0	https://www.npmjs.com/
Ngôn ngữ lập trình chính	TypeScript	5.7.2	https://www.typescriptlang.org/
Xây dựng giao diện	React	18.2.0	https://react.dev/
Trình đóng gói frontend	Vite	5.0.8	https://vite.dev/
Framework CSS	Tailwind CSS	4.1.8	https://tailwindcss.com/
UI components	Shadcn/ui	0.9.5	https://ui.shadcn.com/
Thư viện biểu tượng	Lucide	0.311.0	https://lucide.dev/
Quản lý bảng dữ liệu	react-table	7.8.0	https://www.npmjs.com/package/react-table
Lưu trữ file	supabase/storage-js	2.7.3	https://www.npmjs.com/package/@supabase/storage-js

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Mục đích	Công cụ	Phiên bản	Địa chỉ URL
HTTP client cho frontend	Axios	1.7.9	https://www.npmjs.com/package/axios
Mã hóa mật khẩu	bcrypt	6.0.0	https://www.npmjs.com/package/bcrypt
Xử lý CORS trong backend	cors	2.8.5	https://www.npmjs.com/package/cors
Quản lý biến môi trường	dotenv	16.4.7	https://www.npmjs.com/package/dotenv
Framework backend	Express	4.21.1	https://expressjs.com/
Cơ sở dữ liệu Mongodb	mongodb	6.12.0	https://www.npmjs.com/package/mongodb
ORM cho MongoDB	mongoose	8.13.1	https://www.npmjs.com/package/mongoose
Kiểm tra mã nguồn	ESLint	8.55.0	https://www.npmjs.com/package/eslint
Xác thực JWT	jsonwebtoken	9.0.2	https://www.npmjs.com/package/jsonwebtoken
Ảo hóa danh sách	react-window	1.8.11	https://www.npmjs.com/package/react-window
Popup thông báo	sonner	1.3.1	https://www.npmjs.com/package/sonner
Kiểm tra kiểu dữ liệu	zod	3.25.64	https://www.npmjs.com/package/zod
Quản lý form	react-hook-form	7.58.0	https://www.npmjs.com/package/react-hook-form

4.3.2 Kết quả đạt được

Hệ thống được đóng gói thành nhiều thành phần chính, mỗi thành phần đảm nhận một vai trò cụ thể và phối hợp với nhau để tạo thành một hệ thống hoàn

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

chỉnh. Thành phần Backend, nằm trong thư mục /backend, chịu trách nhiệm xử lý nghiệp vụ, cung cấp API, quản lý dữ liệu, xác thực người dùng, kết nối cơ sở dữ liệu và hỗ trợ các dịch vụ RESTful cho frontend. Thành phần Frontend, nằm tại /frontend, đảm nhiệm vai trò giao diện người dùng, cho phép hiển thị dữ liệu, nhập liệu và tương tác trực tiếp với người dùng cuối. Bên cạnh đó, thành phần AI với tên gọi ollama3.1 tại thư mục /ollama3.1 chứa mã nguồn trí tuệ nhân tạo, các mô hình học máy, cùng các script Python phục vụ cho việc phân tích dữ liệu và hỗ trợ chatbot y tế. Về các file cấu hình quan trọng, hệ thống sử dụng các Dockerfile riêng biệt cho Backend và Frontend để đóng gói từng thành phần thành các image độc lập, đảm bảo dễ dàng triển khai và mở rộng. Thư mục ollama3.1 tập trung các script Python, notebook, cũng như dữ liệu mẫu phục vụ cho các tác vụ AI. Toàn bộ các thành phần trên được đóng gói và điều phối thông qua file cấu hình docker-compose.yml, giúp triển khai nhanh chóng và đồng bộ hệ thống trên máy chủ VPS.

Thống kê tổng quát:

Bảng 4.9: Thống kê thông tin của ứng dụng

Thành phần	Tổng số gói	Tổng số file	Số dòng code	Dung lượng
backend	6	24	2000	160 KB
frontend	20	120	7000	1100 KB
ollama3.1	1	9	500	45 KB
docker-compose.yml	0	1	80	1 KB

Thống kê chi tiết thư mục backend:

Bảng 4.10: Thống kê thông tin thư mục backend

Tên	Loại	Dung lượng (bytes)	Vai trò/chức năng
.dockerignore	file	23	Docker ignore file
.env.example	file	202	Biến môi trường mẫu
Dockerfile	file	113	Đóng gói backend
package.json	file	1,055	Định nghĩa dependency
package-lock.json	file	131,273	Quản lý dependency

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Loại	Dung lượng (bytes)	Vai trò/chức năng
tsconfig.json	file	12,427	TypeScript config
README.md	file	185	Hướng dẫn sử dụng
src	thư mục	23,000	Source code chính

Thống kê chi tiết thư mục frontend:

Bảng 4.11: Thống kê thông tin thư mục frontend

Tên	Loại	Dung lượng (bytes)	Vai trò/chức năng
.dockerignore	file	23	Docker ignore file
.env.example	file	366	Biến môi trường mẫu
.eslintrc.cjs	file	1000	Format code
tailwind.config.js	file	3500	Cấu hình CSS
tsconfig	file	1500	Cấu hình Typescript
index.html	file	1000	File HTML gốc
Dockerfile	file	129	Đóng gói FE
package.json	file	2,566	Định nghĩa dependency
package-lock.json	file	308,250	Quản lý dependency
README.md	file	1,441	Hướng dẫn sử dụng
src	thư mục	828,000	Source code chính

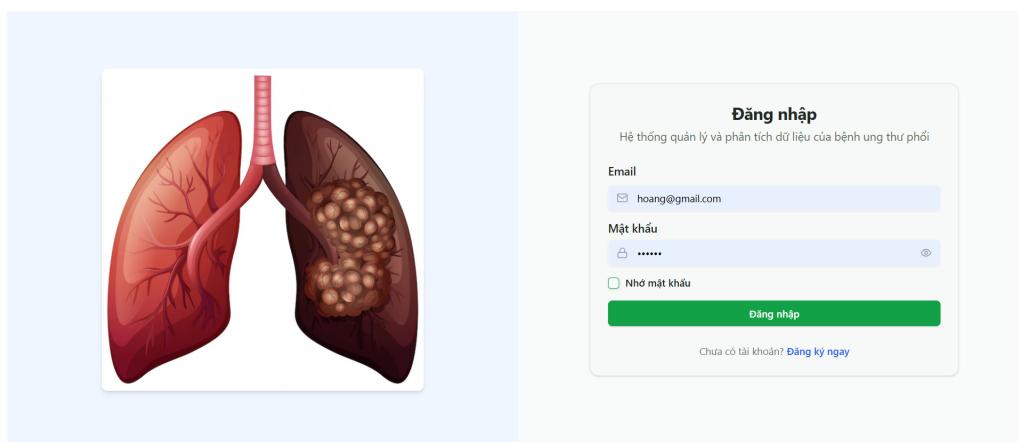
Thống kê chi tiết thư mục ollama3.1:

Bảng 4.12: Thống kê thông tin thư mục ollama3.1

Tên	Loại	Dung lượng (bytes)	Vai trò/chức năng
app.py	file	1,732	Ứng dụng chính Flask
chatbot.py	file	3,164	Chatbot AI
chatbot.ipynb	file	26,373	Notebook mô hình, thử nghiệm
lungcancer.csv	file	2,163	Dữ liệu mẫu AI
utils.py	file	150	Helper functions

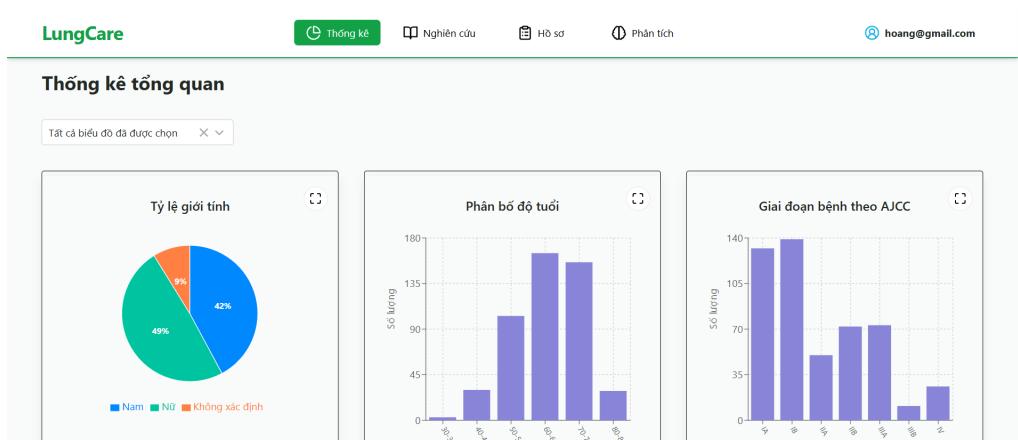
4.3.3 Minh họa các chức năng chính

Giao diện trang đăng nhập được chia thành hai phần rõ rệt. Bên trái là hình ảnh minh họa lá phổi người, trong đó phổi trái (hiển thị bên phải hình) có nhiều khối u màu nâu, cho thấy dấu hiệu của bệnh ung thư phổi, trong khi phổi phải (hiển thị bên trái hình) trông khỏe mạnh với màu sắc tươi sáng và các mạch máu rõ ràng. Bên phải là biểu mẫu đăng nhập dành cho người dùng, cụ thể là các bác sĩ, với các trường thông tin như email, mật khẩu, tùy chọn nhớ mật khẩu, cùng nút đăng nhập màu xanh nổi bật. Phía dưới còn có liên kết để đăng ký tài khoản mới. Trước khi truy cập vào dữ liệu bệnh nhân, người dùng bắt buộc phải đăng nhập nhằm đảm bảo tính bảo mật và phân quyền truy cập, đồng thời bảo vệ thông tin y tế cá nhân một cách an toàn và hiệu quả.



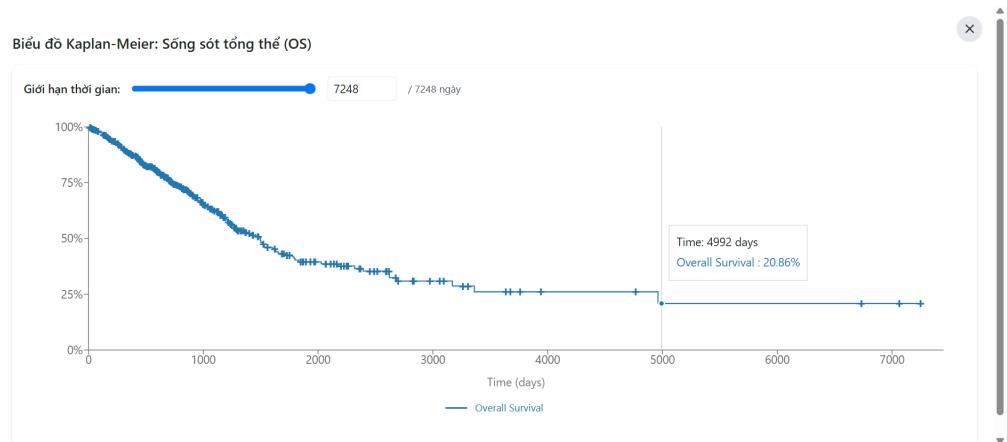
Hình 4.16: Giao diện trang đăng nhập

Giao diện trang thống kê bao gồm nhiều loại biểu đồ (biểu đồ cột, tròn, Kaplan-Meier, ...) cung cấp cái nhìn tổng quan và trực quan về tình trạng bệnh của cộng đồng bệnh nhân, hỗ trợ bác sĩ đưa ra nhận định và quyết định lâm sàng chính xác hơn. Người dùng có thể lựa chọn ẩn/hiện hoặc phóng to biểu đồ để xem chú thích dễ hơn. Đặc biệt, đường cong Kaplan-Meier có thể thay đổi dựa vào thời gian.



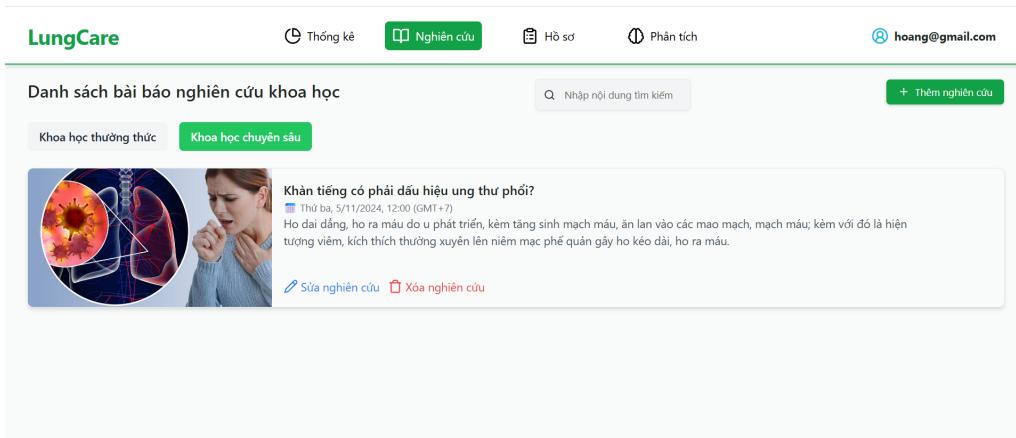
Hình 4.17: Giao diện trang thống kê

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.18: Giao diện biểu đồ Kaplan-Meier khi phóng to

Giao diện trang nghiên cứu quản lý các bài báo và tài liệu nghiên cứu khoa học liên quan đến ung thư phổi. Giao diện được chia thành hai danh mục chính là Khoa học thường thức và Khoa học chuyên sâu, cho phép người dùng dễ dàng phân loại và truy cập nội dung phù hợp. Giao diện cũng cung cấp các nút chức năng như “Sửa nghiên cứu” và “Xóa nghiên cứu”, hỗ trợ người dùng dễ dàng quản lý nội dung.



Hình 4.19: Giao diện trang nghiên cứu

Giao diện trang danh sách mẫu bệnh phẩm gồm tiêu đề "Bảng dữ liệu bệnh phẩm" cùng thanh công cụ hỗ trợ tìm kiếm theo mã mẫu bệnh phẩm, nút "Thêm bản ghi" để thêm mới dữ liệu, và nút "Xuất Excel" để tải bảng dữ liệu. Phần chính của trang là bảng dữ liệu, bao gồm các cột như: Hành động (với các biểu tượng xem, sửa, xoá), STT, mã bệnh nhân, loại ung thư, mã mẫu bệnh phẩm, tuổi chẩn đoán, giới tính, tình trạng sống, giai đoạn AJCC, vị trí sinh thiết, chẩn đoán chính và năm chẩn đoán. Mỗi hàng trong bảng đại diện cho một mẫu bệnh phẩm, với các checkbox để lựa chọn hàng loạt. Cuối bảng là thanh điều hướng phân trang cho phép chọn số bản ghi hiển thị trên mỗi trang (10, 20, 50, 100) và chuyển đổi giữa các trang dữ liệu. Giao diện này thể hiện một thiết kế điển hình của hệ thống quản lý hồ sơ bệnh nhân, hỗ trợ theo dõi, chỉnh sửa và phân tích dữ liệu bệnh học một

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

cách trực quan và hiệu quả.

Hành động	STT	Mã bệnh nhân	Loại ung thư	Mã mẫu bệnh phẩm	Tuổi chẩn đoán	Giới tính	Tình trạng	Giai đoạn AJCC	Vị trí sinh thiết	Chẩn đoán chính	Năm ch.
	1	TCGA-05-4244	Lung Adenocarci...	TCGA-05-4244-01	70	Male	Alive	Stage IV	Lower lobe, lung	Adenocarcinoma...	2009
	2	TCGA-05-4384	Lung Adenocarci...	TCGA-05-4384-01	66	Male	Alive	Stage IIIA	Lower lobe, lung	Adenocarcinoma...	2009
	3	TCGA-05-4389	Lung Adenocarci...	TCGA-05-4389-01	70	Male	Alive	Stage IA	Upper lobe, lung	Adenocarcinoma...	2005
	4	TCGA-05-4402	Lung Adenocarci...	TCGA-05-4402-01	56	Female	10 20 50 100	Stage IV	Lower lobe, lung	Adenocarcinoma...	2007
	5	TCGA-05-4417	Lung Adenocarci...	TCGA-05-4417-01	51	Female	10 50 100	Stage IB	Upper lobe, lung	Adenocarcinoma...	2008

Bản ghi 1-10 của 567 bản ghi

Bản ghi / trang: 1 2 3 4 5 ... 57

Hình 4.20: Giao diện trang danh sách mẫu bệnh phẩm

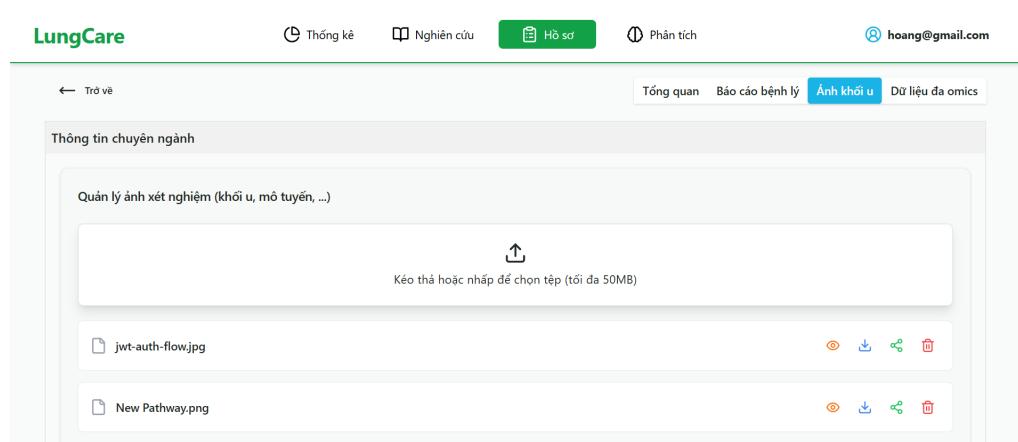
Giao diện trang chi tiết mẫu bệnh phẩm được chia thành hai cột chính, hiển thị đầy đủ các thông tin lâm sàng liên quan đến một mẫu bệnh phẩm cụ thể. Bên trên là nút “Trở về” giúp quay lại trang bảng dữ liệu. Tiếp theo là các tab điều hướng nhỏ gồm: “Tổng quan” (đang được chọn), “Báo cáo bệnh lý”, “Ảnh khói u”, “Dữ liệu thô” và “Dữ liệu đã xử lý” – thể hiện các nhóm dữ liệu liên quan đến mẫu bệnh phẩm.

← Trở về TCGA-05-4244-01		Tổng quan	Báo cáo bệnh lý	Ảnh khói u	Dữ liệu thô	Dữ liệu đã xử lý																																											
Thông tin lâm sàng <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Mã bệnh nhân</td><td>TCGA-05-4244</td></tr> <tr><td>Mã mẫu bệnh phẩm</td><td>TCGA-05-4244-01</td></tr> <tr><td>Tuổi lúc chẩn đoán</td><td>70</td></tr> <tr><td>Vị trí sinh thiết</td><td>Lower lobe, lung</td></tr> <tr><td>Loại ung thư</td><td>Lung Adenocarcinoma</td></tr> <tr><td>Số tháng không bệnh (sau điều trị)</td><td>NA</td></tr> <tr><td>Tình trạng không bệnh</td><td>NA</td></tr> <tr><td>Loại bệnh</td><td>Adenomas and Adenocarcinomas</td></tr> <tr><td>Dân tộc</td><td>NA</td></tr> <tr><td>Tỷ lệ bô gen bị biến đổi</td><td>0.4448</td></tr> <tr><td>Phân loại theo mã ICD-10</td><td>C34.3</td></tr> </table>		Mã bệnh nhân	TCGA-05-4244	Mã mẫu bệnh phẩm	TCGA-05-4244-01	Tuổi lúc chẩn đoán	70	Vị trí sinh thiết	Lower lobe, lung	Loại ung thư	Lung Adenocarcinoma	Số tháng không bệnh (sau điều trị)	NA	Tình trạng không bệnh	NA	Loại bệnh	Adenomas and Adenocarcinomas	Dân tộc	NA	Tỷ lệ bô gen bị biến đổi	0.4448	Phân loại theo mã ICD-10	C34.3	Thông tin lâm sàng <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Mô tả di căn xa (M) theo AJCC</td><td>M1</td></tr> <tr><td>Mô tả hách (N) theo AJCC</td><td>N2</td></tr> <tr><td>Giai đoạn bệnh lý theo AJCC</td><td>Stage IV</td></tr> <tr><td>Mô tả khói u (T) theo AJCC</td><td>T2</td></tr> <tr><td>Chẩn đoán chính</td><td>Adenocarcinoma, NOS</td></tr> <tr><td>Vị trí khói u chính</td><td>Bronchus and lung</td></tr> <tr><td>Có tiền sử ung thư ác tính trước đó</td><td>FALSE</td></tr> <tr><td>Đã từng điều trị trước đó</td><td>FALSE</td></tr> <tr><td>Loại mẫu</td><td>Primary solid Tumor</td></tr> <tr><td>Giới tính</td><td>Male</td></tr> <tr><td>Số năm hút thuốc</td><td>NA</td></tr> </table>	Mô tả di căn xa (M) theo AJCC	M1	Mô tả hách (N) theo AJCC	N2	Giai đoạn bệnh lý theo AJCC	Stage IV	Mô tả khói u (T) theo AJCC	T2	Chẩn đoán chính	Adenocarcinoma, NOS	Vị trí khói u chính	Bronchus and lung	Có tiền sử ung thư ác tính trước đó	FALSE	Đã từng điều trị trước đó	FALSE	Loại mẫu	Primary solid Tumor	Giới tính	Male	Số năm hút thuốc	NA			
Mã bệnh nhân	TCGA-05-4244																																																
Mã mẫu bệnh phẩm	TCGA-05-4244-01																																																
Tuổi lúc chẩn đoán	70																																																
Vị trí sinh thiết	Lower lobe, lung																																																
Loại ung thư	Lung Adenocarcinoma																																																
Số tháng không bệnh (sau điều trị)	NA																																																
Tình trạng không bệnh	NA																																																
Loại bệnh	Adenomas and Adenocarcinomas																																																
Dân tộc	NA																																																
Tỷ lệ bô gen bị biến đổi	0.4448																																																
Phân loại theo mã ICD-10	C34.3																																																
Mô tả di căn xa (M) theo AJCC	M1																																																
Mô tả hách (N) theo AJCC	N2																																																
Giai đoạn bệnh lý theo AJCC	Stage IV																																																
Mô tả khói u (T) theo AJCC	T2																																																
Chẩn đoán chính	Adenocarcinoma, NOS																																																
Vị trí khói u chính	Bronchus and lung																																																
Có tiền sử ung thư ác tính trước đó	FALSE																																																
Đã từng điều trị trước đó	FALSE																																																
Loại mẫu	Primary solid Tumor																																																
Giới tính	Male																																																
Số năm hút thuốc	NA																																																

Hình 4.21: Giao diện trang chi tiết mẫu bệnh phẩm

Giao diện trang quản lý tệp dữ liệu giúp bác sĩ dễ dàng thêm, quản lý và xử lý các file dữ liệu như hình ảnh khói u, kết quả xét nghiệm, dữ liệu lâm sàng, v.v gắn liền với từng bản ghi bệnh nhân, từ đó phục vụ cho việc chẩn đoán, nghiên cứu và theo dõi bệnh một cách trực quan và hiệu quả.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.22: Giao diện trang quản lý tệp dữ liệu

Giao diện trang xem file dữ liệu thô cho phép người dùng truy cập và quan sát trực quan nội dung các tệp định dạng văn bản, như ảnh bên dưới là ví dụ về định lượng miRNA trong nghiên cứu sinh học phân tử, giúp bác sĩ và nhà nghiên cứu kiểm tra nhanh các thông tin như số lượt đọc, vị trí isoform và trạng thái mapping, hỗ trợ phân tích và ra quyết định chính xác.

The screenshot shows the LungCare software interface. On the left, there is a sidebar with 'LungCare' at the top, followed by 'Trở về' (Back), 'TC' (highlighted in blue), and a list of files: 'Quản lý dữ liệu' (Data Management), '19905453', '3f03c090', and '3f03c090'. The main area is titled 'Xem tệp: 3f03c090-78c0-4b3c-9541-3fa95da8db86.mirbase21.isoforms.quantification.txt'. It displays a table with the following data:

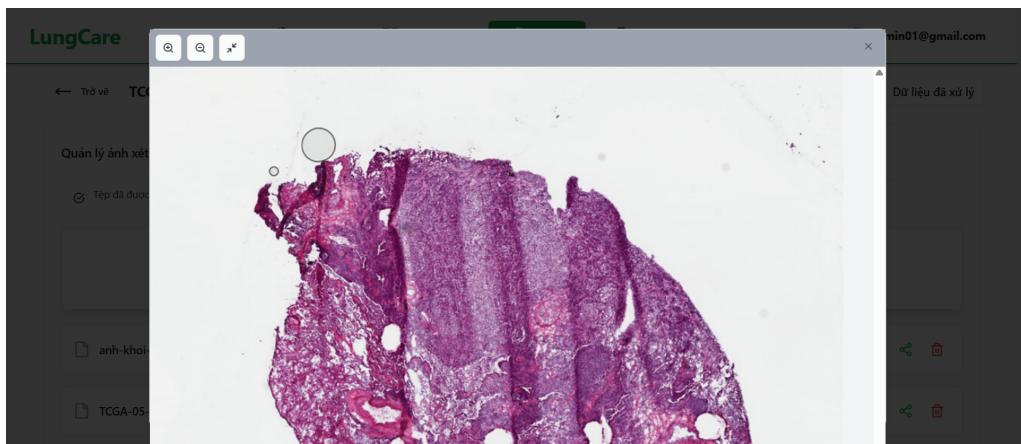
miRNA_ID	isoform_coords	read_count	reads_per_million_miRNA_...	cross-mapped
hsa-let-7a-1	hg38:chr9:94175960-94175...	2	0.785951	N
hsa-let-7a-1	hg38:chr9:94175961-94175...	1	0.392975	N
hsa-let-7a-1	hg38:chr9:94175962-94175...	27	10.610334	N
hsa-let-7a-1	hg38:chr9:94175962-94175...	757	297.482325	N
hsa-let-7a-1	hg38:chr9:94175962-94175...	800	314.380264	N
hsa-let-7a-1	hg38:chr9:94175962-94175...	3099	1217.830548	N
hsa-let-7a-1	hg38:chr9:94175962-94175...	99	38.904558	N
hsa-let-7a-1	hg38:chr9:94175963-94175...	4	1.571901	N
hsa-let-7a-1	hg38:chr9:94175963-94175...	3	1.178926	N
hsa-let-7a-1	hg38:chr9:94175963-94175...	33	12.968186	N
hsa-let-7a-1	hg38:chr9:94175963-94175...	2	0.785951	N
hsa-let-7a-1	hg38:chr9:94175965-94175...	3	1.178926	N
hsa-let-7a-1	hg38:chr9:94175965-94175...	11	4.322729	N
hsa-let-7a-1	hg38:chr9:94175965-94175...	1	0.392975	N
hsa-let-7a-1	hg38:chr9:94175966-94175...	1	0.392975	N

On the right side, there is a sidebar titled 'Dữ liệu đã xử lý' (Processed Data) with three rows of icons for file operations.

Hình 4.23: Giao diện trang xem file dữ liệu thô

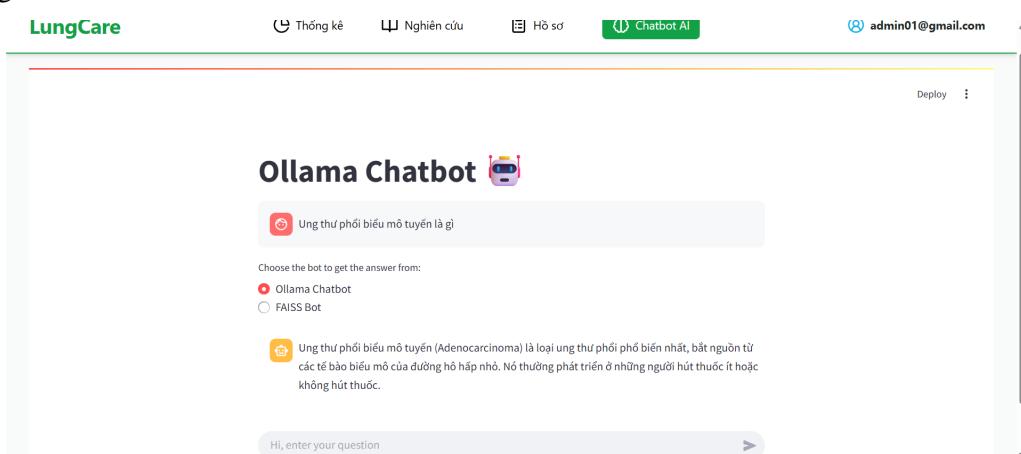
Giao diện trang xem file ảnh xét nghiệm cho phép người dùng phóng to, thu nhỏ và quan sát chi tiết mẫu mô bệnh học với chất lượng hình ảnh cao, hỗ trợ bác sĩ trong việc đánh giá tổn thương, phân tích đặc điểm tế bào và phục vụ cho quá trình chẩn đoán ung thư chính xác hơn.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.24: Giao diện trang xem file ảnh xét nghiệm

Giao diện hỏi đáp với chatbot AI hỗ trợ bác sĩ và người dùng tìm kiếm nhanh các thông tin liên quan đến bệnh ung thư phổi. Người dùng có thể nhập câu hỏi và chọn giữa hai mô hình chatbot để nhận câu trả lời, giúp việc tra cứu trở nên dễ dàng, nhanh chóng và chính xác hơn trong bối cảnh nghiên cứu hoặc tư vấn lâm sàng.



Hình 4.25: Giao diện hỏi đáp với chatbot

4.4 Kiểm thử

4.4.1 Kiểm thử chức năng "Thêm bài báo nghiên cứu"

Áp dụng kiểm thử hộp đen để kiểm thử ràng buộc trường và kiểm thử kiểu dữ liệu. Cụ thể, với mỗi trường được đánh dấu "Bắt buộc", hệ thống phải hiển thị thông báo lỗi nếu người dùng để trống hoặc nhập sai định dạng

Bảng 4.13: Mô tả các trường thông tin form thêm bài nghiên cứu

STT	Tên trường	Mô tả
1	Mã bài nghiên cứu	Bắt buộc. Định dạng chuỗi viết liền
2	Thể loại	Bắt buộc
3	Tiêu đề	Bắt buộc

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

STT	Tên	Mô tả
4	Ngày đăng	Bắt buộc. Định dạng ngày tháng năm
5	Nhóm tác giả	Bắt buộc
6	Nguồn bài viết	Bắt buộc
7	Mô tả	Bắt buộc
8	Ảnh minh họa	Bắt buộc. Định dạng ảnh
9	Nội dung chi tiết	Bắt buộc. Định dạng văn bản

4.4.2 Kiểm thử chức năng "Thêm dữ liệu bệnh phẩm"

Áp dụng kiểm thử hộp đen để kiểm thử ràng buộc trường và kiểm thử kiểu dữ liệu. Hai trường chính “Mã bệnh nhân” và “Mã mẫu bệnh phẩm” được kiểm thử để đảm bảo phải nhập đúng định dạng chuỗi viết liền. Các trường thông tin khác có thể để trống do thiếu thông tin và dữ liệu mẫu có sẵn chưa được chuẩn hóa.

Bảng 4.14: Mô tả các trường thông tin form thêm dữ liệu bệnh phẩm

STT	Tên trường	Mô tả
1	Mã bệnh nhân	Bắt buộc. Định dạng chuỗi viết liền
2	Mã mẫu bệnh phẩm	Bắt buộc. Định dạng chuỗi viết liền
3	Các trường thông tin khác	Không bắt buộc. Định dạng chuỗi hoặc số

4.4.3 Kiểm thử chức năng "Tải lên file dữ liệu"

Áp dụng kiểm thử hộp đen gồm phân vùng tương đương và phân tích giá trị biên. Cụ thể, nghiêm thu cả kịch bản thông thường và kịch bản ngoại lệ qua đó đảm bảo hệ thống vừa xử lý đúng dữ liệu đúng yêu cầu, vừa phát hiện và trả về lỗi chính xác khi gặp đầu vào sai.

Bảng 4.15: Kết quả kiểm thử chức năng "Tải lên file dữ liệu"

STT	Dữ liệu đầu vào	Kết quả mong đợi	Trạng thái
1	Kích thước file lớn hơn 50MB	Thông báo lỗi: Kích thước tệp tin lớn hơn 50MB	Đạt
2	Tên file chứa dấu cách hoặc không đúng định dạng	Thông báo lỗi: Không thể tải tệp lên	Đạt

STT	Dữ liệu đầu vào	Kết quả mong đợi	Trạng thái
3	File đã tồn tại trong bucket	Thông báo lỗi: Tệp tin đã tồn tại	Đạt
4	Tải file lên không đúng bucket	Thông báo lỗi: Bucket không tồn tại	Đạt

4.5 Triển khai

Trong giai đoạn triển khai, toàn bộ ứng dụng được vận hành trên máy chủ ảo (VPS) với địa chỉ IP công khai 45.117.80.79. Máy chủ sử dụng hệ điều hành Ubuntu Server, được cấu hình với 5 lõi CPU, 5 GB bộ nhớ RAM và ổ đĩa dung lượng 80 GB. Cấu hình này đáp ứng đầy đủ nhu cầu vận hành cho các thành phần backend, cơ sở dữ liệu và dịch vụ lưu trữ của hệ thống. Ứng dụng backend (sử dụng Express.js) và frontend (phát triển bằng ReactJS) được đóng gói và triển khai dưới dạng các container Docker, giúp cài đặt môi trường, đơn giản hóa việc triển khai và nâng cấp khi cần thiết. MongoDB được triển khai trong một container riêng biệt, vận hành ở chế độ replica set đơn để đảm bảo tính sẵn sàng của dữ liệu. Dữ liệu hình ảnh và dữ liệu multi-omics được lưu trữ trên Supabase Storage thông qua các bucket chuyên biệt.

Với mô hình triển khai này, hệ thống vừa đảm bảo khả năng vận hành ổn định, vừa dễ dàng mở rộng và bảo trì trong tương lai, đồng thời tạo tiền đề thuận lợi cho việc bổ sung thêm các tính năng phân tích nâng cao, tích hợp trí tuệ nhân tạo hoặc hỗ trợ nghiên cứu dữ liệu y sinh chuyên sâu hơn.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Sau khi hệ thống đã được xây dựng và thử nghiệm, Chương 5 tập trung vào các giải pháp tối ưu hóa được áp dụng để nâng cao hiệu suất, khả năng xử lý dữ liệu và mở rộng chức năng. Chương này sẽ trình bày chi tiết các phương pháp phân tích dữ liệu và công cụ chạy các mô hình ngôn ngữ lớn phục vụ xây dựng chatbot. Đồng thời, chương tổng hợp những đóng góp nổi bật từ quá trình phát triển, làm rõ giá trị mà hệ thống mang lại cho nhóm nghiên cứu bệnh ung thư phổi.

5.1 Phương pháp Kaplan-Meier

Trong đề tài này, việc đánh giá hiệu quả của các phác đồ điều trị là một nhiệm vụ quan trọng, đặc biệt thông qua chỉ số thời gian sống sót tổng thể (OS) – tức thời gian bệnh nhân sống sót sau khi bắt đầu can thiệp điều trị. Tuy nhiên, dữ liệu thời gian đến sự kiện (time-to-event data) thường bị nhiễu, chủ yếu là do các ca bệnh bị kiểm duyệt. Các trường hợp này bao gồm bệnh nhân rút lui, chuyển viện hoặc tử vong do nguyên nhân không liên quan trực tiếp đến ung thư phổi, điều này khiến kết quả phân tích trở nên phức tạp nếu chỉ dựa trên các phương pháp thống kê truyền thống.

Vì vậy, việc áp dụng phương pháp Kaplan–Meier trở thành giải pháp tối ưu nhằm mô tả đường cong sống sót của bệnh nhân theo thời gian, giúp thu được đánh giá trực quan và định lượng về hiệu quả điều trị. Phương pháp Kaplan–Meier được đặt theo tên của hai nhà thống kê là Edward L. Kaplan và Paul Meier, sau khi hợp tác và công bố một bài báo vào năm 1958 về cách xử lý dữ liệu thời gian đến sự kiện. Trong bài báo này, họ đã giới thiệu ước lượng Kaplan–Meier, một công cụ nhằm đo lường tần suất hoặc số lượng bệnh nhân sống sót sau điều trị y tế. [7]

Hệ thống đã tích hợp phương pháp Kaplan–Meier vào hệ thống phân tích dữ liệu ung thư phổi nhằm xây dựng biểu đồ OS trực quan. Phương pháp Kaplan–Meier hoạt động dựa trên nguyên tắc tính toán tỷ lệ sống sót tại mỗi mốc thời gian, bằng cách chia số lượng bệnh nhân còn sống cho số bệnh nhân có nguy cơ tại thời điểm đó, đồng thời xử lý đúng các trường hợp kiểm duyệt. Thông qua cách tiếp cận này, các khoảng thời gian giữa các sự kiện (tử vong hay tiến triển bệnh) được phân tích độc lập, sau đó kết quả được nhân liên tiếp lại để tạo thành đường cong sống sót. Trong hệ thống, dữ liệu bệnh nhân được thu thập và chuyển đổi thành bảng sống còn, từ đó tính toán tạo ra một đường cong Kaplan–Meier chi tiết, biểu diễn trực quan mức sống sót theo thời gian. Giải pháp này không chỉ đảm bảo tính chính xác khi xử lý dữ liệu kiểm duyệt mà còn cung cấp thông tin định lượng như thời gian sống còn trung vị (xác suất sống còn là 50%) và khoảng tin cậy, hỗ trợ bác sĩ trong

việc đánh giá hiệu quả điều trị và đưa ra quyết định lâm sàng kịp thời.

Việc tích hợp Kaplan-Meier đã mang lại những kết quả rõ ràng và thiết thực. Qua các thử nghiệm trên bộ dữ liệu mẫu có thời gian theo dõi kéo dài 24 tháng, đường cong sống sót tổng thể được tạo ra phản ánh một cách trực quan xu hướng sống sót của bệnh nhân, với khả năng xác định giá trị thời gian sống còn trung vị cùng các khoảng tin cậy 95% một cách tự động. Những kết quả này không chỉ khớp với phân tích truyền thống mà còn giúp giảm đáng kể thời gian xử lý và báo cáo dữ liệu, cải thiện hiệu quả ra quyết định trong điều trị. Hơn nữa, biểu đồ Kaplan-Meier cung cấp cho các bác sĩ một công cụ trực quan để đánh giá, so sánh hiệu quả điều trị giữa các nhóm bệnh nhân, từ đó giúp định hướng điều chỉnh phác đồ y tế kịp thời và chính xác. Nhờ đó, hiệu quả của quá trình theo dõi và chăm sóc bệnh nhân được nâng cao, tạo điều kiện thuận lợi cho việc nghiên cứu và hỗ trợ quyết định lâm sàng trong bối cảnh ung thư phổi – một lĩnh vực đòi hỏi sự chính xác và nhanh chóng trong phân tích và xử lý dữ liệu.

5.2 Biểu đồ Kaplan-Meier

Ung thư phổi là một thách thức lớn đối với sức khỏe cộng đồng, với tỷ lệ sống sót biến động đáng kể tùy thuộc vào nhiều yếu tố như giai đoạn chẩn đoán, phương pháp điều trị và đặc điểm nhân khẩu học của bệnh nhân. Để hiểu rõ hơn về sự biến động này và cải thiện kết quả điều trị cho bệnh nhân, phân tích sống sót đóng vai trò quan trọng. Cụ thể, phương pháp ước lượng Kaplan-Meier là một công cụ thống kê không tham số giúp ước tính hàm sống sót từ dữ liệu thời gian sống, từ đó cung cấp cái nhìn sâu sắc về xác suất sống sót theo thời gian. Trong dự án này, chúng tôi tập trung vào việc so sánh trải nghiệm sống sót giữa các nhóm bệnh nhân ung thư phổi khác nhau. Bằng cách phân tích sự khác biệt trong xác suất sống sót giữa các nhóm, chúng tôi có thể xác định các yếu tố liên quan đến kết quả tốt hơn hoặc tệ hơn, từ đó hỗ trợ ra quyết định lâm sàng và định hướng cho các nghiên cứu tương lai.

Để giải quyết vấn đề này, biểu đồ Kaplan-Meier động được xây dựng để biểu diễn các đường cong sống sót cho nhiều nhóm bệnh nhân, mỗi nhóm được phân biệt bằng một màu sắc riêng. Trục x của biểu đồ biểu thị thời gian tính bằng ngày, trong khi trục y biểu thị xác suất sống sót. Một thanh trượt tích hợp cho phép người dùng điều chỉnh động khoảng thời gian, giúp tập trung phân tích vào các khoảng thời gian cụ thể mà họ quan tâm. Ngoài ra, các chú thích công cụ cung cấp thông tin chi tiết về xác suất sống sót tại bất kỳ thời điểm nào được chọn, và một bảng chú giải rõ ràng giúp dễ dàng nhận diện từng nhóm. Cách trình bày trực quan này cho phép đánh giá nhanh chóng sự khác biệt trong mô hình sống sót, làm nổi bật

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

các thời điểm quan trọng mà tại đó xác suất sống sót bắt đầu khác biệt đáng kể, và cho phép khám phá sâu hơn về xu hướng sống sót theo thời gian.



Hình 5.1: Biểu đồ phân tích sống sót Kaplan-Meier theo nhóm giới tính và nhóm tuổi

Biểu đồ bên phải trong ảnh trên là một ví dụ biểu đồ Kaplan–Meier so sánh tỷ lệ sống sót (Overall Survival) theo ba nhóm độ tuổi khác nhau: 24–44 (màu xanh dương), 44–60 (màu cam) và trên 60 (màu xanh lá). Biểu đồ được hiển thị trong một khung ResponsiveContainer rộng, với trục hoành là thời gian tính theo ngày và trục tung là tỷ lệ sống sót, dải giá trị từ 0% đến 100%. Ở phía trên cùng có một thanh trượt “Giới hạn thời gian” kèm ô nhập số, cho phép điều chỉnh mốc thời gian tối đa hiển thị trên trục hoành (ở ví dụ này tối đa là 7248 ngày). Khi kéo thanh trượt hoặc nhập số, đồ thị sẽ cắt bớt dữ liệu sau mốc đó để tập trung vào giai đoạn quan tâm. Mỗi đường cong thể hiện tỷ lệ sống sót lũy tiến của nhóm tương ứng theo kiểu “step-after” (bước nhảy ngay sau mỗi biến cố). Các biến cố (sự kiện tử vong) được đánh dấu bằng ký hiệu “+” lớn dọc theo đường cong, giúp dễ dàng nhận biết thời điểm có biến cố xảy ra. Càng về sau, các đường cong dần hạ thấp, phản ánh sự giảm dần tỷ lệ sống sót theo thời gian. Nhóm trẻ nhất (24–44) giữ tỷ lệ sống cao nhất ở cuối giai đoạn, trong khi nhóm trên 60 tuổi có đường cong xuống thấp nhất, cho thấy nguy cơ tử vong cao hơn ở nhóm lớn tuổi. Phía dưới biểu đồ là chú thích đặt ở giữa, hiển thị tên nhóm kèm màu tương ứng, giúp người xem dễ phân biệt từng đường cong.

Việc triển khai biểu đồ so sánh Kaplan-Meier này đã chứng minh giá trị to lớn trong dự án phân tích dữ liệu ung thư phổi của chúng tôi. Nó cho phép các nhà nghiên cứu khám phá dữ liệu sống sót một cách trực quan và tương tác, dẫn đến sự hiểu biết sâu sắc hơn về cách các yếu tố khác nhau ảnh hưởng đến kết quả của bệnh nhân. Bằng cách tạo điều kiện so sánh các đường cong sống sót giữa các nhóm, biểu đồ đã giúp xác định các xu hướng và mô hình mà có thể bị che khuất trong dữ liệu bảng. Do đó, nó đã trở thành một công cụ thiết yếu để tạo ra các giả thuyết mới và truyền đạt dữ liệu sống sót phức tạp một cách dễ tiếp cận.

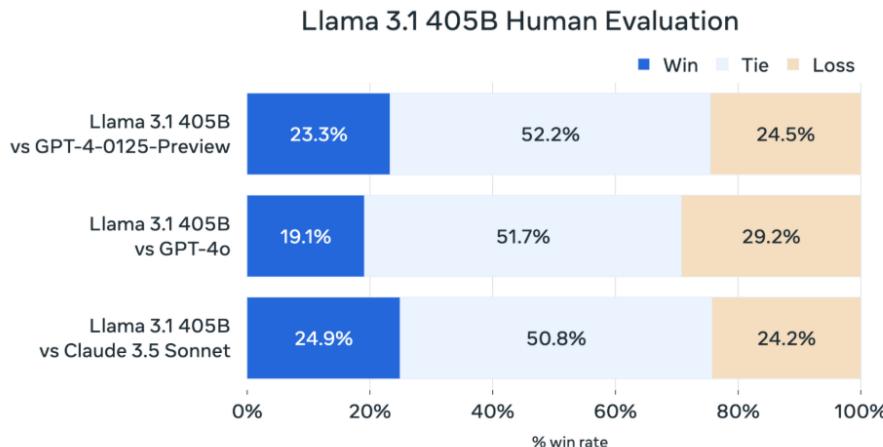
5.3 Chatbot Ollama

Hiện nay, việc cung cấp thông tin chuyên sâu và kịp thời thông qua hệ thống tư vấn tự động là một nhu cầu cấp bách. Các bác sĩ và nhà nghiên cứu thường đối mặt với việc tra cứu phác đồ điều trị, giải đáp thắc mắc của bệnh nhân hoặc rà soát các hướng dẫn lâm sàng từ khối lượng tài liệu khổng lồ như bài báo khoa học và báo cáo hội chẩn. Hơn nữa, yêu cầu về tính bảo mật dữ liệu và phản hồi nhanh khiến việc sử dụng các dịch vụ bên ngoài thông qua đám mây trở nên không phù hợp. Vì vậy, việc triển khai một chatbot nội bộ, chuyên dụng cho lĩnh vực ung thư phổi, dựa trên mô hình Llama 3.1 của Ollama là bước giải quyết thiết yếu, nhằm đảm bảo cung cấp thông tin chính xác, kịp thời và phù hợp với đặc thù của bệnh lý.

Để giải quyết bài toán nêu trên, hệ thống đã tích hợp một chatbot dựa trên phiên bản Llama 3.1 của Ollama, được tích hợp chặt chẽ với framework LangChain. Giải pháp được xây dựng theo mô hình "retrieval-augmented generation" (RAG), nhằm tối ưu hóa khả năng truy xuất thông tin từ dữ liệu y khoa và cung cấp những câu trả lời có căn cứ khi người dùng đặt câu hỏi. Cụ thể, trước hết hệ thống sử dụng các thư viện như SentenceTransformer để nhúng văn bản từ các nguồn tài liệu như hướng dẫn, bài báo và báo cáo hội chẩn thành các vector có chiều (ví dụ: 768 chiều). Các vector này được lưu trữ và lập chỉ mục bằng FAISS, giúp việc truy xuất nhanh chóng các tài liệu liên quan ngay khi có câu hỏi. Khi người dùng tương tác với chatbot, câu hỏi được chuyển đổi thành vector để so khớp với các tài liệu có liên quan. Kết quả của quá trình truy xuất này cùng với lịch sử hội thoại được ghép vào một mẫu prompt có sẵn, được đóng vai trò "hướng dẫn" cho mô hình trả lời. Mẫu prompt này được xây dựng kỹ càng với nội dung yêu cầu trả lời ngắn gọn, không bịa đặt thông tin và chỉ cung cấp dữ liệu có căn cứ từ tài liệu y khoa. Sau đó, mô hình Llama 3.1, chạy trên hạ tầng nội bộ với hiệu năng cao, sẽ sinh ra câu trả lời. Ứng dụng được triển khai thông qua giao diện Streamlit, đảm bảo tính trực quan và phù hợp với yêu cầu bảo mật nội bộ.

Việc triển khai chatbot Ollama với Llama 3.1 đã mang lại những cải tiến đáng kể cho hệ thống tư vấn y khoa trong dự án. Đầu tiên, thời gian phản hồi trung bình được giữ dưới 2 giây (với phần cứng đủ mạnh), đảm bảo trải nghiệm người dùng mượt mà và kịp thời trong các trường hợp cần tư vấn ngay lập tức. Mô hình được tinh chỉnh kỹ lưỡng thông qua prompt chuyên sâu nên giảm thiểu khả năng "hallucination" (bắt buộc cung cấp thông tin khi không có đủ dữ liệu) xuống mức thấp, từ đó tăng độ chính xác của câu trả lời. Hơn nữa, khả năng tự động cập nhật và truy xuất dữ liệu từ các nguồn tài liệu y khoa nội bộ giúp chatbot không chỉ trở thành một công cụ tư vấn hữu ích mà còn là trung tâm xử lý thông tin quan trọng, góp phần giảm thiểu thời gian và công sức của đội ngũ y khoa trong việc tìm kiếm

và xác minh thông tin. Qua đó, dự án đã tạo ra một giải pháp toàn diện, kết hợp giữa hiệu năng xử lý và độ chính xác, đáp ứng tốt nhu cầu đặc thù trong lĩnh vực ung thư phổi.



Hình 5.2: Bảng so sánh mô hình Llama 3.1 405B với các mô hình khác [8]

5.4 Nền tảng Streamlit

Trong thời đại công nghệ số phát triển mạnh mẽ, chatbot AI ngày càng trở thành công cụ không thể thiếu trong nhiều lĩnh vực như hỗ trợ khách hàng, giáo dục và quản lý dữ liệu. Tuy nhiên, phần lớn các giải pháp chatbot hiện nay phụ thuộc vào các dịch vụ đám mây, dẫn đến những lo ngại về bảo mật dữ liệu, chi phí cao và sự phụ thuộc vào kết nối internet. Đối với những cá nhân hoặc tổ chức muốn kiểm soát dữ liệu của mình hoặc giảm chi phí vận hành, việc phát triển một chatbot cục bộ là một nhu cầu cấp thiết. Thách thức nằm ở việc cần kết hợp một mô hình ngôn ngữ mạnh mẽ với một giao diện người dùng thân thiện, điều này có thể gây khó khăn cho những người không có nhiều kinh nghiệm về phát triển web. Trong bối cảnh này, việc tìm kiếm một nền tảng dễ sử dụng nhưng hiệu quả trở thành trọng tâm.

Nền tảng Streamlit, một thư viện Python mã nguồn mở, đã xuất hiện như một giải pháp lý tưởng để giải quyết vấn đề trên. Streamlit cho phép người dùng tạo các ứng dụng web tương tác chỉ với mã Python đơn giản, phù hợp cho các dự án liên quan đến khoa học dữ liệu và học máy [9]. Khi kết hợp với Ollama, một nền tảng mã nguồn mở cho phép chạy các mô hình ngôn ngữ lớn cục bộ, giải pháp này mang lại khả năng bảo mật cao và giảm phụ thuộc vào dịch vụ bên ngoài. Quá trình triển khai bao gồm việc cài đặt Ollama để quản lý các mô hình ngôn ngữ, sau đó sử dụng Streamlit để thiết kế giao diện cho phép người dùng nhập câu hỏi và nhận phản hồi trực tiếp từ mô hình được chọn. Sự tích hợp này không chỉ đơn giản hóa việc phát triển mà còn tạo ra một chatbot linh hoạt, có thể tùy chỉnh theo nhu cầu cụ thể, chẳng hạn như hỗ trợ phân tích dữ liệu y sinh trong phần mềm hiện tại.



Hình 5.3: Giao diện chatbot Ollama chạy trên Streamlit

Nhiều dự án thực tế đã chứng minh hiệu quả của việc kết hợp Streamlit và Ollama trong việc xây dựng chatbot cục bộ. Các hướng dẫn và tài liệu trên các nền tảng như GitHub đã giúp người dùng tạo ra các ứng dụng chatbot với các tính năng như phản hồi theo thời gian thực, chọn mô hình ngôn ngữ và quản lý lịch sử trò chuyện. Những dự án này không chỉ đảm bảo quyền riêng tư dữ liệu mà còn giảm chi phí so với các giải pháp dựa trên đám mây. Trong trường hợp phần mềm hiện tại liên quan đến quản lý dữ liệu y sinh ung thư phổi, chatbot này có thể hỗ trợ các bác sĩ hoặc nhà nghiên cứu bằng cách cung cấp thông tin tức thì dựa trên dữ liệu cục bộ, mang lại hiệu quả cao và bảo mật tối ưu.

5.5 Bản đồ bộ gen ung thư TCGA

Ung thư phổi đặc biệt là ung thư phổi không tế bào nhỏ (NSCLC), là một trong những nguyên nhân hàng đầu gây tử vong do ung thư trên toàn cầu với hàng triệu ca mắc và tử vong mỗi năm theo thống kê của Tổ chức Y tế Thế giới (WHO). Sự phức tạp của bệnh nằm ở sự đa dạng di truyền, nơi các yếu tố di truyền đóng vai trò quan trọng trong việc khởi phát, tiến triển và phản ứng với các phương pháp điều trị. Các phương pháp chẩn đoán truyền thống dựa trên mô bệnh học thường không thể cung cấp cái nhìn toàn diện về các đặc điểm di truyền này, dẫn đến việc điều trị chưa tối ưu và tiên lượng không khả quan ở nhiều bệnh nhân. Điều này đặt ra nhu cầu cần thiết về một nguồn dữ liệu bộ gen toàn diện để hiểu rõ hơn về cơ sở di truyền của ung thư phổi, từ đó hỗ trợ chẩn đoán sớm, dự báo tiên lượng chính xác và phát triển các liệu pháp cá nhân hóa. Trong bối cảnh hệ thống hiện tại, một phần mềm được thiết kế để quản lý và phân tích dữ liệu y sinh ung thư phổi, việc tích hợp dữ liệu bộ gen chất lượng cao là yếu tố quan trọng để nâng cao hiệu quả nghiên cứu và ứng dụng lâm sàng, giúp các bác sĩ và nhà nghiên cứu đưa ra các quyết định tốt hơn.

Dự án The Cancer Genome Atlas (TCGA) đã trở thành một giải pháp quan trọng bằng cách xây dựng bản đồ bộ gen chi tiết cho ung thư phổi. Được phát triển bởi Viện Y tế Quốc gia Hoa Kỳ (NIH) và Viện Ung thư Quốc gia Hoa Kỳ (NCI), TCGA đã phân tích hàng nghìn mẫu ung thư phổi, bao gồm cả ung thư phổi tuyển tính và ung thư phổi tế bào vảy, sử dụng các công nghệ tiên tiến như trình tự hóa gen và phân tích biểu hiện gen. Dự án này đã tạo ra một kho dữ liệu phong phú về các đặc điểm di truyền liên quan đến ung thư phổi, cung cấp thông tin quan trọng để hiểu rõ hơn về bệnh. Dữ liệu này được lưu trữ công khai trên các nền tảng như Genomic Data Commons và cBioPortal, cho phép các nhà nghiên cứu, bác sĩ, và nhà phát triển phần mềm truy cập và sử dụng để phân tích sâu hơn.

Trong hệ thống hiện tại, dữ liệu TCGA có thể được tích hợp để nâng cao khả năng phân tích các đặc điểm di truyền. Hệ thống có thể được thiết kế để nhập dữ liệu của bệnh nhân và so sánh với dữ liệu TCGA, cung cấp thông tin chi tiết để hỗ trợ các quyết định điều trị. Ví dụ, hệ thống có thể giúp bác sĩ đề xuất các phương pháp điều trị phù hợp dựa trên các đặc điểm di truyền của bệnh nhân, trong khi các nhà nghiên cứu có thể sử dụng dữ liệu để khám phá thêm về các cơ chế di truyền và phát triển các chiến lược điều trị mới. Sự tích hợp này không chỉ tăng cường khả năng phân tích mà còn biến hệ thống thành một công cụ mạnh mẽ trong quản lý và nghiên cứu ung thư phổi.

TCGA đã mang lại những tiến bộ đáng kể trong việc hiểu biết về các đặc điểm di truyền của ung thư phổi, góp phần cải thiện chẩn đoán, tiên lượng và điều trị. Dự án đã xác định các mẫu di truyền quan trọng liên quan đến sự phát triển và tiến triển của ung thư phổi, cho phép phân loại bệnh thành các nhóm khác nhau dựa trên đặc điểm di truyền. Những phân loại này đã định hướng các chiến lược điều trị hiệu quả hơn, hỗ trợ phát triển các liệu pháp nhắm mục tiêu và các mô hình dự báo tiên lượng, từ đó cải thiện khả năng sống còn của bệnh nhân. Hơn nữa, TCGA đã thúc đẩy hàng nghìn nghiên cứu khoa học, khẳng định giá trị của nó trong việc nâng cao hiểu biết về ung thư phổi và ứng dụng lâm sàng.

5.6 Docker

Trong quá trình xây dựng hệ thống, em gặp phải khó khăn khi môi trường chạy thử ở máy cá nhân và máy chủ thực tế không giống nhau. Những khác biệt về phiên bản phần mềm, cấu hình cơ sở dữ liệu hay thư viện hỗ trợ thường dẫn đến lỗi chỉ xuất hiện khi triển khai bản demo, gây mất thời gian khắc phục. Hơn nữa, đội ngũ bác sĩ và nhóm nghiên cứu cần một bản thử nghiệm ổn định để đánh giá nghiệp vụ, nhưng nếu phải cài đặt thủ công từng thành phần từ backend, cơ sở dữ liệu đến giao diện thì rất dễ thiếu sót và không đồng bộ.

Để giải quyết vấn đề đó, hệ thống sử dụng Docker để đóng gói toàn bộ ứng dụng vào các container riêng biệt: backend (Express.js), cơ sở dữ liệu (MongoDB), mô-đun chatbot Ollama và frontend (React). Mỗi container đều có Dockerfile định nghĩa chính xác môi trường, phiên bản thư viện và các biến cấu hình. Chỉ việc chạy vài lệnh cơ bản, hệ thống lập tức khởi động đầy đủ mà không lo thiếu môi trường hay thư viện. Cách tiếp cận này đảm bảo rằng bản demo chạy y hệt môi trường đã phát triển, đồng thời giúp người dùng dễ dàng thiết lập để dùng thử, mà không phải lo về cài đặt phức tạp.

Đến nay, em đã hoàn thiện và đóng gói toàn bộ hệ thống dưới dạng Docker. Bản thử nghiệm sắp được bàn giao cho bác sĩ và nhóm nghiên cứu để thu thập đánh giá nghiệp vụ và góp ý về các chức năng cơ bản. Tóm lại, với khả năng đơn giản hóa quy trình, tối ưu hóa tài nguyên và hỗ trợ các công nghệ tiên tiến, Docker đã trở thành một công cụ không thể thiếu trong phát triển ứng dụng hiện đại.

Chương 5 đã trình bày và đánh giá hiệu quả của các giải pháp nhằm nâng cao khả năng phân tích và xử lý dữ liệu. Việc tích hợp công cụ chạy mô hình ngôn ngữ lớn để xây dựng chatbot không chỉ cải thiện chất lượng phản hồi mà còn cho phép hệ thống hỗ trợ tra cứu thông tin chuyên sâu, góp phần giảm tải cho đội ngũ nghiên cứu. Những đóng góp kỹ thuật này không chỉ nâng cao hiệu quả vận hành mà còn cung cấp khả năng mở rộng lâu dài của hệ thống. Trên cơ sở này, Chương 6 sẽ tiến hành tổng kết toàn bộ đồ án và đề xuất hướng phát triển tiếp theo.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chương 6 sẽ tổng kết toàn bộ nội dung đã thực hiện trong đồ án, đánh giá kết quả đạt được và rút ra bài học kinh nghiệm. Ngoài ra, chương này sẽ đề xuất những hướng phát triển tiếp theo cho hệ thống, nhằm nâng cao hiệu quả ứng dụng trong chẩn đoán sớm và điều trị ung thư phổi.

6.1 Kết luận

Xuất phát mong muốn phát triển một phần mềm chuyên sâu nhằm quản lý và phân tích dữ liệu y sinh liên quan đến bệnh ung thư phổi, em đã thực hiện đề tài “Xây dựng hệ thống quản lý và phân tích dữ liệu y sinh bệnh ung thư phổi”. Quá trình thu thập, khảo sát, phân tích yêu cầu và triển khai giải pháp đều đã được trình bày chi tiết trong đồ án tốt nghiệp này.

Đồ án tốt nghiệp đã đáp ứng được những nhiệm vụ và mục tiêu chính đã nêu. Cụ thể, đồ án đã thiết kế và xây dựng một hệ thống quản lý và phân tích dữ liệu y sinh bệnh ung thư phổi chạy trên nền tảng web. Trước tiên, em đã tiến hành khảo sát hai hệ thống lớn gồm cBioPortal và GDC Data Portal, đánh giá chi tiết ưu nhược điểm. Qua việc khảo sát, em rút ra những tính năng quan trọng giúp quản lý và phân tích dữ liệu ung thư phổi một cách hiệu quả, bao gồm từ khâu thu thập dữ liệu bệnh nhân, lưu trữ kết quả xét nghiệm, hình ảnh y khoa, cho đến phân tích đột biến gen, theo dõi tiến triển bệnh, trực quan hóa kết quả phân tích. Hệ thống cần có khả năng tích hợp nhiều nguồn dữ liệu, hỗ trợ các công cụ thống kê và phân tích chuyên sâu, đảm bảo bảo mật thông tin y tế cũng như cung cấp giao diện trực quan, giúp các nhà nghiên cứu và bác sĩ dễ dàng tiếp cận và khai thác dữ liệu để phục vụ điều trị và nghiên cứu. Tiếp theo, em đi sâu vào phân tích và làm rõ các yêu cầu về chức năng và phi chức năng, từ đó thiết kế chi tiết hệ thống và cơ sở dữ liệu. Hệ thống có các tính năng nổi bật như xem biểu đồ thống kê, hỏi đáp với chatbot AI, quản lý dữ liệu y sinh và quản lý các loại tệp dữ liệu như kết quả xét nghiệm, ảnh khối u, dữ liệu thô. Tính năng xem biểu đồ sử dụng thư viện recharts để trực quan hóa dữ liệu. Đặc biệt, đối với biểu đồ phân tích sống sót Kaplan-Meier có thể được tùy biến dựa vào số ngày người dùng chọn. Tính năng hỏi đáp với chatbot giúp người dùng tiết kiệm thời gian tra cứu và nhận được câu trả lời chính xác hơn về bệnh ung thư phổi. Ở phần quản lý dữ liệu y sinh, hệ thống cho phép thao tác với dữ liệu như tìm kiếm, thêm, xem, sửa và xóa dữ liệu. Với từng mã mẫu bệnh phẩm, người dùng có thể quản lý file dữ liệu tương ứng (được chia thành bốn loại gồm kết quả xét nghiệm, ảnh khối u, dữ liệu thô và dữ liệu đã xử lý).

Quá trình thực hiện đồ án đã mang đến cho em nhiều trải nghiệm quý báu và

giúp em tích lũy kiến thức, kỹ năng chuyên sâu. Thông qua việc xây dựng hệ thống đã giúp em có cơ hội được vận dụng, liên kết kiến thức của các môn học để giải quyết các vấn đề xuất hiện ở nhiều bước trong suốt vòng đời phát triển của một phần mềm. Từ việc khảo sát hiện trạng, phân tích nghiệp vụ bài toán, xác định các yêu cầu hệ thống, thiết kế hệ thống cho đến xây dựng, triển khai hệ thống đã giúp em hiểu rõ tầm quan trọng của một quy trình phát triển phần mềm chuyên nghiệp. Ngoài ra, em cũng nắm được kỹ năng lập kế hoạch chi tiết, phân chia thời gian cho từng công việc, tìm kiếm và nghiên cứu tài liệu, viết báo cáo. Những kiến thức trên đều là những kiến thức quan trọng, là hành trang cần thiết giúp em phát triển sự nghiệp sau này.

Tuy đã cố gắng để hoàn thiện đồ án và đạt được mục tiêu ban đầu đề ra, nhưng do năng lực bản thân còn hạn chế và thời gian làm đồ án có giới hạn, hệ thống vẫn còn tồn đọng một số vấn đề chưa được giải quyết. Thứ nhất, việc tích hợp sâu các mô hình dự đoán AI về tiến triển bệnh ung thư phổi chưa được triển khai, hiện tại chỉ dừng ở mức trực quan hóa kết quả thông kê mà chưa cho phép dự báo tự động. Thứ hai, hệ thống chưa có đủ nguồn dữ liệu chuẩn để huấn luyện và kiểm định các mô hình AI, dẫn đến độ chính xác và tính tổng quát của dự đoán còn hạn chế. Cuối cùng, dung lượng lưu trữ hiện tại chưa đáp ứng được khi khối lượng file kết quả xét nghiệm, ảnh y sinh và dữ liệu y sinh ngày càng tăng, khiến việc mở rộng và duy trì hiệu suất hệ thống trở nên khó khăn.

6.2 Hướng phát triển

Các chức năng mà hệ thống cung cấp đã đáp ứng được các mục tiêu của đồ án, tuy nhiên một số chức năng vẫn cần phải được hoàn thiện hơn nữa để mang lại trải nghiệm tốt hơn cho người dùng. Thứ nhất, cần tích hợp chặt chẽ các mô hình dự đoán AI bằng cách xây dựng các pipeline huấn luyện và triển khai ngay trong môi trường backend, cho phép tự động hóa quy trình nạp dữ liệu, tiền xử lý và chạy dự báo; đồng thời bổ sung giao diện để người dùng tùy chọn tham số mô hình, so sánh kết quả giữa các thuật toán như Cox proportional hazards, Random Survival Forest hay deep learning-based survival. Thứ hai, tập trung khai thác thêm các bộ dữ liệu công khai (như TCGA) và phối hợp với phòng xét nghiệm hoặc bệnh viện để xây dựng kho dữ liệu chuẩn, kèm theo bộ dữ liệu đầy đủ về bệnh nhân, quy trình xét nghiệm và thông số kỹ thuật của thiết bị. Cuối cùng, để đáp ứng nhu cầu lưu trữ ngày càng tăng khi số lượng file dữ liệu mở rộng, hệ thống nên chuyển sang kiến trúc lưu trữ phân tán, ví dụ sử dụng object storage (MinIO, AWS S3), đồng thời triển khai các cơ chế nén ảnh để tối ưu chi phí và duy trì hiệu năng trong dài hạn.

Trên đây là toàn bộ nội dung của đồ án “Xây dựng hệ thống quản lý và phân tích

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

dữ liệu y sinh của bệnh ung thư phổi". Với điều kiện thời gian có hạn và những kỹ năng, kinh nghiệm của bản thân còn nhiều hạn chế, quá trình thực hiện đồ án của em khó tránh khỏi những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để hệ thống có thể được hoàn thiện hơn trong tương lai.

Chương 6 đã tổng hợp toàn bộ tiến trình nghiên cứu, từ việc xác định yêu cầu, lựa chọn công nghệ, thiết kế, triển khai đến tối ưu hóa và đánh giá. Đồ án đã đạt được các mục tiêu đề ra về chức năng, hiệu năng và tính mở rộng. Những kết quả này khẳng định tính khả thi và giá trị nghiên cứu của hệ thống. Trên cơ sở đó, đề xuất bổ sung các module phân tích AI nâng cao, mở rộng tích hợp dữ liệu từ nhiều nguồn và triển khai ứng dụng di động sẽ là bước phát triển tiếp theo, phục vụ tốt hơn việc chẩn đoán và theo dõi điều trị ung thư phổi.

TÀI LIỆU THAM KHẢO

- [1] *Cbioportal for cancer genomics*. [Online]. Available: https://www.cbioportal.org/study/summary?id=luad_tcga_gdc (visited on 06/04/2024).
- [2] *Gdc data portal*. [Online]. Available: <https://portal.gdc.cancer.gov/projects/tcga-luad> (visited on 06/04/2024).
- [3] *Shadcn/ui*. [Online]. Available: <https://ui.shadcn.com/docs> (visited on 06/04/2024).
- [4] *Recharts*. [Online]. Available: <https://recharts.org> (visited on 06/04/2024).
- [5] *Supabase storage documentation*. [Online]. Available: <https://supabase.com/docs/guides/storage> (visited on 06/04/2024).
- [6] *React hook form*. [Online]. Available: <https://react-hook-form.com> (visited on 06/04/2024).
- [7] J. K. MK Goel P Khanna, “Understanding survival analysis: Kaplan-meier estimate,” *International Journal of Ayurveda Research*, vol. 1, no. 4, pp. 274–278, 2010.
- [8] *Llama3.1 library*. [Online]. Available: <https://ollama.com/library/llama3.1> (visited on 06/04/2024).
- [9] *Streamlit documentation*. [Online]. Available: <https://docs.streamlit.io> (visited on 06/04/2024).