

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Nguyễn Thành Dũng

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ THƯ VIỆN ẢNH
TÍCH HỢP AI TẠO VIDEO**

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ thông tin

HÀ NỘI - 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



Nguyễn Thành Dũng

XÂY DỰNG ỨNG DỤNG QUẢN LÝ NHÓM CHAT SỰ
KIỆN TÍCH HỢP AI TẠO VIDEO

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Lê Khánh Trình

Cán bộ đồng hướng dẫn: TS. Đặng Trần Bình

HÀ NỘI - 2025

Lời cảm ơn

Em xin gửi lời cảm ơn sâu sắc đến tập thể Quý Thầy Cô giáo Trường Đại học Công nghệ - ĐHQGHN, đặc biệt là các Giảng viên thuộc Khoa Công Nghệ Thông Tin đã tận tâm truyền đạt kiến thức và kinh nghiệm quý báu cho em trong suốt quá trình học tập vừa qua.

Em xin bày tỏ lòng tri ân chân thành tới TS. Lê Khánh Trình, người thầy đã trực tiếp hướng dẫn và giúp đỡ em trong suốt quá trình thực hiện luận văn này. Những lời góp ý tận tình, những buổi trao đổi học thuật và sự động viên kịp thời của Thầy đã giúp em vượt qua nhiều khó khăn để hoàn thành nghiên cứu của mình.

Em cũng muốn gửi lời cảm ơn đến các bạn học viên đã đồng hành cùng em trong hành trình học tập. Nhờ có sự hỗ trợ, chia sẻ kinh nghiệm và những buổi thảo luận bổ ích với các bạn, em đã có cơ hội tiếp cận nhiều công nghệ mới và mở rộng kiến thức chuyên môn của mình.

Sau cùng, em xin gửi lời cảm ơn đặc biệt đến gia đình, người thân đã luôn ở bên cạnh, hỗ trợ em cả về vật chất và tinh thần. Nhờ có sự yêu thương và tin tưởng của gia đình, em đã có thể an tâm tập trung vào việc học tập và hoàn thành luận văn này.

Lời cam đoan

Tôi là Nguyễn Thành Dũng, sinh viên lớp QH-2021-J khóa K66 theo học ngành Công nghệ thông tin tại trường Đại học Công Nghệ - Đại học Quốc gia Hà Nội. Tôi xin cam đoan khoá luận “Xây dựng ứng dụng quản lý thư viện ảnh tích hợp AI tạo video” là công trình nghiên cứu do bản thân tôi thực hiện. Các nội dung nghiên cứu, kết quả trong khoá luận là xác thực.

Các thông tin sử dụng trong khoá luận là có cơ sở và không có nội dung nào sao chép từ các tài liệu mà không ghi rõ trích dẫn tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày 1 tháng 5 năm 2025

Sinh viên

Nguyễn Thành Dũng

Tóm tắt

Khóa luận tập trung xây dựng ứng dụng di động mang tên Smart Gallery nhằm hỗ trợ người dùng quản lý, tổ chức và tìm kiếm hình ảnh thông qua ứng dụng công nghệ trí tuệ nhân tạo. Smart Gallery cung cấp giao diện người dùng thân thiện, cho phép người dùng dễ dàng tương tác với kho ảnh cá nhân. Người dùng có thể xem và quản lý ảnh theo nhiều cách khác nhau như album, thời gian, tags, hoặc địa điểm.

Một trong những điểm nổi bật của Smart Gallery là khả năng tự động phân loại và gắn nhãn ảnh theo các tiêu chí như địa điểm, hoạt động và sự kiện dựa trên nội dung hình ảnh. Hệ thống còn tích hợp công nghệ nhận diện khuôn mặt, cho phép người dùng quản lý và đặt tên cho các nhân vật xuất hiện trong ảnh, từ đó tạo điều kiện thuận lợi cho việc tìm kiếm và phân loại.

Tính năng tìm kiếm thông minh của Smart Gallery cho phép người dùng tìm kiếm ảnh bằng văn bản hoặc giọng nói với nhiều tùy chọn lọc khác nhau như thời gian, album, nhân vật, v.v. Đặc biệt, ứng dụng còn hỗ trợ tạo video slideshow từ bộ sưu tập ảnh với khả năng tùy chỉnh đa dạng như theme, nhạc nền, độ phân giải và thời lượng.

Ngoài ra, Smart Gallery còn cung cấp tính năng quản lý địa điểm cho ảnh và hiển thị vị trí các bức ảnh trên bản đồ, giúp người dùng dễ dàng theo dõi và có cái nhìn tổng quan về những nơi đã đến và địa điểm chụp ảnh.

Từ khóa: Quản lý ảnh thông minh, Gán nhãn ảnh tự động, Nhận diện khuôn mặt, Tạo kịch bản video với AI, Smart Gallery.

Mục lục

Lời cảm ơn	
Lời cam đoan	i
Tóm tắt	ii
Mục lục	iii
Danh sách hình vẽ	vi
Danh sách bảng	viii
Chương 1 ĐẶT VẤN ĐỀ	1
1.1 Giới thiệu bài toán	1
1.2 Hiện trạng thị trường	1
1.2.1 Thị trường trong nước và quốc tế	2
1.2.2 Điểm mạnh so với các ứng dụng trên thị trường	2
1.3 Mục tiêu và đóng góp của khoá luận	3
1.4 Bố cục trình bày	3
Chương 2 KIẾN THỨC CƠ SỞ	5
2.1 Các công nghệ được sử dụng	5
2.1.1 Công nghệ phía người dùng (Front-end)	5
2.1.2 Công nghệ phía máy chủ (Back-end)	6
2.1.3 Cơ sở dữ liệu	6
2.1.4 Công nghệ khác	6
2.2 Các mô hình, thư viện AI được sử dụng và ứng dụng trong hệ thống	7
2.2.1 OpenCLIP	8
2.2.2 Face Recognition	10
Chương 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13

3.1	Đặc tả yêu cầu	13
3.1.1	Yêu cầu chức năng	13
3.1.2	Yêu cầu phi chức năng	14
3.2	Biểu đồ tổng quan ca sử dụng hệ thống	15
3.3	Mô tả ca sử dụng	16
3.3.1	Ca sử dụng điền thông tin cơ bản	17
3.3.2	Ca sử dụng tải ảnh lên	19
3.3.3	Ca sử dụng tạo album	21
3.3.4	Ca sử dụng xem danh sách video recap	24
3.3.5	Ca sử dụng tạo video recap	26
3.3.6	Ca sử dụng xem danh sách khuôn mặt	28
3.3.7	Ca sử dụng xem ảnh theo địa điểm	31
3.3.8	Ca sử dụng thêm vị trí cho ảnh	34
3.3.9	Ca sử dụng tìm kiếm hình ảnh	36
3.4	Thiết kế cơ sở dữ liệu	39
3.4.1	Các bảng dữ liệu chính	39
3.4.2	Tích hợp với các bảng hệ thống của Supabase	41
3.4.3	Triggers và Functions tự động	42

Chương 4 TRIỂN KHAI VÀ KIỂM THỬ ỨNG DỤNG QUẢN LÝ THU VIỆN ẢNH TÍCH HỢP AI TẠO VIDEO

4.1	Triển khai hệ thống	43
4.1.1	Triển khai quy trình tạo video	43
4.1.2	Kiến trúc hệ thống	49
4.2	Các chức năng chính của hệ thống	51
4.2.1	Xác thực người dùng	52
4.2.2	Quản lý ảnh	53
4.2.3	Quản lý album ảnh	54
4.2.4	Giao diện khám phá	55
4.2.5	Quản lý video recap	58
4.2.6	Tìm kiếm ảnh	60
4.3	Kiểm thử cho hệ thống	61
4.3.1	Kiểm thử các xử lý logic	61
4.3.2	Kiểm thử tương tác người dùng trên giao diện ứng dụng	65

Kết luận 67

Tài liệu tham khảo 68

Danh sách hình vẽ

3.1 Biểu đồ ca sử hệ thống.	16
3.2 Biểu đồ tuần tự ca sử dụng điền thông tin cơ bản.	19
3.3 Biểu đồ tuần tự ca sử dụng tải ảnh lên.	21
3.4 Biểu đồ tuần tự ca sử dụng tạo album.	23
3.5 Biểu đồ tuần tự ca sử xem danh sách video recap.	25
3.6 Biểu đồ tuần tự ca sử dụng tạo video recap.	28
3.7 Biểu đồ tuần tự ca sử dụng xem danh sách khuôn mặt.	31
3.8 Biểu đồ tuần tự ca sử dụng xem ảnh theo địa điểm.	33
3.9 Biểu đồ tuần tự ca sử dụng thêm vị trí cho ảnh.	36
3.10 Biểu đồ tuần tự ca sử dụng tìm kiếm hình ảnh.	39
3.11 Cấu trúc cơ sở dữ liệu.	41
4.1 Các thiết kế cho phần Intro của video slideshow.	44
4.2 Prompt Gemini để tạo caption cho video slideshow.	44
4.3 Các style tiêu đề chương.	45
4.4 Thiết kế các style slide ảnh.	45
4.5 Thiết kế phần Special Part của video slideshow.	46
4.6 Thiết kế phần Outro của video slideshow.	46
4.7 Biểu đồ kiến trúc hệ thống.	50
4.8 Giao diện điền thông tin tài khoản.	52
4.9 Giao diện thư viện ảnh.	53
4.10 Giao diện album ảnh.	54
4.11 Giao diện khám phá.	55
4.12 Giao diện quản lý vị trí ảnh.	56
4.13 Giao diện quản lý khuôn mặt.	57
4.14 Giao diện quản lý video recap.	58
4.15 Giao diện tạo video.	59
4.16 Giao diện tìm kiếm.	60

4.17 Độ phủ kiểm thử xử lý logic với JestJS.	62
4.18 Độ phủ kiểm thử xử lý logic với pytest.	62
4.19 Các API endpoint được kiểm thử với Postman.	63

Danh sách bảng

3.1	Mô tả chi tiết ca sử dụng điền thông tin cơ bản	17
3.2	Biểu đồ hoạt động và quan hệ ca sử dụng điền thông tin cơ bản	18
3.3	Mô tả chi tiết ca sử dụng tải ảnh lên	20
3.4	Biểu đồ hoạt động và quan hệ ca sử dụng tải ảnh lên	21
3.5	Mô tả chi tiết ca sử dụng tạo album	22
3.6	Biểu đồ hoạt động và quan hệ ca sử dụng tạo album	23
3.7	Mô tả chi tiết ca sử dụng xem danh sách video recap	24
3.8	Biểu đồ hoạt động và quan hệ ca sử dụng xem danh sách video recap	25
3.9	Mô tả chi tiết ca sử dụng xem ảnh theo địa điểm.	26
3.10	Biểu đồ hoạt động và quan hệ ca sử dụng tạo video recap	27
3.11	Mô tả chi tiết ca sử dụng xem danh sách khuôn mặt	29
3.12	Biểu đồ hoạt động và quan hệ ca sử dụng xem danh sách khuôn mặt	30
3.13	Mô tả chi tiết ca sử dụng xem ảnh theo địa điểm	32
3.14	Biểu đồ hoạt động và quan hệ ca sử dụng xem ảnh theo địa điểm . .	33
3.15	Mô tả chi tiết ca sử dụng thêm vị trí cho ảnh	34
3.16	Biểu đồ hoạt động và quan hệ ca sử dụng thêm vị trí cho ảnh	35
3.17	Mô tả chi tiết ca sử dụng tìm kiếm hình ảnh	37
3.18	Biểu đồ hoạt động và quan hệ ca sử dụng tìm kiếm hình ảnh	38
4.1	Các kịch bản kiểm thử API chính	64
4.2	Các kịch bản kiểm thử tương tác người dùng	65

Chương 1

ĐẶT VẤN ĐỀ

1.1 Giới thiệu bài toán

Tính đến năm 2023, Việt Nam có khoảng 77,93 triệu người dùng Internet, chiếm 79,1% dân số, trong khi đó có 70 triệu người dùng mạng xã hội và chiếm 71% dân số [1]. Cũng theo một khảo sát từ DataReportal vào năm 2024, Việt Nam là một trong những quốc gia có sự tăng trưởng mạnh mẽ trong việc sử dụng internet và các thiết bị số [2]. Việc sử dụng điện thoại thông minh để chụp ảnh trở thành thói quen hàng ngày của nhiều người dân Việt Nam, với khoảng 80% người dùng smartphone chụp ảnh ít nhất 23 tấm hình mỗi tuần [3].

Vì vậy, việc quản lý khôi lượng ảnh khổng lồ đang đặt ra nhiều thách thức cho người dùng:

- Khó khăn trong phân loại và tìm kiếm: Khảo sát của Adobe (2022) [4] chỉ ra rằng 74% người dùng cảm thấy "quá tải" vì không thể tổ chức ảnh hiệu quả, trong khi 62% mất hơn 10 phút để tìm một bức ảnh cũ.
- Tính năng tổng hợp ảnh còn hạn chế: các tính năng tạo "video kỉ niệm" của các ứng dụng quản lý ảnh hiện hành như Google Photo hay Apple Photo còn sơ sài, thiếu tính năng tùy biến và chỉ cung cấp video đơn giản với các hiệu ứng chuyển cảnh cơ bản [5].

Do đó một ứng dụng quản lý ảnh thông minh với khả năng phân loại ảnh tự động theo nhiều mục (người, địa điểm, tags, v.v.) và đặc biệt là tạo video kỉ niệm với nhiều tùy chọn cho người dùng lúc này là hết sức cần thiết.

1.2 Hiện trạng thị trường

Phần này đề cập đến việc phân tích hiện trạng thị trường và nêu bật tầm quan trọng của việc nghiên cứu công trình liên quan trong quá trình phát triển ứng dụng Smart Gallery.

1.2.1 Thị trường trong nước và quốc tế

Trên thị trường Việt Nam và quốc tế đã có không ít các ứng dụng giúp quản lý thư viện ảnh người dùng với các ưu điểm và hạn chế như sau:

- **Google Photos:** Là một trong những ứng dụng quản lý ảnh phổ biến nhất hiện nay. Google Photos cho phép người dùng lưu trữ, chia sẻ và chỉnh sửa ảnh trực tuyến. Tuy nhiên, ứng dụng này yêu cầu người dùng phải có tài khoản Google và có giới hạn dung lượng lưu trữ miễn phí. Ngoài ra tính năng tạo video kỷ niệm của Google Photos còn khá đơn giản và không cho phép người dùng tùy biến nhiều.
- **Apple Photo:** là ứng dụng cung cấp sự tích hợp liền mạch trong hệ sinh thái Apple, tự động đồng bộ hóa ảnh trên iPhone, iPad và Mac thông qua iCloud. Nó cung cấp khả năng tổ chức thành album và tạo video "kỷ niệm" dựa trên các bức ảnh của người dùng, cùng với một bộ công cụ chỉnh sửa ảnh. Với việc giới thiệu Apple Intelligence trên các thiết bị mới hơn, nó hiện tự hào có khả năng tìm kiếm nâng cao bằng cách sử dụng ngôn ngữ tự nhiên. Một nhược điểm lớn là khả năng sử dụng hạn chế bên ngoài hệ sinh thái Apple, khiến nó kém lý tưởng hơn cho người dùng có thiết bị không phải của Apple. Giao diện web chỉ cung cấp các chức năng cơ bản. Hay tính năng tạo video kỷ niệm của Apple Photo cũng không cho phép người dùng tùy biến nhiều.

1.2.2 Điểm mạnh so với các ứng dụng trên thị trường

Đối với các ứng dụng quản lý ảnh hiện có trên thị trường, mặc dù cung cấp các tính năng cơ bản về lưu trữ và phân loại ảnh, nhưng vẫn còn nhiều hạn chế đáng kể. Google Photos bị giới hạn dung lượng lưu trữ miễn phí và thiếu tính năng tùy biến trong việc tạo video kỷ niệm. Apple Photos lại bị giới hạn trong hệ sinh thái riêng, gây khó khăn cho người dùng các thiết bị không phải Apple, đồng thời giao diện web chỉ cung cấp các chức năng cơ bản. Cả hai ứng dụng đều có tính năng tạo video kỷ niệm còn khá sơ sài, thiếu tính năng tùy biến và chỉ là video đơn giản với các hiệu ứng chuyển cảnh cơ bản.

Do đó, khóa luận này đề xuất Smart Gallery - một ứng dụng di động không chỉ kế thừa những ưu điểm mà còn khắc phục những hạn chế của các ứng dụng tiền

nhiệm. Smart Gallery nổi bật với khả năng tự động phân loại và gắn nhãn ảnh theo nhiều tiêu chí như địa điểm, hoạt động và sự kiện dựa trên nội dung hình ảnh. Hệ thống tích hợp công nghệ nhận diện khuôn mặt, cho phép người dùng quản lý và đặt tên cho các nhân vật xuất hiện trong ảnh, từ đó tạo điều kiện thuận lợi cho việc tìm kiếm và phân loại.

Điểm đặc biệt của Smart Gallery so với các ứng dụng hiện có là tính năng tìm kiếm thông minh, cho phép người dùng tìm kiếm ảnh bằng văn bản hoặc giọng nói với nhiều tùy chọn lọc khác nhau như thời gian, album, nhân vật. Ngoài ra, Smart Gallery còn cung cấp khả năng tạo video slideshow từ bộ sưu tập ảnh với khả năng tùy chỉnh đa dạng như theme, nhạc nền, độ phân giải video và thời lượng.

Ngoài ra, Smart Gallery còn cung cấp tính năng quản lý địa điểm cho ảnh và hiển thị vị trí các bức ảnh trên bản đồ, giúp người dùng dễ dàng theo dõi và có cái nhìn tổng quan về những nơi đã đến và địa điểm chụp ảnh.

1.3 Mục tiêu và đóng góp của khóa luận

Mục tiêu của khóa luận này là phát triển một ứng dụng di động quản lý thư viện ảnh thông minh với sự hỗ trợ của trí tuệ nhân tạo. Hệ thống không chỉ giúp người dùng lưu trữ hình ảnh mà còn tự động phân loại, gắn nhãn và tổ chức chúng một cách khoa học. Hệ thống này sẽ giải quyết các vấn đề trong việc quản lý khối lượng ảnh lớn trên thiết bị di động như khó tìm kiếm, phân loại thủ công tốn thời gian, và việc tạo nội dung chia sẻ từ bộ sưu tập còn phức tạp. Đóng góp của khóa luận bao gồm việc xây dựng hệ thống nhận diện và phân loại ảnh thông minh, tích hợp công nghệ nhận diện khuôn mặt nâng cao, phát triển công cụ tạo video slideshow với nhiều tùy chỉnh, và xây dựng bộ công cụ tìm kiếm ảnh bằng văn bản/giọng nói với khả năng hiểu ngữ cảnh. Smart Gallery không chỉ nâng cao trải nghiệm người dùng trong việc quản lý ảnh cá nhân mà còn mở ra hướng phát triển cho các ứng dụng quản lý dữ liệu đa phương tiện dựa trên AI trong tương lai.

1.4 Bố cục trình bày

Phần còn lại của khóa luận được trình bày theo bố cục như sau:

Chương 2 trình bày những kiến thức nền tảng và công nghệ liên quan đến việc

phát triển ứng dụng Smart Gallery. Chương này giới thiệu về các công nghệ được sử dụng trong phát triển ứng dụng, bao gồm công nghệ phía người dùng, phía máy chủ, cơ sở dữ liệu và một số công nghệ bổ sung khác. Đặc biệt, phần sau của chương tập trung vào các mô hình và thư viện AI đóng vai trò quan trọng trong ứng dụng như OpenCLIP và Face Recognition, trong đó cơ sở lý thuyết và cách thức ứng dụng của các công nghệ này được phân tích chi tiết.

Chương 3 trình bày phương pháp tiếp cận và thiết kế giải pháp cho ứng dụng Smart Gallery. Chương này mô tả chi tiết về các đặc tả yêu cầu của hệ thống, các mô tả ca sử dụng và việc thiết kế cơ sở dữ liệu cho ứng dụng. Các ca sử dụng sẽ được mô tả chi tiết kèm theo mô tả chi tiết, các biểu đồ luồng hoạt động và các biểu đồ tuần tự.

Chương 4 tập trung vào thực nghiệm và đánh giá hệ thống. Chương này trình bày về quy trình triển khai kiến trúc hệ thống, cách hệ thống tạo kịch bản cho video, các chức năng chính và giao diện của hệ thống, cũng như các quy trình kiểm thử cho ứng dụng.

Cuối cùng, phần kết luận tóm tắt những kết quả đạt được của khóa luận, đánh giá những đóng góp của ứng dụng Smart Gallery trong việc giải quyết các vấn đề quản lý ảnh của người dùng. Những hạn chế hiện tại và hướng phát triển trong tương lai cũng được đề xuất để tiếp tục cải thiện hệ thống.

Chương 2

KIẾN THỨC CƠ SỞ

Chương này giới thiệu các kiến thức nền tảng và công nghệ chính trong việc phát triển Smart Gallery, bao gồm công nghệ Front-end, Back-end và cơ sở dữ liệu. Trọng tâm chương là các mô hình AI như open-clip và face-recognition, cùng cơ sở lý thuyết của chúng. Các công nghệ này là nền tảng cho các tính năng thông minh của ứng dụng: phân loại ảnh tự động, nhận diện khuôn mặt, tìm kiếm bằng văn bản/giọng nói và tạo video slideshow từ bộ sưu tập ảnh.

2.1 Các công nghệ được sử dụng

Phần này trình bày chi tiết về hệ sinh thái công nghệ được lựa chọn để phát triển ứng dụng Smart Gallery. Các công nghệ được phân tích theo từng thành phần kiến trúc, bao gồm tầng giao diện người dùng (Frontend), tầng xử lý nghiệp vụ (Backend), tầng cơ sở dữ liệu (Database), và các công nghệ hỗ trợ khác. Đối với mỗi công nghệ, luận văn sẽ phân tích cơ sở lý luận cho việc lựa chọn, các đặc điểm kỹ thuật nổi bật, cũng như vai trò cụ thể trong kiến trúc tổng thể của hệ thống.

2.1.1 Công nghệ phía người dùng (Front-end)

Smart Gallery tập trung sử dụng Flutter trong việc phát triển ứng dụng di động ở nền tảng Android. Cũng như sử dụng Supabase Flutter để kết nối, thao tác với cơ sở dữ liệu Supabase và thực hiện xác thực người dùng.

- **Flutter [6]:** Framework UI nguồn mở được phát triển bởi Google, cho phép xây dựng ứng dụng di động đa nền tảng với hiệu năng cao. Flutter sử dụng ngôn ngữ lập trình Dart và hệ thống widget phong phú, giúp tối ưu quá trình phát triển và đảm bảo giao diện người dùng nhất quán.
- **Supabase Flutter [7]:** Thư viện SDK chính thức kết nối Flutter với nền tảng Supabase, cung cấp các API đơn giản để thực hiện các thao tác với cơ sở dữ liệu, xác thực người dùng, lưu trữ tệp và xử lý real-time subscriptions.

2.1.2 Công nghệ phía máy chủ (Back-end)

Ở phía máy chủ, Smart Gallery sử dụng fastAPI và ExpressJS để phát triển các API phục vụ cho ứng dụng. FastAPI được sử dụng để xây dựng các API phục vụ việc phân loại label, định dạng khuôn mặt trong ảnh cũng như nhóm các ảnh có khuôn mặt tương tự nhau. ExpressJS được sử dụng để xây dựng API phục vụ việc tạo schema cho video cũng như tác vụ tạo video slideshow từ các ảnh đã được phân loại.

- **FastAPI** [8]: Là một framework Python hiện đại, nhanh chóng và dễ sử dụng để xây dựng các API RESTful. FastAPI hỗ trợ tự động tạo tài liệu API và có khả năng xử lý đồng thời tốt, giúp tối ưu hóa hiệu suất cho các ứng dụng web.
- **ExpressJS** [9]: Là một framework web phổ biến cho Node.js, giúp xây dựng các ứng dụng web và API một cách nhanh chóng và dễ dàng. ExpressJS cung cấp nhiều tính năng mạnh mẽ như middleware, routing và hỗ trợ cho việc xử lý các yêu cầu HTTP.

2.1.3 Cơ sở dữ liệu

Hệ thống sử dụng cơ sở dữ liệu PostgreSQL để lưu trữ dữ liệu và được host trên Supabase [10] - một nền tảng hosting và quản lý cơ sở dữ liệu.

- **PostgreSQL** [11]: Là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, mạnh mẽ và linh hoạt. PostgreSQL hỗ trợ nhiều tính năng tiên tiến như giao dịch ACID, các kiểu dữ liệu phức tạp và khả năng mở rộng cao.

2.1.4 Công nghệ khác

Ngoài các công nghệ được sử dụng để xây dựng giao diện người dùng và máy chủ, Smart Gallery còn sử dụng một số công nghệ khác để hỗ trợ cho việc phát triển ứng dụng như Redis, Remotion và HTTP Live Streaming (HLS).

- **Redis** [12]: Là một hệ thống lưu trữ dữ liệu trong bộ nhớ (in-memory data store) mã nguồn mở, hỗ trợ nhiều cấu trúc dữ liệu như chuỗi, danh sách, tập

hợp và bản đồ. Redis thường được sử dụng để tăng tốc độ truy xuất dữ liệu và giảm tải cho cơ sở dữ liệu chính.

- **Ứng dụng:** Smart Gallery sử dụng Redis như một queue để lưu trữ trạng thái render video của từng người dùng, ngăn chặn việc tạo video trùng lặp cho cùng một người dùng cũng như giới hạn mỗi người dùng chỉ được tạo một video trong một lúc. Ngoài ra Redis cũng được sử dụng như 1 hàng chờ (queue) để lưu các yêu cầu phân loại ảnh từ người dùng.
- **Remotion [13]:** Là một thư viện JavaScript cho phép tạo video từ mã nguồn. Remotion cho phép lập trình viên sử dụng React để xây dựng các thành phần video, giúp tạo ra các video động và tương tác dễ dàng hơn.
 - **Ứng dụng:** Smart Gallery sử dụng Remotion để tạo video slideshow từ các bức ảnh được chọn bởi người dùng và sau đó render thành định dạng mp4.
- **HTTP Live Streaming (HLS) [14]:** Là một giao thức truyền tải video trực tuyến được phát triển bởi Apple. HLS cho phép truyền tải video qua HTTP, chia video thành các đoạn nhỏ và phát lại chúng theo thời gian thực. HLS hỗ trợ nhiều định dạng video và có khả năng tự động điều chỉnh chất lượng video dựa trên băng thông mạng.
 - **Ứng dụng:** Smart Gallery sử dụng ffmpeg [15] để chuyển đổi video mp4 thành định dạng HLS với các chunk size nhỏ hơn. Sau đó tải các chunk này lên storage của Supabase để video có thể được phát lại trên các thiết bị khác nhau mà không cần tải xuống toàn bộ video, cũng như khắc phục hạn chế của Supabase Storage là không hỗ trợ tải lên những video lớn hơn 150Mb.

2.2 Các mô hình, thư viện AI được sử dụng và ứng dụng trong hệ thống

Phần này trình bày chi tiết về các mô hình trí tuệ nhân tạo và thư viện AI được tích hợp vào hệ thống Smart Gallery. Các mô hình này đóng vai trò nòng cốt trong việc cung cấp các khả năng nhận dạng, phân loại và tìm kiếm ảnh, khuôn mặt.

Luận văn sẽ phân tích về cơ sở lý thuyết của từng mô hình, kiến trúc và cách thức vận hành trong môi trường thực tế của ứng dụng.

2.2.1 OpenCLIP

2.2.1.1 Cơ sở lý thuyết

OpenCLIP [16] là một cài đặt mã nguồn mở của mô hình CLIP (Contrastive Language-Image Pre-training) được phát triển ban đầu bởi OpenAI. Mô hình này được thiết kế để tạo ra sự kết nối giữa hình ảnh và văn bản thông qua phương pháp học tương phản (contrastive learning).

Kiến trúc của OpenCLIP bao gồm hai thành phần chính:

- **Bộ mã hóa hình ảnh (Image Encoder):** Thường là một mạng nơ-ron tích chập (CNN) hoặc mô hình Vision Transformer (ViT) được sử dụng để chuyển đổi hình ảnh thành một vector đặc trưng trong không gian nhúng chung.
- **Bộ mã hóa văn bản (Text Encoder):** Sử dụng một mô hình ngôn ngữ, thường là Transformer, để chuyển đổi văn bản thành vector đặc trưng trong cùng không gian nhúng với hình ảnh.

Quá trình huấn luyện CLIP dựa trên nguyên lý học tương phản, trong đó mô hình được huấn luyện để tối đa hóa sự tương đồng giữa các cặp hình ảnh-văn bản phù hợp và tối thiểu hóa sự tương đồng giữa các cặp không phù hợp. Điều này được thực hiện thông qua hàm mất mát tương phản InfoNCE (Noise Contrastive Estimation).

Phương pháp này cho phép OpenCLIP học được các biểu diễn đa phương tiện mạnh mẽ mà không cần dữ liệu được gán nhãn thủ công. Thay vào đó, nó học từ một khối lượng lớn các cặp hình ảnh-văn bản thu thập từ internet, cho phép mô hình hiểu được các khái niệm trực quan và ngôn ngữ một cách toàn diện.

Điểm đặc biệt của OpenCLIP cũng như Clip là việc học zero-shot, cho phép nó nhận dạng các đối tượng và khái niệm mà nó chưa từng thấy trong quá trình huấn luyện. Điều này được thực hiện bằng cách so sánh hình ảnh đầu vào với các mô tả văn bản khác nhau và chọn mô tả có độ tương đồng cao nhất.

OpenCLIP mở rộng mô hình gốc của OpenAI bằng cách cung cấp các cài đặt mã

nguồn mở, hỗ trợ nhiều kiến trúc mô hình khác nhau, và cho phép huấn luyện trên các tập dữ liệu tùy chỉnh, giúp cộng đồng nghiên cứu tiếp cận và cải tiến mô hình này.

2.2.1.2 Ứng dụng

Trong ứng dụng Smart Gallery, OpenCLIP được triển khai để thực hiện hai chức năng chính: gán nhãn tự động cho hình ảnh và tìm kiếm hình ảnh dựa trên văn bản. Cụ thể hơn, Smart Gallery sử dụng một mô hình OpenCLIP được huấn luyện sẵn, kết hợp với các tập nhãn được định nghĩa trước để phân loại hình ảnh theo các nhãn có độ tương đồng cao nhất.

Gán nhãn tự động cho hình ảnh: Quá trình gán nhãn tự động được thực hiện theo các bước sau:

1. **Chuẩn bị tập nhãn:** Smart Gallery sử dụng ba tập nhãn chính: locations (địa điểm), activities (hoạt động), events (sự kiện), và một tập nhãn đặc biệt để phân loại ảnh có liên quan/không liên quan (relate/unrelate). Các tập nhãn này được mã hóa (encode) bằng bộ mã hóa văn bản của OpenCLIP và được lưu dưới dạng file .pt để sử dụng lại, tránh việc phải thực hiện mã hóa lặp lại.
2. **Lọc ảnh không liên quan:** Khi một ảnh được tải lên, bước đầu tiên là sử dụng tập nhãn đặc biệt để loại bỏ các ảnh không liên quan như giấy tờ, hình ảnh hoạt hình hoặc ảnh chụp màn hình. Điều này giúp giảm thiểu việc gán nhãn không chính xác và tập trung vào các bức ảnh có giá trị cho người dùng.
3. **Tính toán độ tương đồng và gán nhãn:** Đối với các ảnh đã được lọc, hệ thống sẽ mã hóa ảnh thành vector đặc trưng (image features) và tính toán độ tương đồng với các vector đặc trưng của từng nhãn trong các tập nhãn. Quá trình này bao gồm việc chuẩn hóa các vector đặc trưng và sử dụng phép nhân ma trận để tính toán xác suất tương đồng. Sau đó, hệ thống chọn ra top 2 nhãn có độ tương đồng cao nhất cho mỗi tập nhãn locations, activities, và events.
4. **Lưu trữ kết quả:** Sau khi hoàn tất việc gán nhãn, cả vector đặc trưng của ảnh (image features) và các nhãn được gán sẽ được lưu vào cơ sở dữ liệu. Việc lưu trữ vector đặc trưng giúp tối ưu hóa quá trình tìm kiếm sau này.

Tìm kiếm hình ảnh dựa trên văn bản: Khi người dùng nhập một câu truy vấn tìm kiếm, hệ thống sẽ thực hiện các bước sau:

1. Mã hóa câu truy vấn thành vector đặc trưng văn bản (text features) sử dụng bộ mã hóa văn bản của OpenCLIP.
2. Tính toán độ tương đồng cosine giữa vector đặc trưng văn bản và các vector đặc trưng hình ảnh đã được lưu trữ trong cơ sở dữ liệu.
3. Trả về các kết quả có độ tương đồng vượt qua một ngưỡng nhất định, đảm bảo kết quả tìm kiếm có độ chính xác cao.

Cách tiếp cận này cho phép Smart Gallery cung cấp khả năng tìm kiếm hình ảnh thông minh, vượt ra ngoài việc chỉ tìm kiếm dựa trên metadata. Người dùng có thể tìm kiếm hình ảnh bằng các mô tả tự nhiên như "bữa tiệc sinh nhật", "chuyến đi biển" hoặc "buổi picnic ở công viên", và hệ thống sẽ trả về các hình ảnh phù hợp với mô tả đó, ngay cả khi các hình ảnh không chứa thông tin metadata tương ứng.

2.2.2 Face Recognition

2.2.2.1 Cơ sở lý thuyết

Face Recognition [17] là một thư viện Python mã nguồn mở được phát triển bởi Adam Geitgey, được xây dựng trên nền tảng của thư viện dlib [18]. Thư viện này cung cấp một giao diện đơn giản cho các thuật toán nhận diện khuôn mặt tiên tiến, cho phép phát hiện, căn chỉnh và nhận diện khuôn mặt trong hình ảnh.

Quy trình nhận diện khuôn mặt trong thư viện Face Recognition thường bao gồm ba giai đoạn chính:

- **Phát hiện khuôn mặt (Face Detection):** Giai đoạn này sử dụng các thuật toán như Histogram of Oriented Gradients (HOG) hoặc mạng nơ-ron tích chập (CNN) để xác định vị trí của khuôn mặt trong hình ảnh. Phương pháp HOG nhanh nhưng độ chính xác thấp hơn, trong khi phương pháp CNN chính xác hơn nhưng yêu cầu nhiều tài nguyên tính toán hơn.

- **Phát hiện điểm đặc trưng (Facial Landmark Detection):** Sau khi phát hiện khuôn mặt, thuật toán xác định vị trí của 68 điểm đặc trưng trên khuôn mặt (như mắt, mũi, miệng, etc.). Các điểm này giúp căn chỉnh khuôn mặt và chuẩn bị cho quá trình mã hóa khuôn mặt.
- **Mã hóa khuôn mặt (Face Encoding):** Giai đoạn cuối cùng sử dụng một mạng nơ-ron sâu đã được huấn luyện trước để chuyển đổi khuôn mặt đã được căn chỉnh thành một vector đặc trưng 128 chiều. Các vector này còn được gọi là "face embedding" hoặc "face encoding", và chúng đại diện cho các đặc điểm của khuôn mặt trong không gian đa chiều.

Thư viện Face Recognition sử dụng mô hình ResNet-34 được huấn luyện trên tập dữ liệu FaceNet để tạo ra các vector mã hóa khuôn mặt. Mô hình này được huấn luyện bằng phương pháp học triplet, trong đó mô hình học cách tạo ra các embedding sao cho khoảng cách Euclidean giữa các embedding của cùng một người nhỏ, trong khi khoảng cách giữa các embedding của những người khác nhau lớn.

Để so sánh xem hai khuôn mặt có thuộc về cùng một người hay không, thư viện tính toán khoảng cách Euclidean giữa hai vector mã hóa khuôn mặt. Nếu khoảng cách này nhỏ hơn một ngưỡng nhất định (thường là 0.6), hai khuôn mặt được coi là của cùng một người.

Độ chính xác của thư viện Face Recognition đạt khoảng 99.38% trên tập dữ liệu Labeled Faces in the Wild, tương đương với các hệ thống thương mại tiên tiến nhất.

2.2.2.2 Ứng dụng

Trong Smart Gallery, thư viện Face Recognition được ứng dụng để phát hiện và phân loại khuôn mặt xuất hiện trong thư viện ảnh của người dùng, qua hai chức năng chính:

Nhận diện khuôn mặt trong ảnh: Quá trình nhận diện khuôn mặt được thực hiện theo các bước sau:

1. **Tiền xử lý ảnh:** Với những ảnh có kích thước lớn, hệ thống thực hiện giảm kích thước để tối ưu hóa hiệu suất xử lý mà vẫn đảm bảo độ chính xác trong việc nhận diện.

2. **Trích xuất đặc trưng khuôn mặt:** Sử dụng thư viện face_recognition để phát hiện vị trí các khuôn mặt trong ảnh và tính toán vector đặc trưng (face_embedding) cho mỗi khuôn mặt.
3. **Lưu trữ thông tin:** Hệ thống lưu thông tin về vị trí (location) của khuôn mặt trong ảnh cùng với vector đặc trưng tương ứng vào cơ sở dữ liệu để sử dụng cho việc phân nhóm và nhận diện sau này.

Phân nhóm khuôn mặt tương tự: Để người dùng có thể dễ dàng quản lý và phân loại khuôn mặt xuất hiện trong thư viện ảnh, Smart Gallery thực hiện phân nhóm các khuôn mặt tương tự như sau:

1. **Áp dụng thuật toán phân cụm:** Hệ thống sử dụng thuật toán DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [19] để phân cụm các vector đặc trưng khuôn mặt. DBSCAN được lựa chọn vì khả năng phát hiện các cụm với hình dạng tùy ý và không yêu cầu số lượng cụm trước.
2. **Tính toán vector đại diện:** Sau khi phân cụm, hệ thống tính toán centroid (vector trung bình) cho mỗi nhóm khuôn mặt để làm đại diện cho nhóm đó.
3. **Lưu trữ và đặt tên mặc định:** Các centroid được lưu vào cơ sở dữ liệu cùng với tên mặc định cho mỗi nhóm khuôn mặt.
4. **Cập nhật thông tin nhóm:** Khi có thêm khuôn mặt mới được phát hiện, hệ thống thực hiện phân cụm lại và tính toán centroid mới với tập các khuôn mặt mới và khuôn mặt đã được phát hiện. Sau đó, hệ thống tính độ tương đồng cosine [20] giữa các centroid mới và các centroid đã lưu trước đó. Nếu độ tương đồng vượt quá ngưỡng định trước, hệ thống giữ nguyên centroid cũ và cập nhật thông tin nhóm cho các khuôn mặt mới. Cách tiếp cận này đảm bảo sự nhất quán trong việc phân loại khuôn mặt theo thời gian và giúp các nhóm khuôn mặt được cập nhật liên tục khi có thêm dữ liệu mới.

Với cách tiếp cận này, Smart Gallery có thể tự động nhận diện và tổ chức các khuôn mặt xuất hiện trong thư viện ảnh của người dùng, cho phép người dùng dễ dàng xem, tìm kiếm và phân loại ảnh theo các cá nhân xuất hiện trong đó. Người dùng cũng có thể đặt tên cho các nhóm khuôn mặt, giúp cá nhân hóa trải nghiệm và tăng cường khả năng tìm kiếm bằng tên người.

Chương 3

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Chương này trình bày chi tiết về phân tích và thiết kế hệ thống Smart Gallery. Nó bao gồm các yêu cầu chức năng và phi chức năng, sơ đồ use case, thiết kế cơ sở dữ liệu và các thành phần chính của hệ thống.

3.1 Đặc tả yêu cầu

Phần này trình bày chi tiết về các yêu cầu chức năng và phi chức năng của ứng dụng Smart Gallery. Các yêu cầu này được phân loại thành các nhóm chính, mỗi yêu cầu sẽ được mô tả rõ ràng để đảm bảo rằng hệ thống đáp ứng đầy đủ nhu cầu của người dùng và các tiêu chuẩn kỹ thuật đã đề ra.

3.1.1 Yêu cầu chức năng

Để hệ thống hoạt động một cách hiệu quả và thông minh, mang lại trải nghiệm quản lý ảnh trực quan và tiện lợi cho người dùng, Smart Gallery yêu cầu những chức năng sau:

- **Đăng ký và xác thực người dùng:** Hệ thống cần cung cấp chức năng đăng ký và đăng nhập tài khoản người dùng, bao gồm bước xác thực các thông tin người dùng như email, ngày sinh, tên và tải lên ít nhất 3 ảnh vào thư viện người dùng.
- **Quản lý và tổ chức ảnh:** Chức năng cho phép người dùng tải lên, xem và tổ chức hình ảnh theo nhiều cách khác nhau như album, ngày tháng, và các thẻ gắn tự động.
- **Phân loại ảnh thông minh:** Hệ thống cần tự động phân tích và gắn thẻ cho ảnh theo địa điểm, hoạt động và sự kiện có trong ảnh bằng công nghệ AI, giúp người dùng dễ dàng tổ chức và tìm kiếm ảnh mà không cần gắn thẻ thủ công.

- **Nhận diện và quản lý khuôn mặt:** Chức năng tự động phát hiện khuôn mặt trong ảnh, nhóm các khuôn mặt tương tự và cho phép người dùng đặt tên cho từng nhóm. Cũng như cập nhật các khuôn mặt mới vào các nhóm đã có.
- **Tạo video slideshow:** Người dùng cần có khả năng tạo video slideshow từ bộ sưu tập ảnh với nhiều tùy chỉnh như theme, nhạc nền, độ phân giải, tiêu đề và thời lượng. Hệ thống cần hỗ trợ xuất video với nhiều độ phân giải khác nhau (720p, 1080p). Đồng thời cho phép quản lý và xem tiến trình, trạng thái của các video recap.
- **Tìm kiếm thông minh:** Hệ thống cần cung cấp chức năng tìm kiếm ảnh bằng văn bản và giọng nói với nhiều tùy chọn lọc (thời gian, album, nhân vật trong ảnh) và được hỗ trợ bởi AI để hiểu ngữ cảnh tìm kiếm của người dùng.
- **Quản lý địa điểm:** Chức năng cho phép người dùng xem, thêm và chỉnh sửa thông tin địa điểm cho ảnh, cũng như hiển thị vị trí ảnh trên bản đồ để có cái nhìn trực quan về những nơi đã chụp ảnh.

3.1.2 Yêu cầu phi chức năng

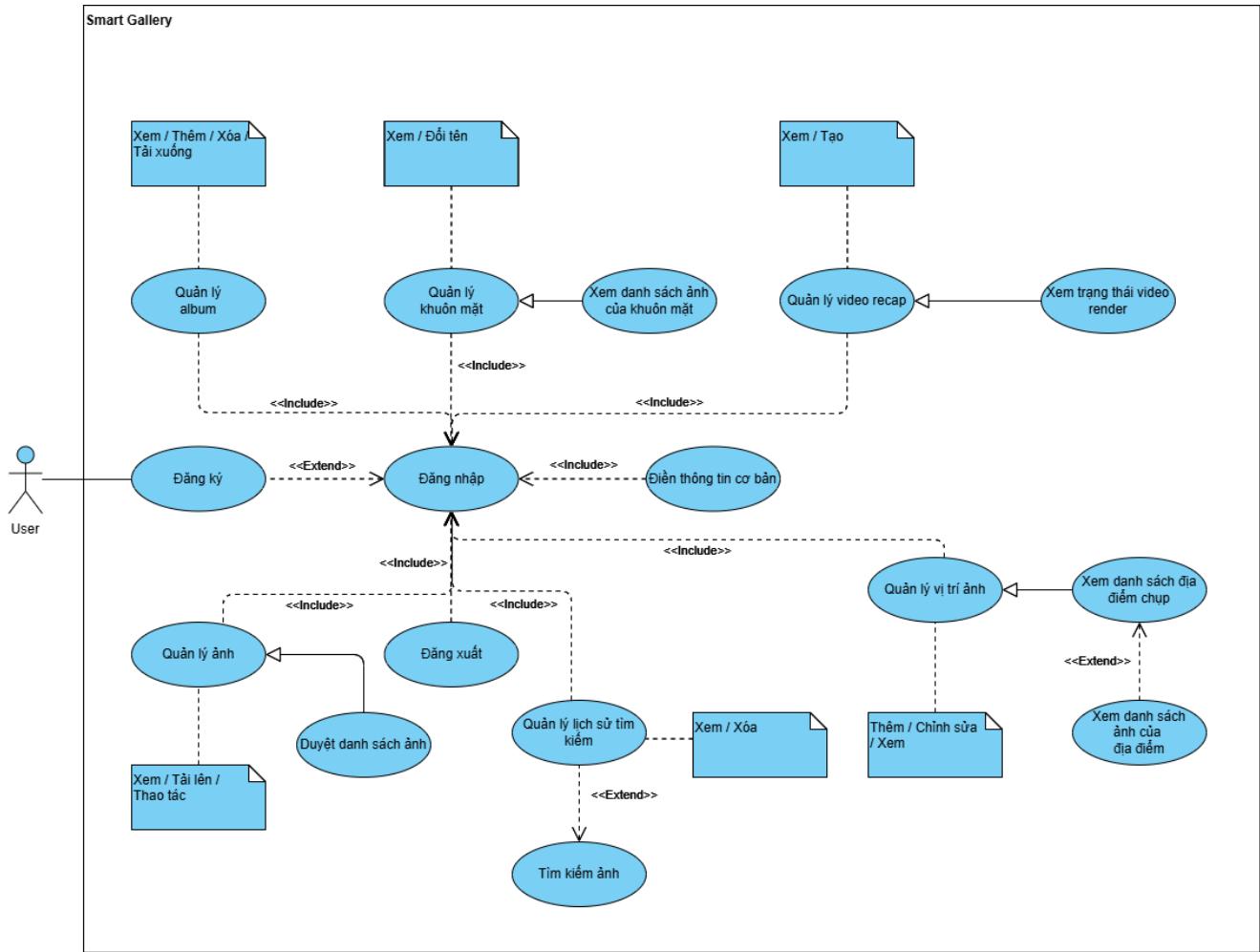
Ứng dụng Smart Gallery cần đáp ứng một số yêu cầu phi chức năng quan trọng để đảm bảo hiệu quả hoạt động và trải nghiệm người dùng tốt nhất:

- **Hiệu năng xử lý:** Hệ thống cần đảm bảo tốc độ phản hồi nhanh dù xử lý bộ sưu tập ảnh lớn. Thời gian tải và hiển thị ảnh cần được tối ưu, đặc biệt khi xem ảnh ở chế độ lướt hoặc album. Thời gian phân loại ảnh tự động không nên vượt quá 5 giây/ảnh trên thiết bị cấu hình trung bình.
- **Bảo mật và bảo vệ dữ liệu:** Hệ thống cần đảm bảo an toàn và bảo mật thông tin cá nhân, đặc biệt là dữ liệu hình ảnh của người dùng. Việc mã hóa dữ liệu và xác thực người dùng phải được thực hiện nghiêm ngặt để ngăn chặn truy cập trái phép.
- **Độ chính xác của AI:** Các thuật toán AI để phân loại ảnh và nhận diện khuôn mặt cần đạt độ chính xác cao. Tỷ lệ gắn nhãn chính xác cho ảnh cần đạt tối thiểu 85%, và nhận diện khuôn mặt cần đạt độ chính xác tối thiểu 90% để đảm bảo trải nghiệm người dùng.

- **Khả năng mở rộng:** Hệ thống cần có khả năng mở rộng để đáp ứng số lượng người dùng và dữ liệu ảnh tăng trưởng theo thời gian mà không làm giảm hiệu suất tổng thể.
- **Thân thiện với người dùng:** Giao diện của ứng dụng cần được thiết kế trực quan, dễ sử dụng và thẩm mỹ. Người dùng mới có thể dễ dàng tiếp cận và sử dụng đầy đủ tính năng mà không cần hướng dẫn phức tạp.
- **Sử dụng tài nguyên tối ưu:** Ứng dụng cần tối ưu hóa việc sử dụng tài nguyên thiết bị, bao gồm CPU, GPU và đặc biệt khi thực hiện các tác vụ xử lý ảnh và AI.
- **Cập nhật thời gian thực:** Hệ thống cần cung cấp khả năng cập nhật thời gian thực cho các quy trình xử lý dài, đặc biệt là trạng thái render video. Người dùng cần được thông báo về tiến độ xử lý video với độ trễ không quá 2 giây, bao gồm các trạng thái "đang xử lý", "đang render", "hoàn thành" hoặc "lỗi". Điều này giúp người dùng theo dõi được quá trình xử lý và đưa ra quyết định phù hợp mà không cần liên tục kiểm tra thủ công.

3.2 Biểu đồ tổng quan ca sử dụng hệ thống

Hình 3.1 biểu diễn tổng quan về các mối quan hệ giữa các ca sử dụng trong hệ thống. Tác nhân chính của hệ thống gồm có: Người dùng.



Hình 3.1: Biểu đồ ca sử dụng.

3.3 Mô tả ca sử dụng

Phần này trình bày chi tiết các ca sử dụng (use case) của hệ thống Smart Gallery, làm rõ tương tác giữa người dùng và hệ thống để đáp ứng các yêu cầu chức năng đã đặc tả. Mỗi ca sử dụng được mô tả theo cấu trúc thống nhất, bao gồm tác nhân, điều kiện tiên quyết, luồng xử lý chính, luồng xử lý thay thế và điều kiện sau khi thực hiện. Việc phân tích các ca sử dụng giúp đảm bảo tính đầy đủ của các tính năng và cung cấp cái nhìn trực quan về cách thức vận hành của ứng dụng từ góc độ người dùng.

3.3.1 Ca sử dụng điền thông tin cơ bản

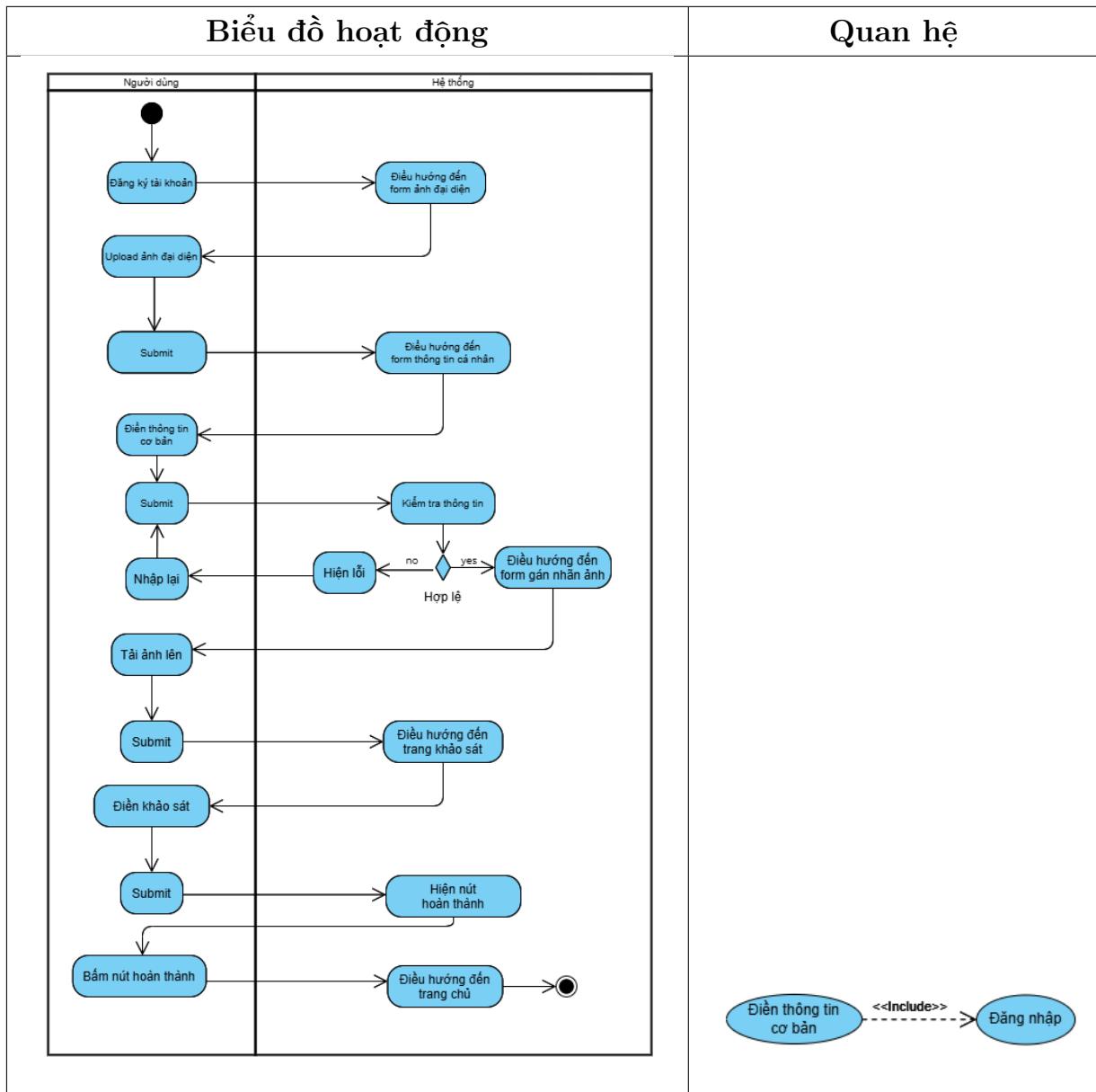
Người dùng sau khi đăng ký hoặc thiếu thông tin sẽ thực hiện việc điền thông tin cơ bản. Sau khi hoàn thành 4 form thông tin, người dùng mới có thể sử dụng các tính năng khác của hệ thống.

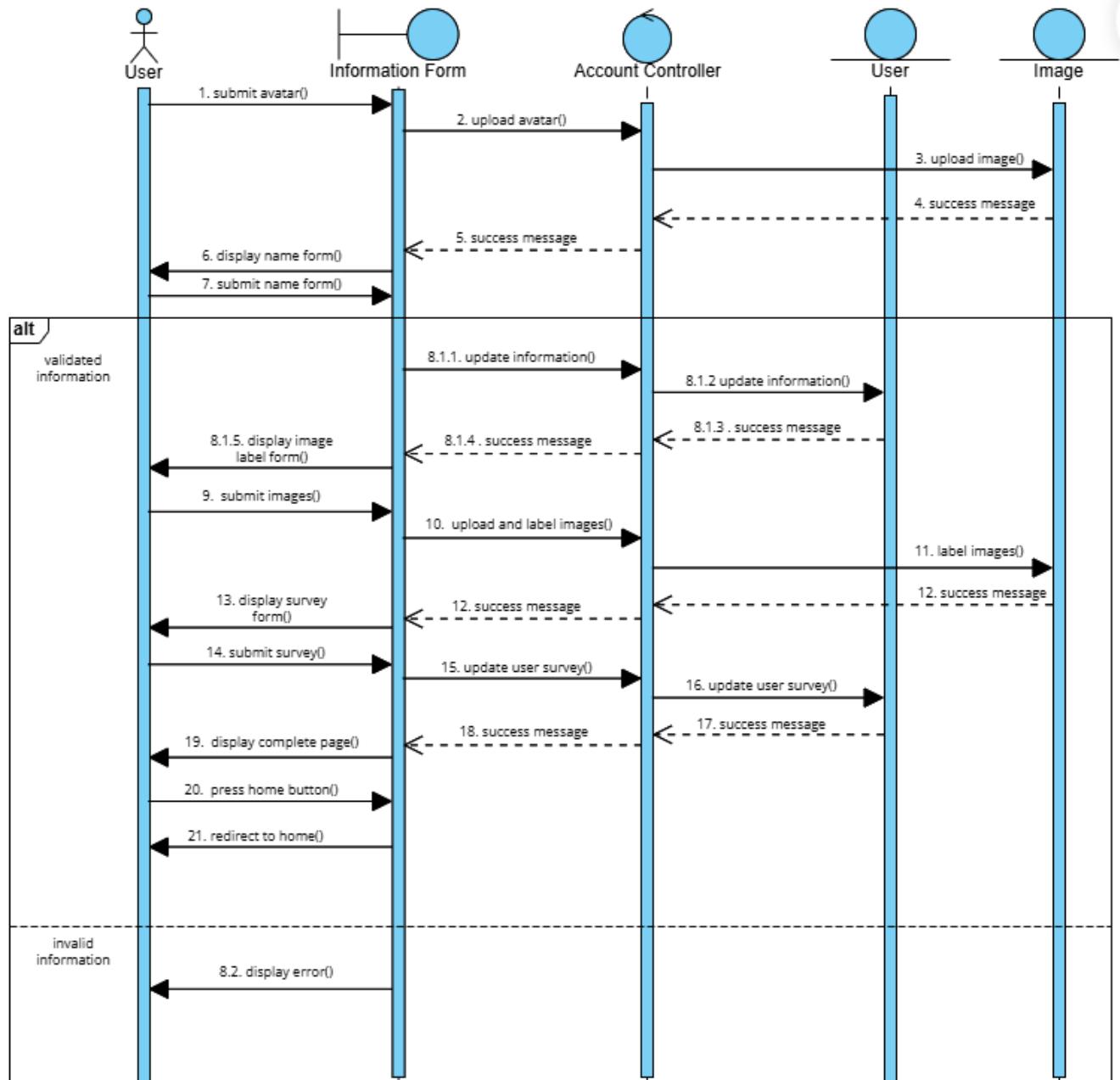
Mô tả chi tiết cho ca sử dụng điền thông tin cơ bản được thể hiện ở Bảng 3.1 dưới đây. Kèm theo là Bảng 3.2 về biểu đồ hoạt động, quan hệ và Hình 3.2 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.1: Mô tả chi tiết ca sử dụng điền thông tin cơ bản

Mô tả	Người dùng cập nhật thông tin cá nhân như ngày sinh, tên, khảo sát, ảnh đại diện và thực hiện 1 số thao tác làm quen với tính năng hệ thống.
Luồng cơ bản	<ol style="list-style-type: none">Người dùng đăng ký tài khoản mới.Người dùng tải ảnh đại diện lên.Người dùng điền tên và ngày sinh.Người dùng upload 3 ảnh lên để hệ thống hiển thị và giới thiệu tính năng gán nhãn ảnh.Người dùng điền form khảo sát.Người dùng bấm nút hoàn thành.Hệ thống điều hướng người dùng đến trang chủ của ứng dụng.
Luồng thay thế	- Nếu thông tin nhập vào không hợp lệ sẽ thông báo lỗi để người dùng nhập lại.
Tiền điều kiện	Người dùng đăng ký tài khoản thành công và chưa hoàn thành điền hết 4 form thông tin cá nhân.
Hậu điều kiện	- Thông tin cá nhân của người dùng được cập nhật. - Ảnh đại diện và 3 ảnh được upload lên sẽ được hệ thống đưa vào thư viện người dùng.
Yêu cầu phi chức năng	Hệ thống xử lý gán nhãn ảnh không quá 5s

Bảng 3.2: Biểu đồ hoạt động và quan hệ ca sử dụng điền thông tin cơ bản





Hình 3.2: Biểu đồ tuần tự ca sử dụng điền thông tin cơ bản.

3.3.2 Ca sử dụng tải ảnh lên

Người dùng có thể tải ảnh lên hệ thống và được tự động gán nhãn, phân loại cho ảnh. Kết quả gán nhãn sẽ được hiển thị trên màn hình sau khi người dùng tải ảnh lên thành công.

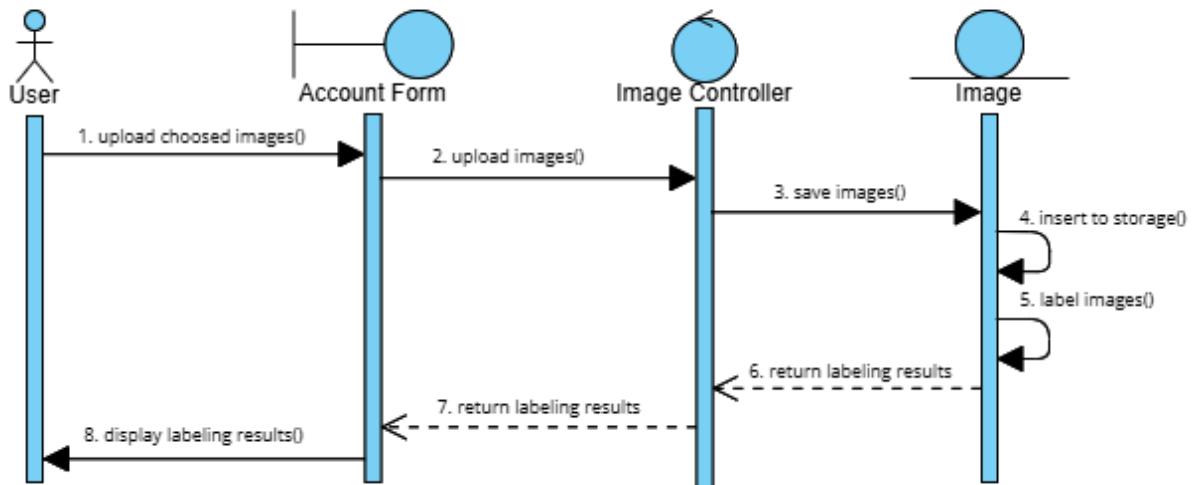
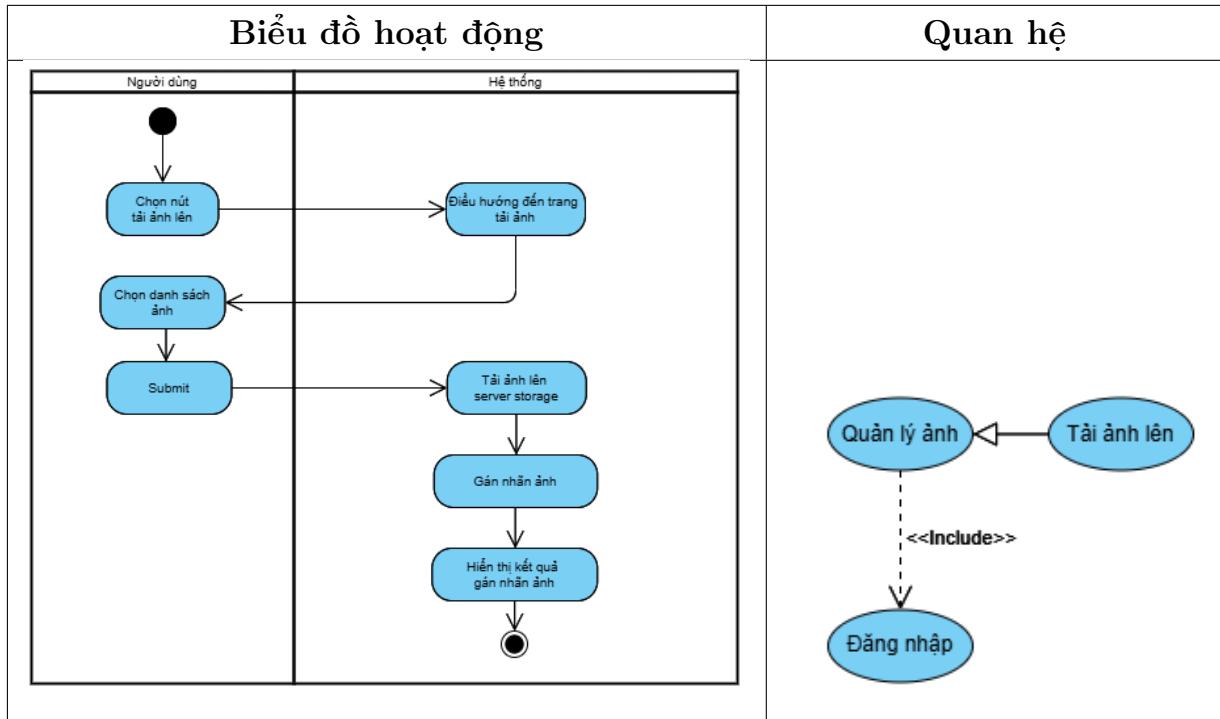
Mô tả chi tiết cho ca sử dụng tải ảnh lên được thể hiện ở Bảng 3.3 dưới đây. Kèm

theo là Bảng 3.4 về biểu đồ hoạt động, quan hệ và Hình 3.3 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.3: Mô tả chi tiết ca sử dụng tải ảnh lên

Mô tả	Người dùng tải ảnh lên để hệ thống tự động gán nhãn cho ảnh.
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng truy cập màn hình tải ảnh lên 2. Người dùng chọn ảnh muốn tải lên trong máy. 3. Người dùng submit danh sách ảnh vừa chọn. 4. Hệ thống tải ảnh lên server ảnh và gán nhãn. 5. Hệ thống hiển thị kết quả gán nhãn ảnh.
Luồng thay thế	<ul style="list-style-type: none"> - Người dùng không chọn ảnh và hệ thống không cho phép submit.
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đăng ký / đăng nhập tài khoản thành công và hoàn thành điền form thông tin cơ bản.
Hậu điều kiện	<ul style="list-style-type: none"> - Danh sách ảnh người dùng được tải lên và cập nhật vào thư viện người dùng.
Yêu cầu phi chức năng	Hệ thống xử lý gán nhãn trên 1 ảnh không quá 2s

Bảng 3.4: Biểu đồ hoạt động và quan hệ ca sử dụng tải ảnh lên



Hình 3.3: Biểu đồ tuần tự ca sử dụng tải ảnh lên.

3.3.3 Ca sử dụng tạo album

Người dùng sau khi tải ảnh lên hệ thống có thể tạo album để phân loại ảnh theo chủ đề. Hệ thống sẽ sắp xếp ảnh theo thứ tự thời gian tải lên và hiển thị dưới dạng

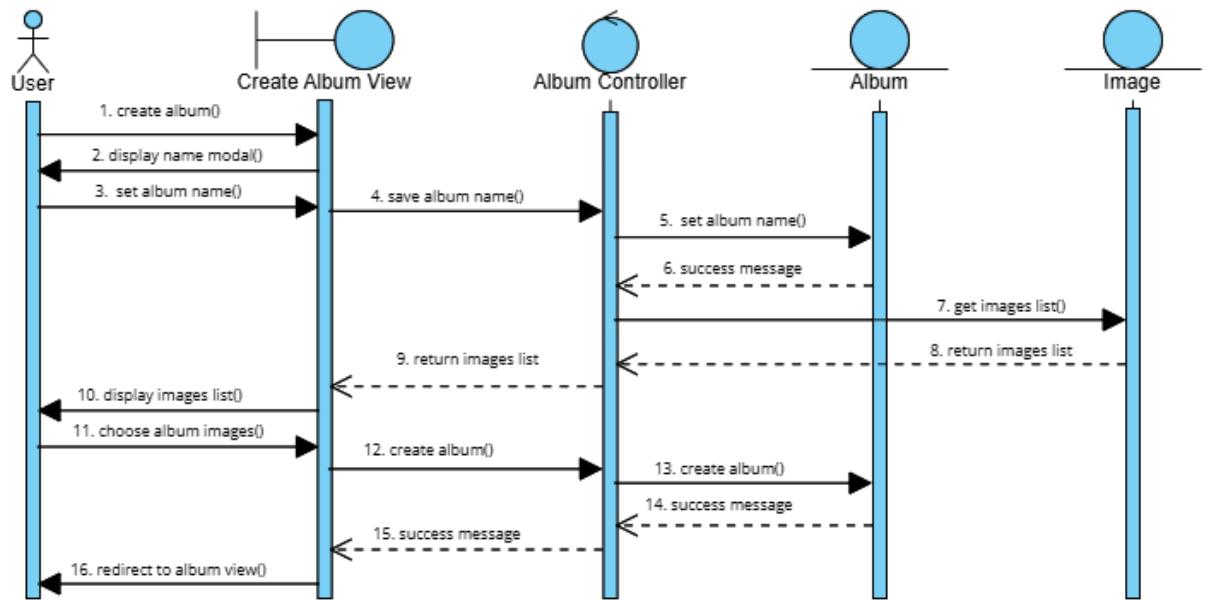
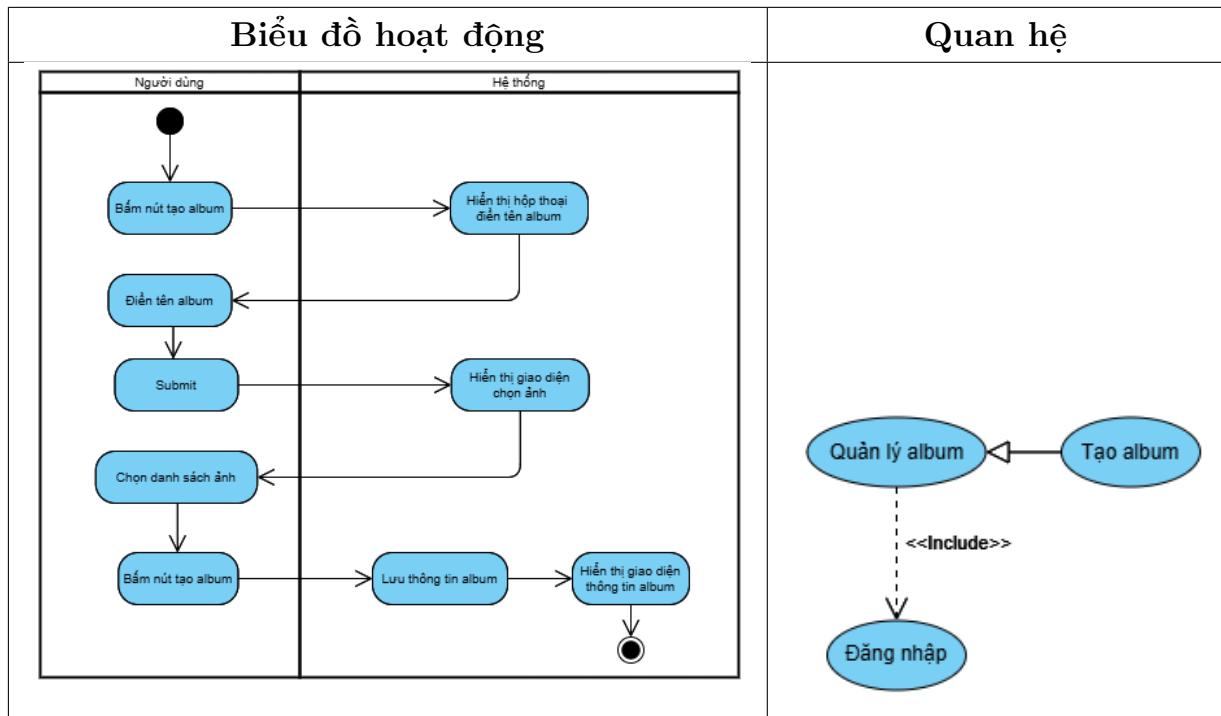
lưới cho người dùng.

Mô tả chi tiết cho ca sử dụng tạo album được thể hiện ở Bảng 3.5 dưới đây. Kèm theo là Bảng 3.6 về biểu đồ hoạt động, quan hệ và Hình 3.4 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.5: Mô tả chi tiết ca sử dụng tạo album

Mô tả	Người dùng tạo album mới trong thư viện ảnh.
Luồng cơ bản	<ol style="list-style-type: none">1. Người dùng bấm nút tạo album trong thanh công cụ ở dưới màn hình.2. Hệ thống hiển thị hộp thoại điền tên album.3. Người dùng nhập tên album và bấm nút tạo album.4. Hệ thống hiển thị hộp thoại chọn ảnh trong album từ danh sách những ảnh người dùng đã tải lên hệ thống.5. Người dùng chọn những ảnh muốn đưa vào album và bấm nút tạo album.6. Hệ thống tạo album và hiển thị giao diện danh sách ảnh trong album và tiêu đề.
Tiền điều kiện	<ul style="list-style-type: none">- Người dùng đã đăng nhập vào hệ thống.- Người dùng đã có ảnh trong thư viện.
Hậu điều kiện	<ul style="list-style-type: none">- Hệ thống cập nhật album mới vào danh sách các album đang có để hiển thị trong trang danh sách album.
Yêu cầu phi chức năng	Hệ thống xử lý tạo album không quá 2s.

Bảng 3.6: Biểu đồ hoạt động và quan hệ ca sử dụng tạo album



Hình 3.4: Biểu đồ tuần tự ca sử dụng tạo album.

3.3.4 Ca sử dụng xem danh sách video recap

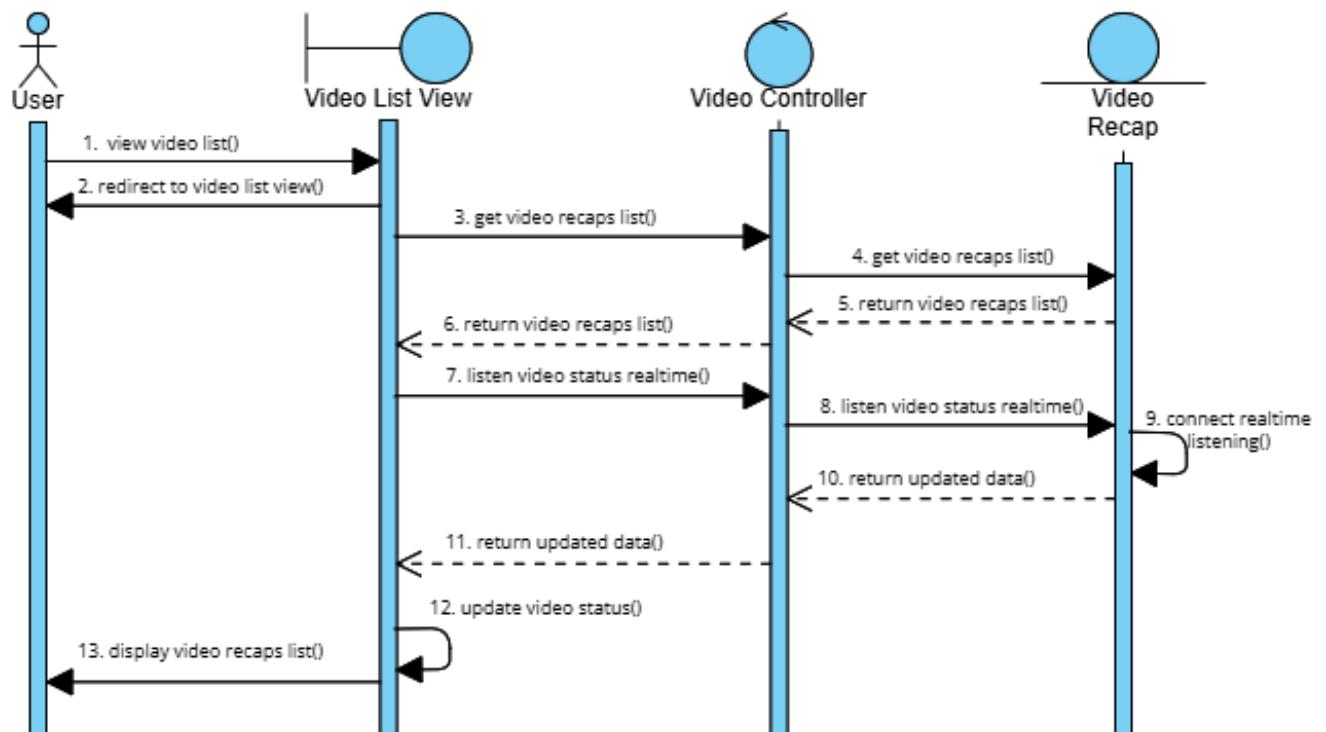
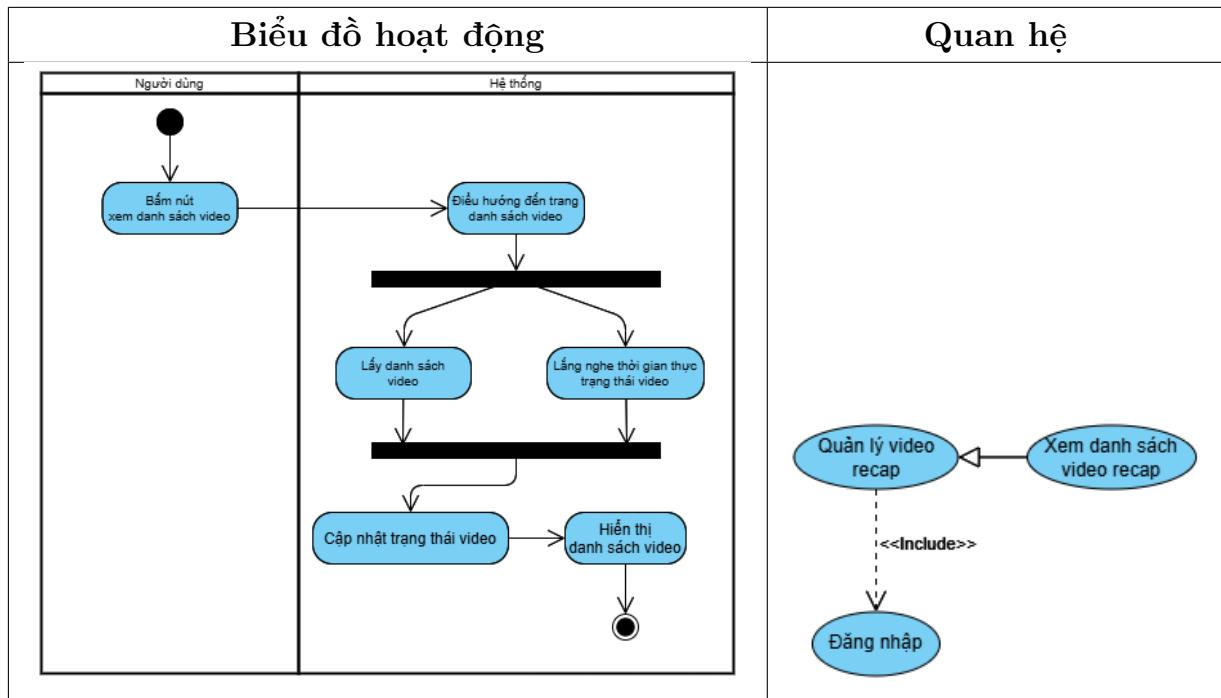
Người dùng sau khi gửi yêu cầu tạo video có thể theo dõi trạng thái video của mình trong danh sách video. Hệ thống sẽ cập nhật trạng thái và tiến độ của video theo thời gian thực.

Mô tả chi tiết cho ca sử dụng xem danh sách video recap được thể hiện ở Bảng 3.7 dưới đây. Kèm theo là Bảng 3.8 về biểu đồ hoạt động, quan hệ và Hình 3.5 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.7: Mô tả chi tiết ca sử dụng xem danh sách video recap

Mô tả	Người dùng xem danh sách và trạng thái của các video đã tạo.
Luồng cơ bản	<ol style="list-style-type: none">Người dùng bấm nút xem danh sách video ở thanh công cụ dưới màn hình.Hệ thống điều hướng đến trang danh sách video và hiển thị.Người dùng nhập tên album và bấm nút tạo album.
Tiền điều kiện	<ul style="list-style-type: none">- Người dùng đã đăng nhập vào hệ thống.- Người dùng đã tạo ít nhất 1 video recap.
Hậu điều kiện	<ul style="list-style-type: none">- Người dùng có thể xem chi tiết trạng thái render của từng video.- Người dùng có thể xem những video đã render xong.- Hệ thống cập nhật trạng thái và tiến độ của các video trong thời gian thực.
Yêu cầu phi chức năng	<ul style="list-style-type: none">- Hệ thống lấy danh sách video không quá 2s.- Hệ thống cập nhật trạng thái video thời gian thực, không quá 0.5s.

Bảng 3.8: Biểu đồ hoạt động và quan hệ ca sử dụng xem danh sách video recap



Hình 3.5: Biểu đồ tuần tự ca sử xem danh sách video recap.

3.3.5 Ca sử dụng tạo video recap

Người dùng có thể tạo video recap từ những ảnh đã tải lên hệ thống cũng như từ máy người dùng. Hệ thống sẽ tự động tạo kịch bản cho video và những tùy chỉnh mặc định cho video từ danh sách ảnh: style khung ảnh, nhạc nền, chất lượng video, thời lượng video, chủ đề video. Người dùng có thể tùy chỉnh lại các thông số này trước khi gửi yêu cầu tạo video.

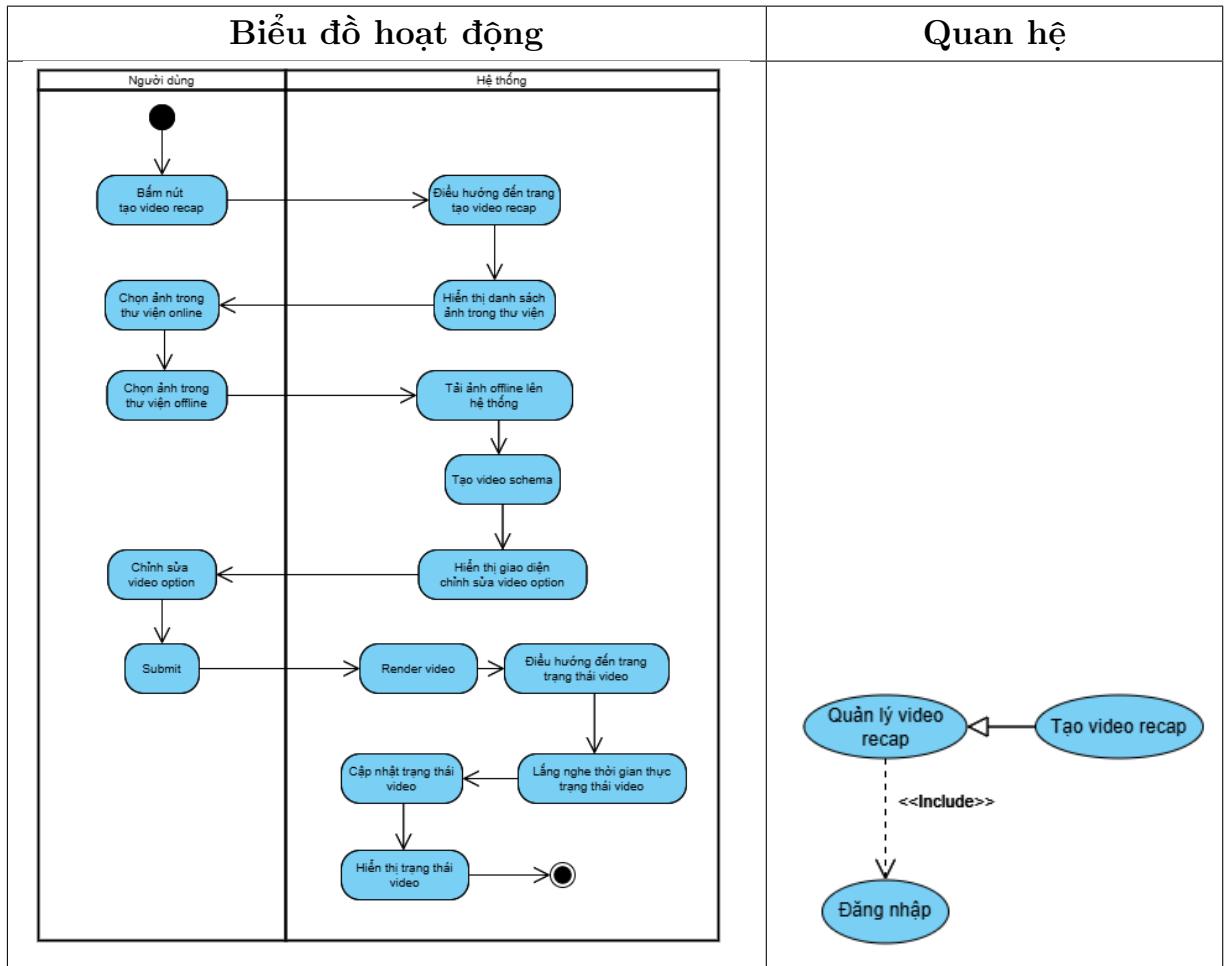
Mô tả chi tiết cho ca sử dụng tạo video recap được thể hiện ở Bảng ?? dưới đây. Kèm theo là Bảng 3.10 về biểu đồ hoạt động, quan hệ và Hình 3.6 về biểu đồ tuần tự của ca sử dụng này.

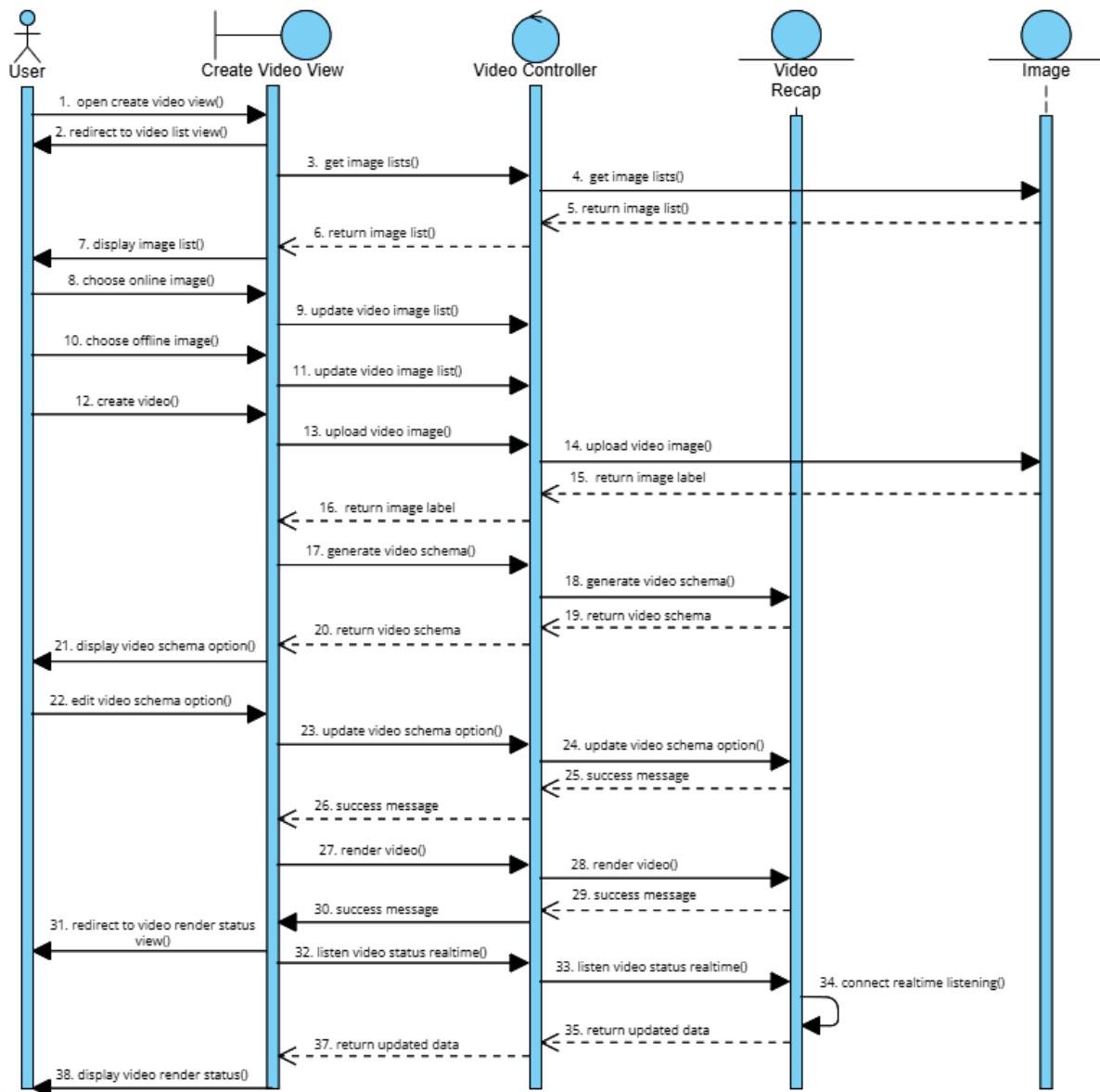
Bảng 3.9: Mô tả chi tiết ca sử dụng xem ảnh theo địa điểm.

Mô tả	Người dùng xem danh sách ảnh được chụp tại 1 địa điểm.
Luồng cơ bản	<ol style="list-style-type: none">Người dùng bấm vào nhóm địa điểm muốn xem danh sách ảnh chụp tại nơi đó.Hệ thống điều hướng đến trang danh sách ảnh của địa điểm.Hệ thống lấy thông tin danh sách ảnh chụp tại địa điểm và nhóm theo ngày.Hệ thống hiển thị hộp thoại cấp quyền thông tin vị trí hiện tại.Người dùng cấp quyền truy cập vị trí hiện tại.Hệ thống hiển thị danh sách ảnh chụp trên bản đồ.
Luồng thay thế	<ul style="list-style-type: none">- Người dùng không cấp quyền truy cập vị trí.- Hệ thống không lấy được dữ liệu ảnh của địa điểm.
Tiền điều kiện	<ul style="list-style-type: none">- Người dùng đã đăng nhập vào hệ thống.- Có ít nhất 1 bức ảnh đã được cập nhật vị trí chụp ảnh.
Hậu điều kiện	<ul style="list-style-type: none">- Người dùng có thể bấm vào ảnh trong danh sách để xem vị trí chi tiết của ảnh trên bản đồ.

Yêu cầu phi chức năng	<ul style="list-style-type: none"> - Hệ thống lấy dữ liệu ảnh của địa điểm không quá 1s. - Hệ thống lấy được dữ liệu hiện tại vị trí người dùng (nếu được cấp quyền).
------------------------------	---

Bảng 3.10: Biểu đồ hoạt động và quan hệ ca sử dụng tạo video recap





Hình 3.6: Biểu đồ tuần tự ca sử dụng tạo video recap.

3.3.6 Ca sử dụng xem danh sách khuôn mặt

Người dùng xem danh sách các khuôn mặt được hệ thống phát hiện trong ảnh. Khi nhận được yêu cầu từ người dùng, hệ thống sẽ lấy dữ liệu khuôn mặt từ ảnh và phân loại chúng thành các nhóm khuôn mặt. Nếu có những nhóm khuôn mặt đã được tạo trước đó thì hệ thống sẽ so sánh nhóm khuôn mặt mới với nhóm khuôn mặt cũ. Nếu phù hợp thì nhóm mới sẽ thay thế nhóm cũ.

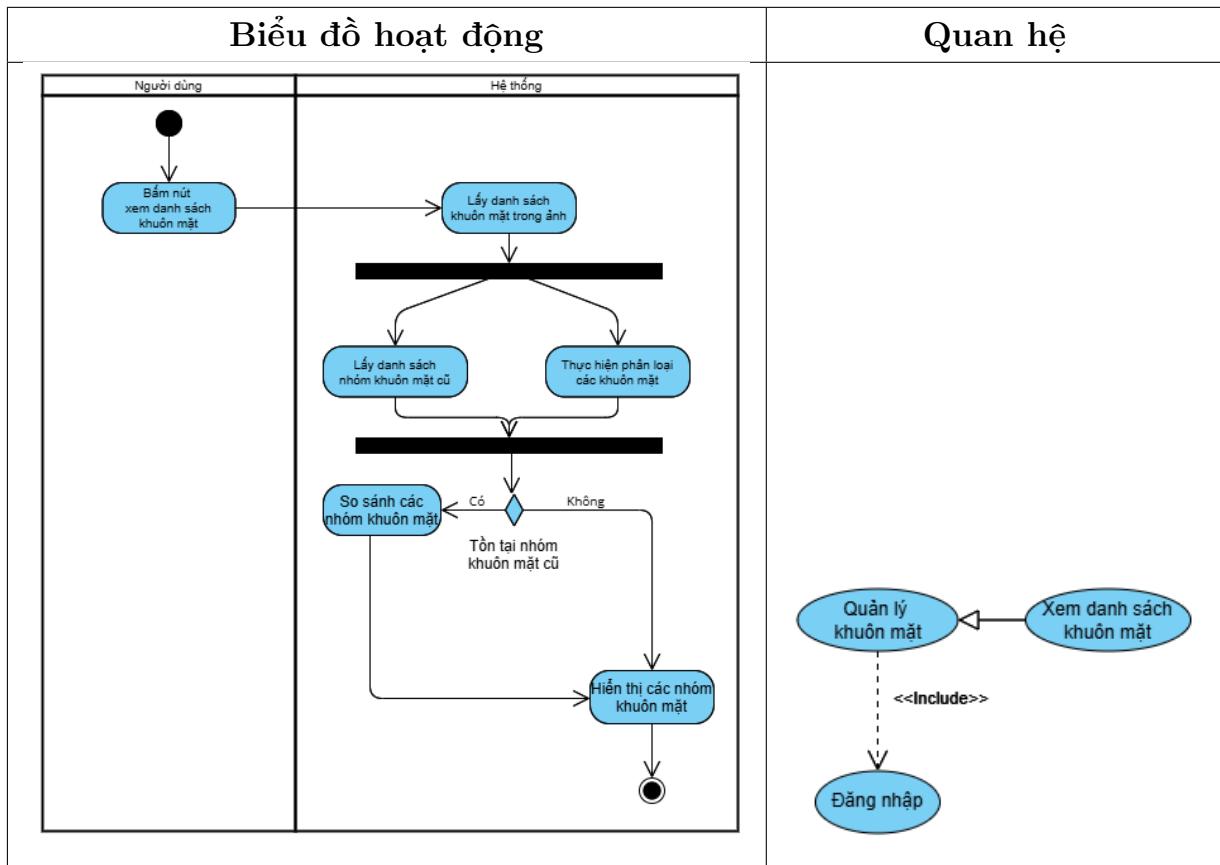
Mô tả chi tiết cho ca sử dụng xem danh sách khuôn mặt được thể hiện ở Bảng

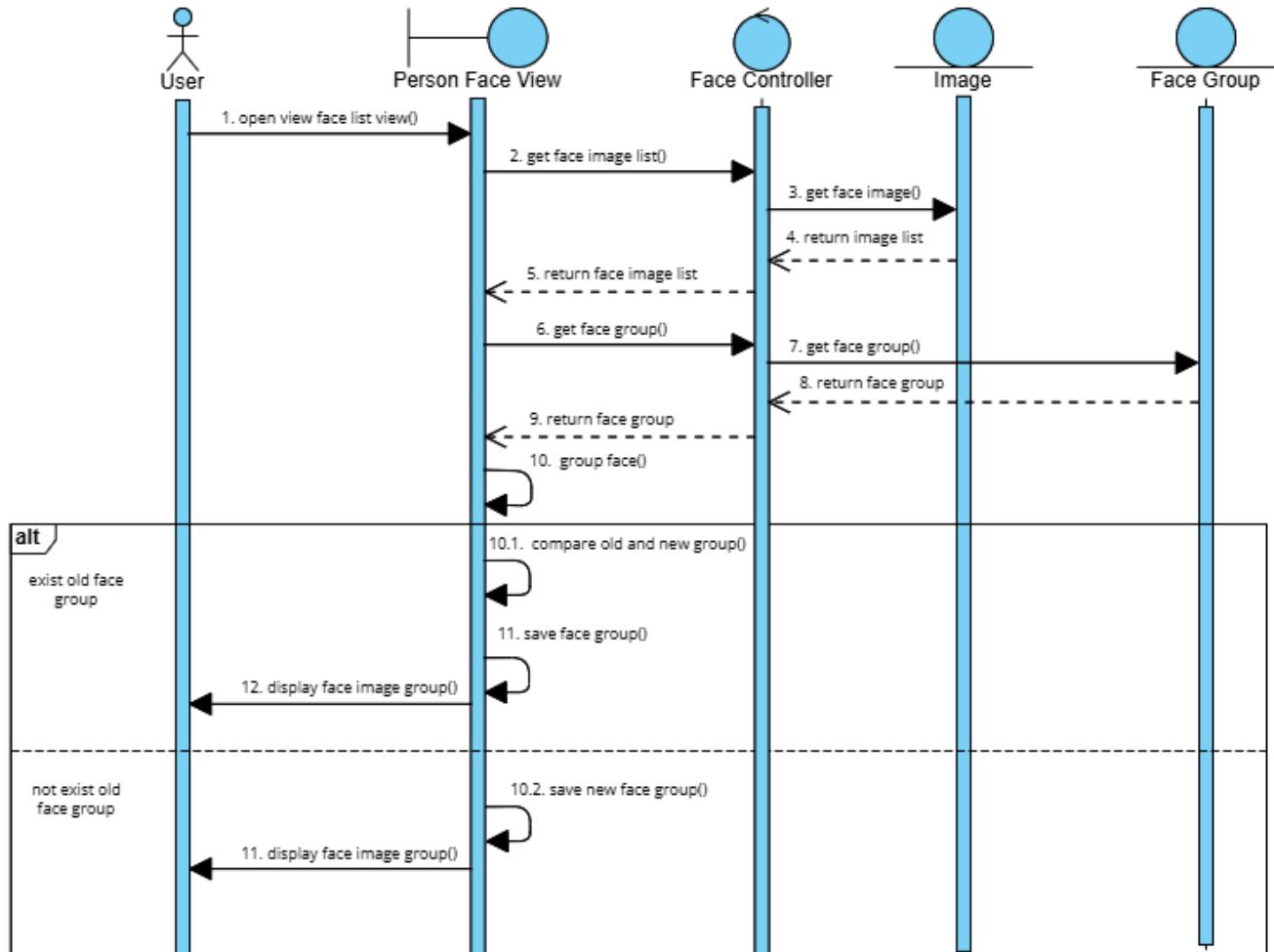
3.11 dưới đây. Kèm theo là Bảng 3.12 về biểu đồ hoạt động, quan hệ và Hình 3.7 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.11: Mô tả chi tiết ca sử dụng xem danh sách khuôn mặt

Mô tả	Người dùng muốn xem các danh sách khuôn mặt xuất hiện trong các bức ảnh tải lên.
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng bấm vào ô khám phá khuôn mặt. 2. Hệ thống lấy dữ liệu những khuôn mặt trong ảnh người dùng. 3. Hệ thống lấy dữ liệu nhóm khuôn mặt đã được tạo trước đó. 4. Hệ thống thực hiện phân loại và so sánh nhóm khuôn mặt mới với nhóm khuôn mặt cũ. 5. Hệ thống hiển thị danh sách các khuôn mặt và tên gọi (nếu có).
Luồng thay thế	<ol style="list-style-type: none"> 2a. Nếu có những nhóm khuôn mặt có sẵn thì hệ thống so sánh nhóm mới với nhóm cũ. Và sau đó nhóm mới sẽ thay thế nhóm cũ nếu phù hợp.
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống. - Hệ thống đã phân loại khuôn mặt trong ảnh.
Hậu điều kiện	<ul style="list-style-type: none"> - Hệ thống hiển thị trạng thái và tiến độ tạo video theo thời gian thực. - Người dùng quay về màn hình danh sách video để theo dõi trạng thái video đã tạo.
Yêu cầu phi chức năng	-Hệ thống xử lý nhóm khuôn mặt không quá 20s.

Bảng 3.12: Biểu đồ hoạt động và quan hệ ca sử dụng xem danh sách khuôn mặt





Hình 3.7: Biểu đồ tuần tự ca sử dụng xem danh sách khuôn mặt.

3.3.7 Ca sử dụng xem ảnh theo địa điểm

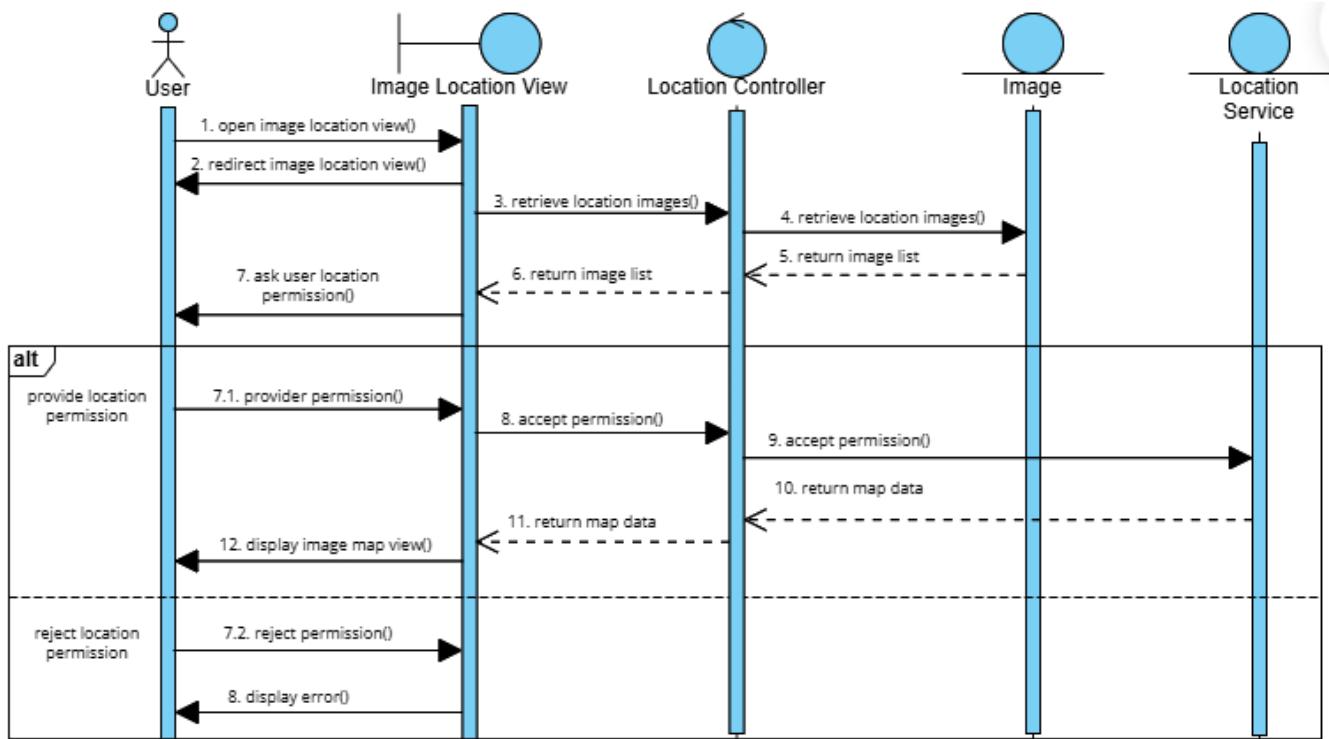
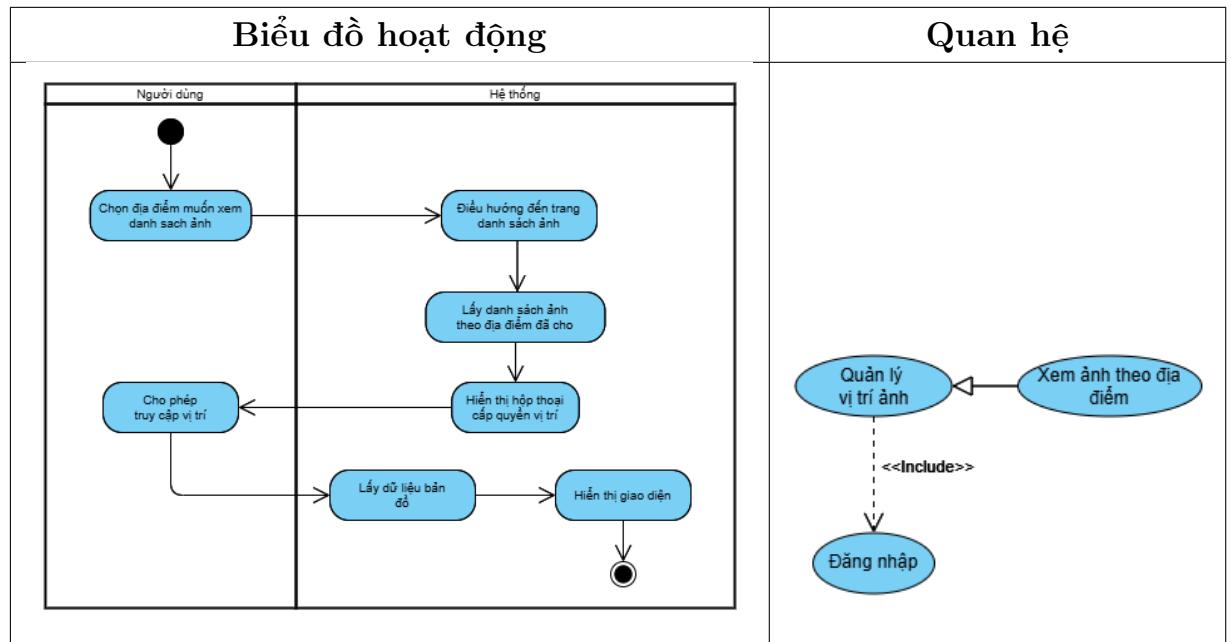
Sau khi cập nhật vị trí cho ảnh, người dùng có thể xem danh sách ảnh đã chụp tại một địa điểm cụ thể. Hệ thống sẽ tự động lấy vị trí hiện tại của người dùng (nếu được cấp quyền) và hiển thị danh sách ảnh trên bản đồ. Người dùng có thể bấm vào ảnh trong danh sách để xem vị trí chi tiết của ảnh trên bản đồ. Với yêu cầu là trước đó, hệ thống đã hoàn thành phân loại các nhóm địa điểm ảnh.

Mô tả chi tiết cho ca sử dụng xem ảnh theo địa điểm được thể hiện ở Bảng 3.13 dưới đây. Kèm theo là Bảng 3.14 về biểu đồ hoạt động, quan hệ và Hình 3.8 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.13: Mô tả chi tiết ca sử dụng xem ảnh theo địa điểm

Mô tả	Người dùng xem danh sách ảnh được chụp tại 1 địa điểm.
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng bấm vào nhóm địa điểm muốn xem danh sách ảnh chụp tại nơi đó. 2. Hệ thống điều hướng đến trang danh sách ảnh của địa điểm. 3. Hệ thống lấy thông tin danh sách ảnh chụp tại địa điểm và nhóm theo ngày. 4. Hệ thống hiển thị hộp thoại cấp quyền thông tin vị trí hiện tại. 5. Người dùng cấp quyền truy cập vị trí hiện tại. 6. Hệ thống hiển thị danh sách ảnh chụp trên bản đồ.
Luồng thay thế	<ul style="list-style-type: none"> - Người dùng không cấp quyền truy cập vị trí. - Hệ thống không lấy được dữ liệu ảnh của địa điểm.
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống. - Có ít nhất 1 bức ảnh đã được cập nhật vị trí chụp ảnh.
Hậu điều kiện	<ul style="list-style-type: none"> - Người dùng có thể bấm vào ảnh trong danh sách để xem vị trí chi tiết của ảnh trên bản đồ.
Yêu cầu phi chức năng	<ul style="list-style-type: none"> - Hệ thống lấy dữ liệu ảnh của địa điểm không quá 1s. - Hệ thống lấy được dữ liệu hiện tại vị trí người dùng (nếu được cấp quyền).

Bảng 3.14: Biểu đồ hoạt động và quan hệ ca sử dụng xem ảnh theo địa điểm



Hình 3.8: Biểu đồ tuần tự ca sử dụng xem ảnh theo địa điểm.

3.3.8 Ca sử dụng thêm vị trí cho ảnh

Người dùng có thể thêm thủ công vị trí cho ảnh đã tải lên trong thư viện của mình. Hệ thống sẽ tự động lấy vị trí hiện tại của người dùng (nếu được cấp quyền) và hiển thị trên bản đồ. Người dùng có thể chọn vị trí trên bản đồ hoặc tìm kiếm địa điểm để thêm vị trí cho ảnh.

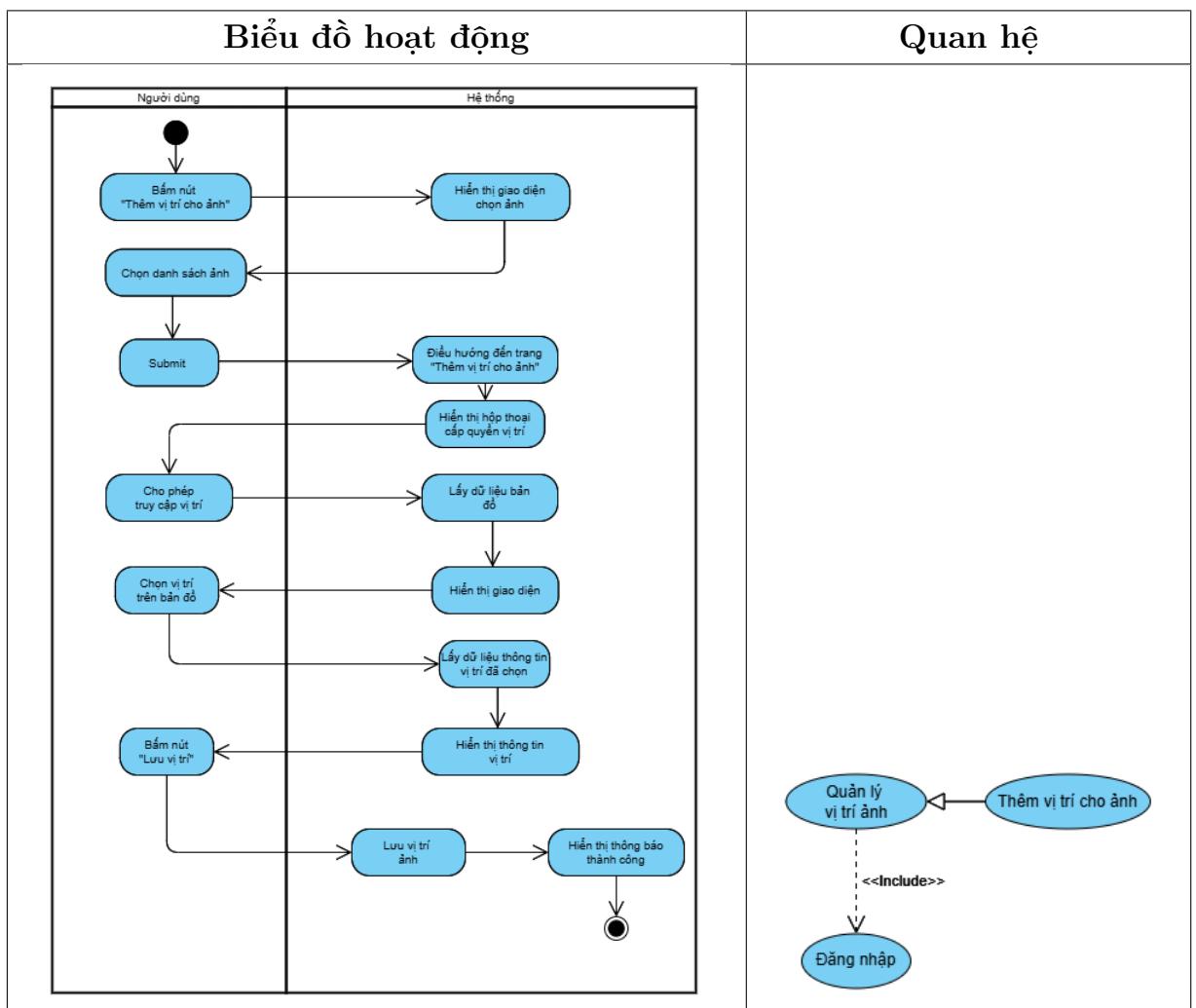
Mô tả chi tiết cho ca sử dụng thêm vị trí cho ảnh được thể hiện ở Bảng 3.15 dưới đây. Kèm theo là Bảng 3.16 về biểu đồ hoạt động, quan hệ và Hình 3.9 về biểu đồ tuần tự của ca sử dụng này.

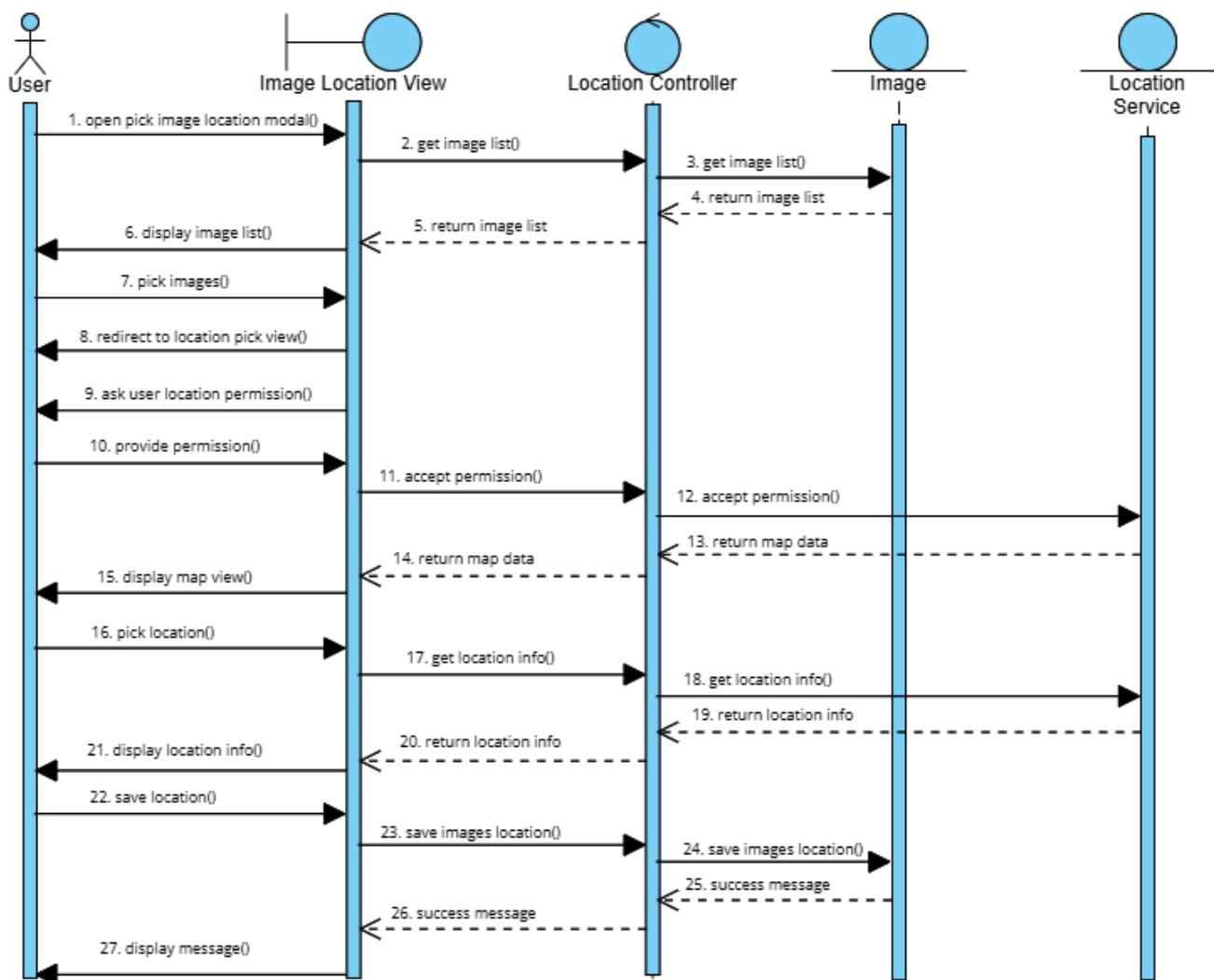
Bảng 3.15: Mô tả chi tiết ca sử dụng thêm vị trí cho ảnh

Mô tả	Người dùng thêm vị trí cho ảnh.
Luồng cơ bản	<ol style="list-style-type: none">Người dùng chọn nút thêm vị trí cho ảnh ở thanh công cụ dưới màn hình.Hệ thống điều hướng đến trang thêm vị trí cho ảnh.Hệ thống hiển thị hộp thoại cấp quyền thông tin vị trí hiện tại.Người dùng cấp quyền truy cập vị trí hiện tại.Hệ thống lấy thông tin vị trí hiện tại của người dùng và hiển thị trên bản đồ.Người dùng chọn vị trí trên bản đồ.Người dùng bấm nút xác nhận để thêm vị trí cho ảnh.Hệ thống lưu vị trí cho ảnh và hiển thị thông báo thành công.
Luồng thay thế	<ul style="list-style-type: none">Người dùng tìm kiếm địa điểm thay vì tự chọn vị trí thủ công trên bản đồ.Hệ thống không lấy được dữ liệu địa điểm từ vị trí người dùng chọn trên bản đồ.
Tiền điều kiện	<ul style="list-style-type: none">Người dùng đã đăng nhập vào hệ thống.Có ít nhất 1 bức ảnh đã tải lên trong thư viện.

Hậu điều kiện	<ul style="list-style-type: none"> - Người dùng có thể xem vị trí hiện tại của bản thân. - Người dùng có thể xem thông tin chi tiết của vị trí vừa chọn.
Yêu cầu phi chức năng	<ul style="list-style-type: none"> - Hệ thống lấy dữ liệu của địa điểm không quá 2s. - Hệ thống lấy được dữ liệu hiện tại vị trí người dùng (nếu được cấp quyền).

Bảng 3.16: Biểu đồ hoạt động và quan hệ ca sử dụng thêm vị trí cho ảnh





Hình 3.9: Biểu đồ tuần tự ca sử dụng thêm vị trí cho ảnh.

3.3.9 Ca sử dụng tìm kiếm hình ảnh

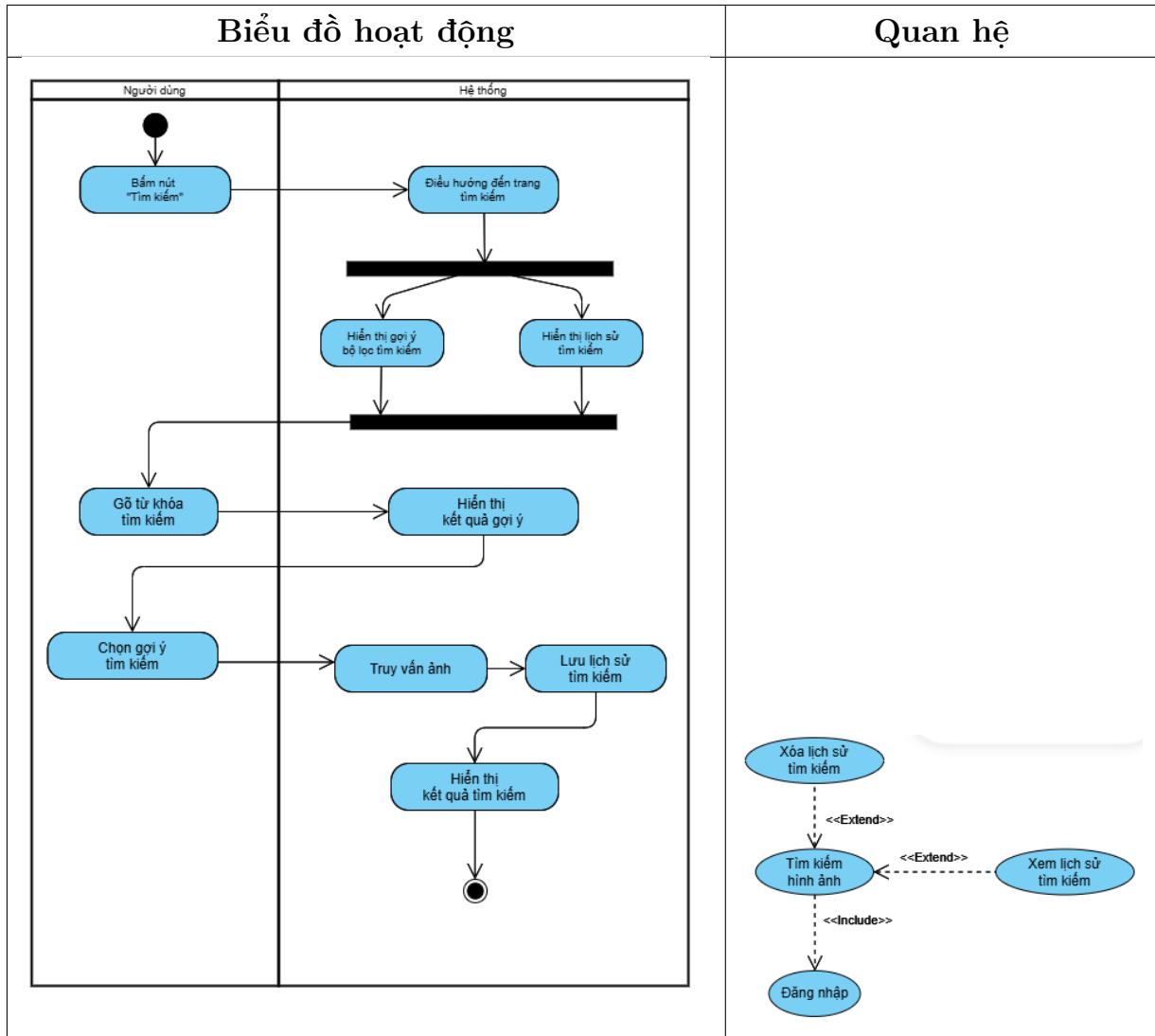
Sau khi tải ảnh lên, người dùng có thể tìm kiếm ảnh theo nhiều tiêu chí khác nhau như thời gian, albums, tên khuôn mặt, tên ảnh, địa điểm, truy vấn AI. Hệ thống sẽ tự động gợi ý cho người dùng những bộ lọc tìm kiếm phù hợp với từ khóa tìm kiếm. Người dùng có thể chọn một trong những gợi ý đó để tìm kiếm ảnh nhanh hơn. Hệ thống sau đó sẽ lưu lịch sử tìm kiếm và hiển thị kết quả cho người dùng.

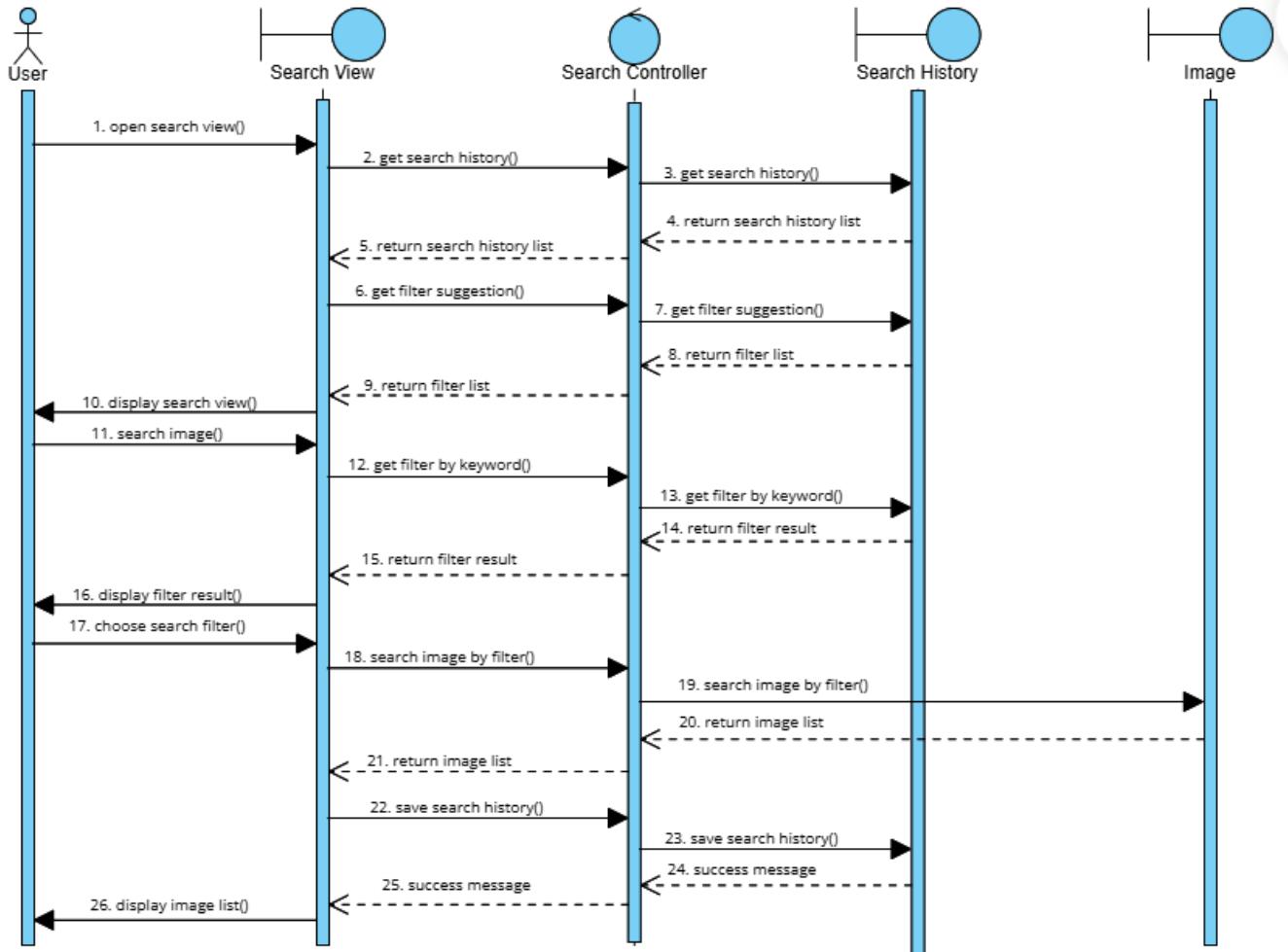
Mô tả chi tiết cho ca sử dụng tìm kiếm hình ảnh được thể hiện ở Bảng 3.17 dưới đây. Kèm theo là Bảng 3.18 về biểu đồ hoạt động, quan hệ và Hình 3.10 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.17: Mô tả chi tiết ca sử dụng tìm kiếm hình ảnh

Mô tả	Người dùng tìm kiếm ảnh với những option hệ thống cung cấp.
Luồng cơ bản	<ol style="list-style-type: none"> 1. Người dùng bấm vào nút tìm kiếm ở thanh công cụ dưới màn hình. 2. Hệ thống điều hướng đến trang tìm kiếm. 3. Hệ thống hiển thị các gợi ý về bộ lọc tìm kiếm (thời gian, albums, tên khuôn mặt, tên ảnh, địa điểm, truy vấn AI) và lịch sử tìm kiếm. 4. Người dùng gõ từ khóa muốn tìm kiếm. 5. Hệ thống hiển thị các gợi ý về bộ lọc có liên quan đến từ khóa tìm kiếm (thời gian, albums, tên khuôn mặt, tên ảnh, địa điểm, truy vấn AI). 6. Người dùng chọn gợi ý tìm kiếm khớp với từ khóa muốn tìm. 7. Hệ thống lưu lịch sử tìm kiếm và hiển thị kết quả theo dạng danh sách.
Luồng thay thế	<ul style="list-style-type: none"> - Người dùng tìm kiếm bằng giọng nói thay vì gõ từ khóa. - Người dùng không gõ từ khóa mà chọn gợi ý tìm kiếm.
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống. - Có ít nhất 1 bức ảnh đã tải lên trong thư viện.
Hậu điều kiện	<ul style="list-style-type: none"> - Người dùng có thể xem thêm thông tin chi tiết về ảnh đã tìm kiếm. - Người dùng có thể xóa bộ lọc tìm kiếm.
Yêu cầu phi chức năng	<ul style="list-style-type: none"> - Hệ thống lấy truy vấn ảnh không quá 2s. - Hệ thống hiển thị gợi ý tìm kiếm không quá 1s.

Bảng 3.18: Biểu đồ hoạt động và quan hệ ca sử dụng tìm kiếm hình ảnh





Hình 3.10: Biểu đồ tuần tự ca sử dụng tìm kiếm hình ảnh.

3.4 Thiết kế cơ sở dữ liệu

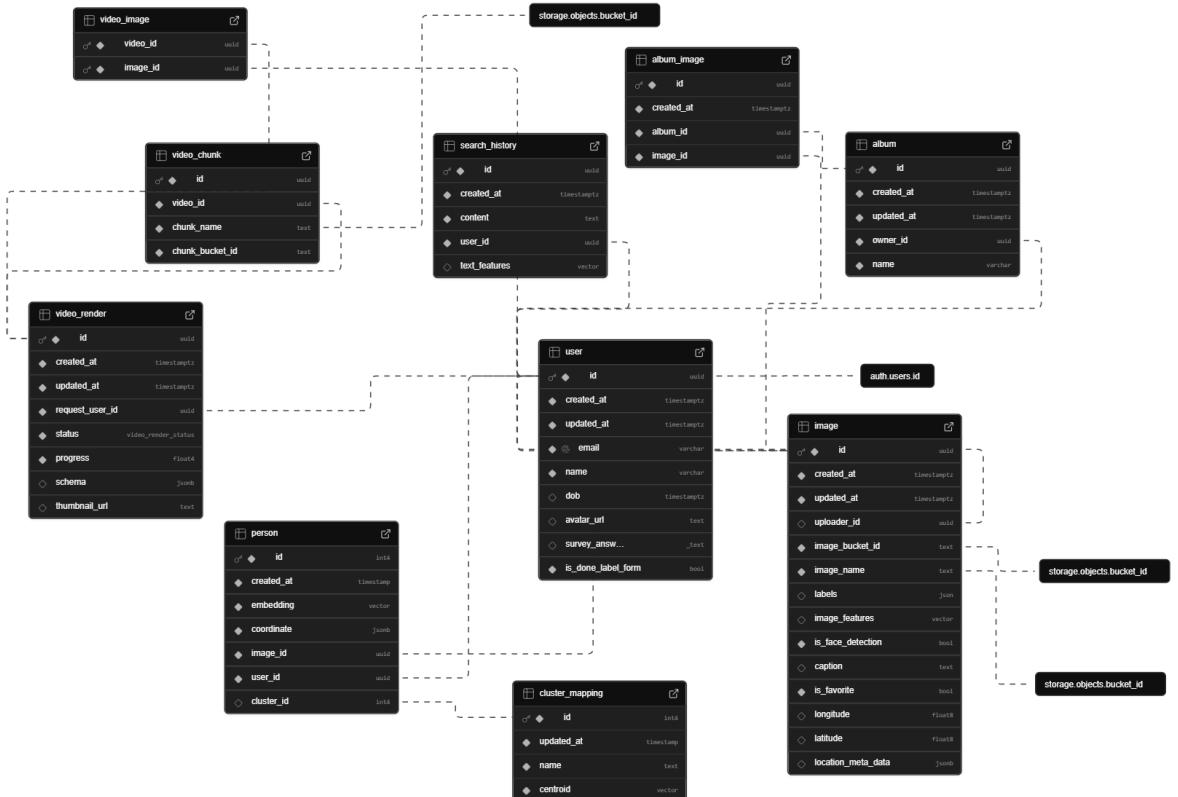
Dựa theo những yêu cầu và thiết kế, cơ sở dữ liệu của hệ thống được thiết kế như Hình 3.11. Hệ thống sử dụng CSDL PostgreSQL được host trên nền tảng Supabase. Với cụ thể thông tin các bảng như sau:

3.4.1 Các bảng dữ liệu chính

- **image**: Lưu trữ thông tin về các hình ảnh được người dùng tải lên. Bảng này chứa các trường quan trọng như "image_bucket_id" và "image_name" (liên kết đến đối tượng lưu trữ trong storage), "uploader_id" (người tải lên), "image_features" (vector đặc trưng của ảnh dùng cho tìm kiếm), "labels"

(nhãn được gắn tự động), thông tin vị trí (longitude, latitude) và trạng thái nhận dạng khuôn mặt.

- **person:** Lưu trữ thông tin về các khuôn mặt được phát hiện trong ảnh. Mỗi bản ghi chứa "embedding" (vector đặc trưng khuôn mặt), "coordinate" (vị trí của khuôn mặt trong ảnh), "image_id" (ảnh chứa khuôn mặt), "user_id" (chủ sở hữu) và "cluster_id" (nhóm khuôn mặt tương tự).
- **cluster_mapping:** Quản lý các nhóm khuôn mặt tương tự nhau, với các trường chính là "name" (tên nhóm, có thể là tên người) và "centroid" (vector đại diện cho nhóm khuôn mặt).
- **album:** Lưu trữ thông tin về các album do người dùng tạo ra, với các trường quan trọng là "owner_id" (người tạo) và "name" (tên album).
- **album_image:** Bảng liên kết giữa album và hình ảnh, cho biết hình ảnh nào thuộc album nào.
- **search_history:** Lưu trữ lịch sử tìm kiếm của người dùng, bao gồm "content" (nội dung tìm kiếm), "user_id" (người tìm kiếm). Ngoài ra trường "text_features" (vector đặc trưng của nội dung tìm kiếm) được sử dụng để tìm kiếm hình ảnh theo câu hỏi truy vấn của người dùng.
- **video_render:** Quản lý thông tin về các video slideshow, bao gồm "request_user_id" (người yêu cầu tạo video), "status" (trạng thái render video), "progress" (tiến độ hoàn thành), "schema" (kịch bản và option video) và "thumbnail_url" (link ảnh đại diện video)
- **video_chunk:** Lưu trữ thông tin về các phân đoạn của video được tạo ra theo chuẩn HLS, liên kết với video_id và có thông tin về tên và vị trí lưu trữ của từng chunk.
- **video_image:** Bảng liên kết giữa video và các hình ảnh được sử dụng trong video đó.



Hình 3.11: Cấu trúc cơ sở dữ liệu.

3.4.2 Tích hợp với các bảng hệ thống của Supabase

Ngoài các bảng được định nghĩa trong schema public, hệ thống còn sử dụng hai bảng đặc biệt có sẵn của Supabase:

- **auth.user**: Quản lý thông tin xác thực và tài khoản người dùng, được cung cấp bởi hệ thống xác thực của Supabase. Bảng này lưu trữ thông tin như email, mật khẩu (đã được mã hóa), thời gian đăng ký và đăng nhập cuối cùng. Nhiều bảng trong hệ thống tham chiếu đến bảng này dưới hình thức khóa ngoại, để khi người dùng xóa tài khoản, các thông tin không cần thiết trong các bảng khác cũng sẽ được xóa theo.
- **storage.objects**: Quản lý các đối tượng lưu trữ (tệp) trong hệ thống lưu trữ của Supabase. Hệ thống sử dụng bảng này để lưu trữ ảnh trong thư viện, ảnh đại diện và các chunk video recap của người dùng.

Thiết kế cơ sở dữ liệu này cho phép hệ thống Smart Gallery lưu trữ và quản lý

hiệu quả các dữ liệu liên quan đến hình ảnh, khuôn mặt, album, tìm kiếm và video, đồng thời tận dụng các tính năng xác thực và lưu trữ có sẵn của Supabase.

3.4.3 Triggers và Functions tự động

Ngoài các bảng dữ liệu chính, hệ thống còn sử dụng 2 tính năng của cơ sở dữ liệu PostgreSQL là triggers và functions để tự động hóa một số quy trình trong cơ sở dữ liệu. Cụ thể:

- **trigger _image _face _detection:** là 1 hàm được kích hoạt khi có 1 ảnh mới được chèn vào bảng "images". Tự động thông báo cho server để xử lý và phát hiện khuôn mặt trong bức ảnh.
- **search _similar _images:** là 1 hàm được sử dụng để tìm kiếm các hình ảnh tương tự trong cơ sở dữ liệu dựa trên vector đặc trưng của hình ảnh, được thiết kế chi tiết như dona. Hàm này sử dụng phương pháp tìm kiếm cosine similarity để tìm ra các hình ảnh có vector tương tự nhất với vector đầu vào. Hàm được gọi từ server khi người dùng thực hiện tìm kiếm hình ảnh. Chi tiết hàm được định nghĩa như sau:

Đoạn mã 3.1: Hàm tìm kiếm hình ảnh tương tự

```
1 BEGIN
2   RETURN QUERY
3   WITH search_history_features AS (
4     SELECT text_features FROM public.search_history
5     WHERE id = search_history_id
6   )
7   SELECT
8     i.id AS image_id, i.image_name, i.image_bucket_id,
9     -- Calculate cosine similarity
10    (1 - (i.image_features <=> sh.text_features)) AS similarity
11   FROM
12     public.image i, search_history_features sh
13   WHERE
14     i.uploader_id = user_id
15    AND (1 - (i.image_features <=> sh.text_features)) >=
16      similarity_threshold
17   ORDER BY similarity DESC;
18 END;
```

Chương 4

TRIỂN KHAI VÀ KIỂM THỬ ỨNG DỤNG QUẢN LÝ THƯ VIỆN ẢNH TÍCH HỢP AI TẠO VIDEO

4.1 Triển khai hệ thống

Phần này trình bày chi tiết về quy trình triển khai hệ thống Smart Gallery, bao gồm các khía cạnh thiết kế kiến trúc và cài đặt các thành phần chức năng. Đầu tiên, luận văn mô tả phương pháp thiết kế và triển khai quy trình tạo video tự động từ bộ sưu tập hình ảnh. Tiếp theo, phân tích cấu trúc tổng thể của hệ thống với các lớp thành phần và mối quan hệ giữa chúng sẽ được trình bày.

4.1.1 Triển khai quy trình tạo video

4.1.1.1 Xây dựng kịch bản video

Một trong những tính năng nổi bật của hệ thống Smart Gallery là khả năng tạo video slideshow tự động từ bộ sưu tập ảnh. Để thực hiện điều này, hệ thống cần xây dựng một kịch bản video có cấu trúc hợp lý, đảm bảo nội dung mạch lạc và hấp dẫn. Kịch bản video được chia thành bốn phần chính:

1. **Intro:** Phần mở đầu của video bao gồm:

- Slide tiêu đề: Hiển thị tiêu đề của video do người dùng đặt, thường được trình bày với hiệu ứng chuyển động và phông chữ đặc biệt.
- Slide tổng quát: Giới thiệu tổng quan về nội dung video, sử dụng các hình ảnh đại diện được chọn lọc từ tập ảnh input của người dùng.



(a) Slide tiêu đề video.



(b) Slide tổng quát video.

Hình 4.1: Các thiết kế cho phần Intro của video slideshow.

2. Content: Phần nội dung chính của video: Phần nội dung chính của video được tổ chức thành các chương (chapters), được phân chia dựa trên nhãn location và event của ảnh. Mỗi chương được cấu trúc như sau:

- Slide tiêu đề chương: Hiển thị tên của chương tương ứng với nhãn location hoặc event.
- Slide trình chiếu ảnh: Hiển thị các ảnh thuộc cùng một nhóm nhãn, kèm theo hiệu ứng chuyển cảnh.

Tiêu đề và caption của các khung hình được tạo tự động bằng mô hình Gemini thông qua prompt sau:

```
// slide[] -> caption[]
export const generateSlidesCaptionPrompt = (
  slide_req: SlideCaptionsRequest
) => {
  const prompt = `Bạn là một nhà biên kịch sáng tạo, hãy tạo ra những caption tiếng Việt KHÁC NHAU thật tự nhiên, ngắn gọn và mỗi caption không vượt quá 10 từ, không được giống các caption khác trong danh sách caption cũng như hấp dẫn cho các slide ảnh trong video recap chuyên nghiệp.
Input của bạn là mảng thông tin về các slide ảnh, bao gồm:
- Địa điểm của slide (place).
- Hoạt động trong slide (activity).
- Các sự kiện diễn ra trong slide (events).
Dựa trên thông tin này, hãy tạo ra một mảng các caption cho từng slide (mỗi phần tử object trong input tạo 1 caption), mỗi caption có độ dài tối đa 10 từ, không có dấu câu hoặc thông tin thừa. Hãy trả về mảng caption dưới dạng JSON string, ví dụ: ["Caption slide 1", "Caption slide 2"].
Không cần thêm bất kỳ thông tin nào khác.
Một số lưu ý:
1. Mỗi caption phải khác nhau, ngay cả khi các slide có địa điểm và hoạt động giống nhau.
2. Mỗi phần tử object trong input chỉ được tạo 1 caption.
3. Các caption có thể sử dụng cho mạng xã hội và phải dễ hiểu, tự nhiên và phổ biến đối với người Việt Nam.
4. Chọn ngẫu nhiên một trong các phong cách sau để tạo caption cho từng slide, đảm bảo mỗi slide có caption theo phong cách khác nhau:
  - Phong cách mô tả thông thường.
  - Phong cách kể chuyện (Storytelling).
  - Phong cách hài hước.
  - Phong cách chuyên nghiệp.
  - Phong cách cảm xúc, chân thật.
  - Phong cách khám phá (Adventure).
5. Không được tạo caption mới giống với các caption đã tạo trước đó.
6. Văn phong caption phải thật tự nhiên với ngôn ngữ tiếng Việt.
Input: ${JSON.stringify(slide_req)}.
Lưu ý: Các caption cần phải ngắn gọn, tự nhiên và dễ hiểu với người Việt.`;

  return prompt;
};
```

Hình 4.2: Prompt Gemini để tạo caption cho video slideshow.

Ngoài ra, với mỗi chương, hệ thống đã chuẩn bị sẵn các phong cách trình bày sau cho slide ảnh:

- 2 Style tiêu đề chương được hệ thống chuẩn bị và chọn sẵn cho người dùng.:



(a) Style tiêu đề 1.



(b) Style tiêu đề 2.

Hình 4.3: Các style tiêu đề chương.

- Các style trình chiếu ảnh mà hệ thống cung cấp cho người dùng:



(a) Style slide ảnh 1.

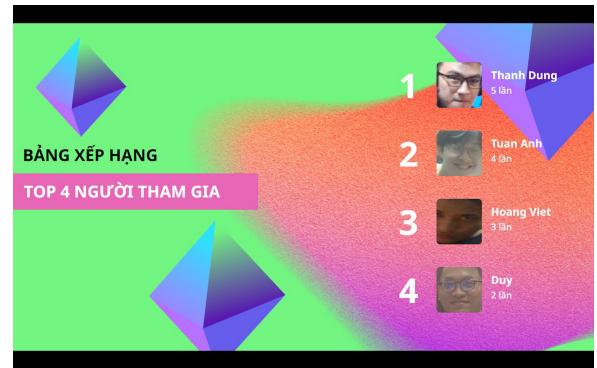


(b) Style slide ảnh 2.

Hình 4.4: Thiết kế các style slide ảnh.

3. Special Part: Phần đặc biệt của video tập trung vào việc điểm lại các thành viên tham gia chuyến đi:

- Chỉ xuất hiện khi bộ sưu tập có ít nhất một khuôn mặt được nhận diện.
- Hiển thị tổng số người đã tham gia trong chuyến đi.
- Trình bày top 4 khuôn mặt xuất hiện nhiều nhất cùng với tên của họ.



(a) Slide giới thiệu phần Special.

(b) Slide hiển thị top khuôn mặt.

Hình 4.5: Thiết kế phần Special Part của video slideshow.

4. Outro: Phần kết thúc của video: Phần này sẽ tóm tắt lại chuyến đi bằng các ảnh tiêu biểu được chọn lọc và các caption được chọn tự động bởi hệ thống.



Hình 4.6: Thiết kế phần Outro của video slideshow.

- Thuật toán xây dựng kịch bản (Content): Phần nội dung chính của video được xây dựng dựa trên một thuật toán phân nhóm và tổ chức ảnh thành các chương logic. Thuật toán này chia tập ảnh input của người dùng thành các nhóm dựa trên nhãn location và event của ảnh. Mỗi nhóm sẽ được tổ chức thành một

chương riêng biệt trong video. Sau đây là cấu trúc JSON của label ảnh:

Đoạn mã 4.1: Cấu trúc JSON của label ảnh

```
1 {
2     "location_labels": [
3         {"temple": 0.9948057532310486},
4         {"palace": 0.0011160931317135692}
5     ],
6     "action_labels": [
7         {"visiting historical sites": 0.8663983941078186},
8         {"visiting museum": 0.039322637021541595}
9     ],
10    "event_labels": [
11        {"Spring": 0.3410003185272217},
12        {"Conference": 0.1293141394853592}
13    ]
14 }
```

Quy trình tạo kịch bản phần Content (nội dung chính) của video bao gồm năm bước chính:

- **Bước 1 - Labeling:** Phát hiện và gán nhãn cho những ảnh chưa có label.
- **Bước 2 - Nhóm ảnh theo nhãn:** Nhóm các ảnh theo nhãn label có độ tin cậy cao nhất của từng ảnh. Ví dụ: {mountain: 2 ảnh}, {park: 3 ảnh}.
- **Bước 3 - Nhóm theo tập nhãn "broader group":** Các nhóm có số lượng ảnh < 3 sẽ được nhóm chung dựa theo tập nhãn rộng hơn. Tập nhãn "broader group" là tập hợp các nhãn location được phân loại vào các nhóm lớn hơn (như "nature" bao gồm "sea", "mountain", "forest", v.v.). Ví dụ: nếu nhóm "sea" có 2 ảnh và nhóm "mountain" có 2 ảnh, cả hai sẽ được gộp thành nhóm "nature" với 4 ảnh.
- **Bước 4 - Tối ưu hóa nhóm:** Kiểm tra lại các nhóm broader location có số lượng < 3 ảnh. Nếu phát hiện, hệ thống sẽ đưa nhóm này vào nhóm location thuộc broader location đó và có số lượng ảnh nhiều nhất. Ví dụ: nhóm

"workspace" (gồm 2 ảnh) sẽ được gộp vào nhóm "classroom" (có 4 ảnh) nếu "classroom" thuộc nhóm broader "workspace" và có số lượng ảnh nhiều nhất trong các nhóm thuộc "workspace".

- **Bước 5 - Xử lý ngoại lệ:** Duyệt lại lần cuối, những nhóm chỉ có 1 ảnh sẽ được gộp vào một chương chung có tên "Small Part" hoặc "Khoảnh khắc khác".
 - **Bước 6 - Tạo nội dung từng chương:** Trong các chương, với tập ảnh đã được nhóm và phân chia từ các bước trước, hệ thống sẽ tạo tiêu đề chương và chia các ảnh trong chương đó thành các slide ảnh. Mỗi slide ảnh được nhóm theo nhãn activity.
 - **Bước 7 - Tạo tiêu đề và caption:** Tiêu đề chương và caption được tạo tự động bằng mô hình Gemini thông qua prompt đã nêu ở trên.
 - **Bước 8 - Chọn phong cách trình bày:** Hệ thống sẽ chọn ngẫu nhiên một trong các phong cách trình bày đã được chuẩn bị sẵn cho tiêu đề chương và slide ảnh (nếu người dùng không chọn phong cách cụ thể nào).
- **Lưu trữ kịch bản video:** Sau khi được xây dựng hoàn chỉnh, kịch bản video được lưu trữ vào cơ sở dữ liệu trong bảng `video_render`, cụ thể là trong trường `schema`. Thông tin này sau đó được sử dụng bởi các module render video để tạo ra video slideshow hoàn chỉnh theo kịch bản đã định. Ngoài ra, việc lưu trữ kịch bản cho phép hệ thống:
- Xem lại và chỉnh sửa kịch bản trước khi render nếu cần
 - Tái sử dụng kịch bản cho các video khác với cùng bộ ảnh
 - Theo dõi và phân tích các mẫu kịch bản phổ biến để cải thiện thuật toán

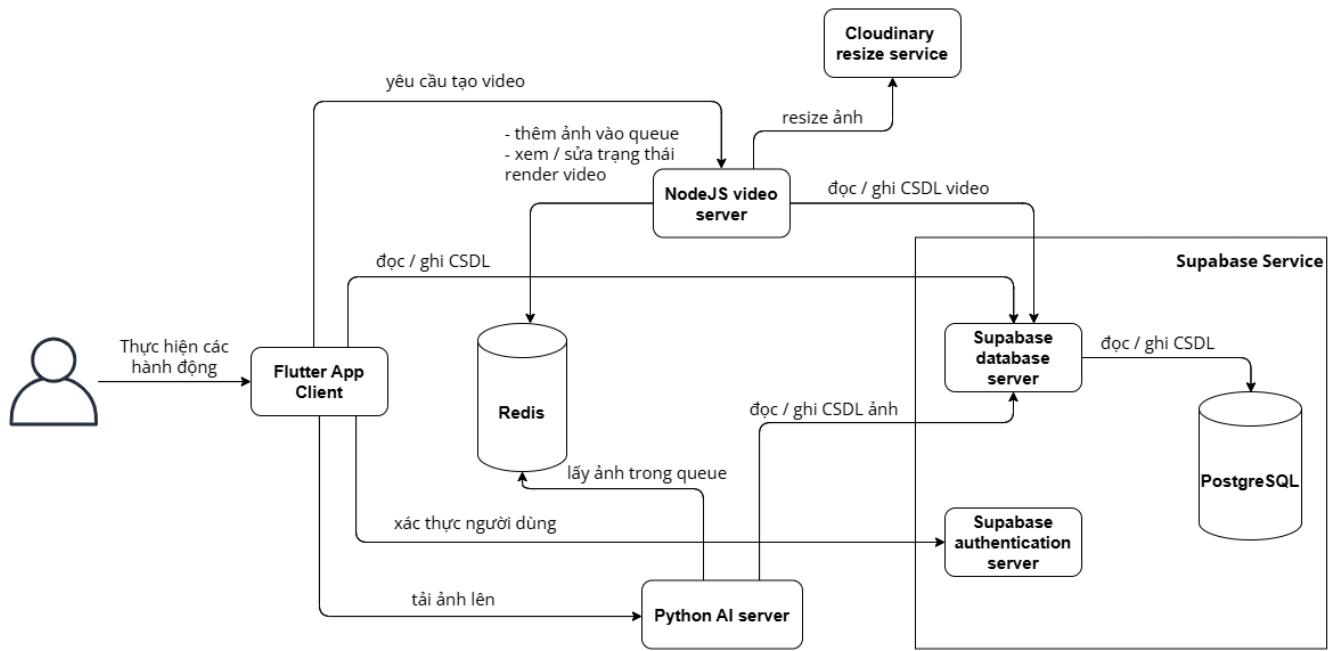
4.1.1.2 Tạo video slideshow

Video sẽ được hệ thống render dựa trên kịch bản đã lưu trong CSDL thông qua những bước sau:

1. **Lấy kịch bản video:** Hệ thống sẽ truy vấn kịch bản video từ bảng `video_render` trong cơ sở dữ liệu.
2. **Tối ưu hóa render:** Trong quá trình thử nghiệm và đo đặc thời gian tạo video, em đã phát hiện ra rằng thời gian render video bị khá lâu ở phần Intro, nên em đã tối ưu bằng cách upload và resize ảnh của video bằng server của Cloudinary. Do hạn chế về dung lượng upload của Cloudinary, hiện tại phần tối ưu hóa render sẽ chỉ được hệ thống áp dụng cho phần Intro của video.
3. **Ghép kịch bản video:** Hệ thống ghép các yếu tố trong kịch bản video vào các khung hình (frame) được lập trình sẵn. Các yếu tố này bao gồm: ảnh, tiêu đề chương, caption, hiệu ứng chuyển cảnh, âm thanh nền và các yếu tố khác.
4. **Render video:** Hệ thống sẽ sử dụng thư viện Remotion để tạo video định dạng mp4 với các khung hình và kịch bản đã lưu. Sau đó lưu video vào 1 thư mục tạm thời trên server. Trong quá trình render, hệ thống sẽ lưu trạng thái render của người dùng vào trong CSDL Redis để tránh tình trạng người dùng tạo nhiều video 1 lúc, qua đó giảm tải cho server.
5. **Tối ưu hóa video cho quá trình streaming:** Sau khi đã hoàn thành video, hệ thống sẽ sử dụng thư viện ffmpeg để chuyển đổi định dạng video sang m3u8, giúp tối ưu hóa cho quá trình streaming video trên web.
6. **Lưu video vào CSDL:** Video sau khi được tối ưu hóa sẽ được lưu vào bảng `video_chunk` trong cơ sở dữ liệu,.

4.1.2 Kiến trúc hệ thống

Kiến trúc của hệ thống hoạt động theo như Hình 4.7



Hình 4.7: Biểu đồ kiến trúc hệ thống.

Flutter App Client: cung cấp giao diện giúp người dùng có thể tương tác với hệ thống. Client sẽ giao tiếp với các dịch vụ thông qua API để thực hiện các chức năng như xác thực người dùng, tương tác với database, render video, đánh nhãn ảnh và phân loại khuôn mặt trong ảnh.

Redis: đóng vai trò là một bộ nhớ cache lưu thông tin tạm thời cho các dịch vụ trong hệ thống. Cụ thể như sau:

- Lưu thông tin render video: NodeJS video server sẽ lưu trữ trạng thái của video trong Redis để có thể truy xuất nhanh chóng và đồng thời hạn chế người dùng chỉ được phép render video một lần trong một khoảng thời gian nhất định.
- Công cụ giao tiếp giữa các dịch vụ: hệ thống sử dụng Redis stream như 1 hàng đợi để đánh dấu những ảnh cần được gán nhãn. Khi NodeJS server phát hiện có ảnh nào trong request tạo video chưa được gán nhãn, server sẽ gửi thông tin ảnh đó vào Redis stream. Python AI server sẽ lắng nghe Redis stream và khi có thông tin ảnh mới, server sẽ lấy ảnh từ kho dữ liệu của Supabase Service, thực hiện gán nhãn cho ảnh và lưu lại kết quả vào kho dữ liệu của Supabase.

NodeJS video server: cung cấp dịch vụ render video cho người dùng. Server

này sẽ nhận request tạo video từ client, sau đó resize ảnh bằng Cloudinary resize service, tạo video và lưu video vào kho dữ liệu của Supabase Service.

Python AI server: cung cấp dịch vụ gán nhãn ảnh và tìm kiếm hình ảnh theo văn bản cho người dùng. Ngoài ra server cũng thực hiện việc nhận diện khuôn mặt trong ảnh và nhóm những khuôn mặt tương tự nhau để người dùng có thể dễ dàng tìm kiếm và phân loại ảnh.

Supabase Service: được sử dụng cho các chức năng như lưu trữ ảnh, video, thông tin người dùng và lắng nghe thời gian thực về sự thay đổi của dữ liệu. Bao gồm các dịch vụ như sau:

- Supabase database server: lưu trữ các bảng dữ liệu như users, videos, images, faces, v.v. và các dữ liệu tệp tin khác như ảnh, video.
- Supabase authentication server: thực hiện xác thực và ủy quyền người dùng. Hệ thống sử dụng JWT token để xác thực người dùng và phân quyền truy cập cho các dịch vụ trong hệ thống.
- PostgreSQL: là hệ quản trị cơ sở dữ liệu được Supabase sử dụng.

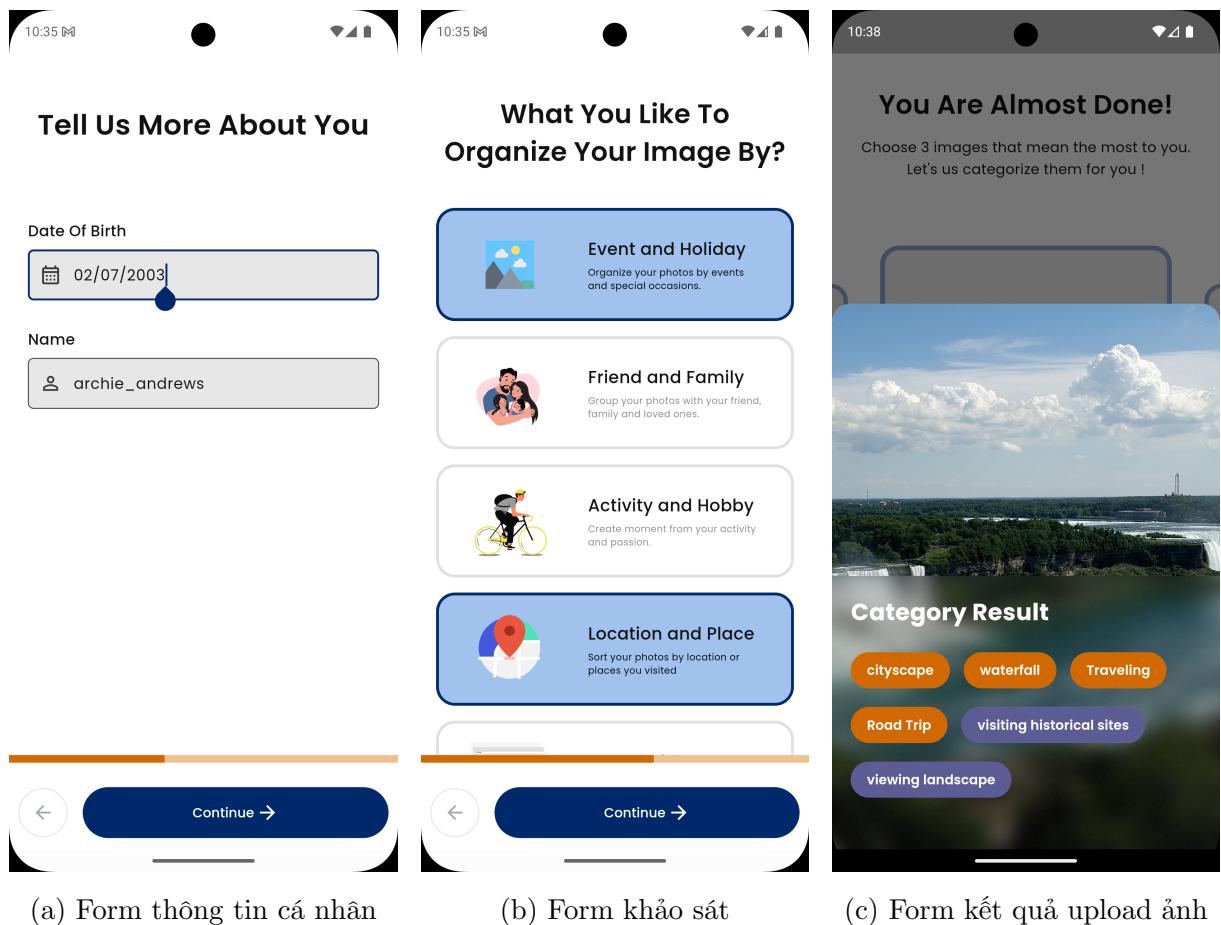
Cloudinary resize service: cung cấp dịch vụ resize ảnh cho hệ thống. Hệ thống sử dụng Cloudinary để resize ảnh trước khi tạo video, nhằm tăng tốc độ tạo và giảm dung lượng tạo video.

4.2 Các chức năng chính của hệ thống

Chương này sẽ trình bày các chức năng chính của hệ thống, bao gồm các chức năng chính mà người dùng có thể sử dụng để tạo ra video slideshow từ bộ sưu tập ảnh của họ. Nội dung chương sẽ bao gồm mô tả chi tiết các chức năng kèm theo hình ảnh thực tế giao diện ứng dụng của hệ thống. Các chức năng chính của hệ thống bao gồm:

4.2.1 Xác thực người dùng

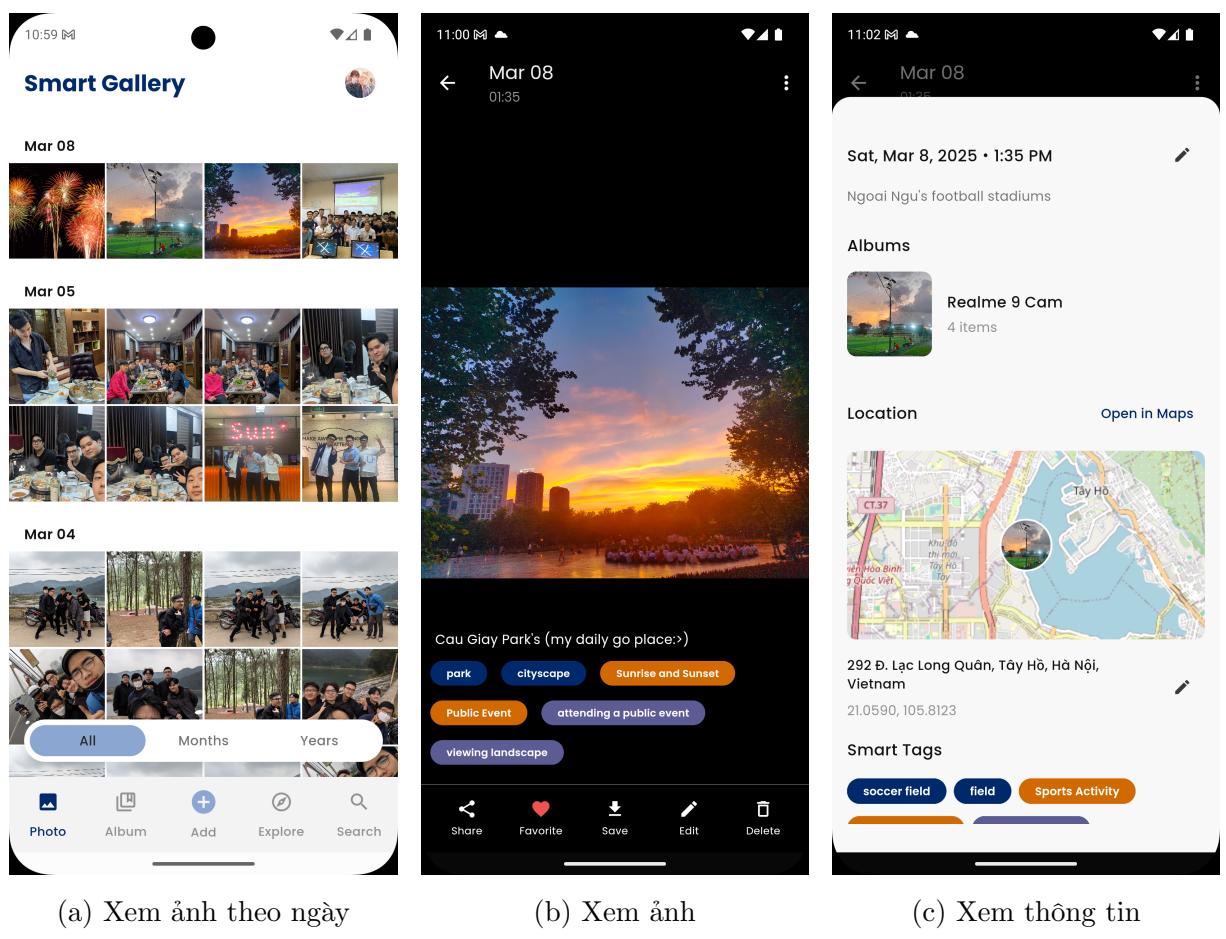
Sau khi người dùng tạo tài khoản thành công, hệ thống sẽ điều hướng người dùng đến trang điền thông tin tài khoản như Hình 4.8. Tại đây, người dùng sẽ phải điền ngày sinh, tên, ảnh đại diện, upload 1 số ảnh cá nhân và thực hiện khảo sát hệ thống trước khi tiến hành sử dụng chức năng chính của hệ thống.



Hình 4.8: Giao diện điền thông tin tài khoản.

4.2.2 Quản lý ảnh

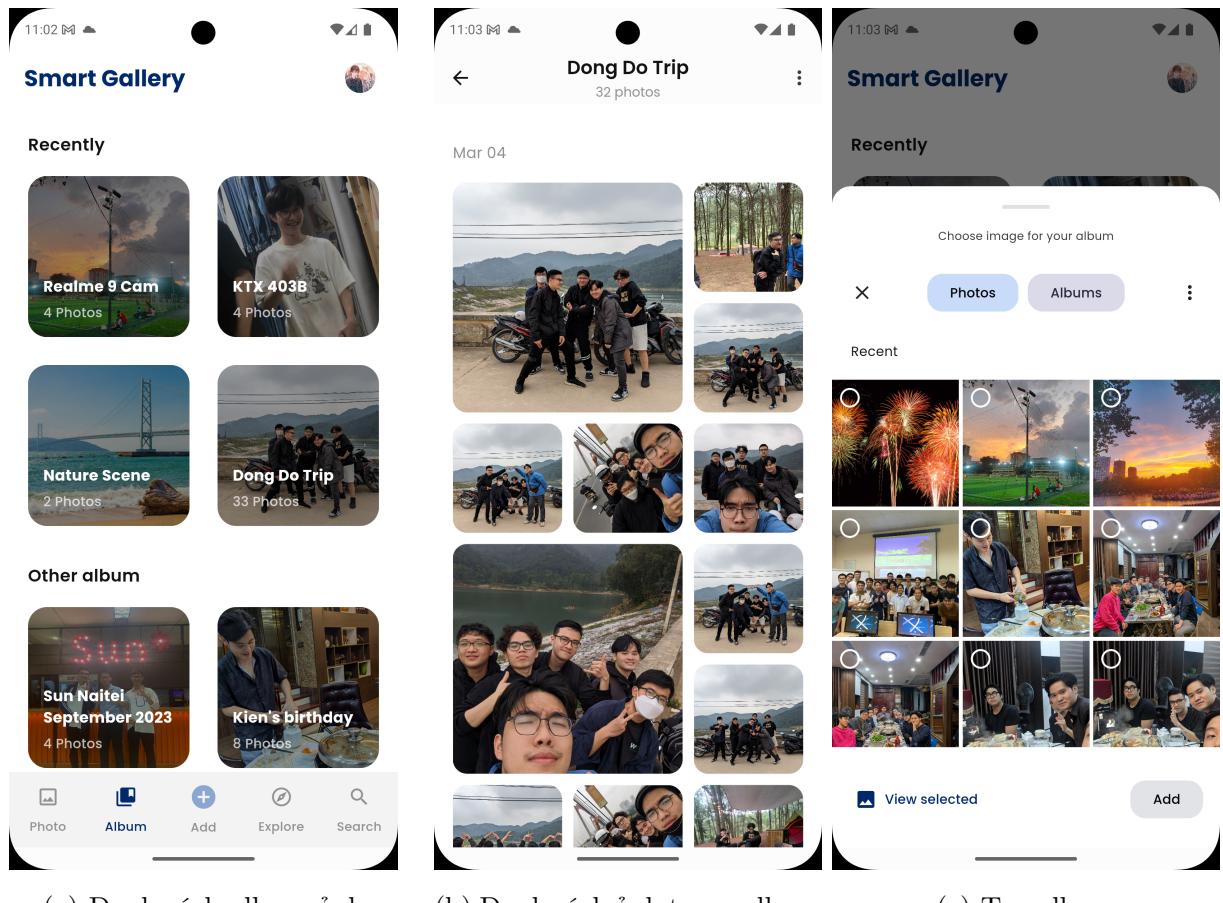
Sau khi đăng nhập thành công, hệ thống sẽ điều hướng người dùng đến trang chủ thư viện ảnh, nơi người dùng có thể xem các ảnh đã tải lên theo ngày / tháng / năm. Ngoài ra người dùng có thể bấm vào từng ảnh để xem thông tin ảnh. tại đây người dùng có thể thực hiện các chức năng như xóa ảnh, yêu thích ảnh, đổi tên ảnh và xem thông tin chi tiết của ảnh như thời gian chụp, vị trí chụp, các tag liên quan đến ảnh và các khuôn mặt được nhận diện trong ảnh như hình 4.9.



Hình 4.9: Giao diện thư viện ảnh.

4.2.3 Quản lý album ảnh

Người dùng có thể quản lý, nhóm ảnh theo các album, xem danh sách các ảnh trong album đó và tạo album như Hình 4.10. Tại đây người dùng có thể xem danh sách các album đã tạo, tải album về máy hay xóa album.

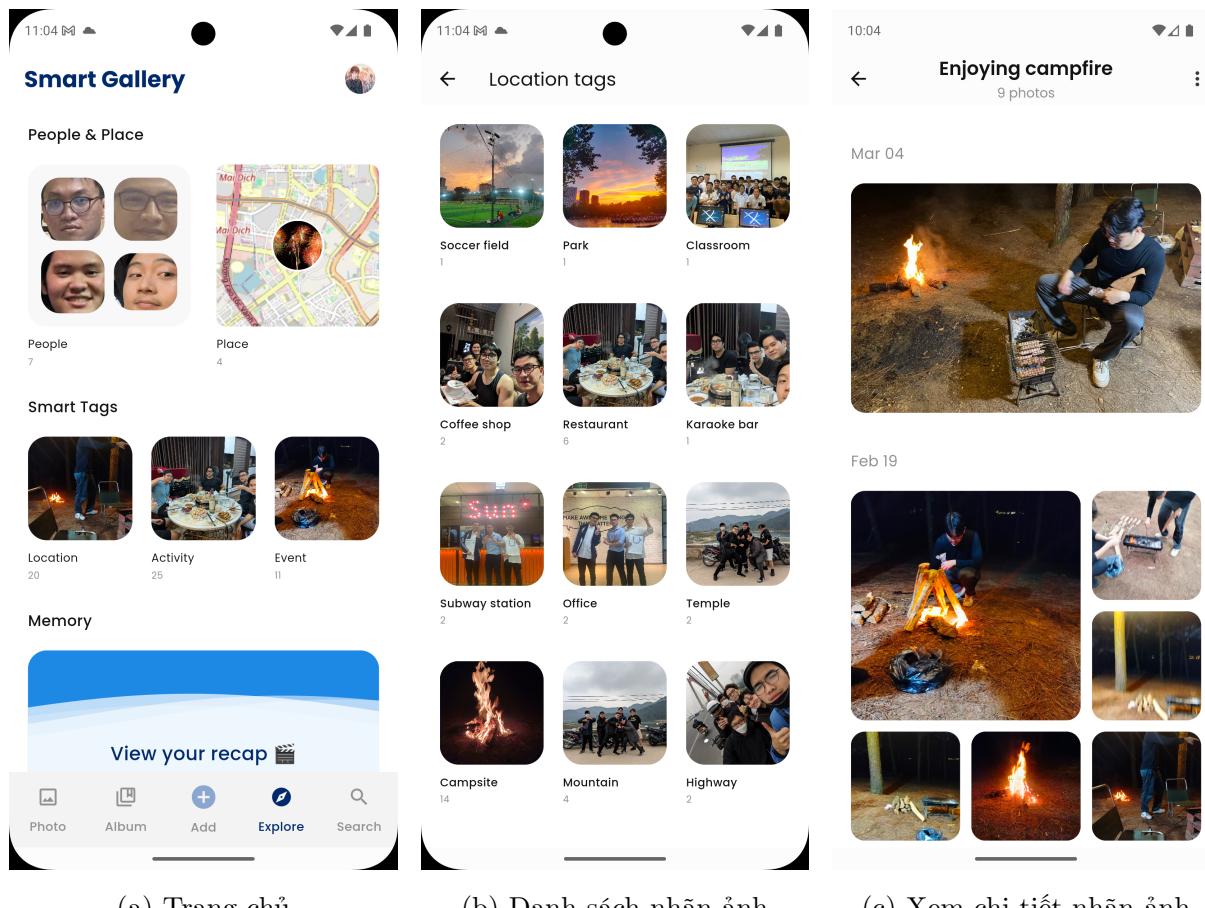


Hình 4.10: Giao diện album ảnh.

4.2.4 Giao diện khám phá

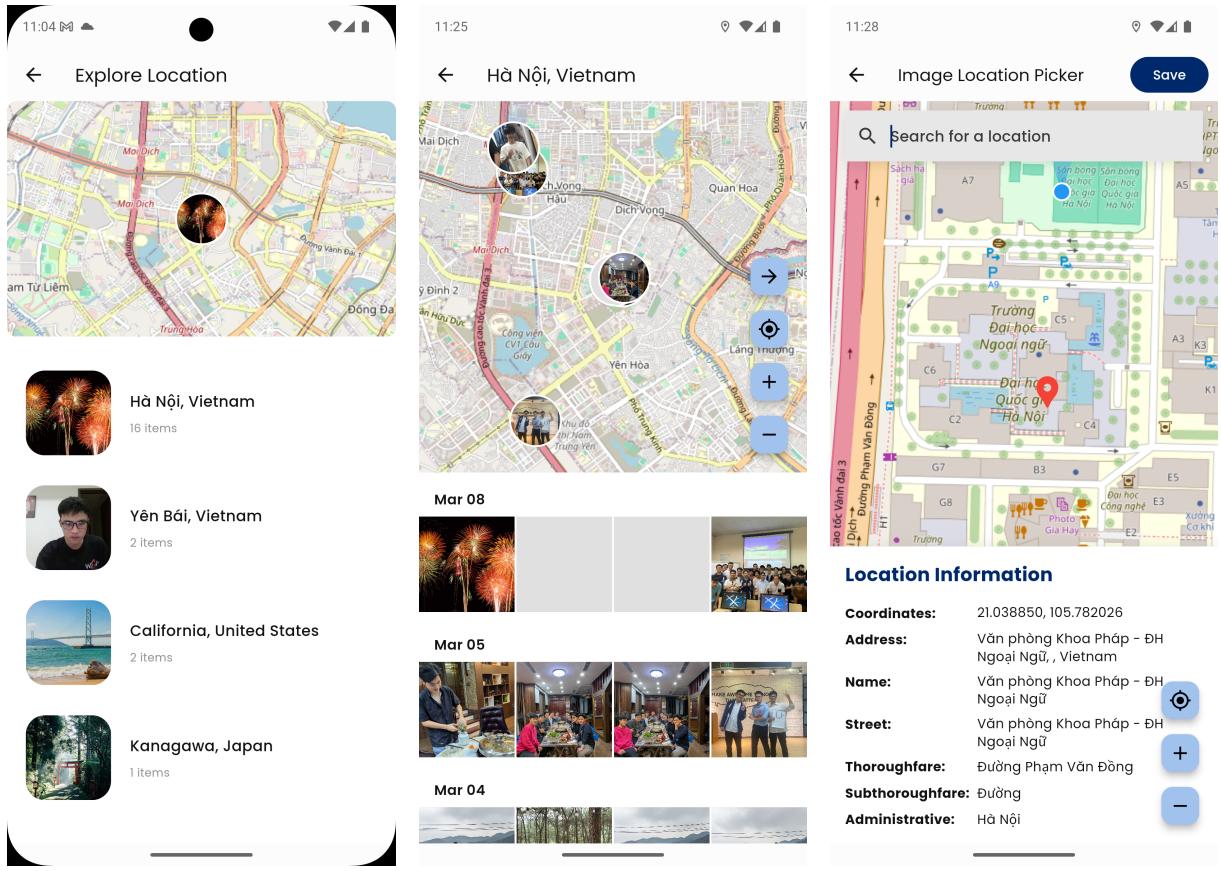
Hệ thống cung cấp chức năng khám phá cho phép người dùng khám phá những khía cạnh khác nhau của ảnh như nhãn, vị trí chụp, khuôn mặt trong bức ảnh.

Các nhãn được phân loại từ các ảnh người dùng tải lên sẽ được hệ thống phân loại, nhóm thành các nhóm khác nhau dựa theo địa điểm, hành động và sự kiện trong ảnh như Hình 4.11.



Hình 4.11: Giao diện khám phá.

Hệ thống cũng cung cấp tính năng quản lý ảnh theo địa điểm. Người dùng có thể xem, chỉnh sửa các vị trí cho ảnh, đồng thời được hệ thống phân nhóm các ảnh theo vị trí chụp như Hình 4.12.



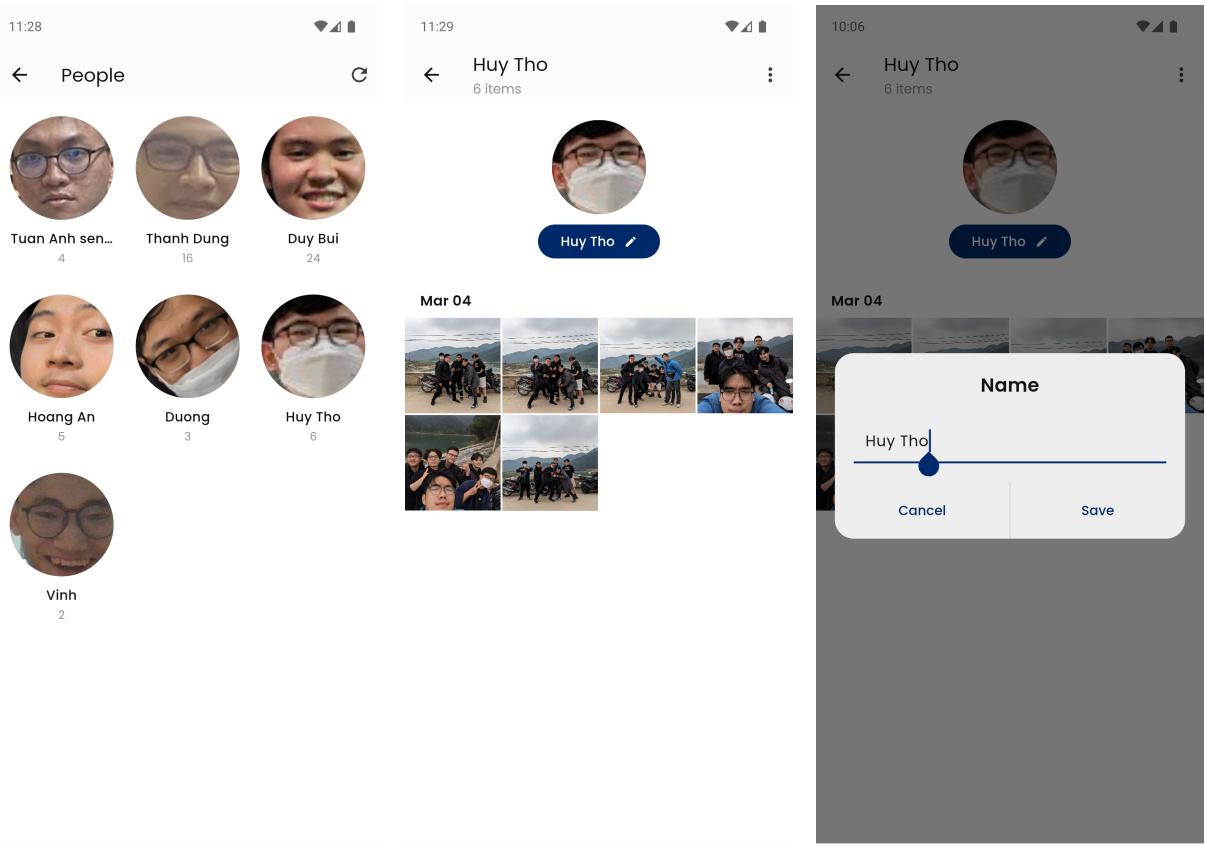
(a) Nhóm ảnh theo vị trí

(b) Xem ảnh theo vị trí

(c) Thêm vị trí cho ảnh

Hình 4.12: Giao diện quản lý vị trí ảnh.

Ngoài ra hệ thống cũng cung cấp tính năng nhận diện khuôn mặt trong ảnh. Người dùng có thể xem, chỉnh sửa các khuôn mặt trong ảnh, đồng thời được hệ thống phân nhóm các ảnh theo khuôn mặt như Hình 4.13.



(a) Danh sách khuôn mặt

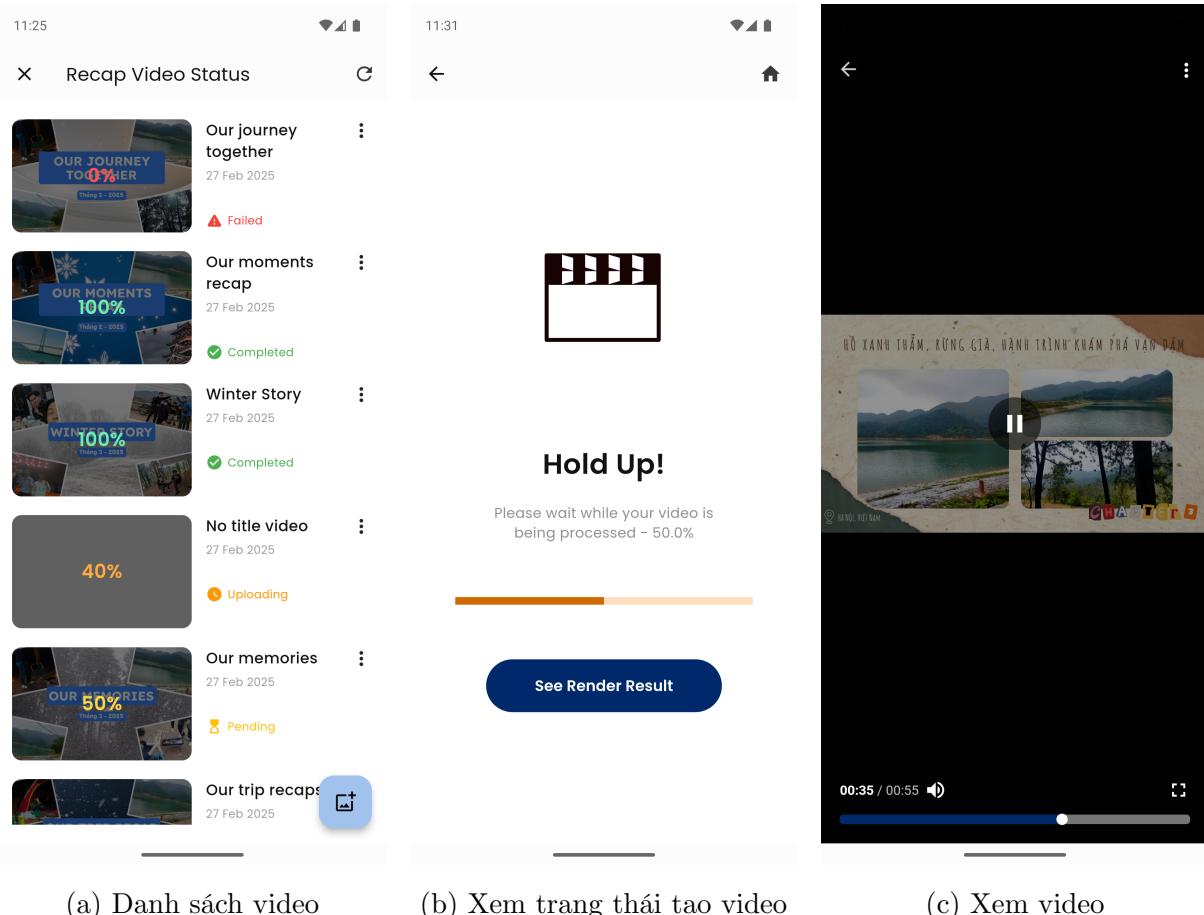
(b) Xem ảnh theo khuôn mặt

(c) Chính sửa tên

Hình 4.13: Giao diện quản lý khuôn mặt.

4.2.5 Quản lý video recap

Người dùng có thể quản lý video recap của mình thông qua chức năng này. Hệ thống cho phép người dùng xem và theo dõi tiến độ tạo video của mình theo thời gian thực như Hình 4.14.



Hình 4.14: Giao diện quản lý video recap.

Ngoài ra, hệ thống cũng cung cấp tính năng tạo video cho người dùng và tùy chỉnh nhiều tùy chọn của video như chất lượng, chủ đề, nhạc nền, tiêu đề, v.v. như Hình 4.15.

11:31



Choose Image For Your Recap Video



Browse your online gallery

or



Upload From Your Device

Import

Add or remove image

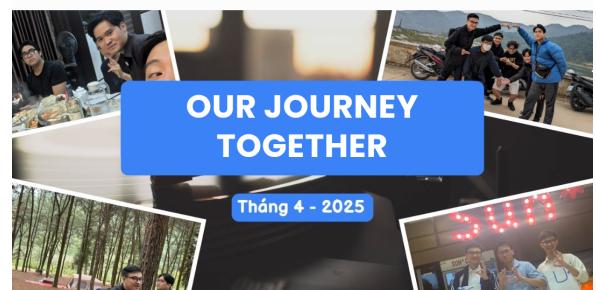


11:35



Edit Video Schema

Create



Our journey

and

Choose title style



Video duration

No limit

Background theme

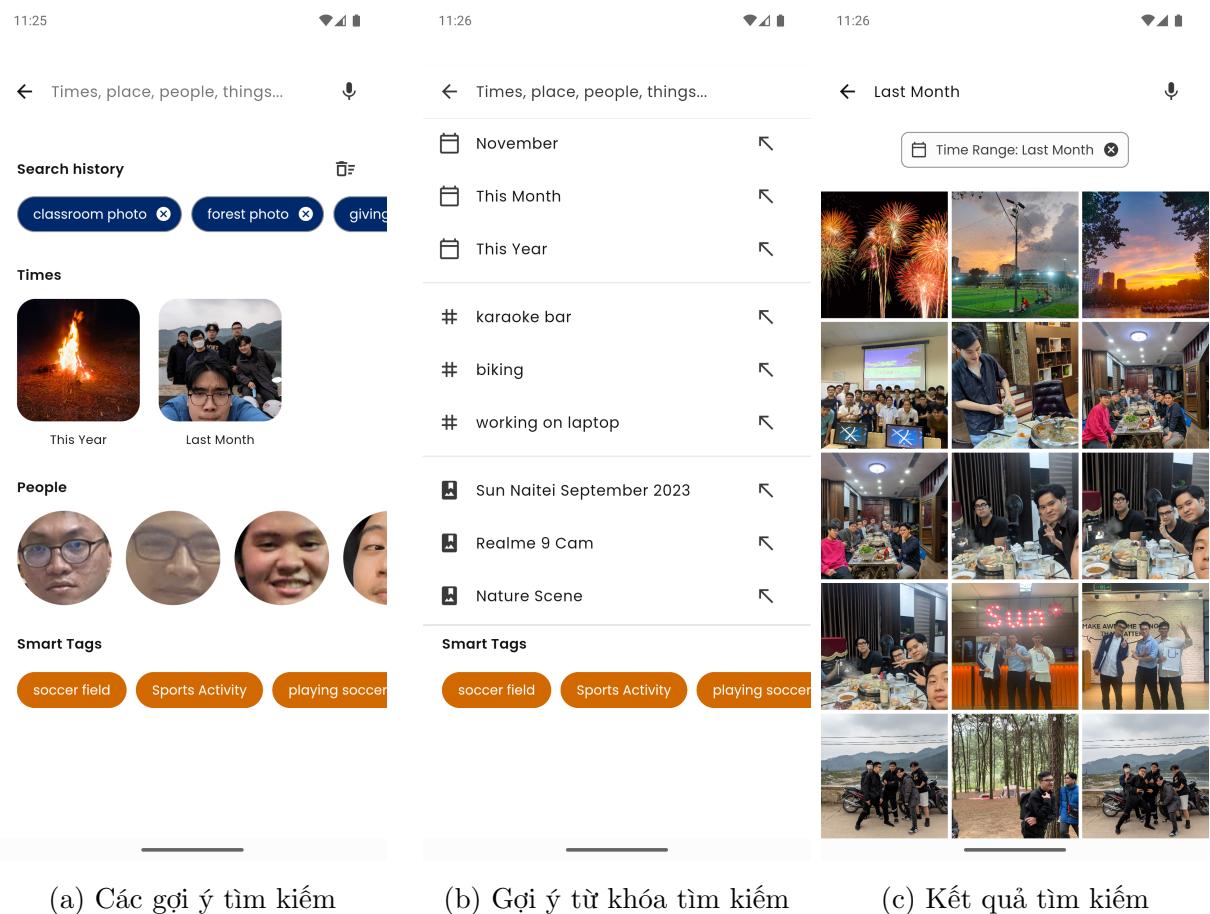


(a) Tạo video

Hình 4.15: Giao diện tạo video.

4.2.6 TÌM KIẾM ẢNH

Người dùng có thể tìm kiếm các ảnh trong hệ thống với các từ khóa khác nhau như tên album, tên ảnh, tên người dùng, hoặc các từ khóa khác liên quan đến ảnh. Giao diện tìm kiếm ảnh được thể hiện trong hình 4.16.



Hình 4.16: Giao diện tìm kiếm.

4.3 Kiểm thử cho hệ thống

Chương này sẽ trình bày về các phương pháp kiểm thử hệ thống. Bao gồm các phương pháp kiểm thử logic như kiểm thử đơn vị và kiểm thử API. Ngoài ra, chương cũng sẽ trình bày về kết quả của các ca kiểm thử tương tác người dùng trên giao diện ứng dụng.

4.3.1 Kiểm thử các xử lý logic

Để đảm bảo hệ thống hoạt động ổn định và chính xác, các xử lý logic của hệ thống cần được kiểm thử kỹ lưỡng. Các kiểm thử này sẽ giúp phát hiện và sửa chữa các lỗi trong mã nguồn, đảm bảo rằng các chức năng của hệ thống hoạt động như mong đợi. Các kiểm thử này bao gồm kiểm thử đơn vị (Unit Testing) và kiểm thử API (API Testing).

4.3.1.1 Kiểm thử đơn vị

Phạm vi kiểm thử

Phạm vi kiểm thử đơn vị cho ứng dụng Smart Gallery bao gồm kiểm thử các hàm xử lý và các lớp xử lý logic của 2 server labeling ảnh và server tạo video.

Môi trường kiểm thử

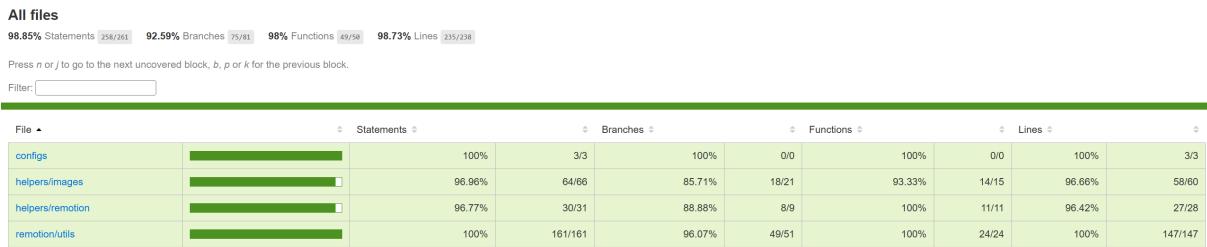
Môi trường kiểm thử cho các xử lý logic gồm có:

- JestJS [21]: là một công cụ được sử dụng để kiểm thử đơn vị được tích hợp sẵn trong NodeJs.
- pytest [22]: là một thư viện kiểm thử đơn vị mạnh mẽ và linh hoạt cho Python. Kết hợp song song với unittest [23], pytest giúp kiểm thử các hàm xử lý logic của server labeling ảnh.

Kết quả kiểm thử

Độ phủ nhánh cho các hàm xử lý logic của server labeling ảnh được miêu tả như Hình 4.17 và đạt 100%, với phạm vi kiểm thử bao gồm các hàm xử lý chính như resize ảnh, phân nhóm khuôn mặt và các hàm xử lý của model AI. Đối với server tạo video, độ phủ nhánh được miêu tả như Hình 4.18 và đạt 92.59%, với phạm vi

kiểm thử bao gồm các hàm xử lý chính như tạo kịch bản video và tạo kịch bản cho video.



Hình 4.17: Độ phủ kiểm thử xử lý logic với JestJS.



Hình 4.18: Độ phủ kiểm thử xử lý logic với pytest.

Đây là kết quả kiểm thử sau khi đã phát hiện và sửa lỗi cho các hàm xử lý logic của 2 server. Một số lỗi đã được phát hiện và sửa chữa trong quá trình kiểm thử, chi tiết các lỗi và cách sửa chữa được mô tả trong các commit trên 2 repository của server: NodeJS Video Server¹ và Python Labeling Server².

¹<https://github.com/LostArrows27/graduation-thesis-nodejs-backend/commits/master/>

²<https://github.com/LostArrows27/graduation-thesis-nodejs-backend/commits/master/>

4.3.1.2 Kiểm thử API

Chi tiết các ca kiểm thử (miêu tả, input và output) được mô tả trong folder “[test]” tại đường dẫn sau đây¹. Hình 4.19 dưới đây mô tả các API endpoint được kiểm thử với Postman [24].

The image shows two side-by-side Postman collections for API testing:

- (a) API server NodeJS:**
 - nodejs server** (selected)
 - home route**
 - GET 1. should return success message**
 - create schema route**
 - POST 0. should deny request without accessToken**
 - POST 1. should check invalid accessToken**
 - POST 2. should deny not exist user**
 - POST 3. should deny request without renderQueueID**
 - POST 4. should deny request without imageIdList**
 - POST 5. should deny request with empty imageIdList array** (highlighted)
 - POST 6. should deny request with invalid renderQueueId**
 - POST 7. should success create video schema without human face**
 - POST 8. should success create video schema with human face**
 - edit schema route**
 - POST 0. should deny request without renderId**
 - POST 1. should deny request without options**
 - POST 2. should deny request with invalid option schema**
 - POST 3. should deny request with invalid renderQueueId**
 - POST 4. should edit schema successfully**
 - render video route**
 - POST 0. should deny request without renderQueueId**
 - POST 1. should deny request with invalid scale**
 - POST 2. should deny request with invalid permission**
 - POST 3. should follow video render status**
 - (b) API server Python:**
 - python server** (selected)
 - home route**
 - GET 1. should return welcome message**
 - image labeling route**
 - POST 0. should reject request with wrong params**
 - POST 1. should reject request wrong image_data structure**
 - POST 2. should reject request wrong user_id**
 - POST 3. should return error message with wrong image information**
 - POST 4. should return empty array with empty image array**
 - POST 5. should return label for image array**
 - image query route**
 - POST 1. should reject wrong input schema request**
 - POST 1. should reject request with wrong user_id**
 - POST 2. should reject request with wrong threshold**
 - POST 3. should reject request with empty query**
 - POST 4. should reject request with invalid user_id**
 - POST 5. should return search query result**
 - person clustering route**
 - POST 0. should reject request with no user_id**
 - POST 1. should return error message with wrong user_id**
 - POST 2. should return no data for user with no person face image**
 - POST 3. should return person clustering group**

Hình 4.19: Các API endpoint được kiểm thử với Postman.

Bảng 4.1 dưới đây mô tả một số kịch bản kiểm thử API chính cho ứng dụng. Đây là các kịch bản kiểm thử và kết quả cho các API chính của ứng dụng sau khi đã sửa lỗi và hoàn thiện.

¹<http://bit.ly/42y7KyF>

Bảng 4.1: Các kịch bản kiểm thử API chính

STT	API	Ca kiểm thử	Kết quả kỳ vọng	Tình trạng
1	API tạo kịch bản video	Tạo kịch bản video từ 5 hình ảnh hợp lệ	Hệ thống tạo được kịch bản, trả về mã 201 và thông tin kịch bản vừa tạo	Đạt
		Tạo kịch bản video không kèm theo ảnh	Hệ thống trả về mã lỗi 400 và thông báo cần gửi kèm danh sách ảnh	Đạt
		Người dùng chưa xác thực tạo kịch bản video	Hệ thống trả về mã lỗi 400 và thông báo cần xác thực	Đạt
2	API sửa kịch bản video	Thay đổi tiêu đề video	Hệ thống thay đổi kịch bản, trả về mã 201 và thông tin kịch bản vừa được cập nhật	Đạt
		Thay đổi kịch bản video với params sai định dạng	Hệ thống trả về mã lỗi 400 và thông báo cần gửi yêu cầu đúng định dạng	Đạt
3	API tạo video	Tạo video với kịch bản có sẵn	Hệ thống trả về ID của video cùng mã 201	Đạt
		Tạo video không kèm theo token người dùng	Hệ thống trả về mã lỗi 400 và thông báo cần xác thực	Đạt
		Yêu cầu tạo cùng 1 video 2 lần	Hệ thống trả về mã lỗi 400 và thông báo lỗi video đang được tạo	Đạt
4	API tìm kiếm hình ảnh	Tìm kiếm với từ khóa	Hệ thống trả về danh sách hình ảnh phù hợp và mã 200	Đạt
		Người dùng chưa xác thực yêu cầu tìm kiếm với từ khóa	Hệ thống trả về mã lỗi 400 và thông báo cần xác thực	Đạt
5	API phân nhóm khuôn mặt	Phân nhóm khuôn mặt với danh sách ảnh không có khuôn mặt	Hệ thống trả về danh sách rỗng và mã 200	Đạt

STT	API	Ca kiểm thử	Kết quả kỳ vọng	Tình trạng
		Phân nhóm khuôn mặt với danh sách 20 ảnh chứa 40 khuôn mặt	Hệ thống trả về danh sách nhóm khuôn mặt tương tự và mã 200	Đạt
6	API gán nhãn ảnh	Phân nhóm khuôn mặt với danh sách ảnh không có khuôn mặt	Hệ thống trả về danh sách rỗng và mã 200	Đạt
		Gán nhãn 1 ảnh	Hệ thống trả nhãn của ảnh và mã 200	Đạt
		Gán nhãn nhiều ảnh	Hệ thống trả về mảng nhãn ảnh và mã 200	Đạt

Trong quá trình kiểm thử các API, một số lỗi đã được phát hiện và khắc phục, đặc biệt là lỗi tại API tạo video khi xử lý trường hợp tạo video với kịch bản không chứa khuôn mặt nào. Lỗi này gây ra việc render video bị thất bại khi không tìm thấy khuôn mặt để hiển thị trong phần cuối video. Vấn đề đã được khắc phục thông qua commit sửa lỗi render video không chứa khuôn mặt¹, trong đó hệ thống được bổ sung logic kiểm tra và xử lý trường hợp đặc biệt này. Sau khi sửa lỗi, API hoạt động ổn định với tất cả các trường hợp kiểm thử.

4.3.2 Kiểm thử tương tác người dùng trên giao diện ứng dụng

Hệ thống triển khai kiểm thử tương tác người dùng trên giao diện với các ca kiểm thử tính năng chính của hệ thống được báo cáo lại trong bảng 4.2.

Bảng 4.2: Các kịch bản kiểm thử tương tác người dùng

STT	Ca kiểm thử	Kết quả kỳ vọng	Tình trạng
1	Đăng nhập với tài khoản hợp lệ	Người dùng được chuyển hướng đến màn hình chính và hiển thị thư viện ảnh	Đạt
2	Đăng nhập với mật khẩu không chính xác	Hiển thị thông báo lỗi "Mật khẩu không chính xác"	Đạt
3	Tải lên ảnh mới từ thiết bị	Ảnh được hiển thị trong thư viện và được phân loại tự động	Đạt

¹<https://github.com/LostArrows27/graduation-thesis-nodejs-backend/commit/0efb26aae61792882dd0937db3be0dfbe6ef75b7>

STT	Ca kiếm thử	Kết quả kỳ vọng	Tình trạng
4	Tìm kiếm ảnh bằng từ khóa	Hiển thị danh sách ảnh có liên quan đến từ khóa	Đạt
5	Phân loại ảnh theo khuôn mặt	Hệ thống nhóm các ảnh có cùng khuôn mặt với độ chính xác trên 85%	Đạt
6	Tạo album mới và thêm ảnh vào	Album được tạo và hiển thị trong danh sách album với đúng số lượng ảnh	Đạt
7	Tạo video recap từ ảnh trong album	Hệ thống tạo video và hiển thị trạng thái xử lý theo thời gian thực	Đạt
8	Xem ảnh theo vị trí trên bản đồ	Bản đồ hiển thị các điểm đánh dấu ảnh theo đúng tọa độ GPS	Đạt
9	Lọc ảnh theo ngày	Hiển thị chính xác những ảnh được chụp trong khoảng thời gian đã chọn	Đạt

Kết luận

Khóa luận này đã xây dựng được hệ thống quản lý thư viện ảnh tích hợp AI tạo video với đầy đủ các chức năng cơ bản như quản lý ảnh, phân loại tự động, tìm kiếm ảnh, và các tính năng nâng cao như tổ chức ảnh theo khuôn mặt, địa điểm, tạo album, và đặc biệt là tạo video recap từ bộ sưu tập ảnh với nhiều tùy chọn về chất lượng, chủ đề, nhạc nền, tiêu đề, v.v.

Bằng việc ứng dụng các mô hình AI như OpenCLIP và Face Recognition, hệ thống có khả năng tự động phân loại và gán nhãn ảnh, nhận diện khuôn mặt người, giúp người dùng quản lý bộ sưu tập hình ảnh của mình một cách thông minh và hiệu quả. Quy trình tạo video slideshow được xây dựng với nhiều thiết kế đa dạng cho phần Intro, Content và Outro, tích hợp thêm trí tuệ nhân tạo để tạo caption phù hợp cho từng video. Những tính năng này không chỉ giúp mang lại trải nghiệm độc đáo cho người dùng trong việc tạo ra những video recap ý nghĩa mà còn giúp họ tiết kiệm thời gian và công sức trong việc tìm kiếm và tổ chức ảnh.

Trong quá trình phát triển, do giới hạn của tài nguyên phần cứng, hệ thống AI và render video hiện chỉ được em chạy trên môi trường local. Đôi khi có sự cạnh tranh tài nguyên GPU giữa các service như phân loại khuôn mặt và gán nhãn ảnh, khiến chúng không thể chạy song song. Thời gian phân loại trung bình cho 300 gương mặt hiện đang là khoảng 10 giây, em sẽ cải thiện thêm về mặt hiệu suất và độ chính xác của thuật toán trong các phiên bản tiếp theo.

Tài liệu tham khảo

- [1] VNetwork. Báo cáo thống kê internet việt nam 2023. <https://www.vnetwork.vn/news/internet-viet-nam-2023-so-lieu-moi-nhat-va-xu-huong-phat-trien/>, 2023.
- [2] DataReportal. Digital 2024: Vietnam. <https://datareportal.com/reports/digital-2024-vietnam>, 2024.
- [3] Q&Me. Vietnam smartphone usage report. <https://qandme.net/en/report/camera-usage-situation-vietnam.html>, 2023. Accessed: 2024.
- [4] CatchLight. The state of photography 2022 report. <https://www.catchlight.io/news/2022/5/2/the-state-of-photography-2022-report>, 2022.
- [5] Usmobile. Apples photos app 2024. <https://www.usmobile.com/blog/apples-photos-app-2024/>, 2024.
- [6] Flutter Team. Docs flutter. <https://docs.flutter.dev/>, 2025.
- [7] Supabase. Supabase flutter sdk. <https://supabase.com/docs/reference/dart/installing>, 2024.
- [8] Sebastián Ramírez. Fastapi documentation. <https://fastapi.tiangolo.com/>, 2023.
- [9] Express.js Team. Express - node.js web application framework. <https://expressjs.com/>, 2023.
- [10] Supabase. Supabase - the open source firebase alternative. <https://supabase.com>, 2025.
- [11] PostgreSQL Global Development Group. Postgresql: The world's most advanced open source relational database. <https://www.postgresql.org/>, 2024.
- [12] Redis Labs. Redis documentation. <https://redis.io/docs/latest/>, 2025.
- [13] Remotion Team. Remotion - create videos programmatically in react. <https://www.remotion.dev/>, 2025.
- [14] Apple Inc. Http live streaming. <https://developer.apple.com/streaming/>, 2023.
- [15] FFmpeg Team. Ffmpeg - the leading multimedia framework. <https://ffmpeg.org/>, 2024.
- [16] ML Foundations. Openclip. https://github.com/mlfoundations/open_clip, 2023.

- [17] Adam Geitgey. Face recognition. https://github.com/ageitgey/face_recognition, 2023.
- [18] Davis King. Dlib c++ library. <http://dlib.net/>, 2023.
- [19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 96(34):226–231, 1996.
- [20] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. An introduction to information retrieval. *Cambridge University Press*, 2008.
- [21] Facebook. Jest - delightful javascript testing. <https://jestjs.io/>, 2023.
- [22] Holger Krekel. pytest - the pytest framework makes it easy to write small tests, yet scales to support complex functional testing for applications and libraries. <https://docs.pytest.org/en/latest/>, 2023.
- [23] Python Software Foundation. unittest - unit testing framework. <https://docs.python.org/3/library/unittest.html>, 2023.
- [24] Postman Inc. Postman - the collaboration platform for api development. <https://www.postman.com/>, 2023.