

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

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
**HỆ THỐNG HỖ TRỢ ĐÁNH GIÁ
ĐIỂM HOẠT ĐỘNG CỘNG ĐỒNG**

Người hướng dẫn: TS. TRƯƠNG NGỌC CHÂU
Sinh viên thực hiện: PHẠM QUỐC TÚ
Số thẻ sinh viên: 102200077
Lớp: 20TCLC_DT1

Đà Nẵng, 06/2024

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

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
HỆ THỐNG HỖ TRỢ ĐÁNH GIÁ
ĐIỂM HOẠT ĐỘNG CỘNG ĐỒNG

Người hướng dẫn: **TS. TRƯƠNG NGỌC CHÂU**
Sinh viên thực hiện: **PHẠM QUỐC TÚ**
Số thẻ sinh viên: **102200077**
Lớp: **20TCLC_DT1**

Đà Nẵng, 06/2024

This image shows a full page of blank handwriting practice paper. It features multiple sets of horizontal lines, each consisting of a solid top line, a dashed midline, and a solid bottom line. These lines are evenly spaced across the entire page, providing a guide for letter height and placement. The background is white, and there are no margins or additional markings.

TS. Trương Ngọc Châu

NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN

[illegible]

Đà Nẵng, ngày ... tháng ... năm 20....

TÓM TẮT

Tên đề tài: Hệ thống hỗ trợ đánh giá điểm hoạt động cộng đồng

Sinh viên thực hiện: Phạm Quốc Tú

Số thẻ SV: 102200077

Lớp: 20TCLC_DT1

Hệ thống hỗ trợ đánh giá điểm hoạt động cộng đồng là một giải pháp công nghệ tiên tiến nhằm tự động hóa và cải thiện quy trình đánh giá sự tham gia của sinh viên vào các hoạt động ngoại khóa. Hệ thống này giúp giảm thiểu khối lượng công việc thủ công của quản trị viên, đảm bảo tính chính xác và minh bạch trong việc chấm điểm. Sinh viên có thể dễ dàng đăng ký tham gia các hoạt động thông qua hệ thống và nộp minh chứng bằng ảnh sau khi tham gia. Hệ thống sử dụng trí tuệ nhân tạo (AI) để nhận diện sinh viên trong ảnh, hỗ trợ quản trị viên duyệt và đánh giá minh chứng một cách nhanh chóng và hiệu quả. Kết quả nhận diện từ AI giúp tăng tốc độ xử lý và đảm bảo công bằng trong chấm điểm, tạo điều kiện thuận lợi cho cả sinh viên và nhà trường. Hệ thống không chỉ thúc đẩy sinh viên tham gia tích cực vào các hoạt động ngoại khóa mà còn tạo ra một môi trường học tập và rèn luyện toàn diện, phát triển kỹ năng mềm và tinh thần cộng đồng của sinh viên.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Phạm Quốc Tú

Số thẻ sinh viên: 102200077

Lớp: 20TCLC_DT1 Khoa: Công Nghệ Thông Tin Ngành: Công Nghệ Phần Mềm

1. Tên đề tài đồ án:

Hệ thống hỗ trợ đánh giá điểm hoạt động cộng đồng

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Không có

4. Nội dung các phần thuyết minh và tính toán:

Giới thiệu đề tài: Phần mở đầu của luận văn, giới thiệu về lý do thực hiện đề tài, đồng thời giới thiệu sơ lược về đề tài và mục tiêu phải đạt được.

Phần 1. Giới thiệu thực trạng và đề xuất giải pháp

Phần 2. Tổng quan về hệ thống. Giới thiệu về kiến trúc microservice, công nghệ sử dụng.

Phần 3. Triển khai thực nghiệm, đánh giá các kết quả đạt được, kết luận và hướng phát triển.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

Không có

6. Tên người hướng dẫn: TS. Trương Ngọc Châu

7. Ngày giao nhiệm vụ đồ án: 10/04/2024.

8. Ngày hoàn thành đồ án: 15/06/2024.

Đà Nẵng, ngày tháng năm 2024

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến thầy hướng dẫn Trương Ngọc Châu đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề tài này. Em cũng xin cảm ơn khoa đã dìu dắt, dạy dỗ và truyền đạt kiến thức, kinh nghiệm quý báu của mình trong suốt quá trình em học tập và nghiên cứu tại trường.

Đề tài “Hệ thống hỗ trợ đánh giá điểm hoạt động cộng đồng” là một giải pháp công nghệ tiên tiến nhằm tự động hóa và cải thiện quy trình đánh giá sự tham gia của sinh viên vào các hoạt động ngoại khóa. Hệ thống này giúp giảm thiểu khối lượng công việc thủ công của quản trị viên, đảm bảo tốt hơn về tính chính xác và minh bạch trong việc chấm điểm.

Trong quá trình thực hiện đề tài, em đã áp dụng nhiều phương pháp nghiên cứu và kỹ thuật khác nhau để xây dựng và triển khai hệ thống. Báo cáo này sẽ trình bày chi tiết về quá trình nghiên cứu, thiết kế và phát triển hệ thống.

Trong quá trình thực hiện chắc hẳn sẽ không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự cảm thông, góp ý và tận tình chỉ bảo của quý thầy cô để đề tài của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn!

Phạm Quốc Tú

CAM ĐOAN

Em xin cam đoan:

1. Nội dung trong đồ án này là do em thực hiện dưới sự hướng dẫn trực tiếp của TS. Trương Ngọc Châu.
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, em xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Phạm Quốc Tú

MỤC LỤC

Tóm tắt	
Nhiệm vụ đồ án	
Lời nói đầu và cảm ơn	i
Lời cam đoan liên chính học thuật	ii
Mục lục	iii
Danh sách các bảng biểu, hình vẽ và sơ đồ	v
Danh sách các ký hiệu, chữ viết tắt	vii
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	2
1.1. Giới thiệu chung	2
1.1.1. Mục đích đề tài	2
1.1.2. Tầm quan trọng của hệ thống đánh giá hoạt động cộng đồng	2
1.2. Cơ sở lý thuyết và nền tảng	2
1.2.1. Khái niệm về hoạt động cộng đồng	2
1.2.2. Tầm quan trọng của hoạt động cộng đồng, ngoại khóa trong giáo dục	2
1.2.3. Các phương pháp đánh giá hoạt động cộng đồng hiện tại	2
1.2.4. Kiến trúc hệ thống và các thành phần chính	3
1.2.5. Công nghệ sử dụng AI, nhận diện khuôn mặt	3
1.2.6. Quy trình hoạt động của hệ thống	3
1.3. Phạm vi nghiên cứu	3
1.3.1. Đối tượng áp dụng	3
1.3.2. Tác động tổng quan	4
CHƯƠNG 2: TỔNG QUAN LÝ THUYẾT	5
2.1. Giới thiệu về kiến trúc Microservice	5
2.1.1. Khái niệm Microservice	5
2.1.2. Ưu điểm và nhược điểm	5
2.2. Giới thiệu về hệ thống nhận diện khuôn mặt	7
2.2.1. Phương pháp và công nghệ sử dụng	7
2.2.2. Quy trình xây dựng hệ thống nhận diện khuôn mặt	8
2.2.3. Tích hợp hệ thống nhận diện khuôn mặt	8
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG	10
3.1. Sơ đồ Usecase tổng quan	10
3.1.1. Sơ đồ usecase phân rã của User	12
3.1.2. Sơ đồ Usecase phân rã của Admin	14

3.2. Yêu cầu hệ thống.....	15
3.2.1. Yêu cầu chức năng.....	15
3.2.2. Yêu cầu phi chức năng	16
3.3. Kiến trúc hệ thống	16
3.3.1. Kiến trúc tổng thể.....	16
3.3.2. Kiến trúc chi tiết.....	17
3.4. Thiết kế cơ sở dữ liệu.....	21
3.4.1. Mô hình ERD (Entity-Relationship Diagram)	21
3.4.2. Bảng dữ liệu	21
CHƯƠNG 4: MÔ TẢ CHỨC NĂNG.....	24
4.1. Chức năng của Admin.....	24
4.1.1. Quản lý User.....	24
4.1.2. Quản lý sự kiện	28
4.1.3. Quản lý tham gia sự kiện.....	32
4.2. Chức năng của User.....	34
4.2.1. Đăng ký tài khoản	34
4.2.2. Chỉnh sửa thông tin tài khoản	35
4.2.3. Đăng ký tham gia sự kiện	36
4.2.4. Xây dựng mô hình nhận diện khuôn mặt.....	37
4.2.5. Xem lịch sử tham gia	38
4.2.6. Nộp minh chứng tham gia sự kiện.....	39
CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ.....	40
5.1. Kết quả.....	40
5.1.1. Giao diện website admin	40
5.1.2. Giao diện website users	47
5.2. Đánh giá	52
KẾT LUẬN	53
TÀI LIỆU THAM KHẢO	

DANH SÁCH CÁC BẢNG BIỂU, HÌNH VẼ VÀ SƠ ĐỒ

BẢNG 3.1 Cấu trúc table Users

BẢNG 3.2 Cấu trúc table Events

BẢNG 3.3 Cấu trúc table Event Images

BẢNG 3.4: Cấu trúc table Registrations

BẢNG 3.5: Cấu trúc table Notification

BẢNG 4.1: Mô tả chức năng xem danh sách người dùng

BẢNG 4.2: Mô tả chức năng thêm người dùng

BẢNG 4.3: Mô tả chức năng xóa người dùng

BẢNG 4.4: Mô tả chức năng chỉnh sửa thông tin người dùng

BẢNG 4.5: Mô tả chức năng xem danh sách sự kiện

BẢNG 4.6: Mô tả chức năng thêm sự kiện

BẢNG 4.7: Mô tả chức năng chỉnh sửa thông tin sự kiện

BẢNG 4.8: Mô tả chức năng xóa sự kiện

BẢNG 4.9: Mô tả chức năng xem danh sách người tham gia

BẢNG 4.10: Mô tả chức năng xét duyệt đơn nộp minh chứng

BẢNG 4.11: Mô tả chức năng đăng ký tài khoản

BẢNG 4.12: Mô tả chức năng chỉnh sửa thông tin tài khoản

BẢNG 4.13: Mô tả chức năng đăng ký tham gia sự kiện

BẢNG 4.14: Mô tả chức năng xây dựng mô hình nhận diện khuôn mặt

BẢNG 4.15: Mô tả chức năng xem lịch sử tham gia

BẢNG 4.16: Mô tả chức năng nộp minh chứng tham gia sự kiện

HÌNH 2.1: Mô hình microservice

HÌNH 3.1: Sơ đồ usecase User

HÌNH 3.2: Sơ đồ usecase Admin

HÌNH 3.3: Sơ đồ usecase xem Event

HÌNH 3.4: Sơ đồ usecase Đăng kí/Hủy tham gia sự kiện

HÌNH 3.5: Sơ đồ usecase Thay đổi thông tin cá nhân

HÌNH 3.6: Sơ đồ usecase xác thực danh tính

HÌNH 3.7: Sơ đồ usecase quản lý người dùng

HÌNH 3.8: Sơ đồ usecase quản lý sự kiện

HÌNH 4.1: Sơ đồ chức năng xem danh sách người dùng

HÌNH 4.2: Sơ đồ chức năng thêm người dùng

HÌNH 4.3: Sơ đồ chức năng xóa người dùng

HÌNH 4.4: Sơ đồ chức năng chỉnh sửa thông tin người dùng

HÌNH 4.5: Sơ đồ chức năng xem danh sách sự kiện

HÌNH 4.6: Sơ đồ chức năng thêm sự kiện

HÌNH 4.7: Sơ đồ chức năng chỉnh sửa thông tin sự kiện

HÌNH 4.8: Sơ đồ chức năng xóa sự kiện

HÌNH 4.9: Sơ đồ chức năng xem danh sách người tham gia

HÌNH 4.10: Sơ đồ chức năng xét duyệt đơn nộp minh chứng

HÌNH 4.11: Sơ đồ chức năng đăng ký tài khoản

HÌNH 4.12: Sơ đồ chức năng chỉnh sửa thông tin tài khoản

HÌNH 4.13: Sơ đồ chức năng đăng ký tham gia sự kiện

HÌNH 4.14: Sơ đồ chức năng xây dựng mô hình nhận diện khuôn mặt

HÌNH 4.15: Sơ đồ chức năng xem lịch sử tham gia

HÌNH 4.16: Sơ đồ chức năng nộp minh chứng tham gia sự kiện

HÌNH 5.1: Chức năng quản lý Event

HÌNH 5.2: Chức năng xem chi tiết Event

HÌNH 5.3: Chức năng thêm sự kiện

HÌNH 5.4: Chỉnh sửa thông tin sự kiện

HÌNH 5.5: Danh sách người tham gia sự kiện

HÌNH 5.6: Danh sách người có điểm tham gia sự kiện nhiều nhất

HÌNH 5.7: Duyệt người dùng nộp minh chứng tham gia sự kiện

HÌNH 5.8: Quản lý người dùng

HÌNH 5.9: Giao diện thêm người dùng mới

HÌNH 5.10: Giao diện sửa đổi thông tin người dùng

HÌNH 5.11: Giao diện danh sách sự kiện của người dùng

HÌNH 5.12: Giao diện chi tiết sự kiện của người dùng

HÌNH 5.13: Giao diện người dùng nộp ảnh để nhận diện khuôn mặt

HÌNH 5.14: Quản lý hoạt động tham gia các sự kiện của người dùng

HÌNH 5.15: Đăng kí tham gia sự kiện

HÌNH 5.16: Giao diện người dùng quản lý trang cá nhân

HÌNH 5.17: Giao diện thông báo của người dùng

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ	Viết tắt	Diễn giải
AI	Artificial Intelligence	Trí tuệ nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu siêu văn bản
CSS	Cascading Style Sheets	Tập tin định kiểu theo tầng
MUI	Material-UI	Thư viện giao diện người dùng dựa trên Material Design
REST	Representational State Transfer	Chuyển đổi trạng thái biểu diễn
JPA	Java Persistence API	Giao diện lập trình lưu trữ Java
ORM	Object-Relational Mapping	Kỹ thuật ánh xạ đối tượng-quan hệ
ERD	Entity-Relationship Diagram	Sơ đồ thực thể-quan hệ
JWT	JSON Web Token	Mã thông báo web JSON

MỞ ĐẦU

Trong bối cảnh giáo dục hiện đại, việc đánh giá và ghi nhận sự tham gia của sinh viên vào các hoạt động cộng đồng và ngoại khóa là vô cùng quan trọng. Những hoạt động này không chỉ giúp phát triển kỹ năng mềm mà còn tạo cơ hội cho sinh viên thể hiện bản thân, xây dựng tinh thần đồng đội và gắn kết với cộng đồng. Tuy nhiên, việc đánh giá hoạt động này thường gặp nhiều khó khăn do khối lượng công việc lớn và yêu cầu tính chính xác, minh bạch cao.

Đề tài "Hệ thống hỗ trợ đánh giá điểm hoạt động cộng đồng" nhằm mục tiêu phát triển một hệ thống tự động hóa quá trình đánh giá và chấm điểm các hoạt động cộng đồng và ngoại khóa của sinh viên. Hệ thống này sẽ sử dụng các công nghệ hiện đại như trí tuệ nhân tạo (AI) và nhận diện khuôn mặt để đảm bảo tính chính xác và hiệu quả trong việc nhận diện và đánh giá sinh viên tham gia hoạt động.

Phạm vi nghiên cứu của đề tài bao gồm việc phát triển hệ thống phần mềm có khả năng:

- Đăng ký và quản lý các hoạt động cộng đồng và ngoại khóa.
- Nhận diện sinh viên thông qua ảnh chụp bằng công nghệ AI.
- Chấm điểm và đánh giá minh chứng sinh viên nộp lên hệ thống. Đối tượng nghiên cứu là sinh viên các trường đại học và các quản trị viên phụ trách việc quản lý và đánh giá hoạt động cộng đồng của sinh viên.

Để thực hiện đề tài, các phương pháp nghiên cứu và phát triển hệ thống sẽ bao gồm:

- Nghiên cứu và phân tích các hệ thống đánh giá hoạt động hiện có.
- Thiết kế hệ thống sử dụng mô hình microservice để đảm bảo tính linh hoạt và khả năng mở rộng.
- Sử dụng các công nghệ Java Spring Boot cho backend và Flask API cho xử lý AI.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu chung

1.1.1. Mục đích đề tài

Mục đích: Đề tài này nhằm phát triển một hệ thống đánh giá hoạt động cộng đồng và ngoại khóa của sinh viên, sử dụng công nghệ hiện đại như AI và nhận diện khuôn mặt để nâng cao hiệu quả và tính chính xác của việc ghi nhận và đánh giá hoạt động của sinh viên.

1.1.2. Tầm quan trọng của hệ thống đánh giá hoạt động cộng đồng

Hệ thống này giúp nhà trường và sinh viên có cái nhìn rõ ràng hơn về sự tham gia và đóng góp của sinh viên vào các hoạt động cộng đồng, từ đó khuyến khích sinh viên tham gia tích cực hơn và giúp nhà trường có cơ sở dữ liệu chính xác để đánh giá và khen thưởng sinh viên.

1.2. Cơ sở lý thuyết và nền tảng

1.2.1. Khái niệm về hoạt động cộng đồng

Hoạt động cộng đồng: là những hoạt động mà sinh viên tham gia để hỗ trợ cộng đồng, từ thiện, bảo vệ môi trường, và các hoạt động xã hội khác.

Ngoại khóa: là các hoạt động ngoài chương trình học chính khóa, bao gồm thể thao, nghệ thuật, câu lạc bộ, và các hoạt động phát triển kỹ năng mềm.

1.2.2. Tầm quan trọng của hoạt động cộng đồng, ngoại khóa trong giáo dục

Phát triển toàn diện: Hoạt động ngoại khóa giúp sinh viên phát triển các kỹ năng mềm, như kỹ năng lãnh đạo, làm việc nhóm, giao tiếp, và quản lý thời gian.

Cơ hội trải nghiệm: Sinh viên có cơ hội trải nghiệm thực tế, áp dụng kiến thức học được vào thực tiễn.

Tăng cường sự kết nối: Giúp sinh viên xây dựng mạng lưới quan hệ và kết nối với cộng đồng.

1.2.3. Các phương pháp đánh giá hoạt động cộng đồng hiện tại

Phương pháp thủ công: Sử dụng giấy tờ, báo cáo bằng văn bản, và các hình thức ghi nhận thủ công.

Phương pháp hiện đại: Sử dụng các công cụ kỹ thuật số, phần mềm quản lý, và hệ thống nhận diện tự động để ghi nhận và đánh giá hoạt động của sinh viên.

1.2.4. Kiến trúc hệ thống và các thành phần chính

Kiến trúc hệ thống: Hệ thống bao gồm các thành phần chính như máy chủ, cơ sở dữ liệu, module nhận diện khuôn mặt, giao diện người dùng, và các thành phần phụ trợ khác.

Các thành phần chính:

- Module nhận diện khuôn mặt: Để nhận diện và theo dõi sinh viên tham gia các hoạt động.
- Cơ sở dữ liệu: Để lưu trữ thông tin về hoạt động của sinh viên.
- Giao diện người dùng: Để sinh viên và nhà trường có thể truy cập và quản lý thông tin.

1.2.5. Công nghệ sử dụng AI, nhận diện khuôn mặt

AI: Sử dụng trí tuệ nhân tạo để phân tích và đánh giá dữ liệu.

Nhận diện khuôn mặt: Sử dụng công nghệ nhận diện khuôn mặt để xác định và theo dõi sinh viên.

1.2.6. Quy trình hoạt động của hệ thống

Bước 1: Sinh viên tham gia hoạt động và được nhận diện bằng công nghệ nhận diện khuôn mặt.

Bước 2: Thông tin về hoạt động của sinh viên được ghi nhận và lưu trữ trong cơ sở dữ liệu.

Bước 3: Hệ thống phân tích và đánh giá dữ liệu, cung cấp báo cáo cho nhà trường và sinh viên.

1.3. Phạm vi nghiên cứu

1.3.1. Đối tượng áp dụng

Đối với sinh viên: Giúp sinh viên theo dõi và đánh giá sự tham gia của mình vào các hoạt động, từ đó cải thiện kỹ năng và hồ sơ cá nhân.

Đối với nhà trường: Cung cấp công cụ để quản lý và đánh giá hoạt động của sinh viên một cách hiệu quả và chính xác.

1.3.2. Tác động tổng quan

Tăng cường chất lượng giáo dục: Giúp nâng cao chất lượng giáo dục thông qua việc khuyến khích sinh viên tham gia tích cực vào các hoạt động cộng đồng và ngoại khóa.

Xây dựng môi trường học tập tích cực: Tạo ra một môi trường học tập tích cực và năng động, khuyến khích sự phát triển toàn diện của sinh viên.

CHƯƠNG 2: TỔNG QUAN LÝ THUYẾT

2.1. Giới thiệu về kiến trúc Microservice

2.1.1. Khái niệm Microservice

Không có định nghĩa phổ quát về thuật ngữ microservice. Định nghĩa đơn giản nhất của microservice, còn được gọi là kiến trúc microservice (microservice architecture), là một kiểu kiến trúc ứng dụng sử dụng các dịch vụ được liên kết lỏng lẻo. Các dịch vụ này có thể được phát triển, triển khai và duy trì độc lập. Chúng hoạt động với tốc độ nhanh hơn và đáng tin cậy hơn nhiều so với các ứng dụng nguyên khối, phức tạp truyền thống. Sử dụng kiến trúc microservice, một tổ chức có kích thước bất kỳ có thể phát triển các ngăn xếp công nghệ phù hợp với khả năng của họ.



Hình 2.1: Mô hình microservice

2.1.2. Ưu điểm và nhược điểm

- Ưu điểm:
 - ❖ *Cải thiện khả năng mở rộng và năng suất:*

Các nhóm lớn thường phải làm việc cùng nhau trong các dự án phức tạp. Với microservice, các dự án có thể được chia thành các đơn vị nhỏ hơn, độc lập. Điều này có nghĩa là các nhóm có thể hành động độc lập về logic nghiệp vụ,

giúp giảm thiểu sự phối hợp và nỗ lực. Trên hết, các nhóm chịu trách nhiệm cho từng microservice có thể đưa ra quyết định công nghệ của riêng họ tùy thuộc vào nhu cầu của họ. Ví dụ: cấu trúc bên trong của mỗi đơn vị hoặc vùng chứa không quan trọng miễn là giao diện hoạt động chính xác. Do đó, bất kỳ ngôn ngữ lập trình nào cũng có thể được sử dụng để viết microservice, vì vậy nhóm chịu trách nhiệm có thể chọn ngôn ngữ tốt nhất cho đồng đội của họ.

❖ *Tích hợp tốt với các hệ thống cũ:*

Hệ thống nguyên khối rất khó để duy trì. Nhiều hệ thống cũ có cấu trúc kém, thử nghiệm kém hoặc phụ thuộc vào các công nghệ lỗi thời. May mắn thay, microservice có thể làm việc cùng với các hệ thống cũ để cải thiện mã và thay thế các phần cũ của hệ thống. Tích hợp rất dễ dàng và có thể giải quyết nhiều vấn đề khiến các hệ thống nguyên khối trở thành quá khứ.

❖ *Phát triển bền vững:*

Kiến trúc microservice tạo ra các hệ thống có thể duy trì trong thời gian dài do các bộ phận khác nhau có thể thay thế. Điều này có nghĩa là một microservice có thể dễ dàng được viết lại mà không ảnh hưởng đến toàn bộ hệ thống. Miễn là sự phụ thuộc giữa các microservice được quản lý phù hợp, các thay đổi có thể dễ dàng được thực hiện để tối ưu hóa nhu cầu và hiệu suất của nhóm.

❖ *Chức năng chéo:*

Microservice là tốt nhất cho các nhóm phân phối. Nếu bạn có các nhóm trên toàn thế giới hoặc các bộ phận khác nhau, microservice cấp các quyền tự do cần thiết và linh hoạt để hoạt động tự chủ. Các quyết định kỹ thuật có thể được đưa ra nhanh chóng, tích hợp với các dịch vụ khác trong nháy mắt. Chức năng chéo chưa bao giờ được dễ dàng hơn.

- Nhược điểm:

- *Triển khai đòi hỏi nhiều nỗ lực hơn:*

Hoạt động của một hệ thống microservice thường đòi hỏi nhiều nỗ lực hơn, vì có nhiều đơn vị có thể triển khai hơn mà mỗi đơn vị phải được triển khai và giám sát. Thay đổi giao diện phải được thực hiện để có thể triển khai độc lập các microservice riêng lẻ.

- *Kiểm thử phải độc lập:*

Vì tất cả các microservice phải được kiểm thử cùng nhau, một microservice có thể làm gián đoạn thử nghiệm và ngăn chặn việc triển khai các microservice khác. Có nhiều giao diện để kiểm tra và kiểm tra phải độc lập cho cả hai mặt của giao diện.

- *Khó thay đổi nhiều microservice:*

Những thay đổi ảnh hưởng đến nhiều microservice có thể khó thực hiện hơn. Trong một hệ thống microservice, các thay đổi đòi hỏi một số triển khai phối hợp.

2.2. Giới thiệu về hệ thống nhận diện khuôn mặt

Hệ thống nhận diện khuôn mặt là một phần quan trọng của hệ thống hỗ trợ đánh giá điểm hoạt động cộng đồng. Công nghệ nhận diện khuôn mặt giúp tự động xác định danh tính của sinh viên từ hình ảnh nộp lên, từ đó đảm bảo tính chính xác và minh bạch trong việc chấm điểm các hoạt động ngoại khóa.

2.2.1. Phương pháp và công nghệ sử dụng

Phương pháp nhận diện khuôn mặt:

Hệ thống sử dụng thư viện **face_recognition** của Python để nhận diện và xác định khuôn mặt. Thư viện này dựa trên các kỹ thuật học sâu (Deep Learning) để trích xuất các đặc trưng khuôn mặt và so sánh với các mẫu đã lưu trữ.

face_recognition cung cấp một cách tiếp cận đơn giản và hiệu quả để nhận diện khuôn mặt, thích hợp cho các ứng dụng yêu cầu độ chính xác cao và tốc độ xử lý nhanh. Điều này giúp cho việc triển khai và tích hợp vào hệ thống trở nên đơn giản hơn, đồng thời đảm bảo sự tin cậy trong việc xác định danh tính của người sử dụng từ các hình ảnh được cung cấp.

Công nghệ sử dụng:

- *Python:* Ngôn ngữ lập trình chính được sử dụng để phát triển hệ thống.

- *Thư viện face_recognition*: Thư viện face_recognition là một thư viện Python phổ biến được sử dụng chủ yếu để nhận diện khuôn mặt. Thư viện này được xây dựng trên nền tảng của Dlib, một thư viện mạnh mẽ về xử lý ảnh và thị giác máy tính. face_recognition cung cấp các công cụ đơn giản để nhận diện và nhận dạng khuôn mặt trong ảnh và video bằng cách sử dụng mô hình đã được huấn luyện trước, giúp cho việc phát triển ứng dụng liên quan đến xác thực và nhận diện khuôn mặt trở nên dễ dàng hơn đáng kể. Thư viện này được sử dụng rộng rãi trong các dự án AI và computer vision do tính đơn giản và hiệu quả của nó trong việc xử lý và nhận diện các đặc trưng của khuôn mặt.
- *OpenCV*: Thư viện hỗ trợ xử lý hình ảnh.
- *Dlib*: Thư viện hỗ trợ các thuật toán máy học và nhận diện khuôn mặt.

2.2.2. Quy trình xây dựng hệ thống nhận diện khuôn mặt

Thu thập dữ liệu:

Sinh viên tiến hành xây dựng model nhận diện khuôn mặt bằng cách nộp các ảnh chụp chân dung của mình lên hệ thống.

Xây dựng Model AI riêng cho từng sinh viên:

Hệ thống sử dụng **face_recognition** để trích xuất các đặc trưng khuôn mặt và xây dựng model nhận diện riêng cho từng sinh viên.

2.2.3. Tích hợp hệ thống nhận diện khuôn mặt

Kiến trúc tích hợp:

Để đảm bảo tính linh hoạt và khả năng mở rộng của hệ thống, hệ thống được xây dựng từ một server API riêng sử dụng Flask. Server API này chịu trách nhiệm xử lý các yêu cầu nhận diện khuôn mặt từ các mô hình khác và trả về kết quả nhận diện. Server API giúp các thành phần khác trong hệ thống dễ dàng tương tác với model AI mà không cần tích hợp trực tiếp.

Quy trình tích hợp:

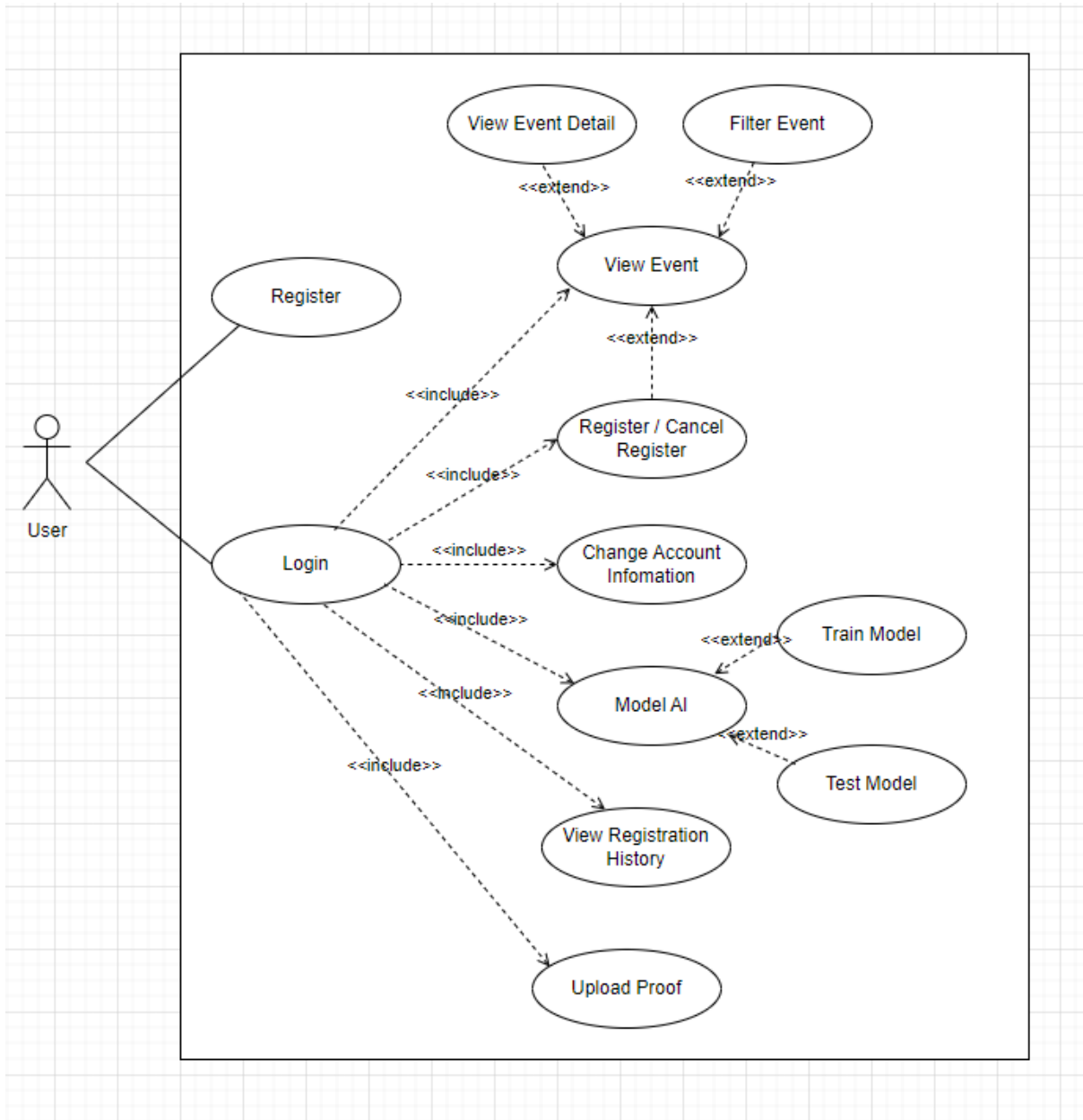
- **Sinh viên nộp hình ảnh minh chứng**: Sinh viên tham gia hoạt động và nộp hình ảnh minh chứng thông qua giao diện người dùng của hệ thống.

- **Gửi yêu cầu đến server API:** Hình ảnh được gửi đến server API nhận diện khuôn mặt thông qua một yêu cầu HTTP POST. Yêu cầu này bao gồm hình ảnh và thông tin nhận dạng sinh viên (user_id).
- **Xử lý yêu cầu trên server API:** Server API nhận hình ảnh và thông tin sinh viên, sau đó tải model nhận diện khuôn mặt tương ứng của sinh viên đó và thực hiện nhận diện khuôn mặt.
- **Trả về kết quả nhận diện:** Server API trả về kết quả nhận diện (bao gồm thông tin nhận dạng hoặc thông báo không nhận diện được) thông qua phản hồi HTTP.
- **Lưu trữ và hiển thị kết quả:** Kết quả nhận diện được lưu trữ trong cơ sở dữ liệu và hiển thị trên giao diện người dùng để quản trị viên có thể xem xét và chấm điểm.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

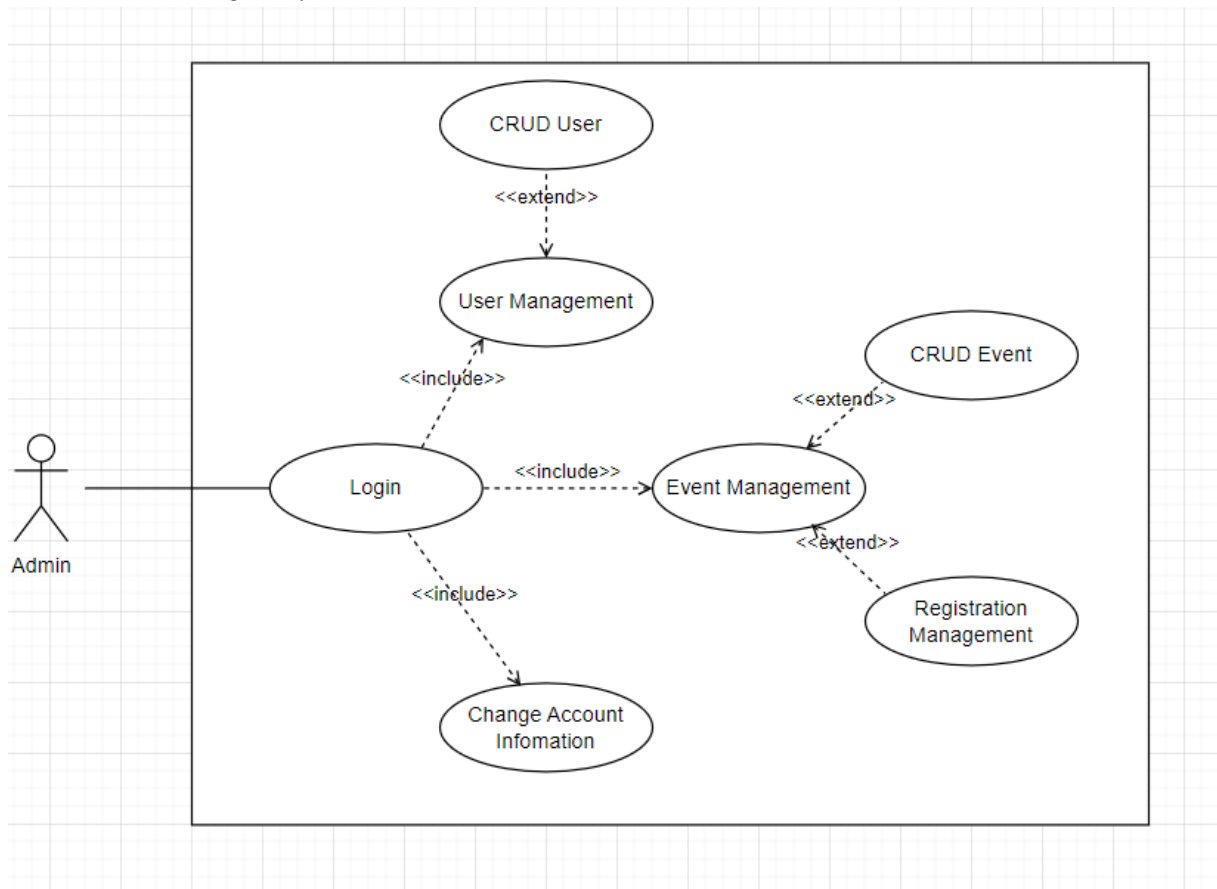
3.1. Sơ đồ Usecase tổng quan

- User:



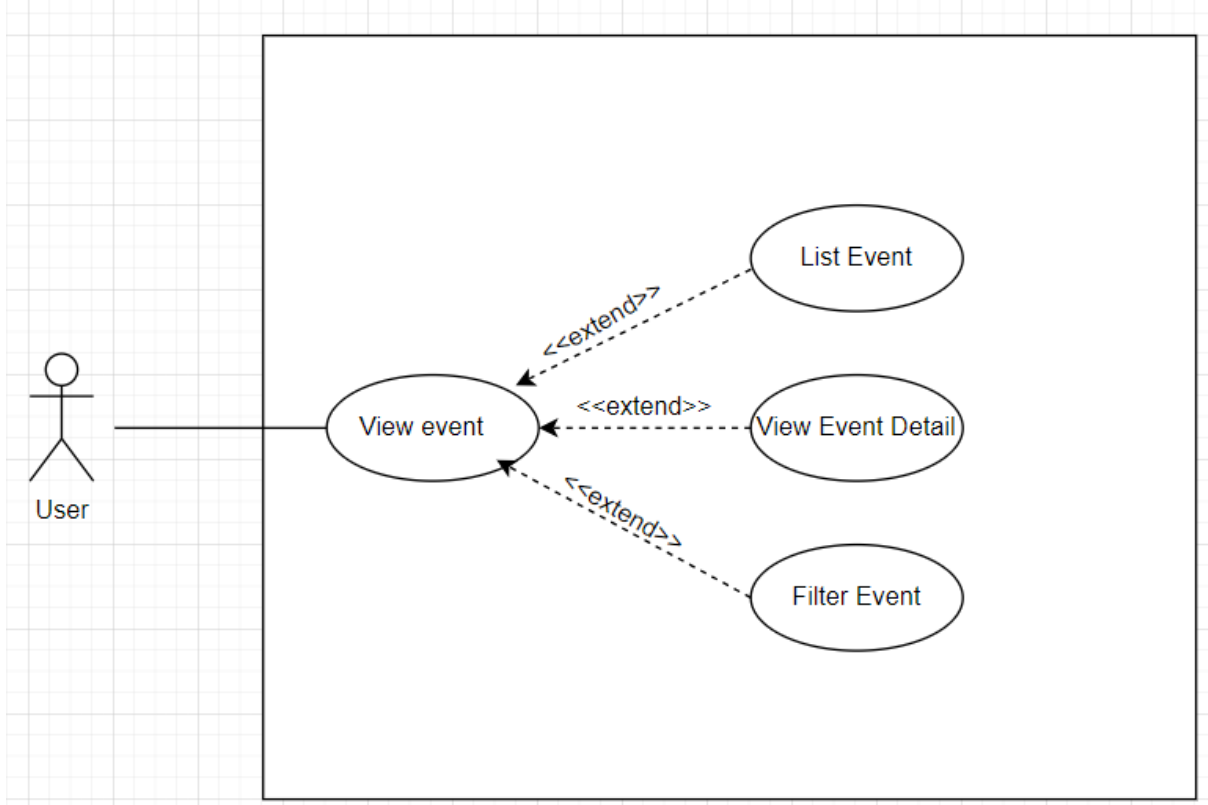
Hình 3.1 Sơ đồ Usecase User

- Admin:

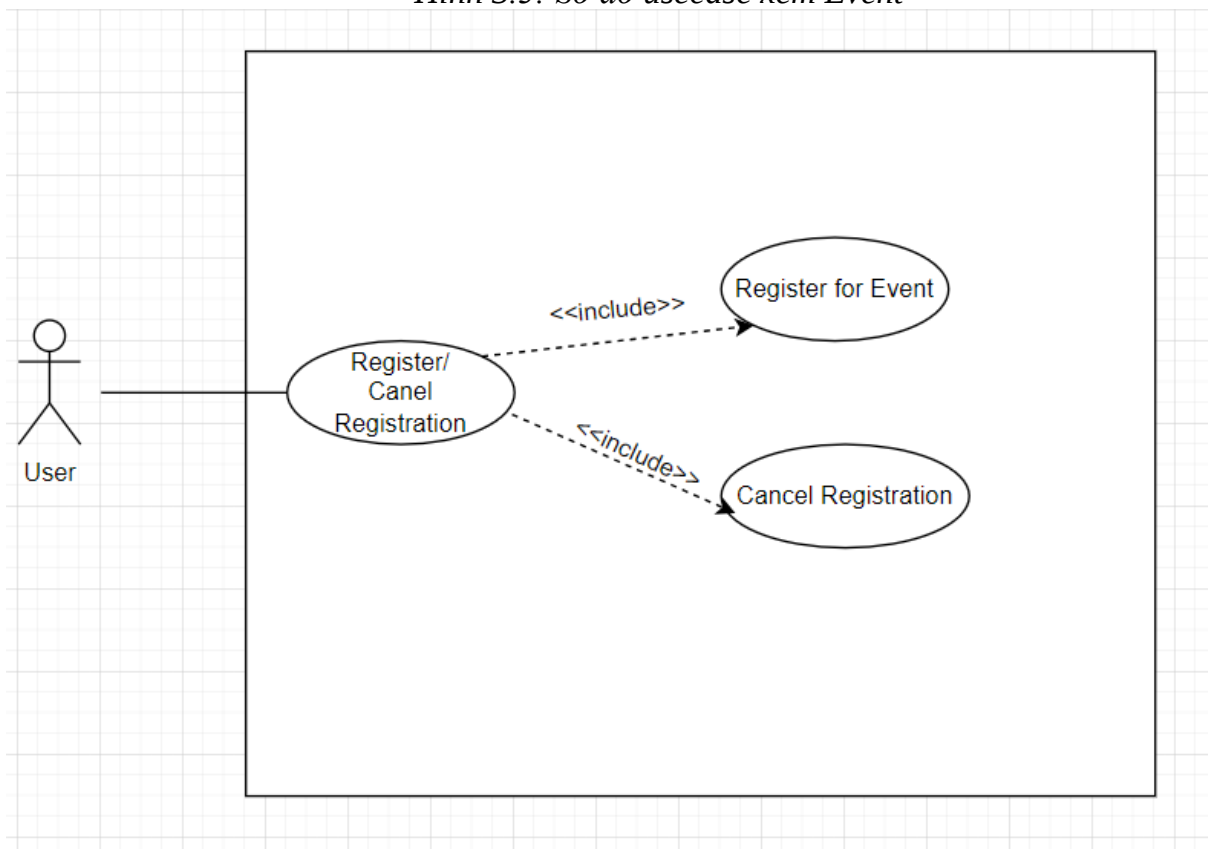


Hình 3.2 Sơ đồ Usecase Admin

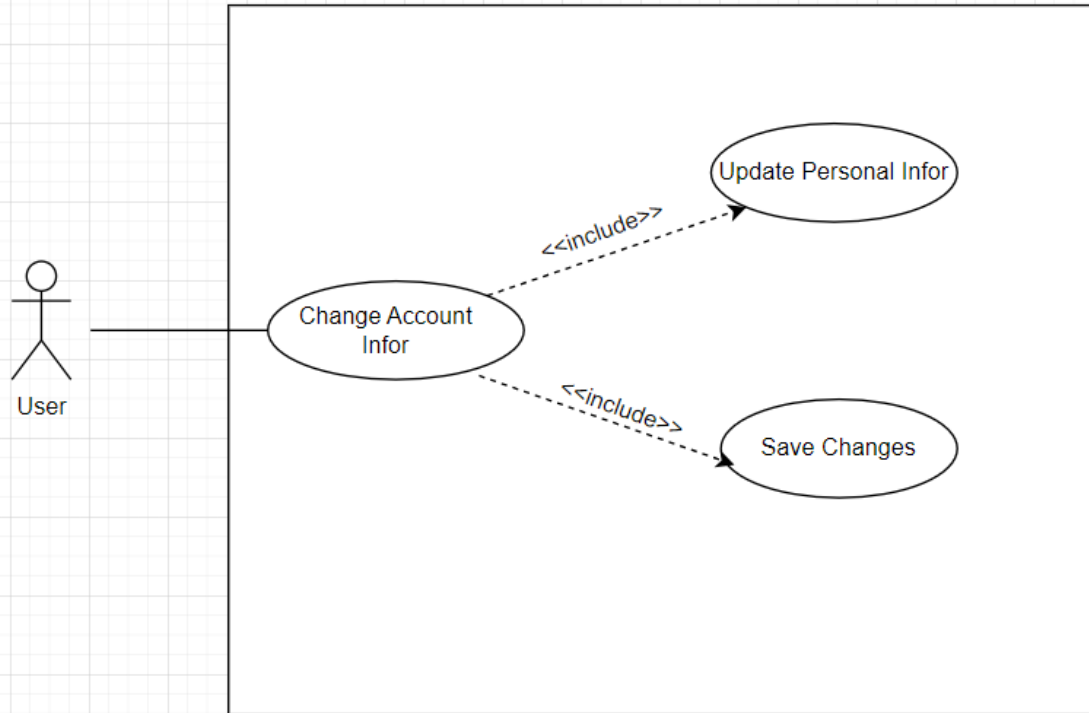
3.1.1. Sơ đồ usecase phân rã của User



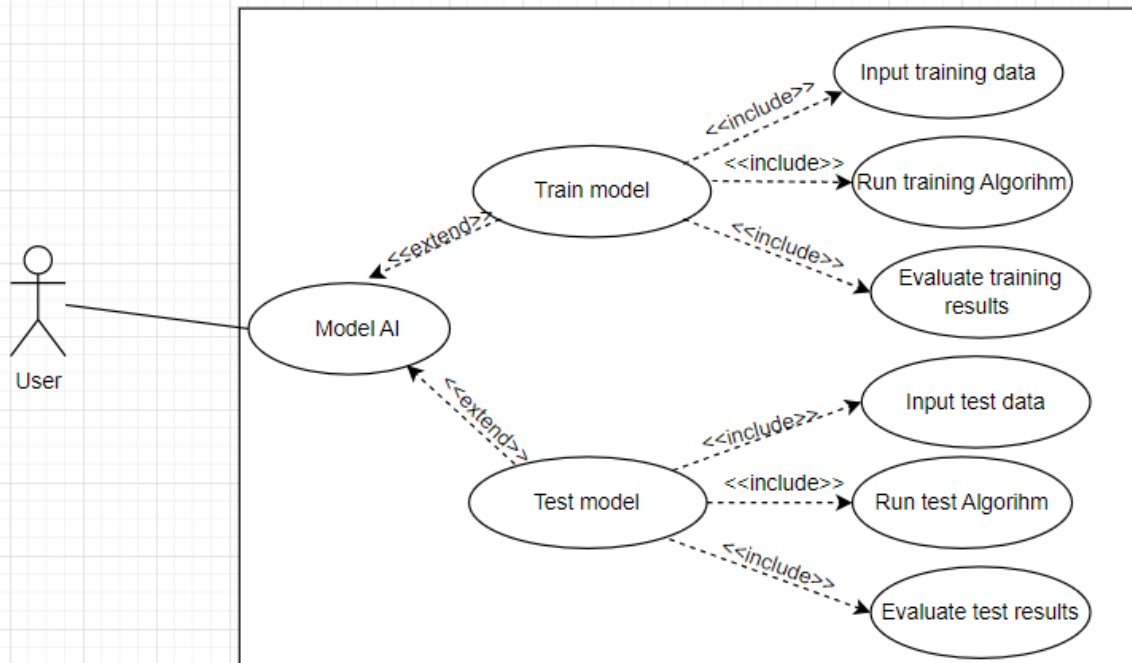
Hình 3.3: Sơ đồ usecase xem Event



Hình 3.4: Sơ đồ usecase Đăng kí/Hủy tham gia sự kiện

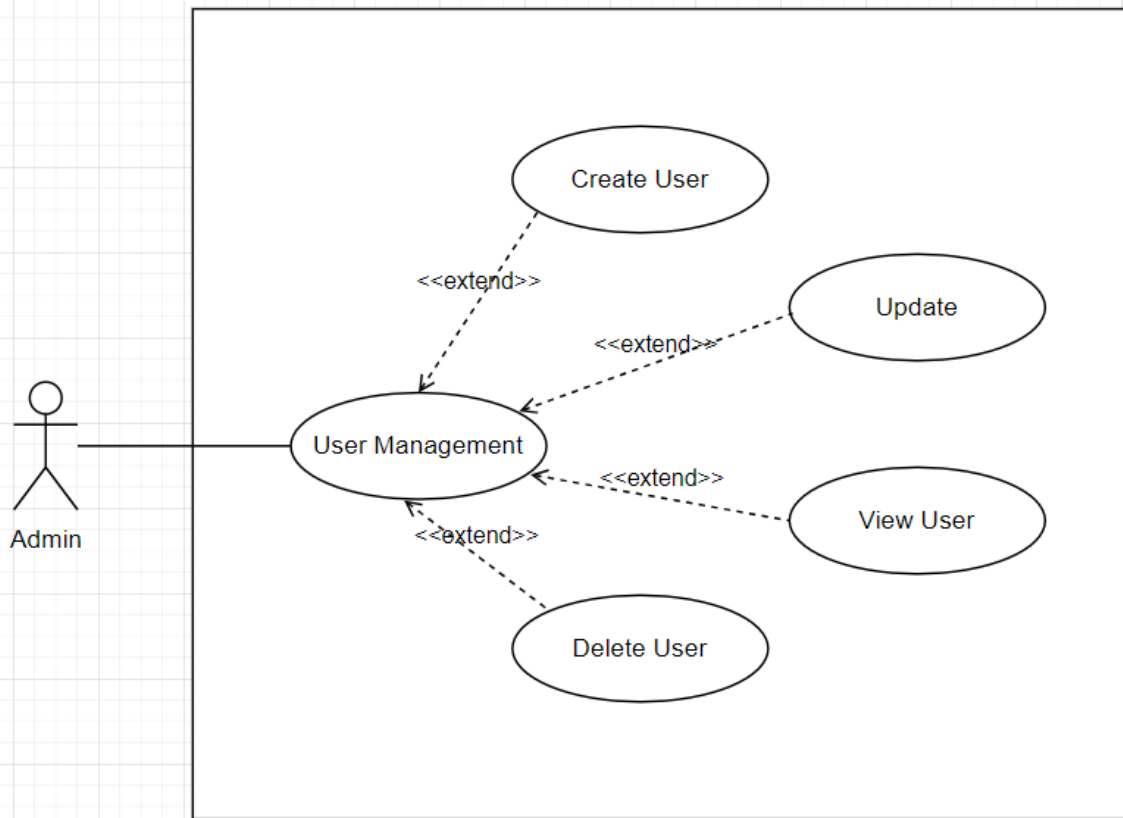


Hình 3.5: Sơ đồ usecase Thay đổi thông tin cá nhân

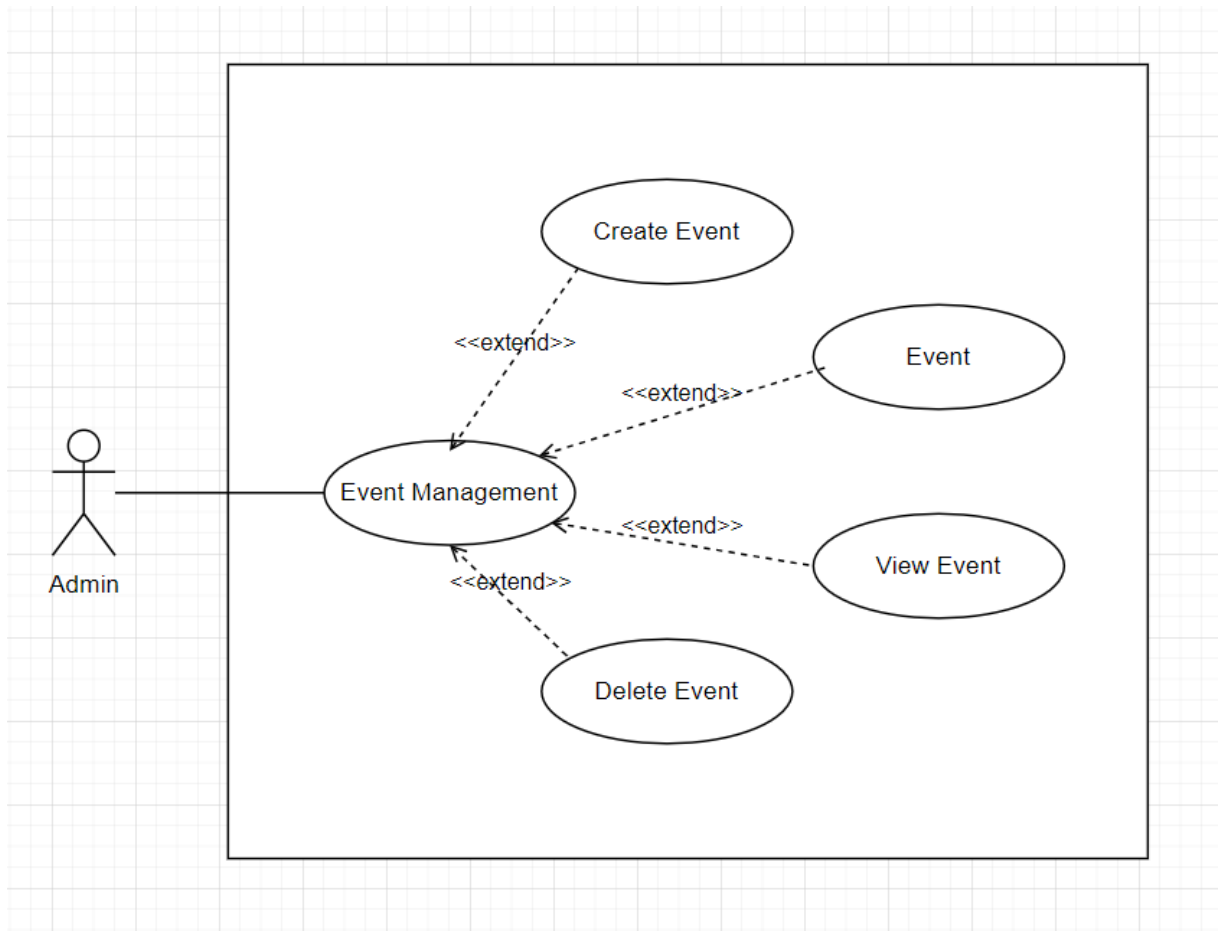


Hình 3.6: Sơ đồ usecase xác thực danh tính

3.1.2. Sơ đồ Usecase phân rã của Admin



Hình 3.7: Sơ đồ usecase quản lý người dùng



Hình 3.8: Sơ đồ usecase quản lý sự kiện

3.2. Yêu cầu hệ thống

3.2.1. Yêu cầu chức năng

- Đăng ký và quản lý tài khoản sinh viên: Sinh viên có thể đăng ký tài khoản mới và quản lý thông tin cá nhân.
- Đăng ký hoạt động: Sinh viên có thể đăng ký tham gia các hoạt động ngoại khóa thông qua hệ thống.
- Nộp minh chứng tham gia: Sinh viên có thể nộp minh chứng tham gia hoạt động bằng ảnh.
- Duyệt và chấm điểm minh chứng: Quản trị viên có thể duyệt và chấm điểm minh chứng sinh viên nộp lên.
- Tự động nhận diện sinh viên trong ảnh: Hệ thống sử dụng AI để nhận diện sinh viên trong ảnh minh chứng.

- Quản lý hoạt động: Quản trị viên có thể tạo, sửa, xóa và quản lý các hoạt động ngoại khóa.
- Báo cáo và thống kê: Hệ thống cung cấp các báo cáo và thống kê về điểm số và hoạt động của sinh viên.

3.2.2. Yêu cầu phi chức năng

- Hiệu suất: Hệ thống phải đảm bảo xử lý nhanh chóng và chính xác.
- Bảo mật: Hệ thống phải bảo mật thông tin cá nhân của sinh viên.
- Khả năng mở rộng: Hệ thống phải có khả năng mở rộng để xử lý số lượng lớn người dùng và dữ liệu.

3.3. Kiến trúc hệ thống

3.3.1. Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình Client-Server với kiến trúc microservice, bao gồm các thành phần chính:

Client:

- Giao diện người dùng, bao gồm giao diện web và ứng dụng di động.
- Các client này tương tác với backend thông qua các API RESTful.

Server:

- *Java Spring Boot*: Xử lý các yêu cầu từ client, quản lý dữ liệu và thực hiện các nghiệp vụ logic chính.
- *Flask API (AI Service)*: Xử lý nhận diện khuôn mặt bằng AI. Flask API này hoạt động như một microservice riêng biệt, chịu trách nhiệm xử lý các yêu cầu liên quan đến AI từ các thành phần khác của hệ thống.

Database:

- Lưu trữ dữ liệu của hệ thống, bao gồm thông tin người dùng, hoạt động, và các kết quả nhận diện khuôn mặt.
- Sử dụng MySQL hoặc các hệ quản trị cơ sở dữ liệu quan hệ khác để lưu trữ dữ liệu.

Kiến trúc Microservice:

- *Spring Boot Service*: Quản lý các nghiệp vụ logic chính của hệ thống, bao gồm quản lý người dùng, hoạt động và các giao dịch dữ liệu khác.
- *Flask AI Service*: Một microservice độc lập chuyên xử lý các yêu cầu liên quan đến nhận diện khuôn mặt, bao gồm huấn luyện và dự đoán nhận diện khuôn mặt.
- *Communication*: Các microservice giao tiếp với nhau thông qua các API RESTful, đảm bảo tính linh hoạt và khả năng mở rộng của hệ thống.

Lợi ích của Kiến trúc Microservice:

- *Tính mô-đun*: Cho phép phát triển, triển khai và mở rộng từng thành phần của hệ thống một cách độc lập.
- *Khả năng mở rộng*: Dễ dàng mở rộng từng microservice mà không ảnh hưởng đến toàn bộ hệ thống.
- *Quản lý lỗi*: Giúp cô lập và quản lý lỗi một cách hiệu quả, đảm bảo rằng lỗi trong một microservice không làm ảnh hưởng đến các phần khác của hệ thống.
- *Tích hợp công nghệ*: Cho phép sử dụng các công nghệ phù hợp nhất cho từng microservice, như Java Spring Boot cho nghiệp vụ logic và Flask cho xử lý AI.

3.3.2. Kiến trúc chi tiết

3.3.2.1. Frontend (Client)

Frontend của hệ thống được xây dựng bằng các công nghệ web hiện đại, tập trung vào việc tạo ra một trải nghiệm người dùng mượt mà và thân thiện. Các công nghệ sử dụng bao gồm HTML, CSS, JavaScript và đặc biệt là framework ReactJS kết hợp với Vite để cải thiện hiệu suất phát triển và tốc độ tải trang. Ngoài ra, Material-UI (MUI) được sử dụng để tạo giao diện người dùng nhất quán và chuyên nghiệp.

- *HTML (HyperText Markup Language)*: Là ngôn ngữ đánh dấu tiêu chuẩn để tạo các trang web. HTML giúp cấu trúc nội dung của trang web thông qua các thẻ và thuộc tính.

- **CSS (Cascading Style Sheets):** Được sử dụng để định dạng và tạo kiểu cho các trang HTML. CSS giúp cải thiện giao diện người dùng bằng cách quản lý màu sắc, phông chữ, bố cục và các yếu tố thiết kế khác.
- **JavaScript:** Là ngôn ngữ lập trình phía client chủ yếu, cho phép tạo các tương tác động trên trang web. JavaScript được sử dụng để cập nhật nội dung, xử lý sự kiện người dùng và thực hiện các thao tác không đồng bộ với server.
- **ReactJS:** Là một thư viện JavaScript để xây dựng giao diện người dùng, phát triển bởi Facebook. ReactJS cho phép tạo ra các component có thể tái sử dụng, giúp quản lý trạng thái ứng dụng một cách hiệu quả. React cũng có một hệ sinh thái phong phú, bao gồm React Router cho việc điều hướng và Redux hoặc Context API để quản lý trạng thái toàn cục.
- **Vite:** Là một build tool hiện đại, được phát triển bởi Evan You (người tạo ra Vue.js). Vite cung cấp một môi trường phát triển nhanh chóng bằng cách sử dụng module ES và có khả năng build sản phẩm cuối cùng với hiệu suất cao.
- **MUI (Material-UI):** Là một thư viện component React, tuân theo các nguyên tắc thiết kế của Material Design do Google phát triển. MUI cung cấp các component giao diện đẹp mắt và dễ sử dụng, giúp tăng tốc quá trình phát triển giao diện người dùng.

Các bước chính trong phát triển Frontend:

1. **Thiết kế giao diện người dùng:** Xây dựng các wireframe và prototype để xác định giao diện và trải nghiệm người dùng.
2. **Tạo cấu trúc dự án:** Khởi tạo dự án ReactJS bằng Vite, cấu trúc thư mục và thiết lập các gói cần thiết.
3. **Phát triển các component:** Xây dựng các component React để tạo nên các phần tử giao diện như form đăng nhập, danh sách hoạt động, và trang quản lý minh chứng. Sử dụng các component của MUI để đảm bảo tính nhất quán và chuyên nghiệp.
4. **Tùy chỉnh giao diện với MUI:** Sử dụng hệ thống chủ đề (theme) của MUI để tùy chỉnh màu sắc, phông chữ và các yếu tố giao diện khác. Áp dụng các phong cách (styles) bằng cách sử dụng makeStyles hoặc styled-components.

5. *Tích hợp API*: Sử dụng Axios hoặc Fetch API để gọi các dịch vụ REST từ backend, xử lý dữ liệu và hiển thị lên giao diện.
6. *Kiểm thử*: Sử dụng các công cụ như Jest và React Testing Library để viết các bài kiểm thử cho các component và logic ứng dụng.
7. *Tối ưu hóa*: Cải thiện hiệu suất và tối ưu hóa trải nghiệm người dùng bằng cách sử dụng các kỹ thuật như lazy loading, code splitting và caching.

3.3.2.2. Backend (Server)

Backend của hệ thống được phát triển bằng Java Spring Boot và Flask. Java Spring Boot được sử dụng để xây dựng các ứng dụng web và dịch vụ RESTful chính, trong khi Flask được sử dụng để tạo ra một server API riêng cho việc xử lý nhận diện khuôn mặt bằng AI.

Java Spring Boot: Java Spring Boot là một framework mạnh mẽ và phổ biến, giúp đơn giản hóa quá trình cấu hình và phát triển ứng dụng. Spring Boot cung cấp các tính năng như tự động cấu hình, các starter dependency và một server nhúng như Tomcat để triển khai ứng dụng một cách nhanh chóng.

- **Java:** Là ngôn ngữ lập trình hướng đối tượng, có tính ổn định cao và được sử dụng rộng rãi trong các ứng dụng doanh nghiệp.
- **Spring Boot:** Là một framework dựa trên Spring, giúp đơn giản hóa quá trình cấu hình và phát triển ứng dụng. Spring Boot cung cấp các tính năng như tự động cấu hình, các starter dependency, và một server nhúng như Tomcat để triển khai ứng dụng một cách nhanh chóng.

Các bước chính trong phát triển Backend:

1. **Khởi tạo dự án Spring Boot:** Sử dụng Spring Initializr hoặc IDE như IntelliJ IDEA để khởi tạo dự án, lựa chọn các dependency cần thiết như Spring Web, Spring Data JPA, và MySQL Driver.
2. **Thiết kế các lớp mô hình (Entity):** Xây dựng các lớp Java đại diện cho các bảng trong cơ sở dữ liệu, sử dụng các annotation như @Entity, @Table, @Id và các mối quan hệ như @OneToMany, @ManyToOne.

3. **Tạo các kho lưu trữ (Repository):** Sử dụng Spring Data JPA để tạo các repository, giúp tương tác với cơ sở dữ liệu một cách dễ dàng thông qua các phương thức CRUD.
4. **Xây dựng các dịch vụ (Service):** Tạo các lớp dịch vụ để xử lý logic nghiệp vụ, sử dụng các annotation như `@Service`, `@Transactional`.
5. **Phát triển các bộ điều khiển (Controller):** Tạo các RESTful API bằng cách sử dụng các annotation như `@RestController`, `@RequestMapping`, `@GetMapping`, `@PostMapping` để xử lý các yêu cầu từ phía client.
6. **Bảo mật:** Sử dụng Spring Security để quản lý xác thực và phân quyền, mã hóa mật khẩu bằng `BCryptPasswordEncoder`, và bảo vệ các endpoint bằng cách cấu hình các lớp bảo mật như JWT Token.

Flask cho API AI: Flask là một micro-framework nhẹ của Python, phù hợp cho việc phát triển các dịch vụ API. Flask được sử dụng để tạo ra một server API riêng biệt cho việc xử lý nhận diện khuôn mặt bằng AI, giúp tăng tính linh hoạt và khả năng mở rộng của hệ thống.

Các bước chính trong phát triển API AI bằng Flask:

1. *Khởi tạo dự án Flask:* Tạo một ứng dụng Flask cơ bản và cấu hình các route cần thiết để xử lý yêu cầu nhận diện khuôn mặt.
2. *Xây dựng các route và logic xử lý:* Tạo các route để nhận yêu cầu từ client, xử lý hình ảnh và thực hiện nhận diện khuôn mặt bằng thư viện **face_recognition**.
3. *Triển khai API:* Chạy ứng dụng Flask trên một server riêng biệt, đảm bảo rằng API có thể nhận và xử lý các yêu cầu HTTP từ hệ thống chính.

3.3.2.3. Database

Hệ thống sử dụng MySQL, một hệ quản trị cơ sở dữ liệu quan hệ phổ biến, để lưu trữ dữ liệu. MySQL cung cấp hiệu suất cao, độ tin cậy và dễ sử dụng, thích hợp cho các ứng dụng web.

- **MySQL:** Là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, hỗ trợ SQL (Structured Query Language) để quản lý và thao tác dữ liệu.

- **Spring Data JPA:** Là một phần của Spring Data, cung cấp một cách dễ dàng để tương tác với cơ sở dữ liệu bằng cách sử dụng các repository, giúp giảm thiểu mã boilerplate.

Các bước chính trong phát triển cơ sở dữ liệu:

- **Thiết kế cơ sở dữ liệu:** Xây dựng mô hình ERD (Entity-Relationship Diagram) để xác định các bảng và mối quan hệ giữa chúng.
- **Khởi tạo cơ sở dữ liệu:** Sử dụng các công cụ như MySQL Workbench hoặc dòng lệnh để tạo cơ sở dữ liệu và các bảng.
- **Cấu hình kết nối:** Cấu hình kết nối đến MySQL trong file application.properties của Spring Boot với các thông số như URL, username, password.
- **Quản lý schema:** Sử dụng Hibernate, một ORM (Object-Relational Mapping) framework tích hợp trong Spring Boot, để tự động tạo và cập nhật schema dựa trên các entity đã định nghĩa.
- **Tối ưu hóa:** Sử dụng các chỉ số (index), khóa ngoại (foreign key), và các kỹ thuật tối ưu hóa truy vấn để cải thiện hiệu suất của cơ sở dữ liệu.

3.4. Thiết kế cơ sở dữ liệu

3.4.1. Mô hình ERD (Entity-Relationship Diagram)

Hệ thống bao gồm các thực thể chính:

- **Người dùng (User):** Thông tin cá nhân, tài khoản đăng nhập.
- **Hoạt động (Event):** Thông tin về các hoạt động ngoại khóa.
- **Đăng ký (Registration):** Quản lý người tham gia các hoạt động.

3.4.2. Bảng dữ liệu

Bảng 3.1: Cấu trúc table Users

Thuộc tính	Kiểu dữ liệu	Mô tả
user_id	int	Khóa chính, tự động tăng, xác định duy nhất mỗi người dùng
username	varchar(255)	Tên đăng nhập của người dùng
password	varchar(255)	Mật khẩu của người dùng
role	varchar(255)	Vai trò của người dùng (admin, student, etc.)

Bảng 3.2: Cấu trúc table Events

Thuộc tính	Kiểu dữ liệu	Mô tả
event_id	int	Khóa chính, tự động tăng, xác định duy nhất mỗi sự kiện
event_name	varchar(255)	Tên của sự kiện
start_time	datetime	Thời gian bắt đầu sự kiện
end_time	datetime	Thời gian kết thúc sự kiện
location	varchar(255)	Địa điểm tổ chức sự kiện
point	int	Điểm thưởng cho sinh viên tham gia sự kiện
description	varchar(1024)	Mô tả chi tiết về sự kiện
max_attender	int	Số lượng tham dự viên tối đa
is_deleted	bit	Trạng thái xóa sự kiện (0: không bị xóa, 1: đã bị xóa)

Bảng 3.3: Cấu trúc table Registrations

Thuộc tính	Kiểu dữ liệu	Mô tả
registration_id	int	Khóa chính, tự động tăng, xác định duy nhất mỗi đăng ký
event_id	int	Khóa ngoại liên kết với bảng events, xác định sự kiện được đăng ký
user_id	int	Khóa ngoại liên kết với bảng users, xác định người dùng đăng ký
registration_time	datetime(6)	Thời gian đăng ký sự kiện
status	enum	Trạng thái đăng ký: ('attended', 'cancelled', 'registered', 'registered_but_not_attended')
image_url	varchar(255)	URL của hình ảnh liên quan đến đăng ký
isaipredicted	bit(1)	Trạng thái dự đoán AI (0: chưa dự đoán, 1: đã dự đoán)

Bảng 3.4: Cấu trúc table Event Images

Thuộc tính	Kiểu dữ liệu	Mô tả
event_id	int	Khóa ngoại liên kết với bảng events, xác định sự kiện chứa hình ảnh
image_url	varchar(255)	URL của hình ảnh sự kiện

Bảng 3.5: Cấu trúc table Notification

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int	Khóa chính, tự động tăng, xác định duy nhất mỗi thông báo
event_id	int	Khóa ngoại liên kết với bảng events, xác định sự kiện liên quan đến thông báo
title	varchar(255)	Tiêu đề của thông báo
description	text	Mô tả chi tiết về thông báo
avatar	varchar(255)	URL của hình ảnh đại diện cho thông báo
type	varchar(255)	Loại thông báo
created_at	timestamp	Thời gian tạo thông báo
is_un_read	tinyint(1)	Trạng thái chưa đọc (0: đã đọc, 1: chưa đọc)
user_id	int	Khóa ngoại liên kết với bảng users, xác định người dùng nhận thông báo

CHƯƠNG 4: MÔ TẢ CHỨC NĂNG

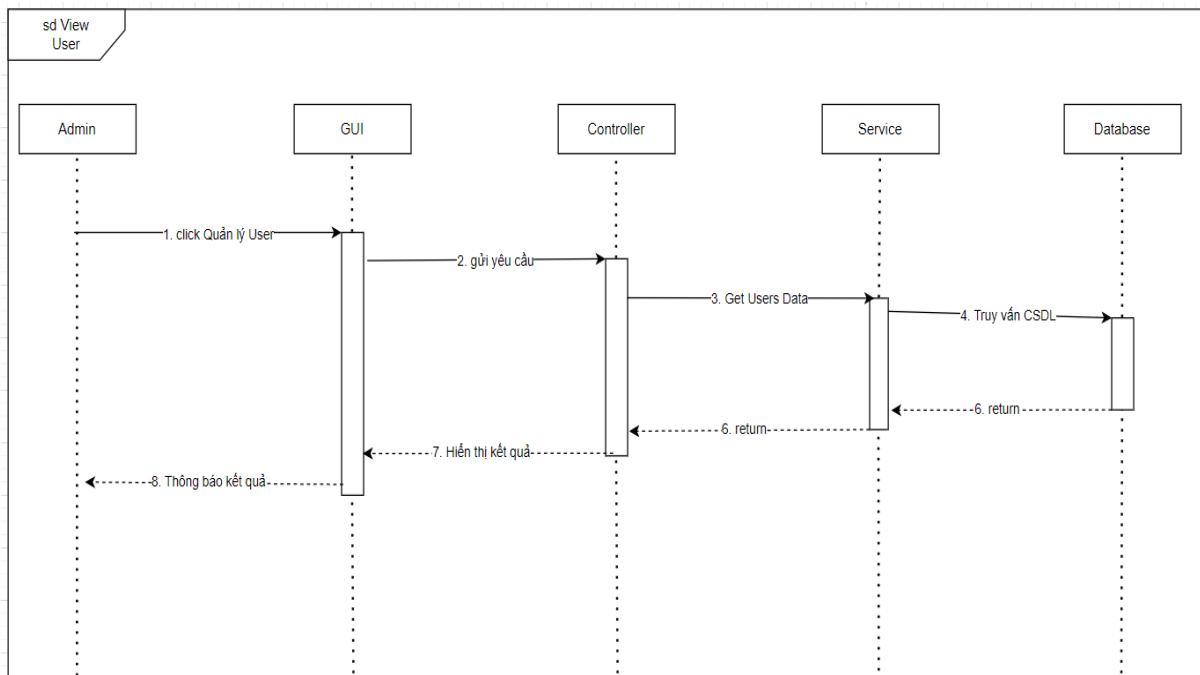
4.1. Chức năng của Admin

4.1.1. Quản lý User

4.1.1.1. Xem danh sách người dùng

Bảng 4.1: Mô tả chức năng xem danh sách người dùng

Tên chức năng	Xem danh sách người dùng
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý User
Kết quả	Hiển thị danh sách User

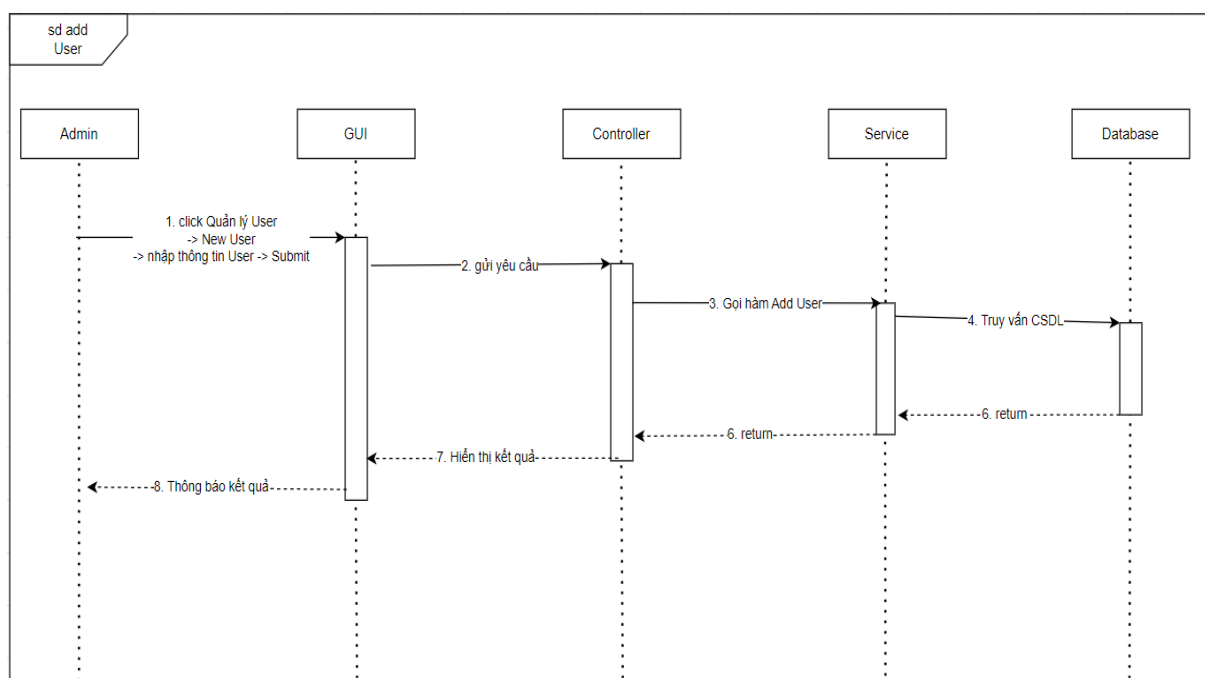


Hình 4.1: Sơ đồ chức năng xem danh sách người dùng

4.1.1.2. Thêm người dùng

Bảng 4.2: Mô tả chức năng thêm người dùng

Tên chức năng	Thêm người dùng
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý User → Nhấn vào button “ <i>New User</i> ” → Điền thông tin và nhấn “ <i>Submit</i> ”
Kết quả	Hiển thị danh sách User

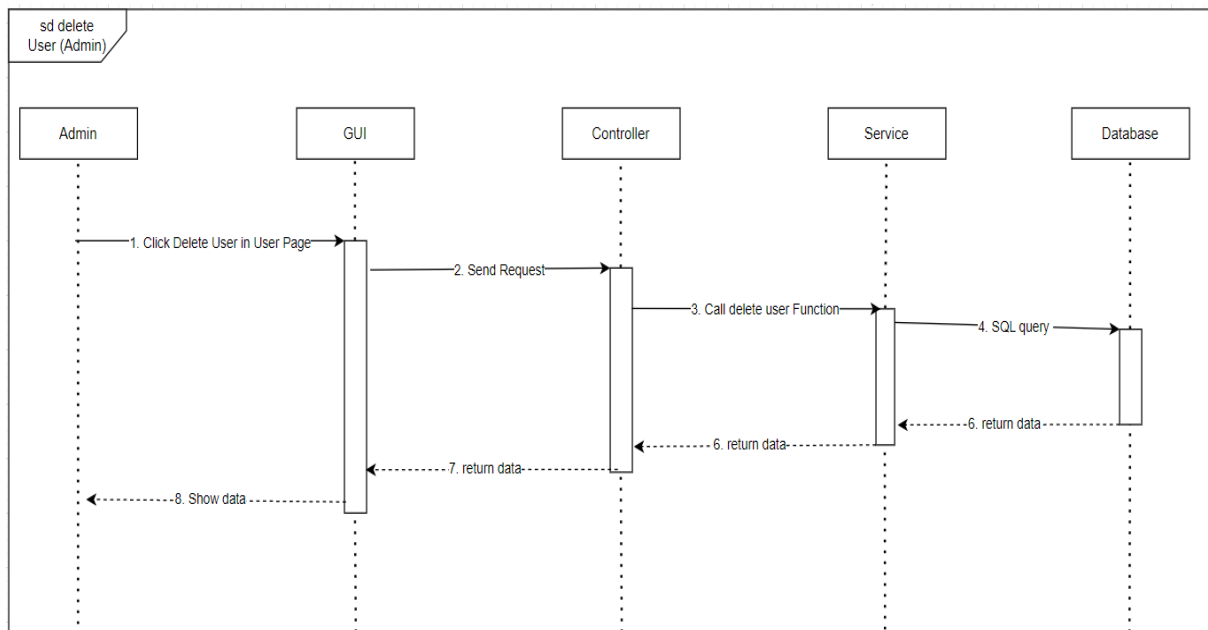


Hình 4.2: Sơ đồ chức năng thêm người dùng

4.1.1.3. Xóa người dùng

Bảng 4.3: Mô tả chức năng xóa người dùng

Tên chức năng	Xóa người dùng
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý User → Nhấn vào “ More option ” của người dùng cần xóa → Chọn “ Delete ”
Kết quả	Người dùng bị xóa (mềm)

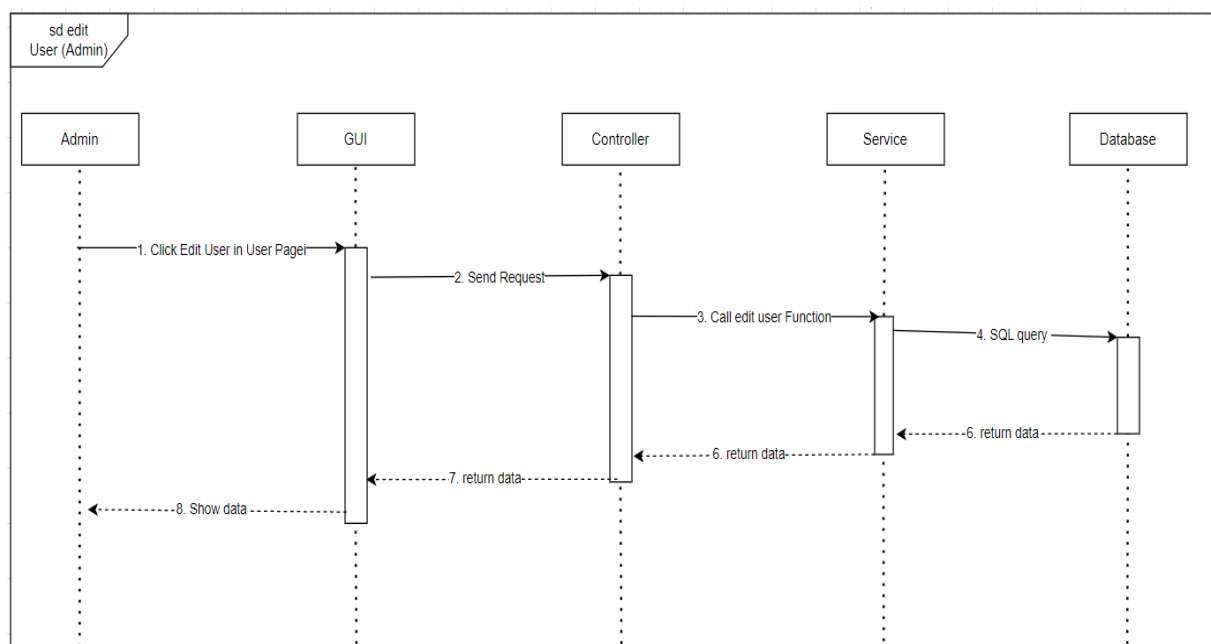


Hình 4.3: Sơ đồ chức năng xóa người dùng

4.1.1.4. Chỉnh sửa thông tin người dùng

Bảng 4.4: Mô tả chức năng chỉnh sửa thông tin người dùng

Tên chức năng	Chỉnh sửa thông tin người dùng.
Đối tượng sử dụng	Admin.
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin.
Quy trình nghiệp vụ	Login → Vào trang quản lý User → Nhấn vào “More option” của người dùng cần chỉnh sửa → Chọn “Edit” → Thay đổi thông tin → nhấn “Submit”
Kết quả	Thay đổi thông tin người dùng thành công.



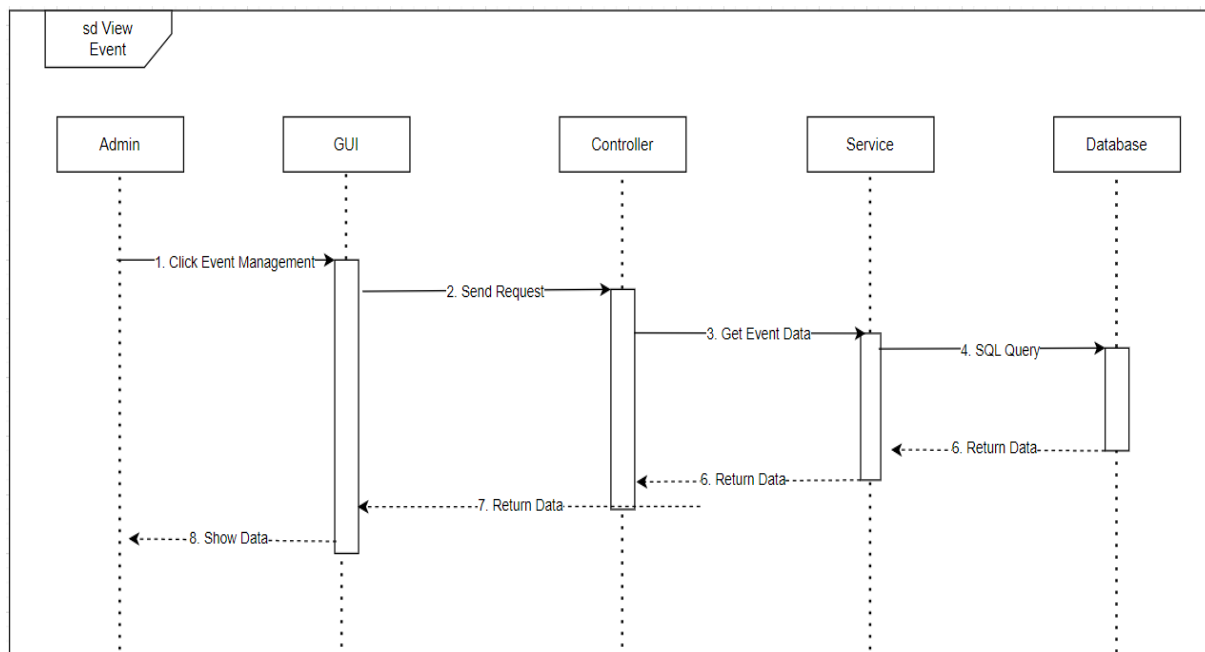
Hình 4.4: Sơ đồ chức năng chỉnh sửa thông tin người dùng

4.1.2. Quản lý sự kiện

4.1.2.1. Xem danh sách sự kiện

Bảng 4.5: Mô tả chức năng xem danh sách sự kiện

Tên chức năng	Xem danh sách sự kiện
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý Event
Kết quả	Hiển thị danh sách Event

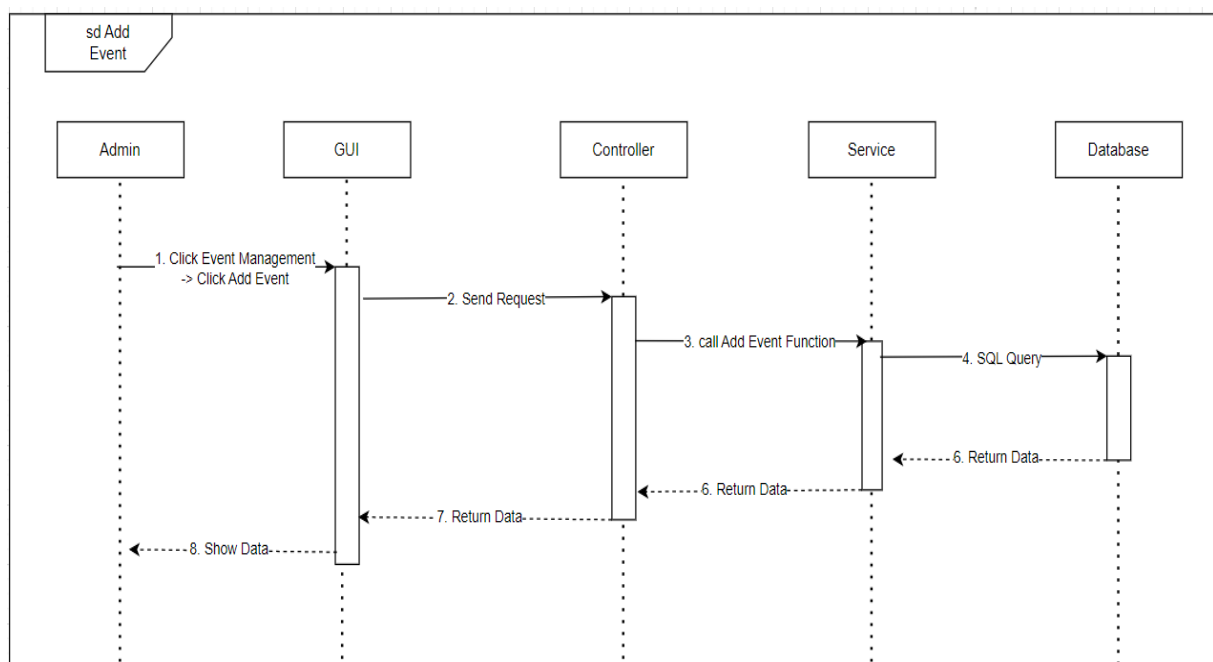


Hình 4.5: Sơ đồ chức năng xem danh sách sự kiện

4.1.2.2. Thêm sự kiện

Bảng 4.6: Mô tả chức năng thêm sự kiện

Tên chức năng	Thêm sự kiện
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý Event → Nhấn vào “ New Event ” → Nhập dữ liệu → Chọn “ Save ”
Kết quả	Thêm sự kiện thành công

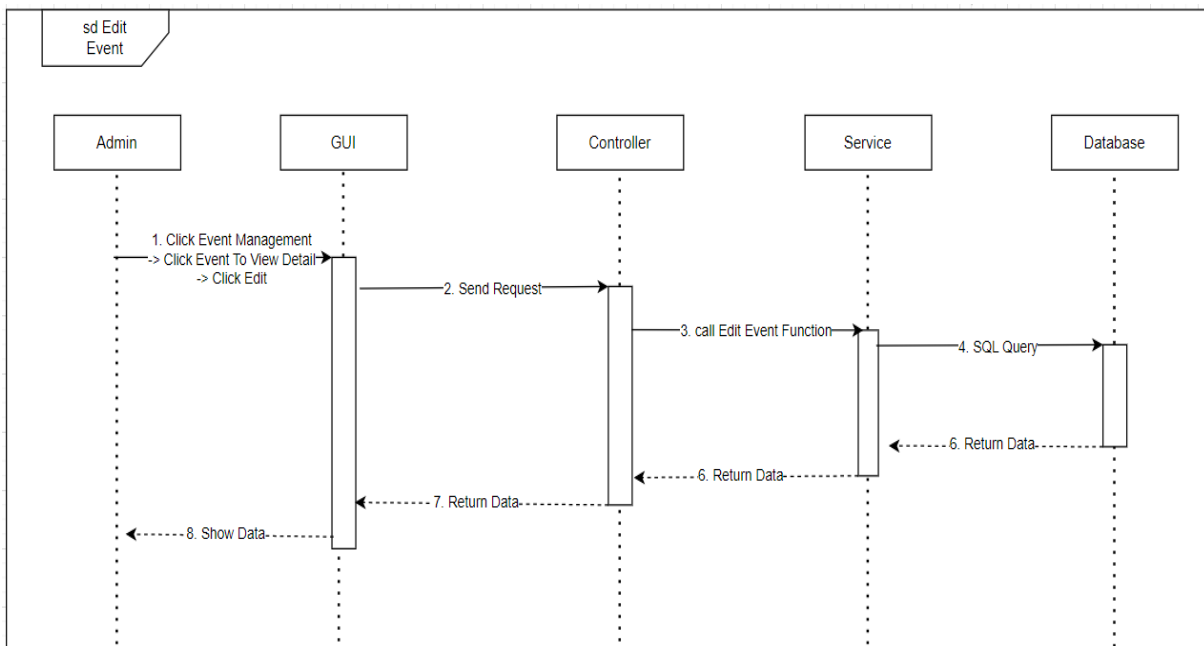


Hình 4.6: Sơ đồ chức năng thêm sự kiện

4.1.2.3. **Chỉnh sửa thông tin sự kiện**

Bảng 4.7: Mô tả chức năng chỉnh sửa thông tin sự kiện

Tên chức năng	Chỉnh sửa thông tin sự kiện
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý Event → Nhấn vào Event cần sửa → Chọn “ Edit Event ” → Sửa thông tin → “ Save ”
Kết quả	Thông tin sự kiện được sửa thành công

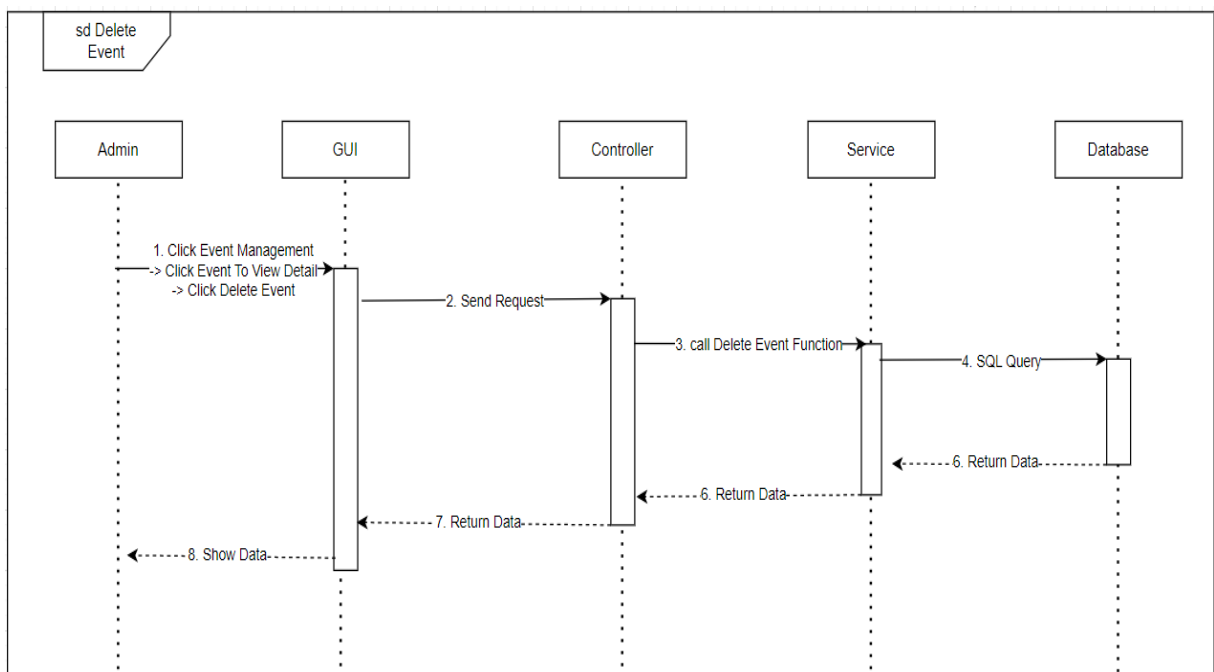


Hình 4.7: Sơ đồ chức năng chỉnh sửa thông tin sự kiện

4.1.2.4. Xóa sự kiện

Bảng 4.8: Mô tả chức năng xóa sự kiện

Tên chức năng	Xóa sự kiện
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý Event → Nhấn vào Event cần xóa → Chọn “Delete Event” → Xác nhận
Kết quả	Sự kiện bị xóa (mềm)



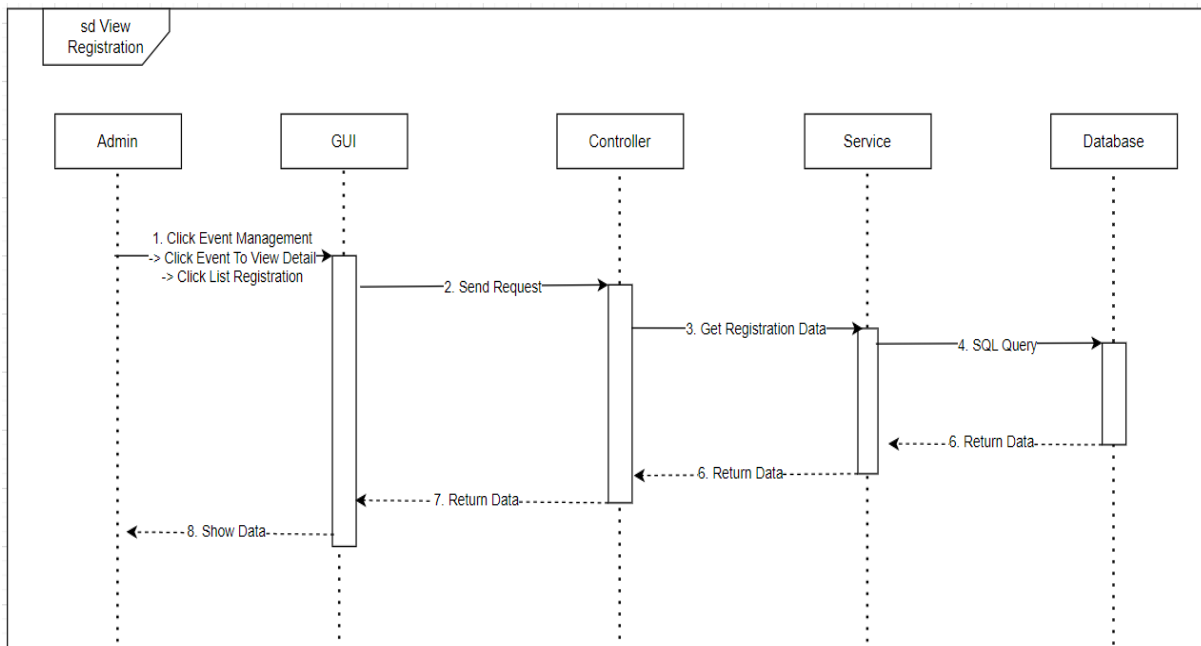
Hình 4.8: Sơ đồ chức năng xóa sự kiện

4.1.3. Quản lý tham gia sự kiện

4.1.3.1. Xem danh sách người tham gia

Bảng 4.9: Mô tả chức năng xem danh sách người tham gia

Tên chức năng	Xem danh sách người tham gia
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý Event → Nhấn vào Event cần xem → Nhấn vào “List Registration”
Kết quả	Hiển thị danh sách những người đăng ký tham gia sự kiện

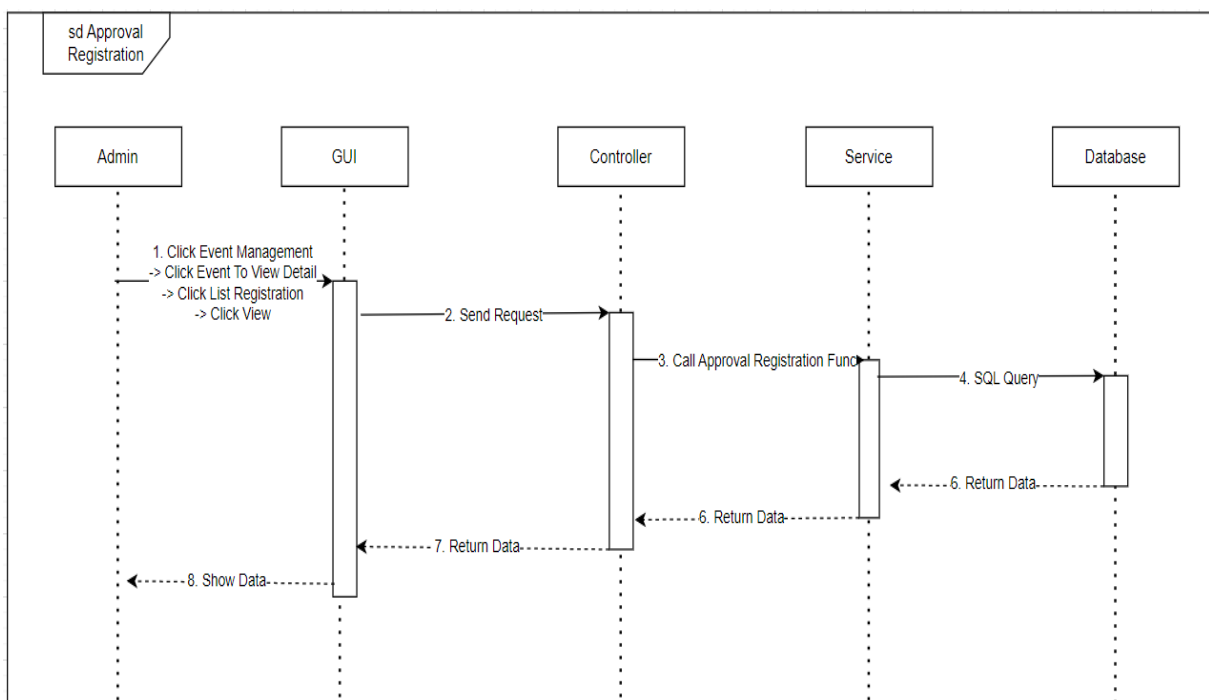


Hình 4.9: Sơ đồ chức năng xem danh sách người tham gia

4.1.3.2. Xét duyệt đơn nộp minh chứng

Bảng 4.10: Mô tả chức năng xét duyệt đơn nộp minh chứng

Tên chức năng	Xét duyệt đơn nộp minh chứng
Đối tượng sử dụng	Admin
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền Admin
Quy trình nghiệp vụ	Login → Vào trang quản lý Event → Nhấn vào Event cần xem → Nhấn vào “List Registration” → “Evidance” → “Accept” or “Reject”
Kết quả	Cập nhật trạng thái đơn và điểm cho người dùng



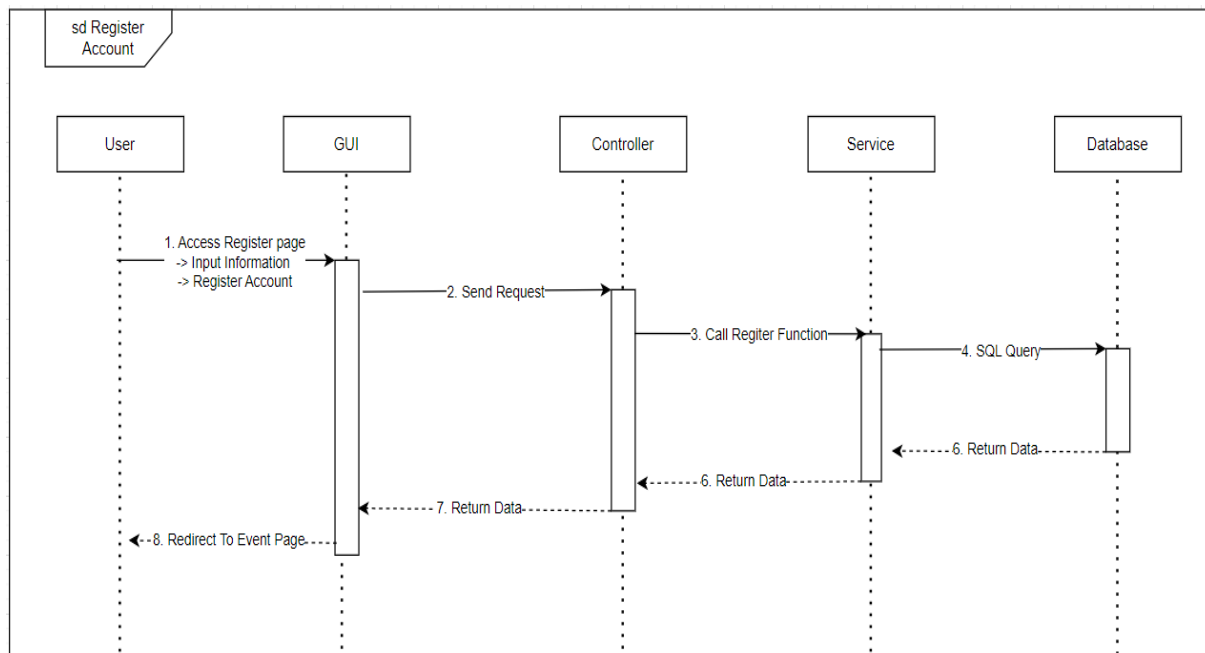
Hình 4.10: Sơ đồ chức năng xét duyệt đơn nộp minh chứng

4.2. Chức năng của User

4.2.1. Đăng ký tài khoản

Bảng 4.11: Mô tả chức năng đăng ký tài khoản

Tên chức năng	Đăng ký tài khoản
Đối tượng sử dụng	User
Điều kiện	Không có
Quy trình nghiệp vụ	Register → Nhập thông tin → Nhấn “ Register ”
Kết quả	Tài khoản được thêm vào hệ thống

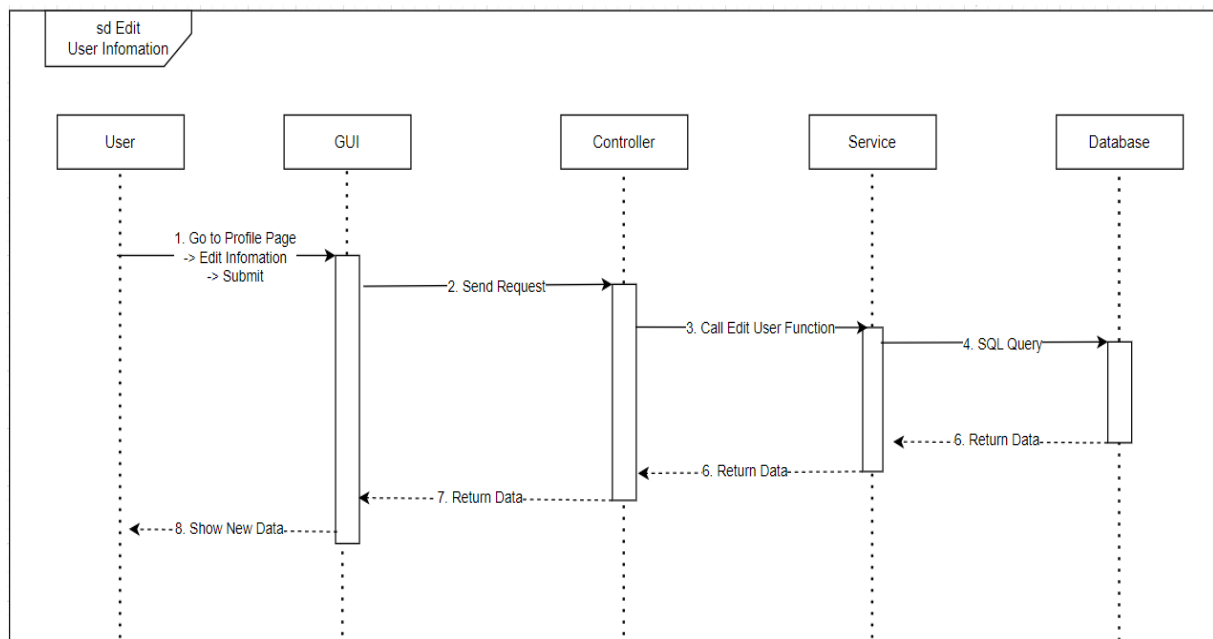


Hình 4.11: Sơ đồ chức năng đăng ký tài khoản

4.2.2. *Chỉnh sửa thông tin tài khoản*

Bảng 4.12: Mô tả chức năng chỉnh sửa thông tin tài khoản

Tên chức năng	Chỉnh sửa thông tin tài khoản
Đối tượng sử dụng	User
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền User
Quy trình nghiệp vụ	Login → Vào trang Profile → Chỉnh sửa thông tin → Chọn “Save”
Kết quả	Thông tin tài khoản được update

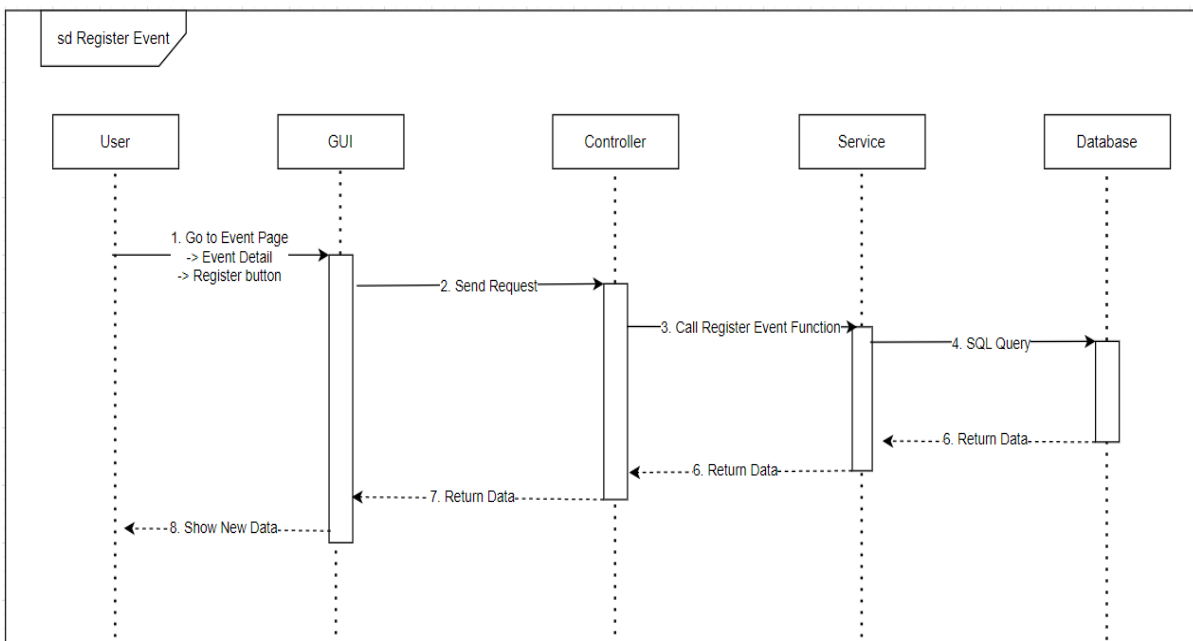


Hình 4.12: Sơ đồ chức năng chỉnh sửa thông tin tài khoản

4.2.3. Đăng ký tham gia sự kiện

Bảng 4.13: Mô tả chức năng đăng ký tham gia sự kiện

Tên chức năng	Đăng ký tham gia sự kiện
Đối tượng sử dụng	User
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền User
Quy trình nghiệp vụ	Login → Vào trang Event → Nhấn vào Event muốn đăng ký → Click Register Event
Kết quả	Người dùng đăng ký thành công

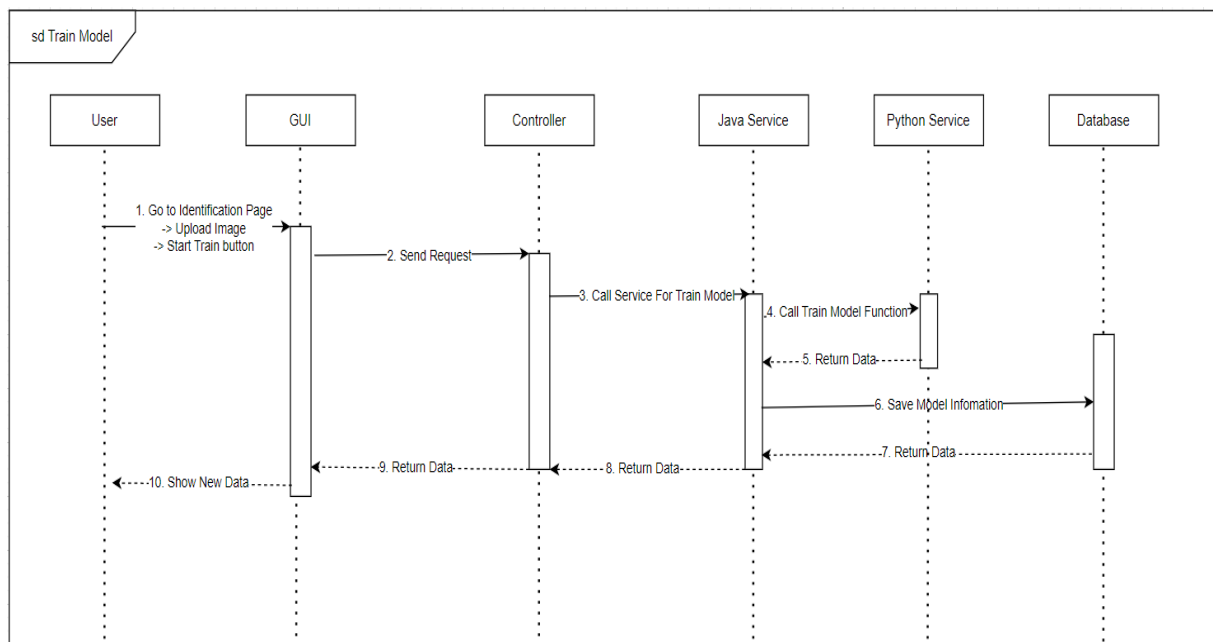


Hình 4.13: Sơ đồ chức năng đăng ký tham gia sự kiện

4.2.4. Xây dựng mô hình nhận diện khuôn mặt

Bảng 4.14: Mô tả chức năng xây dựng mô hình nhận diện khuôn mặt

Tên chức năng	Xây dựng mô hình nhận diện khuôn mặt
Đối tượng sử dụng	User
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền User
Quy trình nghiệp vụ	Login → Vào trang Identification → Upload hình ảnh chân dung → Chọn “ Start train model ”
Kết quả	Xây dựng mô hình thành công

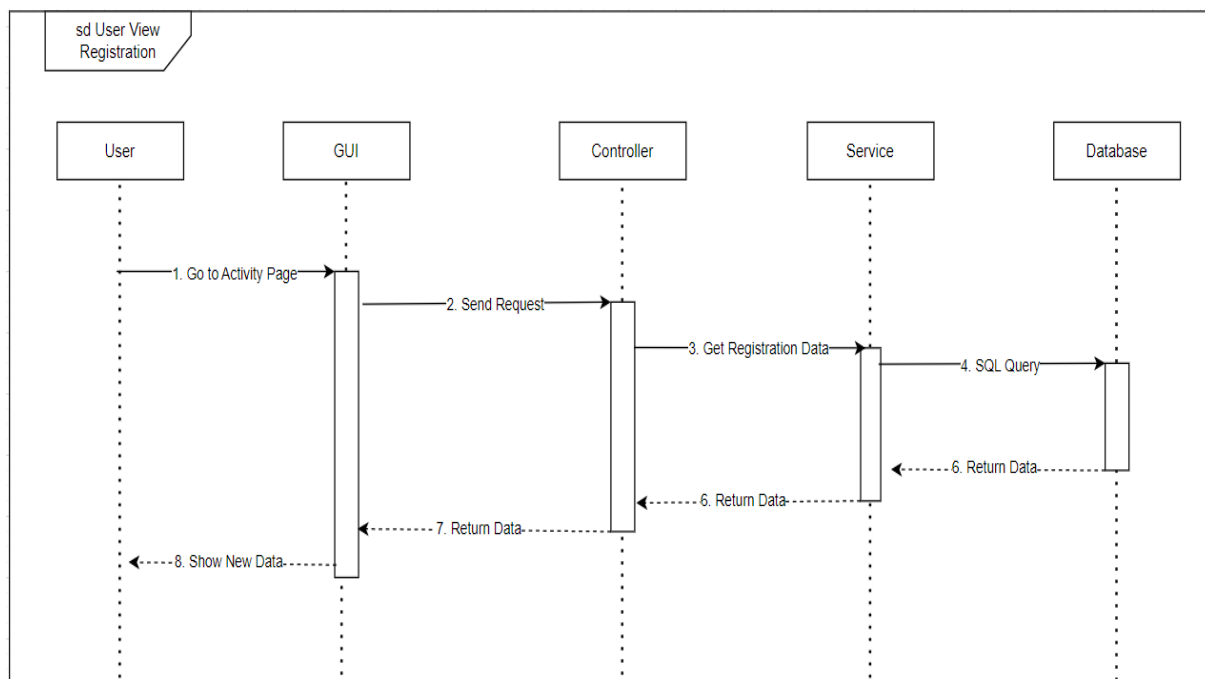


Hình 4.14: Sơ đồ chức năng xây dựng mô hình nhận diện khuôn mặt

4.2.5. Xem lịch sử tham gia

Bảng 4.15: Mô tả chức năng xem lịch sử tham gia

Tên chức năng	Xem lịch sử tham gia
Đối tượng sử dụng	User
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền User
Quy trình nghiệp vụ	Login → Vào trang Activity
Kết quả	Hiển thị các đơn đã đăng ký tham dự của người dùng

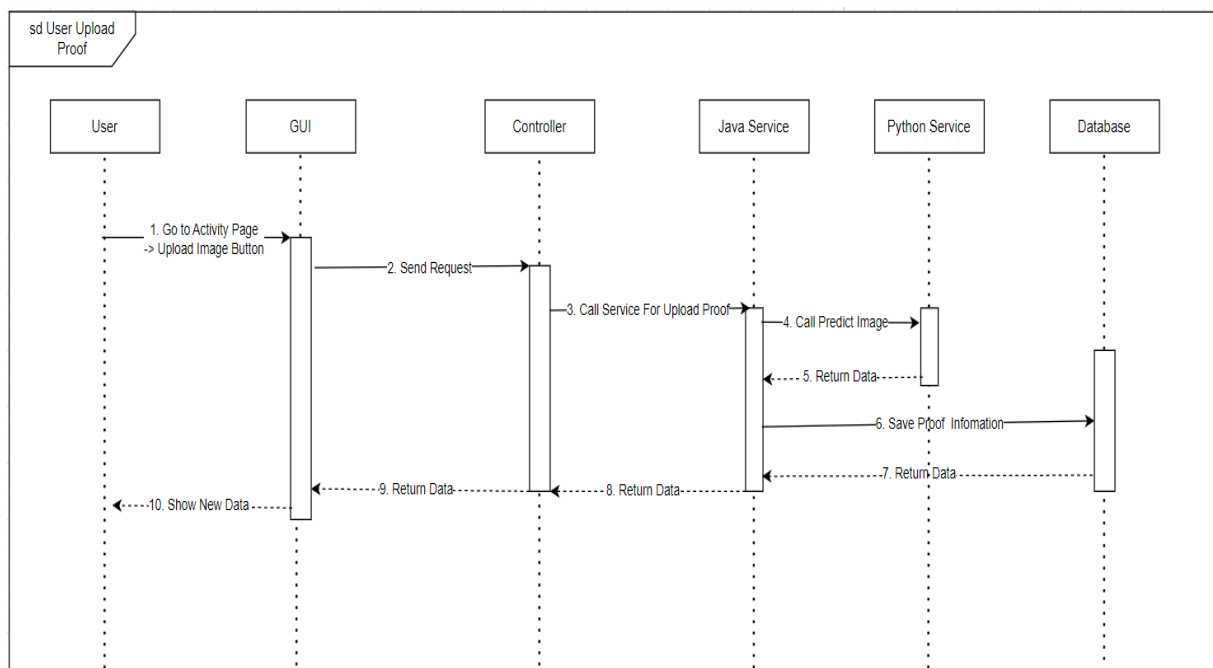


Hình 4.15: Sơ đồ chức năng xem lịch sử tham gia

4.2.6. Nộp minh chứng tham gia sự kiện

Bảng 4.16: Mô tả chức năng nộp minh chứng tham gia sự kiện

Tên chức năng	Nộp minh chứng
Đối tượng sử dụng	User
Điều kiện	Tài khoản tồn tại trên hệ thống, quyền User
Quy trình nghiệp vụ	Login → Vào trang Activity → click Upload button
Kết quả	Cập nhật lại trạng thái nộp minh chứng



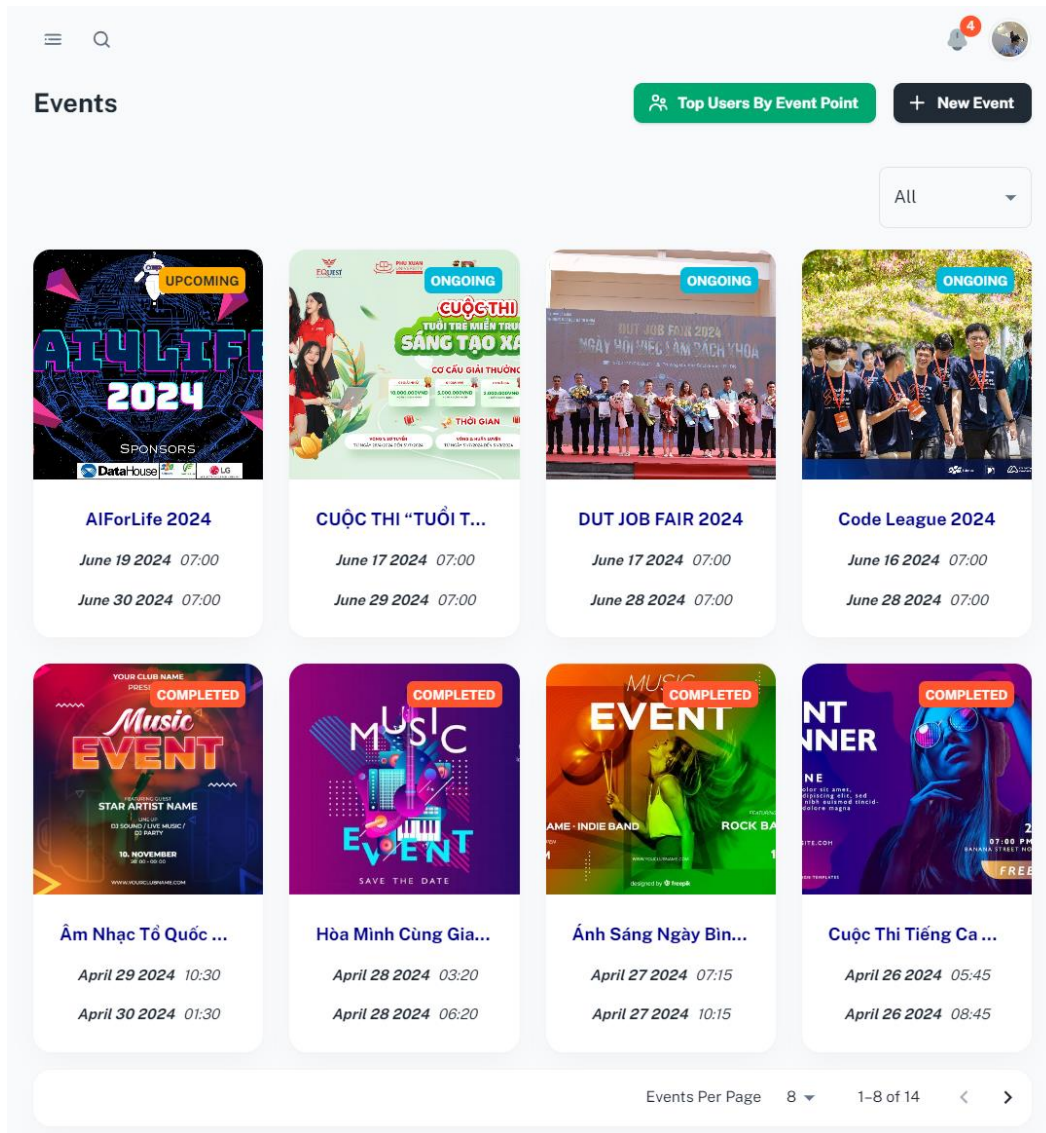
Hình 4.16: Sơ đồ chức năng nộp minh chứng tham gia

CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ

5.1. Kết quả

5.1.1. Giao diện website admin

Dưới đây là hình ảnh minh họa giao diện trang web dành cho admin. Giao diện này được thiết kế để giúp admin quản lý các hoạt động cộng đồng và ngoại khóa của sinh viên một cách hiệu quả.



Hình 5.1: Chức năng quản lý event

4

AIForLife 2024

Edit Event

List Registrants



Point: 10

Max Attenders: 100

Starts: 00:00:00 18-06-2024

Ends: 00:00:00 30-06-2024

Location: Bách Khoa Đà Nẵng

Description: Nhằm tạo sân chơi khoa học công nghệ dành cho sinh viên, đặc biệt là các bạn sinh viên đam mê lập trình, Trường Đại học Bách khoa, Đại học Đà Nẵng phối với với các doanh nghiệp tổ chức cuộc thi "AI4LIFE 2024" với chủ đề "AI Personal Trainer - AIPT".

Delete Event

Hình 5.2: Chức năng xem chi tiết Event

Add Event

Tên sự kiện

Điểm

0

Số người tối đa

0

Địa điểm

Mô tả

Thời gian bắt đầu

Thời gian kết thúc

Upload Images

Thêm

Hình 5.3: Chức năng thêm sự kiện

Chỉnh sửa thông tin

Tên sự kiện

AIForLife 2024

Điểm

10

Số người tối đa

100

Địa điểm

Bách Khoa Đà Nẵng

Mô tả

Nhằm tạo sân chơi khoa học công nghệ dành c

Thời gian bắt đầu

06/21/2024 12:00 AM

Thời gian kết thúc

06/30/2024 12:00 AM

Upload Images

[images.jpg](#) [images.jpg](#)

Lưu

Hình 5.4: Chỉnh sửa thông tin sự kiện

Sinh viên thực hiện: Phạm Quốc Tú

Hướng dẫn: TS Trương Ngọc Châu

42

List of Event Registrants					
Search user...					
Events					All ▾
Name ↑	Email	Gender	Phone	Status	
 phạm hiệu	chinhhiu0@gmail.com	MALE	0348643523	Active	⋮
 jack chín bảy	jack97@gmail.com	FEMALE	0328932112	Active	⋮
 dương công tiến	tiendc@gmail.com	MALE	0998734832	Active	⋮
 phạm chính hiệu	djfkjdjfdkkj@gmail.ccncnc	MALE	0348643572	Active	⋮
Rows per page: 5 ▾ 1-4 of 4 < >					

Hình 5.5: Danh sách người tham gia sự kiện

Top 10 Users By Event Point					
Filter Type: Filter by Month ▾ Month: June ▾ Year: 2024 ▾ <button>Search</button>					
Name ↑	Email	Gender	Amount Of Event	Total Point	
 dương công tiến	tiendc@gmail.com	MALE	1	20	⋮
 phạm chính hiệu	djfkjdjfdkkj@gmail.ccncnc	MALE	1	20	⋮
 phạm hiệu	chinhhiu0@gmail.com	MALE	1	10	⋮
Rows per page: 5 ▾ 1-3 of 3 < >					

Hình 5.6: Danh sách người có điểm tham gia sự kiện nhiều nhất

View submitted Evidence



Hình 5.7: Duyệt người dùng nộp minh chứng đăng kí sự kiện

Users

New User

Search user...

Name ↑	Email	Gender	Phone	Status	
 Phạm Quốc Tú	pumomecus@mailinator.com	MALE	1231231234	Active	⋮
 Quốc Tú Phạm	tutu@mailinator.com	MALE	1231231288	Active	⋮
 Nguyễn Hồng Trường	segofosa@mailinator.com	MALE	0977361239	Active	⋮
 Phan Quyên	qehyk@mailinator.com	MALE	0932417774	Active	⋮
 Phùng Thanh Độ	dochet1989@mailinator.com	FEMALE	3333333333	Active	⋮

Rows per page: 5 1-5 of 17

Hình 5.8: Quản lý người dùng

Users

New User

Search user...

Full Name

Gmail

UserName

Password

Phone Number

Date of birth
06/19/2024

Address

Gender

Submit Close

Name ↑	Email	Gender	Phone	Status	
 Phạm Quốc Tú	pumomecus@mailinator.com	MALE	1231231234	Active	⋮
 Quốc Tú Phạm	tutu@mailinator.com	MALE	1231231288	Active	⋮
 Nguyễn Hồng Trường	segofosa@mailinator.com	MALE	0977361239	Active	⋮
 Phan Quyên	qehyk@mailinator.com	MALE	0932417774	Active	⋮
 Phùng Thanh Độ	dochet1989@mailinator.com	FEMALE	3333333333	Active	⋮

Rows per page: 5 1-5 of 18

Hình 5.9: Giao diện thêm người dùng mới

Users

New User

Q Search

Name ↑

Phạm Quốc Tú

Quốc Tú Phạm

Nguyễn Hoàng

Phan Quyên

Phùng Thanh

User Update

FullName

Phạm Quốc Tú

Gmail

pumomecus@mailinator.com

UserName

jasagq

Phone Number

1231231234

Date of Birth

12/01/2002

Upload Avatar:

Chọn tệp

Không có tệp nào được chọn

Submit

Close

Phone

Status

1231231234

A

Edit

Delete

1231231288

Active

0977361239

Active

0932417774

Active

3333333333

Active

Rows per page:

5

1-5 of 18

Hình 5.10: Giao diện sửa đổi thông tin người dùng

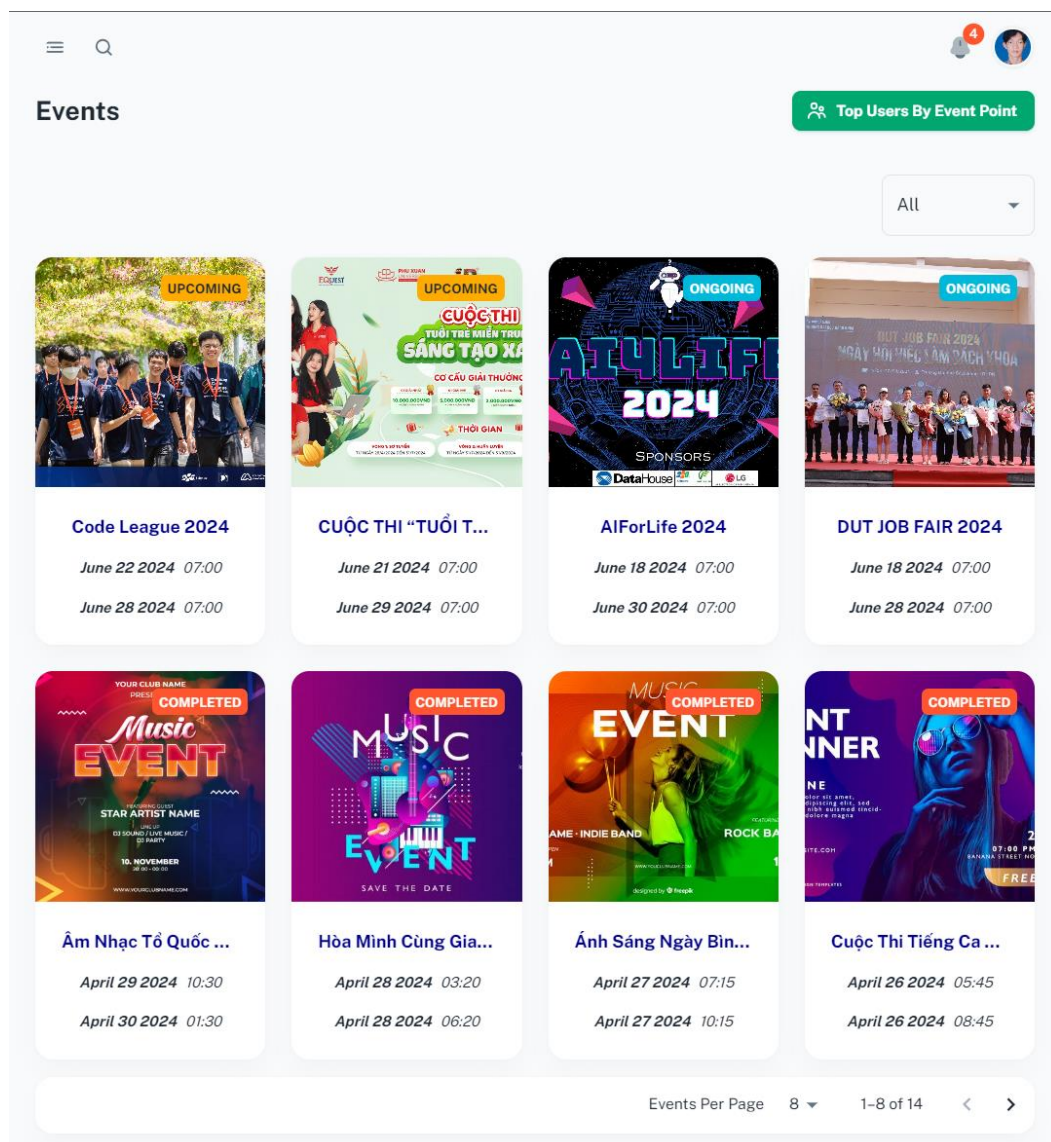
Sinh viên thực hiện: Phạm Quốc Tú

Hướng dẫn: TS Trương Ngọc Châu

46

5.1.2. Giao diện website users

Dưới đây là hình ảnh minh họa giao diện trang web dành cho user. Giao diện này thân thiện và dễ sử dụng, giúp sinh viên dễ dàng theo dõi và đăng ký các hoạt động cộng đồng và ngoại khóa.



Hình 5.11: Giao diện danh sách sự kiện của người dùng

CUỘC THI “TUỔI TRÈ MIỀN TRUNG S...

Cancel Registration

CUỘC THI
TUỔI TRÈ MIỀN TRUNG
SÁNG TẠO XANH

CƠ CẤU GIẢI THƯỞNG

Giải nhất	10,000,000 VND
Giải nhì	5,000,000 VND
Giải ba	2,000,000 VND
Giải khuyến khích	1,000,000 VND

THỜI GIAN

VÒNG 1: SƠ TUYỂN	VÒNG 2: HỮA LUYỆN	VÒNG 3: VÒNG CHUNG KẾT VÀ XẾP HẠNG
Từ ngày 26/06/2024 đến 01/07/2024	Từ ngày 17/07/2024 đến 18/07/2024	Tháng 8/2024

Point: 10

Max Attenders: 900

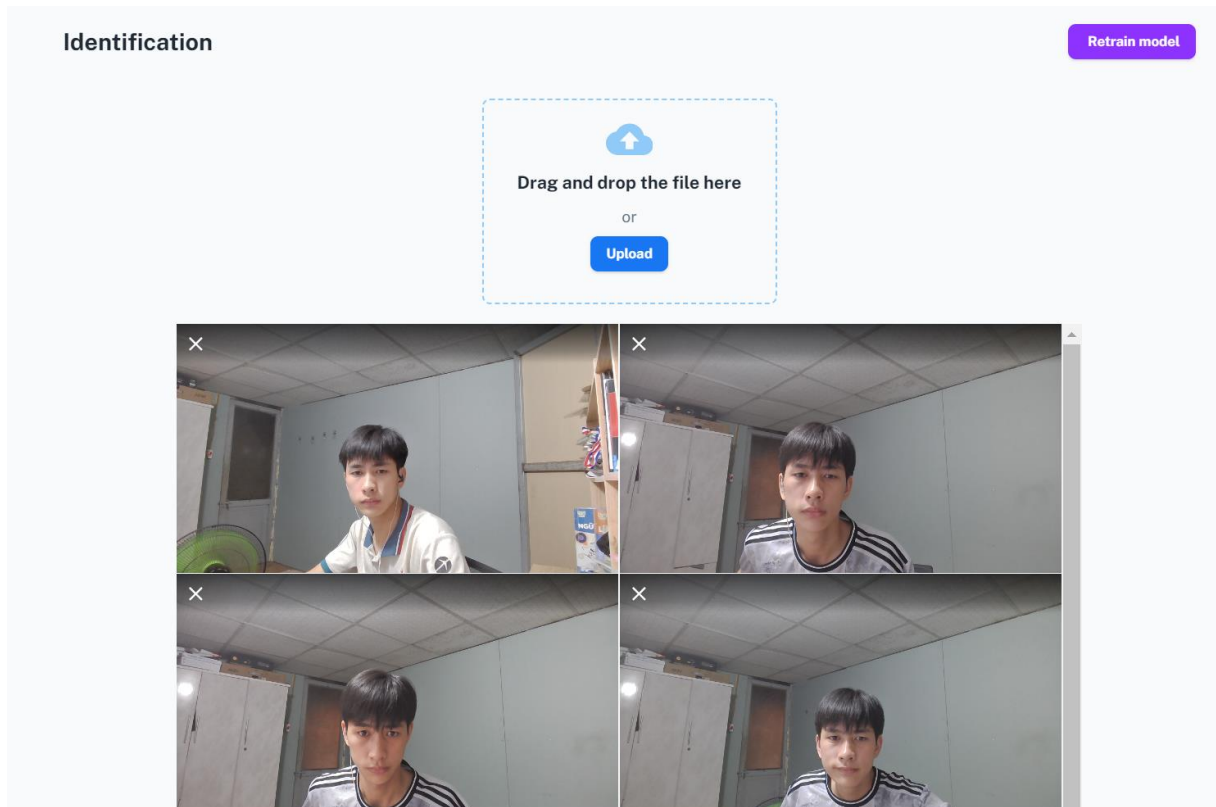
Starts: 00:00:00 21-06-2024

Ends: 00:00:00 29-06-2024

Location: Bách Khoa Đà Nẵng

Description: Cuộc thi “Tuổi trẻ miền Trung sáng tạo xanh 2024” với chủ đề “Khởi nghiệp sáng tạo xanh vì mục tiêu phát triển bền vững” là một sân chơi được đầu tư kỹ lưỡng với quy mô lớn, nằm trong khuôn khổ sự kiện “Chào mừng ngày sở hữu trí tuệ thế giới 2024” được phối hợp tổ chức bởi Tập đoàn Giáo dục EQuest và Trường Đại học Phú Xuân.

Hình 5.12: Giao diện chi tiết sự kiện của người dùng



Hình 5.13: Giao diện người dùng nộp ảnh để nhận diện khuôn mặt

List Event Registered						Total Point: 10	
Event Name	Start Time	End Time	Location	Point	Status	Upload Image	
AIForLife 2024	07:00:00 18-06-2024	07:00:00 30-06-2024	Bách Khoa Đà Nẵng	10	Registered But Not Attended	Not Accepted	⋮
CUỘC THI "TUỔI TRÈ MIỀN TRUNG SÁNG TẠO XANH 2024"	07:00:00 17-06-2024	07:00:00 29-06-2024	Bách Khoa Đà Nẵng	10	attended	Accepted	⋮
DUT JOB FAIR 2024	07:00:00 17-06-2024	07:00:00 28-06-2024	Bách Khoa Đà Nẵng	20	registered	<button>Nộp ảnh</button>	⋮
Code League 2024	07:00:00 16-06-2024	07:00:00 28-06-2024	39 Đà Nẵng	10	registered	<button>Nộp ảnh</button>	⋮

Rows per page: 5 1-4 of 4 < >

Hình 5.14: Quản lý hoạt động tham gia các sự kiện của người dùng

CUỘC THI “TUỔI TRÈ MIỀN TRUNG SẢN...

Register Event



Point: 10

Max Attenders: 900

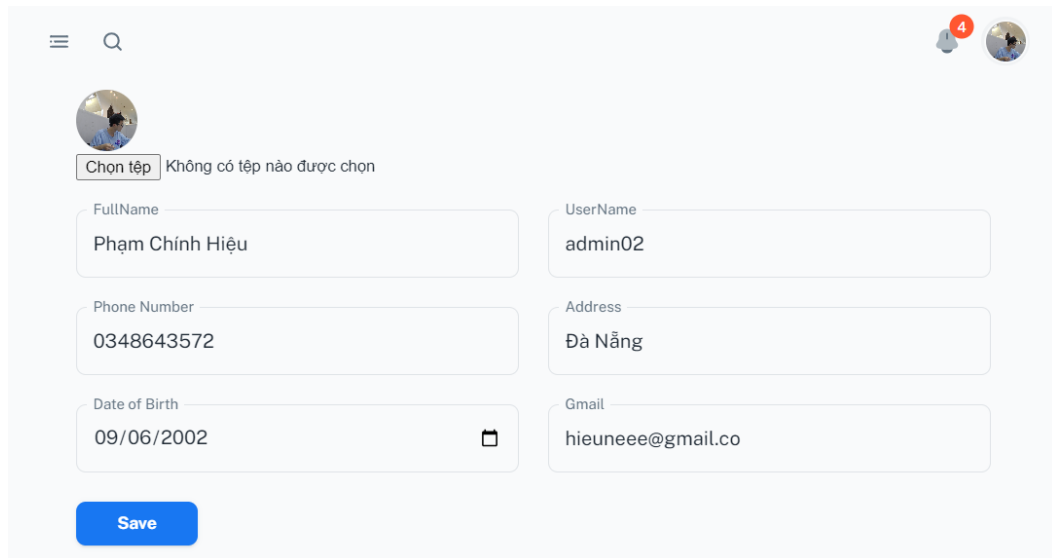
Starts: 00:00:00 21-06-2024

Ends: 00:00:00 29-06-2024

Location: Bách Khoa Đà Nẵng

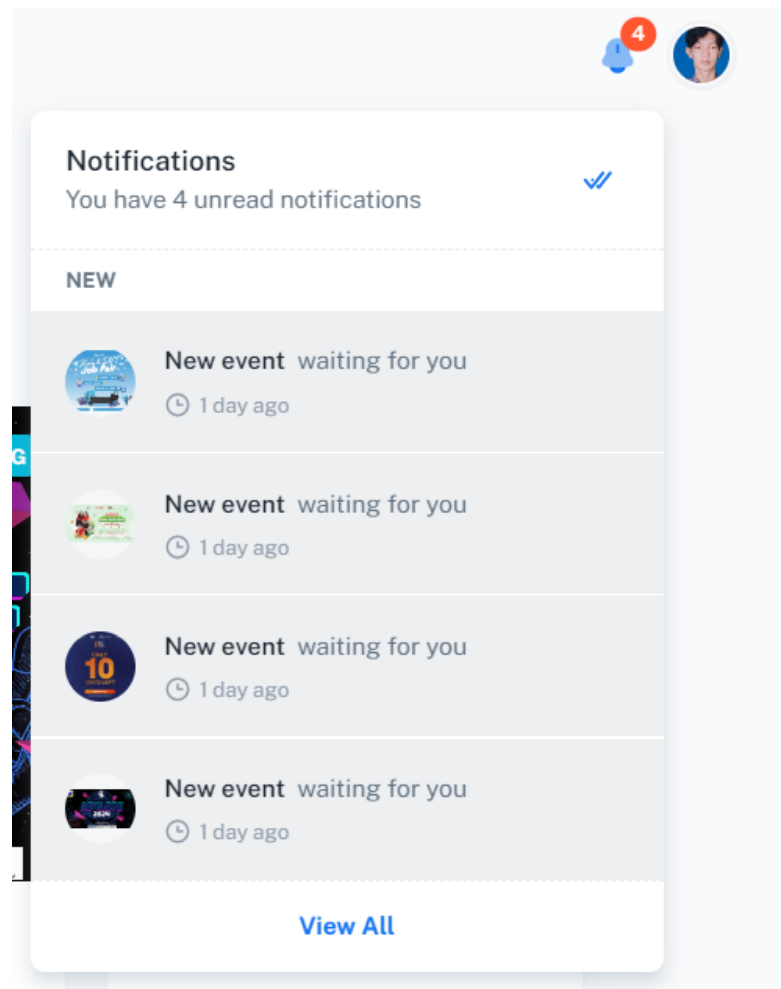
Description: Cuộc thi “Tuổi trẻ miền Trung sáng tạo xanh 2024” với chủ đề “Khởi nghiệp sáng tạo xanh vì mục tiêu phát triển bền vững” là một sân chơi được đầu tư kỹ lưỡng với quy mô lớn, nằm trong khuôn khổ sự kiện “Chào mừng ngày sở hữu trí tuệ thế giới 2024” được phối hợp tổ chức bởi Tập đoàn Giáo dục EQuest và Trường Đại học Phú Xuân.

Hình 5.15: Đăng kí tham gia sự kiện



The screenshot shows a user profile management interface. At the top, there is a search icon and a notification bell with a red badge showing '4'. Below this is a profile picture placeholder with the text 'Chọn tệp' and 'Không có tệp nào được chọn'. The form contains several input fields: 'FullName' with the value 'Phạm Chính Hiệu', 'UserName' with 'admin02', 'Phone Number' with '0348643572', 'Address' with 'Đà Nẵng', 'Date of Birth' with '09/06/2020' and a calendar icon, and 'Gmail' with 'hieuneee@gmail.co'. A blue 'Save' button is at the bottom left.

Hình 5.16: Giao diện người dùng quản lý trang cá nhân



Hình 5.17: Giao diện thông báo của người dùng

5.2. Đánh giá

- **Về giao diện người dùng**

Giao diện người dùng (UI) của hệ thống được thiết kế với tiêu chí thân thiện và dễ sử dụng. Các thành phần giao diện như nút bấm, biểu tượng và menu đều được bố trí hợp lý, giúp người dùng dễ dàng thực hiện các thao tác cơ bản.

Mặc dù còn một số điểm có thể cải thiện về mặt thẩm mỹ, giao diện hiện tại vẫn đáp ứng được nhu cầu sử dụng cơ bản của người dùng.

- **Về hiệu suất hệ thống**

Hệ thống đã được kiểm tra hiệu suất trong điều kiện sử dụng thông thường. Kết quả cho thấy hệ thống có thể xử lý các tác vụ cơ bản một cách mượt mà, với thời gian phản hồi khá nhanh. Tuy nhiên, khi số lượng yêu cầu tăng lên đáng kể, hệ thống bắt đầu xuất hiện một số hiện tượng chậm chạp. Điều này cho thấy có thể cần tối ưu thêm về mặt mã nguồn hoặc cân nhắc nâng cấp phần cứng nếu hệ thống dự kiến sẽ phục vụ cho một lượng người dùng lớn hơn.

- **Về mô hình nhận diện khuôn mặt**

Mô hình nhận diện khuôn mặt được triển khai trong hệ thống sử dụng các thuật toán cơ bản và đạt được độ chính xác khá cao trong các thử nghiệm ban đầu. Khi thử nghiệm trên một bộ dữ liệu nhỏ, mô hình đã nhận diện chính xác khoảng 85-90% các khuôn mặt.

Tuy nhiên, trong các điều kiện ánh sáng yếu hoặc khi khuôn mặt bị che khuất một phần, độ chính xác của mô hình giảm xuống. Điều này chỉ ra rằng mô hình cần được cải tiến thêm và có thể cần huấn luyện trên một bộ dữ liệu lớn và đa dạng hơn để nâng cao hiệu quả nhận diện trong các tình huống phức tạp hơn.

KẾT LUẬN

Trong thời đại hiện nay, việc tham gia các hoạt động cộng đồng và ngoại khóa đã trở thành một phần quan trọng trong quá trình phát triển toàn diện của sinh viên. Hệ thống đánh giá hoạt động cộng đồng và ngoại khóa của chúng tôi được phát triển nhằm mục tiêu hỗ trợ nhà trường và sinh viên trong việc quản lý và theo dõi các hoạt động này một cách hiệu quả. Qua quá trình nghiên cứu và phát triển, hệ thống đã đạt được những kết quả đáng khích lệ:

1. Giao diện thân thiện và dễ sử dụng: Hệ thống được thiết kế với giao diện thân thiện, dễ sử dụng cho cả admin và sinh viên. Điều này giúp người dùng dễ dàng tiếp cận và sử dụng hệ thống mà không gặp khó khăn.
2. Chức năng quản lý hiệu quả: Các chức năng quản lý của hệ thống bao gồm tạo mới hoạt động, theo dõi tiến độ và thành tích của sinh viên. Những chức năng này đã giúp tối ưu hóa công việc quản lý của nhà trường và đảm bảo mọi hoạt động đều được theo dõi chặt chẽ.
3. Tăng cường tính tương tác và động lực tham gia: Hệ thống không chỉ giúp sinh viên dễ dàng đăng ký và theo dõi các hoạt động mà còn tạo động lực cho họ thông qua việc ghi nhận và đánh giá thành tích. Điều này khuyến khích sinh viên tích cực tham gia các hoạt động ngoại khóa, góp phần nâng cao kỹ năng và kinh nghiệm thực tế.
4. Khả năng mở rộng và tích hợp: Hệ thống được thiết kế với khả năng mở rộng và tích hợp cao, cho phép dễ dàng bổ sung các tính năng mới và kết nối với các hệ thống khác như hệ thống quản lý học tập (LMS).

Tuy nhiên, hệ thống vẫn còn một số hạn chế cần được khắc phục và cải tiến trong tương lai:

- Tối ưu hóa hiệu suất: Cần cải thiện hiệu suất hệ thống để đảm bảo hoạt động mượt mà khi số lượng người dùng tăng cao.
- Bảo mật dữ liệu: Tăng cường các biện pháp bảo mật để bảo vệ dữ liệu cá nhân của sinh viên và thông tin hoạt động.

TÀI LIỆU THAM KHẢO

1. <https://thegioimaychu.vn/blog/ai-hpc/huong-dan-xay-dung-he-thong-phat-hien-va-nhan-dien-khuon-mat-p5788/>
2. <https://spring.io/projects/spring-data-jpa>
3. <https://www.codejava.net/frameworks/spring-boot/connect-to-mysql-database-examples>
4. <https://www.codejava.net/frameworks/spring-boot/connect-to-mysql-database-examples>
5. <https://viblo.asia/p/gioi-thieu-ve-kien-truc-microservices-4P8566O35Y3>