Trường đại học Công nghệ

Viện trí tuệ nhân tạo

BÁO CÁO

Bài tập lớn Thực hành phát triển

hệ thống Trí tuệ nhân tạo

ĐỀ TÀI

Hệ Thống Nhận Diện Khuôn

Mặt Trong Thời Gian Thực

Nhóm: *Asteroid Destroyer*

Thành Viên:

STT	*Họ và tên*	*MSV*
1	*Nông Phi Long*	*23020398*
2	*Nguyễn Hữu Hoàng Nam*	*23020405*
3	*Phạm Quân*	*23020418*
4	*Phạm Hải Tiến*	*23020425*
5	*Chu Thanh Tùng*	*23020431*

Mục lục:

I. Giới thiệu chung

Dự án này triển khai một hệ thống nhận diện khuôn mặt thời gian thực, bao gồm các thành phần chính:

- *Frontend người dùng: Giao diện nhận diện khuôn mặt.*
- *Admin dashboard: Giao diện quản trị người dùng.*
- *Backend xử lý AI: Xử lý nhận diện khuôn mặt theo thời gian thực.*

Hệ thống được triển khai sử dụng Docker, với các liên kết Dockerhub:

- *ait_backend*
- *ait_frontend*

II. Mục tiêu

- *Phần mềm chạy trên nền tảng ứng dụng web.*
- *Phần mềm phát triển một hệ thống nhận diện khuôn mặt thời gian thực.*
- *Phần mềm cung cấp giao diện người dùng và quản trị để quản lý hệ thống.*
- *Phần mềm triển khai hệ thống sử dụng Docker triển khai và quản lý.*

III. Công nghệ sử dụng

Docker

- ***Vai trò**: Đóng gói từng thành phần của hệ thống (AI, Backend, Frontend, Database) thành container.*

- ***Lợi ích**:*

    - *Đảm bảo môi trường nhất quán trên mọi máy chủ.*

    - *Tăng tính di động và khả năng mở rộng hệ thống.*

Docker Compose

- ***Vai trò**: Quản lý nhiều container cùng lúc thông qua một tệp cấu hình docker-compose.yml.*
- ***Lợi ích**:*
    - *Triển khai toàn bộ hệ thống (AI, Backend, Frontend, DB) chỉ với một lệnh.*
    - *Dễ cấu hình mạng nội bộ giữa các dịch vụ.*
    - *Quản lý volume, môi trường và thứ tự khởi chạy dễ dàng.*

Python

- ***Vai trò**: Ngôn ngữ lập trình chính cho phần xử lý AI và backend API.*
- ***Lợi ích**:*
    - *Có nhiều thư viện mạnh cho AI/ML (OpenCV, NumPy, DeepFace,...).*
    - *Dễ đọc, dễ bảo trì, cộng đồng hỗ trợ lớn.*
    - *Phù hợp cho cả viết logic backend và xử lý ảnh.*

Django

- ***Vai trò**: Framework phát triển backend API phục vụ quản lý người dùng, xác thực đăng nhập và giao tiếp với AI.*
- ***Lợi ích**:*
    - *Tích hợp sẵn ORM, Auth, Admin – giúp phát triển nhanh chóng.*
    - *Bảo mật cao, có sẵn các tính năng chống CSRF, XSS,...*
    - *Dễ kết nối với các thư viện AI/Python.*

React

- ***Vai trò**: Xây dựng giao diện người dùng (UI/UX) hiện đại cho phần frontend.*
- ***Lợi ích**:*
    - *Hiển thị dữ liệu theo thời gian thực (VD: kết quả nhận diện khuôn mặt).*
    - *Tái sử dụng component, dễ mở rộng và bảo trì.*
    - *Kết hợp tốt với RESTful API từ Django.*

DeepFace

Vai trò*: Thư viện trí tuệ nhân tạo chuyên dùng để nhận diện và phân tích khuôn mặt người từ ảnh hoặc video, thường kết hợp với các thư viện xử lý ảnh như OpenCV để trích xuất đầu vào.*
Lợi ích:

- *Cung cấp các mô hình nhận diện khuôn mặt mạnh mẽ như VGG-Face, Facenet, ArcFace, v.v.*
- *Dễ tích hợp với Python và các hệ thống AI.*
- *Hỗ trợ nhiều tác vụ: xác minh khuôn mặt, phân tích cảm xúc, tuổi, giới tính, dân tộc.*
- *Hiệu quả cao, phù hợp cho cả ứng dụng thực tế và nghiên cứu.*

SQLite

Vai trò: *Tệp cơ sở dữ liệu sử dụng công nghệ SQLite, có nhiệm vụ lưu trữ dữ liệu dạng quan hệ trong một tệp duy nhất, thường được dùng trong các ứng dụng Python như Django để quản lý dữ liệu người dùng, nội dung và cấu trúc hệ thống.*

Lợi ích:

- *Không cần cài đặt máy chủ cơ sở dữ liệu riêng, dễ triển khai trong môi trường phát triển.*
- *Tương thích tốt với Django và các ứng dụng Python khác.*
- *Lưu trữ toàn bộ dữ liệu (bảng, chỉ mục, quan hệ...) gọn gàng trong một tệp duy nhất.*
- *Hiệu năng tốt cho ứng dụng nhỏ và trung bình, phù hợp để phát triển nhanh và kiểm thử.*

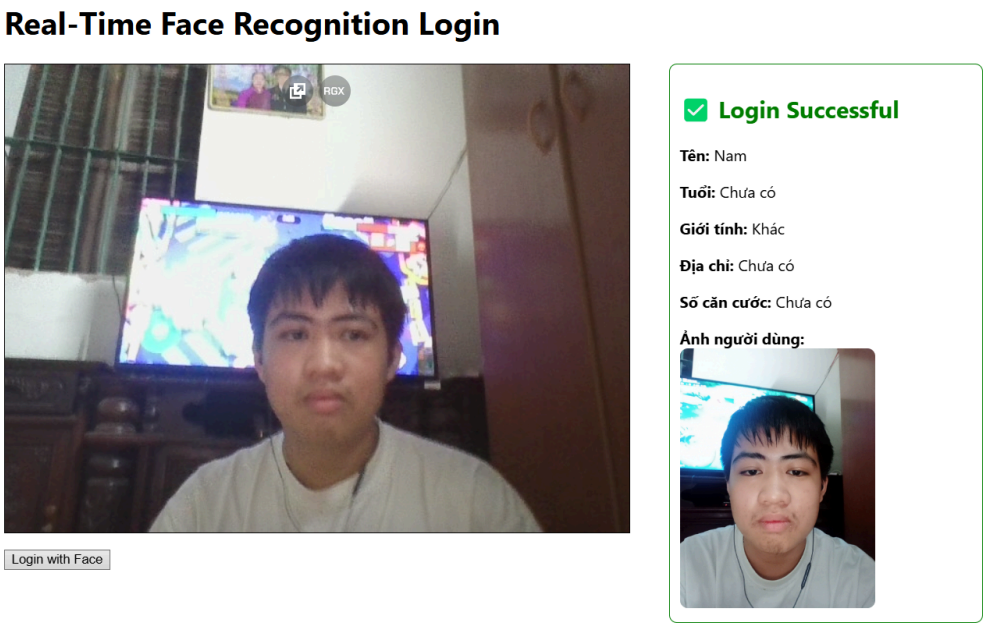
IV. Đặc tả yêu cầu và cài đặt

4.1. Yêu cầu chức năng:

- ***Nhận diện khuôn mặt thời gian thực**:*

  Hệ thống nhận diện khuôn mặt người dùng thông qua webcam và xử lý trực tiếp trên trình duyệt web (khi đã cập nhật ảnh).

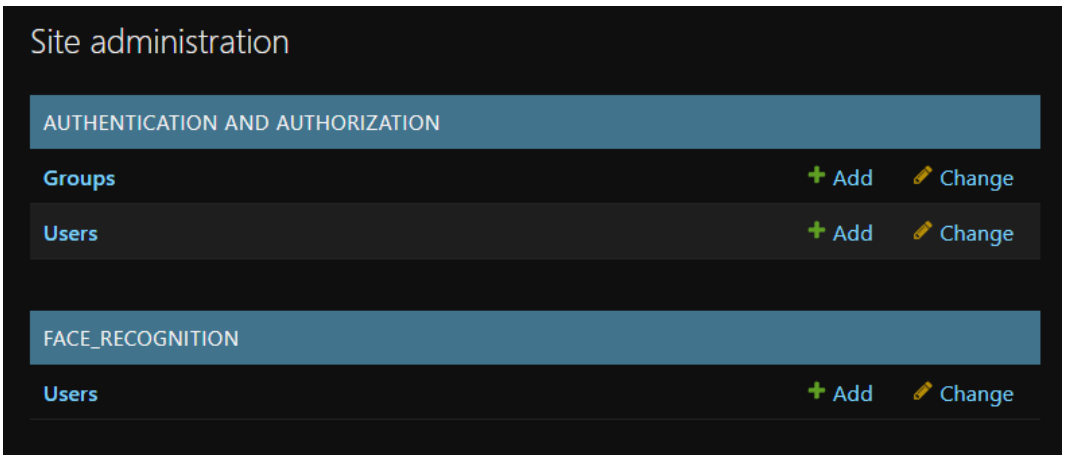
  *Nhận diện thành công:



*Nhận diện thất bại(khi chưa thêm ảnh người dùng):

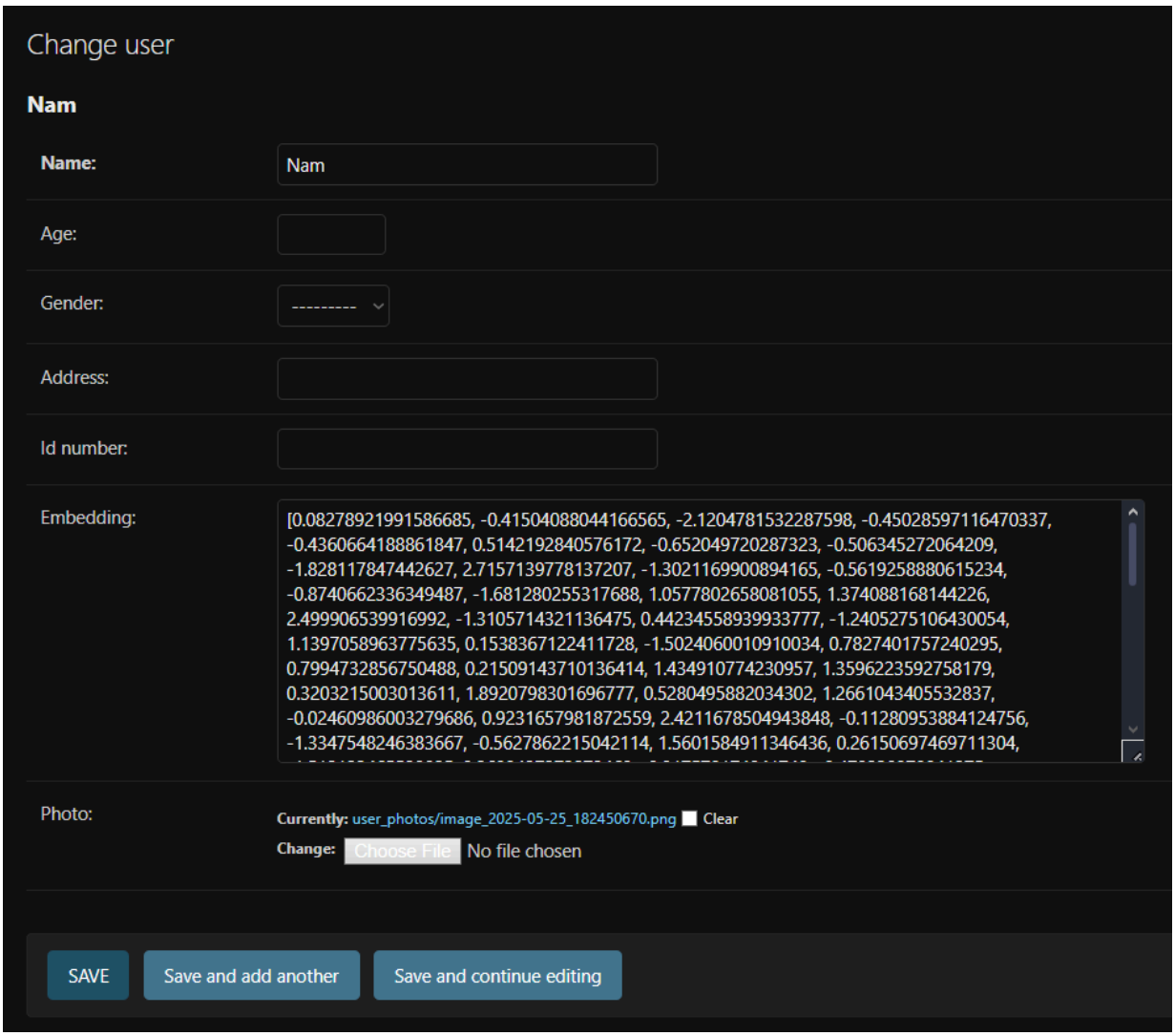
- ***Quản lý người dùng***:

  Hệ thống cho phép quản trị viên thêm, xóa và sửa người dùng.



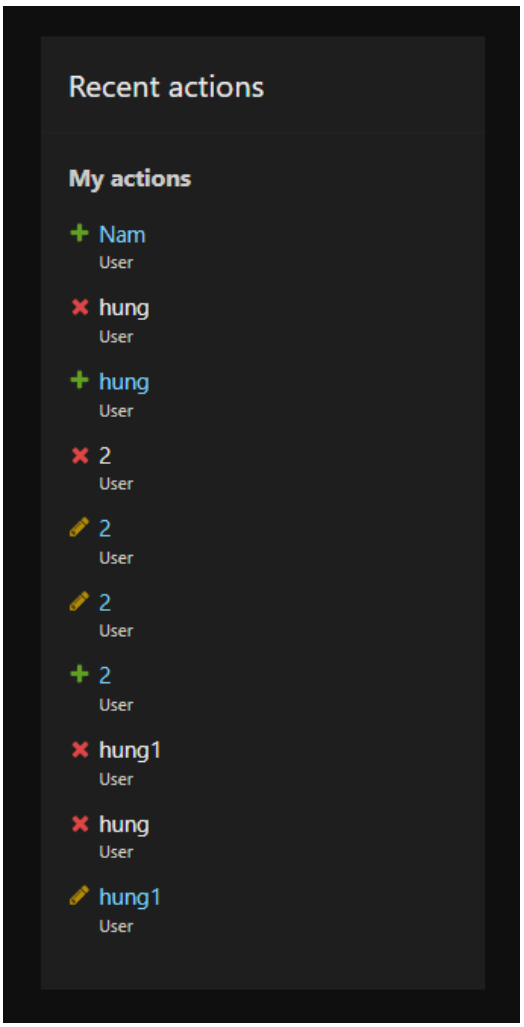
- ***Quản lý thông tin cá nhân***

  Hệ thống cho phép cập nhật và lưu trữ các thông tin cá nhân (tên, tuổi, giới tính,...).



- ***Xem lịch sử chỉnh sửa***

  Hệ thống cho phép xem lại lịch sử thêm, xóa hay chỉnh sửa thông tin người dùng.



4.2. Yêu cầu phi chức năng:

- ***Hiệu suất**:*

  Xử lý nhận diện khuôn mặt trong thời gian thực với độ trễ tối thiểu

- ***Khả năng mở rộng**:*

Sử dụng Docker để dễ dàng mở rộng và triển khai.

- ***Tính tương thích**:*

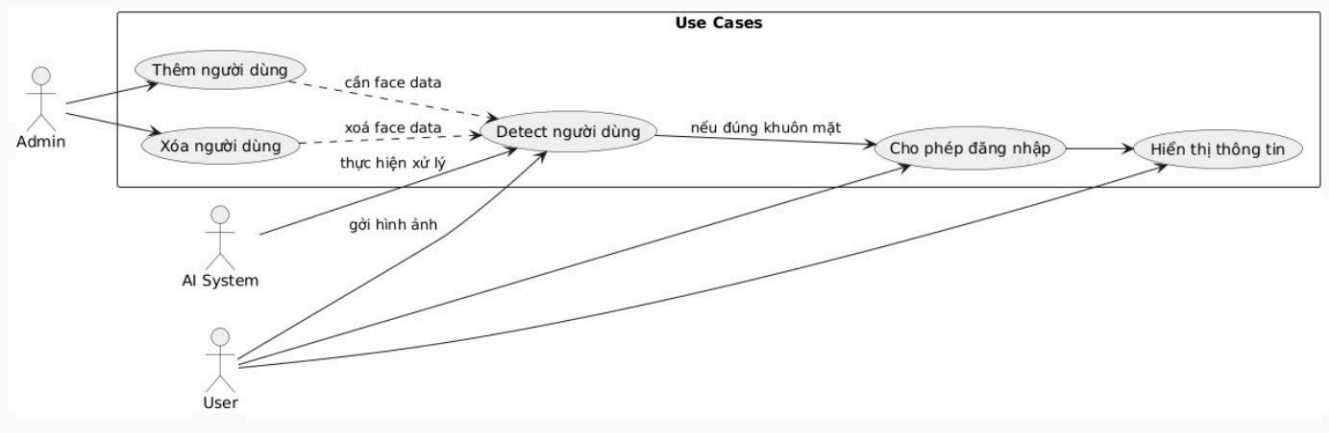
  Hoạt động tốt trên các trình duyệt phổ biến.

- ***Dễ dàng bảo trì**:*

  Dễ dàng cập nhật hoặc thay thế các thành phần.

4.3 Thiết kế biểu đồ cho hệ thống

Use case Diagram



Actor

1. ***Admin**: Quản trị viên có quyền thêm hoặc xóa người dùng.*

2. ***User****: Người dùng hệ thống, gửi hình ảnh để đăng nhập và xem thông tin.*

3. ***AI System****: Hệ thống AI thực hiện việc xử lý và nhận diện khuôn mặt*

Use Cases (Các ca sử dụng):

1. ***Thêm người dùng****:*
    - *Do **Admin** thực hiện.*
    - *Gửi dữ liệu khuôn mặt (face data) đến ca sử dụng **Detect người dùng**.*
2. ***Xóa người dùng****:*
    - *Do **Admin** thực hiện.*
    - *Gửi yêu cầu xóa dữ liệu khuôn mặt cho **Detect người dùng**.*
3. ***Detect người dùng****:*
    - *Nhận hình ảnh từ **User** và **AI System**.*

- *Xử lý dữ liệu, nếu đúng khuôn mặt thì chuyển tiếp đến **Cho phép đăng nhập**.*

- *Là trung tâm nhận và xử lý dữ liệu từ nhiều phía.*

4. ***Cho phép đăng nhập**:*

- *Xảy ra sau khi xác minh khuôn mặt đúng trong **Detect người dùng**.*

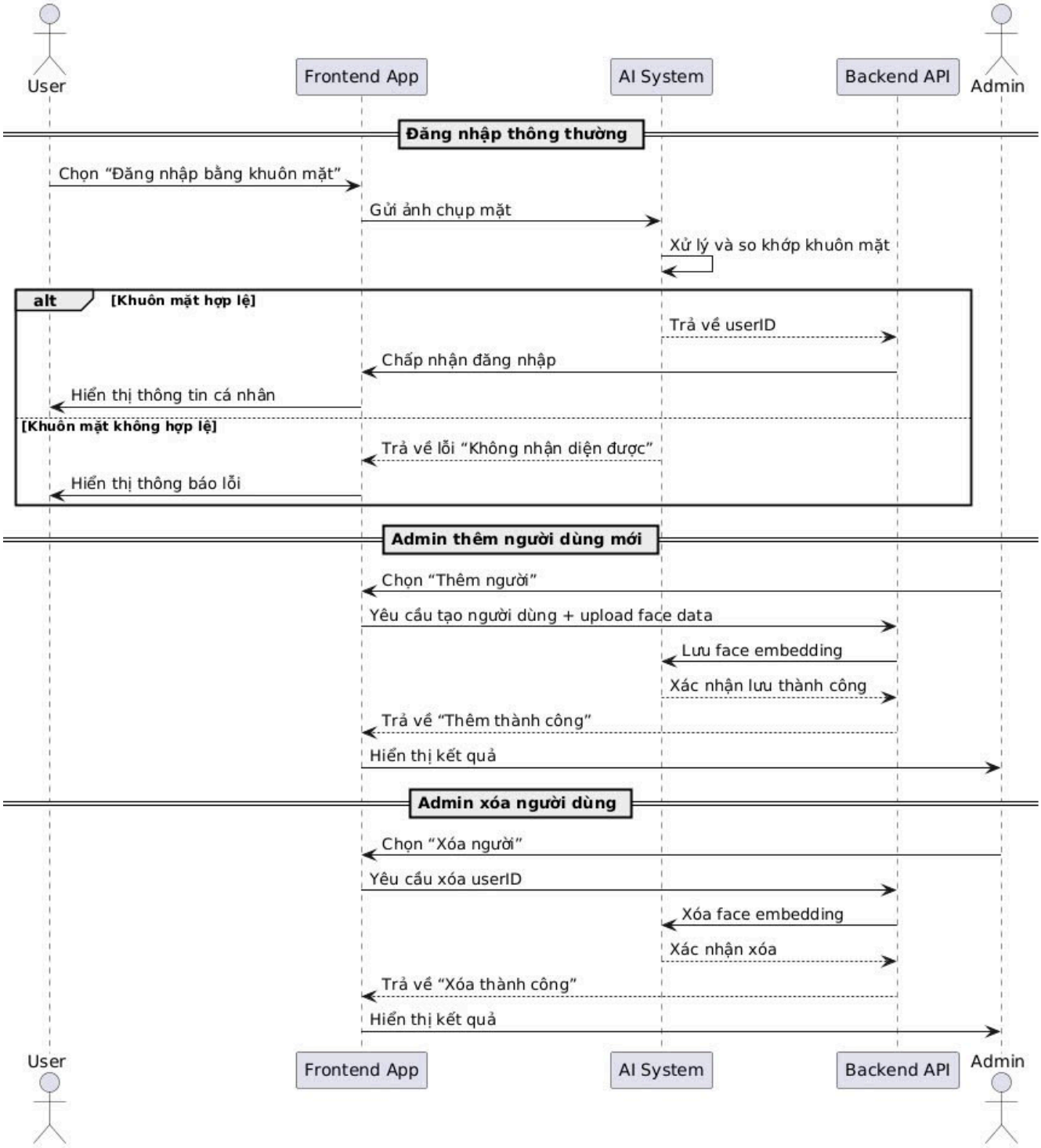
- *Cho phép người dùng truy cập hệ thống.*

5. ***Hiển thị thông tin**:*

- *Chỉ xảy ra khi người dùng đã đăng nhập thành công.*

- *Hiển thị thông tin cá nhân hoặc dữ liệu liên quan đến người dùng.*

Biểu đồ tuần tự



***Actors**:*

- *User: Người dùng hệ thống (thường để đăng nhập).*
- *Admin: Quản trị viên (quản lý người dùng).*

***Các hệ thống tương tác**:*

- *Frontend App: Giao diện người dùng.*
- *AI System: Xử lý ảnh, nhận diện khuôn mặt.*
- *Backend API: Điều phối xử lý logic và dữ liệu backend.*

1. Đăng nhập thông thường

Luồng xử lý:

*1. **User** chọn "Đăng nhập bằng khuôn mặt".*

*2. **Frontend App** gửi ảnh chụp mặt đến **AI System**.*

3. ***AI System*** *xử lý ảnh và so khớp khuôn mặt → nếu hợp lệ, trả lại userID.*

4. ***Backend API*** *nhận userID, xác thực → chấp nhận đăng nhập.*

5. *Nếu hợp lệ:*

    - *Giao diện hiển thị thông tin cá nhân.*

6. *Nếu không hợp lệ:*

    - *Trả về lỗi "Không nhận diện được" → giao diện hiển thị thông báo lỗi.*

Ghi chú*:*

- *Biểu đồ sử dụng nhánh alt để mô tả 2 tình huống (khuôn mặt hợp lệ vs không hợp lệ).*

2. Admin thêm người dùng mới

Luồng xử lý:

1. ***Admin*** *chọn "Thêm người".*

2. ***Frontend App*** *gửi yêu cầu tạo người dùng và dữ liệu khuôn mặt đến* ***AI System****.*

3. ***AI System*** *thực hiện lưu face embedding (đặc trưng khuôn mặt).*

4. ***Backend API*** *xác nhận lưu thành công và gửi phản hồi "Thêm thành công".*

5. ***Frontend App*** *hiển thị kết quả.*

3. Admin xóa người dùng

Luồng xử lý:

1. ***Admin*** *chọn "Xóa người".*

2. ***Frontend App*** *gửi yêu cầu xóa với userID.*

3. ***AI System*** *xóa dữ liệu face embedding tương ứng.*

4. ***Backend API*** *xác nhận xóa thành công → phản hồi "Xóa thành công".*

5. ***Frontend App*** *hiển thị kết quả.*

4.4 Cài đặt

** Liên kết Dockerhub:*

dockerhub ai backend

dockerhub ai frontend

- ***B1. Lưu File Cấu Hình docker-compose.yml***

  Tải và lưu file docker-compose.yml.

- ***B2. Khởi Động Hệ Thống Bằng Docker***

  Mở terminal tại thư mục chứa project và chạy các lệnh sau:

```
docker-compose pull
docker-compose up -d
```

- ***B3. Truy Cập Ứng Dụng***

  Sau khi các container đã được khởi chạy thành công, bạn có thể truy cập các thành phần của hệ thống tại:

** Giao diện nhận diện khuôn mặt (Frontend người dùng):*

frontend người dùng

** Trang quản trị người dùng (Admin Dashboard):*

admin dashboard

V. Kết luận

- Nhìn chung, dự án của nhóm Asteroid Destroyer đã hoàn thành việc xây dựng hệ thống nhận diện khuôn mặt trong thời gian thực. Phần mềm đã hoạt động một cách ổn định, có thể nhận diện khuôn mặt để trích xuất thông tin của người dùng.
- Thông qua dự án này, nhóm chúng em đã tích lũy được thêm kiến thức phát triển hệ thống AI nhằm phục vụ cho việc nhận dạng khuôn mặt theo thời gian thực; đồng thời dự án còn giúp nhóm thành thạo hơn trong việc sử dụng Docker để triển khai và container hóa hệ thống AI.