

**ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



Báo cáo: Project 2

ĐỀ TÀI: Xây dựng dịch vụ định danh bằng ảnh chân dung

Sinh viên thực hiện: Phạm Thanh An – 20224911

Giáo viên hướng dẫn: TS. Nguyễn Bá Ngọc

Hà Nội, Tháng 6–2025

MỤC LỤC

LỜI NÓI ĐẦU	2
CHƯƠNG 1: KHẢO SÁT HIỆN TRẠNG VÀ LỰA CHỌN ĐỀ TÀI	3
1.1. Hiện trạng dịch vụ định danh	3
1.2. Mục tiêu và yêu cầu đề tài	3
1.3. Phạm vi và giới hạn	3
CHƯƠNG 2: ĐẶC TẢ YÊU CẦU BÀI TOÁN	4
2.1. Biểu đồ use case tổng quan	4
2.2. Danh sách các tác nhân	5
2.3. Danh sách các Usecase	5
2.4. Đặc tả use case	6
2.5. Mô hình hóa hành vi	11
CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU	15
CHƯƠNG 4: THIẾT KẾ KIẾN TRÚC	16
4.1. Kiến trúc hệ thống	16
4.2. Mô tả chi tiết từng thành phần trong hệ thống	16
CHƯƠNG 5: THIẾT KẾ GIAO DIỆN	20
5.1. Thiết kế giao diện người dùng	20
5.2. Thiết kế giao diện admin	22
CHƯƠNG 6: Xây dựng và thử nghiệm	24
6.1. Xây dựng	24
6.2. Thử nghiệm	24
CHƯƠNG 7: TỔNG KẾT	25
7.1. Đánh giá kết quả thu được	25
7.1.1. Ưu điểm	25
7.1.2. Hạn chế	25
7.1.3. Những kinh nghiệm và kiến thức đã đạt được	25
7.2. Hướng phát triển	25
7.3. Kết thúc	26

LỜI NÓI ĐẦU

Ngày nay chuyển đổi số ngày càng được phát triển và phổ biến rộng rãi, trong đó việc xác thực danh tính người dùng một cách nhanh chóng và an toàn đóng vai trò then chốt trong nhiều lĩnh vực như tài chính, thương mại. Dịch vụ định danh bằng ảnh chân dung(FaceID) là một giải pháp sử dụng mô hình học sâu, việc nhận diện khuôn mặt trở nên chính xác và hiệu quả. Đề tài này tập trung nghiên cứu và xây dựng một hệ thống định danh người dùng dựa trên ảnh chân dung, từ đó ứng dụng vào bài toán thực tế giao dịch trong ngân hàng. Hệ thống bao gồm các chức năng như trích xuất đặc trưng khuôn mặt, so khớp sinh trắc học và đánh giá độ tin cậy.

CHƯƠNG 1: KHẢO SÁT HIỆN TRẠNG VÀ LỰA CHỌN ĐỀ TÀI

1.1 Hiện trạng dịch vụ định danh

Hiện nay, việc xác minh danh tính người dùng trong các hệ thống nếu thực hiện thủ công thường gặp nhiều bất cập và hạn chế. Các nghiệp vụ như xác minh thông tin cá nhân, đối chiếu giấy tờ, kiểm tra dữ liệu,... dễ xảy ra sai sót và khó kiểm soát khi khối lượng người dùng lớn.

Các cơ quan, tổ chức và doanh nghiệp cũng cần một hệ thống định danh thông minh để dễ dàng kiểm tra, lưu trữ và phân tích dữ liệu khách hàng, cũng như đảm bảo tuân thủ các quy định về bảo mật thông tin cá nhân. Việc không áp dụng công nghệ định danh tự động có thể dẫn đến rủi ro gian lận, mất dữ liệu,...

Mong muốn đặt ra là phát triển một dịch vụ định danh bằng ảnh chân dung hiện tại, sử dụng thư viện deep learning giúp tự động hóa quá trình xác minh danh tính, tăng cường chính xác, giảm thiểu rủi ro, nâng cao trải nghiệm người dùng.

1.2. Mục tiêu và yêu cầu đề tài

Mục tiêu:

- Xây dựng hệ thống đăng ký, nhận diện khuôn mặt từ camera thời gian thực.
- Xác minh danh tính người dùng thông qua ảnh chân dung.
- Quản lý thông tin định danh người dùng, có thể tra cứu khuôn mặt từ ảnh bên ngoài.

Yêu cầu:

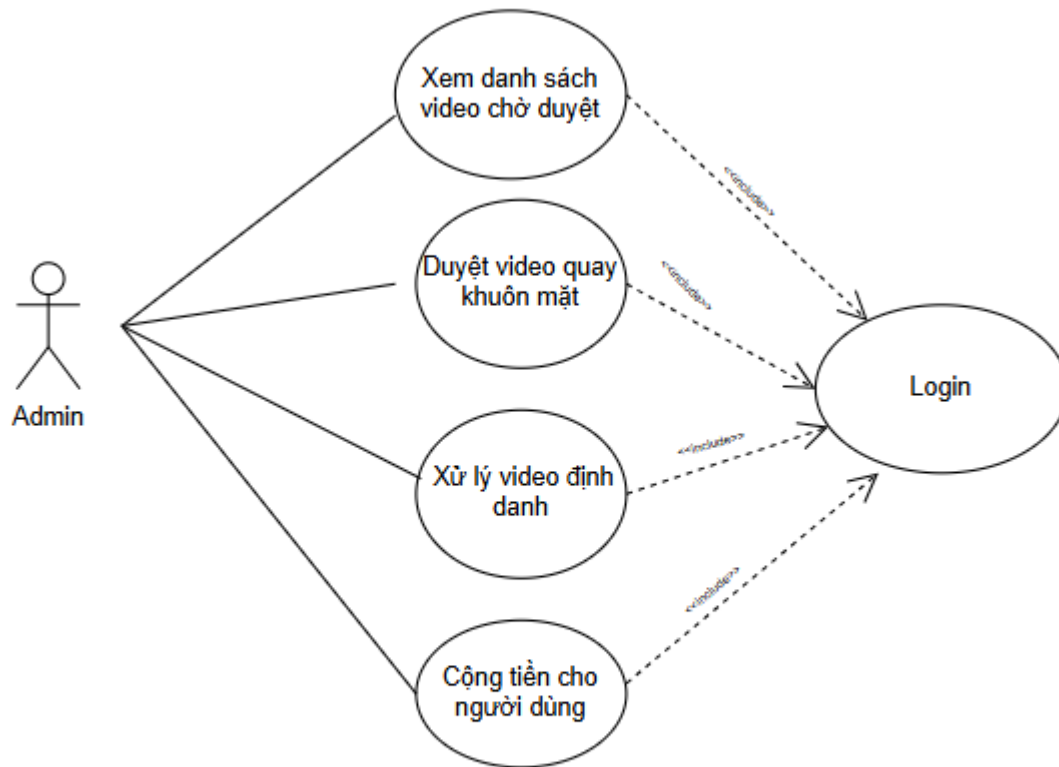
- Giao diện rõ ràng, dễ sử dụng.
- Lưu trữ và quản lý dữ liệu khuôn mặt, thông tin cá nhân và lịch sử giao dịch một cách hiệu quả.
- Hỗ trợ so khớp khuôn mặt người dùng với dữ liệu đã lưu để xác minh nhanh chóng và chính xác.
- Có khả năng tích hợp vào các nền tảng như ngân hàng, ví điện tử, cổng đăng nhập dịch vụ công, hoặc nền tảng thương mại điện tử.
- Cần có khả năng mở rộng để xử lý khối lượng người dùng lớn, hỗ trợ trích xuất báo cáo, thống kê và phát hiện hành vi bất thường.

1.3. Phạm vi và giới hạn

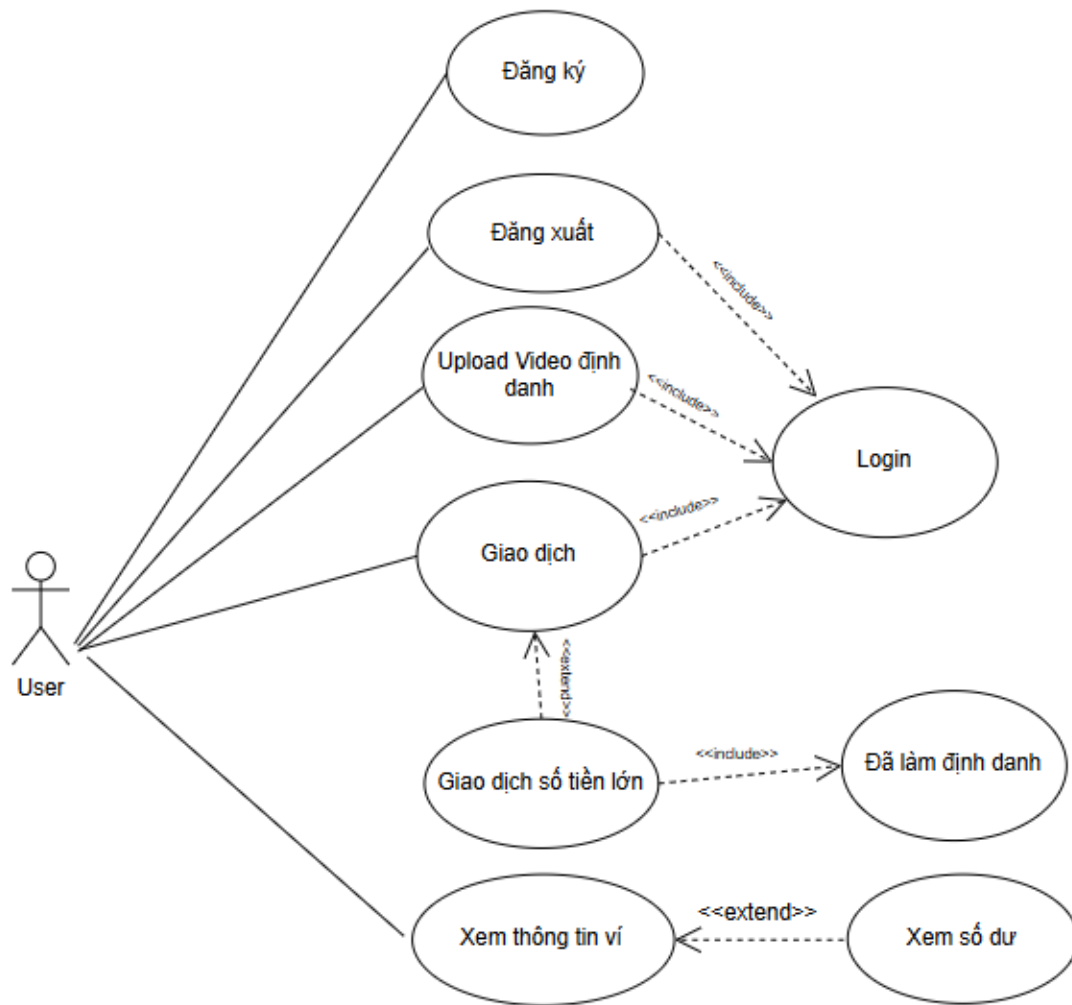
- Dữ liệu thu thập gồm video quay khuôn mặt của 10 người, chất lượng video tốt.
- Hệ thống chạy trên máy tính cá nhân.
- Sử dụng mô hình Facenet512 có sẵn, không thực hiện train lại mà lập trình để gửi dữ liệu đến mô hình, mã hóa lại và gửi dữ liệu đi.

CHƯƠNG 2: ĐẶC TẢ YÊU CẦU BÀI TOÁN

2.1. Biểu đồ use case tổng quan



Hình 1: Usecase trang quản lý (admin)



Hình 2: Usecase trang người dùng

2.2. Danh sách các tác nhân

STT	Tên Actor	Ý nghĩa/Ghi chú
1	Quản trị viên	Duyệt video định danh, Xử lý video định danh, cộng tiền, tìm kiếm người tương đồng bằng hình ảnh
2	Người dùng	Upload video định danh, giao dịch

2.3. Danh sách các Usecase

Mã UseCase	Tên UseCase	Ý nghĩa/Ghi chú
UC101	Đăng nhập vào hệ thống với phân quyền	Người dùng nhập thông tin đăng nhập. Hệ thống phân quyền: người dùng và admin sẽ được điều hướng đến giao diện riêng.
UC102	Đăng xuất	Người dùng nhấn nút đăng xuất, kết thúc phiên làm việc và trở về trang chủ.

UC103	Đăng ký tài khoản	Người dùng tạo tài khoản mới, hệ thống lưu vào cơ sở dữ liệu.
UC201	Upload video định danh	Người dùng chọn video quay khuôn mặt và upload lên hệ thống để yêu cầu định danh.
UC202	Xem danh sách video chờ duyệt	Admin nhấn vào nút xem danh sách video ở trạng thái chờ (waiting) từ người dùng.
UC203	Duyệt video quay khuôn mặt	Admin xét duyệt video, xóa bản ghi trong DB uploadVideos, người dùng vẫn chưa có embedding(để phân biệt với chưa duyệt thì đặt embedding lúc này là 1).
UC204	Xử lý video định danh	Admin nhấn nút xử lý, hệ thống trích xuất embedding từ video và cập nhật vào DB.
UC205	Tìm kiếm người tương đồng	Admin tải một bức ảnh lên và kiểm tra xem trong hệ thống có người nào tương đồng với khuôn mặt xuất hiện không
UC301	Kiểm tra thông tin ví	Hiển thị số dư, lịch sử giao dịch, và các chức năng quản lý ví.
UC302	Giao dịch	Người dùng chuyển tiền; nếu > 3 triệu phải xác thực khuôn mặt.
UC303	Cộng tiền cho người dùng	Admin (hoặc hệ thống) cộng tiền vào ví người dùng (nạp tiền, hỗ trợ).

- Chia danh sách usecase thành 3 nhóm:

- + Nhóm 1: Xác thực & phân quyền người dùng gồm: UC101, UC102, UC103.
- + Nhóm 2: Định danh bằng khuôn mặt gồm: UC201, UC202, UC203, UC204
- + Nhóm 3: Quản lý ví & giao dịch gồm: UC301, UC302, UC303

2.4. Đặc tả use case

- Thực hiện đặc tả đối với các UseCase chính : UC201, UC203, UC204, UC205, UC302

+ Các UseCase thực hiện bởi User: UC201, UC302

Mã UseCase: UC201	Tên UseCase: Upload video định danh		
Mục đích sử dụng	Tải video để yêu cầu định danh sinh trắc học.		
Tác nhân	User		
Tiền điều kiện	+ Người dùng đã đăng nhập. + Người dùng chưa có embedding(embedding= 0, chưa làm định danh)		
	STT	Thực hiện bởi	Hành động
	1	User	Người dùng truy cập giao diện tải video định danh.
	2	User	Người dùng chọn và

Luồng sự kiện chính			upload video.
	3	Hệ thống	Hệ thống lưu video lên Cloudinary.
	4	Hệ thống	Hệ thống thêm bản ghi vào bảng uploadVideos: <ul style="list-style-type: none"> phone = user.phone status = uploading
	5	Hệ thống	Khi video upload thành công: <ul style="list-style-type: none"> status = waiting link = URL trên Cloudinary
	6	Hệ thống	Hệ thống thông báo đã tải lên video thành công
Luồng phụ	Nếu upload lỗi: thông báo lỗi và không cập nhật DB.		
Hậu điều kiện	Bản ghi video ở trạng thái waiting.		

Mã UseCase: UC302	Tên UseCase: Giao dịch		
Mục đích sử dụng	Thực hiện giao dịch chuyển tiền trong nội bộ web		
Tác nhân	User		
Tiền điều kiện	Người dùng đã đăng nhập.		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	User	Người dùng nhập thông tin giao dịch: <ul style="list-style-type: none"> Số điện thoại người nhận (to) Số tiền chuyển (value) Thông điệp (message) tùy chọn
	2	Hệ thống	- Kiểm tra người gửi (from) có đủ balance \geq value không? <ul style="list-style-type: none"> Nếu không đủ thì chuyển sang luồng phụ [LP1] - Kiểm tra người nhận (to) có tồn tại trong bảng users không? <ul style="list-style-type: none"> Nếu không tồn tại thì thông báo lỗi và dừng giao dịch.
			- Kiểm tra nếu số tiền value vượt ngưỡng định danh:

	3	Hệ thống	<ul style="list-style-type: none"> Kiểm tra embedding của người gửi: <ul style="list-style-type: none"> Nếu embedding = 0 hoặc 1: Yêu cầu người dùng thực hiện định danh, dừng giao dịch. Nếu là mảng 512 chiều: <ul style="list-style-type: none"> Yêu cầu người dùng upload ảnh khuôn mặt để xác thực. So sánh ảnh mới với embedding trong DB: <ul style="list-style-type: none"> Nếu khớp thì tiếp tục giao dịch. Nếu không khớp thì chuyển sang luồng phụ [LP2]
	4	Hệ thống	Nếu xác thực thành công hoặc không cần xác thực: <ul style="list-style-type: none"> Trừ số tiền trong người dùng Cộng số tiền vào người nhận Ghi bản ghi mới vào bảng transactions
	5	Hệ thống	Hệ thống thông báo kết quả giao dịch thành công.
Luồng phụ	<ul style="list-style-type: none"> - [LP1] Không đủ số dư <ul style="list-style-type: none"> Hệ thống hiển thị thông báo lỗi số dư không đủ. Giao dịch không được thực hiện. Không ghi vào bảng transactions. - [LP2] Xác thực khuôn mặt thất bại <ul style="list-style-type: none"> Hệ thống thông báo xác thực không thành công Ghi bản ghi vào bảng transactions với status = 0. 		
Hậu điều kiện	<ul style="list-style-type: none"> - Nếu thành công: Giao dịch được lưu với status = 1, số dư cập nhật. - Nếu thất bại (do xác thực): Giao dịch lưu với status = 0. 		

+ Các UseCase thực hiện bởi Admin: UC203, UC204, UC205.

Mã UseCase: UC203	Tên UseCase: Duyệt video quay khuôn mặt		
Mục đích sử dụng	Kiểm duyệt video người dùng gửi lên.		
Tác nhân	Admin		
Tiền điều kiện	+ Admin đã đăng nhập.		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Admin	Admin truy cập danh sách các video có status = waiting.
	2	Admin	Admin xem video qua link.

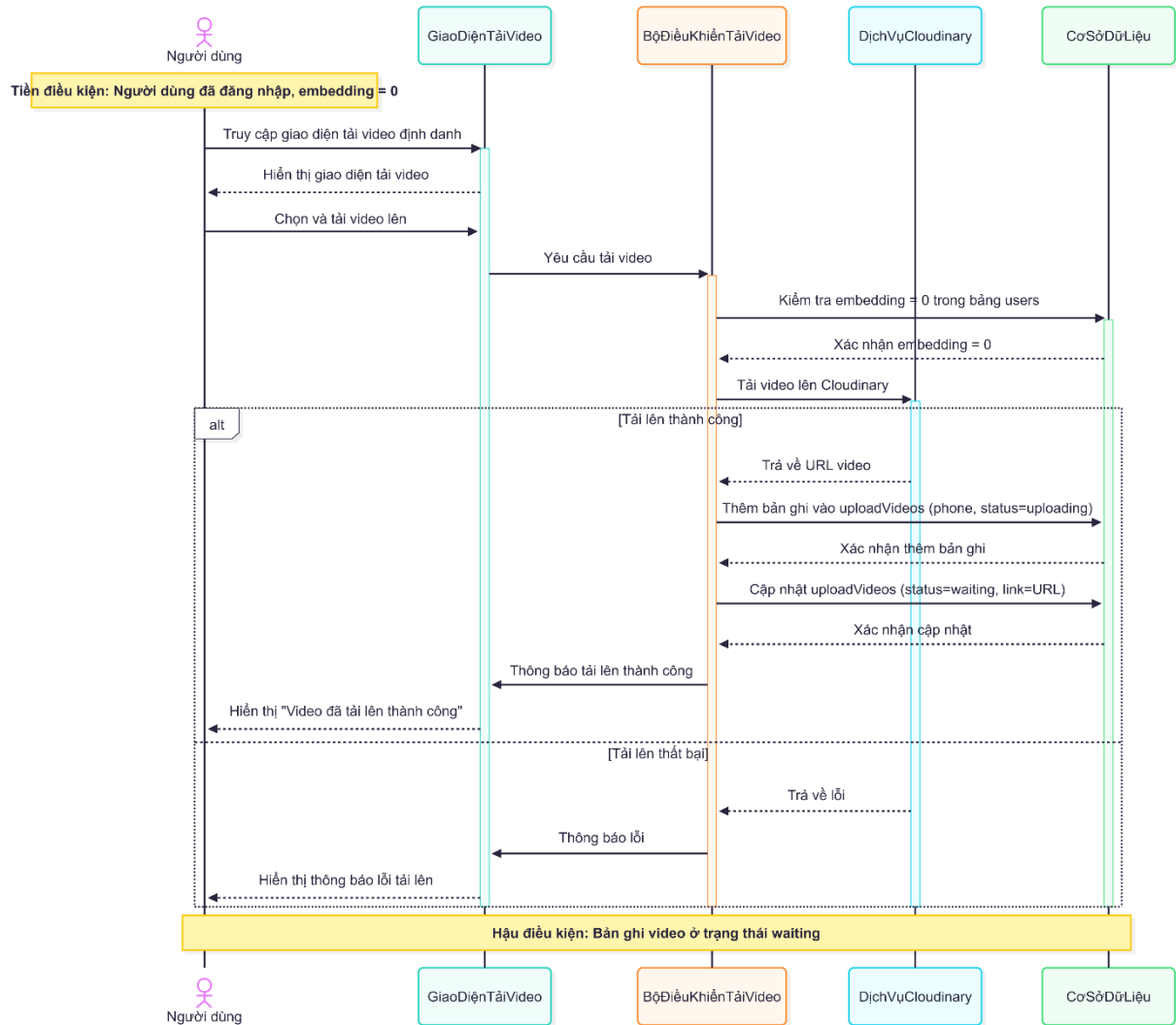
	3	Admin	Khi admin chọn "Duyệt"
	4	Hệ thống	+ Cập nhật uploadVideos.status = accepted + Đổi public id của file mp4 trên cloud từ uploaded/{users.phone} thành accepted/{users.phone} + Cập nhật users.embedding = 1
	5	Hệ thống	Video được đưa vào hàng chờ xử lý embedding.
Hậu điều kiện	Video được chấp nhận và chuẩn bị xử lý embedding.		

Mã UseCase: UC204	Tên UseCase: Xử lý video định danh		
Mục đích sử dụng	Trích xuất embedding khuôn mặt từ video đã được duyệt.		
Tác nhân	Admin		
Tiền điều kiện	Có người đang chờ xử lý (users.embedding = 1).		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Admin	Admin nhấn nút "Xử lý".
	2	Hệ thống	Hệ thống lấy các bản ghi users.embedding = 1
	3	Hệ thống	- Với mỗi video: <ul style="list-style-type: none"> Tải từ Cloundinary. Trích xuất khuôn mặt → vector 512 chiều. Ghi vào embedding.npy. - Sau khi cập nhật toàn bộ vào embedding.npy: Cập nhật tương ứng users.embedding = vector.
	4	Hệ thống	Thông báo hoàn tất.
Hậu điều kiện	+ embedding của user được cập nhật. + Bảng uploadVideos không còn bản ghi liên quan.		

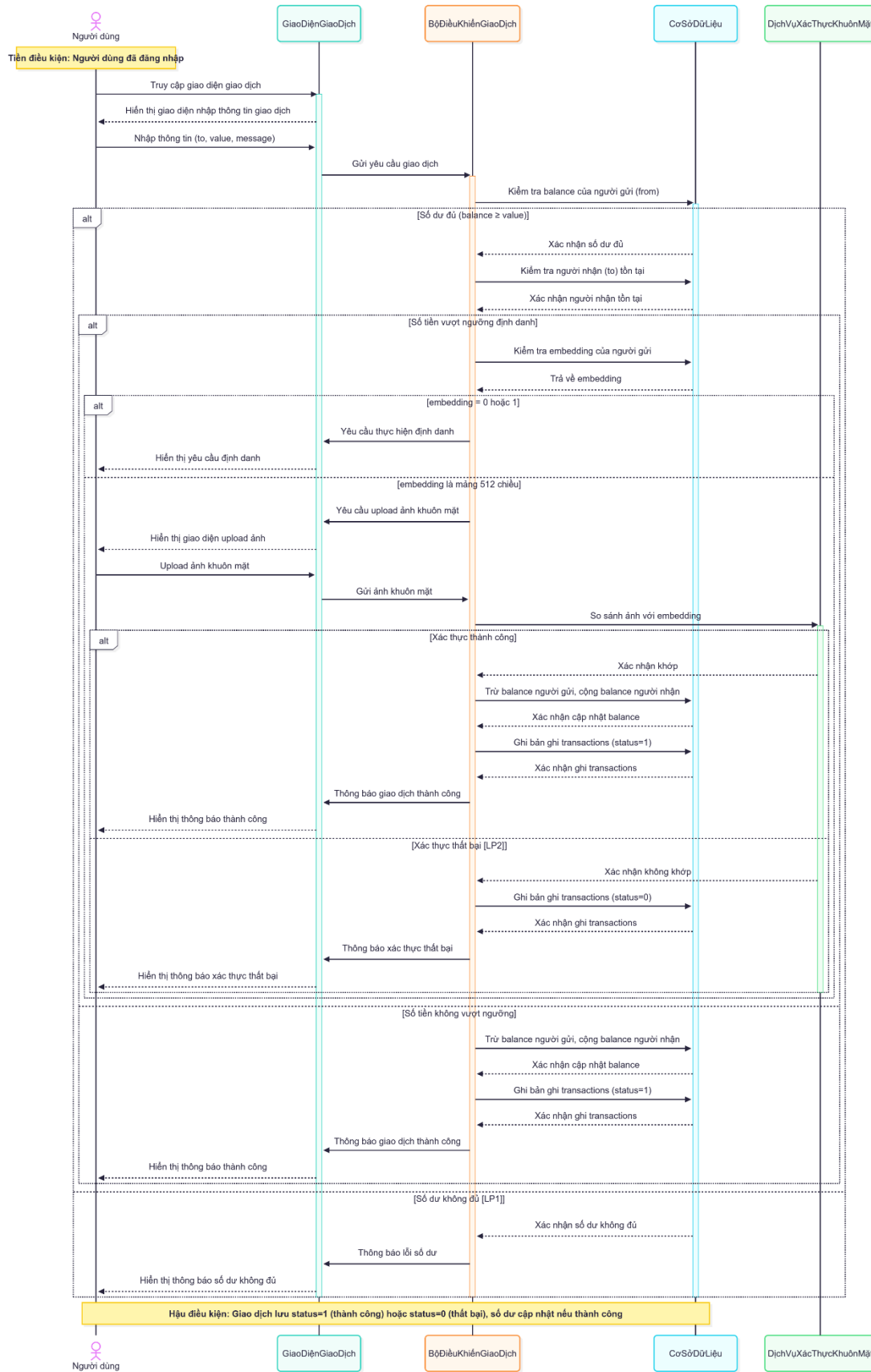
Mã UseCase: UC205	Tên UseCase: Tìm Kiếm Người Tương Đồng		
Mục đích sử dụng	Tìm người dùng có khuôn mặt giống ảnh được cung cấp.		
Tác nhân	Admin		
Tiền điều kiện	Admin đã đăng nhập		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Admin	Admin upload ảnh khuôn mặt.
	2	Hệ thống	Hệ thống trích xuất vector embedding từ ảnh.
	3	Hệ thống	Tìm tất cả người dùng có embedding 512 chiều.
	5	Hệ thống	Tính độ tương đồng (cosine similarity).
	6	Hệ thống	Trả về người dùng có embedding tương đồng cao nhất(>85%), nếu không có thì thông báo không tìm thấy
Hậu điều kiện	Hiển thị thông tin người dùng có khuôn mặt giống, gồm tên và số điện thoại.		

2.5. Mô hình hóa hành vi

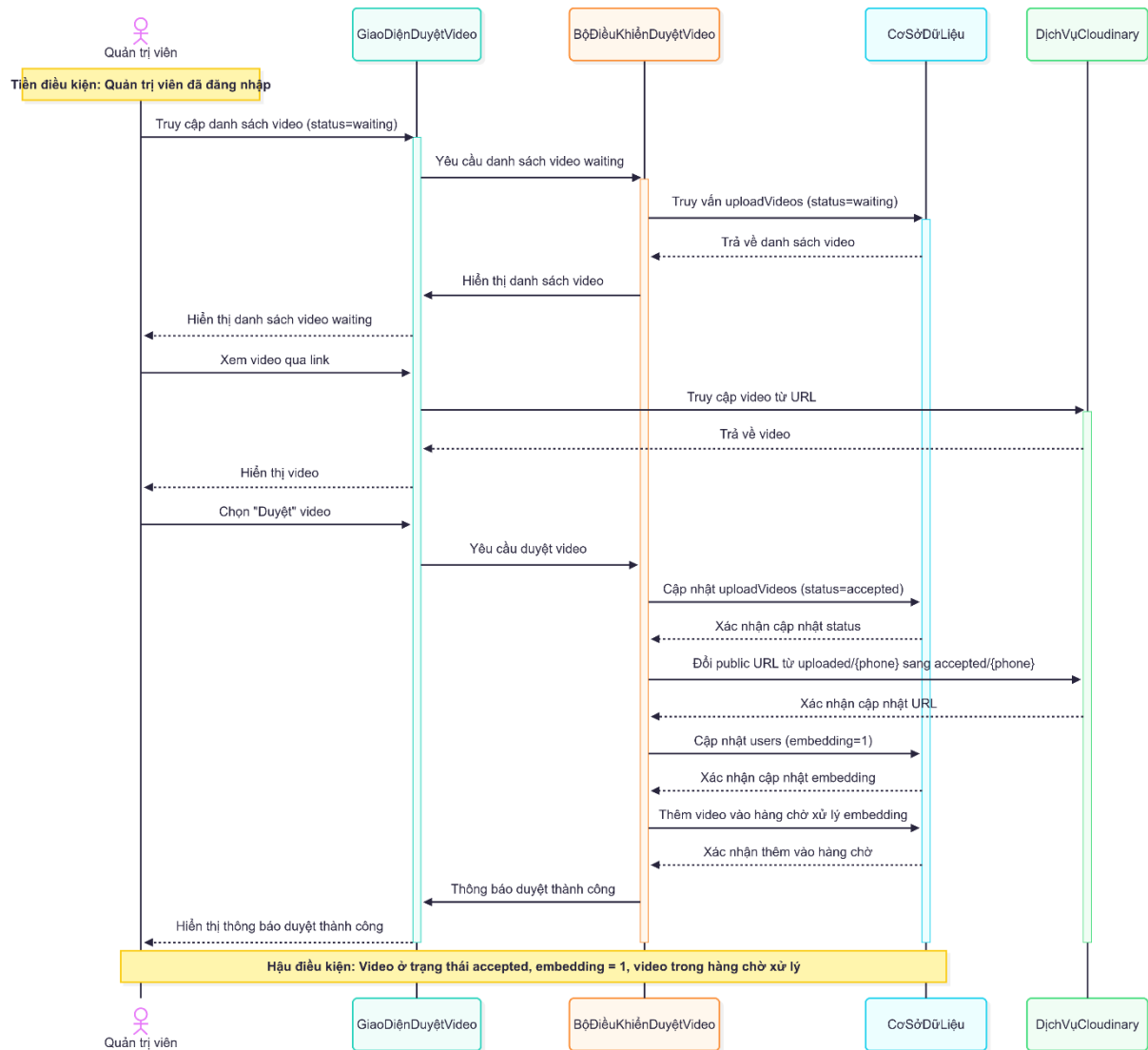
- Mô hình hóa UC201: Upload video định danh



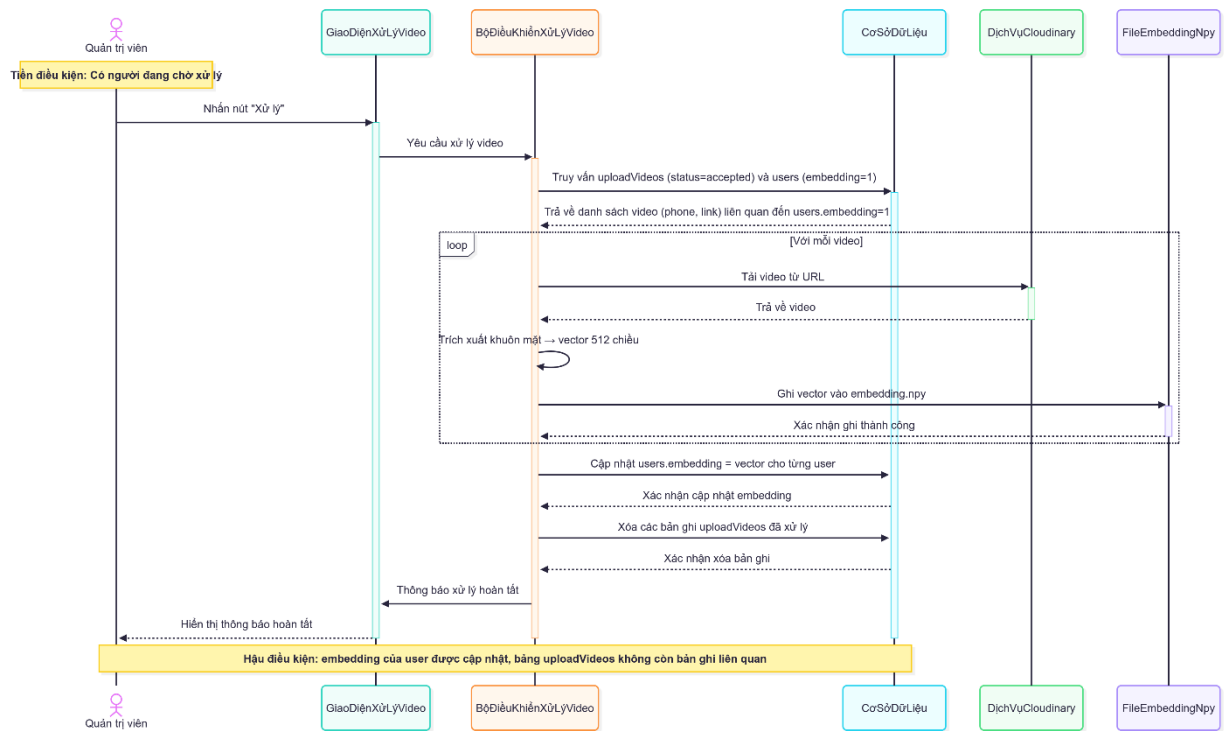
-Mô hình hóa UC302: Giao dịch



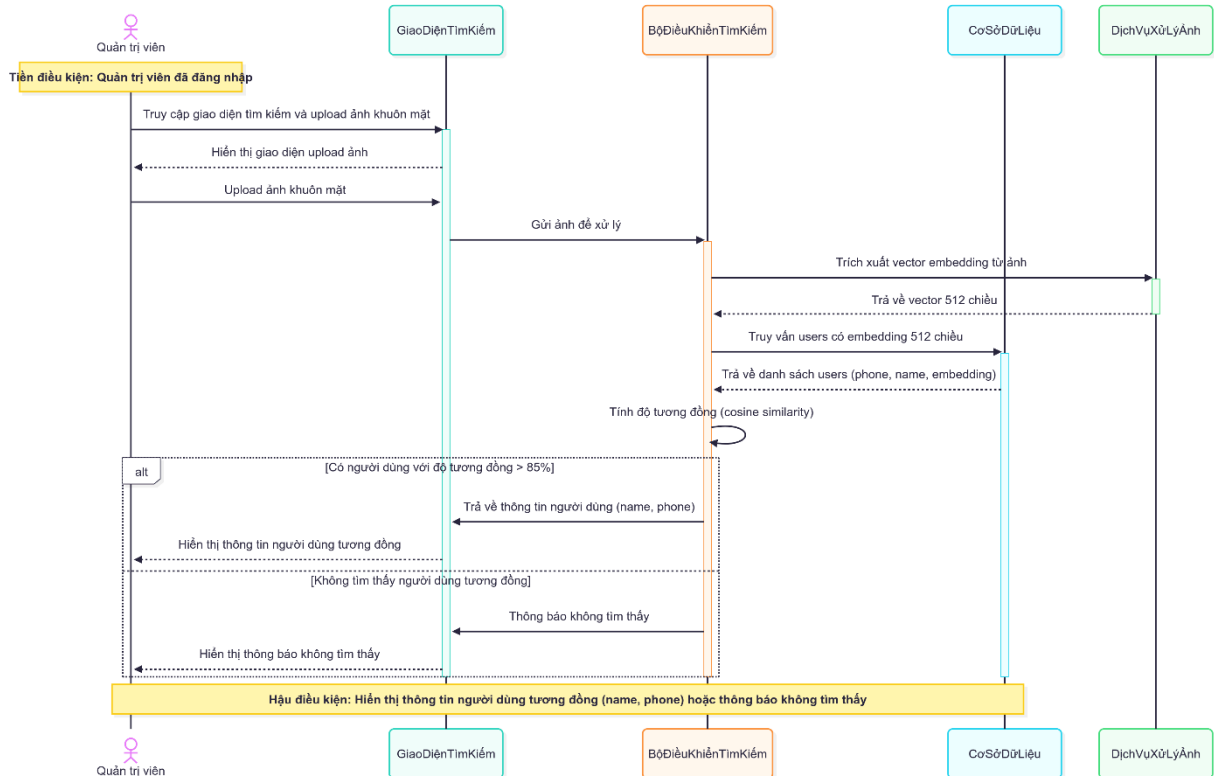
- Mô hình hóa UC203: Duyệt video quay khuôn mặt



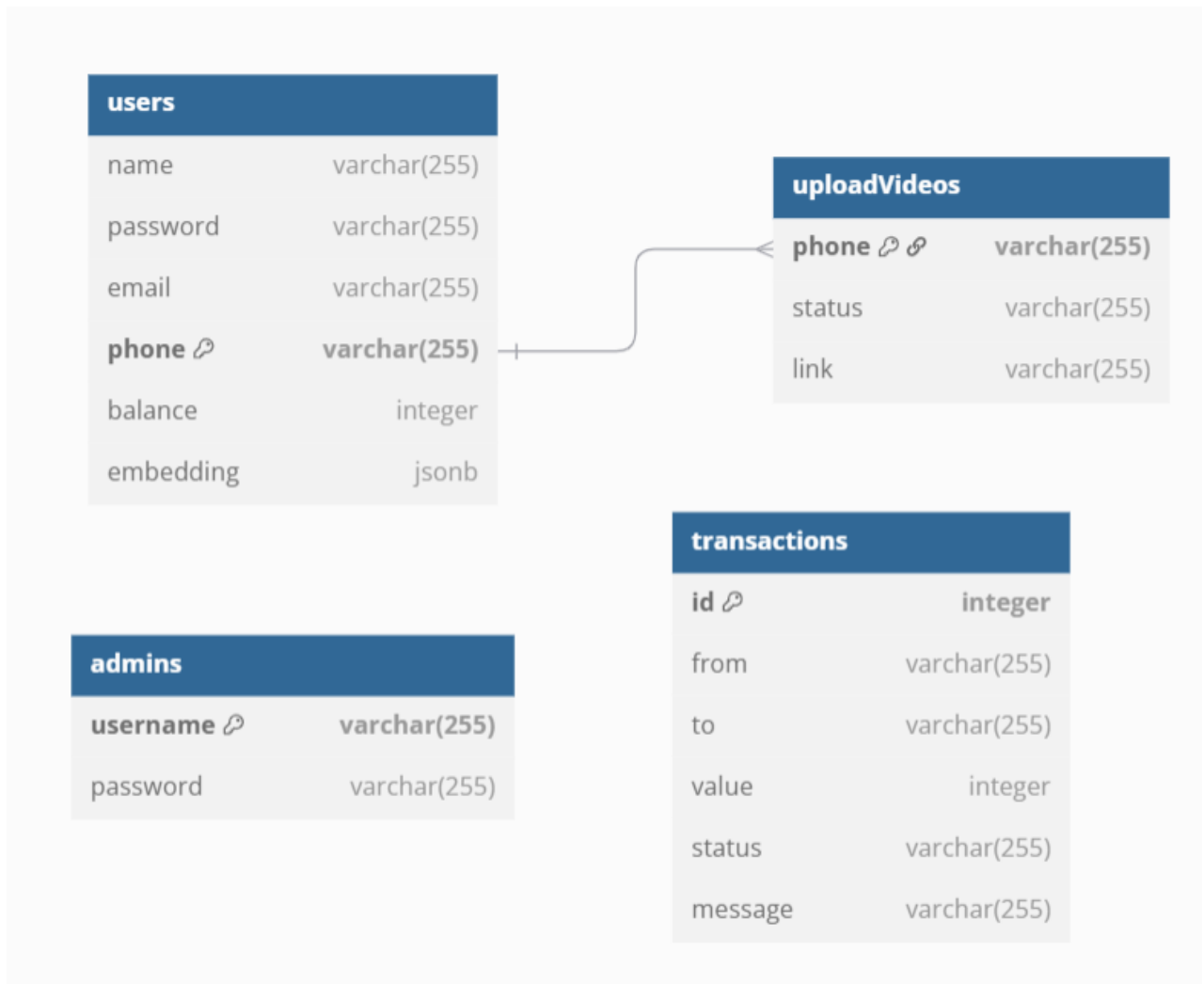
- Mô hình hóa UC204: Xử lý video định danh



- Mô hình hóa UC205: Tìm Kiếm Người Tương Đồng



CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU



Sơ đồ quan hệ cơ sở dữ liệu

- Bảng users:

- + name: tên người dùng
- + password: mật khẩu được băm bằng bcrypt trong Node.js
- + email
- + balance: số dư tài khoản
- + embedding: mảng 512 giá trị được trích xuất từ khuôn mặt người dùng(nếu người dùng mới đăng kí thì embedding có giá trị 0, đang chờ duyệt video định danh là 1, sau khi định danh là mảng)
- + phone: số điện thoại đăng kí, cũng là tên đăng nhập, id chính của người dùng

- Bảng admins:

- + username: tên tài khoản
- +password: mật khẩu

- Bảng uploadVideos:

- + phone: id của người dùng upload video lên hệ thống

+ status: khi người dùng mới bấm tải video lên thì để ở trạng thái uploading (vì hệ thống nhỏ nên chỉ cho 1 video uploading tại một thời điểm), sau khi upload lên server sẽ là waiting(đợi admin duyệt, lúc này vẫn chưa có embedding cho user này)
 + link: khi status là waiting thì có đường dẫn đến clouidary là video người dùng để admin kiểm tra

- Bảng transactions:

- + id: khóa chính
- + from: số điện thoại người chuyển
- + to: số điện thoại người nhận
- + value: số tiền chuyển
- + status: trạng thái sau khi chuyển(1 nếu thành công, 0 nếu thất bại)
- + message: thông điệp truyền đi

CHƯƠNG 4: THIẾT KẾ KIẾN TRÚC

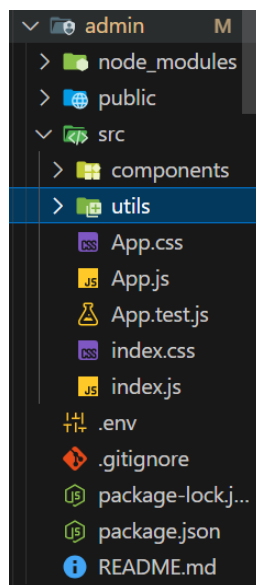
4.1. Kiến trúc hệ thống

Thành phần	Diễn giải
Client	Nơi hệ thống tương tác với người dùng
Server	Nơi xử lý yêu cầu từ client và thực hiện logic nghiệp vụ
Data	Nơi chứa dữ liệu của hệ thống

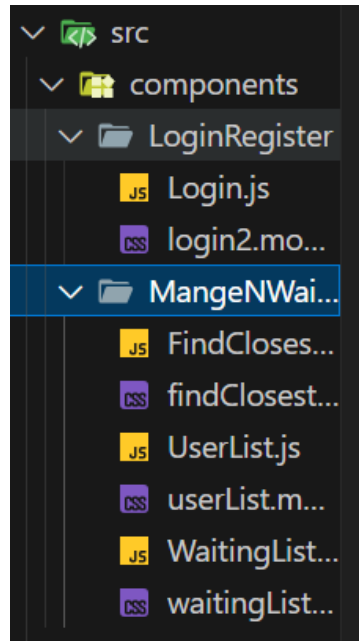
4.2. Mô tả chi tiết từng thành phần trong hệ thống

Ứng dụng được thiết kế theo mô hình MVC phân tách giữa Frontend và Backend gồm:

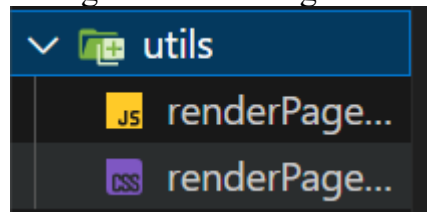
- Frontend: Sử dụng cơ chế React Components
 - Trang quản lý (admin)



1. components: Chứa các thẻ quan trọng gây dựng lên trang web

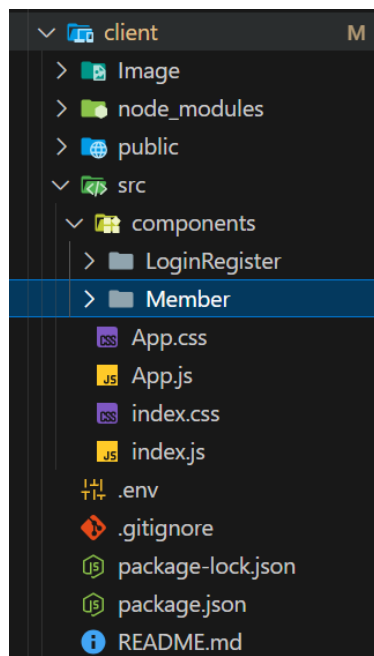


2. utils: chứa hàm hỗ trợ được dùng nhiều lần trong web

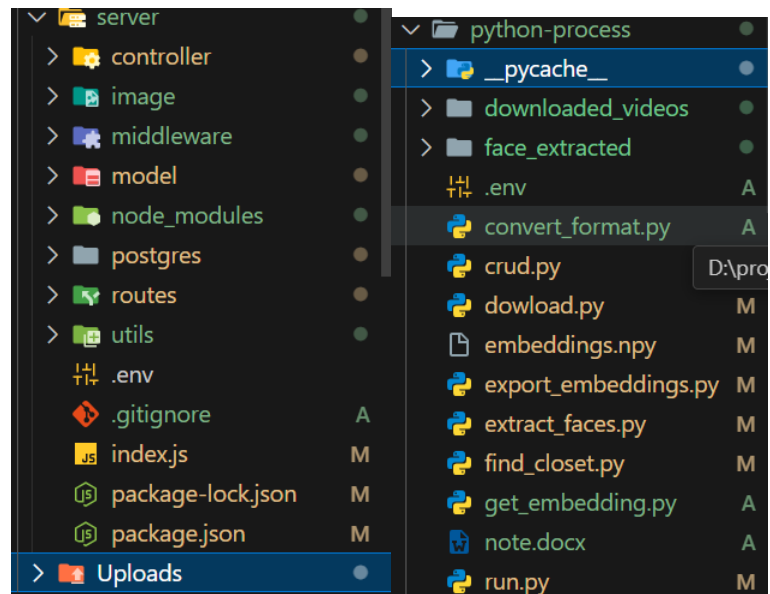


3. App.js: hàm quan trọng chứa bố cục của toàn trang web

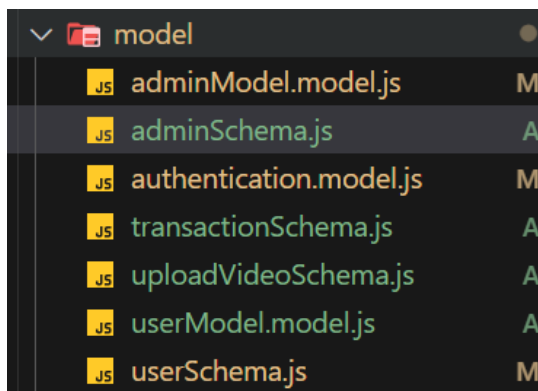
- Trang user:



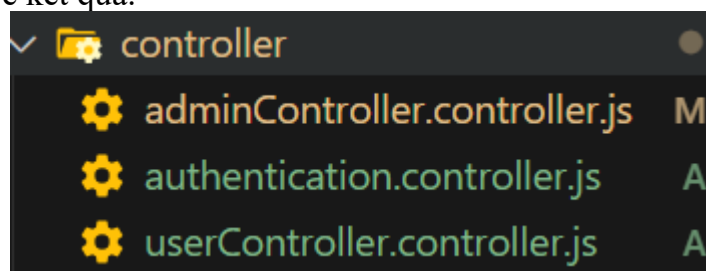
- Backend:
Gồm hai phần là một phần xử lý backend đối với web: mvc; và một phần là tiến trình python xử lý các tác vụ như xử lý ảnh, video của người dùng, sử dụng thư viện deepface để trích xuất đặc trưng khuôn mặt



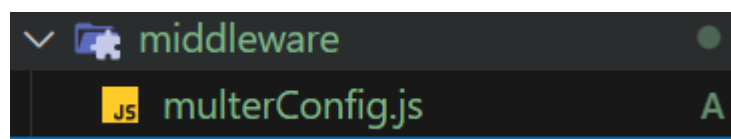
1. Model: để quản lý dữ liệu, xử lý logic nghiệp vụ và truy xuất dữ liệu từ cơ sở dữ liệu, ngoài ra còn tạo bảng với sequelize..



2. Controller: để điều hướng luồng xử lý, nhận yêu cầu từ Frontend, điều phối Model và trả về kết quả.

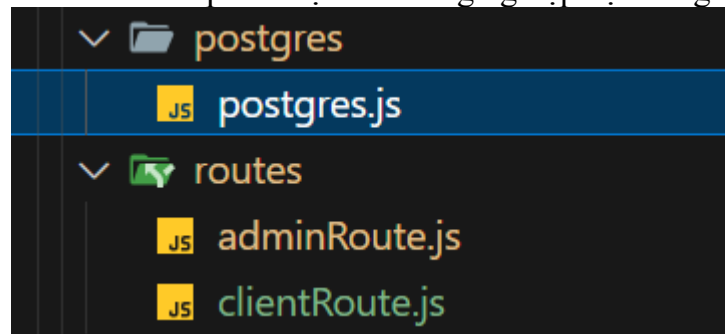


3. Middleware: middleware nhận file gửi lên từ client qua HTTP request

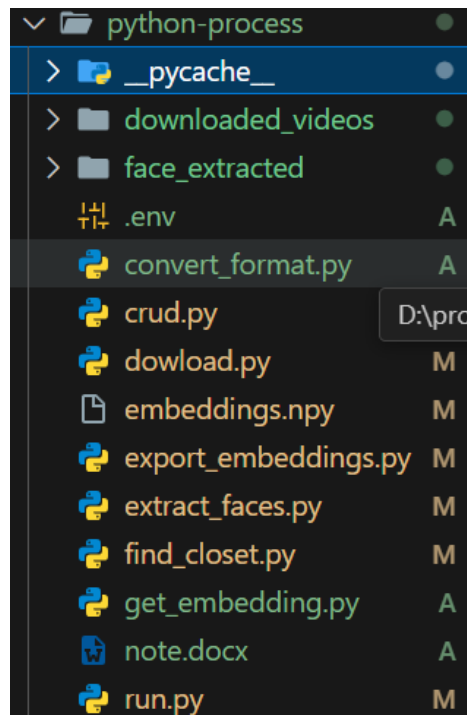


4. Các phần còn lại của backend web

- + postgres: để cấu hình sequelize giúp tương tác với cơ sở dữ liệu quan hệ thông qua mã JavaScript thay vì phải viết SQL thủ công.
- + Route: chứa các API phân loại theo từng nghiệp vụ tương ứng



5. Tiến trình python



- Tiến trình python được gọi từ controller của backend web, bằng child-process trong react, mọi sự kiện được tổng hợp vào file run.py, khi gọi sẽ truyền các tham số như danh sách id để lấy embedding, tên file con chạy trong run,...
 - + crud: thêm, sửa, xóa dữ liệu embedding của người dùng (đã triển khai thêm và xóa) vào file local embeddings.npy (file này chứa thông tin như dictionary với key là số điện thoại của người dùng, value là mảng 512 phần tử, file này dùng cho tiến trình python và để đọc và up lên cơ sở dữ liệu)
 - + download.py để tải video người dùng từ cloud về máy local của admin
 - + extract_faces.py để cắt lấy khuôn mặt trong video sử dụng mtcnn
 - + find_closest: nhận vào một ảnh và từ điển có key là phone, value là embedding (chỉ nhận mảng) tức là chỉ lấy ra những người trên cơ sở dữ liệu mà embedding là mảng 512 phần tử làm từ điển để so khớp với embedding từ một ảnh mới.
 - export_embeddings: đọc file embeddings.npy từ đó parse sang json, chọn id để gửi lên cơ sở dữ liệu

- **Xử lý ảnh bằng FaceNet512 thuộc thư viện DeepFace:**

- Các bước xử lý khuôn mặt trong Facenet512

Bước 1: Phát hiện khuôn mặt (Face Detection)

Dùng MTCNN, RetinaFace hoặc detector khác.

Cắt ra vùng khuôn mặt từ ảnh gốc.

Bước 2: Tiền xử lý khuôn mặt

Resize ảnh về kích thước cố định (160x160).

Bước 3: Trích xuất vector đặc trưng (Embedding)

Mô hình FaceNet512 biến ảnh khuôn mặt thành một vector 512 chiều (embedding).

- Vector đặc trưng (Face Embedding)

- Là đại diện toán học cho khuôn mặt.
- Các khuôn mặt giống nhau thì vector gần nhau.
- Các khuôn mặt khác nhau thì vector cách xa nhau.

- Cách thức so khớp khuôn mặt

$$\text{similarity}(x_1, x_2) = \frac{x_1 \cdot x_2}{\|x_1\| \cdot \|x_2\|}$$

Trong đó:

- $x_1 \cdot x_2$: tích vô hướng
- $\|x_1\|$: chuẩn vector x_1
- Kết quả nằm trong khoảng $[-1, 1]$
 - Gần 1 \rightarrow rất giống
 - Gần 0 \rightarrow không liên quan
 - Gần -1 \rightarrow đối lập
- Khi dùng hàm cosin hệ thống trả về $\text{cosine_distance} = 1 - \text{similarity}$ nên cosine_distance sẽ nằm trong đoạn 0 đến 2, trong đó càng gần 0 thì độ tương đồng càng cao. Nếu độ giống đủ cao (theo một ngưỡng threshold định trước), ta kết luận là cùng người, Facenet512 + cosine: ngưỡng thường $\sim 0.25 - 0.4$ nên ở đây em chọn 0.3 là ngưỡng tối đa (nếu càng nhỏ thì càng tương đồng).

CHƯƠNG 5: THIẾT KẾ GIAO DIỆN

5.1. Thiết kế giao diện người dùng

- Để sử dụng:

- khi người dùng ghé thăm thì cần đăng nhập:

localhost:3000/login

New conversation ... Sản Điểm rèn luyện... Laban Dictionary - T... FULL tài liệu cntt trư... Bỏ sung: Bảng tron... Bài tập cu

Đăng Nhập

Tên khoản:

Mật khẩu:

Đăng ký

Gửi

Sau khi đăng nhập, vào được trang chủ:

Thông tin tài khoản Thông tin ví Đăng xuất

Thông tin tài khoản

★ Họ tên:

★ Email:

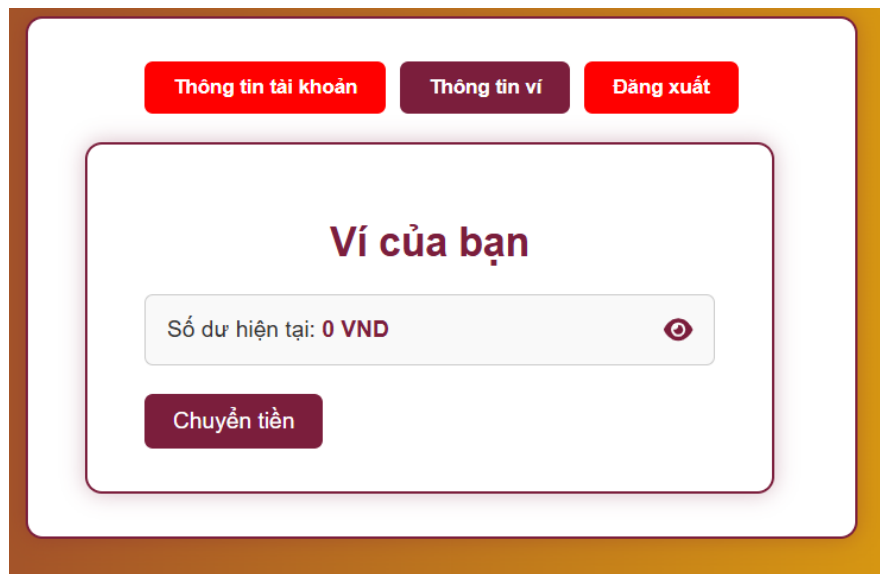
★ Số điện thoại:

Định danh:

Choose File No file chosen

Upload

Có thể xem thông tin ví để thực hiện giao dịch:



Muốn làm định danh thì người dùng chỉ cần tải video của mình lên bằng cách ấn vào choose file rồi upload, sau đó chờ admin duyệt

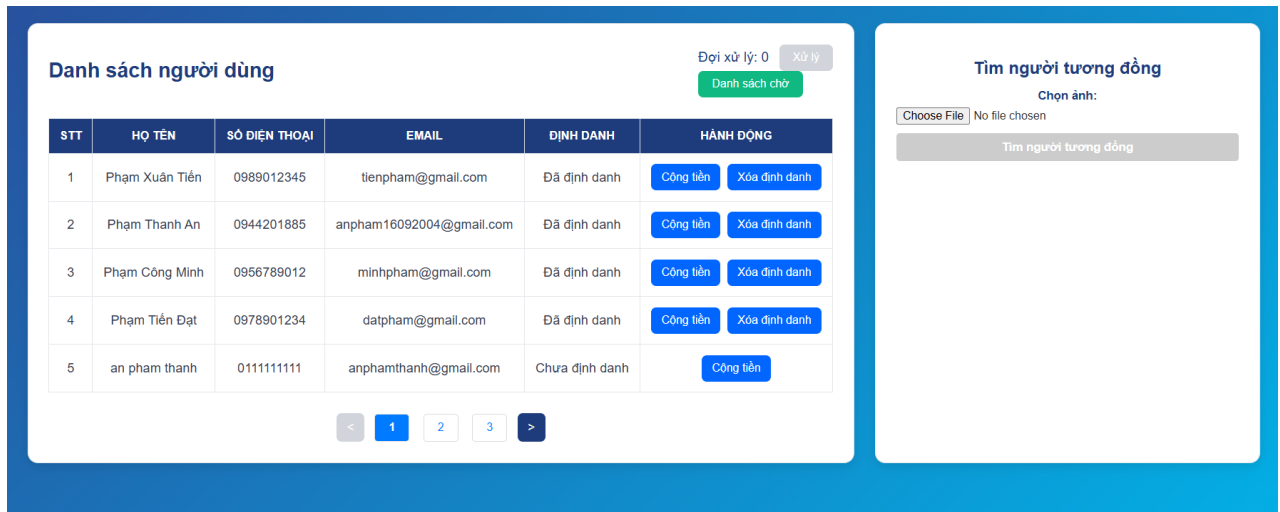
Định danh:

No file chosen

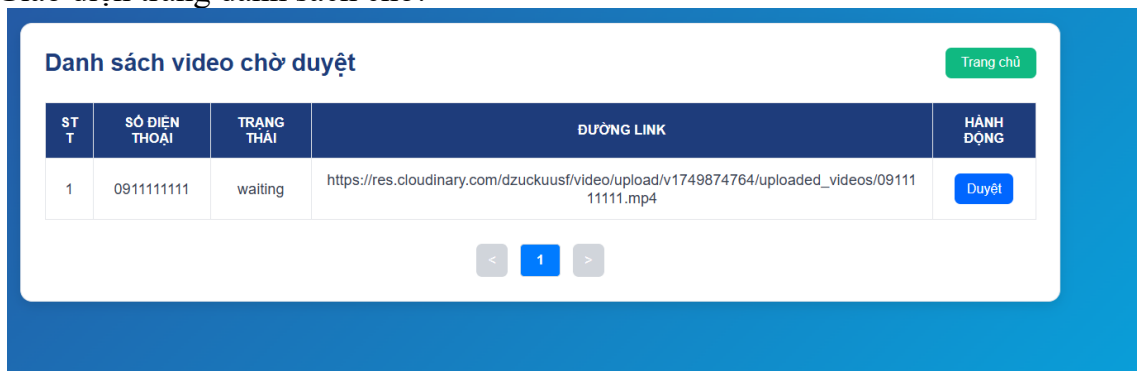
5.2.Thiết kế giao diện admin

- Trang admin cũng được thiết kế thân thiện dễ dùng:
 - Giao diện trang admin(đăng nhập):

- Giao diện trang chủ hiển thị danh sách người dùng và phần tìm kiếm
Có thể ấn vào nút danh sách chờ để chuyển qua trang danh sách chờ



- Giao diện trang danh sách chờ:



- Công nghệ sử dụng:

- **ReactJS:** Xây dựng giao diện người dùng, chia các trang thành các components. mỗi component là 1 thẻ đại diện cho 1 chức năng, dễ quản lý, dễ sửa đổi
- **DeepFace:** thư viện cấp cao được xây dựng trên nền các framework như TensorFlow, Keras, PyTorch, tích hợp sẵn nhiều mô hình học sâu mạnh mẽ để: nhận diện người trong ảnh, trích xuất embedding khuôn mặt, so sánh hai khuôn mặt,...
- **NodeJS:** backend dùng JavaScript để xây dựng máy chủ và API hiệu quả, kết hợp thêm với axios, cloudinary(nền tảng cloud), middleware multer

- Các chức năng chính:

- + Trang người dùng:
 - Upload video quay khuôn mặt để làm định danh
 - Xem thông tin ví, thực hiện giao dịch, xác minh danh tính
- + Trang quản lý:
 - Chấp nhận yêu cầu định danh
 - Xử lý video định danh
 - Cộng tiền cho người dùng

- Tìm kiếm người tương đồng bằng ảnh chân dung
- **Tích hợp API:** Frontend giao tiếp với RESTful API từ Backend để hiển thị và cập nhật dữ liệu.

CHƯƠNG 6: Xây dựng và thử nghiệm

6.1. Xây dựng

Yêu cầu hệ thống

- **Phần mềm cần thiết:** các phần mềm hoặc công cụ phát triển: Node.js, npm, ReactJS, PostgreSQL.
- **Cấu hình hệ thống:** Các yêu cầu phần cứng hoặc cấu hình tối thiểu của máy chủ, trình duyệt hỗ trợ.

Cài đặt môi trường phát triển

- **Bước 1:** Cài đặt Node.js và npm (nếu sử dụng ReactJS hoặc các công cụ tương tự).
- **Bước 2:** Cài đặt các gói và thư viện cần thiết (ví dụ: chạy npm install trong thư mục dự án).
- **Bước 3:** Thiết lập cơ sở dữ liệu
- **Bước 4:** Cấu hình môi trường (các biến môi trường như API keys, các tham số cần thiết cho việc phát triển hoặc chạy thử).
- **Bước 5:** Chạy thử ứng dụng trên môi trường cục bộ (ví dụ, chạy lệnh npm start và kiểm tra ứng dụng trên trình duyệt).

Cài đặt môi trường lưu trữ

- Cấu hình cloudinary, lấy api key, api secret, cloud name, cài đặt preset để lưu trữ video upload lên

6.2. Thử nghiệm

a, Thử nghiệm chức năng trang người dùng

- + Upload video quay khuôn mặt: video được up thành công lên cloudinary với public_id là uploaded/{phone}
- + Xem thông tin ví: hiển thị đúng số dư
- + Giao dịch tiền: giao dịch thành công, cập nhật số dư, nếu giao dịch hơn 5 triệu thì cần xác thực
- + Xác minh danh tính bằng ảnh chụp từ camera: nhận diện khuôn mặt thành công hoặc báo lỗi không phù hợp

b, Thử nghiệm chức năng trang quản lý

- Duyệt yêu cầu định danh: Cập nhật thành công trạng thái người dùng
- Xử lý video định danh: trích xuất khuôn mặt tạo embedding thành công
- Cộng tiền cho người dùng: cập nhật số dư chính xác
- Tìm người tương đồng bằng ảnh: trả về thành công người có độ tương đồng cao nhất hoặc không tìm thấy nếu độ tương đồng không đạt trên 85%

c, Thử nghiệm tích hợp

- Kiểm tra việc giao tiếp API giữa frontend và backend thông qua Axios.
- Kiểm tra upload video lên Cloudinary thành công.
- Kiểm tra bảo mật: middleware kiểm tra quyền truy cập hợp lệ.
- Kiểm tra tốc độ phản hồi và tính ổn định trong nhiều phiên bản trình duyệt

CHƯƠNG 7: TỔNG KẾT

7.1. Đánh giá kết quả thu được

7.1.1. Ưu điểm

- Hệ thống sử dụng DeepFace để nhận diện khuôn mặt nhanh chóng.
- Upload và xử lý video qua Cloudinary giúp lưu trữ dữ liệu lớn, dễ dàng mở rộng quy mô.
- Tìm kiếm người tương đồng bằng ảnh: tính năng truy xuất khuôn mặt có độ tương đồng cao giúp hỗ trợ điều tra hoặc xác thực danh tính phụ.

7.1.2. Hạn chế

- Phụ thuộc vào kết nối internet: hệ thống cần mạng ổn định để upload video và tương tác với Cloudinary hoặc backend.
- Tốc độ xử lý phụ thuộc vào thiết bị và dung lượng video: các video lớn có thể làm chậm quá trình xử lý hoặc nhận diện khuôn mặt (có thể giới hạn dung lượng video được upload lên).
- Khả năng mở rộng chưa tối ưu: với số lượng lớn video và dữ liệu embedding, cần cải thiện cơ sở dữ liệu và hiệu năng xử lý trong tương lai.
- Độ chính xác phụ thuộc vào chất lượng video và môi trường ánh sáng: nếu video mờ hoặc thiếu sáng, mô hình nhận diện có thể cho kết quả sai lệch do đó cần quản trị viên kiểm tra kỹ lưỡng.

7.1.3. Những kinh nghiệm và kiến thức đã đạt được

Phân tích và thiết kế hệ thống

- Tầm quan trọng của Use Case và Sequence Diagram: Việc xây dựng các Use Case rõ ràng giúp nhóm hiểu rõ vai trò, chức năng của từng thành viên hệ thống.
- Cần tối ưu cấu trúc cơ sở dữ liệu (Database Design): Các quan hệ (entities) phải rõ ràng, tránh dư thừa hoặc trùng lặp.

Phát triển Backend

- Các API phải được xây dựng có xác thực và phân quyền.

Phát triển FrontEnd

- Cần xây dựng giao diện dễ dàng sử dụng ấn tượng tốt với người dùng.

Tích hợp Công cụ AI/Deep Learning

- Có thể tích hợp thư viện deep learning tùy ý bằng cách xử lý tách rời từng bộ phận và được giao tiếp với api từ backend.

7.2. Hướng phát triển

Dưới đây là một số hướng phát triển thêm cho web dịch vụ định danh nhằm nâng cao trải nghiệm người dùng và tối ưu hóa quy trình vận hành:

1. Tăng cường bảo mật: tích hợp các cơ chế xác thực 2 bước, mã hóa dữ liệu người dùng, giao dịch
2. Tối ưu hóa tốc độ xử lý video: nâng cao hạ tầng xử lý video bằng cách sử dụng GPU hoặc các nền tảng Cloud trả phí như Google Cloud AI, AWS để rút ngắn thời gian xử lý

3. Cải thiện độ chính xác nhận diện khuôn mặt: fine-tune các mô hình DeepFace với dữ liệu thực tế hoặc các mô hình khác, xây dựng mô hình kiểm tra thật giả,...
4. Thêm chức năng thông báo: cung cấp hệ thống thông báo(quia email hoặc SMS) cho người dùng khi định danh thành công hoặc nạp tiền vào ví.
5. Hỗ trợ nhiều định dạng đầu vào: ngoài video, cho phép người dùng quay trực tiếp từ webcam trong hệ thống.

7.3. Kết thúc

Qua quá trình xây dựng và thử nghiệm, hệ thống đã cơ bản hoàn thiện các chức năng chính như định danh khuôn mặt, quản lý ví và xử lý video hiệu quả. Tuy vẫn còn hạn chế, em đã rút ra được nhiều kinh nghiệm thực tế về lập trình web, phân tích thiết kế hệ thống, xử lý dữ liệu đa phương tiện và tích hợp thư viện AI. em xin cảm ơn thầy Nguyễn Bá Ngọc đã tận tình hướng dẫn, góp ý, định hướng cho em trong quá trình thực hiện đồ án. Em cũng mong nhận được những đóng góp đánh giá của thầy về bài tập lớn này để em có thể cải thiện hơn trong tương lai.