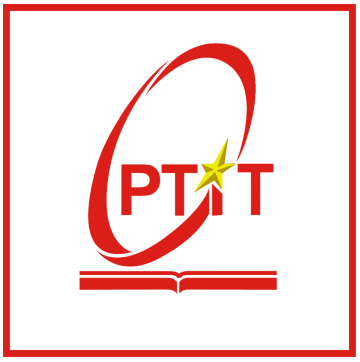
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

* * * * * * * * * * * * *



BÀI TẬP LỚN IOT VÀ ỨNG DỤNG

BÁO CÁO CUỐI KỲ

Đề tài: Hệ thống tủ thông minh

Giảng viên hướng dẫn: TS. Kim Ngọc Bách

Lớp chuyên ngành: D22CNPM02 - Nhóm BTL: 9

Thành viên	Mã sinh viên
Trần Mai Hương	B22DCCN424
Nguyễn Thị Khánh Vân	B22DCCN892
Nguyễn Nam Vũ	B22DCCN916
Nguyễn Thị Yến	B22DCCN928

Hà Nội, Tháng 11/2025

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU CHUNG

1.1. Lý do chọn đề tài

- Nhu cầu bảo mật và tự động hóa trong gia đình, văn phòng, ký túc xá, phòng làm việc, trung tâm thương mại ngày càng lớn.
- Người dùng có thể giám sát từ xa hoặc được cảnh báo khi có hành vi mở khóa trái phép.
- Công nghệ nhận diện khuôn mặt và IoT hiện nay cho phép xây dựng các hệ thống an ninh nhỏ gọn, giá rẻ nhưng hiệu quả cao.
- Hệ thống loại bỏ rủi ro mất chìa khóa, bị sao chép thẻ RFID, đồng thời có khả năng cảnh báo nguy hiểm.

1.2. Tổng quan về dự án

- Trong thời đại công nghệ 4.0, nhu cầu về các thiết bị thông minh, tiện lợi và an toàn ngày càng tăng. Hệ thống tủ thông minh nhận diện khuôn mặt giúp người dùng mở khóa mà không cần chìa, đồng thời có thể điều khiển và giám sát từ xa qua Internet.
- Đề tài này sử dụng ESP32-CAM – một vi điều khiển có camera tích hợp và khả năng kết nối Wi-Fi, để nhận diện khuôn mặt, mở khóa tủ và gửi cảnh báo trong các tình huống bất thường.

1.3. Mục đích của dự án

- Xây dựng hệ thống tủ thông minh có khả năng nhận diện khuôn mặt để mở khóa tự động.
- Cho phép điều khiển đóng/mở tủ từ xa
- Tích hợp tính năng cảnh báo khi phát hiện người lạ hoặc có hành vi truy cập trái phép
- Hệ thống hoạt động ổn định, chi phí thấp, dễ dàng mở rộng**.**

1.4. Đối tượng và phạm vi nghiên cứu

- Đối tượng:
    + 1 module ESP32-CAM (camera và xử lý trung tâm).
    + 1 servo motor để điều khiển cơ cấu đóng/mở tủ.
    + 1 cảm biến chuyển động PIR.
    + Nguồn cấp: lấy từ cổng USB máy tính hoặc adapter 5V.
- Phạm vi nghiên cứu:
    + Phạm vi kết nối:
        ○ Hệ thống hoạt động qua Wi-Fi để điều khiển và gửi cảnh báo.
    + Phạm vi môi trường
        ○ Áp dụng cho tủ cá nhân, tủ văn phòng, ký túc xá, trung tâm thương mại hoặc không gian trong nhà.
        ○ Mô hình demo có thể mở rộng thành hệ thống nhiều tủ có quản lý trung tâm
    + Tính năng:
        ○ Tự động hóa mở khóa tủ thông qua nhận diện khuôn mặt → nâng cao tính bảo mật và tiện lợi cho người dùng.
        ○ Cho phép điều khiển tủ từ xa → người dùng có thể mở tủ hoặc kiểm tra trạng thái dù không có mặt trực tiếp.

    - Tự động gửi cảnh báo an ninh khi phát hiện hành động xâm nhập hoặc nhận diện sai khuôn mặt.
    - Tự động phát hiện chuyển động và nhận diện khuôn mặt.

1.5. Phương pháp nghiên cứu

- Phương pháp nghiên cứu lý thuyết: Tìm hiểu và nghiên cứu các công nghệ phần mềm, AI, công nghệ phần cứng, cách sử dụng thiết bị:
    + Phần cứng: vi điều khiển ESP32-CAM, cảm biến chuyển động PIR, servo 90.
    + Phần mềm: lập trình nhúng (Firmware), backend, frontend
    + AI: YOLOFace, ArcFace
- Phương pháp thực nghiệm: Tiến hành thiết kế, lắp đặt, cấu hình hệ thống thực tế để đánh giá độ chính xác của hệ thống, độ tin cậy trong việc thu thập dữ liệu và tính khả dụng của hệ thống trong các điều kiện hoạt động khác nhau.
- Phương pháp phân tích: Sử dụng các công cụ phân tích để xử lý dữ liệu thu được từ hệ thống đo lường. Đánh giá và so sánh kết quả thực nghiệm với các mô hình lý thuyết đã được nghiên cứu để xác định hiệu quả và tính chính xác của hệ thống. Đồng thời, tiến hành điều chỉnh cần thiết để hoàn thiện hệ thống

1.6. Công nghệ và thiết bị cần thiết

- Phần cứng:
    + 1 module ESP32-CAM (camera và xử lý trung tâm).
    + 1 servo motor để điều khiển cơ cấu đóng/mở tủ.
    + 1 cảm biến chuyển động PIR.
    + Nguồn cấp: lấy từ cổng USB máy tính hoặc adapter 5V.
- Công nghệ sử dụng:
    + Công nghệ phần cứng: ESP32-CAM, servo SG90, cảm biến chyển động PIR.
    + Công nghệ phần mềm: lập trình firmware, backend, frontend.
    + Giao thức truyền thông: Wifi, MQTT, HTTP/HTTPs
    + Công nghệ AI: YOLOFace, ArcFace.

CHƯƠNG 2: NỀN TẢNG LÝ THUYẾT

2.1. Công nghệ phần cứng

Phần cứng là nền tảng của hệ thống IoT, bao gồm các module thu thập dữ liệu, xử lý và thực thi. Trong đề tài **"Tủ thông minh nhận diện khuôn mặt"**, hệ thống được xây dựng dựa trên các thành phần chính sau:

2.1.1. ESP32-CAM - Vi điều khiển trung tâm:

- ESP32-CAM là một module camera phát triển dựa trên vi điều khiển ESP32 của Espressif, tích hợp khả năng xử lý hình ảnh và truyền dữ liệu qua Wi-Fi, rất phù hợp cho các ứng dụng IoT(Internet of Things) như giám sát an ninh, nhận diện khuôn mặt, và các dự án camera thông minh. Với kích thước nhỏ gọn và giá thành thấp, ESP32-CAM đã trở thành một lựa chọn phổ biến trong cộng đồng lập trình viên và các nhà phát triển hệ thống nhúng

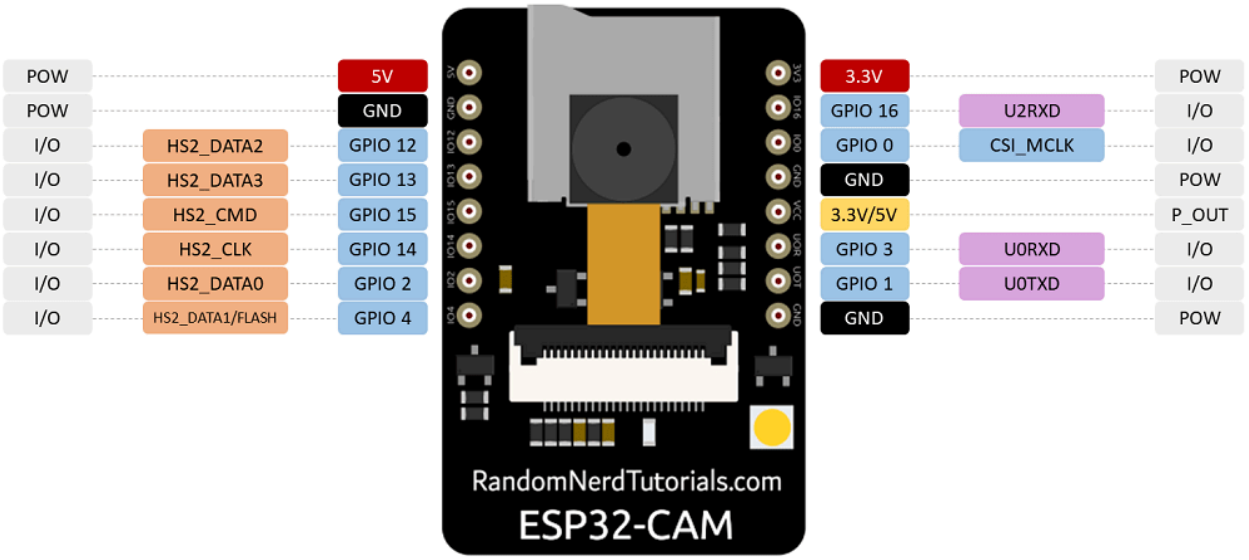
Hình 2.1.1.1: Module ESP32 - CAM - MB

- ESP32-CAM được tích hợp đầy đủ chức năng xử lý hình ảnh, kết nối Wi-Fi, Bluetooth và các cổng giao tiếp mở rộng.
  + Camera OV2640 (2MP, hỗ trợ độ phân giải lên đến 1600x1200)
  + Wi-Fi 802.11 b/g/n và Bluetooth 4.2.
- **Thông số kỹ thuật chính:**

Thuộc tính	Giá trị
Vi xử lý	Dual-core Xtensa LX6, xung nhịp 240 MHz
Bộ nhớ	520 KB SRAM, 4MB Flash
Camera tích hợp	OV2640 – 2MP (hỗ trợ độ phân giải 1600×1200)
Kết nối không dây	Wi-Fi 802.11 b/g/n, Bluetooth 4.2 BLE

Giao tiếp hỗ trợ	UART, SPI, I2C, PWM, GPIO
Nguồn hoạt động	5V DC
Mức tiêu thụ điện	< 200 mA khi hoạt động
Kích thước	Khoảng 40 × 27 mm

- **Sơ đồ chân:**



Hình 2.1.1.2. Sơ đồ chân ESP32-CAM

+ Chân nguồn
  - 5V → cấp điện cho module.
  - GND → nối đất (bắt buộc).
+ Chân giao tiếp UART (lập trình & gửi dữ liệu)
  - U0R (RX) → nhận dữ liệu.
  - U0T (TX) → gửi dữ liệu.
    → Dùng để nạp code qua USB–TTL hoặc truyền dữ liệu đến vi điều khiển khác.
+ Chân điều khiển camera OV2640
  - Camera kết nối nội bộ với ESP32 qua D0–D7, VSYNC, HREF, PCLK (không cần tự nối thủ công).
  - Các chân này phục vụ việc đọc ảnh từ camera.
+ Chân điều khiển đèn Flash
  - GPIO4 → bật/tắt đèn LED Flash.
+ Chân điều khiển (IO)
  - GPIO2, GPIO12, GPIO13, GPIO15, GPIO14…
    → Dùng để gắn cảm biến, relay, nút bấm, truyền lệnh sau xử lý.

+ Chân BOOT
  - GPIO0 → khi nối GND → vào chế độ nạp firmware.

- **Nguyên lý hoạt động**
  + Camera OV2640 chụp ảnh→ Gửi dữ liệu ảnh sang ESP32.

+ ESP32 xử lý dữ liệu → Nén ảnh, nhận diện khuôn mặt hoặc phân tích theo thuật toán.
+ ESP32 truyền dữ liệu qua Wifi→ Gửi ảnh/video hoặc thông tin lên server, app hoặc cloud.
+ ESP32 nhận hoặc gửi lệnh điều khiển → Sau khi xử lý hoặc nhận phản hồi từ broker, ESP32 có thể gửi lệnh điều khiển

- **Ưu điểm nổi bật:**
  + Tích hợp camera, không cần module rời.
  + Giá thành rẻ, khoảng 200.000 VNĐ, dễ mua và thay thế.
  + Lập trình đơn giản qua Arduino IDE, hỗ trợ cập nhật OTA (Over-the-Air).
  + Xử lý ảnh sơ bộ và nén JPEG trực tiếp, giảm tải cho server.
- **Vai trò trong hệ thống:**
  + ESP32-CAM là bộ điều khiển trung tâm, thực hiện:
  + Chụp ảnh khuôn mặt khi phát hiện chuyển động.
  + Gửi ảnh qua Wi-Fi đến server AI để nhận dạng.
  + Nhận phản hồi xác thực và điều khiển servo mở khóa.
  + Cảnh báo người lạ.

2.1.2. Servo Motor SG90 – Bộ truyền động cơ khóa

- Servo SG90 được sử dụng làm bộ chấp hành (actuator) để điều khiển cơ cấu khóa của tủ. Khi hệ thống xác thực khuôn mặt hợp lệ, ESP32-CAM gửi tín hiệu PWM đến servo để mở chốt khóa, sau đó tự động đóng lại sau một khoảng thời gian định sẵn

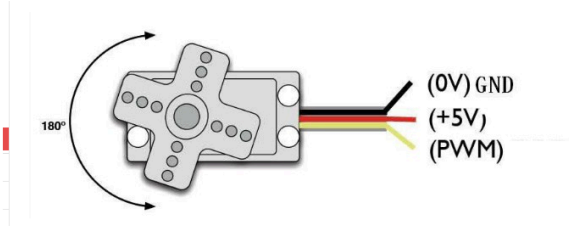
Hình 2.1.2.1: Servo SG90

- **Thông số kỹ thuật:**

Thuộc tính	Giá trị
Điện áp hoạt động	5V DC

Dòng tiêu thụ	Khoảng 500 mA
Góc quay	0° – 180°
Điều khiển	Tín hiệu PWM từ ESP32-CAM
Trọng lượng	9g
Tốc độ phản hồi	0.1 s/60°

- **Sơ đồ chân:**



Hình 2.1.2.2: sơ đồ chân servo SG90

Các chân:

+ VCC → Nguồn 5V
+ GND → mass
+ Signal → nhận tín hiệu PWM từ vi điều khiển

- **Ưu điểm:**
  + Kích thước nhỏ, dễ lắp đặt trong khoang tủ.
  + Hoạt động ổn định và chính xác, phù hợp với nguồn USB 5V.
  + Phản hồi nhanh, hỗ trợ điều khiển mượt qua PWM.

2.1.3. Cảm biến PIR (HC-SR501)

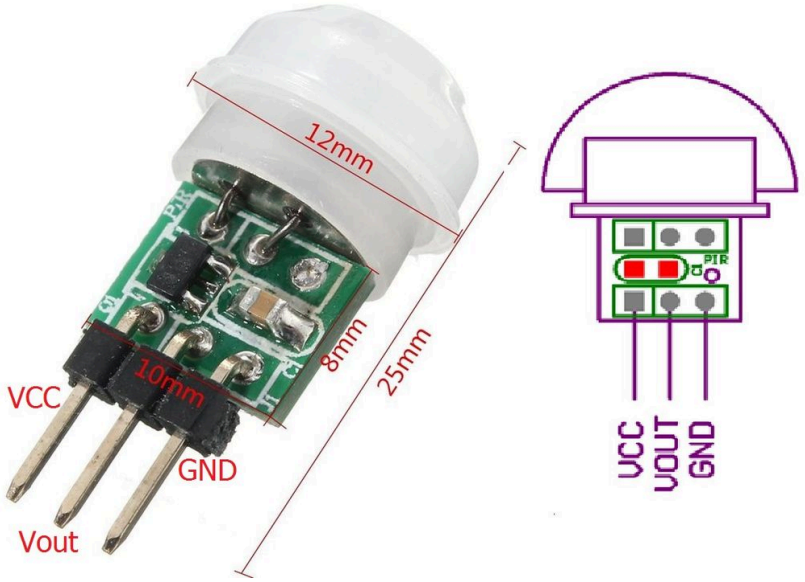
- Cảm biến PIR giúp phát hiện chuyển động của con người để kích hoạt camera chụp ảnh hoặc đánh thức ESP32-CAM khỏi chế độ deep sleep, giúp tiết kiệm năng lượng.

Hình 2.1.3.1: Cảm biến PIR (HC-SR501)

- **Thông số kỹ thuật:**

Thuộc tính	Giá trị
Điện áp hoạt động	5V DC
Dòng tiêu thụ	<50 µA (standby), ~100 µA (hoạt động)
Khoảng cách phát hiện	3 – 7 mét
Góc phát hiện	~120°
Thời gian trễ	0.5 – 200 giây (có thể điều chỉnh)

- **Sơ đồ chân:**



Hình 2.1.3.2: Sơ đồ chân cảm biến chuyển động PIR

+ Chân nguồn
    - VCC → cấp điện (3.3V – 5V)
    - GND → nối đất
+ Chân tín hiệu
    - OUT → ra tín hiệu digital (HIGH khi phát hiện chuyển động, LOW khi không)
+ Chân điều chỉnh (nếu có trên module)
    - ADJ / Sens / Delay → núm vặn để điều chỉnh:
        - Sensitivity: khoảng cách phát hiện
        - Time / Delay: thời gian tín hiệu HIGH giữ nguyên sau khi phát hiện

- **Nguyên lý hoạt động:**

PIR (Passive Infrared Sensor) phát hiện sự thay đổi bức xạ hồng ngoại (nhiệt) từ cơ thể người hoặc động vật.

+ Khi môi trường ổn định → đầu ra LOW.

+ Khi có người di chuyển → IR thay đổi → đầu ra HIGH
+ Thấu kính Fresnel mở rộng góc quét và tăng độ nhạy.

- **Ưu điểm:**
  - Tiêu thụ năng lượng cực thấp, phù hợp IoT.
  - Giúp giảm tải CPU khi không có người đến gần.
  - Nâng cao tính thông minh của hệ thống: camera chỉ kích hoạt khi có người.

2.1.4. Tổng quan hệ thống phần cứng

- Tất cả các module được cấp nguồn 5V DC (qua adapter USB hoặc pin sạc).
- Hệ thống được thiết kế hoạt động độc lập, không cần máy tính trung gian.
- Nguồn cấp:
  - **Nguồn chính:** Adapter 5V – 1A, đảm bảo đủ dòng cho ESP32-CAM và servo.
  - **Nguồn dự phòng:** Pin LiPo 3.7V (kèm mạch boost 5V) hoặc UPS mini 1000 mAh, hoạt động 30–60 phút khi mất điện.
- Kết nối giữa các thành phần:
  - PIR sensor → GPIO Input ESP32-CAM (phát hiện chuyển động).
  - Servo motor → GPIO PWM ESP32-CAM (điều khiển chốt).
  - ESP32-CAM ↔ Wi-Fi ↔ Server Node.js (gửi ảnh, nhận lệnh mở khóa).
  - Server ↔ Web (React/Firebase) để hiển thị và điều khiển realtime

2.2. Các giao thức truyền thông

Giao thức truyền thông là cầu nối giữa các lớp trong kiến trúc IoT, đảm bảo dữ liệu được truyền tải an toàn, ổn định và theo thời gian thực. Trong hệ thống Tủ thông minh nhận diện khuôn mặt, nhóm sử dụng Wi-Fi làm nền tảng kết nối chính, kết hợp MQTT cho truyền thông hai chiều realtime và HTTP/HTTPS cho các dịch vụ web và API cloud.

2.2.1. Wi-Fi (IEEE 802.11)

- Wi-Fi là giao thức truyền thông không dây phổ biến, hoạt động ở dải tần 2.4GHz, cho tốc độ truyền dữ liệu lên tới 150Mbps và phạm vi hoạt động từ 10–50 mét trong môi trường trong nhà.
- Module ESP32-CAM được tích hợp Wi-Fi và có thể hoạt động ở hai chế độ:
  - Station (Client Mode): Kết nối vào mạng Wi-Fi sẵn có để gửi dữ liệu lên server hoặc cloud.
  - Access Point (AP Mode): Tạo mạng nội bộ để người dùng điều khiển và cấu hình hệ thống trực tiếp.
- Ưu điểm:
  - Triển khai dễ dàng, không cần gateway trung gian.
  - Băng thông lớn, đủ để truyền ảnh hoặc video.
  - Hỗ trợ giao tiếp song song với nhiều thiết bị khác trong cùng mạng.
- Hạn chế:
  - Tiêu thụ điện năng cao hơn so với BLE hoặc ZigBee.
  - Phụ thuộc vào độ ổn định mạng Wi-Fi khi truyền dữ liệu liên tục.

2.2.2. MQTT (Message Queuing Telemetry Transport)

- MQTT là giao thức truyền thông nhẹ (lightweight protocol) được thiết kế tối ưu cho các hệ thống IoT có tài nguyên hạn chế.
- Giao thức này hoạt động theo mô hình Publish / Subscribe, giúp các thiết bị gửi và nhận dữ liệu qua broker trung gian mà không cần kết nối trực tiếp.
- Đặc điểm nổi bật:
    - Gói tin nhỏ (header chỉ 2 bytes) → tiết kiệm băng thông.
    - Hỗ trợ QoS (Quality of Service) – đảm bảo tin nhắn được truyền đến nơi nhận:
        - QoS 0: gửi một lần, không xác nhận.
        - QoS 1: gửi ít nhất một lần (dùng cho lệnh mở khóa).
        - QoS 2: gửi đúng một lần (cho dữ liệu quan trọng).
    - Hỗ trợ truyền dữ liệu hai chiều (bi-directional) giữa thiết bị và server theo thời gian thực.
- Vai trò trong hệ thống:
    - Truyền thông báo realtime từ ESP32-CAM đến server khi phát hiện khuôn mặt hoặc xâm nhập.
    - Nhận lệnh điều khiển mở/đóng khóa từ ứng dụng web hoặc mobile.
    - Giúp hệ thống hoạt động ổn định, phản hồi nhanh và tiết kiệm năng lượng hơn so với HTTP polling.

2.2.3. HTTP / HTTPS (HyperText Transfer Protocol Secure)

- HTTP/HTTPS được sử dụng cho các chức năng trao đổi dữ liệu với web server và API cloud.
- Cụ thể, ESP32-CAM gửi hình ảnh khuôn mặt đến server AI hoặc Firebase bằng phương thức HTTP POST, sau đó nhận phản hồi xác thực từ server để điều khiển servo mở khóa.
- Tính năng chính:
    - Mã hóa bảo mật: HTTPS kết hợp TLS/SSL để bảo vệ dữ liệu khuôn mặt và thông tin người dùng.
    - Tương thích cao: Dễ dàng tích hợp với các nền tảng web, cloud và database.
    - Phù hợp với dashboard web: hỗ trợ cảnh báo qua trình duyệt.

2.3. Công nghệ phần mềm

Phần mềm đóng vai trò kết nối, xử lý và hiển thị thông tin, giúp hệ thống Tủ thông minh nhận diện khuôn mặt hoạt động trơn tru từ tầng thiết bị đến người dùng cuối. Các thành phần phần mềm được chia thành 3 lớp chính: lập trình nhúng (firmware), xử lý backend và giao diện frontend.

2.3.1. Lập trình nhúng (Firmware cho ESP32-CAM)

- Công cụ phát triển:
    + Arduino IDE, hỗ trợ ngôn ngữ C/C++, thuận tiện cho lập trình vi điều khiển.
    + Hỗ trợ cập nhật firmware từ xa (OTA – Over-the-Air) giúp dễ bảo trì và khắc phục lỗi.
- Chức năng chính:
    + Quản lý kết nối Wi-Fi và MQTT client để truyền/nhận dữ liệu.
    + Chụp ảnh khuôn mặt bằng camera OV2640, nén ảnh JPEG trước khi gửi.
    + Gửi ảnh đến server qua HTTP POST hoặc MQTT topic.
    + Nhận phản hồi từ server để điều khiển servo SG90 mở/đóng khóa.
    + Gửi log truy cập và cảnh báo bất thường lên server

- Thư viện sử dụng:
    + ESP32Cam.h, WiFi.h, PubSubClient.h, ESPAsyncWebServer.h.
    + Hỗ trợ streaming video, gửi ảnh dạng Base64 hoặc JPEG.
    + Hỗ trợ deep sleep mode để tiết kiệm năng lượng.

2.3.2. Backend – Xử lý dữ liệu và API

- Ngôn ngữ & Framework:
    - Node.js: Môi trường chạy backend, xử lý API, xác thực người dùng và điều khiển thiết bị.
    - MySQL: Cơ sở dữ liệu lưu trữ thông tin người dùng, thiết bị, lịch sử truy cập, embeddings khuôn mặt.
- Chức năng chính:
    - Nhận và xử lý hình ảnh khuôn mặt từ ESP32-CAM qua HTTP POST hoặc MQTT topic.
    - Xác thực và lưu trữ dữ liệu (thông tin người dùng, lịch sử mở khóa, cảnh báo) vào MySQL.
    - Gửi lệnh điều khiển (mở/khóa tủ) và cảnh báo realtime đến frontend qua Socket.IO.
    - Cung cấp API quản lý người dùng: thêm, sửa, xóa, hoặc cập nhật khuôn mặt.

2.3.3. Frontend – Giao diện Web và Ứng dụng người dùng

- Công nghệ sử dụng:
    + **React.js (TypeScript):** Xây dựng giao diện dashboard quản lý thiết bị, người dùng, cảnh báo.
    + **Next.js:** Framework tích hợp frontend và backend (API Routes), hỗ trợ render nhanh và triển khai serverless.
    + **Tailwind CSS:** Thiết kế giao diện hiện đại, tối ưu cho mobile và desktop.
- Chức năng chính:
    - Hiển thị Dashboard realtime:
        - Trạng thái tủ (mở/đóng).
        - Cảnh báo an ninh.
    - Điều khiển từ xa:
        - Gửi lệnh mở tủ qua API hoặc MQTT.
    - Quản lý người dùng:
        - Admin thêm/sửa thông tin, upload ảnh khuôn mặt mới.
        - Theo dõi danh sách thiết bị, hoạt động theo thời gian thực.

2.4. Công nghệ AI

2.4.1. YOLOFacev8

- YOLOFace8 là mô hình YOLOv8 được tối ưu hóa dành riêng cho nhiệm vụ phát hiện khuôn mặt trong ảnh hoặc video.
- Vai trò:
    + Vẽ bounding box quanh khuôn mặt
    + Xuất tọa độ (x, y, width, height) cho mỗi khuôn mặt
- Đặc điểm:
    + Phát hiện vị trí khuôn mặt trong ảnh/video.
    + Phát hiện khuôn mặt nhỏ (small face).
    + Dùng làm bước đầu cho face recognition, face tracking.

- Ứng dụng: Phát hiện (detect) vị trí khuôn mặt trong khung hình
    + Xác định có khuôn mặt hay không
    + Xác định
    + Cắt ảnh khuôn mặt sạch sẽ để đưa vào ArcFace.

2.4.2. ArcFace

- ArcFace tạo ra không gian đặc trưng mà trong đó khuôn mặt của cùng một người thì gần nhau, còn khuôn mặt của hai người khác nhau thì cách xa nhau bằng một khoảng góc rõ ràng.
- ArcFace nhận diện khuôn mặt bằng cách ánh xạ ảnh mặt người thành một vector 512 chiều (embedding).
- ArcFace làm cho các vector này tuân theo quy tắc:
    + Cùng người → góc nhỏ (véc-tơ gần nhau).
    + Khác người → góc lớn.
- Ứng dụng:
    + Trích xuất vector đặc trưng (embedding). Vector này chứa hình dạng mặt, cấu trúc mắt - mũi - miệng, khoảng cách tỷ lệ, đặc điểm cá nhân.
    + So sánh với dữ liệu người dùng đã đăng ký: so sánh với embedding đã lưu bằng khoảng cách cosine. Nếu khoảng cách nhỏ hơn ngưỡng trong hệ thống → Mở khóa thành công.

CHƯƠNG 3: PHÂN TÍCH YÊU CẦU CHỨC NĂNG

3.1. Chức năng Đăng nhập

- Chức năng đăng nhập là bước đầu tiên để truy cập vào hệ thống quản lý tủ thông minh. Giao diện đăng nhập được thiết kế đơn giản, thân thiện và bảo mật tốt.

3.1.1. Quy trình đăng nhập

- Giao diện đăng nhập có 2 ô nhập: email, password và nút Đăng nhập để truy cập vào trang web
- Dữ liệu xác thực được lưu trong bảng "users" của cơ sở dữ liệu MySQL. Gồm 1 số trường như : id (PK), name, email, password,...
- Nếu đăng nhập thành công, web sẽ chuyển hướng tới giao diện chính của người dùng.
- Đăng nhập thất bại, hệ thống sẽ hiển thị thông báo "Tài khoản không đúng" và quay lại giao diện đăng nhập.

3.1.2. Bảo mật đăng nhập

Các biện pháp bảo mật được sử dụng:

- Mã hóa mật khẩu (Password Hashing):
  - Sử dụng thư viện bcryptjs với salt rounds = 10
  - Mật khẩu không bao giờ được lưu dưới dạng plaintext
  - Mỗi mật khẩu được hash thành chuỗi 60 ký tự, không thể revers
- Xác thực JWT (JSON Web Token):
  - Mỗi request sau khi đăng nhập phải chứa JWT token trong header Authorization: Bearer <token>
  - Token được mã hóa bằng secret key riêng biệt
  - Hết hạn sau 24 giờ, yêu cầu người dùng đăng nhập lại
  - Middleware auth-middleware.js kiểm tra token trên mỗi request
- Session Management:
  - Lưu token trong localStorage với thời hạn 24 giờ
  - Người dùng có thể "Đăng Xuất" để xóa token
  - Khi token hết hạn, hệ thống tự động chuyển hướng về trang đăng nhập
- HTTPS/SSL:
  - Sử dụng HTTPS để mã hóa toàn bộ dữ liệu truyền từ client tới server
  - Ngăn chặn man-in-the-middle attack

3.2. Chức năng Quản lý tủ

- Sau khi đăng nhập thành công, người dùng sẽ vào giao diện web nơi có thể lựa chọn các thao tác với tủ thông minh mà người dùng quản lý
- Giao diện quản lý thông tin các tủ hiển thị dưới dạng card với các thông tin như: tên tủ, mã tủ, vị trí, người sở hữu, trạng thái tủ hiện tại (đóng/ mở),.... Và nút để mở tủ từ xa.
- Phân quyền:
  - Admin có quyền xem được tất cả các tủ trong hệ thống
  - User chỉ xem được tủ mình sở hữu.

3.3. Chức năng Giám sát trạng thái real-time

- Trên Dashboard hiển thị các thông tin:
    - Thông tin, trạng thái tủ
    - Thống kê: số tủ theo trạng thái, các cảnh báo, số người dùng,...
    - Activity Log: hiển thị lịch sử truy cập tủ
- Frontend tự động fetch dữ liệu mới từ backend
- Người dùng có thể click nút "Refresh" để cập nhật ngay lập tức

3.4. Chức năng Điều khiển

3.4.1. Chế độ thủ công (Bấm nút trên Web)

- Người dùng và admin có thể truy cập website để gửi lệnh mở/đóng cửa tủ từ xa qua Internet. Lệnh được truyền qua Network Layer (HTTP/MQTT) đến hệ thống, kích hoạt servo motor. Hỗ trợ kiểm tra trạng thái tủ realtime (mở/đóng).
- Trong trường hợp của admin, chỉ thực hiện chức năng khi có yêu cầu từ người dùng.
- Bảo mật trong chế độ thủ công:
    - Chỉ người dùng có quyền mới có thể mở
    - Mỗi lệnh phải được xác thực bằng JWT token
    - Tất cả hành động được ghi log chi tiết
    - Tối đa 10 lệnh/phút (ngăn spam)

3.4.2.Chế độ tự động (Mở bằng nhận diện khuôn mặt)

- Hệ thống sử dụng camera trên ESP32-CAM để chụp hình khuôn mặt người dùng khi tiếp cận tủ. Hình ảnh được gửi đến AI service để so sánh với cơ sở dữ liệu khuôn mặt đã đăng ký.
- Nếu khớp (độ chính xác >65%), servo motor sẽ tự động mở chốt tủ trong vòng <5 giây.
- Nếu không khớp sẽ giữ cảnh báo lên web cho người dùng sở hữu.

3.5. Chức năng Lưu trữ và Báo cáo

3.5.1. Xem lịch sử truy cập (Access Logs)

- Giao diện Access Logs cho phép admin quản lý xem lại các thông tin truy cập tủ, dữ liệu này được lưu trong bảng “access_logs”: mã tủ, người dùng (null nếu không xác định), loại truy cập (tự động / từ xa), trạng thái (thành công hay thất bại), thời gian truy cập,...
- Tính năng:
    - Tự động cập nhật mỗi 10 giây
    - Có thể scroll để xem logs cũ hơn

3.5.2. Xem cảnh báo an ninh (Alerts)

- Các loại cảnh báo:
    - Truy cập từ người không được phép
    - Nhận diện khuôn mặt không khớp
    - Tủ mất kết nối,...

- Nếu có cảnh báo chưa đọc, tab alerts sẽ hiển thị số lượng. Cảnh báo chưa đọc: nền màu đỏ nhạt (còn nổi bật). Cảnh báo đã đọc: nền màu xám (nhạt hơn)
- Mỗi alert sẽ hiển thị với các thông tin như: mã tủ, loại cảnh báo, mô tả chi tiết, ảnh chụp lúc cảnh báo,...

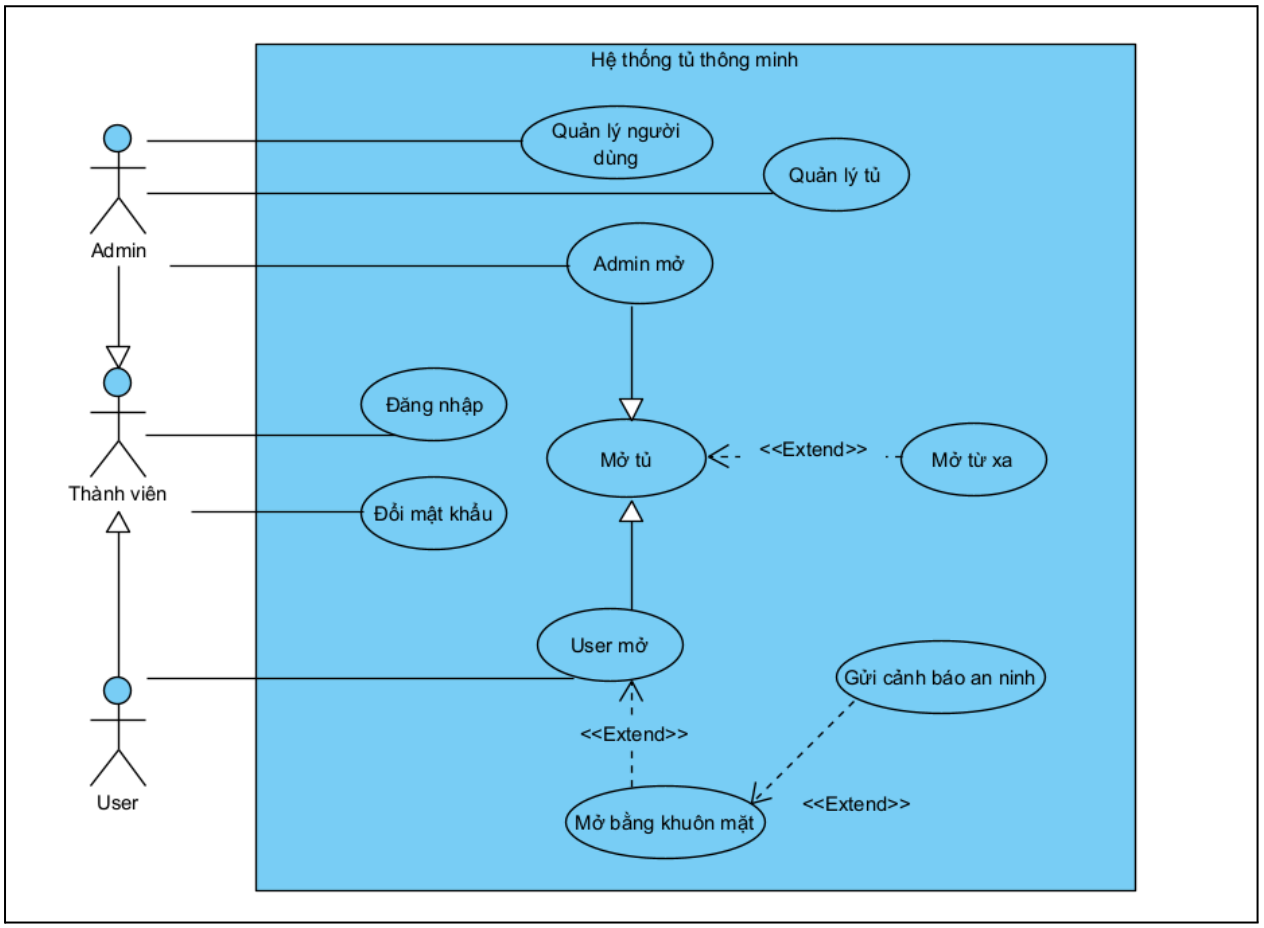
3.5.3. Quản lý người dùng (Admin only)

- Tại tab chức năng quản lý Users, admin có thể thực hiện các chức năng như:
    - Thêm người dùng mới: điền form thông tin (name, email, password)
    - Xem danh sách người dùng với đầy đủ các thông tin: tên, email, vai trò và nút xoá người dùng.
    - Admin có thể xoá người dùng khỏi hệ thống quản lý.

3.6. Sơ đồ use case tổng quan các chức năng của hệ thống

- **Mô tả chi tiết các use case:**
    - Use case đăng nhập và đổi mật khẩu: thành viên hệ thống được quyền đăng nhập và đổi mật khẩu của mình nếu muốn
    - Use case quản lý người dùng: admin có quyền thêm, sửa, xóa thông tin của người dùng
    - Use case quản lý tủ: admin có quyền thêm tủ vào hệ thống, sửa thông tin tủ và xóa tủ ra khỏi hệ thống
    - Usecase mở tủ: thành viên hệ thống có quyền mở tủ từ xa. trong đó:
        - User phải được xác thực thông tin (đăng nhập vào hệ thống hoặc khuôn mặt được nhận diện đúng) thì mới ở được tủ.
        - Admin chỉ được mở tủ từ xa thông qua quyền hệ thống khi có yêu cầu từ người dùng (lỗi, …).
    - Usecase mở bằng khuôn mặt được extend từ usecase user mở tủ: hệ thống tự động nhận diện khuôn mặt của người dùng thông qua model AI và thực hiện cơ chế xác thực để mở tủ
    - Usecase gửi cảnh báo an ninh được extend từ usecase mở bằng khuôn mặt: trong trường hợp xác thực khuôn mặt bị sai (không đúng user sở hữu tủ), hệ thống sẽ gửi cảnh báo an ninh về tài khoản của người dùng

- **Sơ đồ use case tổng quan:**



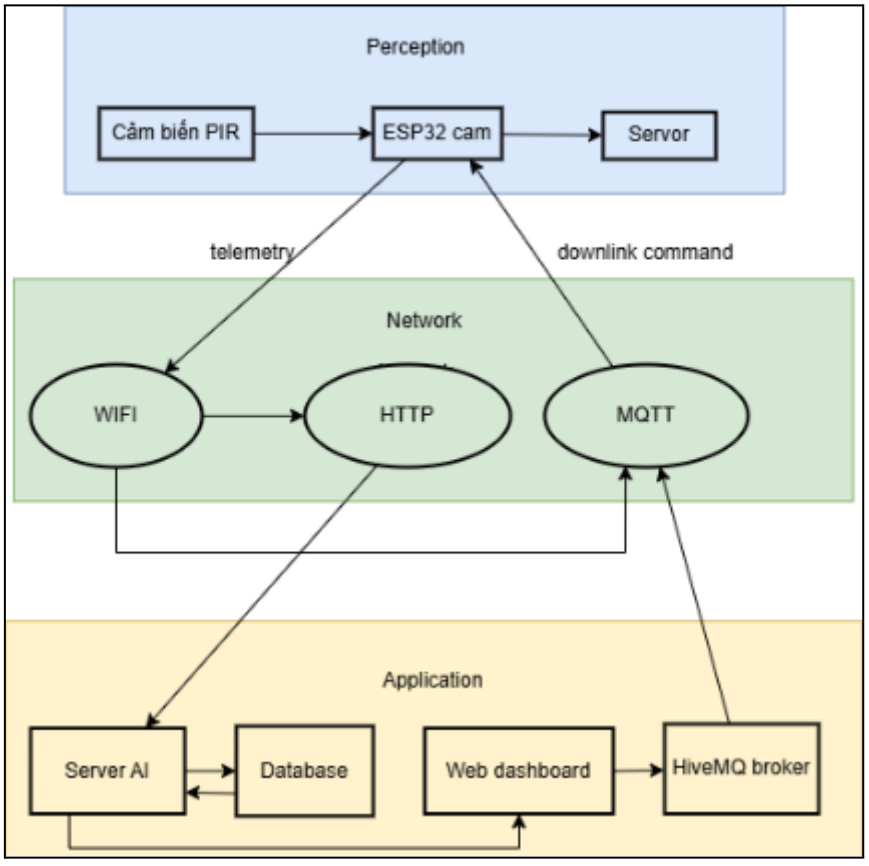
Hình 3.6.1: Sơ đồ use case tổng quan

CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

4.1. Thiết kế Logic

4.1.1. Kiến trúc Logic hệ thống

- Luồng hoạt động:
    - **Phát hiện & Chụp ảnh**: Cảm biến PIR phát hiện chuyển động kích hoạt ESP32 cam hoạt động.
    - **Gửi lên Cloud**: Ảnh chụp từ ESP32 cam được gửi qua WIFI (HTTP) lên Server AI.
    - **Xử lý**: Server AI nhận diện khuôn mặt. Nếu nhận diện hợp lệ, Server AI gửi message tới web
    - **Ra lệnh**: Web sẽ gửi lệnh mở cửa tới HiveMQ Broker
    - **Chấp hành**: Broker gửi lệnh mở cửa qua MQTT về ESP32 cam, kích hoạt servo mở tủ.
- Thiết kế biểu đồ 3 lớp chi tiết:



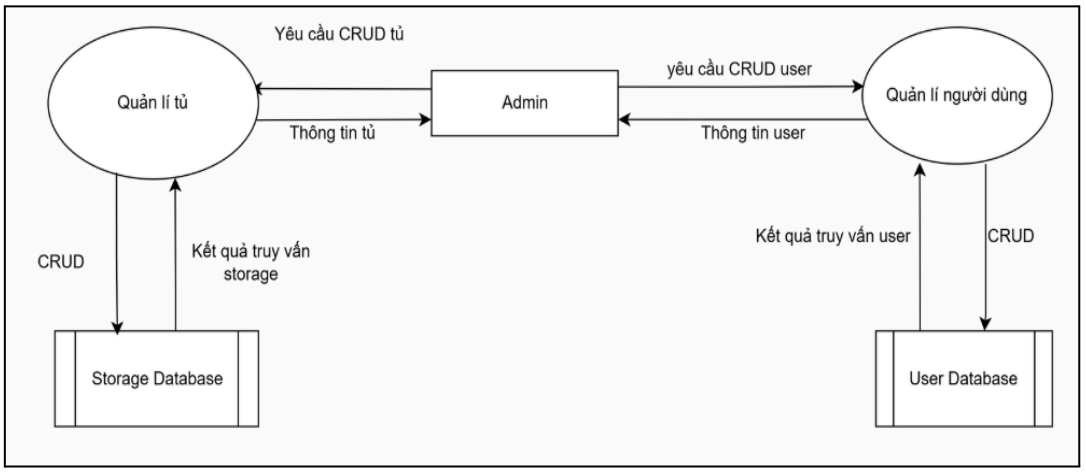
Hình 4.1.1.1: Thiết kế 3 lớp chi tiết

4.1.2. Thiết kế luồng dữ liệu

- Chức năng quản lý tủ, người dùng
    - Các thành phần trong sơ đồ

Thành phần	Vai trò
Storage Database	Chứa thông tin người dùng

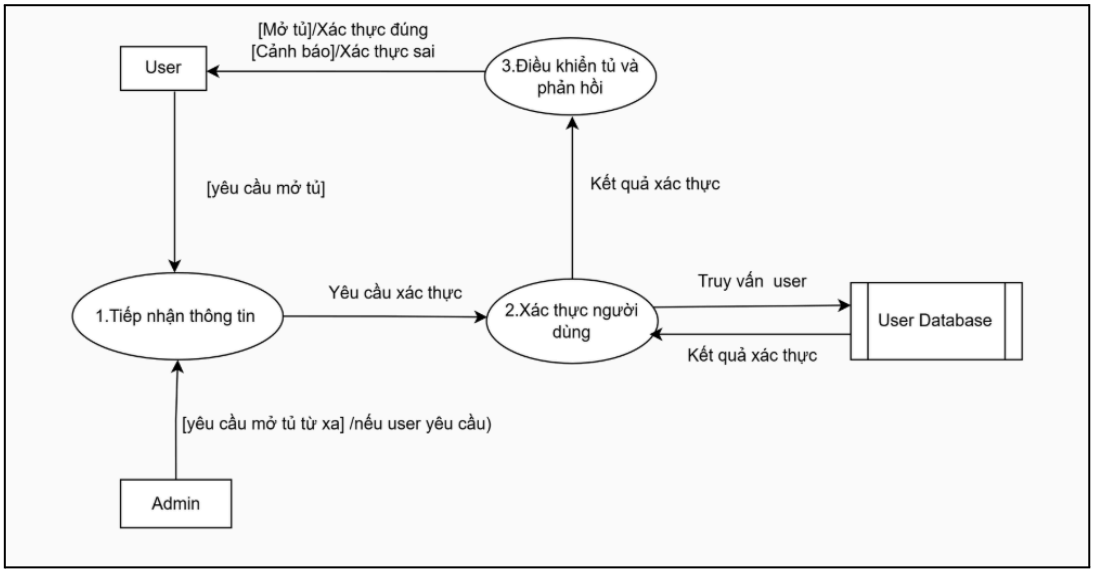
User Database	Chứa thông tin tủ
Admin	Thêm, sửa, xóa tủ Thêm, sửa, xóa người dùng



Hình 4.1.2.1: Sơ đồ luồng dữ liệu chức năng quản lý tủ, người dùng

- Chức năng mở tủ
  - Các thành phần trong sơ đồ

Thành phần	Vai trò
Admin	Điều khiển đóng mở tủ từ xa nếu user yêu cầu
User	Gửi yêu cầu mở tủ
User Database	Chứa thông tin người dùng



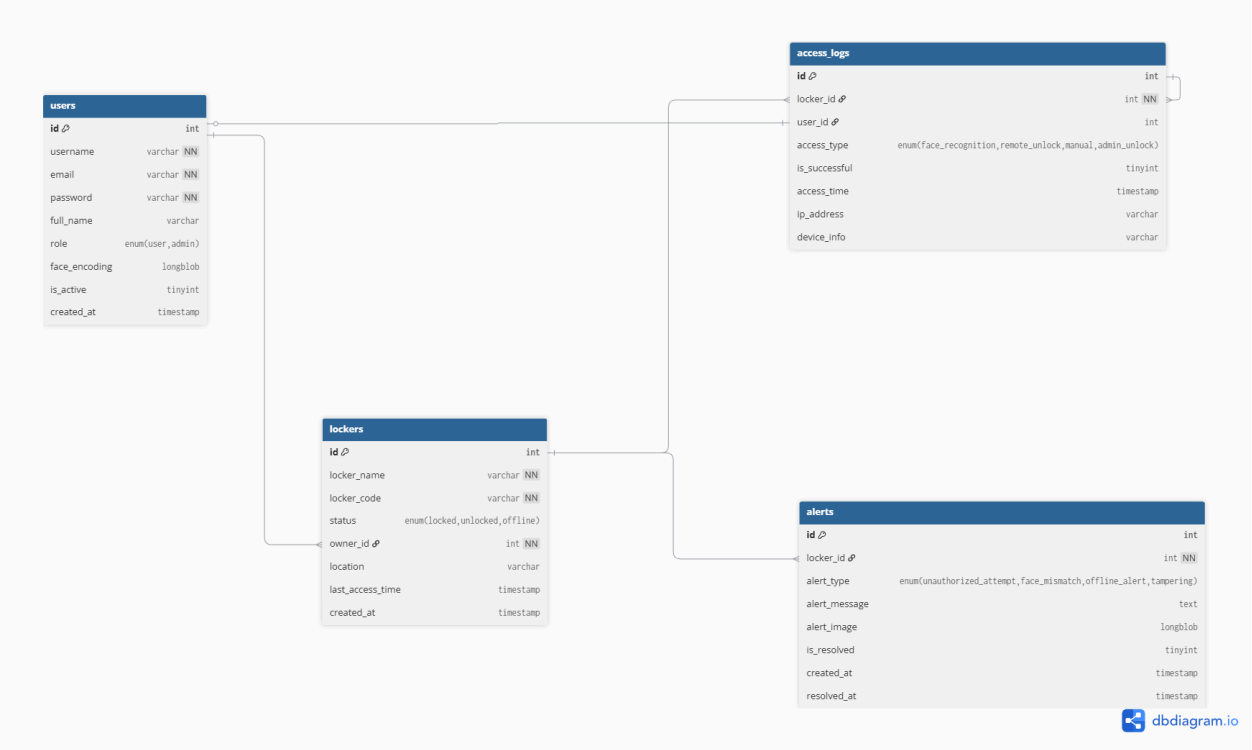
Hình 4.1.2.2: sơ đồ luồng dữ liệu chức năng mở tủ

4.1.3. Thiết kế database

- Cơ sở dữ liệu (CSDL) của Hệ thống Tủ Thông Minh được thiết kế theo mô hình quan hệ (Relational Model), sử dụng MySQL, nhằm đảm bảo tính toàn vẹn, bảo

mật và khả năng truy xuất dữ liệu nhanh chóng cho các nghiệp vụ thời gian thực của IoT và AI.

- **Sơ đồ quan hệ thực thể (ERD):**



Hình 4.1.3.1 Sơ đồ quan hệ thực thể

❖ **Giải thích các thực thể:**

- **Bảng users (quản lý người dùng):** Bảng này lưu trữ thông tin cơ bản và dữ liệu sinh trắc học (khuôn mặt) của tất cả người dùng trong hệ thống (bao gồm cả Admin và Thành viên).

Trường dữ liệu	Kiểu dữ liệu	Mô tả
id	INT (PK)	Khóa chính, ID duy nhất của người dùng.
username	VARCHAR	Tên đăng nhập duy nhất.
email	VARCHAR	Email duy nhất, dùng để đăng nhập.
password	VARCHAR	Mật khẩu đã được Hash (sử dụng bcrypt).
role	ENUM	Vai trò người dùng (user hoặc admin).

face_encoding	LONGBLOB	Vector nhúng khuôn mặt (dữ liệu sinh trắc học) của người dùng, được tạo bởi thuật toán AI (ArcFace). Dùng để so sánh trong quá trình nhận diện.
is_active	TINYINT	Trạng thái hoạt động của tài khoản (1: Active, 0: Inactive).

- **Bảng lockers (quản lý Tủ thông minh):** Bảng này lưu trữ thông tin chi tiết về các thiết bị tủ thông minh được quản lý trong hệ thống.

Trường dữ liệu	Kiểu dữ liệu	Mô tả
id	INT (PK)	Khóa chính, ID duy nhất của tủ.
locker_name	VARCHAR	Tên hiển thị của tủ (ví dụ: Tủ A, Locker 01).
locker_code	VARCHAR	Mã định danh duy nhất (Unique Code) của thiết bị vật lý, dùng để giao tiếp qua MQTT.
status	ENUM	Trạng thái hiện tại của tủ (locked, unlocked, offline).
owner_id	INT (FK)	ID của người sở hữu tủ, liên kết đến users.id.
location	VARCHAR	Vị trí lắp đặt của tủ.
last_access_time	TIMESTAMP	Thời gian lần cuối tủ được truy cập thành công.

- **Bảng access_logs (Nhật ký Truy cập):** Bảng quan trọng lưu trữ tất cả các sự kiện truy cập (đóng/mở) vào tủ, phục vụ cho chức năng Giám sát và Báo cáo.

Trường dữ liệu	Kiểu dữ liệu	Mô tả
id	INT (PK)	Khóa chính, ID của bản ghi

		nhật ký.
locker_id	INT (FK)	Tủ xảy ra sự kiện, liên kết đến lockers.id.
user_id	INT (FK)	Người dùng thực hiện hành động (có thể NULL nếu là người lạ).
access_type	ENUM	Phương thức truy cập: face_recognition, remote_unlock (từ Web), admin_unlock.
is_successful	TINYINT	Kết quả truy cập (1: Thành công, 0: Thất bại).
access_time	TIMESTAMP	Thời gian chính xác sự kiện xảy ra.

- **Bảng alerts (Cảnh báo An ninh):** Bảng này lưu trữ các sự kiện bất thường và cảnh báo an ninh, giúp Admin và User nắm bắt kịp thời các rủi ro.

Trường dữ liệu	Kiểu dữ liệu	Mô tả
id	INT (PK)	Khóa chính, ID của cảnh báo.
locker_id	INT (FK)	Tủ xảy ra cảnh báo, liên kết đến lockers.id.
alert_type	ENUM	Loại cảnh báo: unauthorized_attempt (truy cập trái phép), face_mismatch (khuôn mặt không khớp), offline_alert (tủ mất kết nối), tampering (can thiệp vật lý).
alert_message	TEXT	Nội dung chi tiết của cảnh báo.

alert_image	LONGBLOB	Ảnh chụp bằng chứng tại thời điểm cảnh báo (ví dụ: ảnh người lạ).
is_resolved	TINYINT	Trạng thái xử lý cảnh báo (0: Mới, 1: Đã giải quyết).

❖ **Mối quan hệ giữa các thực thể:**

Mối Quan Hệ	Các Thực Thể Liên Quan	Loại Mối Quan Hệ	Giải thích
Sở hữu Tủ	users và lockers	Một-Nhiều (1:N)	Một người dùng (users.id) có thể sở hữu hoặc tạo ra nhiều tủ (lockers.owner_id).
Ghi nhận Truy cập	lockers và access_logs	Một-Nhiều (1:N)	Một tủ (lockers.id) có thể có nhiều bản ghi nhật ký truy cập (access_logs.locker_id).
Thực hiện Truy cập	users và access_logs	Một-Nhiều (1:N)	Một người dùng (users.id) có thể thực hiện nhiều lần truy cập (access_logs.user_id). (Lưu ý: access_logs.user_id là khóa ngoại chấp nhận giá trị NULL để ghi lại sự kiện người lạ/không xác định).
Cảnh báo Tủ	lockers và alerts	Một-Nhiều (1:N)	Một tủ (lockers.id) có thể phát sinh nhiều cảnh báo an ninh (alerts.locker_id).

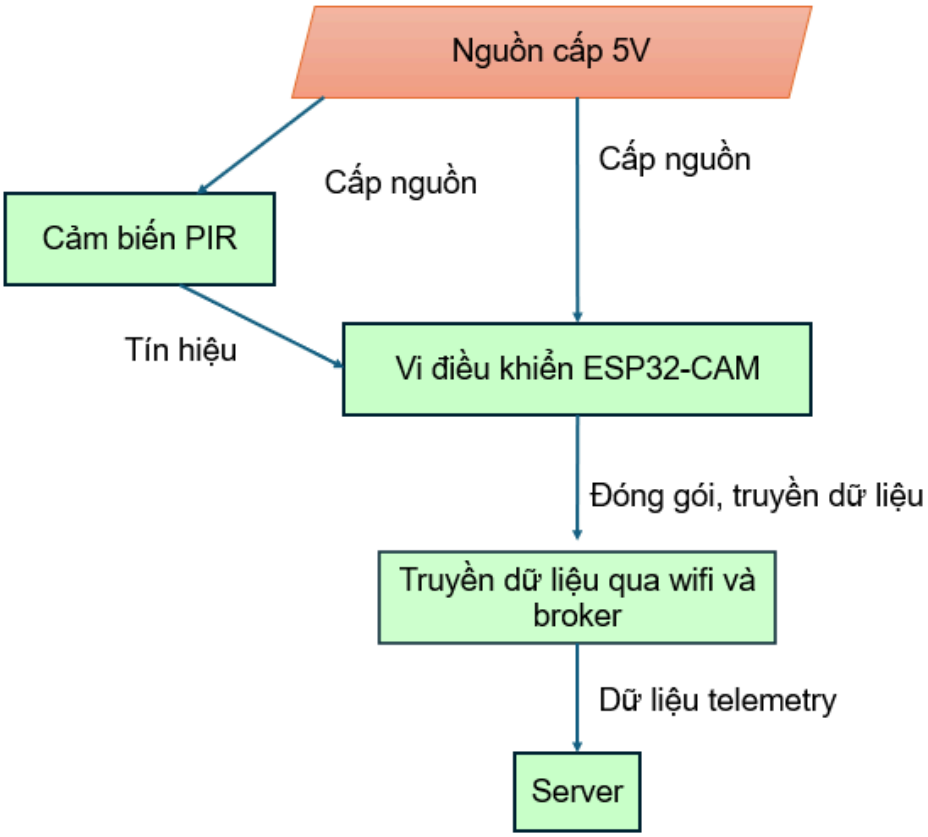
4.1.4. Thiết kế logic an ninh và quản lý

- Mục tiêu: Hệ thống Tủ thông minh nhận diện khuôn mặt không chỉ thực hiện chức năng mở/đóng tủ tự động mà còn phải đảm bảo các yêu cầu về an toàn thông tin, bảo mật dữ liệu sinh trắc học và quyền riêng tư của người dùng. Vì hệ thống liên quan trực tiếp đến dữ liệu khuôn mặt – một loại thông tin cá nhân nhạy cảm – nên việc thiết kế cơ chế an ninh được xem là yếu tố cốt lõi. Mục tiêu của thiết kế an ninh gồm:
    - Ngăn chặn truy cập trái phép vào hệ thống tủ.
    - Phát hiện và cảnh báo kịp thời các hành vi xâm nhập bất hợp pháp.
- Mô hình bảo mật tổng thể:
    - Xác thực dựa trên khuôn mặt: Chỉ người có trong cơ sở dữ liệu khuôn mặt hợp lệ mới được phép mở tủ. Đây là lớp bảo mật sinh trắc học, giúp loại bỏ các rủi ro liên quan đến chìa khóa cơ, thẻ RFID hoặc mật khẩu.
    - Ngưỡng nhận diện an toàn: Hệ thống chỉ cấp quyền khi độ tương đồng khuôn mặt $\geq 65\%$ để giảm xác suất nhận diện sai hoặc giả mạo bằng ảnh.
    - Giới hạn phần cứng: ESP32-CAM chỉ thực hiện nhiệm vụ chụp ảnh và tiền xử lý đơn giản (resize, nén JPEG). Việc nhận diện được xử lý ở server nhằm tránh bị can thiệp sâu vào firmware để đánh cắp dữ liệu.
    - Phân quyền người dùng: Hệ thống chia thành 2 vai trò chính
        - Admin:
            - Thêm / sửa / xóa người dùng.
            - Quản lý danh sách thiết bị (tủ).
            - Xem lịch sử truy cập và cảnh báo.
            - Quản lý dữ liệu khuôn mặt.
        - User:
            - Chỉ được phép mở tủ của mình.
            - Nhận thông báo khi có truy cập trái phép.
            - Không có quyền truy cập dữ liệu của người khác.
- Thiết kế hệ thống quản lý:
    - Quản lý người dùng:
        - Thêm user mới kèm dữ liệu khuôn mặt.
        - Xóa người không còn quyền truy cập.
    - Quản lý thiết bị:
        - Thêm tủ mới (thêm ESP32-CAM).
        - Kiểm tra trạng thái kết nối (online/offline).

4.2. Thiết kế vật lý

4.2.1. Thiết kế phần tử cảm nhận & điều khiển

- Chọn thiết bị cảm biến PIR đo biến động của chuyển động
- Tích hợp vi điều khiển ESP32-CAM: vi điều khiển kết hợp camera để gửi ảnh của người dùng lên server AI cho chức năng nhận diện khuôn mặt
- Yêu cầu nguồn cấp 5V



Hình 4.2.1.1: Thiết kế vật lý

4.2.2. Thiết kế mạng truyền thông

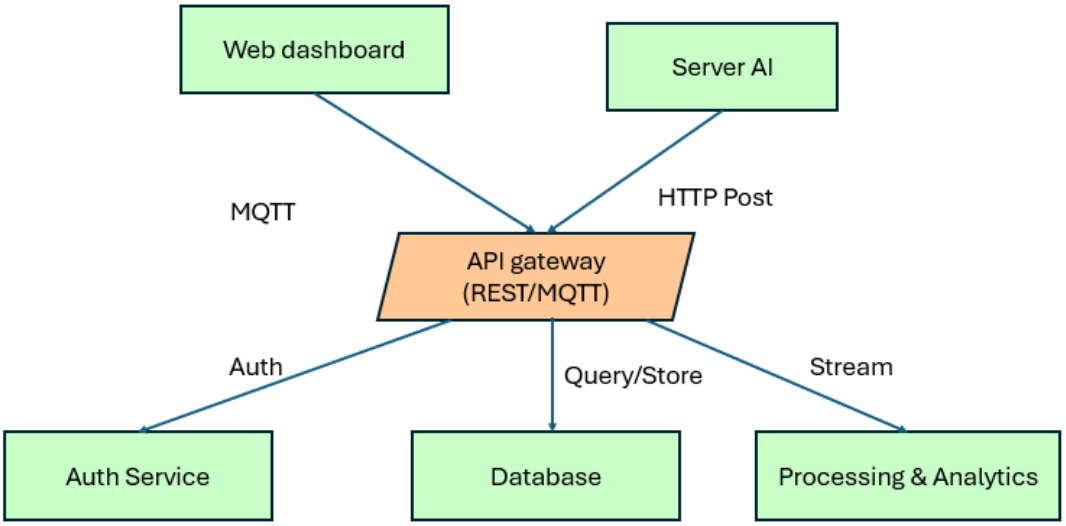
- Lựa chọn công nghệ kết nối: sử dụng Wifi để truyền dữ liệu từ esp32 lên server
- Mô hình giao thức HTTP và MQTT:
    - Giao thức HTTP (Hypertext Transfer Protocol): Dùng cho dữ liệu hình ảnh (Uplink)
        - Vai trò: Truyền tải file ảnh chụp từ người dùng lên AI Server để xử lý.
        - Phương thức: Sử dụng HTTP POST request.
        - Lý do: Dữ liệu ảnh có kích thước lớn (Payload lớn). HTTP hoạt động theo cơ chế Request-Response phù hợp cho việc upload file và chờ kết quả xác nhận từ Server mà không yêu cầu kết nối duy trì liên tục (stateless).
    - Giao thức MQTT (Message Queuing Telemetry Transport): Dùng cho điều khiển và trạng thái (Downlink/Real-time)
        - Vai trò: Truyền lệnh điều khiển mở khóa (Unlock Command) từ Web Dashboard xuống ESP32 và cập nhật trạng thái tủ (Online/Offline).
        - Thành phần: Sử dụng HiveMQ làm Message Broker trung gian.
        - Cơ chế: Publish/Subscribe.
        - Lý do: MQTT cực kỳ nhẹ (lightweight), độ trễ thấp (low latency) và hoạt động ổn định ngay cả khi mạng chập chờn, đảm bảo lệnh mở khóa được thực thi tức thì.
- An ninh truyền thông: sử dụng SSL/TLS để bảo vệ dữ liệu đường truyền, dùng cơ chế xác thực:
    - Mã hóa đường truyền (Transport Encryption):
        - Sử dụng SSL/TLS (Secure Sockets Layer/Transport Layer Security) cho cả hai giao thức.

- Cơ chế xác thực (Authentication):
    - Xác thực thiết bị (Device Auth): ESP32 kết nối với HiveMQ Broker phải cung cấp Username và Password hợp lệ mới được quyền Subscribe/Publish.
    - Xác thực API (API Security): Các request HTTP từ ESP32 lên Server phải đính kèm API Key hoặc Token trong Header để server nhận diện đúng thiết bị, tránh việc kẻ tấn công giả mạo request gửi ảnh rác lên hệ thống.

4.2.3. Thiết kế hạ tầng xử lý và lưu trữ

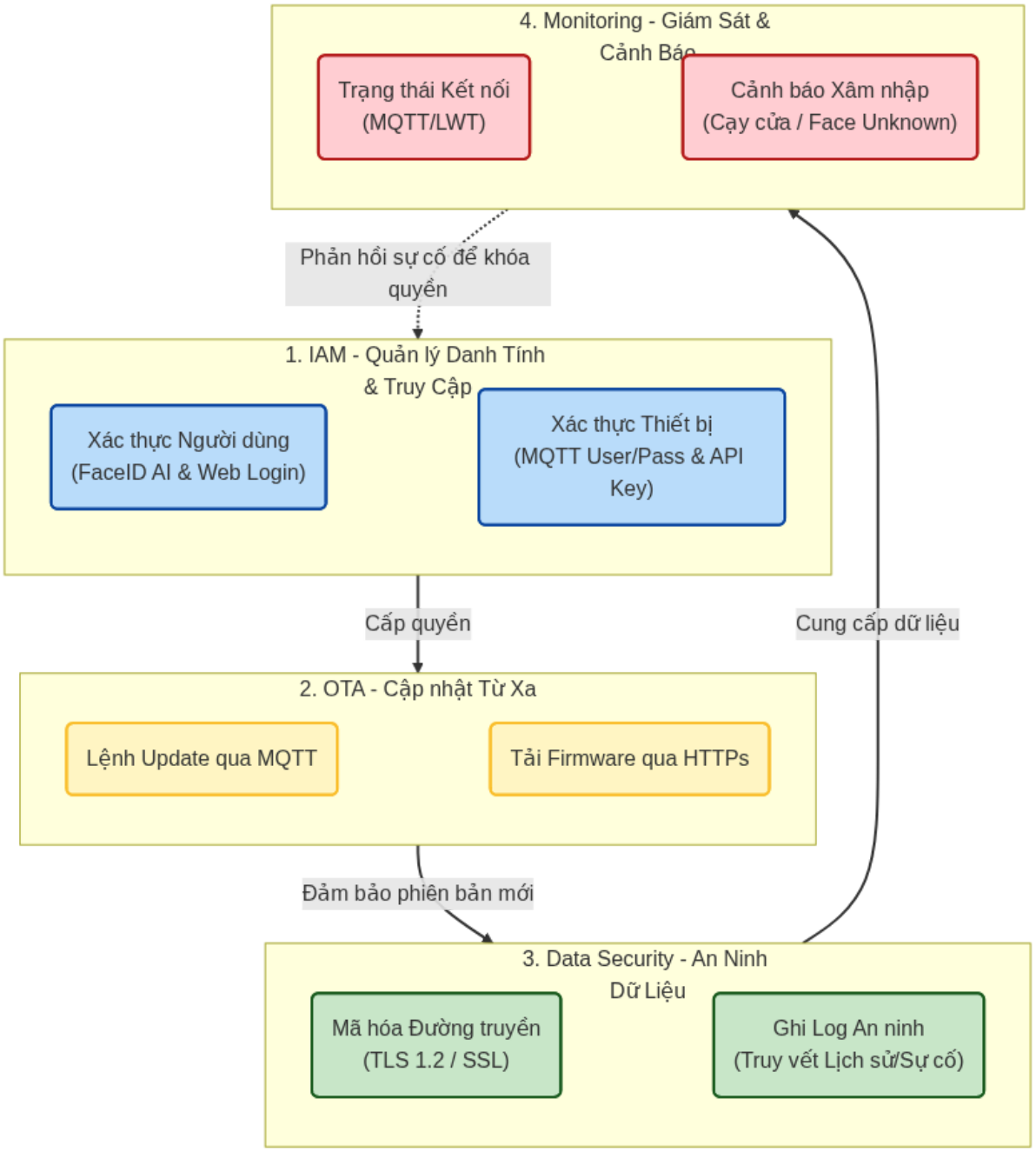
- Ứng dụng người dùng:
    - Loại ứng dụng: Web Dashboard (Bảng điều khiển trên trình duyệt).
    - Đối tượng sử dụng:
        - Quản trị viên (Admin): Giám sát toàn bộ hệ thống, xem logs, quản lý danh sách người dùng.
        - Người dùng cuối (User): Theo dõi trạng thái tủ cá nhân, xem lịch sử truy cập bản thân.
    - Công nghệ lõi:
        - Backend Runtime: Node.js (Sử dụng Express.js hoặc NestJS framework) để xử lý logic API tốc độ cao và quản lý kết nối thời gian thực.
        - Giao thức giao tiếp: HTTP/RESTful API (cho giao tiếp với AI Server/Client) và MQTT (cho giao tiếp với thiết bị qua HiveMQ).
- Chức năng chính:
    - Giám sát trạng thái thời gian thực (Real-time Monitoring):
        - Hệ thống hiển thị trực quan trạng thái kết nối của thiết bị ESP32 ngay trên Dashboard
        - Cơ chế: Server Node.js lắng nghe các gói tin từ HiveMQ để cập nhật giao diện người dùng
    - Quản lý truy cập & Định danh thông minh:
        - Kịch bản AI: Khi AI Server gửi kết quả nhận diện khuôn mặt thành công về Node.js, hệ thống tự động truy vấn cơ sở dữ liệu và hiển thị đầy đủ hồ sơ người dùng (Họ tên, Ảnh đại diện, ID, Thời gian truy cập) lên màn hình Dashboard
        - Kịch bản Tài khoản: Người dùng đăng nhập bằng Username/Password sẽ được chuyển hướng vào trang quản lý riêng biệt cho ngăn tủ được phân quyền sở hữu
    - Điều khiển từ xa (Remote Control):
        - Cung cấp giao diện nút nhấn cho phép người dùng hoặc Admin gửi lệnh "Mở khóa"
        - Lệnh điều khiển được Node.js Backend chuyển tiếp thành bản tin MQTT gửi tới Broker HiveMQ, đảm bảo độ trễ thấp nhất để Servo phản hồi tức thì
- Hệ thống Cảnh báo & An ninh (Alert System):
    - Cảnh báo xâm nhập: Nếu AI Server trả về kết quả "Unknown" (Không nhận diện được) quá số lần quy định hoặc phát hiện cố gắng mở tủ trái phép, Dashboard sẽ hiển thị cảnh báo đỏ

4.2.4. Thiết kế phần mềm ứng dụng



- API Gateway (REST/MQTT)
    - Vai trò: Nó đóng vai trò là cửa ngõ duy nhất tiếp nhận mọi yêu cầu từ bên ngoài (Web, AI, Thiết bị) và điều phối chúng đến các dịch vụ xử lý bên dưới.
    - Giao thức: Hỗ trợ song song 2 giao thức:
        - REST (HTTP): Để nhận dữ liệu từ Server AI (kết quả nhận diện).
        - MQTT: Để duy trì kết nối thời gian thực với Web Dashboard (và cả thiết bị ESP32 qua Broker).
- Các thành phần đầu vào: Hai thành phần này gửi dữ liệu và lệnh vào hệ thống:
    - Web Dashboard:
        - Chức năng: Giao diện quản lý dành cho Admin/User.
        - Kết nối (MQTT): Mũi tên ghi "MQTT" cho thấy Web Dashboard kết nối theo thời gian thực. Khi người dùng nhấn nút "Mở khóa" trên web, lệnh được gửi qua MQTT đến Gateway. Ngược lại, khi tủ mở, Gateway đẩy trạng thái cập nhật ngay lập tức lên Web
    - Server AI:
        - Chức năng: Xử lý hình ảnh khuôn mặt.
        - Kết nối (HTTP Post): Sau khi nhận diện xong, Server AI đóng gói kết quả (ID người dùng, độ chính xác) và gửi một bản tin HTTP POST đến API Gateway để thông báo
- Các thành phần xử lý & Lưu trữ: Đây là các module nội bộ của Node.js Server để xử lý yêu cầu:
    - Auth Service (Dịch vụ xác thực): Kiểm tra tính hợp lệ.Database (Cơ sở dữ liệu): Lưu thông tin người dùng, log lịch sử ra vào, trạng thái hiện tại của các ngăn tủ. Gateway thực hiện truy vấn (Query) để lấy thông tin hiển thị và lưu (Stored) lịch sử hoạt động
- Processing & Analytics (Xử lý & Phân tích: Đây là phần xử lý luồng dữ liệu (Stream)

4.2.5. Thiết kế bảo mật và quản lý



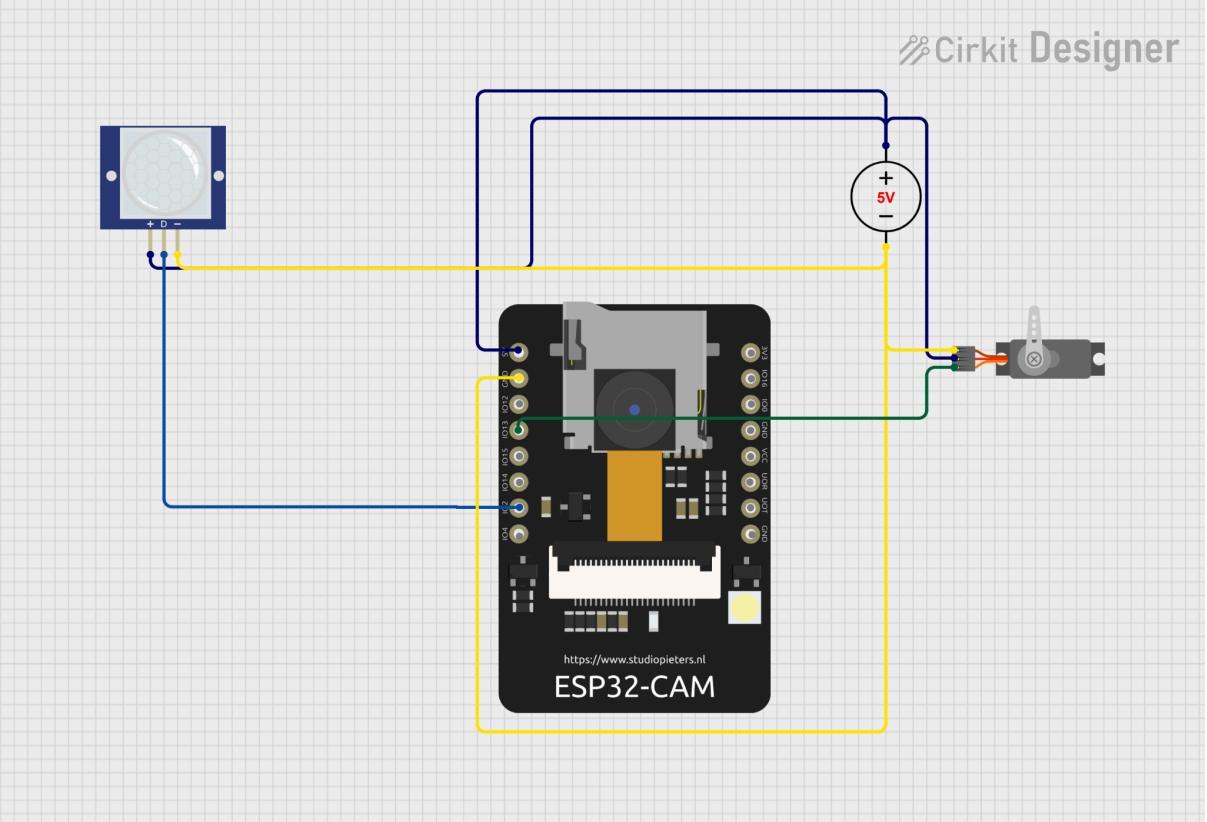
4.2.5.1. Thiết kế bảo mật và quản lý

- IAM: Là cổng vào đầu tiên. Hệ thống tách biệt xác thực cho Người (dùng khuôn mặt hoặc mật khẩu web) và Máy (ESP32 dùng token/password để kết nối Broker). Chỉ khi xác thực đúng mới được tham gia mạng
- OTA: Sau khi thiết bị được xác thực (IAM), hệ thống cho phép cập nhật firmware. Quá trình này được kích hoạt bằng lệnh nhẹ qua MQTT và tải dữ liệu nặng qua HTTPs an toàn.
- Data Security: Mọi dữ liệu đi qua OTA hay lệnh điều khiển đều được bao bọc bởi lớp Mã hóa (Encryption). Đồng thời, mọi hành động đều được Ghi Log (Logging) để phục vụ tra cứu.
- Monitoring: Dựa trên dữ liệu log và trạng thái kết nối thời gian thực (LWT), hệ thống thực hiện giám sát. Nếu phát hiện bất thường (ví dụ: thiết bị mất kết nối đột ngột hoặc có người lạ cố mở cửa), nó sẽ gửi cảnh báo.

- Mũi tên nét đứt quay lại IAM: Thể hiện "Chu trình liên tục". Ví dụ: Nếu Monitoring phát hiện một tài khoản có dấu hiệu tấn công (thử mật khẩu nhiều lần), nó sẽ phản hồi về IAM để tạm khóa tài khoản đó

4.2.6. Thiết kế triển khai vận hành

- **Sơ đồ lắp đặt thiết bị:**



Hình 4.2.6.1: Sơ đồ lắp đặt

  - Vị trí camera: lắp đặt trước cánh tủ
  - Vị trí servo: lắp âm bên trong khung tủ cố định
  - Vị trí cảm biến PIR: lắp tại mép cửa để phát hiện chuyển động
  - Bộ nguồn: đặt tại sau tủ
- Kịch bản kiểm thử:
  - Kiểm thử hiệu năng:
    - Mục tiêu: đánh giá độ trễ
    - Kịch bản: đo thời gian từ lúc người đứng trước camera -> ESP32 Cam chụp ảnh -> Server AI xử lý -> Lệnh mở khóa về tới ESP
  - Kiểm thử ổn định (Endurance Test):
    - Cho hệ thống chạy liên tục 24 giờ, thực hiện đóng/mở khóa định kỳ mỗi 5 phút để kiểm tra hiện tượng quá nhiệt của ESP32 hoặc rò rỉ bộ nhớ (Memory Leak) của Server.

CHƯƠNG 5: KẾT LUẬN

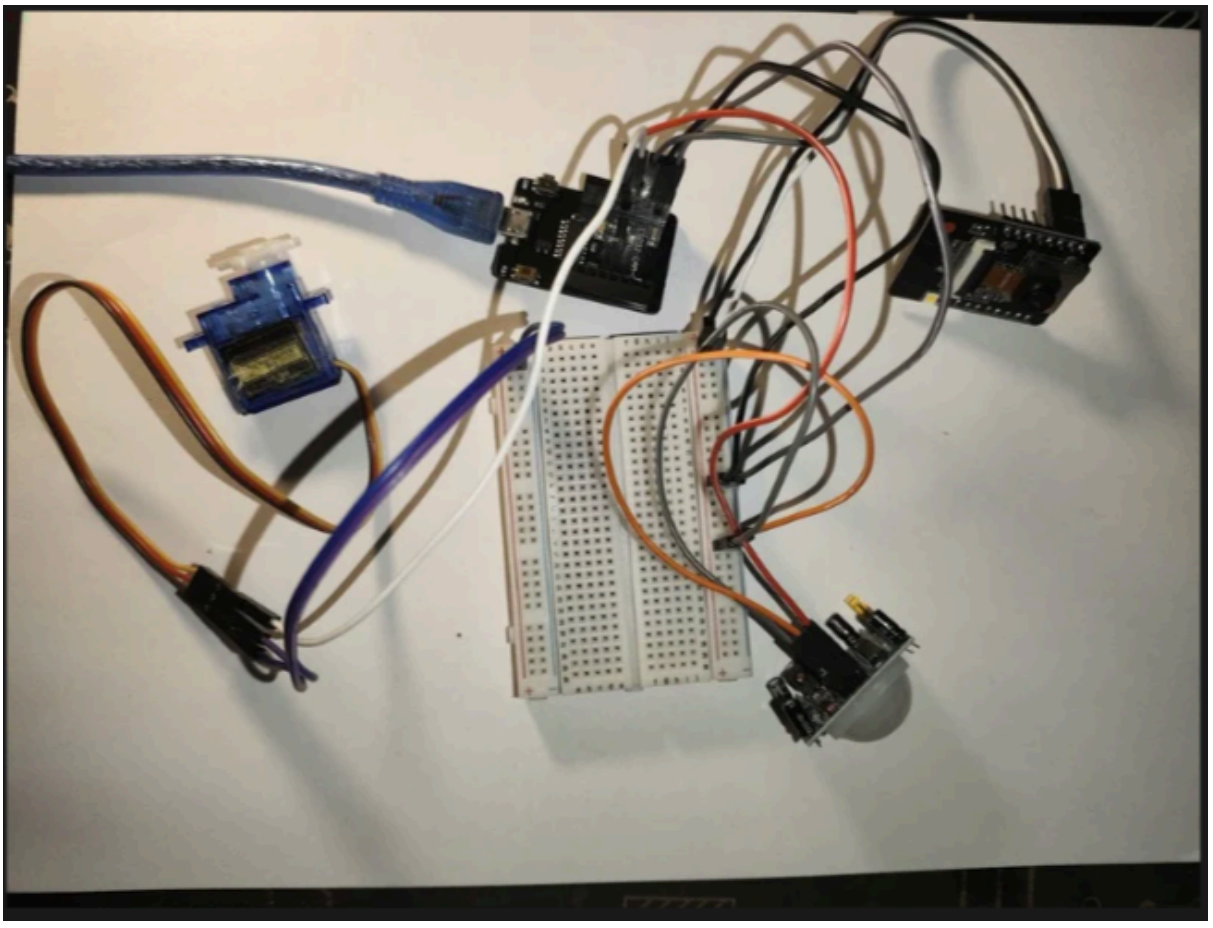
5.1. Kết quả đạt được

5.1.1. Về mặt lý thuyết.

- Qua việc tìm hiểu và nghiên cứu các thiết bị phần cứng như ESP32-CAM, cảm biến PIR (HC-SR501), Servo SG90 nhóm đã hiểu rõ cơ chế hoạt động và cách thức tương tác giữa các thành phần. Các giao thức truyền thông không dây (Wifi, MQTT) được sử dụng hiệu quả để truyền dữ liệu từ cảm biến đến máy chủ, đảm bảo dữ liệu được cập nhật liên tục theo thời gian thực.

5.1.2 Về mặt thực tiễn

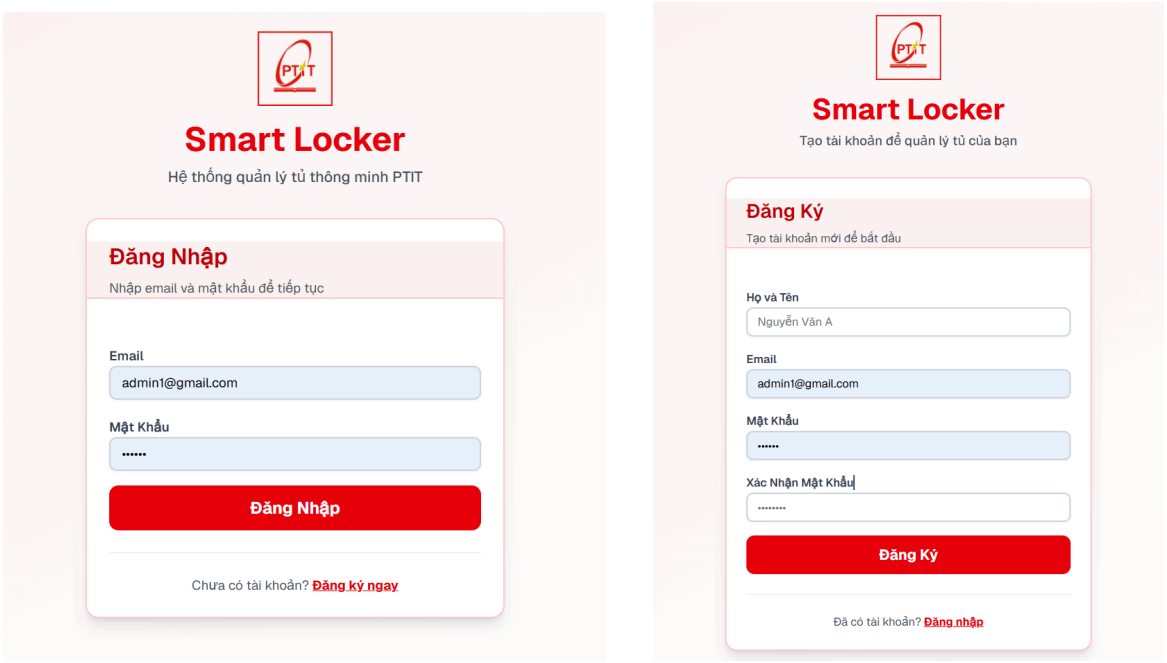
- Hệ thống chạy ổn định , có thể nhận diện chính xác khuôn mặt trong khoảng cách 30 - 50cm với accuracy trong khoảng 65-68%, có thể nhận định được giả mạo bằng hình ảnh..



Hình 5.1.2.1: Hệ thống thu được

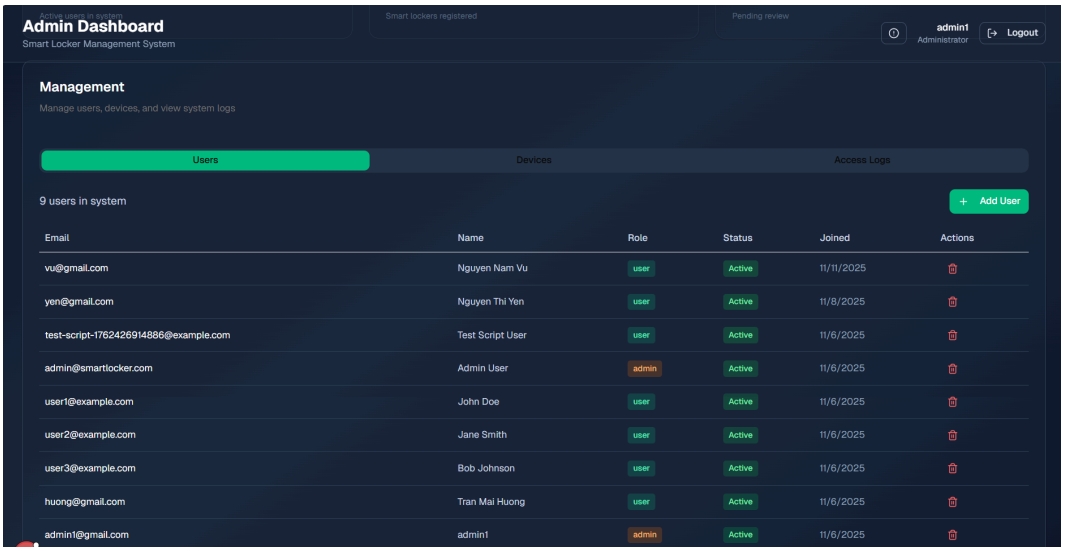
5.1.3. Giao diện ứng dụng (UI)

- **Giao diện Đăng nhập, Đăng ký:**



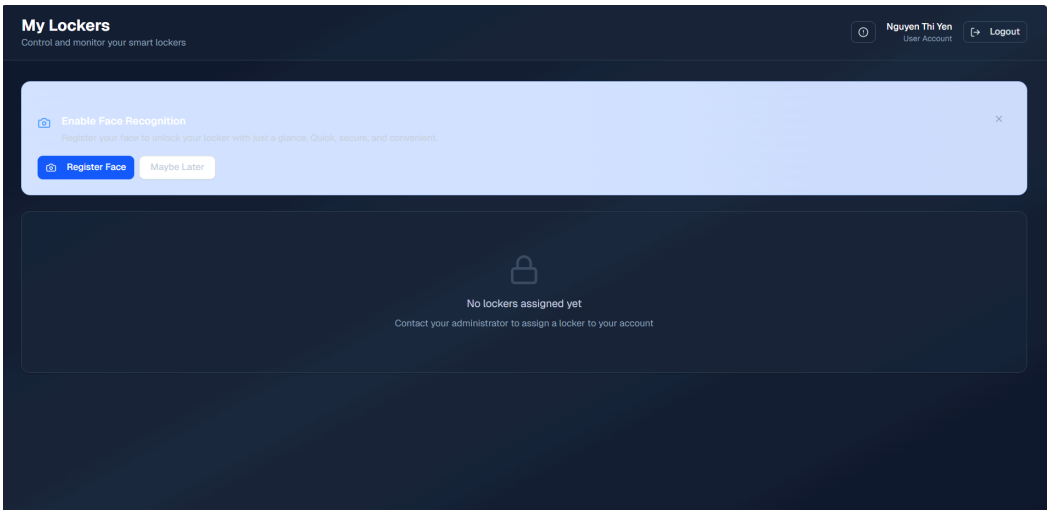
Hình 5.1.3.1: Giao diện đăng nhập đăng ký

- **Giao diện Dashboard Admin:**



Hình 5.1.3.2: Giao diện dashboard admin

- **Giao diện Dashboard User:**



Hình 5.1.3.3: Giao diện dashboard user

5.2. Hạn chế:

- ESP32CAM phụ thuộc vào tốc độ wifi.
- Kết nối giữa ESP32MB và laptop qua cáp USB chưa ổn định.
- Camera phản ứng không tốt với ánh sáng ngoài môi trường.
- Cần tối ưu hóa về mặt năng lượng, chưa có pin dự phòng.

5.3. Hướng phát triển:

- **Mở rộng thiết bị:** Hỗ trợ thêm nhiều tủ thông minh (mỗi tủ 1 ESP32-CAM) mà không cần thay đổi kiến trúc mạng.
- **Mở rộng người dùng:** Cơ sở dữ liệu khuôn mặt và tài khoản có thể mở rộng đến hàng trăm người.
- **Tích hợp hệ thống:** Có thể kết nối với các nền tảng IoT khác (Google Firebase, Node-RED, hoặc MQTT Cloud).

TÀI LIỆU THAM KHẢO

[1] Giới thiệu ESP32-CAM

[2] Hướng dẫn sử dụng servo SG90

[3] Hướng dẫn sử dụng ArcFace

[4] Model ArcFace

[5] Tài liệu YOLOFacev8

[6] Tài liệu hướng dẫn MQTT

[7] Tài liệu hướng dẫn cảm biến chuyển động PIR