A red and yellow logo

Description automatically generatedĐẠI HỌC BÁCH KHOA HÀ NỘI   
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI TẬP LỚN MÔN HỌC  
IOT VÀ ỨNG DỤNG**  
*Đề tài: Hệ thống Quản lý ra vào bằng Sinh trắc học*

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | TS. Đặng Tuấn Linh |
| Mã lớp: | 154060 |
| Nhóm: | 12 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Email | Lớp |
| 1 | Bùi Viết Lăng | 20215604 | lang.bv215604@sis.hust.edu.vn | IT2 – 03 K66 |
| 2 | Hà Văn Tăng | 20215638 | tang.hv215638@sis.hust.edu.vn | IT2 – 03 K66 |
| 3 | Nguyễn Quang Thuận | 20215649 | thuan.nv215649@sis.hust.edu.vn | IT2 – 03 K66 |
| 4 | Trịnh Hà Trung | 20215654 | trung.th215654@sis.hust.edu.vn | IT2 – 03 K66 |

Hà Nội, ngày 23 tháng 12 năm 2024

**MỤC LỤC**

[**Chương 1: Mô tả bài toán** 6](#_Toc185814566)

[**1.1.** **Tổng quan dự án** 6](#_Toc185814567)

[**1.2.** **Yêu cầu cơ bản của hệ thống** 6](#_Toc185814568)

[**1.3.** **Ngữ cảnh hệ thống được xây dựng** 6](#_Toc185814569)

[**Chương 2: Thiết kế, phương pháp** 7](#_Toc185814570)

[**2.1.** **Sơ đồ tổng quan hệ thống** 7](#_Toc185814571)

[**2.2.** **Thiết kế phần cứng** 8](#_Toc185814572)

[**2.2.1. Module ESP32** 8](#_Toc185814573)

[**2.2.2. ESP32 tích hợp module camera OV2640** 10](#_Toc185814574)

[**2.2.3. Cảm biến vân tay AS608** 11](#_Toc185814575)

[**2.2.4. LCD 16x2 tích hợp I2C** 13](#_Toc185814576)

[**2.3. Thiết kế Web Server** 14](#_Toc185814577)

[**2.3.1. Use case tổng quan** 14](#_Toc185814578)

[**2.3.2. Database** 15](#_Toc185814579)

[**Chương 3: Công cụ, thư viện sử dụng** 16](#_Toc185814580)

[**3.1.** **Arduino IDE** 16](#_Toc185814581)

[**3.2.** **Visual Studio Code** 16](#_Toc185814582)

[**3.3.** **Database MongoDB** 17](#_Toc185814583)

[**3.4.** **MQTT Broker HiveMQ** 17](#_Toc185814584)

[**3.5.** **Xây dựng web server bằng Nestjs** 18](#_Toc185814585)

[**3.6.** **Thư viện nhận diện khuôn mặt Face-api.js** 18](#_Toc185814586)

[**3.6.1. Tổng quan về face-api.js:** 18](#_Toc185814587)

[**3.6.2. Các mô hình chính được sử dụng:** 18](#_Toc185814588)

[**Chương 4: Thi công hệ thống, kết quả thực hiện** 20](#_Toc185814589)

[**4.1.** **Sơ đồ luồng hoạt động của hệ thống** 20](#_Toc185814590)

[**4.2.** **Kết quả thực hiện** 21](#_Toc185814591)

[**4.2.1.** **Phần cứng:** 21](#_Toc185814592)

[**4.2.2. Phần mềm:** 23](#_Toc185814593)

[**Chương 5: Kết luận** 24](#_Toc185814594)

**MỤC LỤC HÌNH ẢNH**

[Hình 1. Sơ đồ Tổng quan Hệ thống 7](#_Toc185814619)

[Hình 2. Sơ đồ Kết nối phần cứng 8](#_Toc185814620)

[Hình 3. Sơ đồ Thiết kế Module ESP32 8](#_Toc185814621)

[Hình 4. Sơ đồ Thiết kế Module ESP32-CAM 10](#_Toc185814622)

[Hình 5. Cảm biến vân tay AS608 11](#_Toc185814623)

[Hình 6. Mô phỏng thiết kế bên trong của cảm biến vân tay AS608 12](#_Toc185814624)

[Hình 7. Mô phỏng cách thức hoạt động của cảm biến vân tay AS608 12](#_Toc185814625)

[Hình 8. Mô hình Use case tổng quan 14](#_Toc185814626)

[Hình 9. Giao diện Arduino IDE 16](#_Toc185814627)

[Hình 10. Giao diện Visual Studio Code 16](#_Toc185814628)

[Hình 11. Sơ đồ Luồng hoạt động của hệ thống 20](#_Toc185814629)

[Hình 12. Thử nghiệm sản phẩm 21](#_Toc185814630)

[Hình 13. Mô hình lắp đặt thiết bị 22](#_Toc185814631)

[Hình 14. Giao diện Quản lý nhân sự 23](#_Toc185814632)

[Hình 15. Giao diện Quản lý vào ra 23](#_Toc185814633)

**LỜI NÓI ĐẦU**

Trong xu thế công nghệ 4.0, sinh trắc học đang trở thành một lĩnh vực quan trọng để tăng cường tính bảo mật và hiệu quả trong các hệ thống quản lý. Đề tài "Hệ thống Quản Lý Vào Ra Bằng Sinh Trắc Học" được thực hiện với mục đích tăng cường tính an ninh và đưa ra một giải pháp hiện đại cho các hệ thống quản lý nhân sự. Em hy vọng rằng dự án này sẽ mang lại những đóng góp hữu ích trong việc áp dụng công nghệ sinh trắc học vào đời sống thường nhật.

Nhóm em xin gửi lời cảm ơn Thầy đã hướng dẫn, đưa ra những góp ý giúp nhóm em hoàn thiện Project này.

# **Chương 1: Mô tả bài toán**

## **Tổng quan dự án**

Hệ thống quản lý vào ra bằng sinh trắc học được thiết kế nhằm tối ưu hóa việc giám sát và điều khiển quy trình ra vào của nhân viên trong các tổ chức, doanh nghiệp. Dự án sử dụng hai công nghệ chính: nhận diện khuôn mặt và vân tay. Hai công nghệ này đảm bảo độ chính xác cao và khó có thể giả mạo, đem lại độ bảo mật vượt trội so với các phương pháp truyền thống.

Hệ thống bao gồm:

* **Thiết bị phần cứng:** module camera nhận diện khuôn mặt, cảm biến vân tay, và module ESP32.
* **Phần backend:** Nest.js và Face-API.
* **Phần frontend:** Giao diện Web để quản lý nhân sự và theo dõi lịch sử ra vào.

## **Yêu cầu cơ bản của hệ thống**

* **Độ chính xác cao:** Hệ thống phải nhận diện khuôn mặt và vân tay với độ chính xác cao, giảm tối đa nguy cơ sai lệch.
* **Tính thời gian thực:** Hệ thống cần xử lý nhanh chóng, cho phép truy cập ngay lập tức.
* **Bảo mật cao:** Thông tin sinh trắc học cần được mã hóa để đảm bảo an toàn.
* **Khả năng tích hợp:** Hệ thống có khả năng kết hợp với các hệ thống quản lý khác như ERP hoặc CRM.
* **Dễ sử dụng:** Giao diện đơn giản, tính năng rõ ràng.

## **Ngữ cảnh hệ thống được xây dựng**

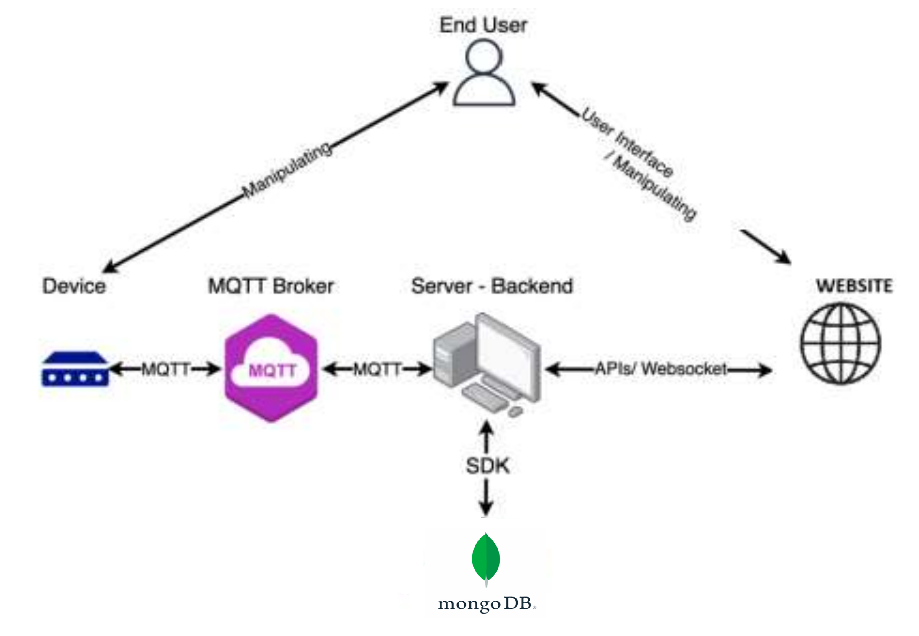
Hệ thống quản lý vào ra bằng sinh trắc học được xây dựng trong bối cảnh nhu cầu tăng cao về tính bảo mật và hiệu quả trong các tổ chức, doanh nghiệp. Các lĩnh vực có thể áp dụng bao gồm:

* **Doanh nghiệp:** Quản lý nhân viên ra vào, theo dõi giờ làm việc.
* **Trường học:** Quản lý sinh viên vào ra các khu vực hạn chế.
* **Các toà nhà văn phòng:** Kiểm soát lối vào ra và bảo vệ.
* **Các khu công nghiệp:** Tăng cường an ninh cho khu nhà máy và nhân công.

Bằng cách kết hợp những tiến bộ công nghệ vào thực tiễn, hệ thống hứa hẹn mang lại giá trị gia tăng đáng kể cho người sử dụng.

# **Chương 2: Thiết kế, phương pháp**

* 1. **Sơ đồ tổng quan hệ thống**



Hình 1. Sơ đồ Tổng quan Hệ thống

## **Thiết kế phần cứng**

A diagram of a circuit board

Description automatically generated

Hình 2. Sơ đồ Kết nối phần cứng

### **2.2.1. Module ESP32**

A diagram of a circuit board

Description automatically generated

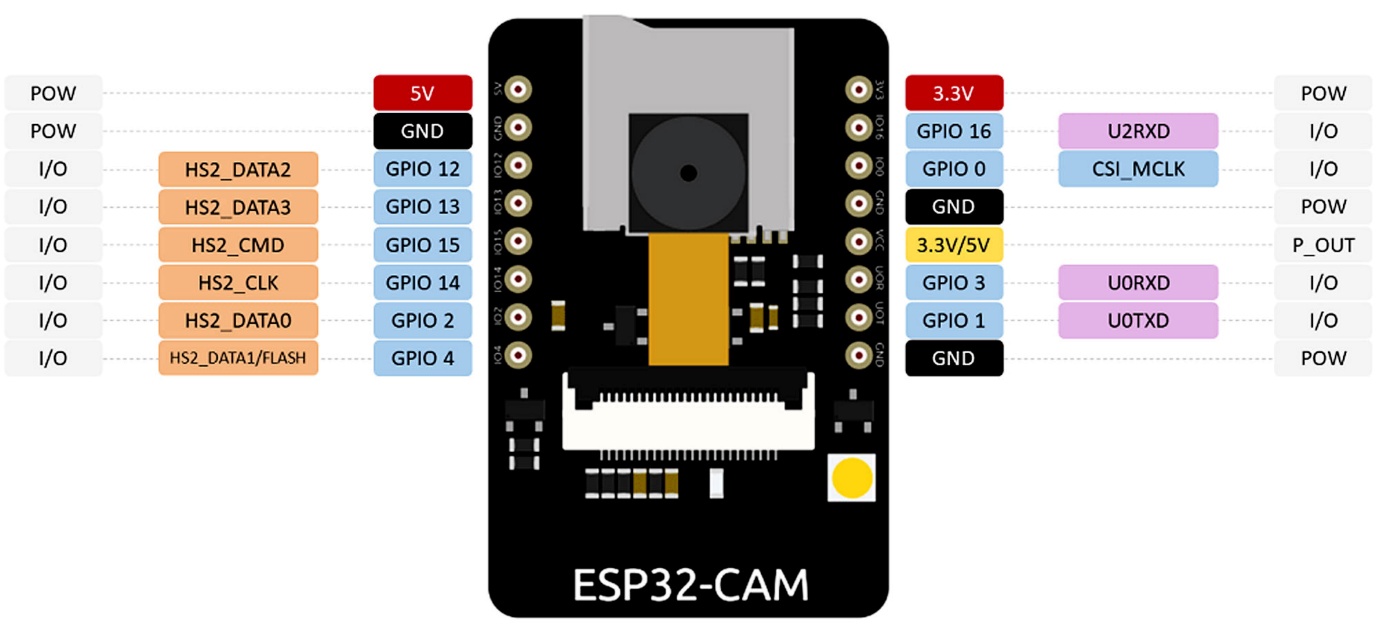
Hình 3. Sơ đồ Thiết kế Module ESP32

ESP32 Dev Module là một bo mạch phát triển mạnh mẽ, đa năng, dựa trên vi điều khiển ESP32 của Espressif Systems. Đây là một trong những lựa chọn phổ biến nhất cho các dự án IoT nhờ hiệu năng cao, hỗ trợ kết nối không dây, và khả năng lập trình linh hoạt.

**Các đặc điểm nổi bật:**

* **Vi điều khiển ESP32:**
* Tích hợp chip ESP32 với hai nhân CPU Xtensa 32-bit LX6, hỗ trợ hoạt động đơn hoặc đa lõi.
* Tần số xung nhịp tối đa 240MHz, hiệu năng cao đáp ứng các ứng dụng phức tạp.
* Hỗ trợ Wi-Fi 802.11 b/g/n và Bluetooth 4.2 (BLE + Classic), lý tưởng cho các ứng dụng IoT và kết nối không dây.
* **Bộ nhớ:**
* RAM: 520KB SRAM nội bộ.
* Bộ nhớ Flash: 4MB (có thể mở rộng tùy phiên bản), đáp ứng tốt cho các ứng dụng lưu trữ mã lệnh và dữ liệu.
* **Các cổng giao tiếp:**
* GPIO: Lên đến 34 chân I/O, có thể lập trình và hỗ trợ các giao thức giao tiếp như UART, SPI, I2C, I2S, PWM.
* ADC và DAC: 16 kênh ADC (12-bit), 2 kênh DAC (8-bit), hỗ trợ xử lý tín hiệu analog.
* Hỗ trợ giao tiếp CAN và Ethernet (qua module bổ sung).
* **Khả năng lập trình:**
* Tương thích với nhiều nền tảng phát triển: Arduino IDE, ESP-IDF, MicroPython, PlatformIO, và Lua.
* Tích hợp các công cụ phát triển mạnh mẽ như OTA (Over-The-Air) để cập nhật chương trình từ xa.
* **Nguồn cấp:**
* Hỗ trợ nguồn cấp từ 3.3V đến 5V thông qua cổng micro-USB hoặc các chân pin.
* Có tích hợp bộ quản lý nguồn giúp tối ưu hiệu năng và tiêu thụ điện năng.

### **2.2.2. ESP32 tích hợp module camera OV2640**



Hình 4. Sơ đồ Thiết kế Module ESP32-CAM

ESP32-CAM là một module phát triển mạnh mẽ, nhỏ gọn và chi phí thấp, được tích hợp vi điều khiển ESP32 và camera OV2640. Module này do AI-Thinker phát triển, được thiết kế để hỗ trợ các ứng dụng IoT với khả năng chụp ảnh và truyền dữ liệu qua mạng Wi-Fi. Đây là một giải pháp lý tưởng cho các dự án như giám sát an ninh, nhận diện khuôn mặt, hoặc các ứng dụng yêu cầu chụp ảnh và xử lý dữ liệu hình ảnh.

**Các đặc điểm nổi bật:**

* **Vi điều khiển ESP32:**
* Sử dụng chip ESP32 với hai nhân CPU Xtensa 32-bit LX6.
* Tần số xung nhịp tối đa 240MHz, tích hợp bộ nhớ Flash 4MB.
* Hỗ trợ kết nối Wi-Fi (802.11 b/g/n) và Bluetooth 4.2.
* **Camera OV2640:**
* Độ phân giải tối đa 2MP (1600x1200).
* Hỗ trợ nhiều chế độ chụp, bao gồm JPEG, BMP, và YUV.
* Góc nhìn rộng và khả năng chỉnh tiêu cự thủ công.
* **Các cổng giao tiếp:**
* GPIO: Có thể kết nối cảm biến, relay, hoặc các thiết bị ngoại vi khác.
* UART, I2C, và SPI để giao tiếp với các thiết bị khác.
* **Bộ nhớ:**
* RAM tích hợp 520KB và PSRAM 4MB (phiên bản nâng cấp), hỗ trợ xử lý các ứng dụng yêu cầu bộ nhớ lớn như xử lý hình ảnh và video.
* **Khả năng mở rộng:**
* Khe cắm thẻ MicroSD hỗ trợ lưu trữ dữ liệu.
* Tích hợp đèn flash LED để hỗ trợ chụp ảnh trong điều kiện ánh sáng yếu.

### **2.2.3. Cảm biến vân tay AS608**

A black rectangular object with colorful wires

Description automatically generated

Hình 5. Cảm biến vân tay AS608

AS608 là loại cảm biến vân tay quang học. Sử dụng ánh sáng nhìn thấy và một máy ảnh, nó hoạt động bằng cách chụp một bức ảnh của dấu vân tay của bạn và chuyển đổi nó thành dữ liệu số. Trong hơn nửa thế kỷ qua, các sản phẩm cảm biến vân tay sinh quang học đã là lựa chọn hàng đầu của nhân viên thực thi pháp luật và các cơ quan chính phủ do tính kinh tế, độ tin cậy và tốc độ chính xác của chúng.

* **Thiết kế phần cứng của cảm biến**

Một cảm biến vân tay quang học bao gồm một lớp bảo vệ bằng kính, một nguồn sáng chiếu sáng như đèn LED, một lăng kính phản xạ ánh sáng từ bề mặt ngón tay, và một ống kính tập trung ánh sáng phản xạ lên cảm biến hình ảnh CCD (Charge Coupled Device) - cảm biến này chụp lại tất cả các đường vân và rãnh đặc trưng để nhận dạng vân tay. Ngoài ra, cảm biến còn chứa một MCU (Bộ điều khiển vi mô) hoặc DSP (Bộ xử lý tín hiệu số) hỗ trợ điều khiển, chuyển đổi dữ liệu và phân tích. Cuối cùng, sử dụng các giao diện UART, SPI hoặc USB, dữ liệu số từ cảm biến vân tay quang học có thể được chuyển đến máy tính hoặc thiết bị di động. Các thành phần chính của một cảm biến vân tay quang học như sau :

* Lớp bảo vệ bằng kính - Hoạt động như một lớp bảo vệ chống bụi và nước.
* Nguồn sáng (đèn LED) - Giúp chiếu sáng ngón tay, làm nổi bật lên đường vân, rãnh của ngón tay
* Cảm biến hình ảnh CCD hoặc CMOS - Chụp lại chi tiết các đường vân và rãnh của dấu vân tay.
* MCU hoặc DSP - Điều khiển, chuyển đổi dữ liệu và phân tích.
* Giao tiếp UART, SPI hoặc USB - Chuyển dữ liệu dấu vân tay đến máy tính hoặc thiết bị di động

Diagram of a diagram of a light beam

Description automatically generated

Hình 6. Mô phỏng thiết kế bên trong của cảm biến vân tay AS608

* **Cách thức hoạt động của cảm biến**

Cảm biến vân tay quang học sử dụng công nghệ hình ảnh quang học để chụp và số hóa hình ảnh ngón tay của một cá nhân. Người dùng chỉ cần đặt ngón tay lên bề mặt kính của máy quét quang học, sau đó các tia sáng chiếu sáng được phát ra để tạo ra hình ảnh của các đường vân và rãnh. Sau đó, các phản xạ này được thu thập bởi một ống kính trước khi được tập trung lên máy ảnh CCD hoặc CMOS để chụp lại dấu vân tay của bạn, và chuyển đổi nó thành hình ảnh số bằng bộ chuyển đổi A/D (bộ chuyển đổi tương tự sang số). Hình ảnh vân tay số này sau đó được nhị phân hóa và các đặc điểm chi tiết của nó được trích xuất bằng nhiều thuật toán khác nhau để tạo ra một mẫu vân tay sinh trắc học. Sau khi quá trình này hoàn tất và mẫu được lưu lại

Diagram of a prism with arrows pointing to the surface

Description automatically generated

Hình 7. Mô phỏng cách thức hoạt động của cảm biến vân tay AS608

* **Thông số kỹ thuật**
* Nguồn cung cấp từ 3.6-6V. Dòng 100mA-150mA.
* Phương thức giao tiếp UART/USB 1.1.
* Chế độ so sánh kiểm tra: có 2 chế độ 1:1 và 1: N.
* Tốc độ Baud: từ 9600-115200bps. Mặc định của module là 57600bps.
* Kích thước tập tin: 256bytes.
* Thời gian kiểm tra: nhỏ hơn 0.5s.
* Kích thước mỗi mẫu: 512bytes.
* Khả năng lưu trữ trong flash: 256 mẫu.
* Mức độ bảo mật: Cấp 5(cao nhất).
* Độ sai số cho phép nhầm: (FAR - False Acceptance Rate): nhỏ hơn 0.001%.
* Độ sai số từ chối nhầm: (FRR - False Reject Rate): nhỏ hơn 0.1%.
* Thời gian tìm kiếm trung bình (1:1000): nhỏ hơn 1s.

### **2.2.4. LCD 16x2 tích hợp I2C**

**LCD I2C** là một loại màn hình LCD (Liquid Crystal Display) được tích hợp thêm giao tiếp I2C (Inter-Integrated Circuit), giúp việc kết nối và điều khiển màn hình trở nên đơn giản và tiết kiệm chân GPIO trên vi điều khiển.

**Các đặc điểm nổi bật:**

* **Giao tiếp I2C:**
* Chỉ sử dụng **2 dây** để giao tiếp: SDA (Dữ liệu) và SCL (Xung nhịp), thay vì 8 hoặc nhiều hơn khi dùng giao tiếp song song.
* Có thể dễ dàng kết nối nhiều thiết bị I2C trên cùng một bus, giúp mở rộng hệ thống mà không cần tăng số lượng chân GPIO.
* **Khả năng hiển thị:**
* Hiển thị ký tự và số, hỗ trợ các ký tự đặc biệt và ký tự tùy chỉnh (Custom Characters).
* Nền LED với đèn nền có thể bật/tắt hoặc điều chỉnh độ sáng.
* Tùy chọn LCD với màu nền xanh hoặc đen, ký tự trắng hoặc vàng.
* **Nguồn cấp:**
* Điện áp hoạt động: 5V hoặc 3.3V (tùy model).
* Tiêu thụ năng lượng thấp, phù hợp cho các dự án yêu cầu tiết kiệm điện.

## **2.3. Thiết kế Web Server**

### **2.3.1. Use case tổng quan**

A diagram of a person with text

Description automatically generated

Hình 8. Mô hình Use case tổng quan

### **2.3.2. Database**

* Schema User (lưu những thông tin cơ bản của người dùng)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| \_id | ObjectId |  | ID duy nhất cho mỗi bản ghi, Tự động sinh bởi MongoDB |
| id\_nvien | string | Required, Unique | Mã nhân viên |
| name | string | Required, Unique | Tên nhân viên |
| finger\_id | number | Required, Unique | finger\_id |
| faceDescriptor | ObjectId | Liên kết tới FaceDescriptor, Unique | Tham chiếu đến mô tả khuôn mặt |
| userlog | Mảng ObjectId | Liên kết tới UserLog | Tham chiếu đến lịch sử hoạt động của user |

* Schema FaceDescriptor (lưu lại thông tin về dữ liệu khuôn mặt)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| \_id | ObjectId |  | ID duy nhất cho mỗi bản ghi, tự động sinh bởi MongoDB |
| descriptor | string | Required | Chuỗi mô tả đặc trưng của khuôn mặt |
| iv | string | Required | Chuỗi IV (Initialization Vector) sử dụng trong mã hóa |
| user | ObjectId | Required, Unique | Liên kết với User để ánh xạ người dùng |

* Schema User\_log (lưu lại thông tin về lịch sử vào ra của người dùng)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| \_id | ObjectId |  | ID duy nhất cho mỗi bản ghi, tự động sinh bởi MongoDB |
| user | ObjectId | Required | Liên kết với User để ánh xạ người dùng |
| date | Date | Required | Ngày ghi nhận hoạt động |
| time\_in | string | Required | Thời gian vào |
| time\_out | string |  | Thời gian ra (nếu có) |

# **Chương 3: Công cụ, thư viện sử dụng**

## **Arduino IDE**

A screenshot of a computer

Description automatically generated**Arduino IDE (Integrated Development Environment)** là một công cụ mã nguồn mở được sử dụng để lập trình và phát triển các ứng dụng nhúng trên các vi điều khiển của nền tảng Arduino. Arduino IDE hỗ trợ viết code bằng ngôn ngữ lập trình Arduino (dựa trên C/C++), biên dịch, và tải code trực tiếp lên các bo mạch Arduino thông qua kết nối USB.

Hình 9. Giao diện Arduino IDE

## **Visual Studio Code**

**Visual Studio Code (VS Code)** là một trình soạn thảo mã nguồn mở, nhẹ và mạnh mẽ, được phát triển bởi Microsoft. VS Code hỗ trợ nhiều ngôn ngữ lập trình như Python, JavaScript, …nhờ vào hệ thống mở rộng phong phú qua các tiện ích (extensions).

A screen shot of a computer program

Description automatically generatedỞ project này, VS Code được sử dụng để phát triển phía back-end và front-end của web.

Hình 10. Giao diện Visual Studio Code

## **Database MongoDB**

**MongoDB** là một cơ sở dữ liệu NoSQL mã nguồn mở, được thiết kế để lưu trữ dữ liệu dưới dạng JSON. Với kiến trúc không sử dụng bảng truyền thống như SQL mà thay vào đó là các collection, MongoDB phù hợp để xử lý dữ liệu không có cấu trúc hoặc bán cấu trúc, cho phép mở rộng quy mô theo chiều ngang một cách dễ dàng.

**Sự phù hợp của MongoDB trong hệ thống IoT:**

* **Linh hoạt với dữ liệu không cấu trúc:** Dữ liệu IoT thường có nhiều loại hình và cấu trúc thay đổi, điều mà MongoDB xử lý tốt nhờ lưu trữ dạng JSON/BSON.
* **Tốc độ ghi cao:** MongoDB tối ưu hóa ghi dữ liệu nhanh, phù hợp cho các hệ thống IoT nhận dữ liệu thời gian thực.
* **Truy vấn mạnh mẽ:** Hỗ trợ tìm kiếm nâng cao và xử lý dữ liệu phân tích, hữu ích cho việc giám sát và quản lý thiết bị IoT.
* **Khả năng phân cụm (sharding):** Giúp phân phối dữ liệu qua nhiều server, đảm bảo hiệu suất khi hệ thống mở rộng.

Với khả năng linh hoạt và hiệu suất cao, MongoDB là một lựa chọn lý tưởng cho việc lưu trữ và phân tích dữ liệu trong các ứng dụng IoT.

## **MQTT Broker HiveMQ**

**MQTT Broker** là một thành phần trung tâm trong giao thức **MQTT (Message Queuing Telemetry Transport)**, chịu trách nhiệm quản lý và điều phối việc truyền thông tin giữa các thiết bị trong hệ thống. MQTT Broker hoạt động theo mô hình **publish/subscribe**, nơi các thiết bị (clients) gửi dữ liệu (publish) hoặc đăng ký nhận dữ liệu (subscribe) qua các "chủ đề" (topics).

**Đặc điểm chính của MQTT Broker:**

* **Nhẹ và hiệu quả:** Tối ưu hóa cho các hệ thống có tài nguyên hạn chế và đường truyền băng thông thấp.
* **Hỗ trợ QoS (Quality of Service):** Đảm bảo mức độ tin cậy khác nhau khi truyền thông tin.
* **Khả năng mở rộng:** Dễ dàng tích hợp và mở rộng để xử lý hàng triệu kết nối đồng thời

## **Xây dựng web server bằng Nestjs**

**NestJS** là một framework back-end mạnh mẽ được thiết kế dựa trên TypeScript. NestJS nổi bật với kiến trúc module hóa, cho phép xây dựng các ứng dụng server-side có khả năng mở rộng, bảo trì tốt và dễ dàng tích hợp với các công nghệ khác.

**Sự phù hợp của NestJS trong xây dựng server cho hệ thống IoT:**

* **Hỗ trợ giao thức đa dạng:** NestJS dễ dàng tích hợp các giao thức phổ biến trong IoT như MQTT, WebSocket, và HTTP, giúp giao tiếp linh hoạt giữa thiết bị và server.
* **Xử lý thời gian thực:** Với tích hợp WebSocket và các thư viện như RxJS, NestJS hỗ trợ xử lý luồng dữ liệu thời gian thực, đáp ứng nhu cầu của hệ thống IoT.
* **Kiến trúc module hóa:** Giúp tổ chức mã nguồn rõ ràng, dễ mở rộng khi số lượng thiết bị và dịch vụ tăng lên.
* **Tích hợp dễ dàng với cơ sở dữ liệu:** Hỗ trợ nhiều cơ sở dữ liệu (trong đó có MongoDB) kết hợp với TypeOrm giúp xây dựng database dễ dàng để lưu trữ và quản lý dữ liệu IoT một cách hiệu quả.

Với tính linh hoạt, hiệu suất cao, và khả năng mở rộng, NestJS là một lựa chọn lý tưởng để xây dựng server trong các hệ thống IoT, hỗ trợ quản lý thiết bị, xử lý dữ liệu thời gian thực, và triển khai các dịch vụ thông minh.

## **Thư viện nhận diện khuôn mặt Face-api.js**

### **3.6.1. Tổng quan về face-api.js:**

***face-api.js*** là một thư viện ***JavaScript*** được xây dựng trên nền tảng ***tensorflow.js***, được phát triển để thực hiện các tác vụ nhận diện và phân tích khuôn mặt trong trình duyệt.

### **3.6.2. Các mô hình chính được sử dụng:**

#### **3.6.2.1. SSD (Single Shot Multibox Detector):**

* Đây là mô hình chính để phát hiện khuôn mặt
* SSD hoạt động bằng cách:
* Tạo ra nhiều khung hình (anchor boxes) với các tỷ lệ và kích thước khác nhau
* Quét qua ảnh đầu vào và dự đoán xem mỗi khung có chứa khuôn mặt hay không
* Sử dụng các lớp tích chập để trích xuất đặc trưng tại nhiều tỉ lệ khác nhau
* Kết hợp các dự đoán từ nhiều lớp để có kết quả cuối cùng

#### **3.6.2.2. FaceNet:**

* Được sử dụng để tạo ra các face embeddings (vector đặc trưng của khuôn mặt)
* Mô hình này được huấn luyện bằng triplet loss
* Nó biến đổi ảnh khuôn mặt thành vector 128 chiều, trong đó:
* Khoảng cách Euclidean giữa các vector của cùng một người sẽ nhỏ
* Khoảng cách giữa các vector của người khác nhau sẽ lớn

#### **3.6.2.3. Quy trình xử lí cụ thể**

* Phát hiện khuôn mặt:
* Ảnh đầu vào được resize về nhiều kích thước khác nhau (tạo image pyramid)
* Mô hình SSD được áp dụng trên mỗi mức của pyramid để phát hiện khuôn mặt
* Non-maximum suppression được áp dụng để loại bỏ các khung trùng lặp
* Kết quả là các bounding box xác định vị trí khuôn mặt
* Trích xuất điểm đặc trưng (landmarks):
* 68 điểm đặc trưng được xác định trên mỗi khuôn mặt
* Các điểm này bao gồm: mắt, mũi, miệng, lông mày, đường viền mặt
* Sử dụng mạng CNN riêng được huấn luyện cho tác vụ này
* Tạo face embeddings:
* Khuôn mặt được căn chỉnh dựa trên các điểm đặc trưng
* FaceNet được sử dụng để tạo vector đặc trưng

# **Chương 4: Thi công hệ thống, kết quả thực hiện**

## **Sơ đồ luồng hoạt động của hệ thống**

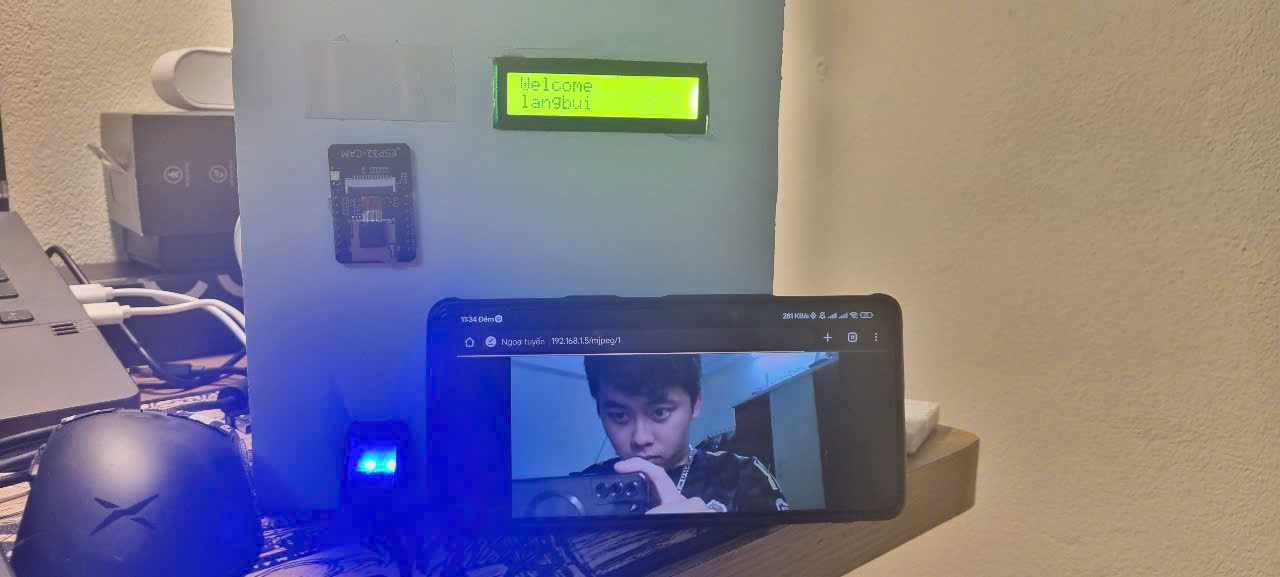
A diagram of a computer

Description automatically generated

Hình 11. Sơ đồ Luồng hoạt động của hệ thống

## **Kết quả thực hiện**

### **Phần cứng:**

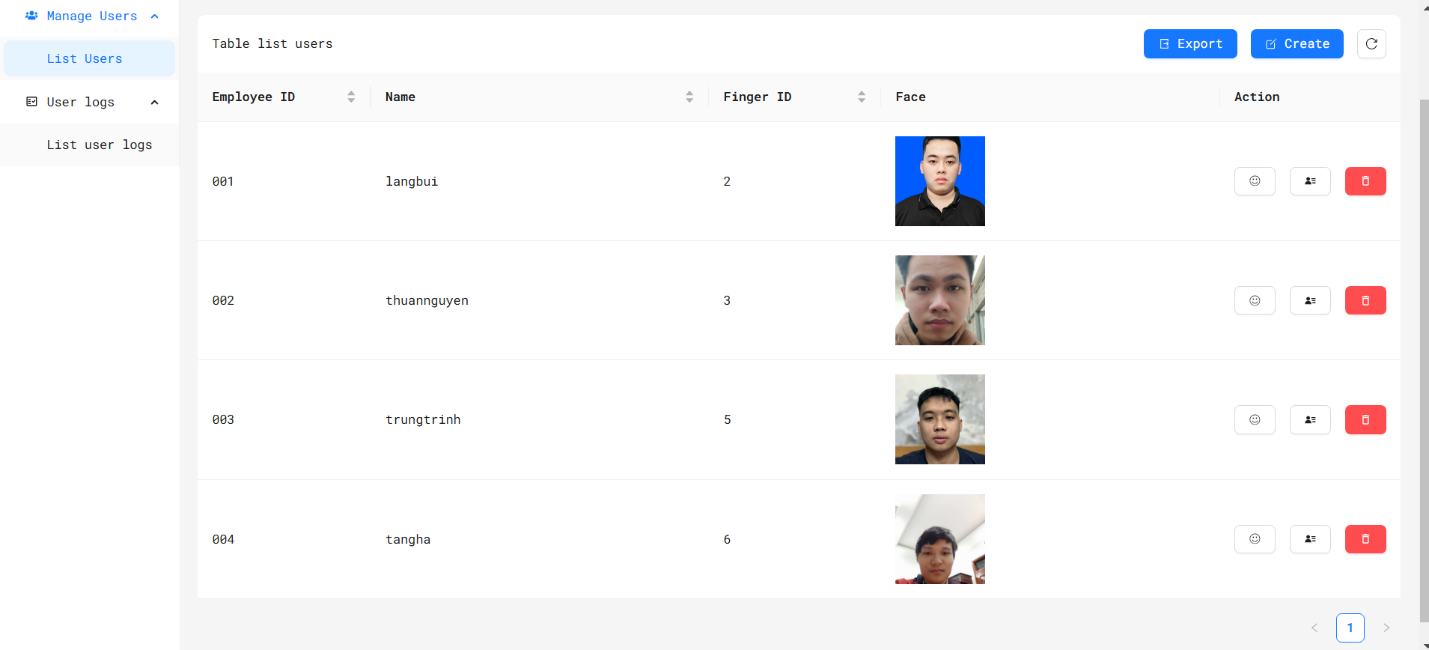


Hình 12. Thử nghiệm sản phẩm

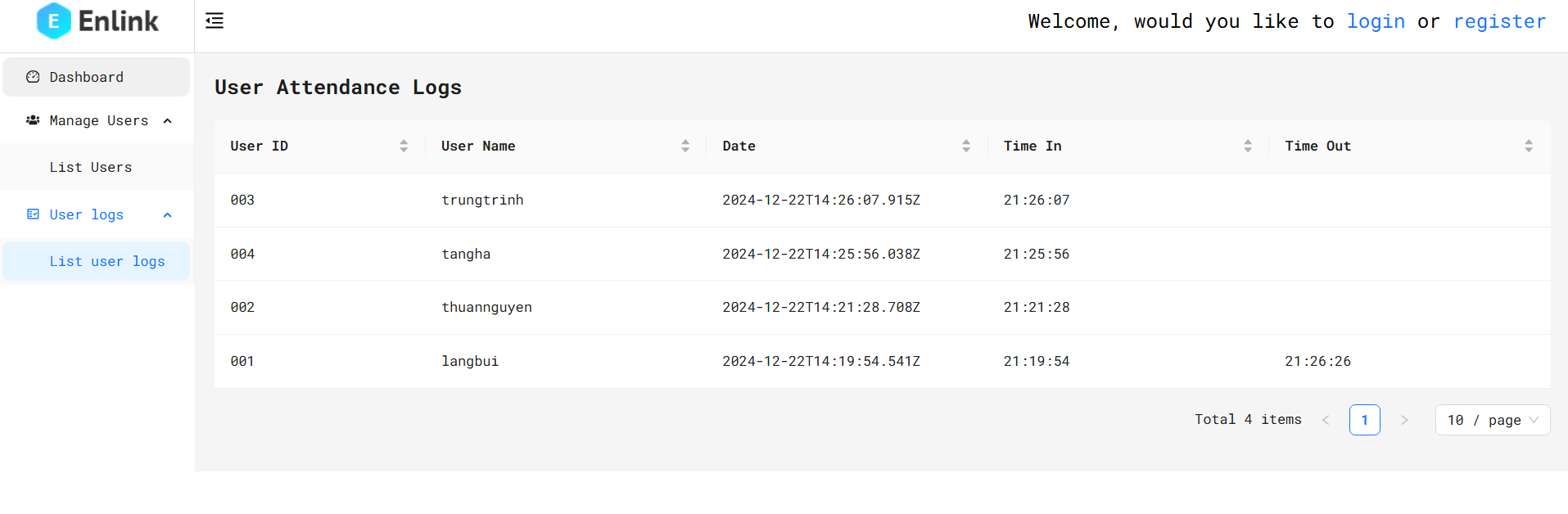


Hình 13. Mô hình lắp đặt thiết bị

### **4.2.2. Phần mềm:**



Hình 14. Giao diện Quản lý nhân sự



Hình 15. Giao diện Quản lý vào ra

# **Chương 5: Kết luận**

Hệ thống quản lý vào ra bằng sinh trắc học sử dụng công nghệ nhận diện khuôn mặt và vân tay không chỉ đáp ứng được các yêu cầu về độ chính xác và bảo mật cao, mà còn mở ra nhiều ứng dụng tiềm năng trong các tổ chức, doanh nghiệp. Việc sử dụng các thiết bị phần cứng như module ESP32, camera OV2640, cảm biến vân tay AS608, và màn hình LCD 16x2 với I2C giúp hệ thống hoạt động ổn định và hiệu quả, đáp ứng các yêu cầu về thời gian thực và tính dễ sử dụng.

Phần mềm được phát triển với công nghệ Nest.js kết hợp Face-API giúp quá trình nhận diện khuôn mặt và vân tay diễn ra nhanh chóng, chính xác, trong khi giao diện Web thân thiện với người dùng cho phép quản lý nhân sự và theo dõi lịch sử ra vào một cách hiệu quả.

Với tính bảo mật cao, khả năng tích hợp linh hoạt và dễ sử dụng, hệ thống này hứa hẹn sẽ là giải pháp tối ưu cho việc quản lý ra vào và giám sát an ninh trong nhiều lĩnh vực, từ doanh nghiệp đến các khu công nghiệp, trường học hay các tòa nhà văn phòng.