

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG HÀ NỘI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG**  
**NHÀ THÔNG MINH CÓ TÍCH HỢP AI**

**NGUYỄN QUANG HUY**

**Hà Nội – Năm 2025**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG HÀ NỘI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG**  
**NHÀ THÔNG MINH CÓ TÍCH HỢP AI**

**Họ tên sinh viên: Nguyễn Quang Huy**  
**Ngành đào tạo: Công Nghệ Thông Tin**

**NGƯỜI HƯỚNG DẪN: ThS. TRƯƠNG MẠNH ĐẠT**

**Hà Nội – Năm 2025**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**BẢN CAM ĐOAN**

Tên tôi là: Nguyễn Quang Huy

Mã sinh viên: 21111064617

Lớp ĐH11C13

Ngành: Công Nghệ Thông Tin

Tôi đã thực hiện khóa luận tốt nghiệp với đề tài: Nghiên cứu và xây dựng hệ thống nhà thông minh có tích hợp AI.

Tôi xin cam đoan đây là đề tài nghiên cứu của riêng tôi và được sự hướng dẫn của Ths. Trương Mạnh Đạt. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa được công bố dưới bất kỳ hình thức nào. Nếu phát hiện có bất kỳ hình thức gian lận nào tôi xin hoàn toàn chịu trách nhiệm trước pháp luật.

*Hà Nội, ngày 13 tháng 04 năm 2025*

**Giáo viên hướng dẫn**

**Sinh viên**

**Ths. Trương Mạnh Đạt**

**Nguyễn Quang Huy**

## LỜI CẢM ƠN

Trong quá trình nghiên cứu đề tài “Nghiên cứu và xây dựng hệ thống nhà thông minh có tích hợp AI”, em xin cảm ơn tới các thầy cô giảng viên khoa Công nghệ thông tin trường Đại học Tài nguyên và Môi trường Hà Nội đã luôn hướng dẫn em để có thể hoàn thành bài khóa luận một cách tốt nhất.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc tới Ths. Trương Mạnh Đạt, người đã hướng dẫn trực tiếp cho em trong khóa luận lần này. Nhờ có sự hướng dẫn chi tiết, tận tình của thầy mà em đã khắc phục được nhiều sai sót trong khóa luận lần này.

Tuy nhiên với những hạn chế về kiến thức và thời gian nên không thể tránh khỏi sai sót, em rất mong nhận được những nhận xét góp ý chỉ bảo của thầy cô để chương trình được hoàn thiện hơn, có tính thực tiễn để có thể áp dụng trong tương lai.

Em xin chân thành cảm ơn!

# MỤC LỤC

<b>BẢN CAM ĐOAN</b> .....	i
<b>LỜI CẢM ƠN</b> .....	ii
<b>DANH MỤC TỪ VIẾT TẮT</b> .....	vii
<b>DANH MỤC BẢNG BIỂU</b> .....	ix
<b>DANH MỤC HÌNH ẢNH</b> .....	x
<b>MỞ ĐẦU</b> .....	<b>1</b>
1. Giới thiệu đơn vị thực tập .....	1
2. Lý do chọn đề tài .....	1
3. Mục tiêu nghiên cứu .....	3
4. Phạm vi và đối tượng nghiên cứu .....	3
4.1 Phạm vi nghiên cứu .....	3
4.2 Đối tượng nghiên cứu .....	4
5. Phương pháp nghiên cứu .....	4
6. Dự kiến kết quả nghiên cứu đạt được .....	5
7. Ý nghĩa khoa học và ý nghĩa thực tiễn của nghiên cứu.....	7
8. Bố cục của báo cáo .....	8
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT</b> .....	<b>9</b>
1.1. Internet of things (IoT) .....	9
1.1.1. Nguyên lý hoạt động của IoT .....	9
1.1.2. Ứng dụng của IoT trong thực tế.....	10
1.1.3. Lợi ích và thách thức của IoT .....	10
1.1.4. So sánh IoT với các công nghệ khác trong nhà thông minh.....	10
1.1.5. Một số hạn chế của IoT khi ứng dụng trong nhà thông minh .....	10
1.1.6. Tương lai của IoT .....	11
1.2. Trí thông minh nhân tạo (AI).....	12
1.2.1. AI là gì? .....	12
1.2.2. Các loại AI .....	12
1.2.3. Ứng dụng của AI trong thực tế .....	12
1.2.4. Lợi ích và thách thức của AI.....	13

1.3. Giới thiệu về nhà thông minh (Smart Home) .....	13
1.3.1. Khái niệm nhà thông minh .....	13
1.3.2. Lịch sử phát triển của nhà thông minh .....	14
1.3.3. Cấu trúc của hệ thống nhà thông minh .....	14
1.3.4. Các tính năng chính của nhà thông minh.....	14
1.3.5. Xu hướng phát triển của nhà thông minh .....	15
1.3.6. Kết luận.....	15
1.4. Các giao thức và công nghệ kết nối trong nhà thông minh .....	15
1.4.1. Giao thức kết nối không dây.....	15
1.4.2. Giao thức kết nối có dây .....	17
1.4.3. Kết luận.....	18
1.5. Giới thiệu về mô hình Client – Server .....	18
1.5.1. Giới thiệu về mô hình Client – Server.....	18
1.5.2. Cấu trúc của mô hình Client – Server.....	19
1.5.3. Cách thức hoạt động của mô hình Client – Server .....	20
1.5.4. Các giao thức truyền thông trong mô hình Client – Server.....	20
1.5.5. Ưu điểm và nhược điểm của mô hình Client – Server .....	20
1.6. Nền tảng và công nghệ áp dụng phía phần mềm .....	21
1.6.1. Nền tảng Android.....	21
1.6.2. Ngôn ngữ lập trình Java.....	22
1.6.3. Thư viện Retrofit .....	23
1.6.4. Thư viện OkHttp .....	24
1.6.5. NodeJS .....	24
1.6.6. WebSocket .....	25
1.6.7. MongoDB .....	26
1.6.8. Google Speech Recognition .....	27
1.6.9. OpenAI.....	27
1.6.10. Blender.....	28
1.6.11. Python .....	28
1.6.12. MTCNN .....	29
1.7. Phần mềm Arduino IDE và các thiết bị phần cứng .....	29
1.7.1. Giới thiệu về phần mềm Arduino IDE.....	29

1.7.2. Module WiFi ESP32.....	31
1.7.3. ESP32 CAM AI THINKER.....	32
1.7.4. Cảm biến nhiệt độ, độ ẩm DHT11 .....	33
1.7.5. Cảm biến khí gas MQ2 .....	34
1.7.6. Màn hình LCD .....	35
1.7.7. Còi 5V .....	36
1.7.8. Led xanh 3v .....	37
1.7.9. Breadboard.....	37
1.7.10. Dây cắm breadboard .....	38
1.7.11. Quạt tản nhiệt 5v.....	39
1.7.12. Servo SG90 .....	40
<b>CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....</b>	<b>42</b>
2.1. Đánh giá các hệ thống liên quan và hướng phát triển của đề tài .....	42
2.1.1. Các dự án tiêu biểu trong và ngoài nước .....	42
2.1.2. Điểm khác biệt và ưu thế của dự án.....	42
2.2. Thiết kế phần mềm .....	43
2.2.1. Mô tả các chức năng và tác nhân của hệ thống .....	43
2.2.2. Sơ đồ usecase .....	45
2.2.3. Sơ đồ usecase phân rã.....	48
2.2.4. Sơ đồ lớp.....	51
2.2.5. Sơ đồ hoạt động .....	52
2.1.6. Thiết kế Database.....	55
2.1.7. Bảo mật hệ thống .....	59
2.1.7. Giao tiếp ứng dụng di động với server .....	60
2.3. Phần cứng .....	60
2.3.1. Sơ đồ lắp ráp mạch .....	60
2.3.2. Các thư viện được dùng .....	61
<b>CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ THỬ NGHIỆM</b>	<b>63</b>
3.1. Yêu cầu môi trường triển khai và các bước triển khai.....	63
3.1.1. Yêu cầu môi trường triển khai .....	63
3.1.2. Các bước triển khai .....	64

3.2. Giao diện và lập trình chức năng phần mềm .....	65
3.2.1. Đăng nhập .....	65
3.2.2. Giao diện trang chủ.....	66
3.2.3. Giao diện quản lý phòng .....	67
3.2.4. Giao diện quản lý thiết bị.....	68
3.2.5. Quản lý AccessKey.....	69
3.2.6. Giao diện quản lý tài khoản .....	70
3.2.7. Giao diện cài đặt hệ thống .....	71
3.2.8. Giao diện quản lý khuôn mặt.....	72
3.2.9. Giao diện trợ lý ảo .....	73
3.2.10. Giao diện quản lý camera .....	75
3.2.11. Giao diện thông báo.....	76
3.2.12. Giao diện biểu đồ thống kê lịch sử nhiệt độ, độ ẩm, khí gas.....	77
3.3. Mô hình phần cứng .....	78
3.4. Đánh giá thử nghiệm hệ thống.....	80
3.4.1. Mục tiêu thử nghiệm.....	80
3.4.2. Môi trường và thiết bị thử nghiệm.....	80
3.4.3. Phương pháp thử nghiệm và kết quả .....	81
3.4.4. Kết luận.....	83
<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>84</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>85</b>



## DANH MỤC TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Từ đầy đủ</b>	<b>Nghĩa</b>
AI	Artificial Intelligence	Trí tuệ nhân tạo - Công nghệ mô phỏng trí thông minh của con người, dùng để tự động hóa, phân tích dữ liệu và xử lý ngôn ngữ.
Android SDK	Android Software Development Kit	Bộ công cụ phát triển phần mềm Android - Tập hợp các công cụ, thư viện và tài liệu để phát triển ứng dụng trên hệ điều hành Android.
AR	Augmented Reality	Thực tế tăng cường - Công nghệ phủ lớp thông tin số (hình ảnh, âm thanh) lên thế giới thực, thường dùng trong giáo dục, giải trí và giao diện người dùng.
iOS	iPhone Operating System	Hệ điều hành iPhone - Hệ điều hành di động do Apple phát triển, được sử dụng trên các thiết bị iPhone, iPad, nổi bật với tính ổn định và bảo mật.
IoT	Internet of Things	Internet vạn vật - Hệ thống các thiết bị kết nối qua internet, cho phép thu thập, chia sẻ và điều khiển từ xa.
MTCNN	Multi-task Cascaded Convolutional Networks	Mạng nơ-ron tích chập đa tác vụ - Thuật toán học sâu dùng để phát hiện và căn chỉnh khuôn mặt trong hình ảnh hoặc video với độ chính xác cao.
OpenAI	Open Artificial Intelligence	Trí tuệ nhân tạo mở - Tổ chức và nền tảng nghiên cứu AI, cung cấp các mô hình học sâu để xử lý ngôn ngữ tự nhiên và các tác vụ thông minh khác.

3D	Three-Dimensional	Ba chiều - Công nghệ biểu diễn hoặc mô phỏng đối tượng trong không gian ba chiều (dài, rộng, cao), thường dùng trong thiết kế và giao diện trực quan.
IDE	Integrated Development Environment	Là môi trường phát triển tích hợp, cung cấp các công cụ như trình soạn thảo mã, trình biên dịch (hoặc thông dịch), gỡ lỗi và các tính năng hỗ trợ lập trình khác trong một giao diện duy nhất.
LCD	Liquid Crystal Display	Là loại màn hình sử dụng tinh thể lỏng để hiển thị hình ảnh.
JVM	Java Virtual Machine	Máy ảo Java, giúp chạy mã Java trên nhiều hệ điều hành bằng cách chuyển đổi bytecode thành mã máy.
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản, dùng để giao tiếp giữa trình duyệt và máy chủ web.
HTTPS	HyperText Transfer Protocol Secure	Phiên bản bảo mật của HTTP, sử dụng SSL/TLS để mã hóa dữ liệu, giúp bảo vệ thông tin truyền tải.
APK	Android Package	Định dạng file ứng dụng dành cho hệ điều hành Android.
SSL	Secure Sockets Layer	Giao thức bảo mật mã hóa dữ liệu truyền trên internet.
AWS	Amazon Web Services	Nền tảng điện toán đám mây cung cấp dịch vụ hạ tầng IT.

## DANH MỤC BẢNG BIỂU

Bảng 2.1. Các yêu cầu chức năng của hệ thống.....	43
Bảng 2.2. Biểu đồ Usecase chức năng đăng nhập.....	49
Bảng 2.3. Biểu đồ Usecase chức năng thêm phòng .....	50
Bảng 2.4. Biểu đồ Usecase chức năng thêm thiết bị.....	51
Bảng 2.5. Bảng Users.....	55
Bảng 2.6. Bảng Rooms.....	56
Bảng 2.7. Bảng Devices .....	56
Bảng 2.8. Bảng Messages .....	57
Bảng 2.9. Bảng Notifications .....	57
Bảng 2.10. Bảng Histories .....	58
Bảng 2.11. Bảng Face_Embeddings .....	58
Bảng 2.12. Bảng Thresholds .....	59
Bảng 2.13. Bảng thư viện dùng cho phần cứng .....	61
Bảng 3.1. Bảng đo độ chính xác của cảm biến .....	81
Bảng 3.2. Bảng đo thời gian hiệu quả cảnh báo.....	82
Bảng 3.3. Bảng đo hiệu suất nhận diện khuôn mặt.....	83
Bảng 3.4. Bảng đo hiệu suất truyền video từ ESP32 - CAM.....	83

## DANH MỤC HÌNH ẢNH

Hình 1.1. Internet of things .....	9
Hình 1.2. Trí thông minh nhân tạo AI.....	12
Hình 1.3. SmartHome .....	13
Hình 1.4. Mô hình Client – Server.....	18
Hình 1.5. Nền tảng Android.....	21
Hình 1.6. Ngôn ngữ lập trình Java .....	22
Hình 1.7. Thư viện Retrofit2.....	23
Hình 1.8. Thư viện OkHttp .....	24
Hình 1.9. NodeJS .....	25
Hình 1.10. WebSocket .....	25
Hình 1.11. MongoDB.....	26
Hình 1.12. Google Speech Recognition.....	27
Hình 1.13. OpenAI.....	27
Hình 1.14. Blender .....	28
Hình 1.15. Python .....	28
Hình 1.16. MTCNN .....	29
Hình 1.17. Phần mềm Arduino IDE.....	29
Hình 1.18. Module WiFi ESP32 .....	31
Hình 1.19. ESP32 CAM AI THINKER.....	32
Hình 1.20. Cảm biến nhiệt độ, độ ẩm DHT11 .....	33
Hình 1.21. Cảm biến khí gas MQ2 .....	34
Hình 1.22. Màn hình LCD 16x2 .....	35
Hình 1.23. Còi 5v .....	36
Hình 1.24. Led xanh 3v.....	37
Hình 1.25. Breadboard .....	37
Hình 1.26. Dây cắm breadboard .....	38
Hình 1.27. Quạt tản nhiệt 5v .....	39
Hình 1.28. Servo SG90 .....	40
Hình 2.1. Sơ đồ usecase .....	45
Hình 2.2. Biểu đồ phân rã usecase đăng nhập .....	48

Hình 2.3. Biểu đồ phân rã usecase quản lý phòng .....	49
Hình 2.4. Biểu đồ phân rã usecase quản lý thiết bị .....	50
Hình 2.5. Sơ đồ lớp .....	51
Hình 2.6. Biểu đồ hoạt động chức năng đăng nhập .....	52
Hình 2.7. Biểu đồ hoạt động chức năng thêm phòng.....	53
Hình 2.8. Biểu đồ hoạt động chức năng thêm thiết bị .....	54
Hình 2.9. Sơ đồ ERD .....	55
Hình 2.11. Sơ đồ lắp ráp mạch.....	60
Hình 3.1. Giao diện đăng nhập.....	65
Hình 3.2. Giao diện trang chủ .....	66
Hình 3.3. Danh sách phòng .....	67
Hình 3.4. Thêm thông tin phòng .....	67
Hình 3.5. Cập nhật và xoá thông tin phòng .....	67
Hình 3.6. Danh sách thiết bị.....	68
Hình 3.7. Thêm thông tin thiết bị.....	68
Hình 3.8. Cập nhật và xoá thiết bị.....	68
Hình 3.9. Giao diện quản lý AccessKey .....	69
Hình 3.10. Giao diện quản lý tài khoản .....	70
Hình 3.11. Giao diện cài đặt hệ thống.....	71
Hình 3.12. Giao diện danh sách khuôn mặt .....	72
Hình 3.13. Giao diện thêm khuôn mặt .....	72
Hình 3.14. Giao diện trò chuyện văn bản .....	73
Hình 3.15. Giao diện trợ lý ảo 3D.....	73
Hình 3.16. Giao diện danh sách camera.....	75
Hình 3.17. Giao diện thêm camera.....	75
Hình 3.18. Giao diện thông báo .....	76
Hình 3.19. Giao diện biểu đồ thống kê lịch sử nhiệt độ, độ ẩm, khí gas .....	77
Hình 3.20. Hình nhìn từ phía trên mô hình .....	78
Hình 3.21. Camera giám sát bên ngoài ngôi nhà .....	78

# MỞ ĐẦU

## 1. Giới thiệu đơn vị thực tập

Công Ty TNHH Izisolution là một doanh nghiệp hoạt động trong lĩnh vực tư vấn và triển khai giải pháp quản trị doanh nghiệp tại Việt Nam. Công ty chuyên cung cấp các dịch vụ và giải pháp nhằm tối ưu hóa quy trình kinh doanh, nâng cao hiệu suất và hiệu quả hoạt động cho các doanh nghiệp.

**Lĩnh vực hoạt động chính của Izisolution bao gồm:**

- **Tư vấn giải pháp quản trị doanh nghiệp:** Izisolution hỗ trợ doanh nghiệp trong việc lựa chọn và triển khai các giải pháp quản trị phù hợp, giúp cải thiện quy trình và tăng cường khả năng cạnh tranh.
- **Triển khai hệ thống ERP:** Công ty cung cấp dịch vụ triển khai hệ thống hoạch định nguồn lực doanh nghiệp (ERP), giúp doanh nghiệp quản lý hiệu quả các nguồn lực và hoạt động kinh doanh.
- **Phát triển phần mềm tùy chỉnh:** Izisolution cung cấp dịch vụ phát triển các giải pháp phần mềm được tùy chỉnh theo nhu cầu cụ thể của từng doanh nghiệp, đảm bảo đáp ứng tối đa yêu cầu và mục tiêu kinh doanh.
- **Đào tạo và hỗ trợ kỹ thuật:** Công ty cung cấp các chương trình đào tạo và dịch vụ hỗ trợ kỹ thuật, giúp doanh nghiệp và nhân viên nắm bắt và sử dụng hiệu quả các giải pháp công nghệ.

Với đội ngũ chuyên gia giàu kinh nghiệm và am hiểu sâu sắc về các giải pháp quản trị doanh nghiệp, Izisolution đã và đang là đối tác tin cậy của nhiều doanh nghiệp trong việc cải thiện hiệu suất và đạt được mục tiêu kinh doanh.

## 2. Lý do chọn đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0, nhà thông minh đang trở thành xu hướng tất yếu, kết hợp giữa Internet of Things (IoT) và Trí tuệ nhân tạo (AI) để nâng cao sự tiện nghi, an ninh và hiệu quả năng lượng. Việc nghiên cứu và phát triển một hệ thống nhà thông minh có tích hợp AI không chỉ đáp ứng nhu cầu thực tiễn mà còn mang lại nhiều lợi ích thiết thực:

- **Tăng cường an ninh và giám sát thông minh:** Với tình trạng an ninh ngày càng trở thành mối quan tâm lớn, việc tích hợp AI vào hệ thống nhà thông minh giúp nhận diện khuôn mặt, phát hiện người lạ và gửi cảnh báo thời gian thực.[5] Các thiết bị như ESP32-CAM kết hợp với WebSocket hỗ trợ truyền video liên tục, giúp người dùng theo dõi không gian sống từ xa và nâng cao tính an toàn cho gia đình.

- **Tối ưu hóa năng lượng và bảo vệ môi trường:** Trong bối cảnh biến đổi khí hậu và chi phí năng lượng gia tăng, hệ thống nhà thông minh có thể tự động điều chỉnh thiết bị điện dựa trên dữ liệu cảm biến và lệnh giọng nói. Điều này không chỉ giúp tiết kiệm điện năng mà còn giảm thiểu tác động tiêu cực đến môi trường, hướng đến lối sống bền vững.[3]

- **Cải thiện trải nghiệm người dùng với công nghệ AI:** Việc kết hợp Google Speech Recognition và OpenAI cho phép hệ thống hiểu và phản hồi các lệnh giọng nói một cách tự nhiên. Ngoài ra, trợ lý ảo 3D không chỉ mang lại giao diện trực quan mà còn giúp người dùng tương tác với thiết bị một cách thông minh hơn.

- **Khả năng mở rộng và ứng dụng rộng rãi:** Hệ thống nhà thông minh tích hợp AI không chỉ phù hợp với hộ gia đình mà còn có thể mở rộng để quản lý văn phòng, chung cư hoặc doanh nghiệp nhỏ. Nhờ sự kết hợp linh hoạt giữa IoT và AI, hệ thống có thể phát triển thêm nhiều tính năng nhằm đáp ứng các nhu cầu khác nhau.

- **Thách thức và cơ hội nghiên cứu:** Mặc dù công nghệ nhà thông minh đã có nhiều bước tiến quan trọng, nhưng vẫn tồn tại những thách thức về hiệu suất, bảo mật dữ liệu và khả năng cá nhân hóa. Việc nghiên cứu và phát triển một hệ thống tối ưu, kết hợp AI và IoT, không chỉ giúp khắc phục những hạn chế này mà còn mở ra tiềm năng mới trong lĩnh vực tự động hóa và trí tuệ nhân tạo.

Từ những lý do trên, đề tài "**Nghiên cứu và xây dựng hệ thống nhà thông minh có tích hợp AI**" được lựa chọn nhằm góp phần phát triển một giải pháp nhà thông minh tiên tiến, hiệu quả và an toàn hơn, đáp ứng nhu cầu ngày càng cao của xã hội hiện đại.

### 3. Mục tiêu nghiên cứu

Nghiên cứu ứng dụng IoT trong thiết kế hệ thống quản lý nhà thông minh với sự tích hợp của ESP32 và các công nghệ AI tiên tiến nhằm đạt được các mục tiêu chính sau:

- **Phát triển hệ thống quản lý nhà thông minh:**
  - Thiết kế kiến trúc hệ thống với ESP32, ESP32-CAM, WebSocket để điều khiển thiết bị và truyền dữ liệu thời gian thực.
  - Xây dựng mô hình phân cứng kết nối thiết bị qua IoT, hỗ trợ giám sát và điều khiển từ xa bằng giọng nói và ứng dụng di động.
- **Tăng cường an ninh:**
  - Ứng dụng nhận diện khuôn mặt để giám sát an ninh.
  - Cảnh báo thời gian thực qua ứng dụng di động hoặc trợ lý ảo 3D.
- **Thiết kế giao diện thân thiện:**
  - Phát triển ứng dụng di động và trợ lý ảo 3D, hỗ trợ điều khiển bằng giọng nói.

Các mục tiêu trên không chỉ thể hiện tính khả thi của nghiên cứu mà còn mang lại giá trị thực tiễn, từ việc nâng cao an ninh, tiết kiệm năng lượng, cải thiện tiện nghi, đến thúc đẩy sự phát triển bền vững. Việc hiện thực hóa hệ thống này, với sự kết hợp giữa ESP32-CAM, AI, WebSocket và trợ lý ảo 3D, sẽ tạo ra một giải pháp quản lý nhà thông minh hiệu quả, đáng tin cậy, đồng thời mở ra nhiều cơ hội ứng dụng và phát triển trong tương lai.

### 4. Phạm vi và đối tượng nghiên cứu

#### 4.1 Phạm vi nghiên cứu

- **Nghiên cứu mô hình hệ thống quản lý nhà thông minh:**
  - Đề tài tập trung vào thiết kế và phát triển một mô hình hệ thống quản lý nhà thông minh ứng dụng công nghệ IoT và AI, sử dụng ESP32 và ESP32-CAM làm trung tâm điều khiển. Mô hình bao gồm các cảm biến (nhiệt độ, chuyển động), thiết bị điều khiển (đèn, quạt, điều hòa), và camera ESP32-CAM để truyền video thời gian thực qua WebSocket, hỗ trợ nhận diện khuôn mặt bằng AI.



- Hệ thống được triển khai trong môi trường mô phỏng tại trường học, phục vụ mục đích thử nghiệm và học tập. Phạm vi nghiên cứu giới hạn ở việc xây dựng mô hình thử nghiệm, không bao gồm triển khai thương mại hoặc mở rộng quy mô ngoài môi trường giáo dục.
- **Mục tiêu cụ thể:**
  - Thiết kế giao diện người dùng đa dạng, bao gồm ứng dụng di động (dùng Java và Android SDK) và trợ lý ảo 3D, tích hợp Google Speech Recognition và OpenAI, cho phép điều khiển bằng giọng nói và giám sát trực quan qua video từ ESP32-CAM.
  - Thử nghiệm hệ thống trong các kịch bản thực tế (giám sát an ninh qua nhận diện khuôn mặt, tự động hóa thiết bị để tiết kiệm năng lượng), đánh giá hiệu quả và khả năng tương tác của người dùng với các tính năng thông minh.

#### **4.2 Đối tượng nghiên cứu**

- Mô hình hệ thống quản lý nhà thông minh được phát triển dựa trên các thiết bị và công nghệ hiện đại:
  - **Thiết bị và cảm biến:** Sử dụng ESP32 làm bộ điều khiển trung tâm, kết hợp ESP32-CAM để quay và truyền video, cùng các cảm biến như DHT11 (nhiệt độ, độ ẩm), và thiết bị điều khiển (đèn LED, quạt, công tắc).
  - **Tích hợp AI và IoT:** ESP32 kết nối qua Wi-Fi, hỗ trợ WebSocket để truyền dữ liệu thời gian thực (bao gồm video và lệnh), trong khi Google Speech Recognition và OpenAI xử lý lệnh giọng nói, và AI nhận diện khuôn mặt tăng cường an ninh.
- Người dùng có thể giám sát và điều khiển từ xa thông qua ứng dụng di động hoặc tương tác trực tiếp với trợ lý ảo 3D, dựa trên dữ liệu từ cảm biến và video ESP32-CAM.

#### **5. Phương pháp nghiên cứu**

- **Phương pháp khảo sát và phân tích:**
  - Tác giả tiến hành khảo sát nhu cầu người dùng (qua bảng hỏi hoặc phỏng vấn) để xác định các tính năng mong muốn, như điều khiển bằng giọng nói, giám sát video, hoặc tự động hóa an ninh. Kết quả phân tích sẽ định hình yêu

cầu kỹ thuật, ví dụ: tích hợp ESP32-CAM cho video hay OpenAI cho tự động hóa.

- Tham khảo tài liệu khoa học, bài báo về IoT và AI (như nghiên cứu về WebSocket, nhận diện khuôn mặt), để hiểu xu hướng công nghệ và áp dụng vào hệ thống.

- **Phương pháp thiết kế và triển khai:**

- Thiết kế mô hình phần cứng mô phỏng ngôi nhà thông minh với ESP32, ESP32-CAM, cảm biến và thiết bị điều khiển. Lập trình bằng C++ cho ESP32/ESP32-CAM (sử dụng thư viện WebSocket, camera), Java cho ứng dụng di động (Android SDK, Retrofit), và tích hợp Google Speech Recognition cùng OpenAI cho trợ lý ảo 3D.
- Triển khai hệ thống trong môi trường mô phỏng tại trường học, kết nối qua Wi-Fi để thử nghiệm giám sát video, điều khiển giọng nói, và tự động hóa thiết bị.

- **Phương pháp kiểm tra và đánh giá:**

- Thực hiện các thử nghiệm với kịch bản thực tế: phát hiện chuyển động qua ESP32-CAM và nhận diện khuôn mặt, tự động bật/tắt quạt dựa trên nhiệt độ, gửi cảnh báo qua WebSocket và trợ lý ảo 3D.
- Thu thập phản hồi từ sinh viên và người dùng qua bảng hỏi hoặc phỏng vấn sau khi trải nghiệm, đánh giá tính hiệu quả (an ninh, tiết kiệm năng lượng, tiện nghi), và điều chỉnh hệ thống để tối ưu hóa hiệu suất dựa trên kết quả.

## **6. Dự kiến kết quả nghiên cứu đạt được**

- **Hệ thống quản lý nhà thông minh hoàn chỉnh:** Một mô hình nhà thông minh toàn diện sẽ được hoàn thiện, sử dụng ESP32 và ESP32-CAM làm trung tâm điều khiển, kết nối các cảm biến (nhiệt độ, chuyển động) và thiết bị (đèn, quạt, điều hòa). Hệ thống hỗ trợ truyền video thời gian thực qua WebSocket từ ESP32-CAM và điều khiển từ xa qua ứng dụng di động hoặc giọng nói với Google Speech Recognition, mang lại khả năng giám sát và quản lý hiệu quả.

- **Giao diện ứng dụng và trợ lý ảo thân thiện:** Ứng dụng di động (phát triển bằng Java và Android SDK) sẽ có giao diện trực quan, hiển thị dữ liệu từ cảm biến, video từ ESP32-CAM, và điều khiển thiết bị. Đồng thời, mô hình 3D trợ lý ảo, tích hợp Google Speech Recognition và OpenAI, cung cấp trải nghiệm tương tác sinh động qua giọng nói, nâng cao sự tiện lợi và hấp dẫn cho người dùng.
- **Tính năng tự động hóa thông minh:** Hệ thống sẽ tự động điều chỉnh thiết bị dựa trên dữ liệu cảm biến và phân tích từ OpenAI. Ví dụ, khi nhiệt độ tăng cao, quạt tự động bật; khi ESP32-CAM phát hiện người lạ qua nhận diện khuôn mặt, hệ thống kích hoạt cảnh báo qua WebSocket. Các kịch bản này đảm bảo an ninh và tối ưu hóa năng lượng một cách thông minh.
- **Tiết kiệm năng lượng:** Thông qua giám sát thời gian thực và tự động hóa, mô hình dự kiến giảm tiêu thụ điện năng bằng cách tắt thiết bị không cần thiết hoặc điều chỉnh hoạt động dựa trên nhu cầu, với dữ liệu phân tích được hiển thị qua ứng dụng di động hoặc báo cáo từ trợ lý ảo 3D, giúp người dùng quản lý năng lượng hiệu quả.
- **Nâng cao an ninh:** Hệ thống tích hợp nhận diện khuôn mặt qua ESP32-CAM, phát hiện sự cố (như người lạ xâm nhập) và gửi thông báo tức thì qua WebSocket đến ứng dụng di động hoặc thông qua giọng nói từ trợ lý ảo 3D. Điều này tăng cường khả năng bảo vệ an toàn cho gia đình và tài sản.
- **Phản hồi tích cực từ người dùng:** Dự kiến sinh viên và người thử nghiệm sẽ đánh giá cao tính dễ sử dụng, hiệu quả quản lý và khả năng ứng dụng thực tế của hệ thống. Phản hồi từ họ, thu thập qua trải nghiệm thực tế tại môi trường mô phỏng, sẽ cung cấp dữ liệu quý giá để cải tiến hệ thống trong tương lai.
- **Tiềm năng phát triển và mở rộng:** Kết quả nghiên cứu sẽ đặt nền tảng cho các cải tiến tiếp theo, như tích hợp sâu hơn với AI (OpenAI) để nâng cao tự động hóa, hoặc mở rộng mô hình từ hộ gia đình sang quản lý chung cư/doanh nghiệp nhỏ nhờ tính linh hoạt của ESP32, ESP32-CAM, và WebSocket. Mô hình phần cứng mô phỏng cũng mở ra cơ hội nghiên cứu thêm về giám sát video và tương tác thông minh.

## 7. Ý nghĩa khoa học và ý nghĩa thực tiễn của nghiên cứu

### Ý nghĩa khoa học

- **Góp phần vào phát triển lĩnh vực IoT và AI:** Đề tài góp phần làm phong phú thêm lĩnh vực nghiên cứu nhà thông minh bằng cách tích hợp công nghệ IoT (ESP32, ESP32-CAM) và trí tuệ nhân tạo (OpenAI, Google Speech Recognition, nhận diện khuôn mặt). Đặc biệt, việc sử dụng WebSocket để truyền video thời gian thực giúp mở rộng khả năng tương tác giữa người và thiết bị trong môi trường sống hiện đại.
- **Tài liệu tham khảo cho nghiên cứu và giảng dạy:** Với quy trình nghiên cứu rõ ràng, từ khảo sát nhu cầu thực tế đến thiết kế hệ thống và thử nghiệm, đề tài trở thành một tài liệu thực hành giá trị cho sinh viên, nhà nghiên cứu và giảng viên trong việc giảng dạy và triển khai các ứng dụng IoT/AI.
- **Xây dựng mô hình nghiên cứu toàn diện:** Đề tài không chỉ tập trung vào giải pháp kỹ thuật, mà còn lồng ghép yếu tố trải nghiệm người dùng và tính bảo mật. Sự kết hợp hài hòa giữa phần cứng (cảm biến, vi điều khiển), phần mềm (lập trình C++, Java, Android SDK) và AI mang đến một mô hình nghiên cứu đa chiều, dễ áp dụng và phát triển trong tương lai.
- **Khuyến khích tư duy liên ngành:** Việc vận dụng nhiều lĩnh vực như kỹ thuật điện tử, khoa học máy tính, AI và giao diện người dùng giúp người học và người nghiên cứu tiếp cận được phương pháp tích hợp công nghệ theo hướng toàn diện, mở ra cơ hội cho các sáng kiến đổi mới sáng tạo trong thời kỳ công nghiệp 4.0.

### Ý nghĩa thực tiễn

- **Cải thiện chất lượng cuộc sống hàng ngày:** Hệ thống giúp người dùng dễ dàng điều khiển các thiết bị điện trong nhà thông qua ứng dụng di động hoặc giọng nói, mang lại sự linh hoạt và thuận tiện trong việc quản lý các thiết bị như đèn, quạt, điều hòa, và các thiết bị điện khác. Trợ lý ảo 3D hỗ trợ giao tiếp tự nhiên, dễ dàng tương tác với người dùng, đặc biệt phù hợp cho người lớn tuổi hoặc những người ít tiếp xúc với công nghệ. Điều này góp phần mang lại sự tiện nghi, giảm thiểu công sức và thời gian, cải thiện chất lượng cuộc sống hàng ngày.

- **Tăng cường an ninh và giám sát thông minh:** Tính năng nhận diện khuôn mặt từ ESP32-CAM không chỉ giúp xác định người lạ mà còn có khả năng phát hiện chuyển động bất thường trong khu vực quan sát, gửi cảnh báo ngay lập tức qua ứng dụng điện thoại. Điều này giúp người dùng nhanh chóng nhận diện các nguy cơ tiềm ẩn về an ninh, đặc biệt hữu ích khi không có mặt tại nhà.
- **Thúc đẩy ứng dụng công nghệ trong cộng đồng:** Mô hình hệ thống được thiết kế đơn giản và chi phí hợp lý, dễ triển khai tại các hộ gia đình, văn phòng nhỏ hoặc các không gian sống và làm việc khác. Với khả năng tích hợp dễ dàng và sử dụng trực quan, hệ thống này giúp người dùng nhanh chóng làm quen và áp dụng công nghệ vào đời sống, nâng cao nhận thức cộng đồng về các tiện ích mà công nghệ thông minh mang lại.
- **Tiềm năng ứng dụng rộng rãi:** Hệ thống có thể dễ dàng mở rộng và ứng dụng vào các lĩnh vực khác như giám sát văn phòng, phòng trọ, các tòa nhà nhỏ hoặc khu dân cư cần đảm bảo an ninh cơ bản. Với thiết kế linh hoạt về phần mềm và phần cứng, hệ thống có thể được tùy chỉnh để phù hợp với các nhu cầu cụ thể, từ việc quản lý thiết bị điện trong các hộ gia đình đến các ứng dụng giám sát an ninh trong các khu vực cộng đồng hoặc tòa nhà văn phòng.

## **8. Bố cục của báo cáo**

**Chương 1:** Cơ sở lý thuyết

**Chương 2:** Phân tích thiết kế hệ thống

**Chương 3:** Triển khai và đánh giá thử nghiệm hệ thống

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1. Internet of things (IoT)



Hình 1.1. Internet of things

**IoT (Internet of Things)** là một hệ thống các thiết bị được kết nối với nhau thông qua mạng Internet, cho phép chúng thu thập, trao đổi và xử lý dữ liệu một cách tự động mà không cần sự can thiệp trực tiếp của con người. IoT giúp tạo ra một môi trường thông minh, nơi các thiết bị có thể giao tiếp và thực hiện các tác vụ một cách hiệu quả.

#### 1.1.1. Nguyên lý hoạt động của IoT

IoT hoạt động dựa trên bốn thành phần chính:

- **Thiết bị cảm biến:** Các thiết bị này thu thập dữ liệu từ môi trường xung quanh (như nhiệt độ, độ ẩm, ánh sáng, chuyển động, v.v.).
- **Kết nối mạng:** Dữ liệu từ cảm biến được truyền qua các giao thức kết nối như Wi-Fi, Bluetooth, Zigbee, LoRa, 4G/5G.
- **Xử lý dữ liệu:** Dữ liệu được gửi đến máy chủ, điện toán đám mây hoặc các hệ thống xử lý trung tâm để phân tích và đưa ra quyết định.
- **Tác động hoặc phản hồi:** Sau khi dữ liệu được xử lý, hệ thống có thể tự động điều khiển thiết bị hoặc gửi cảnh báo cho người dùng.

### ***1.1.2. Ứng dụng của IoT trong thực tế***

IoT đang được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

- **Nhà thông minh (Smart Home):** Hệ thống đèn, điều hòa, khóa cửa, camera giám sát có thể được điều khiển từ xa thông qua ứng dụng di động.
- **Y tế thông minh (Smart Healthcare):** Các thiết bị đeo tay theo dõi sức khỏe, cảm biến đo nhịp tim, huyết áp giúp bác sĩ theo dõi bệnh nhân từ xa.
- **Giao thông thông minh (Smart Transportation):** Xe tự lái, hệ thống giám sát giao thông và bãi đỗ xe thông minh giúp giảm ùn tắc và tai nạn.[1]
- **Nông nghiệp thông minh (Smart Agriculture):** Hệ thống cảm biến giúp theo dõi độ ẩm đất, nhiệt độ, lượng nước tưới để tối ưu hóa sản xuất nông nghiệp.
- **Công nghiệp 4.0:** Các nhà máy tự động hóa sử dụng IoT để quản lý dây chuyền sản xuất, bảo trì thiết bị và giám sát hoạt động từ xa.

### ***1.1.3. Lợi ích và thách thức của IoT***

#### **Lợi ích:**

- Tăng cường tự động hóa, tối ưu vận hành, giảm chi phí.
- Cải thiện trải nghiệm người dùng, thu thập và phân tích dữ liệu theo thời gian thực.

#### **Thách thức:**

- Rủi ro bảo mật, chi phí đầu tư cao.
- Độ tin cậy kết nối mạng và khả năng mở rộng hệ thống.

### ***1.1.4. So sánh IoT với các công nghệ khác trong nhà thông minh***

- **IoT vs. PLC:** IoT linh hoạt, kết nối Internet, dễ mở rộng; PLC bền bỉ, không cần mạng, phù hợp công nghiệp.
- **IoT vs. Zigbee/Z-Wave:** IoT đa dạng giao thức, tích hợp đám mây; Zigbee/Z-Wave tiết kiệm năng lượng, ổn định nhưng giới hạn kết nối.
- **IoT vs. AI độc lập:** IoT + AI tận dụng dữ liệu lớn, học hỏi liên tục; AI độc lập bảo mật tốt hơn nhưng xử lý hạn chế.

### ***1.1.5. Một số hạn chế của IoT khi ứng dụng trong nhà thông minh***

Mặc dù IoT mang lại nhiều lợi ích cho nhà thông minh, nó cũng đối mặt với một số hạn chế cụ thể:

- **Độ trễ khi truyền dữ liệu:**
  - Trong hệ thống IoT, dữ liệu thường được gửi qua mạng Internet đến máy chủ đám mây để xử lý trước khi phản hồi lại thiết bị. Điều này có thể gây ra độ trễ, đặc biệt khi mạng không ổn định hoặc quá tải. Ví dụ, bật đèn thông minh qua ứng dụng có thể mất vài giây nếu kết nối Wi-Fi yếu, làm giảm trải nghiệm người dùng so với công tắc truyền thống.
- **Chi phí đầu tư ban đầu cao:**
  - Việc triển khai IoT trong nhà thông minh đòi hỏi mua sắm các thiết bị tương thích (đèn thông minh, cảm biến, khóa cửa, v.v.), cùng với việc cài đặt hệ thống mạng mạnh và ổn định. Chi phí này có thể là rào cản lớn đối với người dùng phổ thông, đặc biệt khi so sánh với các giải pháp tự động hóa đơn giản hơn như công tắc hẹn giờ.
- **Phụ thuộc vào kết nối mạng:**
  - IoT trong nhà thông minh mất hiệu quả nếu không có Internet hoặc khi mạng bị gián đoạn. Ví dụ, nếu Wi-Fi bị ngắt, người dùng không thể điều khiển đèn hoặc xem camera từ xa, làm giảm tính tiện lợi.
- **Vấn đề tương thích giữa các thiết bị:**
  - Không phải tất cả các thiết bị IoT đều hoạt động tốt với nhau do sử dụng giao thức hoặc nền tảng khác nhau (ví dụ: thiết bị của Amazon Alexa không tương thích hoàn toàn với Google Home). Điều này buộc người dùng phải mua thiết bị từ cùng một hệ sinh thái, hạn chế sự lựa chọn và tăng chi phí.
- **Tiêu thụ năng lượng:**
  - Các thiết bị IoT thường xuyên kết nối mạng có thể tiêu tốn nhiều điện năng hơn so với thiết bị truyền thống hoặc thiết bị dùng giao thức tiết kiệm năng lượng như Zigbee. Điều này dẫn đến hóa đơn tiền điện cao hơn trong dài hạn.

#### ***1.1.6. Tương lai của IoT***

Với sự phát triển của công nghệ 5G, AI (trí tuệ nhân tạo) và điện toán biên (Edge Computing), IoT sẽ ngày càng trở nên mạnh mẽ hơn. Trong tương lai, IoT sẽ tiếp tục đóng vai trò quan trọng trong việc xây dựng các thành phố thông minh.



## 1.2. Trí thông minh nhân tạo (AI)

### 1.2.1. AI là gì?



Hình 1.2. Trí thông minh nhân tạo AI

Trí thông minh nhân tạo (AI - Artificial Intelligence) là một lĩnh vực trong khoa học máy tính, tập trung vào việc phát triển các hệ thống có khả năng mô phỏng trí tuệ con người. AI giúp máy móc có thể học hỏi, suy luận, nhận thức và thực hiện các nhiệm vụ mà trước đây chỉ con người mới có thể làm được.

### 1.2.2. Các loại AI

AI có thể được chia thành ba cấp độ chính:

- **AI yếu (Narrow AI):** Đây là dạng AI phổ biến nhất hiện nay, được thiết kế để thực hiện một nhiệm vụ cụ thể, như trợ lý ảo (Siri, Google Assistant), hệ thống nhận diện khuôn mặt, hoặc AI trong xe tự lái.
- **AI mạnh (General AI):** Một hệ thống AI có thể thực hiện nhiều nhiệm vụ khác nhau với trí tuệ tương đương con người. Hiện nay, AI mạnh vẫn đang trong quá trình nghiên cứu và phát triển.
- **AI siêu việt (Super AI):** Đây là giai đoạn mà AI có thể vượt qua trí thông minh của con người, có khả năng tự suy nghĩ, sáng tạo và ra quyết định mà không cần sự can thiệp của con người.

### 1.2.3. Ứng dụng của AI trong thực tế

AI đang được ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm:

- **Y tế:** Chẩn đoán bệnh dựa trên hình ảnh y khoa, phân tích hồ sơ bệnh nhân, hỗ trợ bác sĩ trong việc điều trị.
- **Giáo dục:** Hệ thống giảng dạy thông minh, trợ lý học tập ảo.
- **Tài chính:** Phân tích dữ liệu tài chính, phát hiện gian lận, hỗ trợ giao dịch chứng

khoán.

- **Sản xuất:** Tự động hóa dây chuyền sản xuất, kiểm soát chất lượng sản phẩm.
- **Giao thông:** Xe tự lái, hệ thống quản lý giao thông thông minh.
- **Dịch vụ khách hàng:** Chatbot, tổng đài tự động giúp giải quyết yêu cầu của khách hàng nhanh chóng.

#### ***1.2.4. Lợi ích và thách thức của AI***

##### **Lợi ích:**

- Tăng hiệu suất và tự động hóa công việc.
- Giảm chi phí nhân lực và thời gian xử lý.
- Cải thiện độ chính xác trong nhiều lĩnh vực như y tế, tài chính.
- Hỗ trợ ra quyết định dựa trên dữ liệu lớn.

##### **Thách thức:**

- Rủi ro về bảo mật và quyền riêng tư.
- Tác động đến thị trường lao động, có thể thay thế một số công việc truyền thống.
- AI có thể bị thiên vị nếu dữ liệu huấn luyện không đa dạng và công bằng.
- Việc kiểm soát AI mạnh trong tương lai có thể đặt ra nhiều thách thức về đạo đức và pháp lý.

### **1.3. Giới thiệu về nhà thông minh (Smart Home)**

#### ***1.3.1. Khái niệm nhà thông minh***



*Hình 1.3. SmartHome*

Nhà thông minh (Smart Home) là một hệ thống nhà ở hiện đại được trang bị các thiết bị điện tử và công nghệ IoT (Internet of Things) nhằm tự động hóa và điều khiển từ xa các chức năng trong gia đình. Các thiết bị trong hệ thống có thể bao gồm đèn chiếu sáng, điều hòa nhiệt độ, khóa cửa, camera an ninh, rèm cửa, thiết bị

giải trí, và nhiều thiết bị khác. Người dùng có thể quản lý và điều khiển các thiết bị này thông qua điện thoại thông minh, máy tính bảng hoặc trợ lý ảo bằng giọng nói.

### ***1.3.2. Lịch sử phát triển của nhà thông minh***

Ý tưởng về nhà thông minh xuất hiện từ những năm 1970 khi các hệ thống điều khiển tự động hóa trong gia đình bắt đầu được nghiên cứu. Đến những năm 1990, sự phát triển của Internet đã mở ra tiềm năng kết nối các thiết bị gia dụng với nhau. Vào đầu thế kỷ 21, nhờ sự tiến bộ của công nghệ IoT, điện toán đám mây, trí tuệ nhân tạo (AI) và các giao thức kết nối không dây, nhà thông minh đã trở nên phổ biến và ngày càng phát triển mạnh mẽ.

### ***1.3.3. Cấu trúc của hệ thống nhà thông minh***

Một hệ thống nhà thông minh thường bao gồm các thành phần chính sau:

- **Thiết bị cảm biến:** Thu thập dữ liệu từ môi trường như nhiệt độ, độ ẩm, ánh sáng, chuyển động.
- **Bộ điều khiển trung tâm (Hub/Controller):** Là bộ não của hệ thống, giúp kết nối và điều khiển các thiết bị.
- **Thiết bị đầu ra:** Bao gồm đèn thông minh, điều hòa, khóa cửa, rèm cửa tự động, hệ thống âm thanh, v.v.
- **Hệ thống kết nối:** Sử dụng các giao thức như Wi-Fi, Zigbee, Z-Wave, Bluetooth để truyền tải dữ liệu giữa các thiết bị.
- **Phần mềm điều khiển:** Ứng dụng trên điện thoại hoặc trợ lý ảo như Google Assistant, Amazon Alexa giúp người dùng giám sát và điều khiển hệ thống.

### ***1.3.4. Các tính năng chính của nhà thông minh***

- **Tự động hóa:** Hệ thống có thể tự động bật/tắt thiết bị theo lịch trình hoặc cảm biến. Ví dụ, đèn tự động bật khi có người bước vào phòng.
- **Điều khiển từ xa:** Người dùng có thể điều khiển thiết bị thông qua ứng dụng di động hoặc giọng nói dù ở bất cứ đâu.
- **Tiết kiệm năng lượng:** Hệ thống giúp tối ưu hóa điện năng tiêu thụ bằng cách tự động điều chỉnh thiết bị khi không có người sử dụng.
- **An ninh và giám sát:** Camera thông minh, khóa cửa điện tử và hệ thống báo động giúp bảo vệ ngôi nhà khỏi các nguy cơ xâm nhập.

- **Tích hợp AI:** Các hệ thống nhà thông minh hiện đại có thể học thói quen của người dùng để tối ưu hóa trải nghiệm sống.

#### ***1.3.5. Xu hướng phát triển của nhà thông minh***

- **Kết nối 5G và IoT:** Với sự phát triển của mạng 5G, tốc độ truyền tải dữ liệu giữa các thiết bị sẽ nhanh hơn và ổn định hơn.
- **Trí tuệ nhân tạo (AI) và học máy (Machine Learning):** Các hệ thống nhà thông minh ngày càng trở nên thông minh hơn, có khả năng dự đoán và tự động điều chỉnh thiết bị dựa trên thói quen của người dùng.
- **Công nghệ điện toán biên (Edge Computing):** Giúp xử lý dữ liệu ngay tại thiết bị thay vì phải gửi lên đám mây, giúp tăng tốc độ phản hồi và bảo mật hơn.
- **Nhà thông minh bền vững:** Xu hướng phát triển các thiết bị sử dụng năng lượng tái tạo như năng lượng mặt trời, pin lưu trữ thông minh để giảm thiểu tác động đến môi trường.

#### ***1.3.6. Kết luận***

Nhà thông minh không chỉ mang lại sự tiện nghi mà còn giúp tối ưu hóa năng lượng, nâng cao mức độ an toàn và bảo mật cho người dùng. Với sự phát triển nhanh chóng của công nghệ, nhà thông minh hứa hẹn sẽ trở thành một phần không thể thiếu trong cuộc sống hiện đại, giúp nâng cao chất lượng sống và tạo ra những trải nghiệm sống thông minh hơn.

### **1.4. Các giao thức và công nghệ kết nối trong nhà thông minh**

#### ***1.4.1. Giao thức kết nối không dây***

- **Wi-Fi**
  - **Mô tả:** Wi-Fi là một trong những giao thức phổ biến nhất, cho phép các thiết bị nhà thông minh kết nối với mạng Internet để điều khiển từ xa.
  - **Ưu điểm:**
    - Phổ biến, dễ thiết lập và sử dụng.
    - Tốc độ truyền tải dữ liệu cao, hỗ trợ nhiều thiết bị.
    - Có thể điều khiển từ xa thông qua Internet.
  - **Nhược điểm:**
    - Tiêu tốn năng lượng nhiều hơn so với các giao thức khác.

- Khi có quá nhiều thiết bị kết nối, mạng có thể bị tắc nghẽn.
- **Ứng dụng:** Camera an ninh, loa thông minh, công tắc thông minh.
- **Zigbee**
  - **Mô tả:** Zigbee là một giao thức truyền thông không dây chuyên dụng cho các thiết bị IoT, hoạt động trên băng tần 2.4GHz.
  - **Ưu điểm:**
    - Tiêu thụ điện năng thấp, phù hợp với các thiết bị chạy pin.
    - Có thể kết nối lên đến hàng trăm thiết bị trong một mạng.
    - Hoạt động trên mô hình **Mesh Network**, giúp tăng phạm vi kết nối.
  - **Nhược điểm:**
    - Tốc độ truyền dữ liệu thấp hơn Wi-Fi.
    - Cần có một bộ điều khiển trung tâm (Hub) để kết nối với Internet.
  - **Ứng dụng:** Cảm biến nhiệt độ, bóng đèn thông minh, công tắc thông minh.
- **Z-Wave**
  - **Mô tả:** Z-Wave là một giao thức truyền thông không dây hoạt động ở băng tần 800-900MHz, giúp giảm nhiễu so với Wi-Fi và Zigbee.
  - **Ưu điểm:**
    - Ít bị nhiễu hơn do không sử dụng băng tần 2.4GHz.
    - Tiêu thụ điện năng thấp, phù hợp với các thiết bị dùng pin.
    - Hỗ trợ kết nối thiết bị theo mô hình **Mesh Network**.
  - **Nhược điểm:**
    - Chi phí cao hơn so với Zigbee.
    - Cần có một bộ điều khiển trung tâm (Hub) để kết nối với Internet.
  - **Ứng dụng:** Cảm biến chuyển động, khóa cửa thông minh, bộ điều khiển rèm cửa.
- **Bluetooth Low Energy (BLE)**
  - **Mô tả:** BLE là phiên bản tiết kiệm năng lượng của Bluetooth, được sử dụng để kết nối các thiết bị gần nhau mà không cần mạng Wi-Fi.

- **Ưu điểm:**
    - Tiêu thụ năng lượng thấp, kéo dài tuổi thọ pin của thiết bị.
    - Không cần bộ điều khiển trung tâm, có thể kết nối trực tiếp với điện thoại.
  - **Nhược điểm:**
    - Phạm vi kết nối gần (khoảng 10-30m).
    - Tốc độ truyền dữ liệu thấp hơn Wi-Fi.
  - **Ứng dụng:** Khóa cửa thông minh, cảm biến sức khỏe, thiết bị đeo tay.
- **LoRa (Long Range)**
- **Mô tả:** LoRa là một công nghệ truyền dữ liệu tầm xa, sử dụng băng tần thấp để truyền tín hiệu ở khoảng cách xa mà không tiêu tốn nhiều năng lượng.
  - **Ưu điểm:**
    - Phạm vi kết nối rất xa (hơn 10km).
    - Tiêu thụ năng lượng cực thấp, phù hợp với các thiết bị IoT ngoài trời.
  - **Nhược điểm:**
    - Tốc độ truyền dữ liệu thấp.
    - Chưa phổ biến trong các hệ thống nhà thông minh gia đình.
  - **Ứng dụng:** Hệ thống cảm biến giám sát môi trường, hệ thống tưới tiêu thông minh.

#### ***1.4.2. Giao thức kết nối có dây***

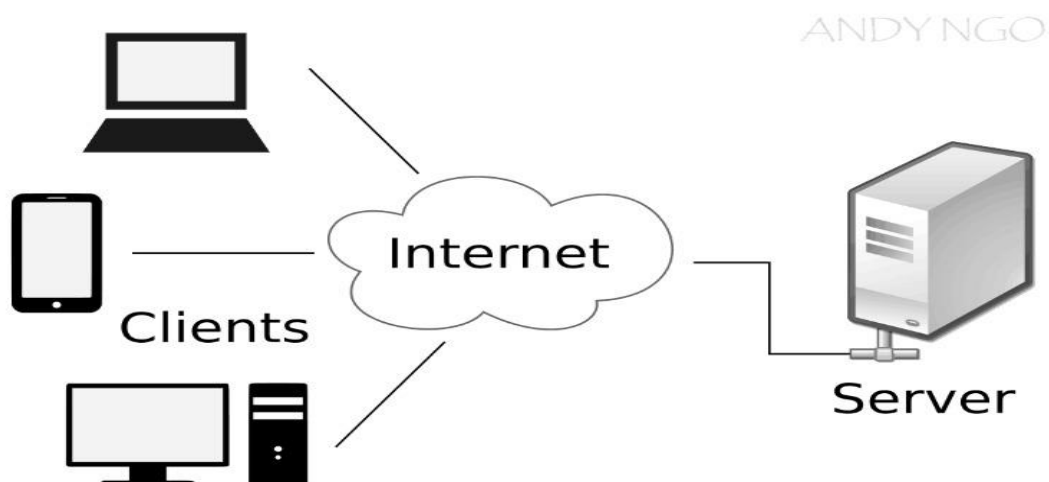
- **Ethernet**
- **Mô tả:** Ethernet là giao thức kết nối có dây thông qua cáp mạng LAN, giúp đảm bảo tốc độ truyền dữ liệu nhanh và ổn định.
  - **Ưu điểm:**
    - Độ ổn định cao, không bị nhiễu sóng như Wi-Fi.
    - Tốc độ truyền tải dữ liệu nhanh (lên đến 10Gbps).
  - **Nhược điểm:**
    - Cần đi dây, khó lắp đặt với các hệ thống nhà có sẵn.
    - Thiếu tính linh hoạt so với các giao thức không dây.

- **Ứng dụng:** Camera an ninh, bộ điều khiển trung tâm, hệ thống giải trí gia đình.
- **Powerline Communication (PLC)**
  - **Mô tả:** PLC là công nghệ truyền dữ liệu qua đường dây điện, giúp các thiết bị nhà thông minh có thể kết nối mà không cần đi dây mạng riêng.
  - **Ưu điểm:**
    - Không cần lắp đặt dây mạng mới, tận dụng đường dây điện sẵn có.
    - Kết nối ổn định hơn so với Wi-Fi.
  - **Nhược điểm:**
    - Tốc độ truyền dữ liệu phụ thuộc vào chất lượng dây điện trong nhà.
    - Dễ bị nhiễu bởi các thiết bị điện khác.
  - **Ứng dụng:** Hệ thống an ninh, điều khiển thiết bị điện trong nhà.

#### 1.4.3. Kết luận

Việc lựa chọn giao thức và công nghệ kết nối phù hợp cho hệ thống nhà thông minh phụ thuộc vào nhiều yếu tố như phạm vi kết nối, tốc độ truyền tải, mức tiêu thụ năng lượng và yêu cầu bảo mật.

### 1.5. Giới thiệu về mô hình Client – Server



Hình 1.4. Mô hình Client – Server

#### 1.5.1. Giới thiệu về mô hình Client – Server

Mô hình **Client – Server** là một kiến trúc mạng phổ biến trong phát triển hệ thống và ứng dụng, trong đó **Client** (máy khách) gửi yêu cầu và **Server** (máy chủ)

xử lý yêu cầu rồi phản hồi lại. Đây là mô hình nền tảng của hầu hết các ứng dụng web, hệ thống nhà thông minh, ứng dụng di động, và các dịch vụ trực tuyến hiện nay.

### ***1.5.2. Cấu trúc của mô hình Client – Server***

Mô hình này bao gồm hai thành phần chính:

- **Client (Máy khách)**

- Là thiết bị hoặc phần mềm gửi yêu cầu đến máy chủ để nhận dữ liệu hoặc thực hiện một thao tác nào đó.
- Có thể là trình duyệt web, ứng dụng di động, thiết bị IoT, hoặc bất kỳ phần mềm nào có khả năng gửi yêu cầu đến máy chủ.
- Thông thường, Client chỉ đảm nhận giao diện và gửi yêu cầu, không xử lý logic nghiệp vụ phức tạp.

**Ví dụ:**

- Khi người dùng truy cập một website, trình duyệt web (Client) gửi yêu cầu đến Server để lấy nội dung trang web.
- Trong hệ thống nhà thông minh, ứng dụng trên điện thoại (Client) gửi lệnh bật/tắt đèn đến Server.

- **Server (Máy chủ)**

- Là một hệ thống phần mềm hoặc phần cứng chịu trách nhiệm xử lý yêu cầu từ Client và phản hồi kết quả.
- Chứa cơ sở dữ liệu, logic xử lý và tài nguyên cần thiết để cung cấp dịch vụ cho Client.
- Có thể phục vụ nhiều Client cùng lúc.

**Ví dụ:**

- Server của Facebook xử lý yêu cầu hiển thị bài viết khi người dùng mở ứng dụng Facebook.
- Server của nhà thông minh xử lý lệnh điều khiển từ xa của người dùng và gửi phản hồi về trạng thái thiết bị.



### ***1.5.3. Cách thức hoạt động của mô hình Client – Server***

- **Client gửi yêu cầu** đến Server thông qua giao thức truyền thông (HTTP, WebSocket, MQTT, v.v.).
- **Server nhận yêu cầu**, xử lý dữ liệu và thực hiện logic nghiệp vụ.
- **Server gửi phản hồi** cho Client, có thể là dữ liệu (JSON, XML, HTML) hoặc trạng thái của hệ thống.
- **Client nhận phản hồi**, hiển thị dữ liệu cho người dùng hoặc thực hiện hành động tương ứng.

Ví dụ về quá trình hoạt động của mô hình Client – Server trong hệ thống nhà thông minh:

- Người dùng mở ứng dụng trên điện thoại (Client) và nhấn nút “Bật đèn”.
- Ứng dụng gửi yêu cầu bật đèn đến Server thông qua mạng Internet.

### ***1.5.4. Các giao thức truyền thông trong mô hình Client – Server***

Mô hình này sử dụng nhiều giao thức để giao tiếp giữa Client và Server, phổ biến nhất là:

- **HTTP/HTTPS**: Được sử dụng trong các ứng dụng web và API.
- **WebSocket**: Kết nối thời gian thực giữa Client và Server, thường dùng trong chat, thông báo đẩy.
- **MQTT**: Giao thức nhẹ dành cho các thiết bị IoT, như trong nhà thông minh.
- **FTP**: Truyền tải tập tin giữa Client và Server.

### ***1.5.5. Ưu điểm và nhược điểm của mô hình Client – Server***

- **Ưu điểm**
  - **Dễ mở rộng**: Server có thể phục vụ nhiều Client cùng lúc.
  - **Bảo mật tốt hơn**: Dữ liệu quan trọng được lưu trên Server thay vì Client.
  - **Dễ bảo trì và nâng cấp**: Các thay đổi chỉ cần thực hiện trên Server mà không ảnh hưởng đến Client.
  - **Quản lý tập trung**: Dữ liệu và logic xử lý tập trung tại Server giúp kiểm soát hệ thống dễ dàng.
- **Nhược điểm**

- **Dễ mở rộng:** Server có thể phục vụ nhiều Client cùng lúc.
- **Bảo mật tốt hơn:** Dữ liệu quan trọng được lưu trên Server thay vì Client.
- **Dễ bảo trì và nâng cấp:** Các thay đổi chỉ cần thực hiện trên Server mà không ảnh hưởng đến Client.
- **Quản lý tập trung:** Dữ liệu và logic xử lý tập trung tại Server giúp kiểm soát hệ thống dễ dàng.

## 1.6. Nền tảng và công nghệ áp dụng phía phần mềm

### 1.6.1. Nền tảng Android



*Hình 1.5. Nền tảng Android*

Android là một hệ điều hành mã nguồn mở do Google phát triển, dựa trên nhân Linux, được thiết kế chủ yếu cho các thiết bị di động như điện thoại thông minh, máy tính bảng, đồng hồ thông minh, TV thông minh và hệ thống xe hơi. Ra mắt lần đầu vào năm 2008, Android nhanh chóng trở thành hệ điều hành di động phổ biến nhất thế giới nhờ khả năng tùy biến cao, hỗ trợ đa dạng thiết bị và hệ sinh thái ứng dụng phong phú thông qua Google Play.

Với cấu trúc linh hoạt và cộng đồng phát triển rộng lớn, Android không chỉ đáp ứng nhu cầu người dùng cá nhân mà còn là nền tảng lý tưởng cho các ứng dụng nhúng và hệ thống IoT, đặc biệt phù hợp với các giải pháp như nhà thông minh, thiết bị giám sát, và các hệ thống tự động hóa hiện đại.

### 1.6.2. Ngôn ngữ lập trình Java



Hình 1.6. Ngôn ngữ lập trình Java

**Java** là một ngôn ngữ lập trình hướng đối tượng, được phát triển bởi **Sun Microsystems**, hiện thuộc sở hữu của **Oracle**.

Các đặc điểm chính của Java bao gồm:

- **Đa nền tảng:** Chương trình Java có thể chạy trên bất kỳ hệ điều hành nào có hỗ trợ JVM, bao gồm Windows, macOS, Linux, và thậm chí các thiết bị di động. Điều này mang lại sự linh hoạt tối đa cho các nhà phát triển, giúp giảm thiểu sự phụ thuộc vào nền tảng phần cứng cụ thể.
- **Hướng đối tượng:** Java là một ngôn ngữ hướng đối tượng (OOP), nghĩa là mọi thành phần trong chương trình đều được xây dựng dưới dạng đối tượng, với các thuộc tính và phương thức.
- **Bộ nhớ tự quản lý:** Java sử dụng cơ chế **Garbage Collection**, tự động giải phóng bộ nhớ không còn sử dụng, giúp giảm thiểu vấn đề rò rỉ bộ nhớ và nâng cao hiệu suất ứng dụng.
- **Bảo mật:** Java cung cấp một mô hình bảo mật mạnh mẽ, đặc biệt là trong các ứng dụng web và mạng.
- **Thư viện phong phú:** Java có một bộ API và thư viện cực kỳ phong phú, bao gồm các thư viện cho giao diện người dùng (Swing, JavaFX), kết nối cơ sở dữ liệu (JDBC), lập trình mạng, và các công cụ phát triển ứng dụng web. Điều này giúp lập trình viên dễ dàng xây dựng các ứng dụng từ desktop đến các ứng dụng

di động, web và dịch vụ mạng.

### 1.6.3. Thư viện Retrofit



Hình 1.7. Thư viện Retrofit2

Retrofit2 là một thư viện HTTP client mạnh mẽ cho Android và Java, được phát triển bởi Square. Nó giúp đơn giản hóa việc giao tiếp với các API RESTful bằng cách chuyển đổi các API thành các interface, cho phép thực hiện các cuộc gọi mạng một cách dễ dàng và hiệu quả.

#### Các đặc điểm chính của Retrofit2:

- **Cấu trúc rõ ràng:** Retrofit cho phép bạn định nghĩa các API bằng cách sử dụng các interface, giúp mã nguồn dễ đọc và bảo trì hơn.
- **Hỗ trợ nhiều loại định dạng dữ liệu:** Retrofit có thể chuyển đổi dữ liệu JSON, XML, và các định dạng khác thông qua các thư viện như Gson, Jackson, hoặc Moshi.
- **Xử lý bất đồng bộ:** Retrofit hỗ trợ cả cuộc gọi đồng bộ và bất đồng bộ, cho phép bạn chọn phương pháp phù hợp với nhu cầu của ứng dụng.
- **Tích hợp với OkHttp:** Retrofit sử dụng OkHttp làm client HTTP, mang lại hiệu suất cao và khả năng xử lý các yêu cầu mạng hiệu quả.
- **Hỗ trợ tùy chỉnh:** Bạn có thể tùy chỉnh các yêu cầu và phản hồi, thêm các interceptor, hoặc tạo các converter riêng để xử lý dữ liệu theo nhu cầu.
- **Quản lý lỗi dễ dàng:** Retrofit cung cấp các cơ chế để xử lý lỗi một cách dễ dàng

và có thể mở rộng.

#### 1.6.4. Thư viện OkHttp



Hình 1.8. Thư viện OkHttp

OkHttp là một thư viện HTTP client dành cho Android và Java, được phát triển bởi Square. Nó giúp đơn giản hóa việc gửi và nhận các yêu cầu HTTP, hỗ trợ cả kết nối đồng bộ và bất đồng bộ. OkHttp nổi bật với các tính năng như:

- **Hiệu suất cao:** Tối ưu hóa kết nối HTTP/2, cho phép nhiều yêu cầu trong một kết nối duy nhất.
- **Quản lý cache:** Hỗ trợ lưu trữ và phục hồi dữ liệu, giảm thiểu băng thông.
- **Xử lý lỗi tự động:** Tự động xử lý các lỗi mạng và retry các yêu cầu.
- **Hỗ trợ WebSocket:** Cho phép giao tiếp theo thời gian thực giữa máy khách và máy chủ.
- **Dễ sử dụng:** Cung cấp API rõ ràng và dễ hiểu cho lập trình viên.

#### 1.6.5. NodeJS



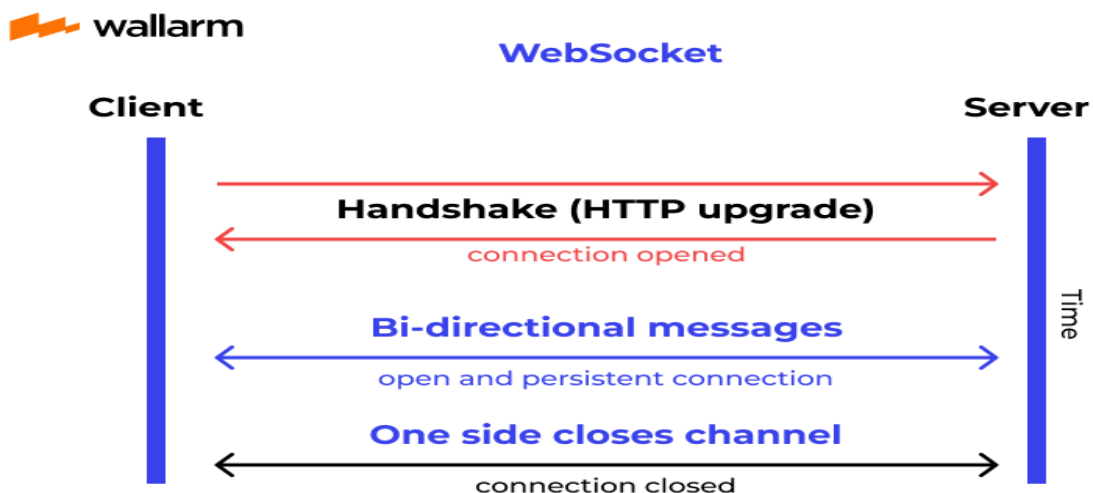
Hình 1.9. NodeJS

Node.js là một môi trường chạy JavaScript bên ngoài trình duyệt, được xây dựng trên nền tảng Chrome V8 JavaScript Engine. Được phát triển bởi Ryan Dahl và ra mắt vào năm 2009, Node.js cho phép chạy JavaScript trên máy chủ, tạo ra các ứng dụng mạng nhanh chóng và hiệu quả.

#### Các đặc điểm chính của Node.js:

- **Non-blocking I/O:** Node.js sử dụng mô hình nhập/xuất phi đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không bị chặn, giúp tăng hiệu suất cho các ứng dụng mạng.
- **Single-threaded:** Node.js chạy trên một luồng đơn, nhưng có thể xử lý hàng triệu kết nối đồng thời nhờ cơ chế sự kiện và callback.
- **Mô-đun hóa:** Node.js có hệ thống quản lý mô-đun thông qua npm (Node Package Manager), cho phép dễ dàng cài đặt và sử dụng lại các thư viện và gói phần mềm.
- **Chạy trên nhiều nền tảng:** Node.js có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux.
- **Thích hợp cho ứng dụng thời gian thực:** Node.js rất phù hợp cho việc phát triển các ứng dụng cần thời gian thực như chat, game trực tuyến và các dịch vụ API.

#### 1.6.6. WebSocket



Hình 1.10. WebSocket

WebSocket là một giao thức truyền thông mạng cho phép tạo kết nối hai chiều giữa máy khách và máy chủ, cho phép gửi và nhận dữ liệu theo thời gian thực. WebSocket được phát triển như một phần của HTML5 và là một giải pháp hiệu quả cho việc trao đổi thông tin trong các ứng dụng web.

**Các đặc điểm chính của WebSocket:**

- **Kết nối hai chiều:** WebSocket cho phép cả máy khách và máy chủ đều có thể gửi dữ liệu cho nhau mà không cần phải thực hiện yêu cầu HTTP mới.
- **Tốc độ cao:** Do giảm thiểu overhead của giao thức HTTP, WebSocket cho phép truyền dữ liệu nhanh hơn và hiệu quả hơn cho các ứng dụng cần phản hồi ngay lập tức.
- **Duy trì kết nối:** Sau khi kết nối được thiết lập, WebSocket duy trì kết nối mở, cho phép gửi và nhận dữ liệu liên tục mà không cần phải thiết lập lại kết nối.
- **Giảm băng thông:** WebSocket tiêu thụ băng thông ít hơn so với các phương pháp truyền thông truyền thống như AJAX, vì nó không cần phải gửi các header lớn trong mỗi yêu cầu.
- **Thích hợp cho ứng dụng thời gian thực:** WebSocket thường được sử dụng cho các ứng dụng như chat trực tuyến, game đa người chơi, và các dịch vụ thông báo thời gian thực.

**1.6.7. MongoDB**



*Hình 1.11. MongoDB*

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở, lưu trữ dữ liệu dưới dạng tài liệu JSON (hoặc BSON). Nó cho phép lưu trữ dữ liệu phi cấu trúc với mô hình dữ liệu linh hoạt, không yêu cầu cấu trúc cố định. MongoDB hỗ trợ mở rộng ngang, cho phép phân tán dữ liệu trên nhiều máy chủ, giúp xử lý lượng lớn dữ liệu

một cách hiệu quả. Với hiệu suất cao, ngôn ngữ truy vấn phong phú và khả năng sao lưu, MongoDB là lựa chọn phổ biến cho các ứng dụng web và di động hiện đại.

#### **1.6.8. Google Speech Recognition**



*Hình 1.12. Google Speech Recognition*

Google Speech Recognition là một dịch vụ nhận diện giọng nói do Google phát triển, dựa trên công nghệ trí tuệ nhân tạo và học sâu. Nó có khả năng chuyển đổi âm thanh thành văn bản với độ chính xác cao, hỗ trợ nhiều ngôn ngữ và được ứng dụng rộng rãi trong các sản phẩm như trợ lý ảo, ứng dụng di động và thiết bị thông minh.

#### **1.6.9. OpenAI**



*Hình 1.13. OpenAI*

**OpenAI** là tổ chức nghiên cứu trí tuệ nhân tạo (AI), nổi bật với các mô hình tiên tiến như xử lý ngôn ngữ tự nhiên và học máy. Thành lập vào năm 2015 bởi **Elon Musk**, **Sam Altman** và các nhà sáng lập khác, OpenAI tập trung phát triển các công cụ mạnh mẽ cho việc xây dựng ứng dụng thông minh. Các sản phẩm của OpenAI, như **GPT** và **DALL·E**, ứng dụng trong nhiều lĩnh vực như chatbot, phân tích dữ liệu và sáng tạo nội dung. Mục tiêu của OpenAI là phát triển công nghệ AI an toàn và hữu ích cho xã hội, giúp tối ưu hóa các quy trình và cải thiện trải nghiệm người dùng.



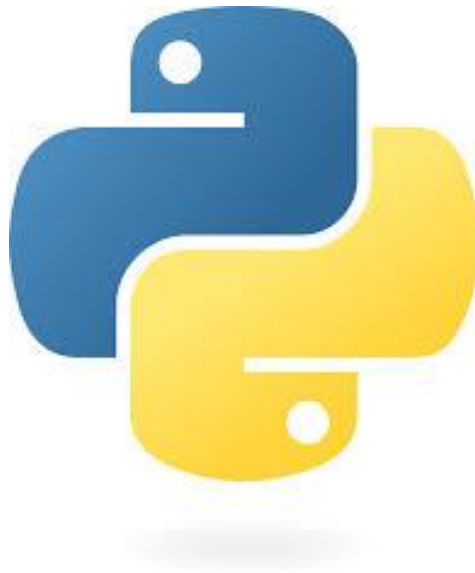
#### ***1.6.10. Blender***



*Hình 1.14. Blender*

Blender là một phần mềm mã nguồn mở dùng để thiết kế đồ họa 3D, được cộng đồng phát triển từ năm 1998. Nó hỗ trợ tạo mô hình, hoạt hình, render và chỉnh sửa video, phù hợp cho các nhà thiết kế, nghệ sĩ và nhà phát triển muốn xây dựng nội dung 3D chất lượng cao mà không mất phí.

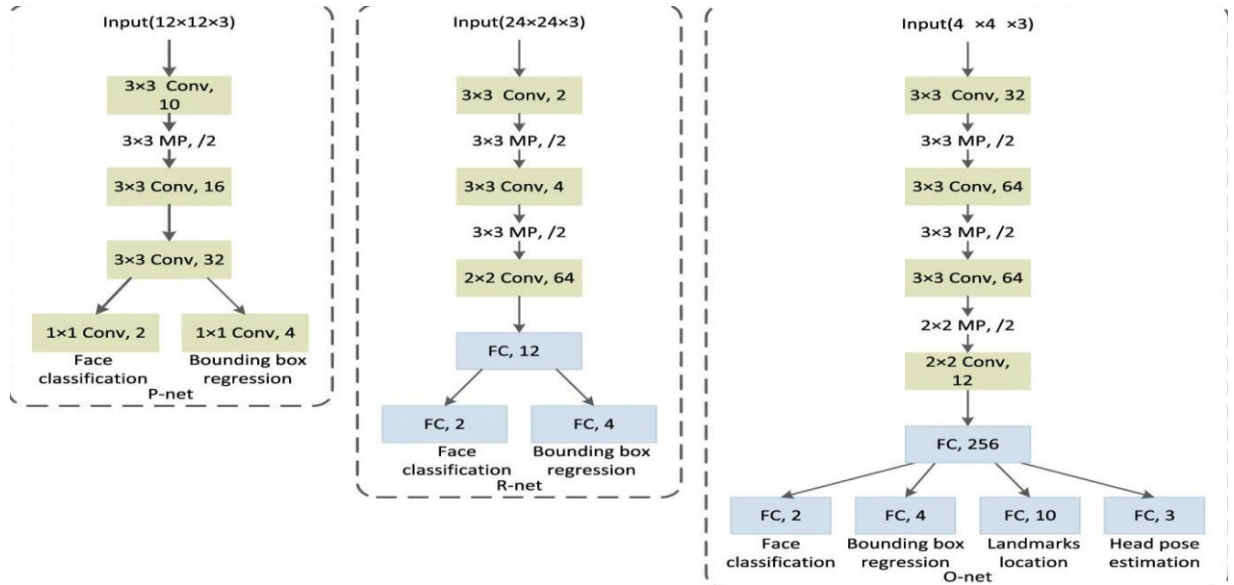
#### ***1.6.11. Python***



*Hình 1.15. Python*

Python là một ngôn ngữ lập trình bậc cao, được Guido van Rossum tạo ra vào năm 1991. Với cú pháp đơn giản, dễ học và thư viện phong phú, Python được sử dụng rộng rãi trong phát triển web, khoa học dữ liệu, trí tuệ nhân tạo và tự động hóa, trở thành một trong những ngôn ngữ phổ biến nhất thế giới.[2]

### 1.6.12. MTCNN



Hình 1.16. MTCNN

MTCNN (Multi-task Cascaded Convolutional Networks) là một thuật toán học sâu chuyên về nhận diện khuôn mặt, được phát triển dựa trên mạng nơ-ron tích chập. Nó nổi bật với khả năng phát hiện và căn chỉnh khuôn mặt chính xác trong hình ảnh hoặc video, thường được dùng trong các ứng dụng bảo mật và phân tích hình ảnh.

## 1.7. Phần mềm Arduino IDE và các thiết bị phần cứng

### 1.7.1. Giới thiệu về phần mềm Arduino IDE



Hình 1.17. Phần mềm Arduino IDE

**Arduino IDE (Integrated Development Environment)** là phần mềm phát triển mạnh mẽ, được sử dụng rộng rãi để lập trình các bo mạch Arduino và các thiết bị điện tử tương thích. Đây là công cụ lý tưởng cho những ai muốn phát triển ứng dụng điện tử nhúng, từ các dự án học tập cho đến các sản phẩm thương mại. Dưới đây là một số thông tin chi tiết về Arduino IDE:

### **Giới thiệu**

- **Định nghĩa:** Arduino IDE là một môi trường phát triển tích hợp, cho phép người dùng viết, biên dịch và tải lên mã nguồn vào các bo mạch Arduino, hỗ trợ các dự án từ đơn giản đến phức tạp.
- **Ngôn ngữ lập trình:** IDE sử dụng một phiên bản rút gọn của ngôn ngữ C/C++, kết hợp với các thư viện và hàm được tối ưu hóa để lập trình các bo mạch Arduino, giúp người dùng dễ dàng tương tác với phần cứng.

### **Tính năng chính**

- **Giao diện đơn giản:** Giao diện của Arduino IDE được thiết kế thân thiện và dễ sử dụng, phù hợp cho cả người mới bắt đầu lẫn lập trình viên có kinh nghiệm, giúp tối ưu hóa quy trình phát triển.
- **Trình biên dịch:** Arduino IDE tích hợp một trình biên dịch mạnh mẽ để biên dịch mã nguồn thành mã máy, sau đó tải lên bo mạch Arduino qua cổng USB, hỗ trợ các dự án điện tử thực tế.
- **Thư viện phong phú:** IDE cung cấp một bộ thư viện tích hợp đa dạng cho các linh kiện và cảm biến phổ biến, giúp người dùng tiết kiệm thời gian và dễ dàng xây dựng các hệ thống phức tạp mà không cần phải viết lại mã từ đầu.
- **Ghi chú và ví dụ:** Có sẵn nhiều ví dụ và tài liệu hướng dẫn chi tiết, giúp người dùng tìm hiểu cách sử dụng các hàm và thư viện có sẵn để thực hiện các tác vụ phổ biến trong lập trình Arduino.
- **Hỗ trợ nhiều nền tảng:** Arduino IDE có sẵn cho các hệ điều hành Windows, macOS và Linux, giúp người dùng có thể làm việc trên nhiều nền tảng mà không gặp phải trở ngại.



- **Giao thức:** Hỗ trợ TCP/IP, UDP, HTTP, MQTT, WebSocket.
- **Pin GPIO:** Khoảng 36 pin GPIO, hỗ trợ SPI, I2C, UART, ADC, DAC.
- **Điện áp hoạt động:** 2.7 - 3.6 V.
- **Tiêu thụ năng lượng:**
  - Chế độ hoạt động: Khoảng 100-240 mA (tùy tác vụ).
  - Chế độ ngủ sâu: Khoảng 5  $\mu$ A.

### 1.7.3. ESP32 CAM AI THINKER



Hình 1.19. ESP32 CAM AI THINKER

#### Giới thiệu:

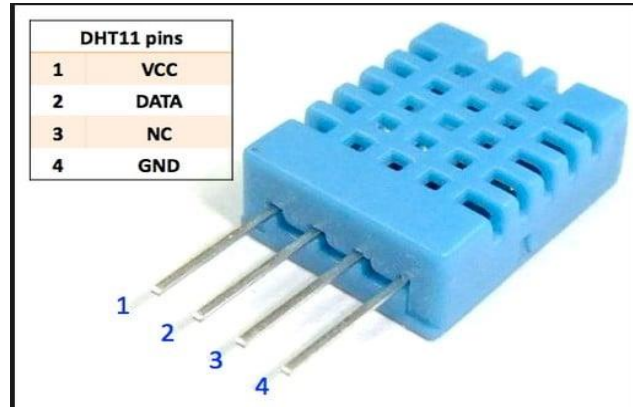
- **Mục đích:** Kết hợp vi điều khiển và camera cho các ứng dụng IoT liên quan đến hình ảnh và giám sát.
- **Tính năng:** Nhỏ gọn, tích hợp camera, hỗ trợ Wi-Fi/Bluetooth, lý tưởng cho dự án DIY như hệ thống an ninh hoặc truyền video.

#### Thông số kỹ thuật chính:

- **Chipset:** ESP32-S (dựa trên ESP32).
- **Tần số hoạt động:** 2.4 GHz (Wi-Fi 802.11 b/g/n, Bluetooth 4.2).
- **Băng thông:** Tối đa 150 Mbps (Wi-Fi).
- **Dung lượng bộ nhớ:**
  - RAM: 520 KB SRAM + 4 MB PSRAM.
  - Flash: 4 MB (mở rộng qua thẻ microSD).
- **Giao thức:** Hỗ trợ TCP/IP, UDP, HTTP, WebSocket.
- **Pin GPIO:** 10 pin GPIO khả dụng, hỗ trợ SPI, UART, I2C.

- **Camera:** OV2640 (2MP, góc nhìn 66.5°).
- **Điện áp hoạt động:** 5 V (qua USB) hoặc 3.3 V (qua pin).
- **Tiêu thụ năng lượng:**
  - Chế độ hoạt động: Khoảng 180-250 mA (khi dùng camera).
  - Chế độ ngủ: Khoảng 20  $\mu$ A.

#### 1.7.4. Cảm biến nhiệt độ, độ ẩm DHT11



Hình 1.20. Cảm biến nhiệt độ, độ ẩm DHT11

DHT11 là một cảm biến đo nhiệt độ và độ ẩm phổ biến, thường được sử dụng trong các dự án IoT và tự động hóa. Dưới đây là một số thông tin về DHT11 cùng với thông số kỹ thuật chính:

##### Giới thiệu

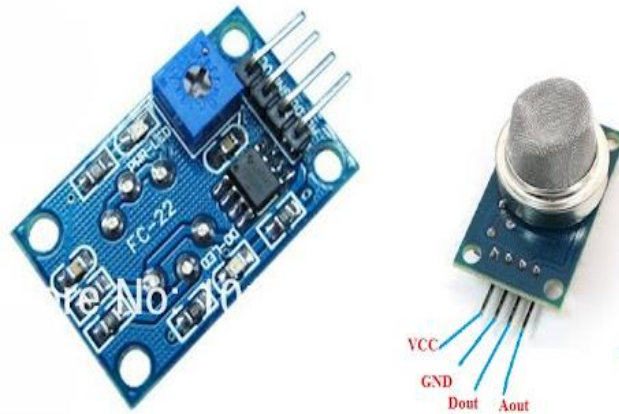
- **Mục đích:** Đo lường nhiệt độ và độ ẩm trong không khí.
- **Ứng dụng:** Thích hợp cho các ứng dụng trong nhà thông minh, dự báo thời tiết, và điều khiển môi trường.

##### Thông số kỹ thuật chính

- **Phạm vi đo:**
  - Nhiệt độ: 0 - 50 °C ( $\pm 2$  °C).
  - Độ ẩm: 20 - 80% RH ( $\pm 5$ % RH).
- **Độ phân giải:**
  - Nhiệt độ: 1 °C.
  - Độ ẩm: 1% RH.
- **Thời gian phản hồi:** Khoảng 1 giây.

- **Điện áp hoạt động:** 3.5 - 5.5 V.
- **Kết nối:** Sử dụng giao thức 1-wire, giúp dễ dàng kết nối với vi điều khiển như Arduino hoặc ESP32.
- **Kích thước:** Nhỏ gọn, thường có dạng module với kích thước khoảng 15mm x 25mm.

#### 1.7.5. Cảm biến khí gas MQ2



Hình 1.21. Cảm biến khí gas MQ2

Cảm biến khí gas MQ-2 là một trong những cảm biến phổ biến nhất trong dòng cảm biến MQ, được sử dụng để phát hiện sự hiện diện của nhiều loại khí khác nhau, đặc biệt là khí metan ( $\text{CH}_4$ ), propane ( $\text{C}_3\text{H}_8$ ), và khói. Dưới đây là một số thông tin chi tiết về cảm biến MQ-2:

#### Giới thiệu

- **Mục đích:** Phát hiện khí gas và khói trong môi trường, giúp cảnh báo về sự rò rỉ khí và đảm bảo an toàn.
- **Ứng dụng:** Sử dụng trong các hệ thống an toàn gia đình, các thiết bị cảnh báo, và các dự án IoT.

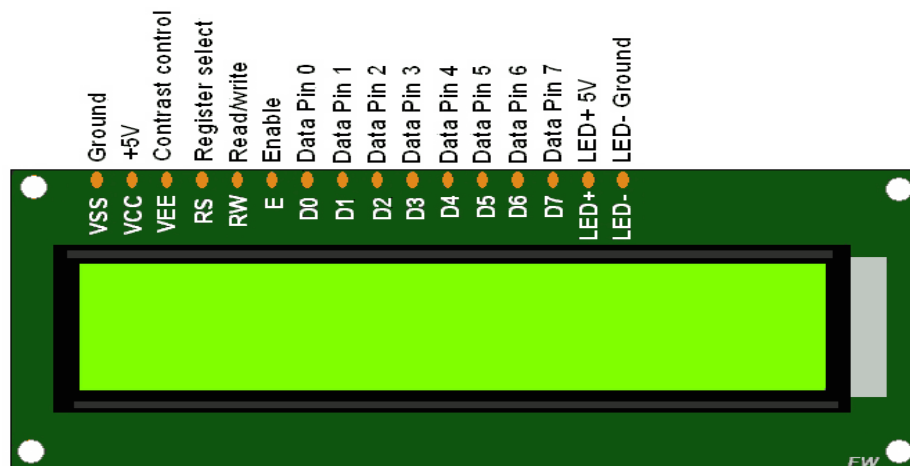
#### Thông số kỹ thuật chính

- **Điện áp hoạt động:** 5V DC.
- **Dòng tiêu thụ:** Khoảng 150mA (khi hoạt động).



- **Phạm vi phát hiện:**
  - Khí metan: 300 - 10,000 ppm (phần triệu).
  - Khí propane: 200 - 10,000 ppm.
- **Khối:** 0.1 - 10% (thể tích).
- **Thời gian phản hồi:** Nhanh, thường trong vài giây.
- **Kích thước:** Khoảng 32mm x 20mm x 25mm.
- **Mô hình đầu ra:** Tín hiệu analog và digital.

#### 1.7.6. Màn hình LCD:



Hình 1.22. Màn hình LCD 16x2

Màn hình LCD (Liquid Crystal Display) là một loại màn hình phẳng sử dụng công nghệ tinh thể lỏng để hiển thị hình ảnh. Đây là công nghệ phổ biến được sử dụng trong nhiều thiết bị điện tử như tivi, máy tính, điện thoại di động, và nhiều thiết bị khác. Dưới đây là một số thông tin chi tiết về màn hình LCD:

#### Giới thiệu

- **Cấu tạo:** Màn hình LCD bao gồm nhiều lớp, bao gồm một lớp tinh thể lỏng nằm giữa hai lớp kính hoặc nhựa. Các lớp này được chiếu sáng bởi đèn nền, thường là LED hoặc CCFL.
- **Nguyên lý hoạt động:** Khi có điện áp được áp dụng, các phân tử tinh thể lỏng thay đổi hướng và điều chỉnh ánh sáng đi qua, từ đó tạo ra hình ảnh.

#### Ưu điểm

- **Kích thước mỏng và nhẹ:** Màn hình LCD thường mỏng hơn so với màn hình



CRT (Cathode Ray Tube) truyền thống, giúp tiết kiệm không gian.

- **Tiết kiệm năng lượng:** Tiêu thụ năng lượng thấp hơn, đặc biệt khi sử dụng đèn nền LED.
- **Chất lượng hình ảnh tốt:** Có độ phân giải cao và khả năng hiển thị màu sắc tốt.

#### Nhược điểm

- **Góc nhìn hạn chế:** Chất lượng hình ảnh có thể giảm khi nhìn từ các góc khác nhau.
- **Thời gian phản hồi:** Có thể chậm hơn so với các công nghệ như OLED, gây hiện tượng mờ khi hiển thị các hình ảnh chuyển động nhanh.

#### 1.7.7. Còi 5V



Hình 1.23. Còi 5v

Còi 5V (hay còn gọi là buzzer 5V) là một thiết bị điện tử sử dụng để phát ra âm thanh, thường được sử dụng trong các ứng dụng cảnh báo, thông báo, hoặc trong các dự án DIY. Dưới đây là một số thông tin chi tiết về còi 5V:

#### Giới thiệu

- **Mục đích:** Còi 5V được sử dụng để phát ra âm thanh cảnh báo hoặc thông báo trong các thiết bị điện tử.
- **Loại:** Có hai loại chính:
  - **Còi thụ động (Passive Buzzer):** Phát âm thanh khi có tín hiệu tần số được cung cấp từ vi điều khiển.
  - **Còi chủ động (Active Buzzer):** Phát ra âm thanh khi có điện áp cấp vào mà không cần tín hiệu tần số.

#### Thông số kỹ thuật chính

- **Điện áp hoạt động:** 5V DC (có thể có loại 3V hoặc 12V).

- **Dòng tiêu thụ:** Khoảng 20 - 30 mA (tùy thuộc vào loại và kích thước).
- **Tần số âm thanh:** Thường từ 2 kHz đến 4 kHz đối với còi chủ động; còi thụ động có thể phát ra âm thanh ở nhiều tần số khác nhau.

#### 1.7.8. Led xanh 3v



Hình 1.24. Led xanh 3v

LED xanh 3V là một loại diode phát sáng (LED) có khả năng phát ra ánh sáng màu xanh khi có dòng điện đi qua. Dưới đây là thông tin chi tiết về LED xanh 3V:

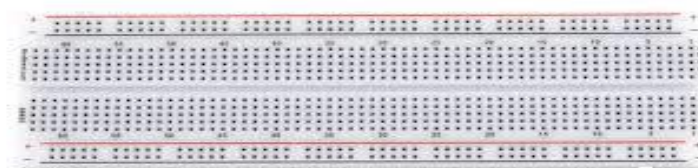
#### Giới thiệu

- **Mục đích:** LED xanh thường được sử dụng trong các ứng dụng chiếu sáng, trang trí, và báo hiệu.
- **Cấu tạo:** Gồm một khối bán dẫn, thường là gallium phosphide (GaP) hoặc gallium nitride (GaN) để tạo ra ánh sáng xanh.

#### Thông số kỹ thuật chính

- **Điện áp hoạt động:** Khoảng 2.0V - 3.2V (thường là 3V cho LED xanh).
- **Dòng điện tối ưu:** Thường từ 20mA đến 30mA để đạt được độ sáng tối ưu.
- **Màu sắc:** Ánh sáng xanh (tần số khoảng 450-495 nm).
- **Kích thước:** Có nhiều kích thước khác nhau, thường là 5mm, 10mm, hoặc loại SMD (Surface Mount Device).

#### 1.7.9. Breadboard



Hình 1.25. Breadboard

Breadboard là một công cụ quan trọng trong việc thiết kế và thử nghiệm các mạch điện mà không cần hàn. Dưới đây là thông tin chi tiết về breadboard:

### Giới thiệu

- **Định nghĩa:** Breadboard là một bảng mạch không cần hàn, được thiết kế để lắp ráp các linh kiện điện tử tạm thời. Nó cho phép người dùng kết nối và thử nghiệm mạch điện một cách dễ dàng và nhanh chóng.
- **Chất liệu:** Thường làm bằng nhựa với các hàng cắm bằng đồng mạ kẽm bên trong.

### Cấu trúc

- **Hàng cắm (Terminal Strips):** Các hàng cắm chính trên breadboard thường được chia thành hai phần:
  - **Hàng ngang:** Được sử dụng để kết nối nguồn (VCC) và đất (GND).
  - **Hàng dọc:** Dùng để kết nối các linh kiện điện tử, như điện trở, LED, transistor, vi điều khiển, v.v.
- **Các khe cắm:** Breadboard có nhiều khe cắm để linh kiện được gắn vào. Mỗi khe cắm tương ứng với một chân của linh kiện.

#### 1.7.10. Dây cắm breadboard



Hình 1.26. Dây cắm breadboard

Dây cắm breadboard (hay còn gọi là jumper wires) là các dây dẫn được sử dụng để kết nối các linh kiện điện tử trên breadboard. Dưới đây là thông tin chi tiết về dây cắm breadboard:

### Giới thiệu

- **Định nghĩa:** Dây cắm breadboard là những đoạn dây ngắn, thường có hai đầu cắm (hoặc đầu cắm duy nhất), dùng để tạo kết nối giữa các linh kiện trên breadboard mà không cần hàn.
- **Chất liệu:** Dây thường được làm từ đồng mạ kẽm hoặc đồng không gỉ, được bọc nhựa để cách điện.

### Đặc điểm

- **Độ dài:** Có nhiều độ dài khác nhau, từ vài cm đến hơn 30 cm, để phù hợp với nhiều ứng dụng.
- **Kích thước đầu cắm:** Đầu cắm thường là chân cắm dạng male (nam) hoặc female (nữ):
- **Male (Nam):** Đầu cắm có chân, có thể cắm trực tiếp vào các khe trên breadboard.
- **Female (Nữ):** Đầu cắm có khe để tiếp xúc với chân của linh kiện hoặc đầu cắm male khác.

#### 1.7.11. Quạt tản nhiệt 5v



Hình 1.27. Quạt tản nhiệt 5v

Quạt tản nhiệt 5V là một thiết bị điện cơ đơn giản nhưng hiệu quả, thường được sử dụng trong nhiều ứng dụng điện tử để duy trì nhiệt độ hoạt động của các linh kiện trong phạm vi an toàn.:

#### Giới thiệu:

- **Định nghĩa:** Quạt tản nhiệt 5V là loại quạt điện hoạt động bằng nguồn điện 5V, được sử dụng chủ yếu để làm mát các linh kiện điện tử, giảm nhiệt độ của bo mạch, vi xử lý và các thiết bị điện tử khác, từ đó giúp đảm bảo hiệu suất hoạt động ổn định và kéo dài tuổi thọ cho các linh kiện.
- **Chất liệu:** Quạt thường được chế tạo từ các chất liệu bền, như nhựa hoặc kim loại, giúp bảo vệ và gia tăng độ bền cho quạt. Cánh quạt thường được làm bằng nhựa, giúp giảm trọng lượng và chi phí sản xuất.

### Đặc điểm:

- **Điện áp hoạt động:** Quạt tản nhiệt 5V hoạt động trên nguồn điện 5V, là mức điện áp tiêu chuẩn cho nhiều thiết bị điện tử nhỏ, như các hệ thống máy tính, Arduino, Raspberry Pi và các ứng dụng USB khác. Điều này giúp dễ dàng tích hợp với các thiết bị sử dụng nguồn 5V, mà không cần thêm bộ chuyển đổi điện áp phức tạp.
- **Kích thước:** Quạt 5V có đa dạng kích thước, phù hợp với nhiều nhu cầu sử dụng khác nhau. Các kích thước phổ biến bao gồm 40mm, 60mm và 80mm, giúp người dùng có thể lựa chọn sản phẩm phù hợp với không gian và yêu cầu tản nhiệt của từng hệ thống.
- **Công suất:** Quạt tản nhiệt 5V tiêu thụ công suất khá thấp, thường từ 1-5W, giúp tiết kiệm năng lượng và giảm thiểu mức tiêu thụ điện cho các ứng dụng điện tử nhỏ. Nhờ vào công suất thấp, quạt này có thể hoạt động liên tục mà không gây quá tải cho các mạch điện hoặc bộ nguồn.

Với những đặc điểm này, quạt tản nhiệt 5V trở thành một phần không thể thiếu trong việc đảm bảo hiệu suất ổn định và độ bền của các thiết bị điện tử, đặc biệt trong các hệ thống đòi hỏi làm mát hiệu quả nhưng tiết kiệm năng lượng.

#### 1.7.12. Servo SG90



Hình 1.28. Servo SG90

Servo SG90 là một loại servo nhỏ gọn và phổ biến, thường được sử dụng trong các dự án điện tử và robotics.

## **Giới thiệu**

- **Định nghĩa:** Servo SG90 là một loại servo động cơ DC được sử dụng để điều khiển góc quay. Nó có khả năng điều chỉnh vị trí của một bộ phận, thường được dùng trong các ứng dụng điều khiển chính xác như robot, mô hình máy bay, và các dự án DIY.
- **Kích thước:** Là loại servo nhỏ gọn, dễ dàng tích hợp vào các thiết bị hoặc mô hình.

## **Thông số kỹ thuật**

- **Điện áp hoạt động:** 4.8V đến 6.0V
- **Góc quay:** Khoảng 180 độ
- **Tốc độ:** Khoảng 0.1s/60 độ ở 6V
- **Lực kéo:** Khoảng 1.5 kg/cm (ở 6V)
- **Kích thước:** Khoảng 22.5 x 11.5 x 22 mm
- **Trọng lượng:** Khoảng 9 g

## CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Đánh giá các hệ thống liên quan và hướng phát triển của đề tài

#### 2.1.1. Các dự án tiêu biểu trong và ngoài nước

##### - Dự án trong nước:

- **Hệ thống nhà thông minh của Viettel:** Viettel cung cấp nền tảng quản lý nhà thông minh qua ứng dụng di động, nhưng gặp hạn chế trong việc tích hợp nhiều loại thiết bị từ các nhà sản xuất khác nhau. Hệ thống này cũng chưa hỗ trợ truyền video thời gian thực hay nhận diện khuôn mặt để xác thực người dùng.

- **Nghiên cứu tại Đại học Bách Khoa TP.HCM:** Các nghiên cứu tại đây tập trung vào ứng dụng IoT trong điều khiển thiết bị, nhưng chưa khai thác khả năng tích hợp trí tuệ nhân tạo (AI) để nâng cao bảo mật và trải nghiệm người dùng.

- **Dự án Smart City Đà Nẵng:** Đà Nẵng đã triển khai IoT vào quản lý đô thị và nhà thông minh, nhưng vẫn gặp vấn đề về kết nối thiết bị và bảo mật dữ liệu do thiếu một nền tảng trung gian hiệu quả.

##### - Dự án quốc tế:

- **Google Nest:** Hệ thống nhà thông minh của Google có khả năng điều khiển tự động, nhưng thiếu tính linh hoạt trong việc tích hợp phần cứng giá rẻ như ESP32, làm cho chi phí triển khai cao hơn so với các giải pháp mở.

- **Nghiên cứu tại MIT:** Một nhóm nghiên cứu tại MIT đã phát triển hệ thống quản lý nhà thông minh với nhiều tính năng tối ưu, nhưng chi phí triển khai quá cao khiến việc áp dụng rộng rãi gặp nhiều hạn chế.

- **Dự án Smart Home tại Singapore:** Singapore đã tích hợp IoT vào các hệ thống nhà thông minh tiên tiến, nhưng vẫn tồn tại vấn đề bảo mật và khó khăn trong việc kết nối các thiết bị từ nhiều nhà sản xuất.

#### 2.1.2. Điểm khác biệt và ưu thế của dự án

Dự án "**Xây dựng hệ thống quản lý nhà thông minh ứng dụng IoT và AI sử dụng ESP32 và ESP32-CAM**" có một số điểm khác biệt nổi bật so với các hệ thống hiện có trong và ngoài nước:

**- Tích hợp nhận diện khuôn mặt để phát hiện người lạ và gửi cảnh báo:** Khác với các hệ thống chỉ dừng lại ở mức điều khiển thiết bị, hệ thống trong đề tài có khả năng sử dụng ESP32-CAM để nhận diện khuôn mặt. Nếu phát hiện người lạ, hệ thống tự động gửi cảnh báo đến người dùng qua ứng dụng, giúp tăng cường giám sát và đảm bảo an toàn cho ngôi nhà.

**- Truyền video thời gian thực qua WebSocket:** Hệ thống sử dụng ESP32-CAM kết hợp WebSocket để truyền video trực tiếp đến ứng dụng, hỗ trợ giám sát linh hoạt và kịp thời. Đây là một điểm cải tiến so với nhiều hệ thống hiện tại ở Việt Nam chưa tích hợp khả năng truyền video hoặc còn phụ thuộc vào các giao thức truyền thống.

**- Điều khiển thiết bị và giao tiếp bằng giọng nói thông qua trợ lý ảo 3D:** Việc kết hợp trợ lý ảo 3D và Google Speech Recognition mang lại trải nghiệm điều khiển trực quan và thân thiện hơn cho người dùng, giúp họ có thể tương tác với hệ thống bằng giọng nói một cách tự nhiên.

**- Tính mở và khả năng tùy chỉnh linh hoạt:** Thay vì phụ thuộc vào hệ sinh thái khép kín như Google Nest, hệ thống trong đề tài được thiết kế dựa trên các nền tảng phổ biến như ESP32, giúp dễ dàng mở rộng và tích hợp thêm thiết bị trong tương lai.

## 2.2. Thiết kế phần mềm

### 2.2.1. Mô tả các chức năng và tác nhân của hệ thống

#### a) Các yêu cầu chức năng của hệ thống

*Bảng 2.1. Các yêu cầu chức năng của hệ thống*

Tác nhân	Chức năng
Người dùng	<ul style="list-style-type: none"> <li>- Đăng nhập</li> <li>- Thống kê dữ liệu đo từ cảm biến</li> <li>- Quản lý phòng</li> <li>- Quản lý thiết bị</li> <li>- Quản lý kết nối AccessKey</li> <li>- Quản lý camera</li> <li>- Quản lý tài khoản</li> <li>- Cài đặt hệ thống</li> <li>- Quản lý khuôn mặt</li> <li>- Trợ lý ảo</li> <li>- Thông báo</li> </ul>



### ***b) Các yêu cầu phi chức năng của hệ thống***

#### **- Hiệu suất**

- Hệ thống cần đảm bảo xử lý và truyền dữ liệu thời gian thực (bao gồm video từ ESP32-CAM và lệnh qua WebSocket) với độ trễ tối đa 2 giây trong điều kiện mạng Wi-Fi ổn định (băng thông tối thiểu 10 Mbps).

- Ứng dụng di động và trợ lý ảo 3D phải phản hồi lệnh giọng nói từ Google Speech Recognition trong vòng 1 giây khi mạng internet hoạt động bình thường.

- Hệ thống phải duy trì hoạt động ổn định khi kết nối đồng thời ít nhất 5 thiết bị (như đèn, quạt, cảm biến) qua ESP32 mà không xảy ra gián đoạn.

#### **- Bảo mật**

- Dữ liệu truyền qua WebSocket (video, thông tin cảm biến) cần được mã hóa bằng giao thức bảo mật như AES-128 hoặc TLS để ngăn chặn truy cập trái phép.

- Hệ thống yêu cầu xác thực người dùng thông qua nhận diện khuôn mặt (MTCNN) hoặc mật khẩu trên ứng dụng, với tỷ lệ lỗi nhận diện dưới 5%.

- Thông tin cá nhân và dữ liệu video từ ESP32-CAM phải được bảo vệ an toàn, không để lộ nếu không có sự cho phép của người dùng.

#### **- Khả năng mở rộng**

- Hệ thống cần hỗ trợ thêm tối thiểu 10 thiết bị hoặc cảm biến mới mà không cần thay đổi cấu trúc phần cứng chính của ESP32 và ESP32-CAM.

- Ứng dụng di động phải duy trì hiệu suất ổn định khi mở rộng sang các nền tảng khác (như iOS) hoặc khi có đến 50 người dùng truy cập cùng lúc.

#### **- Độ tin cậy**

- Hệ thống phải hoạt động liên tục trong 24 giờ mà không gặp lỗi nghiêm trọng trong điều kiện môi trường thông thường (nhiệt độ 20-35°C, độ ẩm 40-70%).

#### **- Tính khả dụng**

- Giao diện ứng dụng di động (dùng Android SDK) và trợ lý ảo 3D cần thiết kế thân thiện, giúp người dùng không có kinh nghiệm công nghệ làm quen trong vòng 5 phút.

- Hệ thống phải cung cấp thông báo rõ ràng qua ứng dụng hoặc giọng nói khi xảy ra sự cố, đảm bảo 100% cảnh báo được gửi trong vòng 3 giây.

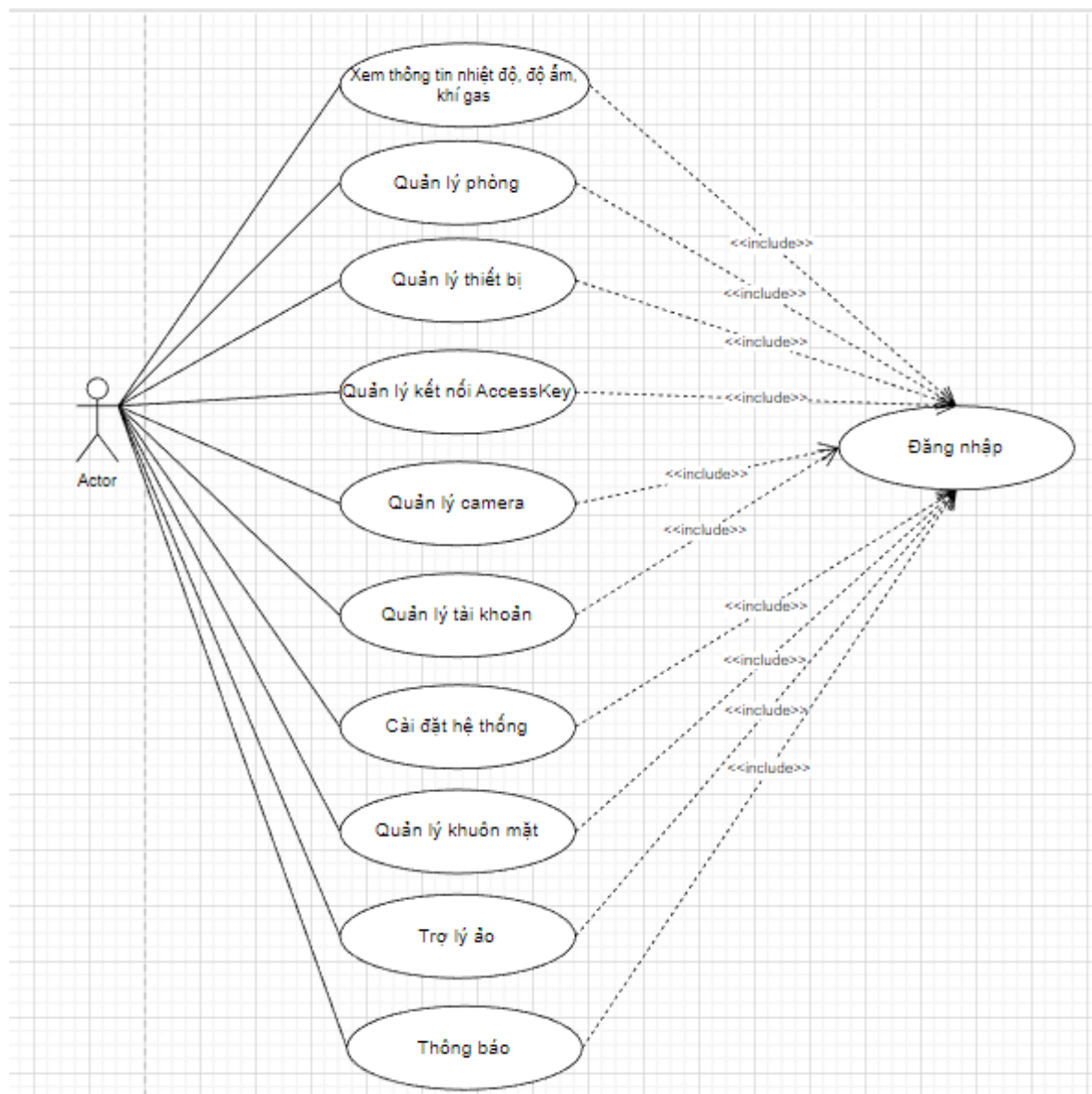
**- Khả năng tương thích**

- Hệ thống cần hoạt động tốt trên thiết bị Android từ phiên bản 8.0 trở lên và tương thích với mạng Wi-Fi chuẩn 802.11 b/g/n.
- Các thành phần phần mềm (Python, Java) phải tương thích với các thư viện mã nguồn mở phổ biến như Retrofit hay WebSocket để dễ dàng tích hợp và bảo trì.

**- Khả năng bảo trì**

- Mã nguồn (Python, C++, Java) cần được tổ chức rõ ràng, đi kèm tài liệu hướng dẫn, để nhóm phát triển khác có thể cập nhật trong vòng 1 tuần.

**2.2.2. Sơ đồ usecase**



Hình 2.1. Sơ đồ usecase

**Dưới đây là mô tả chi tiết về từng use case:**

**- Đăng nhập**

- Người dùng nhập số điện thoại và mật khẩu để truy cập hệ thống.
- Hệ thống xác thực thông tin đăng nhập dựa trên dữ liệu có trong cơ sở dữ liệu.
- Nếu thông tin chính xác, hệ thống cấp quyền truy cập và chuyển hướng người dùng đến giao diện chính.
- Nếu thông tin sai, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.

**- Quản lý phòng**

- Người dùng có thể tạo phòng mới bằng cách nhập tên phòng và các thông tin liên quan.
- Người dùng có thể sửa đổi thông tin phòng, bao gồm tên phòng, mô tả,...
- Người dùng có thể xóa phòng nếu trong phòng không còn thiết bị nào được liên kết.
- Khi xóa phòng, hệ thống yêu cầu xác nhận trước khi thực hiện hành động này.

**- Quản lý thiết bị**

- Người dùng có thể thêm thiết bị vào hệ thống bằng cách nhập thông tin thiết bị.
- Khi thêm thiết bị, hệ thống kiểm tra xem đã tồn tại chưa.
- Người dùng có thể sửa đổi thông tin thiết bị, bao gồm trạng thái hoạt động, tên hiển thị.
- Khi người dùng muốn xóa thiết bị, hệ thống sẽ yêu cầu xác nhận trước khi thực hiện.
- Người dùng có thể bật/tắt thiết bị từ giao diện quản lý.
- Người dùng cũng có thể tiến hành điều khiển thiết bị thông qua giọng nói.

**- Quản lý kết nối AccessKey**

- Người dùng có thể nhập mã AccessKey để tiến hành kết nối đến ESP32 có mã tương ứng.

#### **- Quản lý camera**

- Người dùng có thể thêm camera giám sát vào hệ thống bằng cách nhập thông tin kết nối.
- Hệ thống sẽ kiểm tra kết nối và xác nhận xem camera có hoạt động hay không.
- Người dùng có thể xem trực tiếp video từ camera thông qua giao diện hệ thống.

#### **- Quản lý tài khoản**

- Người dùng có thể cập nhật thông tin cá nhân như tên, số điện thoại,...
- Hệ thống yêu cầu xác thực trước khi thay đổi thông tin quan trọng như mật khẩu.
- Người dùng có thể đổi mật khẩu theo các yêu cầu bảo mật (chẳng hạn ít nhất 8 ký tự).
- Nếu nhập sai mật khẩu liên tục tại khi đổi mật khẩu, hệ thống sẽ hiển thị cảnh báo.
- Người dùng có thể thiết lập ảnh đại diện và cập nhật thông tin bảo mật bổ sung.

#### **- Cài đặt hệ thống**

- Người dùng có thể truy cập phần cài đặt để điều chỉnh các thông số chung của hệ thống.
- Cài đặt bao gồm tùy chọn các ngưỡng cảnh báo từ cảm biến hoặc thiết bị.
- Khi thay đổi cài đặt, hệ thống sẽ yêu cầu xác nhận và lưu lại thông tin.
- Một số cài đặt yêu cầu khởi động lại hệ thống để có hiệu lực.

#### **- Quản lý khuôn mặt**

- Người dùng có thể đăng ký khuôn mặt của mình để sử dụng tính năng nhận diện.
- Hệ thống sẽ quét khuôn mặt và lưu trữ dữ liệu sinh trắc học để xác thực danh tính.
- Nếu hệ thống không nhận diện được khuôn mặt, người dùng có thể thử lại

hoặc sử dụng phương thức xác thực khác.

- Người dùng có thể xóa dữ liệu khuôn mặt đã lưu nếu không muốn sử dụng nữa.

#### - Trợ lý ảo

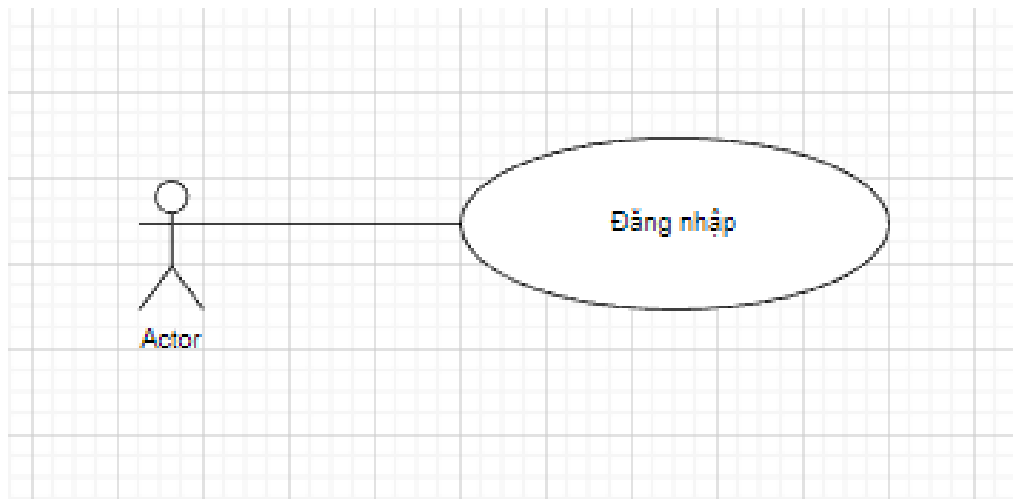
- Hệ thống hỗ trợ trợ lý ảo giúp người dùng có thể trò chuyện, giải đáp thắc mắc
- Ngoài ra còn có trợ lý ảo 3D giúp nâng cao trải nghiệm người dùng

#### - Thông báo

- Hệ thống gửi thông báo cho người dùng khi có sự kiện quan trọng xảy ra, chẳng hạn như nhiệt độ, độ ẩm, khí gas nằm trong ngưỡng cảnh báo.
- Người dùng có thể nhận thông báo trên giao diện hệ thống hoặc qua SMS tùy theo cài đặt.
- Các loại thông báo bao gồm cảnh báo nguy hiểm (rò rỉ khí gas, nhiệt độ cao, độ ẩm), thông báo cập nhật hệ thống, và nhắc nhở bảo trì.

### 2.2.3. Sơ đồ usecase phân rã

#### a) Chức năng đăng nhập

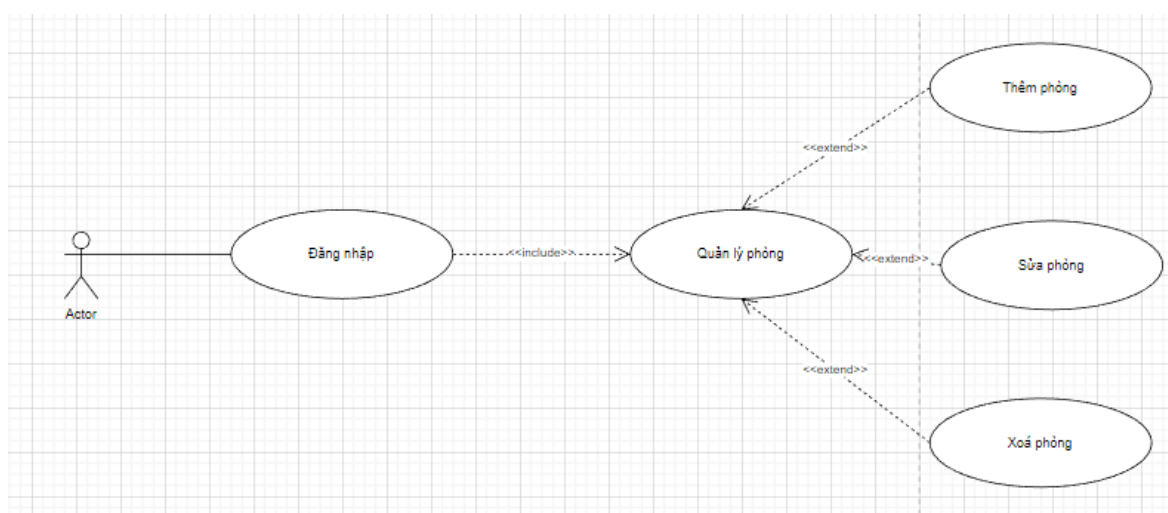


Hình 2.2. Biểu đồ phân rã usecase đăng nhập

Bảng 2.1. Biểu đồ Usecase chức năng đăng nhập

<b>Usecase</b>	Đăng nhập
<b>Mô tả</b>	Đối với các tác nhân có tài khoản hệ thống, cho phép các tác nhân đã đăng nhập vào hệ thống để thực hiện các chức năng bên trong hệ thống
<b>Tác nhân</b>	Người dùng
<b>Tiền điều kiện</b>	Tác nhân truy cập vào giao diện đăng nhập của hệ thống, tác nhân có tài khoản hệ thống.
<b>Hậu điều kiện</b>	Tác nhân đăng nhập thành công vào hệ thống.
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Tác nhân truy cập vào giao diện đăng nhập của hệ thống.</li> <li>2. Hệ thống hiển thị giao diện đăng nhập.</li> <li>3. Tác nhân nhập thông tin .</li> <li>4. Tác nhân chọn “Đăng nhập”.</li> <li>5. Hệ thống kiểm tra thông tin đăng nhập.</li> <li>6. Hệ thống thông báo đăng nhập thành công.</li> <li>7. Hệ thống hiển thị giao diện trang chủ.</li> </ol>
<b>Luồng rẽ nhánh</b>	Hệ thống thông báo lỗi. Tác nhân thực nhập lại thông tin đăng nhập hoặc thoát khỏi hệ thống

**b) Chức năng thêm phòng**

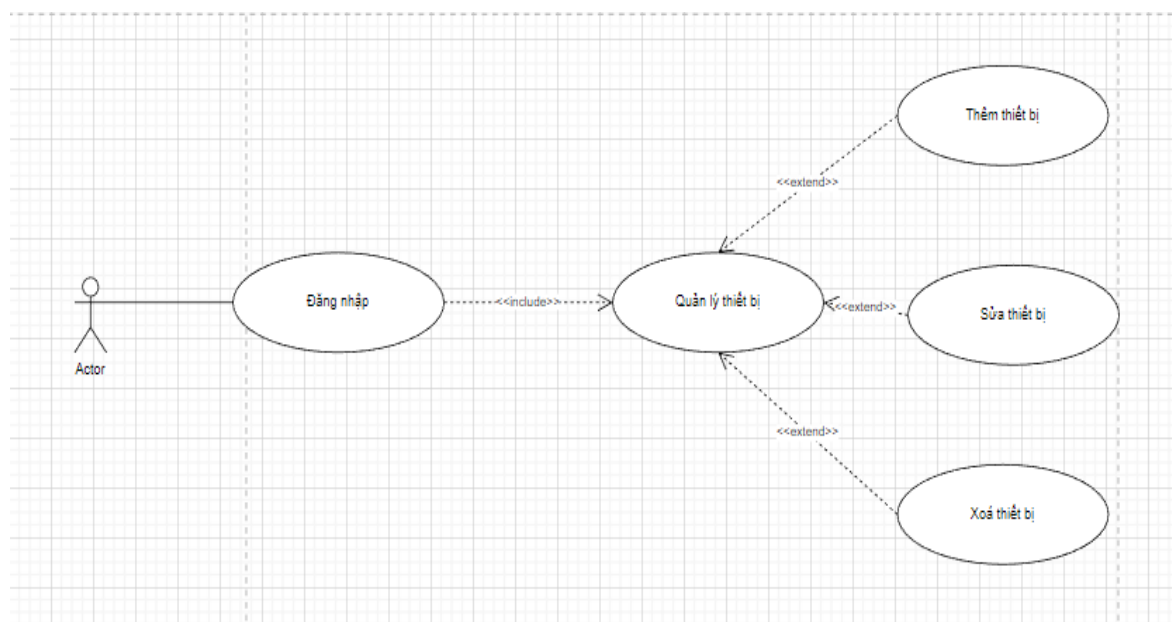


Hình 2.3. Biểu đồ phân rã usecase quản lý phòng

Bảng 2.3. Biểu đồ Usecase chức năng thêm phòng

<b>Usecase</b>	Thêm phòng
<b>Mô tả</b>	Chức năng thêm phòng
<b>Tác nhân</b>	Người dùng
<b>Tiền điều kiện</b>	Tác nhân đăng nhập thành công vào hệ thống, tác nhân chọn chức năng thêm phòng.
<b>Hậu điều kiện</b>	Tác nhân thêm phòng thành công.
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Tác nhân chọn chức năng quản lý phòng.</li> <li>2. Hệ thống hiển thị giao diện danh sách phòng.</li> <li>3. Tác nhân chọn thêm phòng .</li> <li>4. Tác nhân nhập thông tin.</li> <li>5. Hệ thống kiểm tra thông tin.</li> <li>6. Hệ thống thông báo thêm phòng thành công.</li> <li>7. Hệ thống hiển thị giao diện danh sách phòng.</li> </ol>
<b>Luồng rẽ nhánh</b>	Hệ thống thông báo lỗi. Tác nhân thực nhập lại thông tin thêm phòng.

*c) Chức năng thêm thiết bị*

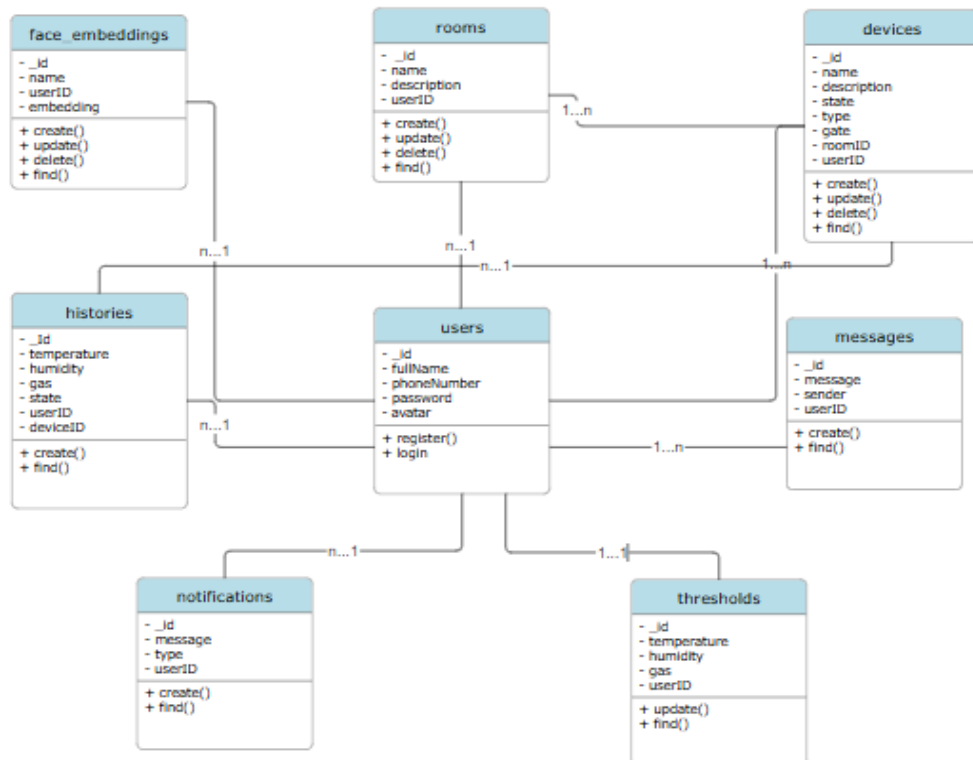


Hình 2.4. Biểu đồ phân rã usecase quản lý thiết bị

Bảng 2.4. Biểu đồ Usecase chức năng thêm thiết bị

<b>Usecase</b>	Thêm thiết bị
<b>Mô tả</b>	Chức năng thêm thiết bị
<b>Tác nhân</b>	Người dùng
<b>Tiền điều kiện</b>	Tác nhân đăng nhập thành công vào hệ thống, tác nhân chọn chức năng thêm thiết bị
<b>Hậu điều kiện</b>	Tác nhân thêm thiết bị thành công.
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Tác nhân chọn chức năng quản lý thiết bị.</li> <li>2. Hệ thống hiển thị giao diện danh sách thiết bị.</li> <li>3. Tác nhân chọn thêm thiết bị .</li> <li>4. Tác nhân nhập thông tin.</li> <li>5. Hệ thống kiểm tra thông tin.</li> <li>6. Hệ thống thông báo thêm thiết bị thành công.</li> <li>7. Hệ thống hiển thị giao diện danh sách thiết bị.</li> </ol>
<b>Luồng rẽ nhánh</b>	Hệ thống thông báo lỗi. Tác nhân thực nhập lại thông tin thêm thiết bị.

#### 2.2.4. Sơ đồ lớp

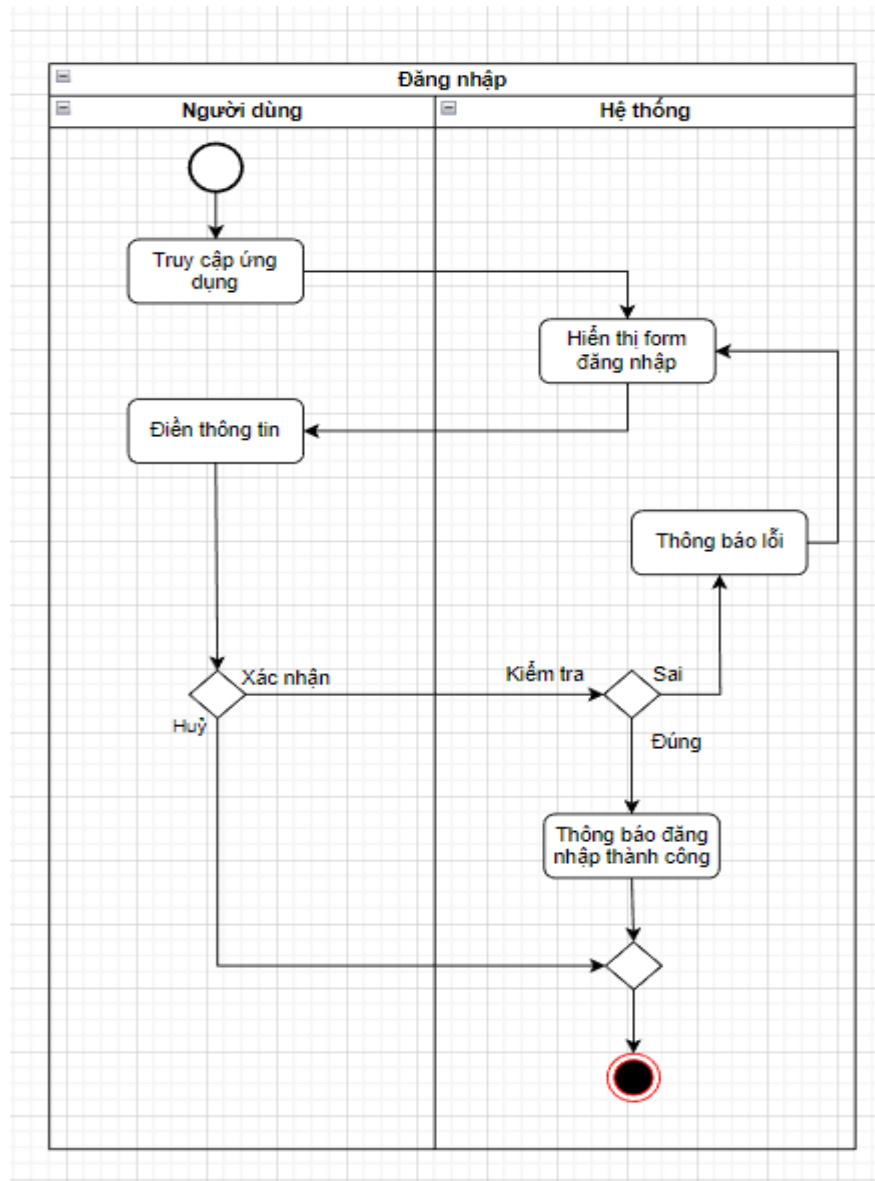


Hình 2.5. Sơ đồ lớp



### 2.2.5. Sơ đồ hoạt động

#### a) Chức năng đăng nhập

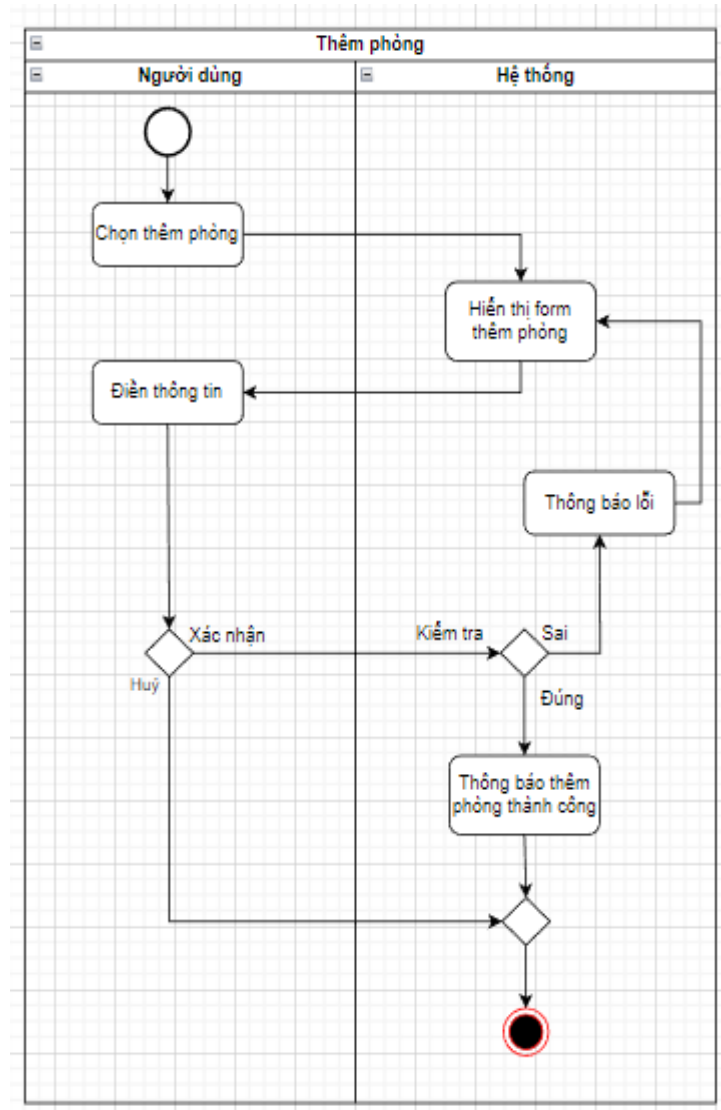


Hình 2.6. Biểu đồ hoạt động chức năng đăng nhập

#### Mô tả:

- Người dùng truy cập ứng dụng.
- Hệ thống hiển thị form đăng nhập và người dùng điền thông tin.
- Hệ thống kiểm tra thông tin đăng nhập:
  - Nếu thông tin sai, hệ thống thông báo lỗi, người dùng phải nhập lại.
  - Nếu thông tin đúng, hệ thống thông báo đăng nhập thành công.
- Quá trình đăng nhập kết thúc.

***b) Chức năng thêm phòng***

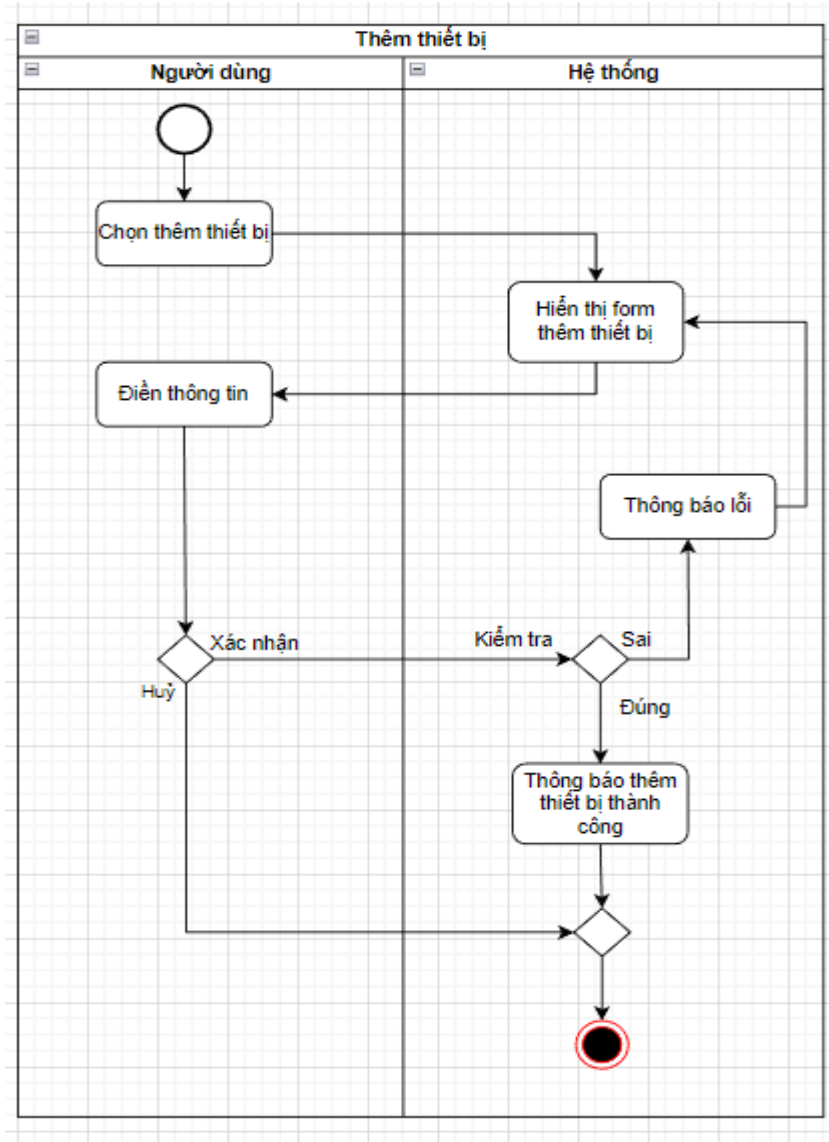


*Hình 2.7. Biểu đồ hoạt động chức năng thêm phòng*

**Mô tả:**

- Người dùng chọn thêm phòng.
- Hệ thống hiển thị form thêm phòng để người dùng nhập thông tin.
- Người dùng điền thông tin và nhấn Xác nhận.
- Hệ thống kiểm tra thông tin:
  - Nếu thông tin sai, hệ thống thông báo lỗi, người dùng phải nhập lại.
  - Nếu thông tin đúng, hệ thống thông báo thêm phòng thành công.
- Quá trình thêm phòng kết thúc.

**c) Chức năng thêm thiết bị**

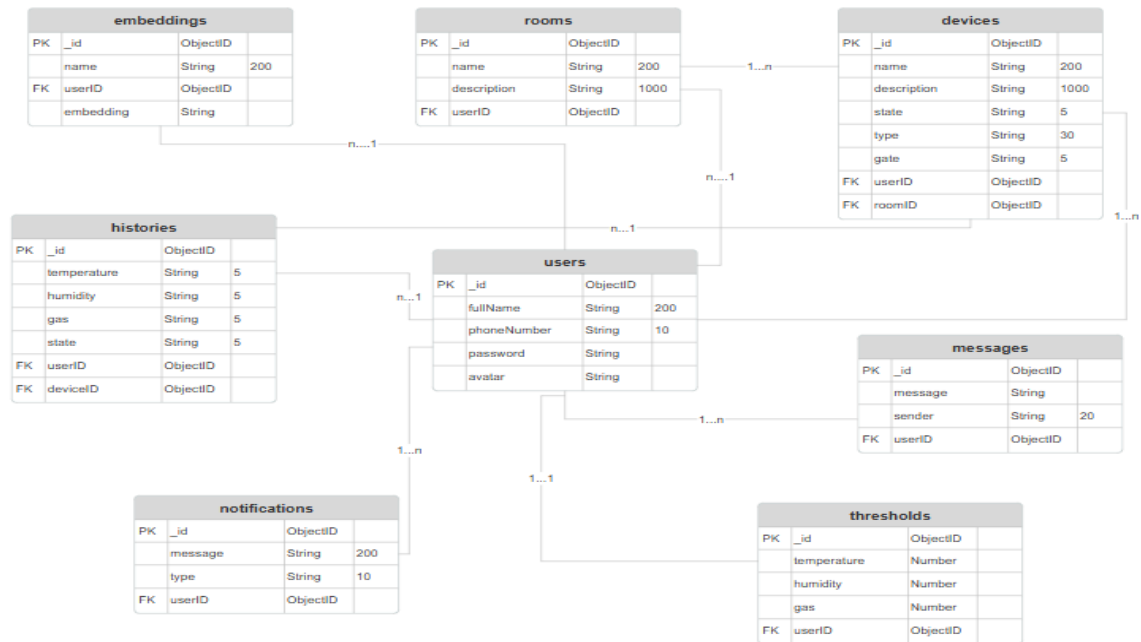


Hình 2.8. Biểu đồ hoạt động chức năng thêm thiết bị

**Mô tả:**

- Người dùng chọn thêm thiết bị.
- Hệ thống hiển thị form thêm thiết bị để người dùng nhập thông tin.
- Người dùng điền thông tin và nhấn Xác nhận.
- Hệ thống kiểm tra thông tin:
  - Nếu thông tin sai, hệ thống thông báo lỗi, người dùng phải nhập lại.
  - Nếu thông tin đúng, hệ thống thông báo thêm thiết bị thành công.
- Quá trình thêm thiết bị kết thúc.

### 2.1.6. Thiết kế Database



Hình 2.9. Sơ đồ ERD

#### c) Chi tiết các bảng

##### - Bảng Users:

Bảng 2.5. Bảng Users

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id người dùng		OBJECTID	
fullName	Tên đầy đủ		STRING	200
phoneNumber	Số điện thoại người dùng		STRING	10
password	Mật khẩu người dùng		STRING	100
avatar	Ảnh đại diện	x	STRING	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Rooms:**

*Bảng 2.6. Bảng Rooms*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id phòng		OBJECTID	
name	Tên phòng		STRING	200
description	Mô tả phòng	x	STRING	1000
userID	Id người tạo phòng		OBJECTID	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Devices:**

*Bảng 2.7. Bảng Devices*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id thiết bị		OBJECTID	
name	Tên thiết bị		STRING	200
description	Mô tả thiết bị	x	STRING	1000
state	Trạng thái thiết bị		STRING	5
roomID	Id phòng		OBJECTID	
type	Kiểu thiết bị		STRING	30
gate	Cổng của thiết bị		STRING	5
userID	Id người tạo thiết bị		OBJECTID	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Messages:**

*Bảng 2.8. Bảng Messages*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id tin nhắn		OBJECTID	
message	Tin nhắn		STRING	12
sender	Loại người gửi		STRING	20
userID	Id người gửi		OBJECTID	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Notification:**

*Bảng 2.9. Bảng Notifications*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id thông báo		OBJECTID	
message	Tin nhắn thông báo		STRING	200
deviceID	Id thiết bị thông báo		OBJECTID	
type	Kiểu thông báo		STRING	10
userID	Id người dùng		OBJECTID	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Histories:**

*Bảng 2.10. Bảng Histories*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id phòng		OBJECTID	
deviceId	Id thiết bị		OBJECTID	
temperature	Giá trị nhiệt độ	x	STRING	5
humidity	Giá trị độ ẩm	x	STRING	5
gas	Giá trị khí gas	x	STRING	5
state	Trạng thái thiết bị	x	STRING	10
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Face\_Embeddings:**

*Bảng 2.11. Bảng Face\_Embeddings*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id khuôn mặt		OBJECTID	
embedding	Đặc trưng khuôn mặt		ARRAY	
name	Tên		STRING	200
userID	Id người sở hữu		OBJECTID	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**- Bảng Thresholds:**

*Bảng 2.12. Bảng Thresholds*

Tên	Mô tả	Cho phép null	Kiểu dữ liệu	Độ dài
_id	Id ngưỡng		OBJECTID	
temperature	Nhiệt độ		NUMBER	
humidity	Độ ẩm		NUMBER	
gas	Khí gas		NUMBER	
userID	Id người dùng		OBJECTID	
createdAt	Thời gian tạo		TIMESTAMP	
updatedAt	Thời gian cập nhật		TIMESTAMP	

**2.1.7. Bảo mật hệ thống**

Hệ thống nhà thông minh tích hợp IoT và AI mang lại sự tiện nghi nhưng cũng tiềm ẩn nhiều nguy cơ về bảo mật, bao gồm tấn công mạng, rò rỉ dữ liệu và chiếm quyền điều khiển thiết bị. Để đảm bảo an toàn, hệ thống áp dụng các biện pháp bảo mật nghiêm ngặt.

**Các giải pháp bảo mật trong hệ thống**

**- Mã hóa dữ liệu**

- Toàn bộ dữ liệu trao đổi giữa **client – server – thiết bị IoT** được mã hóa bằng **HTTPS** và **WebSocket Secure (WSS)** nhằm ngăn chặn việc đánh cắp thông tin.
- Hình ảnh nhận diện khuôn mặt từ ESP32-CAM được mã hóa **AES-256** trước khi gửi đến server để xử lý.



- **Xác thực người**
  - Hệ thống chỉ cho phép người dùng **đã xác thực** điều khiển thiết bị.
- **Bảo mật thiết bị IoT**
  - Mỗi ESP 32 hoặc ESP32 CAM được cấp một **Access Key duy nhất**, giúp xác minh nguồn gốc và ngăn chặn truy cập trái phép.
  - Thiết bị IoT được kiểm tra định kỳ để phát hiện các lỗi hỏng bảo mật.

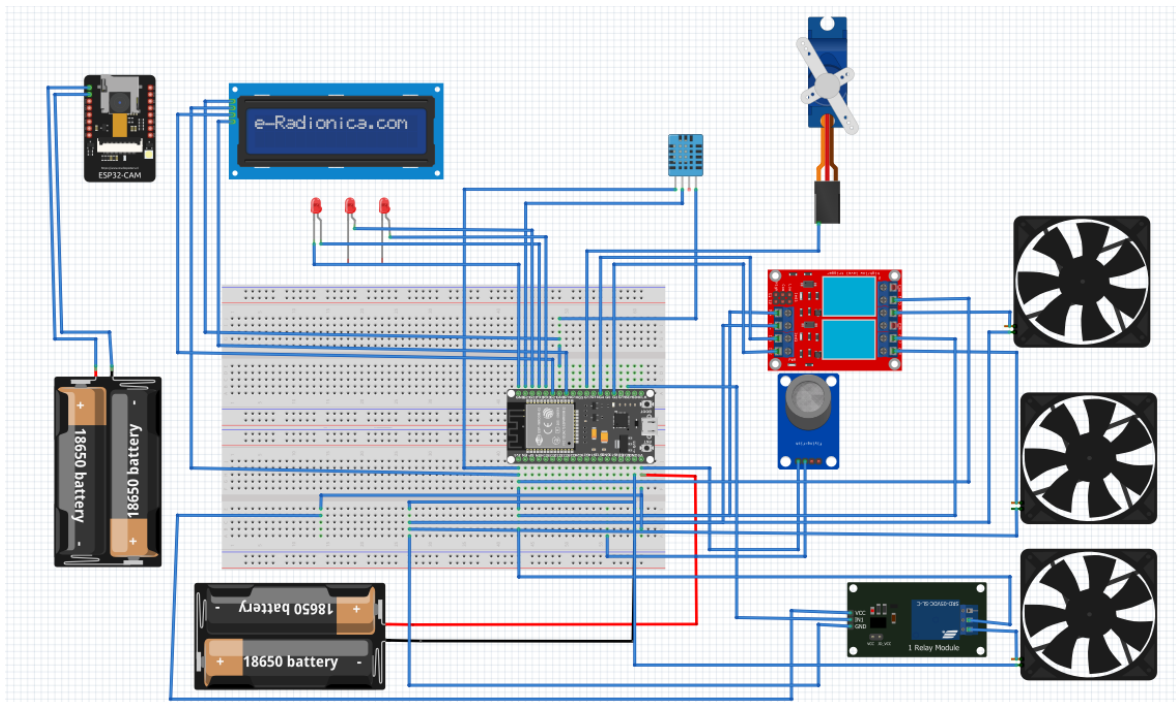
### 2.1.7. Giao tiếp ứng dụng di động với server

Ứng dụng di động giao tiếp với server thông qua các giao thức sau:

- **API RESTful**: Xử lý yêu cầu HTTP để thực hiện các thao tác như đăng nhập, đăng ký, truy xuất thông tin phòng, thiết bị và một số thao tác không yêu cầu cập nhật theo thời gian thực.
- **WebSocket**: Dùng để **truyền dữ liệu thời gian thực**, giúp phản hồi nhanh chóng khi điều khiển thiết bị hoặc nhận sự kiện từ cảm biến, đồng thời giúp truyền video từ ESP32 CAM theo thời gian thực giúp người dùng dễ dàng xem xét các tình huống hoặc nhận diện khuôn mặt theo thời gian thực.

## 2.3. Phần cứng

### 2.3.1. Sơ đồ lắp ráp mạch



Hình 2.11. Sơ đồ lắp ráp mạch

### 2.3.2. Các thư viện được dùng

Bảng 2.13. Bảng thư viện dùng cho phần cứng

STT	Tên các thư viện dùng trong Arduino	Chức năng của từng thư viện	Thiết bị sử dụng thư viện
2	<WebSocketsClient.h>	- Thư viện này giúp ESP32 hoặc các thiết bị hỗ trợ khác có thể giao tiếp với server thông qua giao thức WebSocket. WebSocket là một giao thức liên lạc hai chiều (full-duplex) giúp việc truyền tải dữ liệu trở nên nhanh chóng và liên tục mà không cần phải liên tục mở và đóng kết nối như HTTP. Bạn có thể sử dụng nó để gửi/nhận dữ liệu giữa server và thiết bị theo thời gian thực.	- ESP32, ESP32 CAM AI THINKER
3	<ArduinoJson.h>	- Thư viện này được sử dụng để phân tích (parse) và tạo dữ liệu JSON trong các ứng dụng Arduino. JSON là định dạng dữ liệu phổ biến cho việc trao đổi thông tin giữa các hệ thống, đặc biệt trong các ứng dụng IoT. ArduinoJson hỗ trợ tạo JSON từ các giá trị Arduino và ngược lại, phân tích dữ liệu JSON nhận được từ các thiết bị khác hoặc từ server.	- Không yêu cầu thiết bị phần cứng cụ thể, thường dùng khi cần trao đổi dữ liệu dạng JSON.
4	<DHT.h>	- Thư viện này hỗ trợ đọc dữ liệu từ các cảm biến DHT (bao gồm DHT11, DHT22,	- Cảm biến DHT11

		AM2302). Các cảm biến này được sử dụng để đo nhiệt độ và độ ẩm. Thư viện cung cấp các phương thức để đọc giá trị nhiệt độ (C/F) và độ ẩm (%).	
5	<Wire.h>	- Thư viện Wire hỗ trợ giao tiếp I2C (Inter-Integrated Circuit), một giao thức truyền thông nối tiếp giữa Arduino và các thiết bị khác như cảm biến, màn hình LCD, EEPROM, v.v. Nó cho phép Arduino điều khiển các thiết bị I2C thông qua các chân SDA (data) và SCL (clock).	- Các thiết bị sử dụng giao thức I2C
6	<LiquidCrystal_I2C.h>	- Thư viện này cung cấp các hàm để điều khiển màn hình LCD sử dụng giao thức I2C. Nó hỗ trợ hiển thị văn bản, số và các ký tự khác trên màn hình LCD thông qua giao tiếp I2C, giúp giảm số chân cần sử dụng trên Arduino.	- Màn hình LCD với module I2C
7	<ESP32Servo.h>	- Thư viện này dùng để điều khiển động cơ servo, cung cấp các hàm để điều khiển vị trí góc của servo (thường trong khoảng từ 0 đến 180 độ). Nó hỗ trợ việc điều chỉnh chính xác vị trí của trục động cơ thông qua tín hiệu PWM.	- Động cơ servo như SG90

## CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ THỬ NGHIỆM

### 3.1. Yêu cầu môi trường triển khai và các bước triển khai

#### 3.1.1. Yêu cầu môi trường triển khai

Hệ thống được triển khai trên nền tảng Android và hoạt động theo mô hình Client-Server. Hệ thống gồm hai server, một chạy Node.js và một chạy Python, phục vụ các chức năng khác nhau, đồng thời đảm bảo hiệu suất và khả năng mở rộng linh hoạt. Các yêu cầu về môi trường triển khai cho hệ thống bao gồm:

#### - Phía Client (Thiết bị Android):

- **Hệ điều hành:** Thiết bị Android cần chạy hệ điều hành Android 7.0 trở lên để đảm bảo tương thích với ứng dụng và các tính năng mới nhất.
- **Kết nối internet:** Thiết bị cần có kết nối internet ổn định để thực hiện các giao dịch với server và nhận các phản hồi từ hệ thống.
- **Cài đặt ứng dụng:** Ứng dụng sẽ được cài đặt từ file APK để đơn giản hóa quá trình triển khai trên các thiết bị Android khác nhau.

#### - Phía Server:

- **Server 1 - Backend Node.js (Triển khai trên Render):**
  - **Hệ điều hành:** Đảm bảo hệ điều hành là Ubuntu 22.04 hoặc Windows Server 2019 trở lên.
  - **Công nghệ:** Node.js v16+ cho phần backend, MongoDB cho cơ sở dữ liệu, và WebSocket để hỗ trợ giao tiếp thời gian thực.
  - **Triển khai:** Server này được triển khai trên nền tảng Render, đảm bảo khả năng mở rộng tự động và tối ưu hóa hiệu suất khi có tải cao.
- **Server 2 - Python (Xử lý nhận diện khuôn mặt, triển khai trên AWS):**
  - **Hệ điều hành:** Server sử dụng Ubuntu 22.04 hoặc Windows Server 2019.
  - **Công nghệ:** Python 3.10+ cùng các thư viện chuyên biệt cho việc nhận diện khuôn mặt như OpenCV hoặc dlib.
  - **Triển khai:** Dịch vụ Python này được triển khai trên AWS EC2, tận dụng sức mạnh hạ tầng của AWS để đảm bảo khả năng xử lý nhanh chóng và khả năng mở rộng cao.

### 3.1.2. Các bước triển khai

#### - Triển khai phía Server

##### **Bước 1: Triển khai Server Node.js trên Render**

1. **Đẩy mã nguồn lên GitHub** và thiết lập kết nối giữa GitHub và Render để dễ dàng quản lý và triển khai mã nguồn.
2. **Tạo Web Service** trên Render để triển khai ứng dụng và quản lý các kết nối WebSocket giữa client và server.
3. **Cấu hình biến môi trường** cùng các thông số liên quan đến database, API và các lệnh cần thiết cho ứng dụng Node.js.
4. **Deploy và kiểm tra kết nối** giữa client và server Node.js để đảm bảo ứng dụng hoạt động ổn định và có thể xử lý các yêu cầu từ người dùng.

##### **Bước 2: Triển khai Server Python trên AWS**

1. **Chọn dịch vụ AWS EC2** và cài đặt môi trường Python 3.10+, thiết lập các thư viện nhận diện khuôn mặt cần thiết như OpenCV, dlib.
2. **Cài đặt các thư viện** và các công cụ hỗ trợ cho việc nhận diện khuôn mặt, xử lý ảnh và gửi phản hồi về client.
3. **Cấu hình Nginx** làm Reverse Proxy để chuyển tiếp các yêu cầu từ client đến đúng dịch vụ nhận diện khuôn mặt.
4. **Cài đặt SSL với Let's Encrypt** để đảm bảo mọi giao dịch qua mạng đều được mã hóa và bảo mật.
5. **Khởi động và kiểm tra** dịch vụ Python để xác minh rằng server có thể xử lý các yêu cầu đúng cách và cung cấp các dịch vụ nhận diện khuôn mặt chính xác.

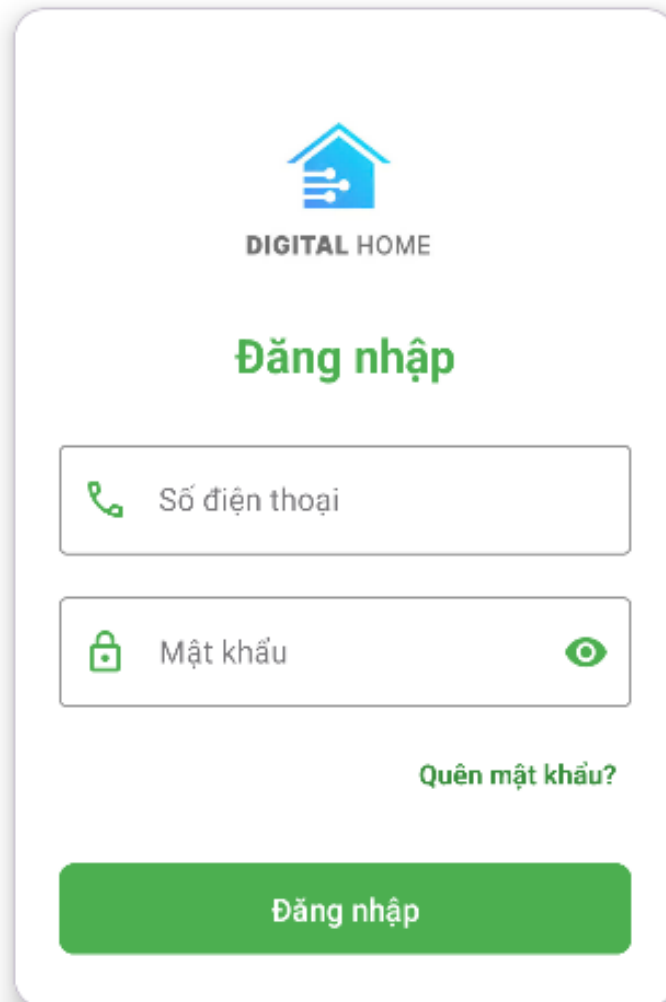
#### - Triển khai phía Client

1. **Biên dịch ứng dụng Android** từ mã nguồn .
2. **Xuất file APK** và cài đặt trên các thiết bị Android thực tế để thử nghiệm với nhiều loại thiết bị khác nhau, đảm bảo tính tương thích.
3. **Kiểm tra kết nối** và các chức năng giữa ứng dụng Android và cả hai server (Node.js và Python), đảm bảo hệ thống hoạt động ổn định.

## 3.2. Giao diện và lập trình chức năng phần mềm

### 3.2.1. Đăng nhập

- Giao diện:



Hình 3.1. Giao diện đăng nhập

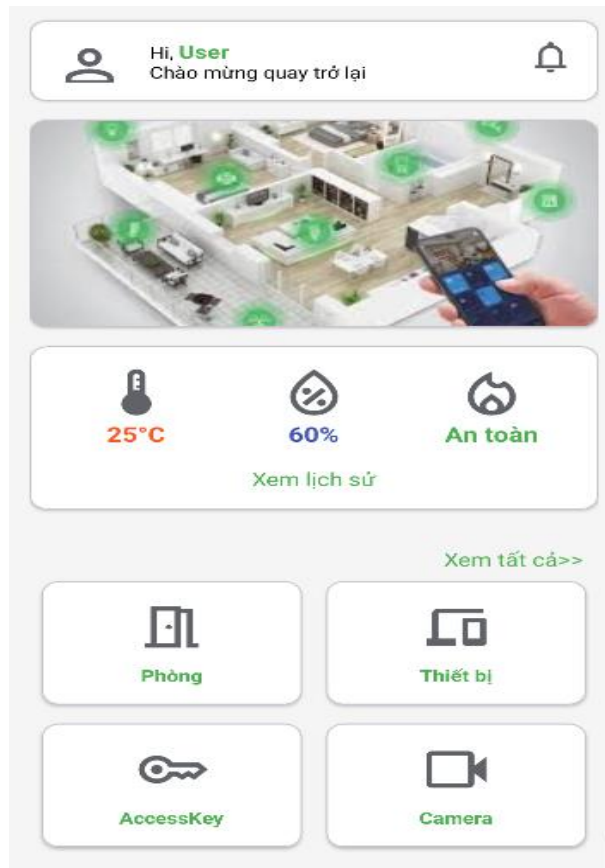
- Mô tả:

Giao diện đăng nhập bao gồm hai trường:

- **phoneNumber:** Người dùng nhập số điện thoại để xác định tài khoản.
- **password:** Người dùng nhập mật khẩu để xác thực.

Sau khi điền đủ thông tin, người dùng nhấn nút đăng nhập để truy cập vào hệ thống. Ngoài ra, hệ thống còn cung cấp thêm 2 phương thức đăng nhập qua vân tay hoặc FaceID nếu người dùng kích hoạt.

### 3.2.2. Giao diện trang chủ



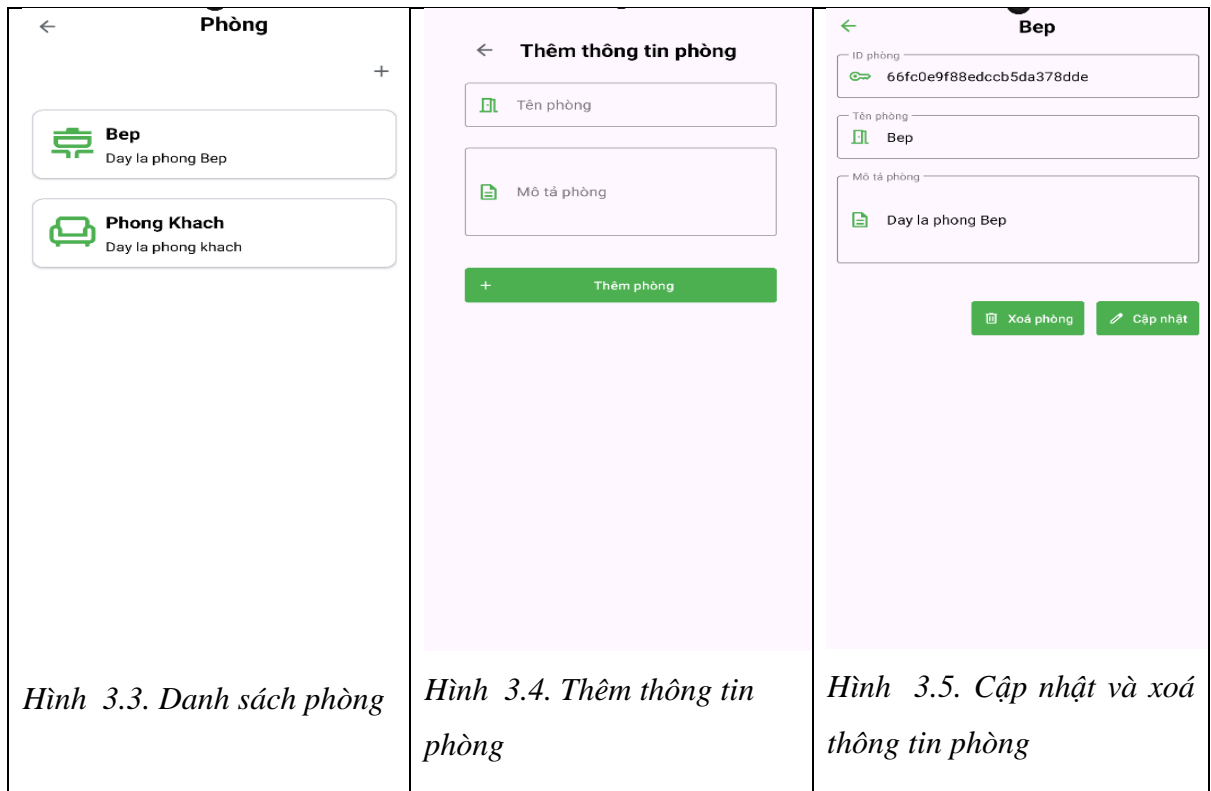
Hình 3.2. Giao diện trang chủ

#### - Mô tả:

Giao diện này là trang chính của hệ thống giám sát và quản lý thiết bị, có các thành phần chính sau:

- Thông tin giám sát:
  - Hiển thị nhiệt độ (°C), độ ẩm (%), và trạng thái an toàn của môi trường.
  - Màu sắc của các chỉ số giúp người dùng dễ dàng nhận biết: màu đỏ cho nhiệt độ, màu xanh cho độ ẩm và màu xanh lá cho trạng thái an toàn.
  - Có liên kết "Xem lịch sử" để người dùng xem lại dữ liệu giám sát.
- Hiển thị một số chức năng:
  - Phòng: Quản lý và theo dõi các phòng trong hệ thống.
  - Thiết bị: Quản lý và theo dõi các thiết bị trong hệ thống.
  - AccessKey: Quản lý AccessKey.
  - Camera: Quản lý camera.

### 3.2.3. Giao diện quản lý phòng



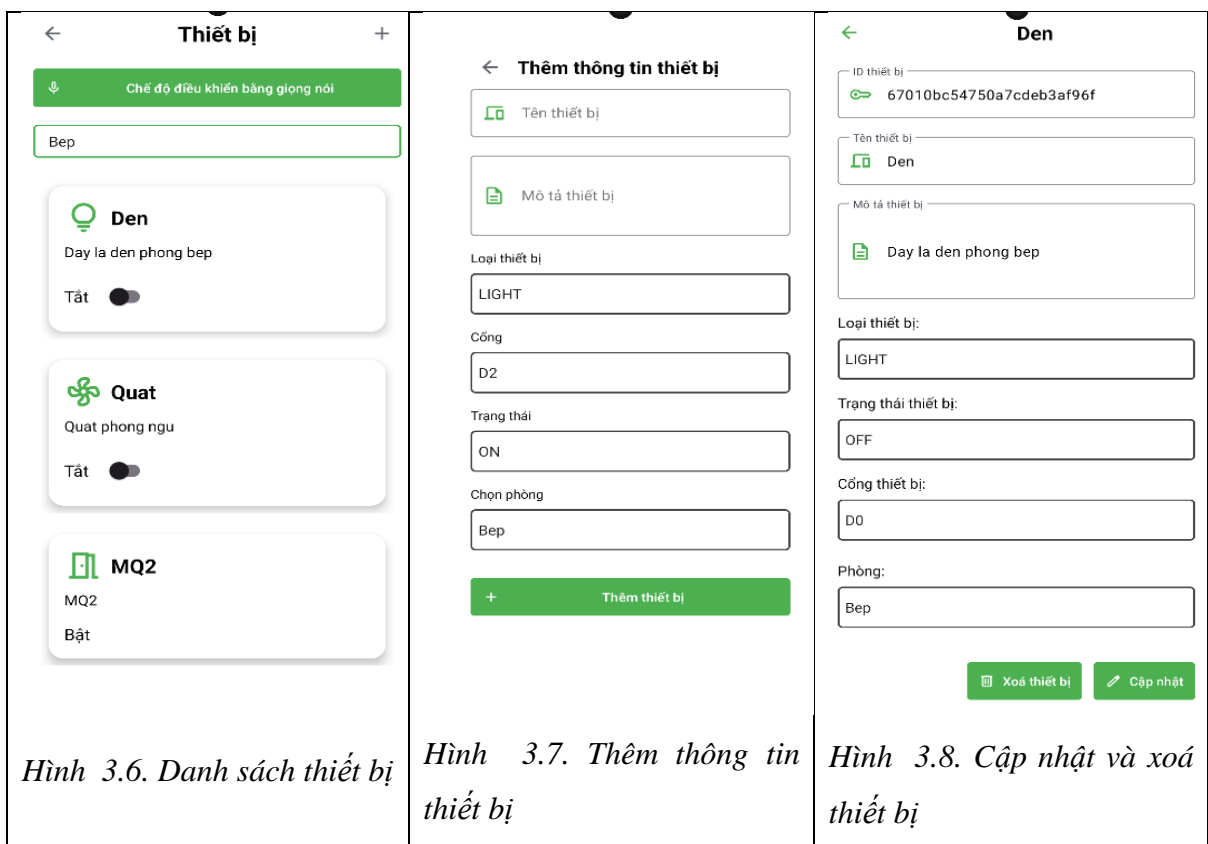
#### - Mô tả:

- Giao diện Danh sách phòng (Hình 3.3):
  - Hiện thị tổng quan tất cả các phòng trong hệ thống
  - Cho phép người dùng xem nhanh các phòng hiện có
  - Có nút "+" để thêm phòng mới
  - Cho phép chọn vào từng phòng để xem chi tiết/chỉnh sửa
  - Mỗi phòng được hiển thị với icon đặc trưng để dễ phân biệt (ví dụ: icon bếp, icon ghế sofa cho phòng khách)
- Giao diện Thêm thông tin phòng (Hình 3.4):
  - Cho phép người dùng tạo phòng mới
  - Có form nhập liệu với 2 trường thông tin:
    - Tên phòng: đặt tên cho phòng mới
    - Mô tả phòng: thêm thông tin chi tiết về phòng
  - Nút "Thêm phòng" để lưu thông tin phòng mới vào hệ thống



- Giao diện Cập nhật và xoá thông tin phòng (Hình 3.5):
  - Hiện thị chi tiết thông tin của một phòng cụ thể
  - Cho phép xem ID phòng
  - Có thể chỉnh sửa tên và mô tả phòng
  - Cung cấp 2 chức năng chính:
    - Nút "Cập nhật": lưu các thay đổi thông tin
    - Nút "Xoá phòng": xoá phòng khỏi hệ thống

### 3.2.4. Giao diện quản lý thiết bị



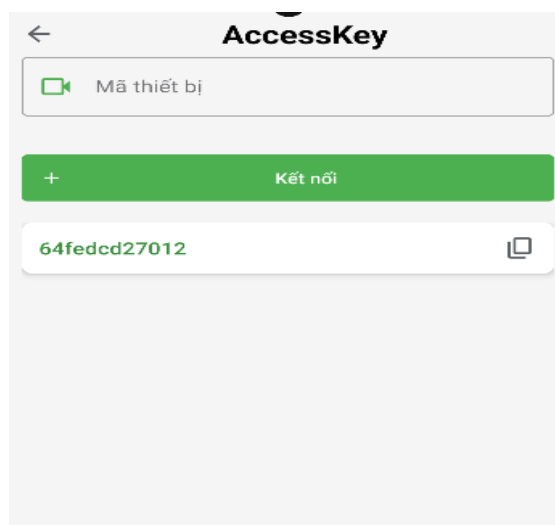
#### - Mô tả:

Ba giao diện này liên quan đến quản lý thiết bị:

- Hình 3.6 (Danh sách thiết bị):
  - Hiện thị danh sách các thiết bị
  - Mỗi thiết bị có biểu tượng riêng, tên và mô tả
  - Có công tắc bật/tắt (Tắt/Bật) cho từng thiết bị
  - Nút "Chế độ điều khiển bằng giọng nói" màu xanh lá

- Hình 3.7 (Thêm thông tin thiết bị):
  - Form nhập thông tin thiết bị mới với các trường:
    - Tên thiết bị
    - Mô tả thiết bị
    - Loại thiết bị
    - Cổng
    - Trạng thái
    - Chọn phòng
  - Nút "Thêm thiết bị" ở dưới cùng
- Hình 3.8 (Cập nhật và xoá thiết bị):
  - Hiển thị chi tiết một thiết bị với:
    - ID thiết bị
    - Tên thiết bị
    - Mô tả thiết bị
    - Loại thiết bị
    - Trạng thái
    - Cổng thiết bị
    - Phòng
  - Hai nút chức năng: "Xoá thiết bị" và "Cập nhật"

### 3.2.5. Quản lý AccessKey



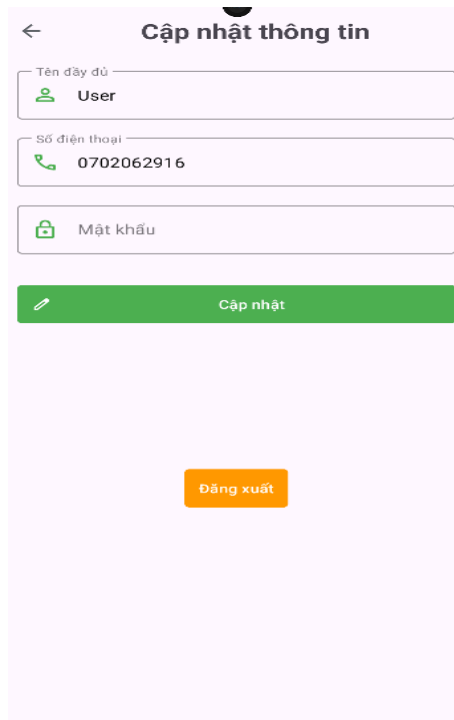
Hình 3.9. Giao diện quản lý AccessKey

**- Mô tả:**

Giao diện này là trang quản lý AccessKey của hệ thống, có các thành phần chính sau:

- Textbox nhập accesskey để kết nối tới ESP32 tương ứng.
- Danh sách AccessKey: Hiển thị AccessKey đã kết nối.

**3.2.6. Giao diện quản lý tài khoản**



*Hình 3.10. Giao diện quản lý tài khoản*

**- Mô tả:**

Giao diện này là trang Cập nhật thông tin của người dùng, với các thành phần chính sau:

- Thông tin cá nhân:
  - Tên đầy đủ: Trường nhập để người dùng cập nhật tên hiển thị của mình.
  - Số điện thoại: Trường nhập số điện thoại, dùng để xác định tài khoản của người dùng.
  - Mật khẩu: Trường nhập mật khẩu mới để người dùng có thể cập nhật hoặc thay đổi.
- Nút "Cập nhật": Nút màu xanh lá cho phép người dùng lưu lại các thay đổi sau

khi cập nhật thông tin.

- Nút "Đăng xuất": Nút màu cam cho phép người dùng đăng xuất khỏi tài khoản.

### 3.2.7. Giao diện cài đặt hệ thống



← Cài đặt hệ thống

Ngưỡng nhiệt độ (°C)

10.0 40.0

Ngưỡng độ ẩm (%)

30.0 90.0

Ngưỡng khí gas (ppm)

500.0

Cập nhật

Hình 3.11. Giao diện cài đặt hệ thống

#### - Mô tả:

- Đây là giao diện **Cài đặt hệ thống**, cho phép người dùng thiết lập và điều chỉnh các ngưỡng cảnh báo quan trọng liên quan đến **hiệt độ, độ ẩm và khí gas**. Khi giá trị đo thực tế vượt quá giới hạn được thiết lập, hệ thống sẽ kích hoạt cảnh báo để đảm bảo an toàn.
- Giao diện được thiết kế theo bố cục đơn giản, trực quan, với mỗi thông số hiển thị theo hàng ngang. **Giá trị tối thiểu** nằm bên trái, **giá trị tối đa** bên phải, giúp người dùng dễ dàng theo dõi và nhập dữ liệu. Phong nền trắng với chữ đen đảm bảo tính rõ ràng và dễ đọc.
- Phía dưới cùng là nút "**Cập nhật**" màu xanh lá, đi kèm biểu tượng chỉnh sửa. Khi nhấn vào, hệ thống sẽ lưu lại các thay đổi và áp dụng ngay các ngưỡng mới vào quá trình giám sát.

### 3.2.8. Giao diện quản lý khuôn mặt



Hình 3.12. Giao diện danh sách khuôn mặt

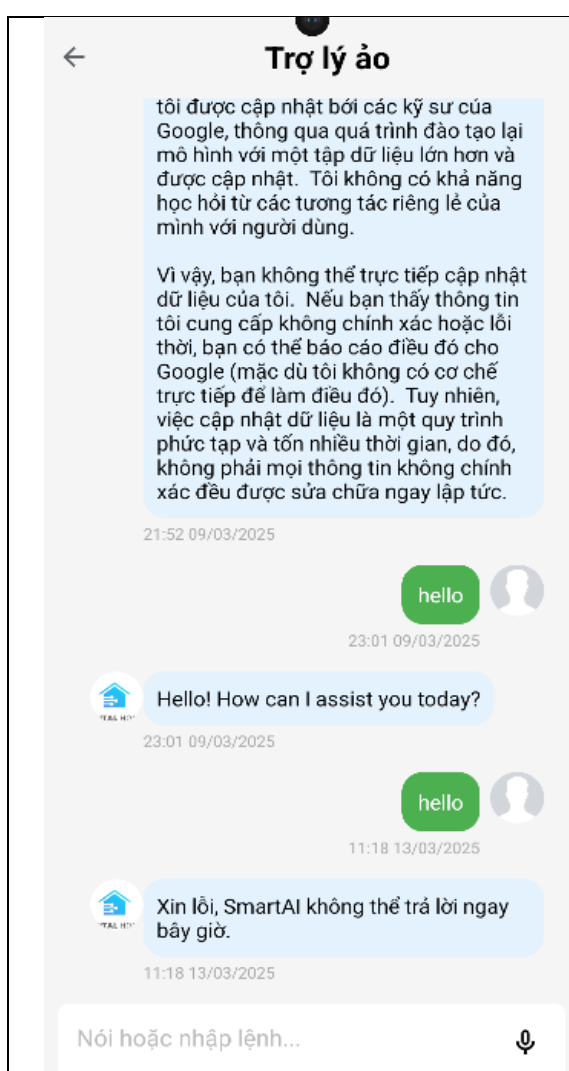
Hình 3.13. Giao diện thêm khuôn mặt

#### - Mô tả:

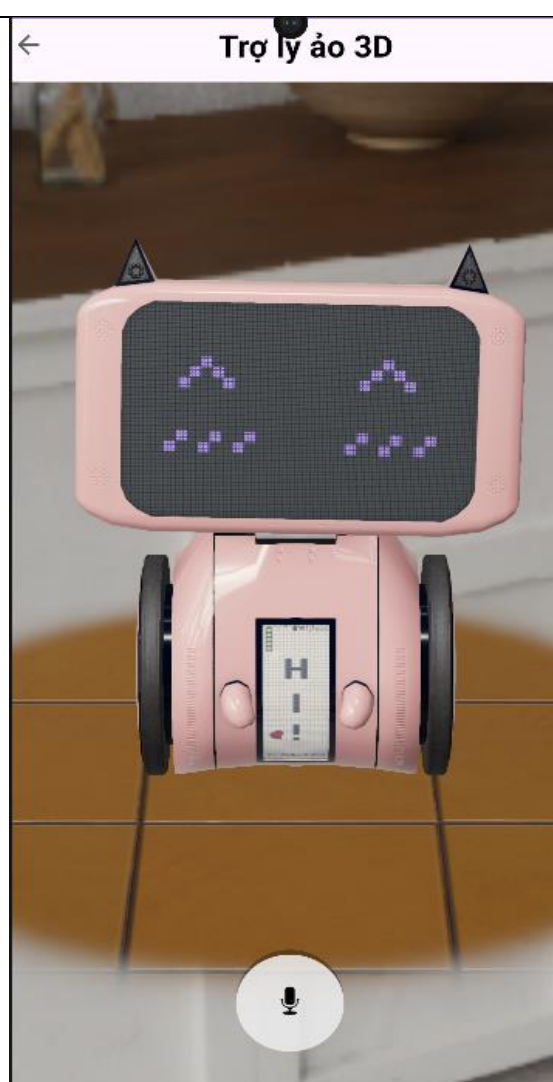
- Hình 3.12 (Giao diện danh sách khuôn mặt)
  - Hiện thị danh sách các khuôn mặt đã được nhận diện, bao gồm ID (chuỗi mã nhận diện), Tên (nếu đã đặt) và các thao tác.
  - Người dùng có thể chỉnh sửa tên khuôn mặt bằng cách nhấn vào biểu tượng cây bút hoặc xóa khuôn mặt bằng biểu tượng thùng rác.
  - Nếu khuôn mặt chưa được đặt tên, hệ thống có thể hiển thị tên mặc định như "unknown".
  - Nút thêm khuôn mặt mới (+) ở góc trên giúp người dùng thêm khuôn mặt mới vào hệ thống.
- Hình 3.13 (Giao diện thêm khuôn mặt)
  - Hiện thị ảnh khuôn mặt mà hệ thống đang cố nhận diện. Nếu không nhận diện được, sẽ có thông báo lỗi “Không phát hiện khuôn mặt”.

- Yêu cầu người dùng điều chỉnh góc nhìn (Nhìn thẳng, Quay trái, Quay phải) để tăng độ chính xác khi nhận diện. Nếu các góc chưa đạt yêu cầu, hệ thống sẽ hiển thị dấu x.
- Người dùng có thể tải ảnh lên nếu hệ thống không thể quét được khuôn mặt từ camera.
- Mục đích của giao diện này là giúp hệ thống thu thập dữ liệu khuôn mặt để sử dụng trong các chức năng nhận diện sau này.

### 3.2.9. Giao diện trợ lý ảo



Hình 3.14. Giao diện trò chuyện văn bản



Hình 3.15. Giao diện trợ lý ảo 3D

## Mô tả:

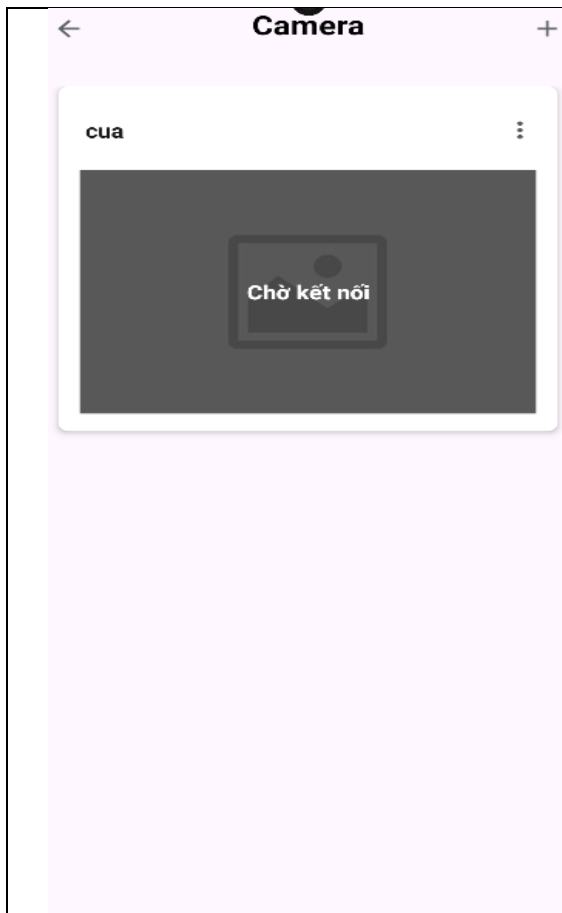
Hình 3.14 (Giao diện trò chuyện văn bản - Chatbot AI)

- Giao diện trò chuyện văn bản cho phép người dùng giao tiếp với trợ lý ảo qua tin nhắn văn bản. Người dùng có thể dễ dàng gửi câu hỏi hoặc yêu cầu trợ giúp, và AI sẽ phản hồi nhanh chóng.
- Trợ lý AI phản hồi dưới dạng văn bản và có khả năng hỗ trợ nhiều ngôn ngữ, giúp người dùng từ các quốc gia khác nhau có thể dễ dàng tương tác với hệ thống.
- Tuy nhiên, trong một số trường hợp, AI có thể không thể phản hồi ngay lập tức nếu hệ thống gặp lỗi hoặc không tìm thấy dữ liệu phù hợp với yêu cầu của người dùng.
- Hệ thống hỗ trợ nhiều ngôn ngữ, bao gồm tiếng Anh và tiếng Việt, giúp đáp ứng nhu cầu của người dùng đa dạng về ngôn ngữ.
- Khi AI không thể trả lời câu hỏi hoặc gặp sự cố, hệ thống sẽ thông báo rõ ràng về lỗi hoặc ngoại lệ, đảm bảo người dùng biết được tình trạng hiện tại của hệ thống.
- Giao diện này có khả năng lưu trữ lịch sử trò chuyện, giúp AI hiểu rõ hơn về ngữ cảnh cuộc trò chuyện, từ đó cung cấp các phản hồi chính xác và phù hợp hơn trong tương lai.

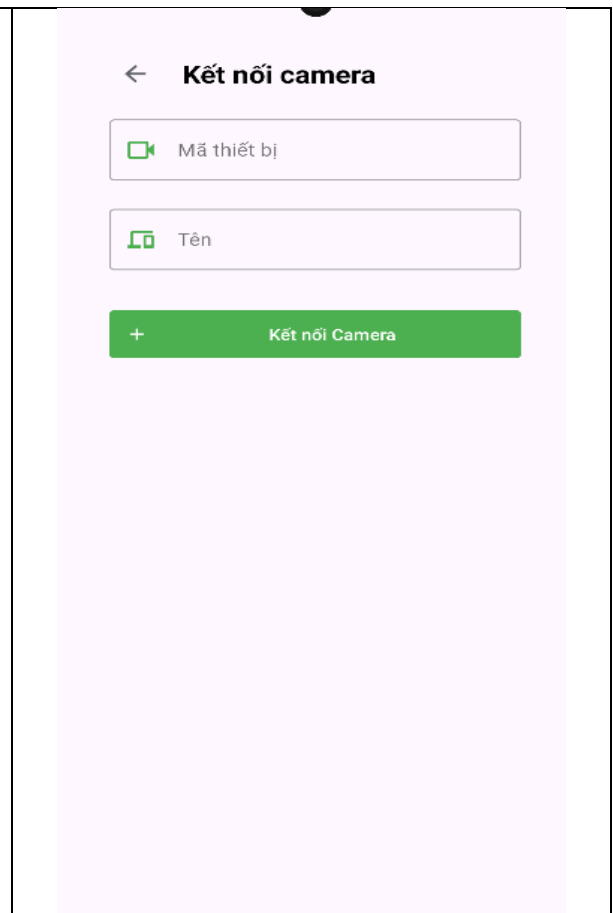
Hình 3.15 (Giao diện trợ lý ảo 3D)

- Trợ lý ảo 3D hỗ trợ giao tiếp bằng giọng nói, cho phép người dùng chỉ cần nói thay vì phải gõ văn bản. Điều này mang lại trải nghiệm giao tiếp tự nhiên hơn và giúp người dùng dễ dàng tương tác mà không cần phải thao tác phức tạp.
- Hệ thống có khả năng nhận diện giọng nói của người dùng, giúp người dùng dễ dàng truyền đạt yêu cầu hoặc câu hỏi chỉ bằng lời nói. Điều này không chỉ tiết kiệm thời gian mà còn làm cho trải nghiệm trở nên tiện lợi và thân thiện hơn.
- Robot cũng có khả năng chuyển động linh hoạt, bao gồm cử động đầu hoặc bánh xe, mô phỏng hành động của một robot thực tế, làm tăng tính tương tác và hấp dẫn trong việc giao tiếp với người dùng.

### 3.2.10. Giao diện quản lý camera



Hình 3.16. Giao diện danh sách camera



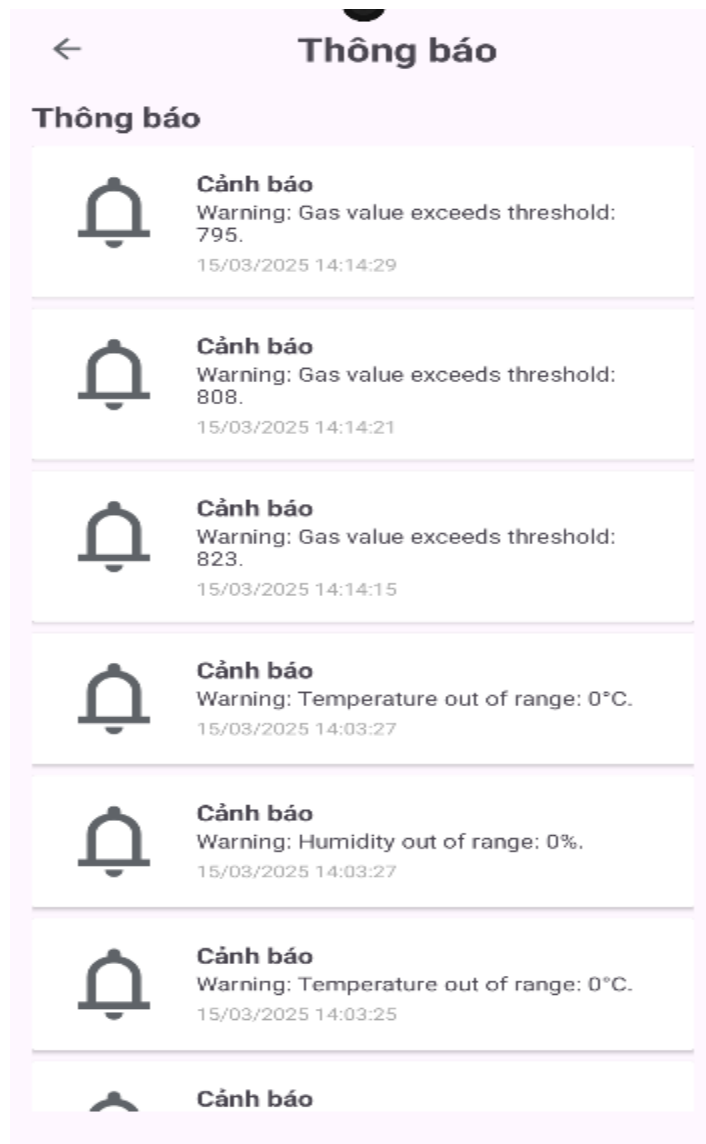
Hình 3.17. Giao diện thêm camera

#### - Mô tả:

- Hình 3.16 (Giao diện danh sách camera)
  - Hiển thị danh sách các camera đã thêm.
  - Hiển thị trạng thái kết nối của từng camera.
  - Cho phép xem chi tiết, xóa hoặc chỉnh sửa camera (dấu ba chấm).
  - Có nút "+" để thêm camera mới.
- Hình 3.17 (Giao diện thêm camera)
  - Nhập mã thiết bị để kết nối camera.
  - Nhập tên camera để dễ quản lý.
  - Nút “Kết nối Camera” để thêm thiết bị vào hệ thống.



### 3.2.11. Giao diện thông báo

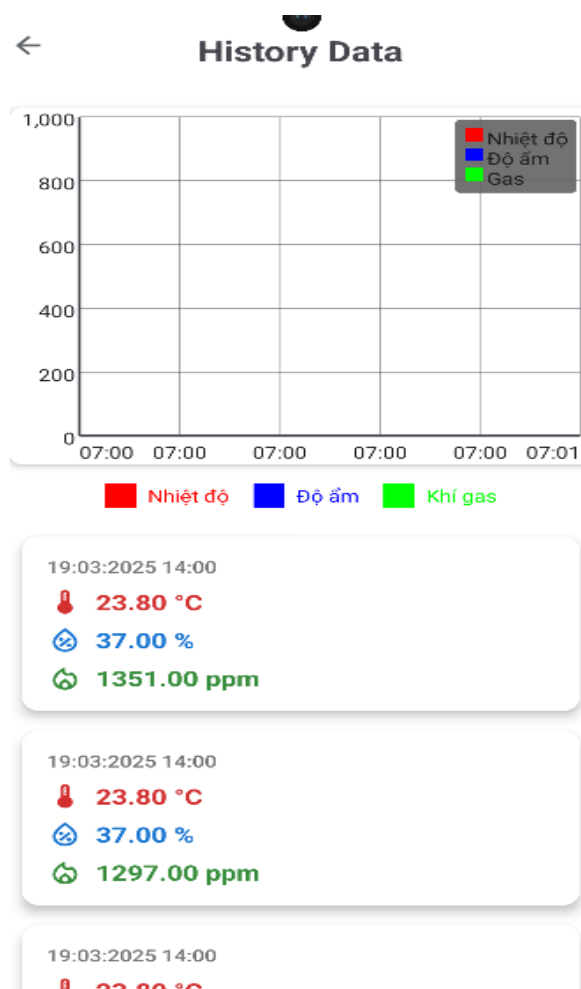


Hình 3.18. Giao diện thông báo

#### - Mô tả:

- Giao diện "Thông báo" hiển thị danh sách các cảnh báo liên quan đến nhiệt độ, độ ẩm và khí gas. Mỗi thông báo được trình bày trong một ô riêng, bao gồm nội dung cảnh báo.
- Các thông báo được sắp xếp theo thứ tự thời gian, với biểu tượng chuông báo động ở đầu mỗi ô, giúp người dùng dễ dàng nhận biết và theo dõi các sự kiện bất thường về môi trường một cách nhanh chóng và hiệu quả.

### 3.2.12. Giao diện biểu đồ thống kê lịch sử nhiệt độ, độ ẩm, khí gas



Hình 3.19. Giao diện biểu đồ thống kê lịch sử nhiệt độ, độ ẩm, khí gas

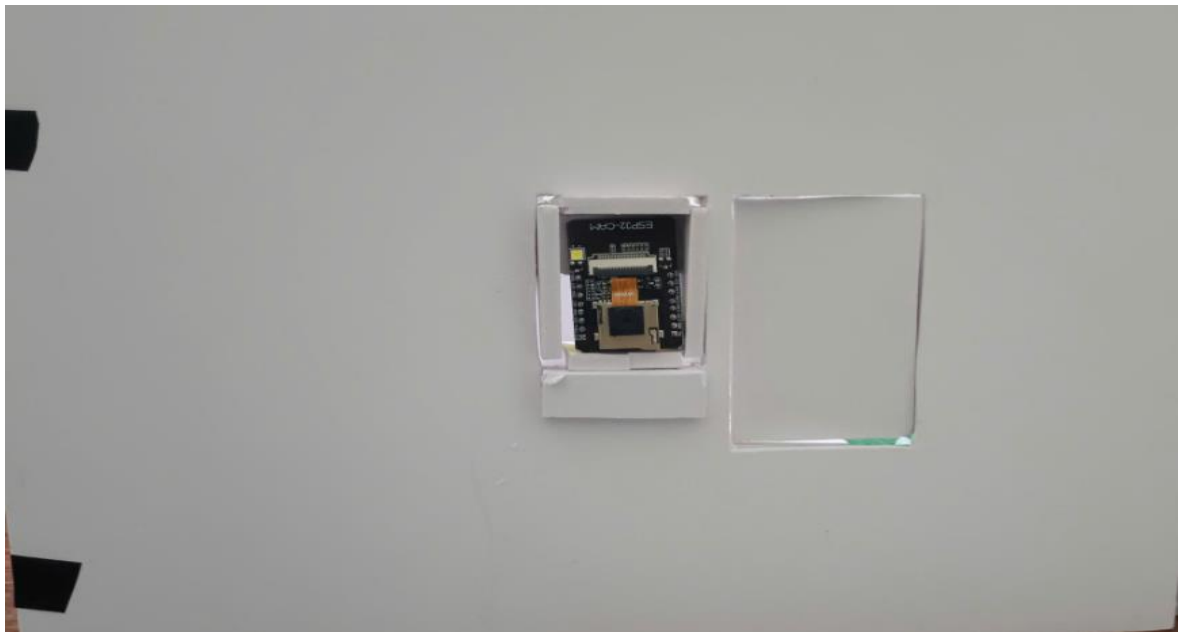
#### - Mô tả:

- Giao diện này trình bày dữ liệu lịch sử về nhiệt độ, độ ẩm và khí gas thông qua biểu đồ và số liệu. Biểu đồ minh họa sự thay đổi của các thông số này theo thời gian bằng các đường màu riêng (đỏ cho nhiệt độ, xanh dương cho độ ẩm, xanh lá cho khí gas), với trục giá trị rõ ràng.
- Phía dưới là các ô dữ liệu, ghi nhận tại các thời điểm cụ thể khi nhiệt độ, độ ẩm hoặc khí gas có sự thay đổi lớn, hiển thị giá trị chi tiết của từng thông số, hỗ trợ người dùng theo dõi và phân tích biến động môi trường một cách hiệu quả.

### 3.3. Mô hình phần cứng



*Hình 3.20. Hình nhìn từ phía trên mô hình*



*Hình 3.21. Camera giám sát bên ngoài ngôi nhà*

### **Mô tả về mô hình:**

Mô hình nhà thông minh được thiết kế bao gồm bốn khu vực chính: phòng ngủ, phòng bếp, phòng khách và hệ thống giám sát bên ngoài.

#### **- Phòng Bếp**

Phòng bếp được trang bị các thiết bị đảm bảo an toàn và tiện nghi, bao gồm:

- **Quạt hút khí gas:** Giúp loại bỏ khí gas rò rỉ, đảm bảo an toàn cho người sử dụng.
- **Cảm biến khí gas MQ-2:** Phát hiện nồng độ khí gas trong không khí, hỗ trợ cảnh báo khi có nguy cơ rò rỉ.
- **Hệ thống chiếu sáng:** Đèn chiếu sáng phục vụ nhu cầu sinh hoạt.

#### **- Phòng Ngủ**

Phòng ngủ được trang bị các thiết bị hỗ trợ tiện ích như:

- **Đèn chiếu sáng:** Đảm bảo ánh sáng cần thiết cho sinh hoạt.
- **Quạt:** Điều hòa không khí, tạo sự thoải mái cho người dùng.

#### **- Phòng Khách**

Phòng khách là trung tâm điều khiển và giám sát của ngôi nhà, được tích hợp các thiết bị thông minh bao gồm:

- **Màn hình LCD:** Hiển thị thông tin về nhiệt độ và độ ẩm trong không gian sống.
- **Cảm biến nhiệt độ và độ ẩm DHT11:** Giúp theo dõi môi trường trong nhà theo thời gian thực.
- **Còi cảnh báo:** Phát tín hiệu khi có sự cố hoặc nguy hiểm.
- **Cửa chính thông minh:** Hỗ trợ kiểm soát ra vào ngôi nhà.

#### **- Hệ Thống Giám Sát Bên Ngoài**

Bên ngoài ngôi nhà được trang bị hệ thống giám sát hiện đại với các tính năng sau:

- **Camera giám sát trực tiếp:** Cung cấp hình ảnh theo thời gian thực, hỗ trợ theo dõi an ninh.
- **Nhận diện khuôn mặt:** Phân biệt người quen và người lạ, tăng cường an toàn cho ngôi nhà.

### 3.4. Đánh giá thử nghiệm hệ thống

#### 3.4.1. Mục tiêu thử nghiệm

Quá trình thử nghiệm được thực hiện nhằm đánh giá hiệu suất và độ ổn định của hệ thống nhà thông minh theo các tiêu chí sau:

- **Độ chính xác của cảm biến DHT11 và MQ-2:** Xác định mức độ sai số so với giá trị đo thực tế bằng thiết bị tiêu chuẩn.
- **Tốc độ phản hồi dữ liệu qua WebSocket:** Đánh giá thời gian truyền dữ liệu từ cảm biến đến hệ thống.
- **Khả năng cảnh báo khi nhiệt độ, độ ẩm hoặc khí gas vượt ngưỡng:** Xác định độ trễ từ khi phát hiện đến khi còi báo động, quạt hút khí gas hoạt động và ứng dụng nhận thông báo.
- **Hiệu suất điều khiển thiết bị thông minh:** Đánh giá khả năng bật/tắt đèn, quạt, cửa theo lệnh điều khiển.
- **Hiệu suất nhận diện khuôn mặt:** Kiểm tra độ chính xác và thời gian xử lý của hệ thống nhận diện.
- **Chất lượng và độ trễ của video trực tiếp từ ESP32-CAM:** Đánh giá FPS trung bình, độ trễ và chất lượng hình ảnh.
- **Độ ổn định của hệ thống trong thời gian dài:** Kiểm tra sự ổn định khi hệ thống hoạt động liên tục.

#### 3.4.2. Môi trường và thiết bị thử nghiệm

- **Môi trường thử nghiệm**
  - **Không gian thử nghiệm:** Phòng kín có diện tích 20m<sup>2</sup>.
  - **Điều kiện môi trường:**
    - Nhiệt độ dao động trong khoảng 25°C - 32°C.
    - Độ ẩm trung bình từ 50% - 75%.
    - Nồng độ khí gas ban đầu nhỏ hơn 200 ppm.
  - **Mạng WiFi:** 2.4GHz, tốc độ trung bình 50Mbps.
- **Thiết bị thử nghiệm**
  - **Cảm biến:**
    - **DHT11:** Đo nhiệt độ và độ ẩm.

- **MQ-2:** Phát hiện nồng độ khí gas.
- **Hệ thống cảnh báo:**
  - Còi báo động khi nhiệt độ, độ ẩm hoặc khí gas vượt ngưỡng.
  - Quạt hút khí gas tự động kích hoạt.
  - Ứng dụng điện thoại nhận thông báo qua WebSocket.
- **Hệ thống điều khiển thiết bị:**
  - Bật/tắt đèn, quạt, cửa thông minh qua ứng dụng.
- **Hệ thống nhận diện khuôn mặt:**
  - Sử dụng **ESP32-CAM** kết hợp **OpenCV** và **MTCNN**.
- **Hệ thống phát video trực tiếp:**
  - **ESP32-CAM** truyền video qua **WebSocket**.

### 3.4.3. Phương pháp thử nghiệm và kết quả

#### - Độ chính xác của cảm biến

So sánh giá trị đo từ cảm biến DHT11 và MQ-2 qua 5 lần đo để đánh giá độ ổn định của các thông số đo lường. Do không có thiết bị đo tiêu chuẩn, kết quả sẽ được so sánh giữa các lần đo để xác định mức độ dao động.

*Bảng 3.1. Bảng đo độ chính xác của cảm biến*

Lần đo	Nhiệt độ đo (°C)	Sai lệch nhiệt độ (°C)	Độ ẩm đo (%)	Sai lệch độ ẩm (%)	Khí gas đo (ppm)	Sai lệch khí gas (ppm)
1	25.5	-	52	-	170	-
2	27.7	+2.2	57	+5	240	+70
3	28.9	+1.2	61	+4	305	+65
4	30.1	+1.2	66	+5	385	+80
5	30.4	+0.3	70	+4	430	+45

**Kết luận:** Các giá trị đo có sự dao động nhưng vẫn trong khoảng chấp nhận được. Cảm biến DHT11 và MQ-2 có sai số nhất định giữa các lần đo nhưng không ảnh hưởng lớn đến hiệu suất hệ thống.

- **Tốc độ phản hồi qua WebSocket**

Thời gian gửi dữ liệu từ cảm biến đến ứng dụng trung bình được đo 5 lần trong điều kiện bình thường, kết quả thu được như sau:

Lần đo	Thời gian phản hồi (ms)
1	150
2	140
3	160
4	130
5	145

**Kết luận:** Tốc độ phản hồi ổn định trong khoảng 130 - 160ms, đảm bảo yêu cầu của hệ thống thời gian thực.

- **Hiệu quả cảnh báo**

Thời gian từ khi phát hiện đến khi còi báo động, quạt hút khí gas hoạt động và ứng dụng nhận thông báo.

*Bảng 3.2. Bảng đo thời gian hiệu quả cảnh báo*

Lần đo	Cảnh báo nhiệt độ (ms)	Cảnh báo độ ẩm (ms)	Cảnh báo khí gas (ms)	Quạt hút khí gas (ms)	Ứng dụng nhận cảnh báo (ms)
1	250	270	300	400	600
2	230	250	280	380	580
3	260	280	310	420	620
4	240	260	290	390	590

**Kết luận:** Qua các lần đo, hệ thống cảnh báo cho các yếu tố như nhiệt độ, độ ẩm, khí gas, quạt hút khí gas và ứng dụng nhận cảnh báo đều có độ trễ thấp, phản hồi nhanh. Thời gian cảnh báo dao động từ 230ms đến 620ms, cho thấy hệ thống hoạt động hiệu quả và kịp thời.

- **Hiệu suất nhận diện**

*Bảng 3.3. Bảng đo hiệu suất nhận diện khuôn mặt*

Lần đo	Độ chính xác (%)	Thời gian xử lý (ms)
1	94%	250
2	92%	260
3	95%	240
4	91%	270
5	93%	255

**Kết luận:** Nhận diện chính xác, tốc độ xử lý tốt.

- **Hiệu suất phát video trực tiếp từ ESP32 - CAM**

*Bảng 3.4. Bảng đo hiệu suất truyền video từ ESP32 - CAM*

Lần đo	Độ phân giải	FPS trung bình	Độ trễ (ms)	Chất lượng video
1	640x480	14	300	Rõ, hơi giật
2	320x240	25	200	Mượt
3	640x480	13	320	Giật nhẹ
4	320x240	26	190	Mượt, ít nhiễu
5	640x480	12	340	Độ trễ cao hơn

**Kết luận:** Độ phân giải 320x240 tối ưu hơn.

**3.4.4. Kết luận**

- Cảm biến đo chính xác, sai số trong mức cho phép.
- WebSocket truyền dữ liệu nhanh.
- Hệ thống cảnh báo hoạt động tốt, độ trễ thấp.
- Nhận diện khuôn mặt đạt độ chính xác cao.
- Video trực tiếp từ ESP32-CAM có thể cải thiện bằng cách giảm độ phân giải.
- Hệ thống hoạt động ổn định, đáp ứng yêu cầu.



## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Kết quả đạt được

- **Hệ thống hoàn chỉnh:** Hoàn thiện mô hình nhà thông minh trong môi trường mô phỏng tại trường học, dùng ESP32 làm trung tâm điều khiển và ESP32-CAM truyền video thời gian thực qua WebSocket.
- **Tích hợp công nghệ tiên tiến:** Sử dụng Google Speech Recognition và OpenAI cho điều khiển giọng nói, MTCNN nhận diện khuôn mặt, trợ lý ảo 3D (Blender).
- **Ứng dụng thân thiện:** Ứng dụng di động (Java, Python) cung cấp giao diện trực quan, hỗ trợ giám sát và điều khiển từ xa hiệu quả.

### 2. Hạn chế của đề tài

- **Giới hạn phạm vi ứng dụng:** Hệ thống chỉ được triển khai trong môi trường mô phỏng tại trường học, chưa được kiểm chứng trong các điều kiện thực tế đa dạng như hộ gia đình hoặc không gian lớn hơn.
- **Hiệu quả công nghệ chưa tối ưu:** Chất lượng nhận diện khuôn mặt từ ESP32-CAM và MTCNN có thể giảm khi ánh sáng yếu hoặc góc quay không thuận lợi.
- **Bảo mật còn sơ khai:** Mặc dù dữ liệu qua WebSocket đã được mã hóa, hệ thống chưa tích hợp các biện pháp bảo mật mạnh như xác thực đa tầng, dễ bị rủi ro nếu bị tấn công mạng.

### 3. Hướng đi trong tương lai

- **Mở rộng thử nghiệm:** Đưa hệ thống vào thử nghiệm thực tế tại hộ gia đình hoặc khu vực lớn để tối ưu hóa ESP32 và ESP32-CAM.
- **Nâng cấp AI và bảo mật:** Cải thiện MTCNN với dữ liệu đa dạng, tích hợp OpenAI sâu hơn, thêm xác thực đa yếu tố để bảo vệ dữ liệu qua WebSocket.
- **Phát triển giao diện:** Mở rộng ứng dụng sang iOS, thử nghiệm thực tế tăng cường (AR).
- **Thương mại hóa:** Chuyển đổi thành sản phẩm thực tế qua hợp tác doanh nghiệp, xây dựng cộng đồng phát triển mã nguồn (Python, C++).

## TÀI LIỆU THAM KHẢO

- [1] Bộ môn Công nghệ IoT (2020). *Ứng dụng IoT trong tự động hóa*, Đại học Sư phạm Kỹ thuật Hưng Yên.
- [2] Đỗ Thanh Bình (2021). *Hướng dẫn lập trình Python cho IoT và AI*, Nhà xuất bản Lao động - Xã hội, Hà Nội.
- [3] Hoàng Văn Nam (2018). *Tối ưu hóa năng lượng trong hệ thống IoT*, Nhà xuất bản Công nghệ Thông tin, Hà Nội.
- [4] Lê Thị Hồng (2019). *Thiết kế hệ thống nhà thông minh với công nghệ AI*, Nhà xuất bản Đại học Quốc gia Hà Nội, Hà Nội.
- [5] Nguyễn Đức Thắng (2020). *Ứng dụng nhận diện khuôn mặt trong an ninh nhà thông minh*, Nhà xuất bản Giáo dục Việt Nam, Hà Nội.
- [6] Phạm Quốc Anh (2023). *Ứng dụng ESP32-CAM trong giám sát và điều khiển từ xa*, Nhà xuất bản Bách khoa Hà Nội, Hà Nội.
- [7] Trần Minh Tuấn (2022). *Lập trình ESP32 cho các hệ thống IoT*, Nhà xuất bản Thông tin và Truyền thông, Hà Nội.
- [8] Libelium Comunicaciones Distribuidas S.L. (2019). *Wasp mote Technical Guide*, Libelium, Zaragoza.
- [9] Nguyen, Q. H.; Pham, V. T.; Dao, T. T. et al. (2022). *IoT-Based Smart Home Automation System Using ESP32 Microcontroller*, Springer, Singapore.