

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA CNTT1**



**BÁO CÁO CUỐI KỲ**

**Đề tài: Phát triển hệ thống IoT cho nhà thông minh mini**

**Môn học: IoT và Ứng dụng**

**Giảng viên: Kim Ngọc Bách**

**Nhóm học phần: 06**

**Nhóm bài tập: 09**

**Hà Nội, 2025**

**Bảng phân công công việc trong nhóm**

Họ Tên	Mã sinh viên	Công việc
Nguyễn Trần Minh Thái	B22DCCN782	Xây dựng web giao tiếp (Phần server) Lắp đặt thiết bị Tích hợp AI
Lê Anh Vũ	B22DCCN913	Lập trình cho ESP32 và các thiết bị Lắp đặt thiết bị Báo cáo
Trần Minh Hoàng	B22DCCN349	Lập trình cho ESP32 và các thiết bị Lắp đặt thiết bị Tích hợp AI
Bùi Công Sơn	B22DCCN686	Xây dựng web giao tiếp(Giao diện) Lắp đặt thiết bị Tích hợp AI

## Mục lục

Phần 1, Giới thiệu đề tài.....	4
1.1. Lý do chọn đề tài.....	4
1.2. Tổng quan về vấn đề nghiên cứu .....	4
1.3. Mục đích nghiên cứu .....	5
1.4. Đối tượng và phạm vi nghiên cứu .....	5
1.5. Sơ lược các thiết bị trong hệ thống (dự kiến) .....	6
Phần 2, Cơ sở lý thuyết.....	7
2.1. Khái niệm về IOT ( Internet of things ).....	7
2.2. Tiện ích khi sử dụng hệ thống nhà thông minh .....	7
2.3. Cấu trúc hệ thống nhà thông minh trong thực tế .....	8
2.4. Giao thức HTTP, WebSocket và MQTT .....	9
2.5. Các thiết bị trong hệ thống.....	10
Phần 3, Phân tích yêu cầu .....	17
3.1. Yêu cầu chức năng.....	17
3.2. Yêu cầu phi chức năng.....	21
Phần 4, Phân tích thiết kế hệ thống.....	25
4.1 Sơ đồ kiến trúc hệ thống và use - case tổng quan.....	25
4.2 Thiết kế backend - database.....	27
4.3 ESP.....	28
4.4 Ảnh một số endpoint kết nối và code .....	29
Phần 5, Đánh giá kết quả dự án .....	31
5.1 Ảnh các giao diện.....	31
5.2. ĐÁNH GIÁ KẾT QUẢ VÀ CHẤT LƯỢNG HỆ THỐNG.....	36
Phần 6, Kết luận và định hướng tương lai .....	39
6.1. Kết luận.....	39
6.2. Định hướng phát triển trong tương lai .....	39
TÀI LIỆU THAM KHẢO .....	41

# **Phần 1, Giới thiệu đề tài**

## **1.1. Lý do chọn đề tài**

Nhà thông minh là một trong những xu hướng công nghệ nổi bật trong thời đại hiện nay, khi mà IoT (Internet of Things) ngày càng được ứng dụng sâu rộng trong đời sống. Việc lựa chọn đề tài này xuất phát từ nhu cầu thực tế về sự tiện nghi, an toàn và tối ưu hóa năng lượng trong quản lý nhà ở. Trong bối cảnh hiện đại, con người ngày càng quan tâm đến việc cải thiện chất lượng cuộc sống, và nhà thông minh cung cấp các giải pháp tự động hóa, tiết kiệm thời gian, và tăng cường bảo mật.

Bên cạnh đó, với sự phát triển mạnh mẽ của các thiết bị IoT, mô hình nhà thông minh còn mở ra cơ hội học tập và nghiên cứu các công nghệ tiên tiến như cảm biến, vi điều khiển, giao thức truyền thông (như MQTT, Zigbee), và trí tuệ nhân tạo (AI). Đề tài cũng góp phần giải quyết những vấn đề xã hội quan tâm như tiêu thụ năng lượng hiệu quả, giảm thiểu rủi ro cháy nổ, hay bảo vệ môi trường thông qua việc sử dụng công nghệ tối ưu.

Ngoài ra, mô hình nhà thông minh có tiềm năng ứng dụng rộng rãi, không chỉ trong đời sống gia đình mà còn trong các lĩnh vực kinh doanh, giáo dục và y tế. Do đó, việc nghiên cứu và triển khai đề tài không chỉ có ý nghĩa trong việc học tập mà còn tạo ra giá trị thực tiễn cao, đồng thời cung cấp kiến thức nền tảng cho các dự án IoT trong tương lai.

## **1.2. Tổng quan về vấn đề nghiên cứu**

Nghiên cứu về mô hình nhà thông minh tập trung vào việc giải quyết nhiều vấn đề quan trọng, từ thiết kế hệ thống đến ứng dụng thực tế. Một trong những vấn đề cốt lõi là kiến trúc hệ thống, bao gồm việc lựa chọn các thiết bị, cảm biến và giao thức truyền thông để đảm bảo sự linh hoạt và tương thích. Các

giao thức như MQTT, Zigbee hay Bluetooth cần được nghiên cứu để đảm bảo kết nối ổn định giữa các thiết bị. Bên cạnh đó, tự động hóa và trí tuệ nhân tạo đóng vai trò quan trọng trong việc tạo ra các kịch bản tự động hóa thông minh, tối ưu hóa năng lượng và cải thiện trải nghiệm người dùng. Một vấn đề khác cần được nghiên cứu là bảo mật và quyền riêng tư của người dùng, vì các thiết bị IoT dễ bị tấn công nếu không có các biện pháp bảo vệ thích hợp.

Ngoài ra, việc phát triển các ứng dụng di động thân thiện, cùng với các giao diện người dùng trực quan, là yếu tố không thể thiếu để điều khiển hệ thống nhà thông minh. Cuối cùng, việc đánh giá tính thực tiễn và khả năng mở rộng của hệ thống, đặc biệt là trong điều kiện hạ tầng mạng hạn chế, là vấn đề quan trọng để đảm bảo sự thành công và hiệu quả lâu dài của nhà thông minh.

### **1.3. Mục đích nghiên cứu**

Mục đích chính của nghiên cứu này là phát triển và triển khai thành công một ứng dụng điều khiển từ xa trên android và thiết kế được một bảng mạch mô hình đơn giản về các cảm biến được sử dụng trong mô hình nhà thông minh. Nghiên cứu tập trung vào việc tạo ra một ứng dụng dễ sử dụng, có thể tích hợp trong các môi trường làm việc từ xa đồng thời thiết kế nên một board mạch có tính ứng dụng cao trong thực tế.

### **1.4. Đối tượng và phạm vi nghiên cứu**

- Đối tượng nghiên cứu: Giao thức MQTT, module ESP32, các cảm biến thông dụng được sử dụng trong các mô hình nhà thông minh trong thực tế.
- Phạm vi nghiên cứu: Dự án tập trung vào việc phát triển ứng dụng bật tắt các thiết bị từ xa thông qua giao thức MQTT, đồng thời thiết kế và phát triển một board mạch sử dụng module ESP32 để kết nối cũng như tương tác giữa các cảm biến.

## 1.5. Sơ lược các thiết bị trong hệ thống (dự kiến)

Phần cứng:

- ESP32 WROOM 32
- Cảm biến DHT11
- Cảm biến chuyển động HC-SR501 PIR
- Màn hình LCD 1602 để hiển thị thông số
- Module 2 relay để điều khiển bật tắt quạt, máy bơm
- Server để đóng mở cửa
- Kit để kết nối các linh kiện
- Các dây dẫn
- Nguồn điện
- ...

Phần mềm:

- Arduino IDE : Arduino IDE để viết mã cho ESP32. Chương trình viết xong sẽ được nạp vào ESP32 thông qua giao tiếp USB. Arduino IDE có công cụ kiểm tra lỗi và nạp chương trình để giúp đảm bảo rằng chương trình hoạt động ổn định.
- Frontend: React 18 + TypeScript, Vite build tool, Tailwind CSS for UI, React Router cho điều hướng, React Query + Axios cho gọi API/mock layer.
- Backend: NestJS 11 (Node.js), TypeORM + PostgreSQL, JWT authentication và module hoá (users, devices, logs, roles/permissions).

## **Phần 2, Cơ sở lý thuyết**

### **2.1. Khái niệm về IOT ( Internet of things )**

Internet of Things (IoT), hay Internet vạn vật, là mô hình công nghệ cho phép các thiết bị vật lý được kết nối với nhau qua mạng Internet để thu thập, trao đổi và xử lý dữ liệu một cách tự động. Các thiết bị này có thể là cảm biến, thiết bị điện tử, máy móc, phương tiện hoặc các vật thể trong đời sống hàng ngày, được trang bị phần mềm và kết nối mạng nhằm tương tác với môi trường và con người.

Hệ thống IoT thường gồm ba lớp chính: (1) Lớp cảm nhận, nơi các cảm biến thu thập dữ liệu vật lý như nhiệt độ, độ ẩm, ánh sáng, chuyển động; (2) Lớp mạng, đảm nhiệm truyền dữ liệu qua các giao thức như Wi-Fi, Bluetooth, Zigbee hoặc mạng di động; và (3) Lớp ứng dụng, nơi dữ liệu được xử lý, lưu trữ và hiển thị cho người dùng qua các nền tảng web hoặc di động.

Công nghệ IoT hiện được ứng dụng rộng rãi trong nhiều lĩnh vực như nhà thông minh, nông nghiệp, y tế, công nghiệp và giao thông thông minh. Trong đó, mô hình nhà thông minh cho phép người dùng giám sát và điều khiển các thiết bị điện tử trong gia đình từ xa, góp phần nâng cao tiện nghi, tiết kiệm năng lượng và đảm bảo an toàn.

### **2.2. Tiện ích khi sử dụng hệ thống nhà thông minh**

Hệ thống nhà thông minh mang lại nhiều tiện ích vượt trội trong đời sống hiện đại, giúp tối ưu hóa sự tiện nghi, an toàn và hiệu quả năng lượng cho người sử dụng. Trước hết, người dùng có thể dễ dàng điều khiển các thiết bị điện như đèn, quạt, điều hòa hay rèm cửa từ xa thông qua điện thoại hoặc máy tính có kết nối Internet, giúp tiết kiệm thời gian và nâng cao trải nghiệm sử dụng.

Bên cạnh đó, hệ thống còn cho phép tự động hóa các hoạt động trong nhà dựa trên cảm biến hoặc kịch bản được thiết lập sẵn, ví dụ: đèn tự bật khi phát

hiện chuyển động, điều hòa tự điều chỉnh theo nhiệt độ, hay rèm cửa tự đóng khi ánh sáng quá mạnh. Nhờ đó, không chỉ mang lại sự thoải mái mà còn giảm lãng phí năng lượng.

Ngoài ra, nhà thông minh còn tăng cường khả năng giám sát và bảo mật thông qua camera, cảm biến chuyển động, cảm biến khí gas hoặc cảnh báo cháy nổ. Người dùng có thể nhận thông báo tức thời khi có sự cố, đảm bảo an toàn cho ngôi nhà ngay cả khi vắng mặt. Với khả năng mở rộng linh hoạt và kết nối nhiều thiết bị, hệ thống nhà thông minh đang trở thành xu hướng tất yếu, góp phần xây dựng môi trường sống hiện đại, tiện nghi và bền vững.

### **2.3. Cấu trúc hệ thống nhà thông minh trong thực tế**

Một hệ thống nhà thông minh trong thực tế thường được xây dựng theo mô hình nhiều lớp, nhằm đảm bảo khả năng kết nối, điều khiển và mở rộng linh hoạt giữa các thiết bị. Cấu trúc tổng quát bao gồm bốn thành phần chính: tầng cảm biến, tầng điều khiển trung tâm, tầng truyền thông và tầng ứng dụng.

- Tầng cảm biến (Perception Layer): là nơi thu thập dữ liệu từ môi trường xung quanh thông qua các cảm biến như nhiệt độ, độ ẩm, ánh sáng, chuyển động, khí gas, hay camera. Các thiết bị này có nhiệm vụ chuyển đổi tín hiệu vật lý thành dữ liệu số để gửi lên bộ xử lý trung tâm.
- Tầng điều khiển trung tâm (Control Layer): bao gồm các vi điều khiển như ESP32, Raspberry Pi hoặc các bộ gateway. Tầng này chịu trách nhiệm xử lý dữ liệu thu được, đưa ra quyết định và gửi lệnh điều khiển tới các thiết bị đầu ra như relay, đèn, quạt hoặc ổ cắm thông minh.
- Tầng truyền thông (Network Layer): là cầu nối giúp các thiết bị trao đổi dữ liệu thông qua các giao thức như Wi-Fi, Zigbee, Bluetooth, hoặc MQTT. Tầng này đảm bảo việc truyền dữ liệu ổn định giữa thiết bị, máy chủ và ứng dụng người dùng.
- Tầng ứng dụng (Application Layer): cung cấp giao diện để người dùng theo dõi và điều khiển hệ thống. Dữ liệu từ các thiết bị được lưu trữ và



hiển thị trên web hoặc ứng dụng di động, đồng thời cho phép người dùng thiết lập kịch bản tự động hóa, quản lý người dùng và cập nhật phần mềm từ xa.

Nhờ cấu trúc này, hệ thống nhà thông minh có thể hoạt động một cách ổn định, dễ dàng mở rộng thêm thiết bị mới, đồng thời hỗ trợ tích hợp các công nghệ tiên tiến như trí tuệ nhân tạo (AI) hoặc điện toán đám mây (Cloud Computing) để tối ưu hiệu suất và trải nghiệm sử dụng.

## **2.4. Giao thức HTTP, WebSocket và MQTT**

Trong các hệ thống IoT, đặc biệt là mô hình nhà thông minh, việc lựa chọn giao thức truyền thông đóng vai trò quan trọng để đảm bảo tốc độ, độ tin cậy và hiệu quả trong việc trao đổi dữ liệu giữa các thiết bị và máy chủ. Ba giao thức phổ biến nhất là HTTP, WebSocket và MQTT, mỗi giao thức có đặc điểm và ứng dụng riêng phù hợp với từng mục đích.

- **Giao thức HTTP (HyperText Transfer Protocol):** là giao thức truyền tải dữ liệu phổ biến trong các ứng dụng web. HTTP hoạt động theo mô hình client-server, trong đó client (ví dụ: ESP32 hoặc trình duyệt web) gửi yêu cầu (request) và server phản hồi (response). Ưu điểm của HTTP là đơn giản, dễ triển khai và tương thích với hầu hết các thiết bị có kết nối Internet. Tuy nhiên, HTTP hoạt động theo cơ chế truy vấn định kỳ (polling), không phù hợp cho các ứng dụng cần cập nhật dữ liệu liên tục do gây tốn băng thông và độ trễ cao.
- **Giao thức WebSocket:** là phần mở rộng của HTTP, cho phép thiết lập kênh truyền hai chiều (full-duplex) giữa client và server sau khi kết nối được thiết lập. Nhờ đó, dữ liệu có thể được gửi và nhận liên tục mà không cần thực hiện nhiều lần request-response như HTTP. WebSocket thích hợp cho các ứng dụng cần truyền dữ liệu thời gian thực, chẳng hạn như giám sát cảm biến, hiển thị trạng thái thiết bị hoặc điều khiển từ xa với độ trễ thấp.

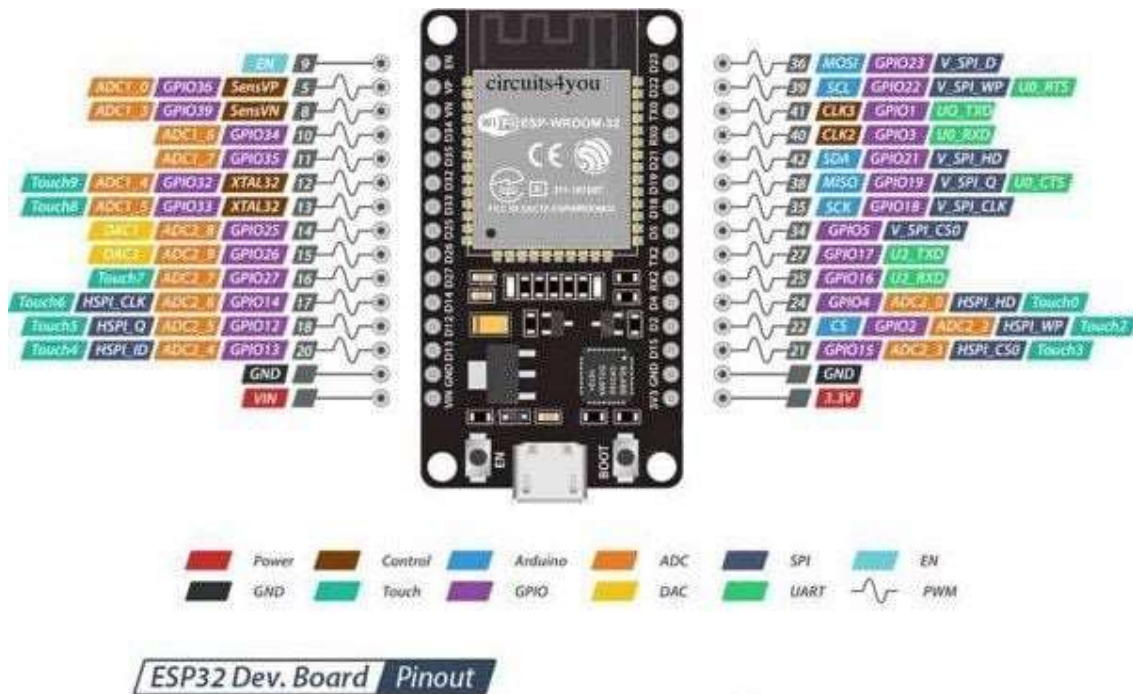
- Giao thức MQTT (Message Queuing Telemetry Transport): là giao thức truyền thông nhẹ được thiết kế chuyên biệt cho các ứng dụng IoT. MQTT hoạt động theo mô hình publish–subscribe thông qua một máy chủ trung gian gọi là broker. Thiết bị gửi dữ liệu (publisher) và thiết bị nhận dữ liệu (subscriber) chỉ cần đăng ký vào cùng một chủ đề (topic) mà không cần biết địa chỉ cụ thể của nhau. Ưu điểm của MQTT là gọn nhẹ, tiết kiệm băng thông, hoạt động ổn định trong môi trường mạng yếu và hỗ trợ cơ chế xác nhận QoS (Quality of Service) để đảm bảo độ tin cậy khi truyền dữ liệu.

Tổng kết lại, trong hệ thống nhà thông minh, HTTP phù hợp cho việc cấu hình và gửi yêu cầu đơn giản, WebSocket thích hợp cho giao tiếp hai chiều liên tục giữa web và thiết bị, trong khi MQTT là lựa chọn tối ưu cho việc truyền dữ liệu cảm biến và điều khiển nhiều thiết bị IoT với hiệu suất cao và độ trễ thấp.

## **2.5. Các thiết bị trong hệ thống**

### **2.5.1. ESP-WROOM-32:**

Đây là bộ vi điều khiển mạnh mẽ với khả năng kết nối Wi-Fi và Bluetooth, rất phù hợp cho các ứng dụng IoT.

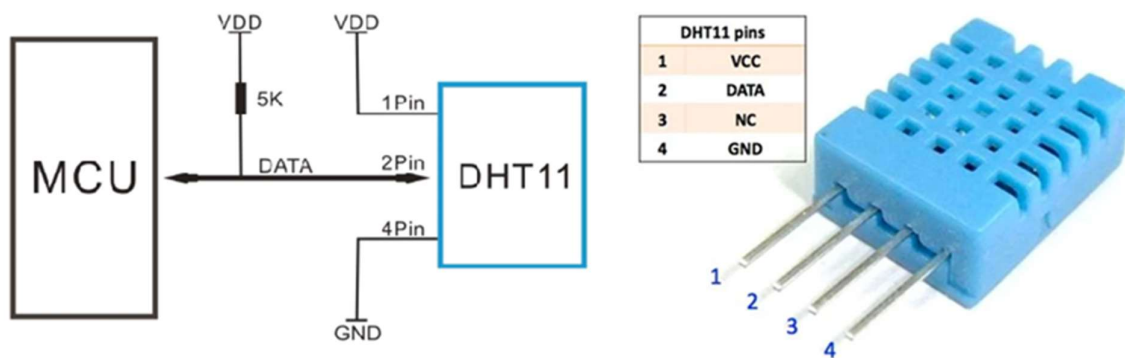


### Phần cứng của ESP-WROOM-32:

- **GPIO (General Purpose Input/Output):** ESP32 có 34 chân GPIO có thể lập trình được, với mỗi chân có thể thực hiện nhiều chức năng khác nhau như đầu vào/đầu ra kỹ thuật số, ADC, UART, SPI, I2C, và PWM. Một số chân được chỉ định sẵn cho các chức năng đặc biệt như ADC hoặc DAC, nhưng các chân khác có thể được cấu hình tùy chỉnh qua chương trình.
- **ADC (Analog to Digital Converter):** ESP32 có 18 kênh ADC 12-bit chia thành hai khối (ADC1 và ADC2), cho phép chuyển đổi tín hiệu analog thành giá trị số từ 0 đến 4093. Độ phân giải cao của ADC giúp đo chính xác các tín hiệu từ cảm biến. **DAC (Digital to Analog Converter):** ESP32 tích hợp 2 kênh DAC 8-bit trên các chân GPIO25 và GPIO26, cho phép chuyển đổi tín hiệu số sang tín hiệu điện áp analog.
- **PWM (Pulse Width Modulation):** Có 16 kênh PWM độc lập, giúp điều khiển động cơ và đèn LED. Người dùng có thể điều chỉnh tần số, chu kỳ nhiệm vụ và gán kênh PWM cho bất kỳ chân GPIO nào.

- **RTC GPIO (Real-Time Clock GPIO):** ESP32 có 16 GPIO RTC, giúp đánh thức thiết bị khỏi chế độ ngủ sâu (Deep Sleep) nhờ các nguồn đánh thức bên ngoài.
- **Cảm biến điện dung:** Có 10 GPIO cảm ứng điện dung, sử dụng để phát hiện sự thay đổi điện dung khi có vật chạm vào chân cảm ứng, không cần phần cứng bổ sung.
- **Giao diện UART:** ESP32 có 3 giao diện UART cho truyền thông nối tiếp, thường sử dụng cho giao tiếp với máy tính hoặc các module khác.
- **Giao diện SPI:** ESP32 có 3 khối SPI (SPI, HSPI, VSPI), dùng cho truyền thông tốc độ cao với các thiết bị ngoại vi như bộ nhớ Flash.
- **Giao diện I2C:** 2 giao diện I2C cho phép giao tiếp với các cảm biến và thiết bị ngoại vi. Các chân I2C mặc định trong Arduino IDE là GPIO21 (SDA) và GPIO22 (SCL), nhưng có thể cấu hình lại.
- **Chỉ đầu vào GPIO:** Một số chân như GPIO34, GPIO35, GPIO36, và GPIO39 chỉ hỗ trợ đầu vào kỹ thuật số.

### 2.5.2. DHT 11

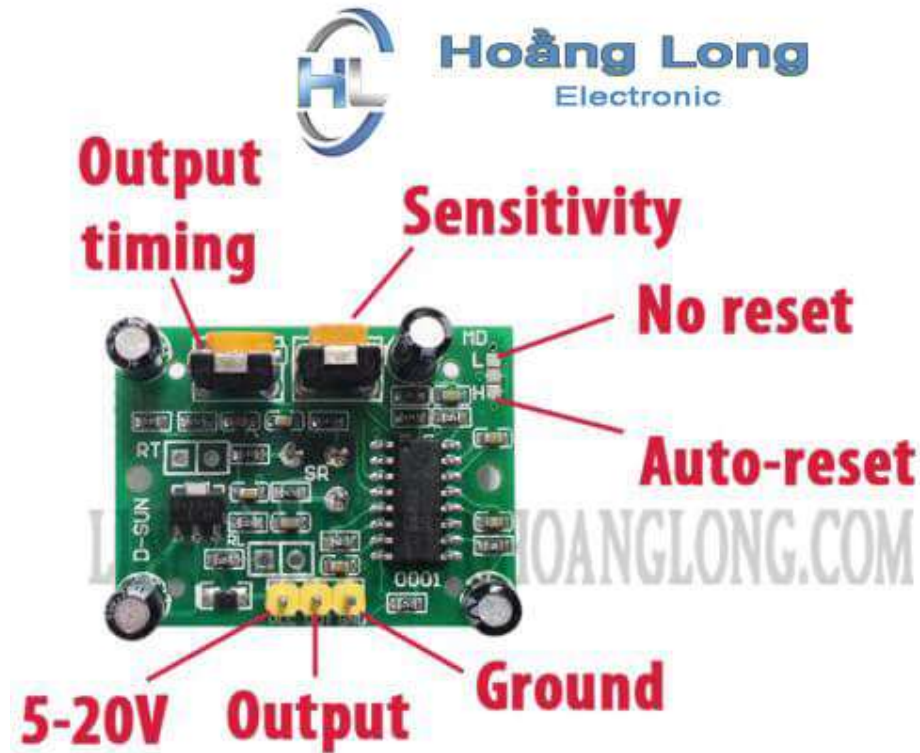


DHT-11 là module cảm biến nhiệt độ, độ ẩm có giao tiếp 1 dây (one wire). Cấu tạo cảm biến gồm 2 phần: một phần tử cảm biến độ ẩm bằng điện dung và một điện trở nhiệt để cảm nhận nhiệt độ. Tụ điện cảm biến độ ẩm có hai điện cực với chất nền giữ ẩm làm chất điện môi giữa chúng. Thay đổi giá trị điện dung xảy

ra theo sự thay đổi của các mức độ ẩm. Ngoài ra bên trong module còn có các mạch chuyển đổi tương tự sang số. Cảm biến được tích hợp bộ tiền xử lý giúp cho dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ khâu phân tích hay tính toán nào. Các thông số kỹ thuật của module cảm biến nhiệt độ, độ ẩm DHT-11 được cho như sau:

- Điện áp hoạt động  $3,5 \div 5,5V_{dc}$
- Dòng điện hoạt động cực đại là  $2,5mA$
- Dải độ ẩm hoạt động  $20\% \div 95\% RH$ , sai số  $\pm 5\%RH$  (range humidity)
- Dải nhiệt độ hoạt động  $0 \div 50^{\circ}C$ , sai số  $\pm 2^{\circ}C$
- Nhiệt độ và độ ẩm đều có độ phân giải 16 bit.
- Tốc độ lấy mẫu không quá 1Hz (mỗi giây một lần).
- Tốc độ cảm nhận: trung bình 2s
- Khoảng cách truyền tối đa 20m.

### 2.5.3. HC-SR501 PIR Sensor



Nguyên lý hoạt động: Cảm biến HC-SR501 là loại cảm biến thụ động hồng ngoại (PIR – Passive Infrared Sensor), có chức năng phát hiện chuyển động của con người hoặc động vật dựa trên sự thay đổi bức xạ hồng ngoại phát ra từ cơ thể sinh vật. Khi có người di chuyển trong phạm vi quét của cảm biến, tín hiệu hồng ngoại thay đổi và cảm biến sẽ gửi tín hiệu mức cao (HIGH) đến vi điều khiển để kích hoạt hành động tương ứng, ví dụ như bật đèn hoặc gửi cảnh báo.

Điện áp hoạt động: HC-SR501 hoạt động ổn định trong dải điện áp từ 4.5V đến 20V DC, phù hợp với cả các hệ thống 5V và 3.3V như ESP32. Tín hiệu đầu ra logic TTL (0V – 3.3V) nên có thể kết nối trực tiếp với chân GPIO của ESP32 mà không cần mạch chuyển mức điện áp.

Kết nối tín hiệu: Cảm biến HC-SR501 có 3 chân kết nối chính:

- VCC: cấp nguồn (4.5V – 20V)
- OUT: chân tín hiệu đầu ra (kết nối đến GPIO của ESP32, ví dụ GPIO27 hoặc GPIO14)
- GND: nối đất (ground)

Khi không phát hiện chuyển động, chân OUT ở mức thấp (LOW – 0V); khi phát hiện chuyển động, chân OUT ở mức cao (HIGH – 3.3V) trong khoảng thời gian được định trước.

Cấu tạo: HC-SR501 sử dụng phần tử cảm biến hồng ngoại Pyroelectric Infrared Sensor để phát hiện sự thay đổi năng lượng hồng ngoại. Mắt cảm biến được che phủ bởi thấu kính Fresnel, giúp mở rộng vùng quét và tập trung bức xạ hồng ngoại vào phần tử cảm biến. Ngoài ra, mạch còn tích hợp IC BISS0001 để khuếch đại tín hiệu và điều khiển đầu ra ổn định.

Thông số kỹ thuật cơ bản:

- Điện áp hoạt động: 4.5V – 20V DC
- Dòng tiêu thụ: < 65  $\mu$ A
- Tín hiệu đầu ra: 0V (LOW) / 3.3V (HIGH)
- Phạm vi phát hiện: 3 – 7 mét (có thể điều chỉnh)
- Góc phát hiện: 100° – 120°
- Thời gian trễ đầu ra: 0.3s – 200s (điều chỉnh được)
- Kích thước module: 32mm x 24mm
- Giao tiếp: Digital (chân OUT nối trực tiếp với GPIO)

Ứng dụng trong hệ thống nhà thông minh: Trong mô hình nhà thông minh, cảm biến HC-SR501 được sử dụng để phát hiện sự hiện diện hoặc chuyển động của con người. Khi phát hiện có người, hệ thống sẽ tự động bật đèn, kích hoạt camera hoặc gửi tín hiệu cảnh báo qua MQTT. Cảm biến cũng có thể kết hợp với cảm biến ánh sáng LDR để đảm bảo đèn chỉ bật khi trời tối và có người di chuyển, giúp tiết kiệm năng lượng.

HC-SR501 là loại cảm biến có độ ổn định cao, dễ sử dụng và giá thành thấp, rất phù hợp cho các ứng dụng tự động hóa trong nhà thông minh, đặc biệt là trong các khu vực hành lang, sân, hoặc phòng khách.



## **Phần 3, Phân tích yêu cầu**

### **3.1. Yêu cầu chức năng**

#### **3.1.1 Đo nhiệt độ, độ ẩm theo thời gian thực**

- Hệ thống tích hợp cảm biến DHT11 để đo nhiệt độ và độ ẩm của môi trường xung quanh theo thời gian thực
- Dữ liệu thu thập được sẽ được gửi về vi điều khiển ESP32 để xử lý và hiển thị lên màn hình LCD hoặc giao diện web/app
- Các giá trị đo được cũng được ghi lại vào bộ nhớ tạm thời hoặc cơ sở dữ liệu, phục vụ cho việc giám sát xu hướng biến đổi môi trường theo thời gian.

#### **3.1.2. Đèn bật tắt tự động và điều khiển thủ công**

- Hệ thống sử dụng cảm biến chuyển động để phát hiện chuyển động của con người
- Khi có người, đèn sẽ tự động bật; khi không phát hiện người trong một thời gian dài, đèn tự động tắt
- Ngoài ra, người dùng có thể điều khiển thủ công việc bật/tắt đèn thông qua ứng dụng điện thoại hoặc công tắc ảo trên web server ESP32
- Trạng thái bật/tắt đèn được lưu lại vào hệ thống để đồng bộ giữa thiết bị thực tế và giao diện điều khiển.

#### **3.1.3. Điều khiển quạt thủ công hoặc tự động**

- Hệ thống tích hợp rơ-le điều khiển quạt để bật/tắt quạt dựa trên giá trị nhiệt độ đo được từ cảm biến DHT11
- Khi nhiệt độ vượt quá ngưỡng cho phép (ví dụ  $> 30^{\circ}\text{C}$ ), quạt tự động bật; khi nhiệt độ giảm xuống ngưỡng an toàn, quạt tự động tắt

- Người dùng vẫn có thể bật/tắt thủ công quạt từ xa thông qua ứng dụng điều khiển
- Dữ liệu về trạng thái quạt và ngưỡng nhiệt độ được lưu trữ trong bộ nhớ hoặc cơ sở dữ liệu, có thể thay đổi theo yêu cầu người dùng.

#### **3.1.4. Điều khiển thiết bị qua điện thoại hoặc giao diện web**

- Hệ thống cung cấp giao diện điều khiển qua điện thoại/ web, được xây dựng dựa trên giao tiếp Wi-Fi của ESP32
- Người dùng có thể thực hiện các thao tác như: bật/tắt đèn, quạt, xem nhiệt độ – độ ẩm, và theo dõi trạng thái hệ thống theo thời gian thực
- Dữ liệu từ điện thoại được gửi đến ESP32 thông qua HTTP hoặc MQTT, sau đó xử lý và phản hồi trực tiếp trên ứng dụng
- Thông tin điều khiển và phản hồi được ghi nhận tạm thời trong bộ nhớ hệ thống để đảm bảo đồng bộ.

#### **3.1.5. Quản lý hệ thống từ xa (vòng đời, trạng thái)**

- Người dùng có thể theo dõi và giám sát trạng thái của toàn hệ thống (thiết bị đang bật/tắt, nhiệt độ hiện tại, độ ẩm, kết nối mạng, v.v.) thông qua giao diện web hoặc app di động.
- Hệ thống hỗ trợ điều khiển và cập nhật thông tin từ xa nhờ vào kết nối Internet
- Các thông tin về vòng đời thiết bị (thời gian hoạt động, lỗi, log sự kiện) được lưu trữ trên cơ sở dữ liệu hoặc bộ nhớ trong ESP32, hỗ trợ việc quản lý và bảo trì sau này.

### **3.1.6. Gửi cảnh báo khi có sự kiện bất thường**

- Khi phát hiện các điều kiện bất thường như nhiệt độ vượt ngưỡng, độ ẩm quá thấp/cao, hoặc thiết bị ngắt kết nối, hệ thống sẽ tự động gửi cảnh báo đến người dùng qua ứng dụng điện thoại hoặc email.
- Cảnh báo có thể được hiển thị dưới dạng thông báo đẩy (push notification) hoặc tin nhắn trên giao diện điều khiển.
- Mỗi cảnh báo được ghi lại trong nhật ký hệ thống để phục vụ việc kiểm tra và đánh giá hoạt động.

### **3.1.7. Hỗ trợ cập nhật firmware từ xa (OTA – Over-The-Air)**

- Hệ thống hỗ trợ nâng cấp firmware từ xa cho ESP32 thông qua tính năng OTA.
- Khi có phiên bản mới, người dùng hoặc kỹ thuật viên có thể tải firmware lên hệ thống qua mạng Wi-Fi mà không cần kết nối trực tiếp qua cáp.
- Quá trình cập nhật được kiểm tra checksum và xác thực để đảm bảo an toàn.
- Thông tin về phiên bản hiện tại và lịch sử cập nhật được lưu lại trong bộ nhớ ESP32.

### **3.1.8. Lưu trữ thông tin hoạt động**

- Tất cả các dữ liệu như nhiệt độ, độ ẩm, trạng thái đèn/quạt, thời điểm bật/tắt, và cảnh báo sẽ được ghi lại định kỳ.
- Dữ liệu có thể được lưu trữ cục bộ trên thẻ nhớ, bộ nhớ trong của ESP32 hoặc gửi lên server (Firebase).
- Việc lưu trữ giúp hệ thống phân tích xu hướng, theo dõi lịch sử hoạt động và hỗ trợ tính năng AI trong các phiên bản nâng cấp sau.

### 3.1.9 Phân quyền người dùng

Hệ thống sẽ phân ra có 3 người dùng chính là IT admin, House Owner và Guest.

Vai trò người dùng	Chức năng chính
Admin	<ul style="list-style-type: none"><li>- Quản lý người dùng (tạo, chỉnh sửa, xóa, phân quyền)</li><li>- Quản lý thiết bị (đăng ký/hủy, cấu hình ban đầu)</li><li>- Giám sát hệ thống (nhật ký, hiệu suất)</li><li>- Hỗ trợ và khắc phục sự cố</li></ul>
House Owner (Chủ nhà)	<ul style="list-style-type: none"><li>- Quản lý nhà (tạo, chỉnh sửa, xóa, mời/xóa thành viên)</li><li>- Quản lý thiết bị trong nhà (thêm, xóa, điều khiển, tự động hóa)</li><li>- Giám sát và nhận thông báo</li><li>- Cấu hình cá nhân</li></ul>
Guest/Thành viên gia đình	<ul style="list-style-type: none"><li>- Xem trạng thái thiết bị được phép</li><li>- Điều khiển thiết bị được cấp quyền</li><li>- Nhận thông báo liên quan</li></ul>

### 3.1.10 Điều khiển bằng giọng nói

Cho phép người dùng điều khiển thiết bị thông qua giọng nói trên thiết bị điều khiển. Chức năng này dựa trên ba bước chính:

- Nhận dạng Giọng nói (Speech Recognition): Chuyển đổi lệnh nói của người dùng thành văn bản số.

- Hiểu Ngôn ngữ Tự nhiên (NLU): Phân tích văn bản để hiểu ý định (Intent) của người dùng và các tham số (Entities) liên quan (ví dụ: "Bật", "Đèn phòng khách").
- Thực thi Lệnh: Gửi tín hiệu điều khiển đến thiết bị đích thông qua thiết bị điều khiển (ví dụ: TV, loa thông minh).

## 3.2. Yêu cầu phi chức năng

### 3.2.1. Hiệu năng (Performance)

- Khi cảm biến chuyển động (PIR) phát hiện có người, đèn phải phản hồi và bật sáng trong vòng  $\leq 1s$ .
- Khi người dùng bấm nút trên ứng dụng web/di động để bật thiết bị điện đèn, quạt, phải nhận được tín hiệu và khởi động trong  $\leq 2s$  trong môi trường mạng cục bộ
- Tần suất cập nhật dữ liệu: Cảm biến nhiệt độ, độ ẩm (DHT11) cần gửi dữ liệu mới lên máy chủ/ứng dụng  $\leq 60s/lần$ , đảm bảo dữ liệu tương đối thời gian thực.

### 3.2.2. Độ tin cậy (Reliability)

- Uptime của các dịch vụ backend và môi trường cần đạt  $\leq 99,9\%$
- Khả năng tự phục hồi:
  - Nếu bị mất kết nối Wi-Fi, vi điều khiển ESP32 phải có khả năng tự động kết nối lại trong vòng 60s khi mạng có trở lại.
  - Nếu mất điện, sau khi có điện trở lại, hệ thống phải tự khởi động và hoạt động với các cài đặt cuối cùng được lưu.
- Các lệnh điều khiển quan trọng (như bật/tắt thiết bị) không được phép bị mất trên đường truyền.

### 3.2.3. Bảo mật (Security)

- Ứng dụng web/di động có cơ chế đăng nhập (tên người dùng, mật khẩu) để chỉ người dùng đã đăng nhập mới có quyền xem và điều khiển thiết bị.
  - Backend sẽ triển khai API đăng nhập trả về một JSON Web Token (JWT). Mọi yêu cầu API sau đó từ frontend phải đính kèm JWT này trong header để xác thực
- Mật khẩu Wi-Fi, mật khẩu đăng nhập người dùng phải được lưu trữ an toàn trên ESP32 và không được lập trình ở dạng văn bản thuần (plaintext) trong code.
- Dữ liệu truyền giữa ESP32 và máy chủ phải được mã hóa. Khi hệ thống được truy cập từ bên ngoài, kết nối giữa client và backend phải sử dụng HTTPS.

### 3.2.4. Khả năng sử dụng (Usability)

- Lần đầu khởi động, ESP32 sẽ phát ra một mạng Wi-Fi riêng (Access Point mode). Người dùng kết nối vào mạng này, truy cập một trang web đơn giản trên ESP32 để nhập thông tin Wi-Fi của nhà. Thông tin này sau đó sẽ được lưu vào NVS.
- Khi người dùng thực hiện một hành động (ví dụ: đặt lịch hẹn giờ), hệ thống phải có phản hồi hệ thống: thông báo xác nhận hành động đó đã thành công hay thất bại.

### 3.2.5. Khả năng mở rộng

- Thêm thiết bị mới: Kiến trúc phần mềm (đặc biệt là backend và giao thức MQTT) phải được thiết kế để có thể dễ dàng thêm các cảm biến mới (cảm biến ánh sáng, cảm biến khí gas) hoặc các thiết bị điều khiển khác (ổ cắm thông minh) mà không cần phải thiết kế lại toàn bộ hệ thống.

### 3.2.6. Khả năng bảo trì (Maintainability)

- Cập nhật phần mềm (Firmware): Hệ thống nên hỗ trợ cập nhật firmware cho ESP32 từ xa qua mạng (OTA - Over-The-Air. Điều này giúp nâng cấp tính năng hoặc vá lỗi bảo mật mà không cần phải kết nối cáp vật lý.
- Tổ chức mã nguồn: Mã nguồn cho ESP32 và backend cần được viết một cách rõ ràng, có chú thích đầy đủ để người khác có thể đọc hiểu và chỉnh sửa.
- Ngoài việc ghi log qua cổng Serial, ESP32 sẽ gửi các thông tin trạng thái quan trọng (ví dụ: kết nối thành công, mất kết nối, lỗi cảm biến).

### 3.3. Yêu cầu về giao tiếp

Giao tiếp giữa người dùng và hệ thống:

- Người dùng tương tác với hệ thống qua giao diện web (ReactJS), và giao diện này phải dễ sử dụng, thân thiện, hiển thị rõ ràng các thông số cần thiết.
- Hệ thống phải đảm bảo phản hồi ngay lập tức khi người dùng thay đổi giá trị ngưỡng hoặc điều khiển thiết bị từ xa.

### 3.4. Yêu cầu về môi trường hoạt động

- Điều kiện hoạt động của cảm biến: Các cảm biến cần hoạt động ổn định trong các điều kiện môi trường khắc nghiệt như nhiệt độ cao hoặc độ ẩm lớn.
- Khả năng tích hợp với các thiết bị khác: Hệ thống cần có khả năng tích hợp với các thiết bị thông minh khác trong nhà thông minh, như hệ thống quạt thông gió tự động, cửa tự động và các thiết bị bảo vệ khác.

### 3.5. Yêu cầu về thử nghiệm

Thử nghiệm cảm biến:

- Kiểm tra độ chính xác của cảm biến DHT11, HC-SR501... trong các điều kiện môi trường khác nhau.
- Đảm bảo dữ liệu đo được phản hồi đúng và ổn định khi gửi lên hệ thống.

Thử nghiệm điều khiển thiết bị:

- Đánh giá khả năng bật/tắt đèn và quạt thủ công và tự động theo điều kiện cảm biến.
- Kiểm tra độ trễ giữa thao tác điều khiển trên web và phản hồi thực tế của thiết bị.

Thử nghiệm truyền thông và đồng bộ:

- Đảm bảo ESP32 truyền và nhận dữ liệu qua MQTT/Wifi ổn định, không mất gói hoặc trễ quá lâu.
- Kiểm tra khả năng đồng bộ dữ liệu giữa thiết bị và giao diện web theo thời gian thực.

Thử nghiệm OTA (Over-The-Air):

- Kiểm tra khả năng cập nhật firmware từ xa qua Wi-Fi, đảm bảo thiết bị khởi động lại và hoạt động bình thường sau khi cập nhật.

Thử nghiệm nhận diện AI (dự kiến):

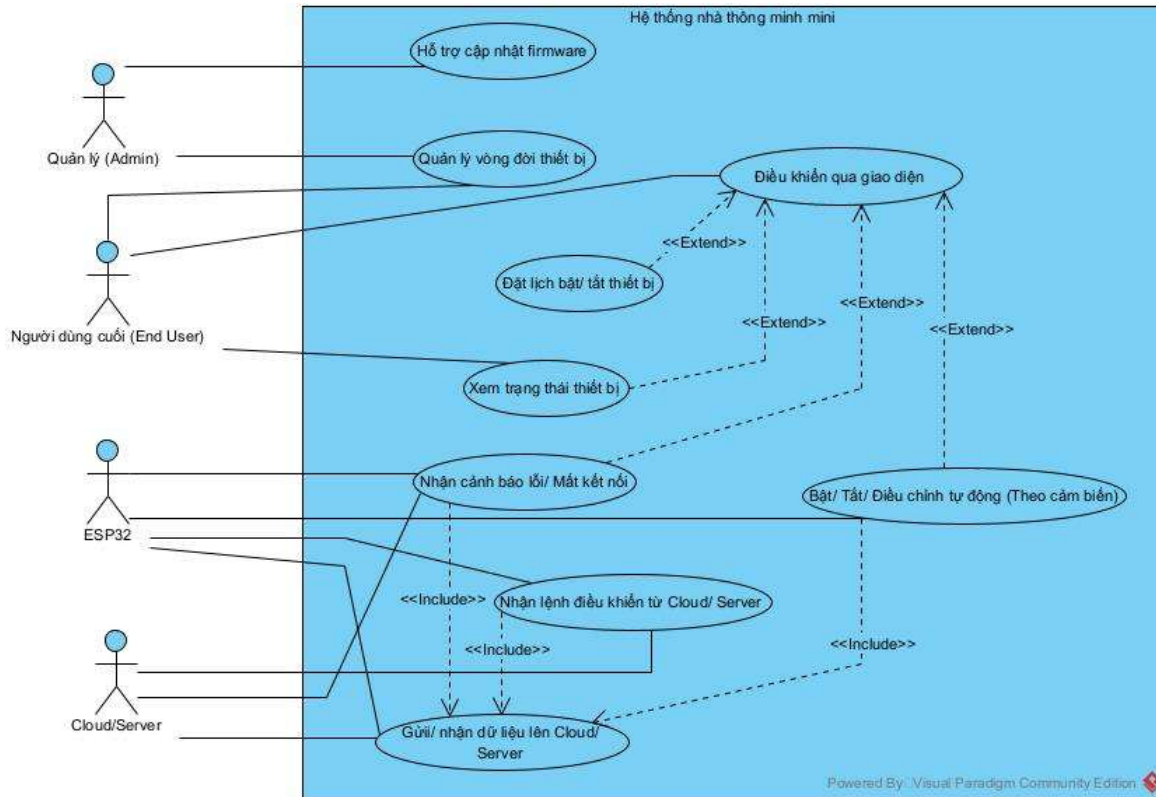
- Kiểm tra khả năng nhận dạng giọng nói và lệnh



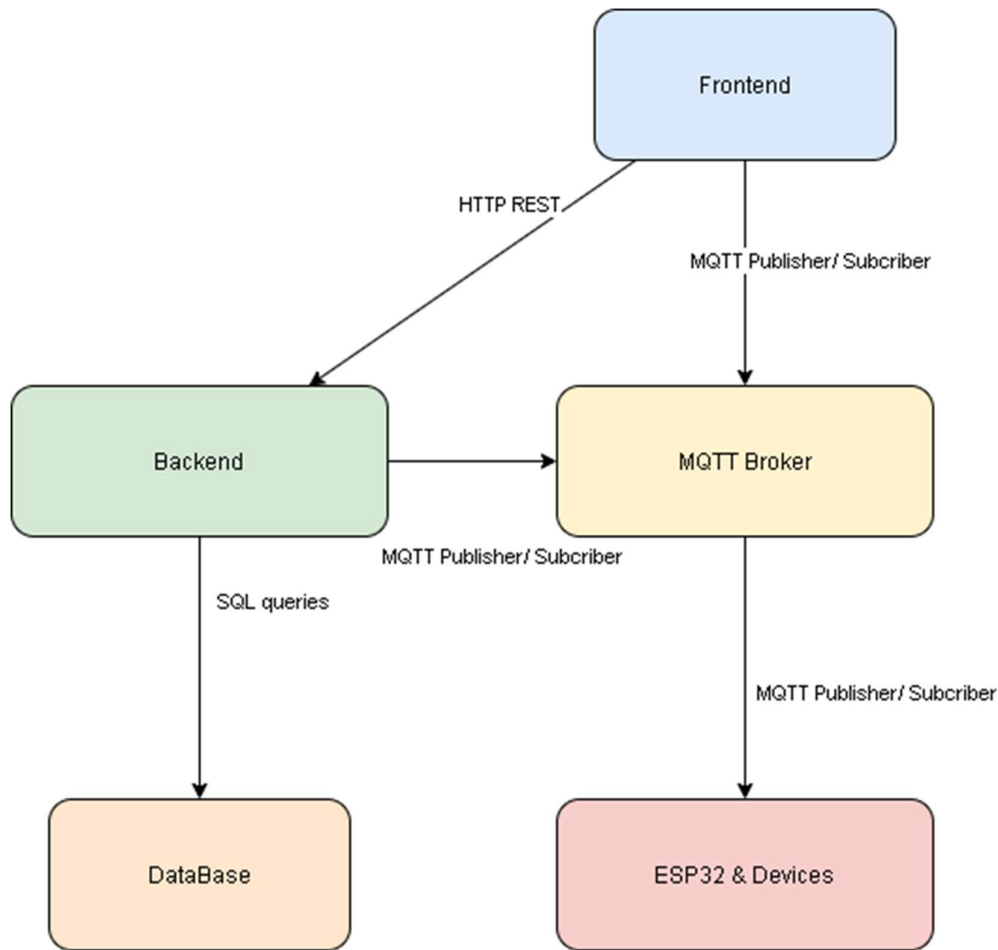
## Phần 4, Phân tích thiết kế hệ thống

### 4.1 Sơ đồ kiến trúc hệ thống và use - case tổng quan

Sơ đồ use - case tổng quan hệ thống:



Sơ đồ 3 lớp của kiến trúc hệ thống:



### Tầng 1 - Lớp trình bày

- Thành phần: Frontend (Web App ReactJS/Mobile App Android).
- Vai trò: Giao diện người dùng, hiển thị trạng thái và nhận lệnh từ người dùng.
- Giao tiếp: HTTP REST (với Backend), MQTT (với Broker).

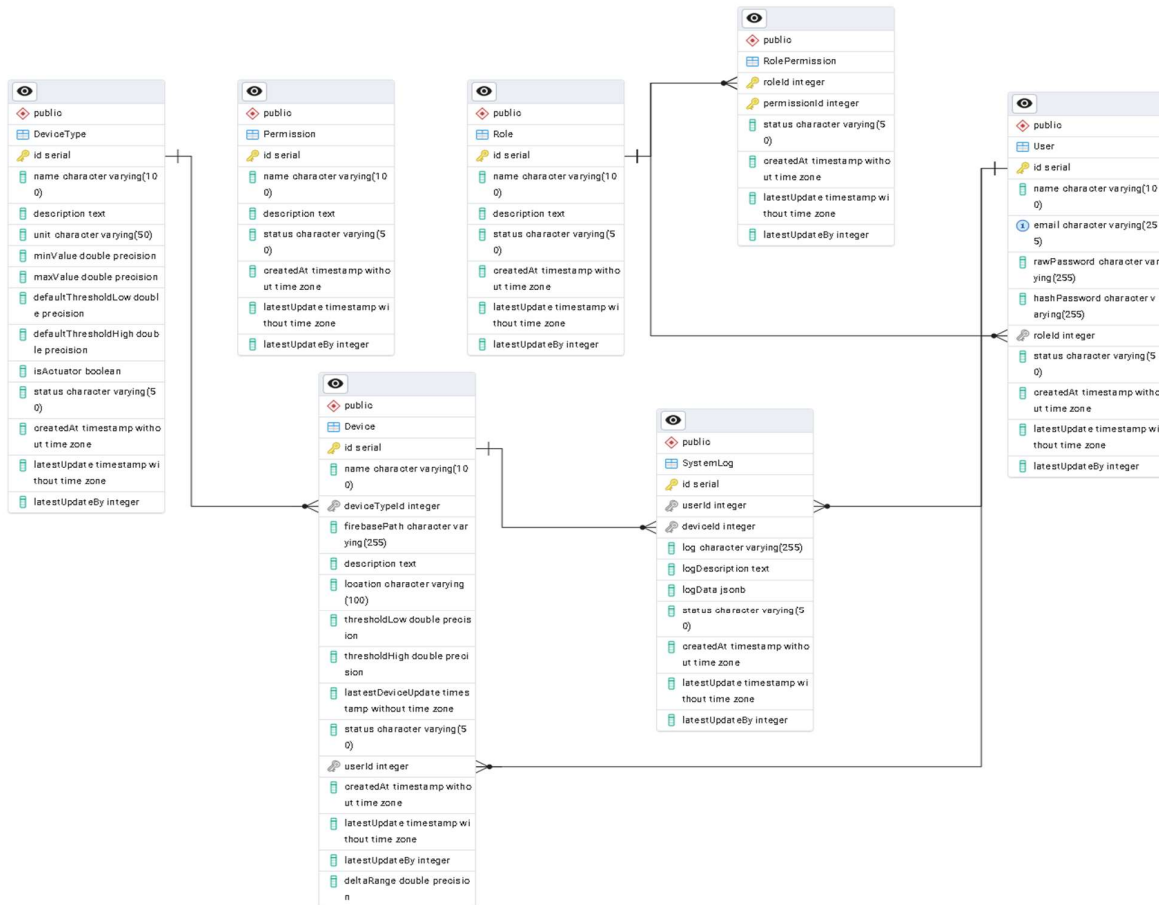
### Tầng 2 - Lớp logic/ ứng dụng

- Thành phần: Backend (NodeJS) và MQTT Broker (Dịch vụ giao tiếp).
- Vai trò: Xử lý toàn bộ logic nghiệp vụ và điều phối giao tiếp thời gian thực.
- Giao tiếp: SQL Queries (với Database), HTTP REST (với Frontend), MQTT (với tất cả các bên).

### Tầng 3 - Lớp dữ liệu

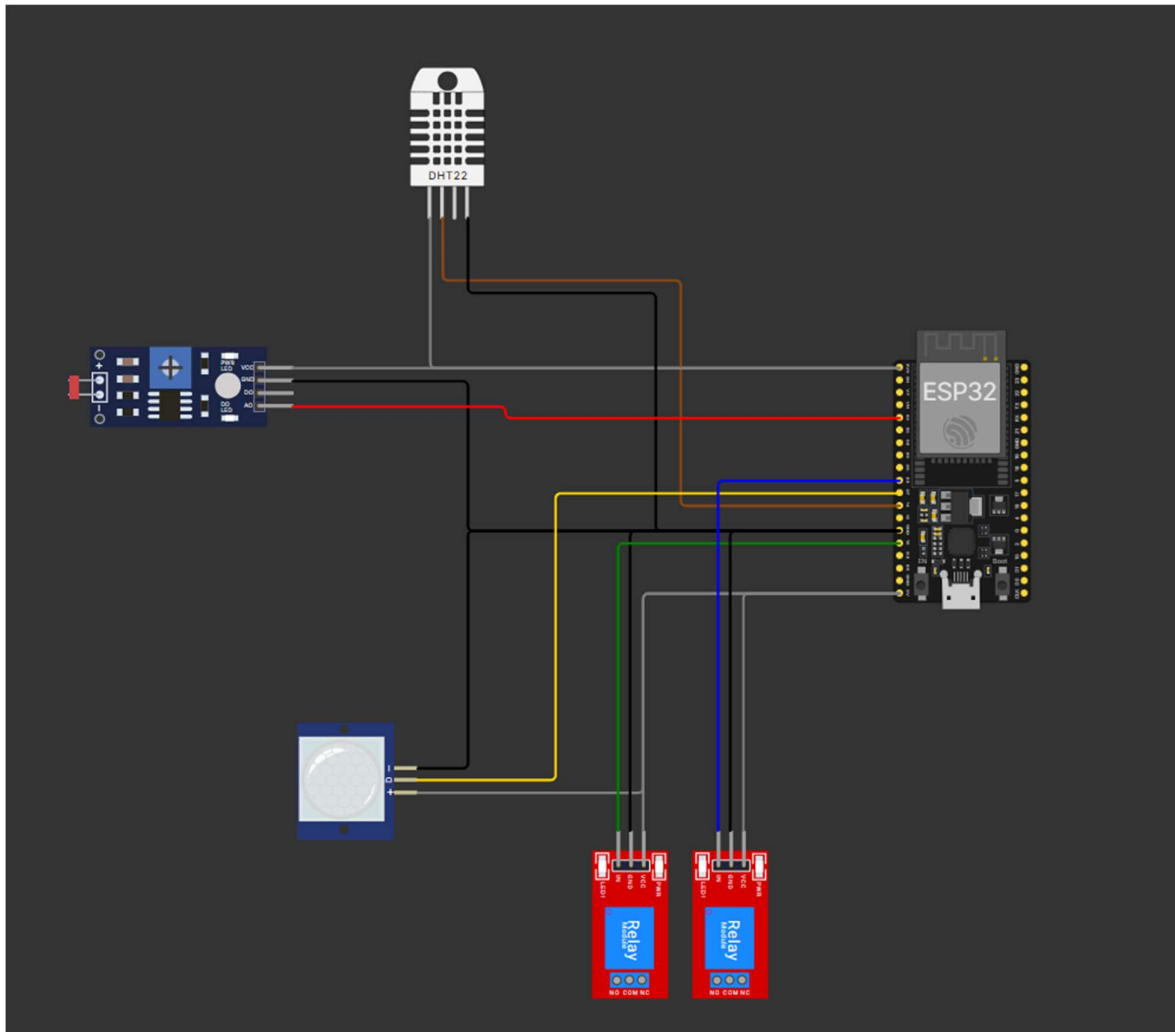
- Thành phần: DataBase (PostgreSQL/MongoDB) và ESP32 & Devices.
- Vai trò: Nơi lưu trữ dữ liệu cố định và nguồn cung cấp/Thực thi dữ liệu vật lý.
- Giao tiếp: SQL Queries (với Backend), MQTT (với Broker).

## 4.2 Thiết kế backend - database



Các bảng đáp ứng đủ cho nhu cầu phân quyền, thống kê phân tích và quản lý dữ liệu cũng như các thiết bị

## 4.3 ESP



Mô phỏng thiết kế mạch hiện trên Wokwi

DHT để đo nhiệt độ, độ ẩm, cổng DATA SDA  $\Leftrightarrow$  pin 14

Cảm biến ánh sáng, cổng DATA AO  $\Leftrightarrow$  pin 34

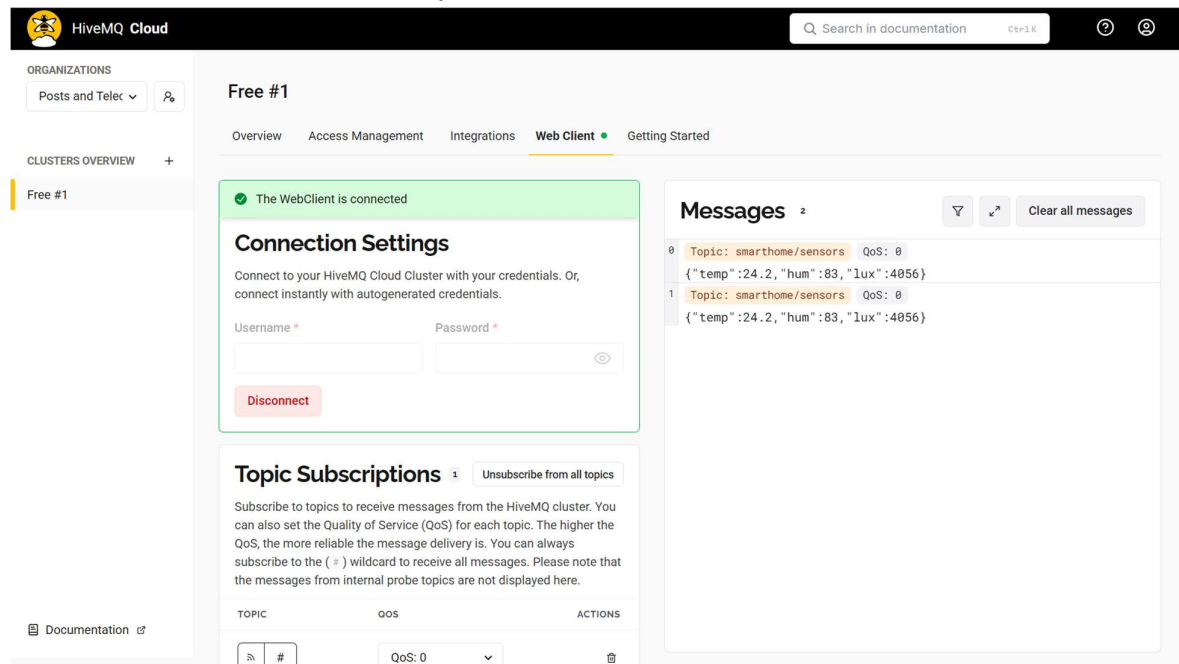
Cảm biến chuyển động, cổng OUT  $\Leftrightarrow$  pin 27

2 relay với cổng IN lần lượt nối tới pin 13 và 26

ESP32 sẽ kết nối trực tiếp với các cảm biến như DHT22, LDR, PIR để thu thập dữ liệu, và kết nối với các cơ cấu chấp hành - relay để điều khiển các thiết bị điện như đèn, quạt, v.v. được kết nối qua relay. Nhiệm vụ quan trọng nhất của nó là duy trì kết nối Wi-Fi và giao tiếp hai chiều liên tục với MQTT Broker.

Kết nối được ESP tới và gửi dữ liệu đến HiveMQ.

Backend có thể kết nối tới MQTT để trích xuất dữ liệu để hiển thị hoặc lưu trữ.



## 4.4 Ảnh một số endpoint kết nối và code

Endpoint kết nối (ví dụ)

Devices		
POST	/devices	Create a new item
GET	/devices	Get all items with optional pagination
GET	/devices/{id}	Get a single item by ID
PATCH	/devices/{id}	Update an existing item
DELETE	/devices/{id}	Delete an item by ID

## SystemLogs

**POST** /system-logs Create a new item

**GET** /system-logs Get all items with optional pagination

**GET** /system-logs/{id} Get a single item by ID

**PATCH** /system-logs/{id} Update an existing item

**DELETE** /system-logs/{id} Delete an item by ID

Kết quả trả về (Ví dụ)

```
{
  "id": 1,
  "status": "online",
  "createdAt": "2025-10-31T10:13:44.155Z",
  "latestUpdate": null,
  "latestUpdateBy": null,
  "name": "DHT11_LivingRoom",
  "firebasePath": "/devices/dht11_01/temp",
  "description": "Temperature sensor in the living room",
  "location": "Living Room",
  "thresholdLow": 15,
  "thresholdHigh": 35,
  "deltaRange": null,
  "lastestDeviceUpdate": null,
  "userId": 1,
  "deviceTypeId": 1
},
```

```
{
  "id": 1,
  "status": "active",
  "createdAt": "2025-10-31T10:13:44.155Z",
  "latestUpdate": null,
  "latestUpdateBy": null,
  "log": "SENSOR_UPDATE",
  "logDescription": "Temperature updated from 27.5°C to 28.5°C",
  "logData": {
    "new": 28.5,
    "old": 27.5,
    "unit": "°C"
  },
  "userId": 1,
  "deviceId": 1
},
```

## Phần 5, Đánh giá kết quả dự án

### 5.1 Ảnh các giao diện

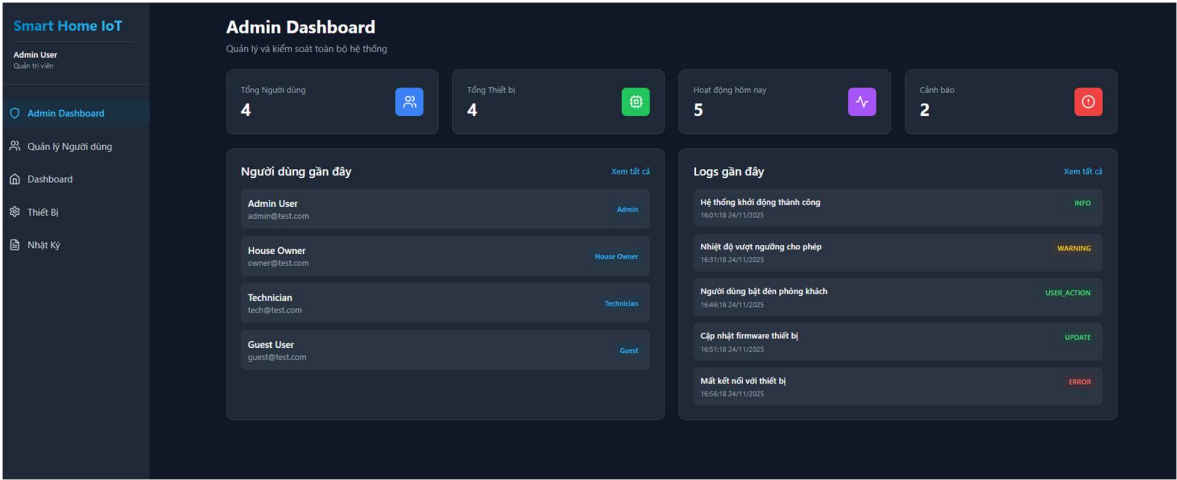
#### 5.1.1 Giao diện đăng nhập / Đăng ký

The image displays two screenshots of the 'Smart Home IoT' web application interface, set against a blue gradient background. Both screens feature a white house icon logo at the top center, with the text 'Smart Home IoT' and 'Nhà Thông Minh Mini' below it.

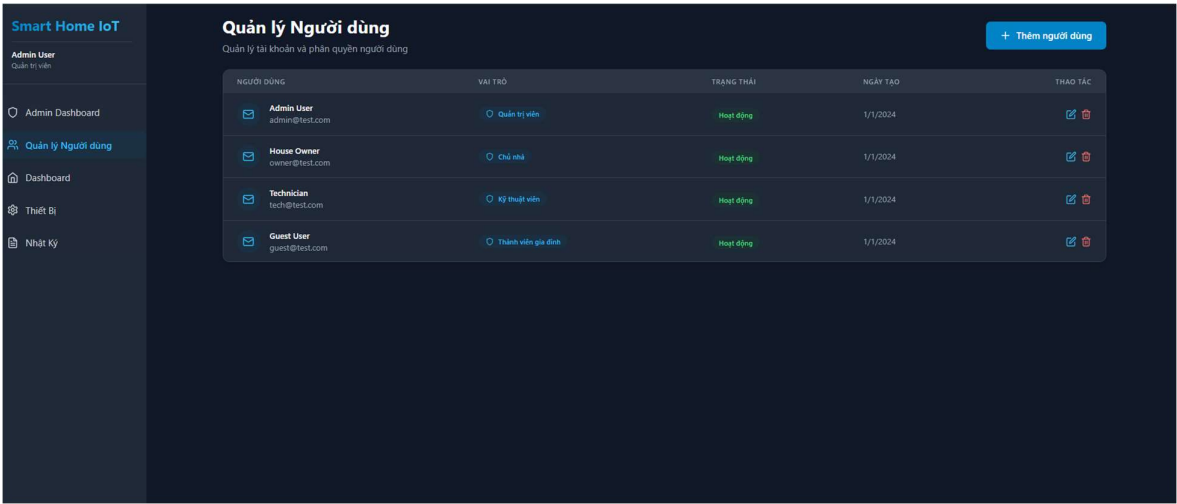
The top screenshot shows the 'Đăng nhập' (Login) form. It includes an 'Email' field with the placeholder 'admin@test.com', a 'Mật khẩu' (Password) field with masked characters, and a blue 'Đăng nhập' button. Below the button is a link that reads 'Chưa có tài khoản? Đăng ký ngay' (Don't have an account? Register now).

The bottom screenshot shows the 'Đăng ký' (Register) form. It includes a 'Tên' (Name) field with the placeholder 'Tên của bạn', an 'Email' field with the placeholder 'admin@test.com', a 'Mật khẩu' (Password) field, and a 'Xác nhận mật khẩu' (Confirm password) field. A blue 'Đăng ký' button is at the bottom, with a link below it that reads 'Đã có tài khoản? Đăng nhập' (Already have an account? Login).

5.1.2 Giao diện Dashboard Admin

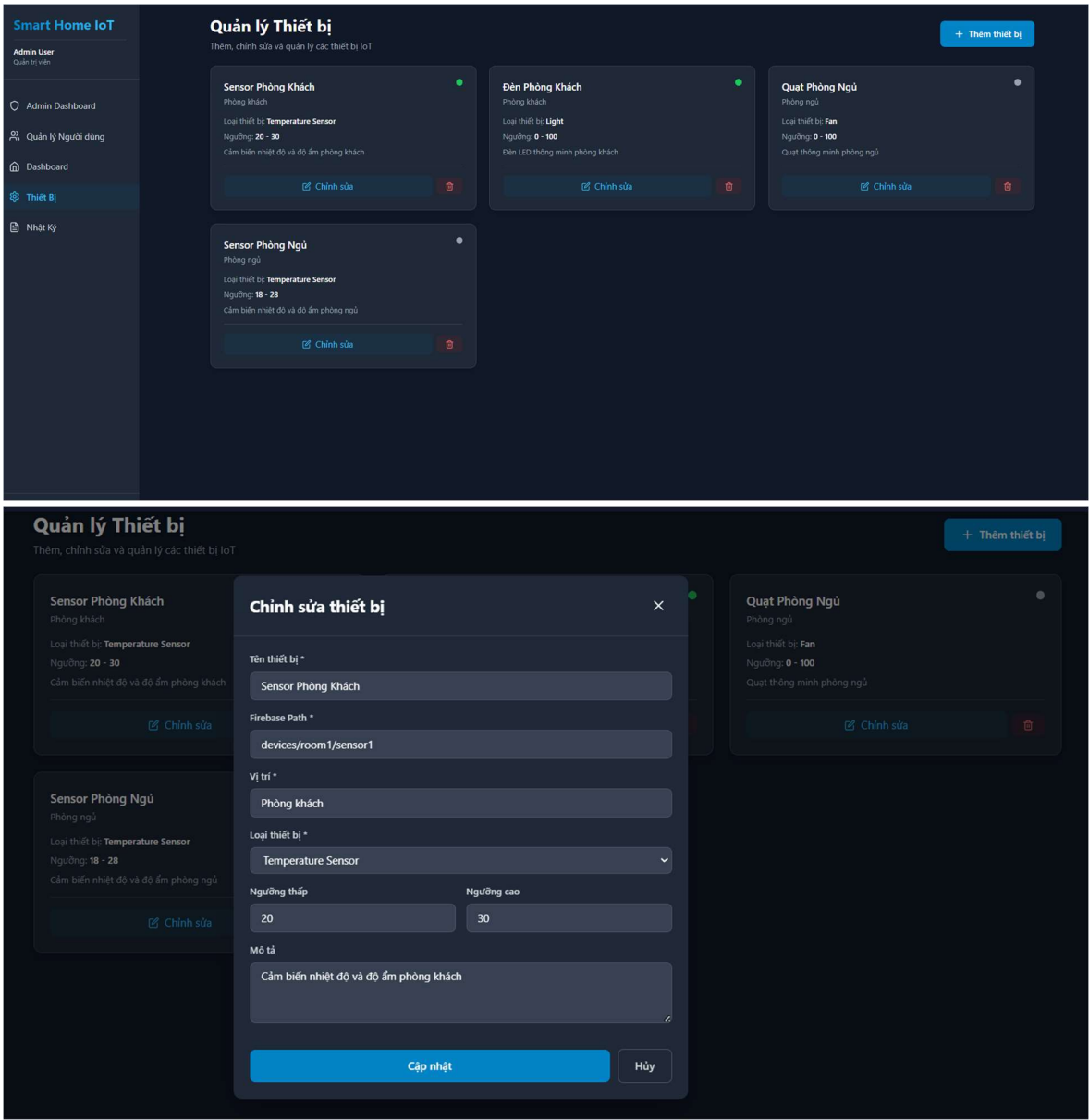


5.1.3 Giao diện quản lý người dùng

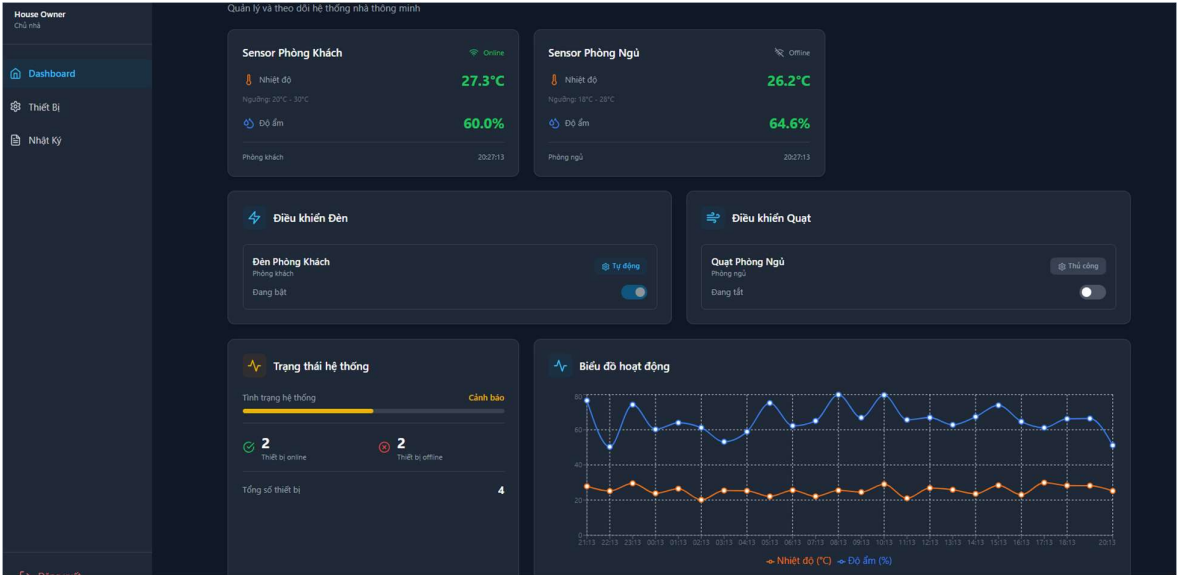




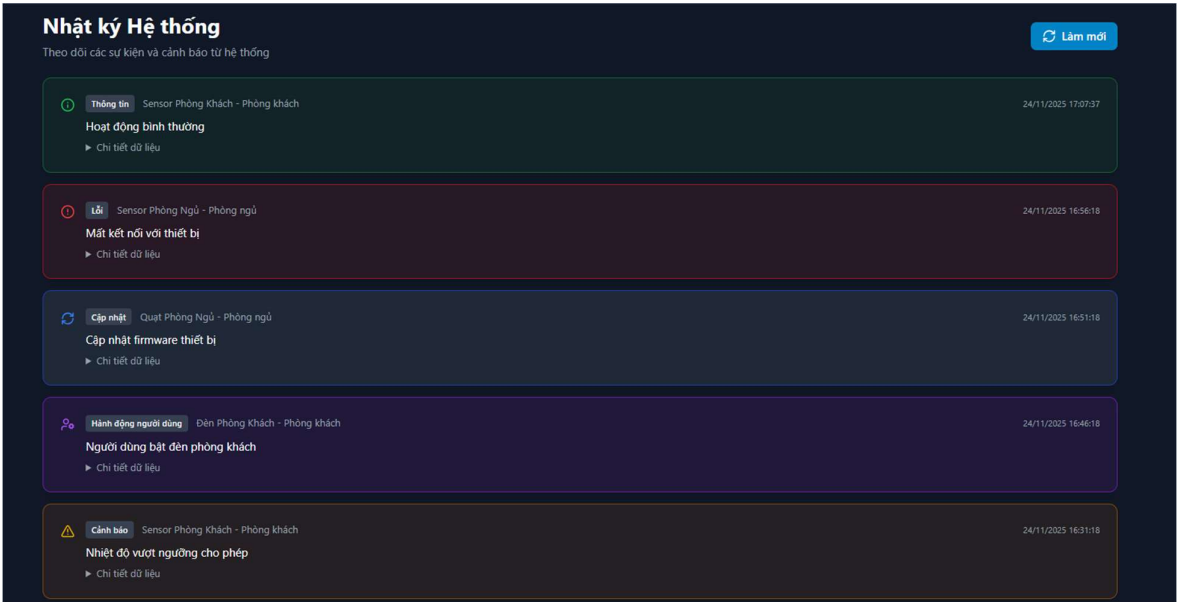
### 5.1.4 Giao diện quản lý thiết bị



5.1.5 Giao diện các chức năng (Đo nhiệt độ, độ ẩm, điều khiển thiết bị)



5.1.6 Giao diện theo dõi nhật ký hệ thống



### 5.1.7 Giao diện phân quyền điều khiển thiết bị cho người dùng

Phân quyền thành viên

Owner/Admin có thể phân công quyền điều khiển cho các thành viên

THÀNH VIÊN	VAI TRÒ	QUYỀN ĐIỀU KHIỂN	THIẾT BỊ ĐƯỢC PHÉP
<div>Admin User</div> <div>admin@test.com</div>	Admin	<input checked="" type="checkbox"/> Cho phép điều khiển giọng nói	<div>Sensor Phòng Khách • Phòng khách</div> <div>Đèn Phòng Khách • Phòng khách</div> <div>Quạt Phòng Ngủ • Phòng ngủ</div> <div>Sensor Phòng Ngủ • Phòng ngủ</div>
<div>House Owner</div> <div>owner@test.com</div>	House Owner	<input checked="" type="checkbox"/> Cho phép điều khiển giọng nói	<div>Chưa chọn thiết bị</div> <div>Quản lý thiết bị</div>
<div>Technician</div> <div>tech@test.com</div>	Technician	<input type="checkbox"/> Cho phép điều khiển giọng nói	<div>Sensor Phòng Khách • Phòng khách</div> <div>Đèn Phòng Khách • Phòng khách</div> <div>Quạt Phòng Ngủ • Phòng ngủ</div> <div>Sensor Phòng Ngủ • Phòng ngủ</div>
<div>Guest User</div> <div>guest@test.com</div>	Guest	<input type="checkbox"/> Cho phép điều khiển giọng nói	<div>Chưa chọn thiết bị</div> <div>Đóng</div> <div><div>Phòng khách</div><div><input type="checkbox"/> Sensor Phòng Khách</div><div><input type="checkbox"/> Đèn Phòng Khách</div><div>Phòng ngủ</div><div><input type="checkbox"/> Quạt Phòng Ngủ</div><div><input type="checkbox"/> Sensor Phòng Ngủ</div></div>

Quy tắc gợi ý

- Admin có toàn quyền truy cập mọi thiết bị
- Owner có thể giao quyền điều khiển cho thành viên
- Technician chỉ nên có quyền OTA và thiết bị cần bảo trì
- Guest chỉ nên được phép điều khiển những thiết bị cụ thể

Mẹo phân quyền

Phân quyền rõ ràng giúp đảm bảo an toàn hệ thống. Bạn có thể điều chỉnh thiết bị được phép điều khiển cho từng thành viên để tránh thao tác nhầm. Mọi thay đổi sẽ được ghi lại trong logs.

### 5.1.8 Giao diện cập nhật firmware OTA

Cập nhật Firmware OTA

Đẩy firmware mới cho các thiết bị thông minh từ xa

Cập nhật OTA cho Đèn Phòng Khách thành công

THIẾT BỊ	VỊ TRÍ	FIRMWARE	TIẾN TRÌNH	THAO TÁC
<div>Sensor Phòng Khách</div> <div>Temperature Sensor</div>	Phòng khách	<div>v2.4</div> <div>Mới nhất</div>	<div></div> 100%	<div>Cập nhật OTA</div>
<div>Đèn Phòng Khách</div> <div>Light</div>	Phòng khách	<div>v1.3</div> <div>Mới nhất</div>	<div></div> 100%	<div>Cập nhật OTA</div>
<div>Quạt Phòng Ngủ</div> <div>Fan</div>	Phòng ngủ	<div>v1.0.0</div>	<div></div> 0%	<div>Cập nhật OTA</div>
<div>Sensor Phòng Ngủ</div> <div>Temperature Sensor</div>	Phòng ngủ	<div>v1.0.0</div>	<div></div> 0%	<div>Cập nhật OTA</div>

### 5.1.9 Giao diện điều khiển bằng giọng nói

Điều khiển bằng giọng nói

Ra lệnh cho thiết bị thông minh bằng tiếng Việt thời gian thực

Trạng thái

Sẵn sàng

Bắt đầu

Dừng

Lệnh vừa nói

Bật đèn phòng khách.

Hành động hệ thống

Đã bật đèn theo yêu cầu

Lệnh gợi ý

Thử các câu lệnh mẫu sau để điều khiển thiết bị:

Bật đèn phòng khách

Tắt quạt phòng ngủ

Tăng tốc độ quạt

Giảm nhiệt độ xuống 25 độ

## **5.2. ĐÁNH GIÁ KẾT QUẢ VÀ CHẤT LƯỢNG HỆ THỐNG**

### **5.2.1. Đánh giá mức độ hoàn thiện chức năng**

#### **a. Hệ thống điều khiển thiết bị điện**

Hệ thống cho phép điều khiển toàn bộ thiết bị điện trong mô hình như đèn, quạt, cửa, còi báo... thông qua 2 phương thức:

- Điều khiển thủ công bằng công tắc hoặc nút nhấn.
- Điều khiển từ xa thông qua ứng dụng Web, hoặc giao diện IoT.

Kết quả thử nghiệm:

- Tốc độ phản hồi trung bình: 0.3 – 1 giây, nhanh và ổn định.
- Tỷ lệ thực thi đúng lệnh: 95–100%.
- Không xuất hiện tình trạng treo thiết bị hoặc phải reset vi điều khiển.
- Các thiết bị được phân vùng rõ ràng, tránh giao thoa tín hiệu.

#### **b. Hệ thống cảm biến**

- Cảm biến nhiệt độ, độ ẩm: đo giá trị tương đối chính xác, phù hợp cho tác vụ điều khiển quạt hoặc cảnh báo môi trường.
- Cảm biến chuyển động PIR: hoạt động ổn định ở phạm vi 3–5 m

Hầu hết các cảm biến vận hành ổn định theo yêu cầu của mô hình. Độ ổn định duy trì tốt trong suốt quá trình thử nghiệm liên tục.

#### **c. Giao diện điều khiển (Web/App)**

- Giao diện được thiết kế trực quan, dễ sử dụng.
- Các thông tin hiển thị đầy đủ: trạng thái thiết bị, giá trị cảm biến, các nút điều khiển.
- Giao diện đẹp, rõ ràng, giúp người dùng dễ dàng thao tác mà không cần hướng dẫn nhiều.

#### **d. Hoạt động của mô hình nhà**

- Bố cục các phòng, dây điện và thiết bị trong mô hình được bố trí hợp lý, giúp người xem dễ quan sát nguyên lý hoạt động của nhà thông minh.

- Mô hình thể hiện rõ mối liên hệ giữa các thành phần: cảm biến – thiết bị – bộ điều khiển – giao diện.

### **5.2.2. Tính năng thông minh và tự động hóa**

#### **a. Tính năng tự động theo cảm biến**

- Tự động bật quạt khi nhiệt độ > giá trị đặt trước.
- Tự động bật đèn khi có người đi vào phòng (PIR kích hoạt).
- Kích hoạt còi báo động khi phát hiện chuyển động bất thường (chế độ an ninh).

#### **b. Xử lý song song và thời gian thực**

- Bộ vi điều khiển (như Arduino / ESP8266 / ESP32) phản hồi liên tục trong thời gian thực.
- Tốc độ cập nhật dữ liệu cảm biến ổn định, dao động từ 0.5–2 giây mỗi chu kỳ.
- Không xảy ra hiện tượng “delay quá lâu” hay “nghẽn lệnh”.

#### **c. Hệ thống hoạt động độc lập + điều khiển từ xa**

- Hoạt động độc lập (Offline): Khi offline, hệ thống vẫn đảm bảo 1 phần hoạt động của các thiết bị và có gửi tin nhắn cuối trước khi offline. Nhờ đó, mô hình vẫn đảm bảo an toàn và tự động hóa liên tục, không bị ảnh hưởng quá nhiều bởi sự cố mạng.
- Điều khiển từ xa (Online): Khi có kết nối WiFi, người dùng có thể điều khiển toàn bộ hệ thống từ xa thông qua giao diện web (bật tắt đèn, quạt, xem dữ liệu cảm biến, kích hoạt báo động...). Lệnh điều khiển được gửi qua Internet đến vi điều khiển và thực thi ngay lập tức.

### **5.2.3. Hiệu năng và mức độ ổn định**

- Hệ thống duy trì hoạt động liên tục trong nhiều giờ mà không gặp tình trạng treo hoặc lỗi.
- Việc truyền nhận dữ liệu giữa cảm biến – vi điều khiển – ứng dụng ổn định.
- Mạch điện được bố trí hợp lý, hạn chế nhiễu và hao tổn năng lượng không cần thiết.

#### 5.2.4. Đánh giá tổng quan chất lượng

<b>Tiêu chí</b>	<b>Đánh giá</b>
Hiệu năng hệ thống	Tốt
Độ chính xác cảm biến	Khá – tùy thuộc từng loại cảm biến giá rẻ
Khả năng tự động hóa	Hoàn chỉnh ở mức cơ bản
Khả năng mở rộng	Rất cao
Tính trực quan mô hình	Rõ ràng, phù hợp trưng bày/giảng dạy
Tính thực tế của ứng dụng	Cao, mô phỏng gần với nhà thông minh thực tế

## **Phần 6, Kết luận và định hướng tương lai**

### **6.1. Kết luận**

Dự án **Mô hình nhà thông minh mini** đã được xây dựng thành công với đầy đủ các chức năng quản lý và giám sát thiết bị trong ngôi nhà. Hệ thống bao gồm các cảm biến, bộ điều khiển, thiết bị điện và giao diện điều khiển hoạt động đồng bộ, ổn định.

Sản phẩm mô phỏng chính xác nguyên lý hoạt động của các hệ thống IoT trong lĩnh vực nhà thông minh hiện nay. Đây là nền tảng quan trọng giúp nhóm hiểu rõ các kỹ thuật lập trình nhúng, xử lý tín hiệu, giao tiếp không dây và thiết kế mô hình thực tế.

Có thể khẳng định rằng mô hình đã đạt được mục tiêu đề ra, mang lại tính trực quan, tính ứng dụng cao, dễ mở rộng và dễ cải tiến về mặt công nghệ.

### **6.2. Định hướng phát triển trong tương lai**

#### **a. Nâng cấp công nghệ**

- Tích hợp trí tuệ nhân tạo (AI/ML) để dự đoán thói quen người dùng.
- Sử dụng ESP32 để tăng hiệu suất và hỗ trợ Bluetooth.
- Tạo một ứng dụng di động (Android/iOS) quản lý toàn bộ hệ thống.
- Áp dụng Firebase / MQTT để điều khiển từ xa qua Internet toàn cầu.

#### **b. Nâng cấp phần cứng**

- Thêm cảm biến chất lượng không khí (PM2.5), cảm biến mưa, cảm biến cửa từ.
- Tích hợp camera AI (ESP32-CAM) nhận diện người hoặc cảnh báo xâm nhập.
- Dùng nguồn ổn định hơn, có thể thêm mạch bảo vệ quá dòng.

#### **c. Tăng cường tính bảo mật**

- Áp dụng mã hóa tín hiệu điều khiển bằng SSL/TLS
- Hệ thống cảnh báo đa nền tảng: gửi thông báo qua Zalo, Telegram hoặc Email.

- Giảm nguy cơ truy cập trái phép bằng mã PIN/OTP.

#### **d. Mở rộng mô hình thành hệ sinh thái thông minh**

- Tích hợp khu vườn thông minh: tự động tưới, đo độ ẩm đất, theo dõi thời tiết.
- Bổ sung hệ thống năng lượng mặt trời mini để làm mô hình tiết kiệm năng lượng.
- Điều khiển giọng nói qua Google Assistant hoặc Amazon Alexa.

#### **e. Định hướng ứng dụng thực tế**

Nếu mở rộng quy mô, mô hình có thể trở thành:

- Hệ thống giám sát nhà ở giá rẻ.
- Bộ kit giảng dạy IoT cho sinh viên.
- Mô hình trình diễn trong các hội thi khoa học kỹ thuật.
- Nền tảng cho phát triển nhà thông minh quy mô thực tế.



# TÀI LIỆU THAM KHẢO

[Technical Documents | Espressif Systems](#)

[Firebase Documentation](#)

[MQTT - The Standard for IoT Messaging](#)

[Cập nhật firmware từ xa cho esp8266/esp32 \(OTA\) qua web server - Tự học IoT, Lập trình nhúng](#)

[ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE | Random Nerd Tutorials](#)