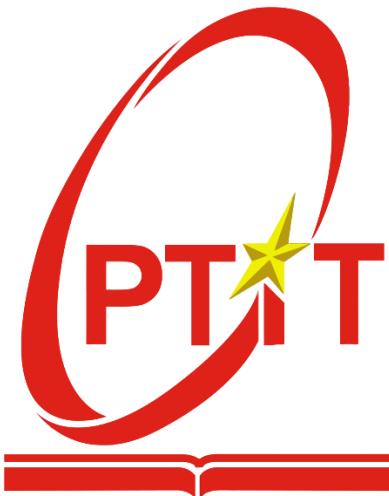


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



**BÁO CÁO CUỐI KỲ  
HỌC PHẦN: IOT VÀ ỨNG DỤNG**

**Đề tài: Thiết kế hệ thống báo cháy**

**Giảng viên hướng dẫn: Thầy Kim Ngọc Bách**

**Lớp: D22CNPM01 (D22 - N06)**

**Nhóm sinh viên thực hiện: Nhóm 08**

Sinh viên	Mã sinh viên
Dào Đức Duy	B22DCCN145
Ong Thế Minh	B22DCCN541
Đỗ Gia Phong	B22DCCN613
Phan Thị Hồng Thắm	B22DCCN806

Hà Nội, tháng 11 năm 2025

## MỤC LỤC

<b>MỤC LỤC.....</b>	<b>1</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>3</b>
<b>DANH MỤC BẢNG.....</b>	<b>3</b>
<b>CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI .....</b>	<b>4</b>
1.1    Mô tả dự án.....	4
1.2    Mục đích và ý nghĩa của dự án.....	4
1.3    Tổng quan phương hướng của dự án .....	5
1.4    Các thiết bị sử dụng trong hệ thống.....	5
1.4.1    Phần cứng .....	5
1.4.2    Phần mềm .....	12
<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....</b>	<b>14</b>
2.1    Khái niệm về IoT (Internet of Things).....	14
2.2    Tầm quan trọng của IoT trong cảnh báo cháy nổ .....	14
2.3    Cấu trúc của hệ thống báo cháy .....	15
2.3.1    Cấu trúc chung của một hệ thống IoT .....	15
2.3.2    Cấu trúc hệ thống của dự án .....	16
<b>CHƯƠNG 3: PHÂN TÍCH YÊU CẦU.....</b>	<b>17</b>
3.1    Mục tiêu của hệ thống.....	17
3.2    Phạm vi của hệ thống.....	17
3.3    Tiêu chí thành công .....	17
3.4    Yêu cầu từ các bên liên quan .....	18
3.5    Yêu cầu chức năng.....	19
3.5.1    Các chức năng của hệ thống .....	19
3.5.2    Sơ đồ Usecase toàn hệ thống .....	19
3.6    Yêu cầu phi chức năng .....	19
3.7    Ràng buộc .....	20
3.7.1    Môi trường hoạt động.....	20

3.7.2	Ràng buộc pháp lý.....	21
3.8	Mô hình hóa yêu cầu.....	22
3.9	Xác minh và phê duyệt .....	22
3.9.1	Xem xét lại yêu cầu với tất cả bên liên quan:.....	22
3.9.2	Ưu tiên hóa yêu cầu:.....	23
<b>CHƯƠNG 4:</b>	<b>TRIỂN KHAI KẾ HOẠCH.....</b>	<b>24</b>
4.1	Các công việc chính cần thực hiện .....	24
4.2	Bảng phân chia công việc .....	24
4.3	Thiết kế hệ thống .....	25
4.3.1	Sơ đồ mạch thiết bị.....	25
4.3.2	Sơ đồ Use Case hệ thống.....	27
4.3.3	Sơ đồ lớp thực thể toàn hệ thống.....	27
4.3.4	Sơ đồ cơ sở dữ liệu toàn hệ thống .....	29
4.3.5	Sơ đồ luồng dữ liệu (DFD).....	30
4.4	Kết quả đạt được .....	31
4.5	Kết luận và phương hướng mở rộng.....	32

## **DANH MỤC HÌNH ẢNH**

Hình 1.1: Cảm biến lửa.....	5
Hình 1.2: Cảm biến khói MP2.....	7
Hình 1.3: Sơ đồ chân của Vi xử lý ESP32 WROOM 32 TYPE-C 30PIN CP2102.....	8
Hình 1.4: Cảm biến nhiệt/độ ẩm DHT22 .....	9
Hình 1.5: Còi báo cháy .....	10
Hình 1.6: Máy bơm 5V.....	11
Hình 1.7: Relay 5V.....	11
Hình 2.1: Sơ đồ cấu trúc của một hệ thống IoT.....	15
Hình 3.1: Sơ đồ Usecase toàn hệ thống .....	19
Hình 3.2: Mô hình 4 lớp IoT .....	22
Hình 4.1: Sơ đồ mạch nối thiết bị.....	25
Hình 4.2: Sơ đồ Use case toàn hệ thống.....	27
Hình 4.3: Sơ đồ lớp thực thể toàn hệ thống.....	27
Hình 4.4: Thiết kế cơ sở dữ liệu của hệ thống.....	29
Hình 4.5: Sơ đồ luồng dữ liệu của hệ thống .....	30
Hình 4.6: Giao diện hiển thị của ứng dụng cảnh báo cháy .....	31

## **DANH MỤC BẢNG**

Bảng 3.1: Bảng liệt kê các chức năng của hệ thống theo vai trò người dùng.....	19
Bảng 4.1: Bảng phân công công việc cho thành viên nhóm.....	24
Bảng 4.2: Bảng chân nối chi tiết giữa các cảm biến, tài và ESP32 .....	26
Bảng 4.3: Bảng mô tả chi tiết quan hệ giữa các bảng trong cơ sở dữ liệu.....	30

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## 1.1 Mô tả dự án

Hệ thống báo cháy, hay fire alarm system, là một hệ thống được thiết kế để phát hiện sự xuất hiện của lửa, khói hoặc nhiệt độ có thể gây cháy trong một khu vực cụ thể và thông báo ngay lập tức để người ta có thể thực hiện các biện pháp an toàn. Hệ thống này chủ yếu bao gồm các cảm biến, thiết bị báo động và một trung tâm điều khiển.

Một hệ thống báo cháy có một số lượng thiết bị hoạt động cùng nhau để phát hiện và cảnh báo mọi người thông qua các thiết bị nghe nhìn khi khói, lửa, khí CO<sub>2</sub> hoặc các trường hợp khẩn cấp khác.

Hiệu quả của hệ thống phát hiện và báo cháy phụ thuộc vào giai đoạn cháy mà hệ thống đó được vận hành. Để tất cả những người cư ngụ có thể thoát ra ngoài mà không gặp quá nhiều khó khăn, cần báo động sớm trước khi các lối thoát hiểm bị khói dày đặc đến mức khiến những người cư ngụ gặp khó khăn trong việc tìm đường ra khỏi tòa nhà.

Hệ thống báo cháy được sử dụng để bảo vệ tính mạng và tài sản. Nó bảo vệ bằng cách:

- (1) Phát hiện đám cháy ở giai đoạn đầu.
- (2) Cảnh báo cho người cư trú.
- (3) Thông báo cho các nhân sự có liên quan.

## 1.2 Mục đích và ý nghĩa của dự án

Đề tài “Hệ thống cảnh báo cháy sử dụng IoT” được thực hiện nhằm giúp người học nắm vững cấu trúc phần cứng, sơ đồ khói và nguyên lý hoạt động của mạch điều khiển. Thông qua việc phân tích và tìm hiểu chi tiết từng khói chức năng, bài tập lớn hướng đến mục tiêu cung cấp kiến thức về thiết kế hệ thống điều khiển tự động và ứng dụng các linh kiện điện tử trong thực tế.

Bên cạnh đó, đề tài tập trung nghiên cứu về lập trình Arduino, qua đó giúp hiểu rõ cách thức giao tiếp giữa vi điều khiển với các cảm biến và thiết bị ngoại vi. Việc tìm hiểu và thực hành lập trình Arduino góp phần nâng cao khả năng tư duy lập trình, đồng thời tạo nền tảng cho việc phát triển các ứng dụng IoT khác trong tương lai.

Thông qua quá trình thực hiện, bài tập lớn giúp rèn luyện kỹ năng triển khai một dự án kỹ thuật hoàn chỉnh, bao gồm các bước từ thiết kế phần cứng, lập trình phần mềm cho đến kiểm thử và đánh giá hoạt động của hệ thống. Đây là cơ sở quan trọng để áp dụng vào các đồ án hoặc đề tài nghiên cứu có quy mô lớn hơn sau này.

Sản phẩm của đề tài là mô hình hệ thống cảnh báo cháy hoạt động ổn định, tích hợp đầy đủ các chức năng cần thiết như phát hiện khói hoặc nhiệt độ bất thường, kích hoạt còi báo động và gửi cảnh báo đến người

dùng thông qua nền tảng IoT. Hệ thống thể hiện tính ứng dụng cao, đồng thời minh chứng cho khả năng kết hợp giữa lý thuyết và thực hành trong lĩnh vực điện – điện tử và công nghệ IoT.

### 1.3 Tổng quan phương hướng của dự án

Dự án được định hướng phát triển theo mô hình hệ thống cảnh báo cháy thông minh ứng dụng công nghệ IoT, có khả năng giám sát, phát hiện và cảnh báo sớm các nguy cơ cháy nổ. Phương hướng triển khai tập trung vào tích hợp các thiết bị cảm biến và vi điều khiển ESP32 để thu thập, xử lý dữ liệu và kết nối với cơ sở dữ liệu

Về tổng thể, dự án được chia thành hai phần chính:

(1) Phần cứng: Bao gồm các cảm biến (khói, lửa, nhiệt độ, độ ẩm), vi điều khiển ESP32, còi báo cháy và máy bơm mini. Các thiết bị này phối hợp với nhau để phát hiện và phản ứng với các yếu tố nguy hiểm như nhiệt độ tăng cao hoặc có khói, lửa.

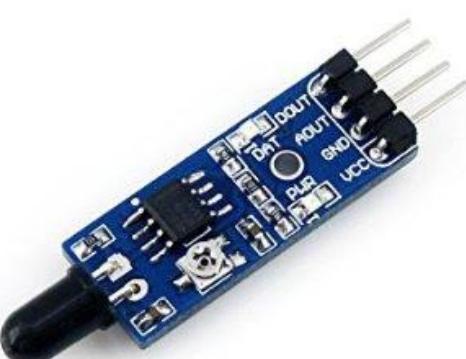
(2) Phần mềm: Gồm chương trình điều khiển viết trên Arduino IDE, cơ sở dữ liệu để lưu trữ và MQTT Broker để làm cầu nối giao tiếp giữa thiết bị và server, ứng dụng Android (Kotlin) phục vụ việc hiển thị và nhận cảnh báo, cùng API Backend (Java) làm cầu nối giao tiếp giữa ứng dụng và cơ sở dữ liệu (PostgreSQL).

Phương hướng phát triển của dự án hướng tới việc tạo ra một hệ thống hoạt động ổn định, dễ mở rộng và có tính ứng dụng cao trong thực tế. Trong tương lai, hệ thống có thể được nâng cấp để hỗ trợ thêm nhiều cảm biến, chức năng tự động xử lý (như kích hoạt phun nước), hoặc áp dụng trí tuệ nhân tạo (AI) nhằm dự đoán và ngăn chặn nguy cơ cháy sớm hơn.

### 1.4 Các thiết bị sử dụng trong hệ thống

#### 1.4.1 Phần cứng

##### 1.4.1.1 Cảm biến lửa



Hình 1.1: Cảm biến lửa

(1) Thông số:

- Nguồn cấp: 3.3V – 5VDC
- Dòng tiêu thụ: 15mA
- Tín hiệu ra: Digital 3.3 – 5VDC tùy nguồn cấp hoặc Analog.
- Khoảng cách: 80cm
- Góc quét: 60 độ
- Kích thước: 3.2 x 1.4 cm

(2) Sơ đồ chân:

- VCC → 3.3V ~ 5.3V
- GND → power supply ground
- AOUT (AO) → analog output
- DOUT (DO) → digital output

(3) Nguyên lý hoạt động:

- Ngọn lửa khi cháy sẽ phát ra các loại bức xạ, trong đó chủ yếu là tia hồng ngoại (IR) hoặc tia tử ngoại (UV).
  - Cảm biến lửa có các thành phần nhạy cảm với một trong hai loại tia này (thường là hồng ngoại). Khi có ngọn lửa xuất hiện, cảm biến sẽ phát hiện mức độ của những tia này
  - Khi mức bức xạ vượt ngưỡng, cảm biến sẽ kích hoạt tín hiệu HIGH (hoặc tín hiệu cụ thể tùy loại cảm biến) để báo hiệu có lửa.

#### 1.4.1.2 Cảm biến khói/khí gas MP-2



Hình 1.2: Cảm biến khói MP2

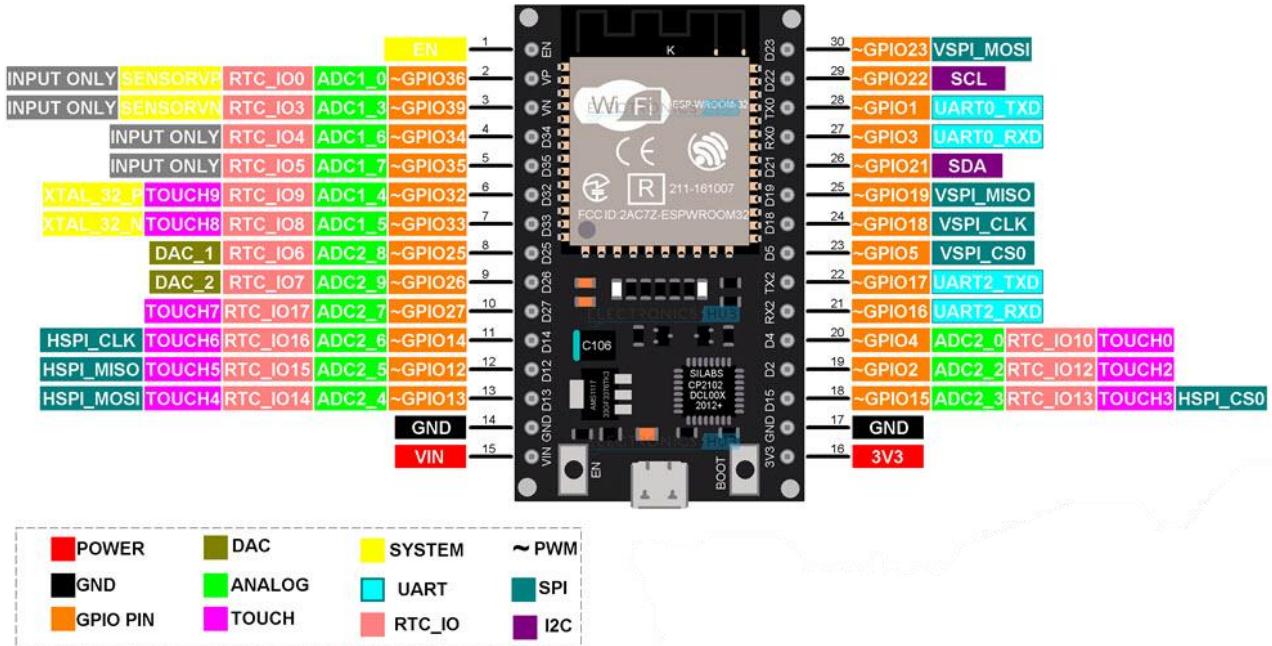
(1) Thông số kỹ thuật:

- Nguồn hoạt động: 5V
- Loại dữ liệu: Analog
- Dòng điện: 180mA
- Công suất: 900mW
- Chuẩn truyền: Digital & Analog (A0)
- Dãy hoạt động: 300ppm – 10000ppm
- Số chân: 4

(2) Nguyên lý hoạt động

- Cảm biến khói/khí gas MP2 sử dụng phân tử SnO<sub>2</sub> có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của SnO<sub>2</sub> sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện.
  - Cảm biến khí gas MP2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mêtan (CH<sub>4</sub>) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau.
  - Cảm biến xuất ra cả hai dạng tín hiệu là Analog và Digital, tín hiệu Digital có thể điều chỉnh mức báo bằng biến trở.

#### 1.4.1.3 ESP32 WROOM 32 TYPE-C 30PIN CP2102



Hình 1.3: Sơ đồ chân của Vi xử lý ESP32 WROOM 32 TYPE-C 30PIN CP2102

ESP32 WROOM 32 type-c 30pin CP2102 được tích hợp anten và RF, hoạt động tiết kiệm năng lượng, ổn định, chống nhiễu tốt, đây là giải pháp chi phí thấp nhất cho 1 dự án với một mạch sử dụng wifi 2.4Ghz và bluetooth TSMC công nghệ 40nm năng lượng thấp.

Đây là module NodeMCU dựa trên ESP32, có kết nối WiFi + Bluetooth, CP2102 trên bo mạch và các nút bấm. Hơn nữa, tất cả các chân I/O của mô-đun ESP-WROOM-32 đều có thể truy cập được thông qua GPIO.

Nhờ tài nguyên nguồn mở phong phú, NodeMCU ESP32-S hỗ trợ phát triển theo nhiều cách khác nhau như lệnh Lua / AT commands/ mã nguồn MicroPython / Arduino / IOT, v.v. giúp bạn nhanh chóng tạo các ứng dụng prototype IoT.

- (1) Thông số cơ bản ESP32 WROOM 32 TYPE-C 30PIN CP2102
  - Điện áp nguồn (USB): 5V DC
  - Đầu vào/Đầu ra điện áp: 3.3V DC
  - Dòng điện: 5µA trong hệ thống treo chế độ
  - In/OUT: 24
  - Mô hình: ESP32 30 chân
  - Thạch anh: 240Mhz
  - Các chuẩn giao tiếp: I2C, SPI, UART / USART, USB, CAN

- Nhiệt độ hoạt động: -40°C ~ 85°C

- Cổng ADC: 12bit - 18 Kênh

- Chip USB-Serial: CP2102

- Timer: Có

(2) Thông số Wifi và Bluetooth

- Loại: Wifi + Bluetooth Module

- Hiệu suất: Lên đến 600 DMIPS

- Tân số: lên đến 240MHz

- Wifi: 802.11 B/g/n/E/I (802.11N - 2.4 GHz lên đến 150 Mbit/S)

- Bluetooth: 4.2 BR/EDR BLE 2 chế độ điều khiển

- Ảng ten: PCB

- Bảo mật: IEEE 802.11, bao gồm cả WFA, WPA/WPA2 và WAPI

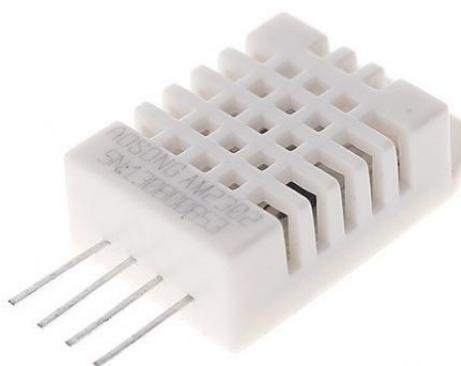
(3) Phần cứng tăng tốc mật mã học

- Các thuật toán hỗ trợ: AES, SHA-2, RSA, ECC (Elliptic Curve Cryptography)

- Bộ sinh số ngẫu nhiên (RNG): Có tích hợp

- Chức năng: Tăng tốc các tác vụ mã hóa, đảm bảo an toàn dữ liệu khi truyền nhận qua mạng WiFi/Bluetooth

#### 1.4.1.4 Cảm biến nhiệt/độ ẩm DHT22



Hình 1.4: Cảm biến nhiệt/độ ẩm DHT22

Cảm biến nhiệt độ - độ ẩm DHT22 bao gồm cảm biến điện trở NTC và cảm biến độ ẩm kiểu điện trở kết hợp cùng một IC VĐK 8-bit được hiệu chỉnh chính xác từ nhà máy, cho tốc độ đáp ứng nhanh, dữ liệu chính xác, ổn định.

Thông số kỹ thuật:

- Hãng sản xuất: ASAIR
- Dải đo: 20~90%RH / 0~50°C
- Sai số cảm biến:  $\pm 5\%$ RH /  $\pm 2^\circ\text{C}$
- Sai số phép đo:  $\pm 1\%$ RH /  $\pm 1^\circ\text{C}$
- Giao tiếp với VĐK theo chuẩn One-Wire, khung truyền 5 bytes, bao gồm 4 bytes dữ liệu nhiệt độ - độ ẩm và 1 byte checksum
- Điện áp hoạt động: 3~5.5Vdc

#### 1.4.1.5 Còi báo cháy Buzzer:



Hình 1.5: Còi báo cháy

Thông số kỹ thuật

- Nguồn: 3.5V - 5.5V
- Dòng điện tiêu thụ: <25mA
- Tần số cộng hưởng:  $2300\text{Hz} \pm 500\text{Hz}$
- Biên độ âm thanh: >80 dB
- Nhiệt độ hoạt động: -20 °C đến +70 °C
- Kích thước: Đường kính 12mm, cao 9,7mm

#### 1.4.1.6 Máy bơm nước mini 3V-5V

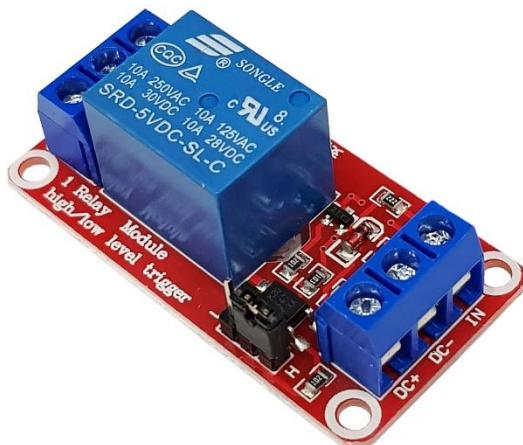


Hình 1.6: Máy bơm 5V

Thông số kỹ thuật:

- Điện áp: DC3-5V
- Dòng điện: 100-200mA
- Đường kính ống: 8mm
- Lưu lượng 1.2-1.6L / phút
- Dây dẫn, ống nước mềm,...

#### 1.4.1.7 Relay 5V



Hình 1.7: Relay 5V

Nguyên lý hoạt động: điều khiển đóng mở máy bơm, dùng điện áp nhỏ để kích mở điện áp lớn

Thông số kỹ thuật

- Nguồn đầu vào: 5V (DC)
- Chân IN: kích mở relay

- Jump H/L level trigger: thiết lập mức điều khiển relay. Có 2 mức: HIGH / LOW.
- COM: Tiếp điểm relay 220V 10A (là tiếp điểm, không phải điện áp ra)
- NO: chân thường mở, NC: chân thường đóng

### 1.4.2 Phần mềm

#### 1.4.2.1 Arduino IDE

Arduino IDE là môi trường phát triển tích hợp (Integrated Development Environment) dùng để viết, biên dịch và nạp chương trình cho vi điều khiển ESP32. Giao diện của Arduino IDE thân thiện, dễ sử dụng, phù hợp cho cả người mới bắt đầu và người đã có kinh nghiệm lập trình nhúng.

Trong đề tài này, Arduino IDE được sử dụng để lập trình cho board ESP32 WROOM-32, nhằm điều khiển cảm biến MP2, thu thập dữ liệu nhiệt độ, khói và thực hiện gửi các giá trị đó lên MQTT Broker. Sau khi viết xong chương trình, mã lệnh sẽ được biên dịch và nạp vào ESP32 thông qua cổng giao tiếp USB (chip CP2102).

Arduino IDE còn tích hợp các công cụ kiểm tra cú pháp và gỡ lỗi cơ bản, giúp phát hiện lỗi trong quá trình viết mã. Ngoài ra, nó cho phép người dùng theo dõi dữ liệu cảm biến trực tiếp thông qua Serial Monitor, hỗ trợ việc kiểm tra và hiệu chỉnh hệ thống một cách thuận tiện.

#### 1.4.2.2 HiveMQ (MQTT Broker)

HiveMQ là một nền tảng MQTT broker mạnh mẽ, được thiết kế để hỗ trợ giao tiếp giữa thiết bị IoT và các hệ thống backend một cách ổn định, an toàn và có khả năng mở rộng cao. Với khả năng xử lý hàng triệu kết nối đồng thời, HiveMQ thường được sử dụng trong các giải pháp IoT quy mô lớn như công nghiệp, vận tải, năng lượng và smart home.

HiveMQ có các đặc điểm sau:

- (1) Khả năng mở rộng linh hoạt: HiveMQ hỗ trợ mở rộng theo chiều ngang thông qua clustering, cho phép hệ thống dễ dàng xử lý thêm nhiều thiết bị mà không ảnh hưởng đến hiệu suất.
- (2) Độ tin cậy và ổn định cao: Nền tảng được thiết kế để hoạt động liên tục trong môi trường IoT yêu cầu độ ổn định lớn, đảm bảo kết nối không bị gián đoạn trong thời gian dài.
- (3) Tuân thủ chuẩn MQTT đầy đủ: HiveMQ hỗ trợ toàn diện các phiên bản MQTT, giúp đảm bảo tính tương thích với nhiều loại thiết bị và ứng dụng IoT.
- (4) Dễ triển khai trên nhiều môi trường: Có thể cài đặt và vận hành trên cloud, on-premise hoặc hybrid, đáp ứng đa dạng nhu cầu của doanh nghiệp.
- (5) Khả năng tích hợp mạnh mẽ: HiveMQ cung cấp các extension và connector giúp kết nối dễ dàng với các hệ thống như Kafka, cơ sở dữ liệu, hoặc các nền tảng doanh nghiệp khác.

(6) Bảo mật cao: Hỗ trợ mã hóa TLS, cơ chế xác thực linh hoạt và quản lý quyền truy cập chi tiết nhằm bảo vệ dữ liệu và thiết bị khỏi các rủi ro an ninh mạng.

#### 1.4.2.3 Frontend (Ứng dụng người dùng – Kotlin)

Phần Frontend của hệ thống được xây dựng bằng ngôn ngữ Kotlin trên nền tảng Android, nhằm tạo ra một ứng dụng di động cho phép người dùng tương tác trực tiếp với hệ thống cảnh báo cháy.

Ứng dụng có các chức năng chính như:

- (1) Hiển thị giá trị nhiệt độ và khói theo thời gian thực từ MQTT Broker.
- (2) Cho phép người dùng thiết lập ngưỡng cảnh báo nhiệt độ hoặc khói.
- (3) Gửi thông báo (notification) đến người dùng khi hệ thống phát hiện dấu hiệu cháy.

Giao diện ứng dụng được thiết kế đơn giản, thân thiện và dễ thao tác, giúp người dùng có thể giám sát tình trạng môi trường trong nhà hoặc khu vực cần bảo vệ mọi lúc, mọi nơi.

#### 1.4.2.4 Backend (API – Java)

Phần Backend của hệ thống được phát triển bằng ngôn ngữ Java, đóng vai trò trung gian giữa Frontend (ứng dụng Kotlin) và dữ liệu nhận được từ MQTT Broker.

Backend chịu trách nhiệm xây dựng API (Application Programming Interface) giúp:

- (1) Giao tiếp, truy xuất và cập nhật dữ liệu từ Broker.
- (2) Xử lý các yêu cầu gửi từ ứng dụng di động (Frontend).
- (3) Quản lý việc xác thực, phân quyền người dùng và xử lý thông báo hệ thống.

Nhờ có Backend, quá trình trao đổi dữ liệu giữa ứng dụng và cơ sở dữ liệu trở nên ổn định, bảo mật và hiệu quả hơn. Bên cạnh đó, việc tách riêng tầng Backend cũng giúp hệ thống dễ bảo trì, mở rộng và tích hợp thêm các chức năng mới trong tương lai (ví dụ như thống kê dữ liệu, lưu lịch sử cảnh báo, ...).

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Khái niệm về IoT (Internet of Things)

Internet of Things (IoT), hay còn gọi là Internet vạn vật, là một mạng lưới các thiết bị vật lý được kết nối với nhau thông qua Internet, cho phép chúng thu thập, trao đổi và xử lý dữ liệu mà không cần sự can thiệp trực tiếp của con người.

Các thiết bị này có thể là cảm biến, bộ điều khiển, máy móc, phương tiện, thiết bị điện tử hoặc các hệ thống nhúng, được trang bị cảm biến, phần mềm và khả năng kết nối mạng để giao tiếp với nhau hoặc với máy chủ trung tâm.

IoT giúp các thiết bị trong hệ thống tự động thu thập thông tin từ môi trường, phân tích dữ liệu, và thực hiện hành động phản hồi phù hợp, góp phần nâng cao mức độ tự động hóa và hiệu quả vận hành.

### 2.2 Tầm quan trọng của IoT trong cảnh báo cháy nổ

Cháy, nổ là một trong những sự cố nguy hiểm có thể gây thiệt hại nghiêm trọng về người và tài sản. Trong nhiều trường hợp, nguyên nhân đến từ việc phát hiện muộn hoặc không có hệ thống giám sát kịp thời. Do đó, việc áp dụng công nghệ Internet of Things (IoT) vào hệ thống cảnh báo cháy nổ mang ý nghĩa quan trọng trong việc nâng cao khả năng phát hiện sớm và xử lý tình huống khẩn cấp.

Công nghệ IoT cho phép các thiết bị cảm biến như cảm biến khói, cảm biến nhiệt độ, khí gas, độ ẩm,... kết nối với nhau và truyền dữ liệu liên tục về trung tâm điều khiển hoặc lên nền tảng đám mây. Nhờ đó, hệ thống có thể giám sát trạng thái môi trường theo thời gian thực, nhanh chóng phát hiện các dấu hiệu bất thường như nhiệt độ tăng cao hoặc nồng độ khí cháy vượt ngưỡng.

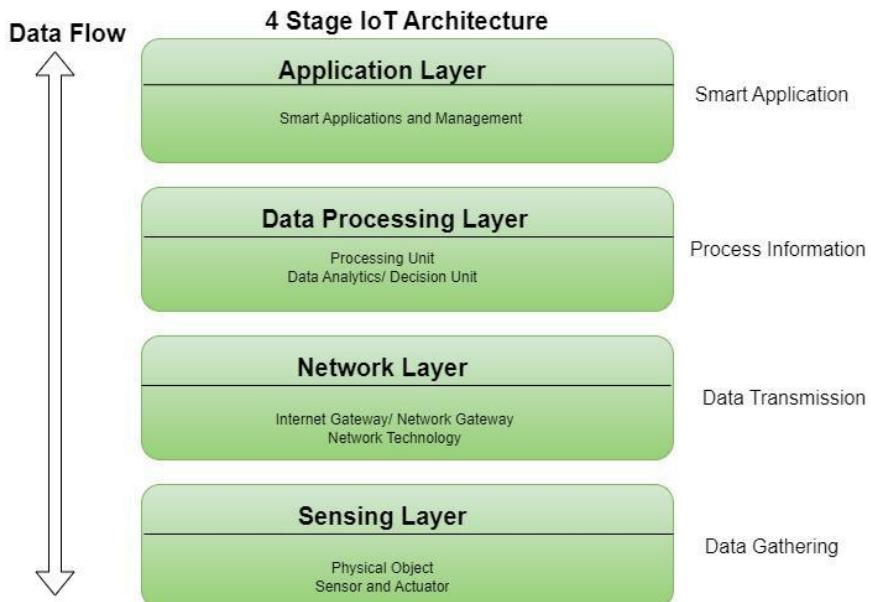
Khi có nguy cơ cháy nổ, hệ thống IoT có thể tự động gửi cảnh báo tức thời đến người quản lý hoặc người dùng thông qua ứng dụng di động, tin nhắn hoặc email, giúp họ kịp thời phản ứng, cắt nguồn điện, kích hoạt hệ thống chữa cháy, hoặc sơ tán người ra khỏi khu vực nguy hiểm. Đây là ưu điểm vượt trội so với các hệ thống cảnh báo truyền thống chỉ hoạt động cục bộ mà không có khả năng truyền thông tin từ xa.

Bên cạnh đó, việc lưu trữ dữ liệu trên nền tảng đám mây còn cho phép phân tích và đánh giá xu hướng nguy cơ cháy nổ theo thời gian, hỗ trợ công tác quản lý, bảo trì thiết bị và nâng cao hiệu quả an toàn phòng cháy chữa cháy trong tương lai.

IoT không chỉ mang lại khả năng tự động hóa và giám sát thông minh, mà còn mở ra hướng phát triển cho các thành phố thông minh (Smart City), nơi mọi hệ thống an ninh, cảnh báo, năng lượng và môi trường được liên kết chặt chẽ, đảm bảo cuộc sống an toàn, tiện nghi và bền vững hơn.

## 2.3 Cấu trúc của hệ thống báo cháy

### 2.3.1 Cấu trúc chung của một hệ thống IoT



Hình 2.1: Sơ đồ cấu trúc của một hệ thống IoT

#### 2.3.1.1 Lớp cảm biến (Sensing Layer)

Lớp cảm biến là lớp chịu trách nhiệm thu thập dữ liệu từ những nguồn khác nhau.

Lớp này bao gồm các cảm biến và bộ truyền động được đặt trong môi trường để thu thập thông tin về nhiệt độ, độ ẩm, ánh sáng, âm thanh và các thông số vật lý khác. Các thiết bị này được kết nối với lớp mạng thông qua các giao thức truyền thông có dây hoặc không dây.

#### 2.3.1.2 Lớp mạng (Network Layer)

Lớp mạng của kiến trúc IoT chịu trách nhiệm cung cấp thông tin liên lạc và kết nối giữa các thiết bị trong hệ thống IoT.

Lớp mạng bao gồm các giao thức và công nghệ cho phép các thiết bị kết nối và liên lạc với nhau cũng như với mạng Internet rộng hơn. Đối với hệ thống này, công nghệ mạng được sử dụng là Wifi.

Ngoài ra, lớp mạng có thể bao gồm các cổng và bộ định tuyến đóng vai trò trung gian giữa các thiết bị và mạng Internet rộng hơn, đồng thời cũng có thể bao gồm các tính năng bảo mật như mã hóa và xác thực để bảo vệ khỏi truy cập trái phép.

#### 2.3.1.3 Lớp xử lý dữ liệu (Data Processing Layer)

Lớp xử lý dữ liệu của kiến trúc IoT để cập đến các thành phần phần mềm và phần cứng chịu trách nhiệm thu thập, phân tích và giải thích dữ liệu từ các thiết bị IoT. Lớp này chịu trách nhiệm nhận dữ liệu thông tin từ thiết bị, xử lý và cung cấp dữ liệu đó để phân tích hoặc hành động thêm.

Lớp xử lý dữ liệu bao gồm nhiều công nghệ và công cụ khác nhau, chẳng hạn như hệ thống quản lý dữ liệu, nền tảng phân tích và thuật toán học máy. Những công cụ này được sử dụng để trích xuất những hiểu biết sâu sắc có ý nghĩa từ dữ liệu và đưa ra quyết định dựa trên dữ liệu đó. Ví dụ về công nghệ được sử dụng trong lớp xử lý dữ liệu là hồ dữ liệu, là kho lưu trữ tập trung để lưu trữ dữ liệu thô từ các thiết bị IoT.

#### 2.3.1.4 Lớp ứng dụng (Application Layer)

Lớp ứng dụng của kiến trúc IoT là lớp trên cùng tương tác trực tiếp với người dùng cuối. Nó chịu trách nhiệm cung cấp các giao diện và chức năng thân thiện với người dùng cho phép người dùng truy cập và kiểm soát các thiết bị IoT.

Lớp này bao gồm các phần mềm và ứng dụng khác nhau như ứng dụng di động, cổng web và các giao diện người dùng khác được thiết kế để tương tác với cơ sở hạ tầng IoT cơ bản. Nó cũng bao gồm các dịch vụ phần mềm trung gian cho phép các thiết bị và hệ thống IoT khác nhau giao tiếp và chia sẻ dữ liệu liền mạch.

Lớp ứng dụng cũng bao gồm khả năng phân tích và xử lý cho phép dữ liệu được phân tích và chuyển đổi thành thông tin chi tiết có ý nghĩa. Điều này có thể bao gồm các thuật toán học máy, công cụ trực quan hóa dữ liệu và các khả năng phân tích nâng cao khác.

### 2.3.2 Cấu trúc hệ thống của dự án

Trong đề tài này, hệ thống được lập trình và triển khai trên nền tảng thiết bị di động Android, nhằm cung cấp thông tin giám sát môi trường tại khu vực cần bảo vệ và gửi thông báo cảnh báo đến người dùng khi phát hiện cháy xảy ra.

Hệ thống được thiết kế theo mô hình nhiều lớp, bao gồm các thành phần chính như sau:

- (1) Lớp cảm biến: ESP32, cảm biến khói, cảm biến lửa, cảm biến nhiệt độ, đèn còi.
- (2) Lớp mạng: Mạng Wi-Fi và MQTT Broker.
- (3) Lớp xử lý dữ liệu: ESP32 xử lý và gửi dữ liệu.
- (4) Lớp ứng dụng: Phát triển ứng dụng Android để hiển thị dữ liệu và server để nhận và xử lý dữ liệu từ thiết bị, lưu trữ dữ liệu vào cơ sở dữ liệu
- (5) Giao thức sử dụng:
  - Wi-Fi (802.11 b/g/n): Được sử dụng cho kết nối giữa ESP8266 và mạng Wi-Fi.
  - HTTP/HTTPS: Giao thức sử dụng để gửi và nhận dữ liệu giữa server và ứng dụng người dùng
  - MQTT: gửi nhận dữ liệu từ thiết bị và server.
  - WebSocket: mở một kết nối duy nhất để gửi nhận dữ liệu realtime

## CHƯƠNG 3: PHÂN TÍCH YÊU CẦU

### 3.1 Mục tiêu của hệ thống

Hệ thống báo cháy cần đạt được hai mục tiêu chính như sau:

- (1) Giám sát và cảnh báo cháy: Hệ thống báo cháy tự động giúp phát hiện các yếu tố nguy hiểm như nhiệt độ cao, khí gas độc hại, hoặc có ngọn lửa. Khi phát hiện nguy cơ cháy, hệ thống sẽ kích hoạt còi báo động để cảnh báo người dùng.
- (2) Lưu trữ dữ liệu của thiết bị và cảm biến: Dữ liệu từ các cảm biến sẽ được gửi lên MQTT Broker, và server có nhiệm vụ lấy và lưu trữ dữ liệu vào cơ sở dữ liệu (PostgreSQL).
- (3) Điều khiển thiết bị: người dùng có thể điều khiển thiết bị từ xa như bật tắt còi báo, bật tắt máy bơm thông qua ứng dụng người dùng.

### 3.2 Phạm vi của hệ thống

- (1) Số lượng thiết bị (mỗi loại 1 thiết bị, sử dụng cho quá trình thử nghiệm):

- Vị điều khiển ESP32
- Cảm biến khói/khí gas (MP-2)
- Cảm biến lửa (MH-Sensor)
- Cảm biến nhiệt độ và độ ẩm (DHT22)
- Còi báo cháy
- Máy bơm nước mini (mô phỏng hệ thống chữa cháy tự động)

- (2) Môi trường hoạt động:

- Hệ thống được thiết kế để hoạt động trong nhà, trong điều kiện nhiệt độ môi trường ổn định.
- Các cảm biến có khả năng chịu được mức nhiệt cao trong giới hạn cho phép, đảm bảo hoạt động ổn định khi phát hiện cháy.

### 3.3 Tiêu chí thành công

- (1) Độ chính xác:

- Tỷ lệ phát hiện cháy đúng (khói hoặc nhiệt độ bất thường)  $\geq 96\%$ .
- Tỷ lệ cảnh báo giả (không có cháy thật)  $< 10\%$ .
- Sai số đo cảm biến nhiệt độ  $< \pm 2^\circ\text{C}$  so với nhiệt kế chuẩn.

- (2) Đỗ trễ:

- Dữ liệu cảm biến (khói/nhiệt độ) cập nhật lên server trong vòng  $< 5$  giây.
- Thời gian từ khi phát hiện cháy đến khi phát cảnh báo (đèn/còi/app)  $< 3$  giây

(3) Độ tin cậy:

- Tỷ lệ truyền dữ liệu thành công  $> 95\%$ .
- Hệ thống hoạt động ổn định liên tục  $\geq 95\%$
- Sau khi mất kết nối mạng, cảm biến tự động khôi phục và gửi dữ liệu lại trong  $< 30$  giây.

(4) Chi phí:

- Tổng chi phí đầu tư ban đầu cho mô hình demo  $< 500000$  VNĐ

### **3.4 Yêu cầu từ các bên liên quan**

Trong quá trình xây dựng hệ thống IoT cảnh báo cháy, các yêu cầu từ các bên liên quan được xác định nhằm đảm bảo hệ thống đáp ứng đúng nhu cầu sử dụng thực tế và mang lại trải nghiệm thuận tiện cho người dùng. Các yêu cầu cụ thể như sau:

(1) Hiển thị:

- Người dùng mong muốn có thể quan sát trực tiếp tình trạng hoạt động của hệ thống thông qua ứng dụng di động.
- Các thông số quan trọng như nhiệt độ, nồng độ khói và trạng thái cảm biến lửa cần được hiển thị rõ ràng, cập nhật theo thời gian thực để người dùng dễ dàng theo dõi và đánh giá mức độ an toàn trong khu vực được giám sát.

(2) Cảnh báo:

- Hệ thống phải có khả năng gửi cảnh báo tức thời đến người dùng khi phát hiện các dấu hiệu bất thường như nhiệt độ tăng cao, có khói hoặc xuất hiện ngọn lửa.
- Thông báo cần được hiển thị dưới dạng thông báo đẩy (push notification) trên ứng dụng để người dùng kịp thời nắm bắt tình hình và đưa ra biện pháp xử lý nhanh chóng.

(3) Điều khiển:

- Người dùng cần có khả năng tương tác và điều khiển một số chức năng cơ bản của hệ thống thông qua ứng dụng, điển hình là tắt còi báo cháy sau khi đã nhận được cảnh báo. Việc này giúp người dùng chủ động trong việc kiểm soát hệ thống và tránh gây tiếng ồn kéo dài khi tình huống đã được xử lý.

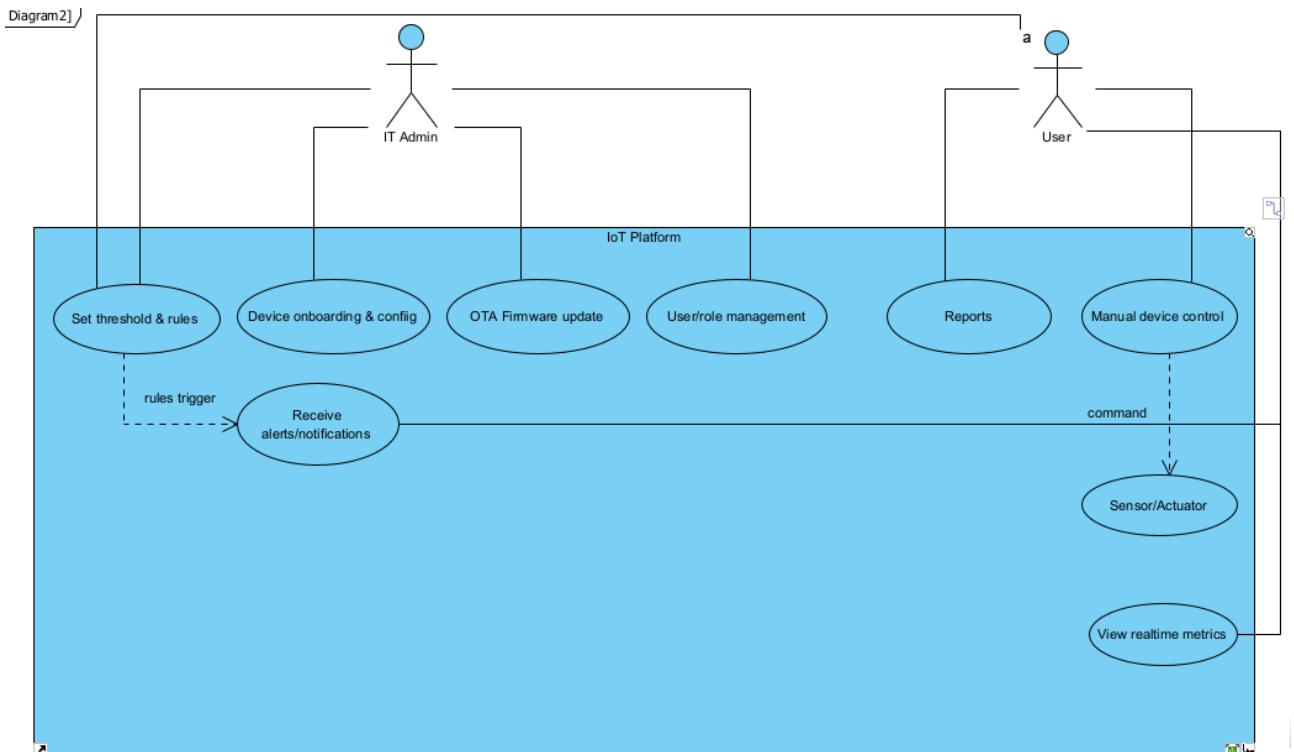
### 3.5 Yêu cầu chức năng

#### 3.5.1 Các chức năng của hệ thống

STT	Chức năng	Tác nhân
1	Hiển thị dữ liệu thời gian thực	Người dùng (User)
2	Nhận thông báo	Người dùng (User)
3	Điều khiển thiết bị thủ công	Người dùng (User)
4	Thiết lập ngưỡng và quy tắc	Quản trị viên (Iot Admin)
5	Thiết lập báo cáo định kỳ	Người dùng (User)
6	Device onboarding & config	Quản trị viên (Iot Admin)
7	OTA firmware update	Quản trị viên (Iot Admin)
8	User/role management	Quản trị viên (Iot Admin)

Bảng 3.1: Bảng liệt kê các chức năng của hệ thống theo vai trò người dùng

#### 3.5.2 Sơ đồ Usecase toàn hệ thống



Hình 3.1: Sơ đồ Usecase toàn hệ thống

### 3.6 Yêu cầu phi chức năng

- (1) Hiệu năng:

- Hệ thống cần phải phản hồi nhanh chóng khi phát hiện khói và nhiệt vượt ngưỡng.

- Thời gian phản hồi giữa cảm biến và còi báo cháy cần nhỏ hơn 3 giây

(2) Bảo mật:

- Mã hóa: Dữ liệu được truyền bảo mật dựa trên TLS 1.2/TLS 1.3

- Phân quyền người dùng: đảm bảo chỉ có người dùng được phép mới có thể truy cập và quản lý dữ liệu.

- Ghi log truy cập, truy xuất dữ liệu

(3) Độ tin cậy:

- Hệ thống phải hoạt động ổn định, đảm bảo liên tục giám sát nhiệt độ, khói trong môi trường mà không bị gián đoạn.

- Cơ sở dữ liệu cần đồng bộ hóa dữ liệu một cách mượt mà giữa thiết bị ESP32 và giao diện người dùng.

- Khi mất kết nối WiFi, hệ thống tự động kết nối lại và tiếp tục gửi dữ liệu.

- Backup dữ liệu theo ngày để tránh mất dữ liệu

(4) Khả năng mở rộng

- Cho phép thêm nhiều cảm biến vào hệ thống nếu cần.

- Có khả năng mở rộng phần mềm nhu tách microservices.

- Có khả năng tích hợp thêm AI cho tương lai

(5) Chi phí, năng lượng:

- Chi phí phần cứng dưới 500.000 VNĐ vẫn đảm bảo chức năng phát hiện và cảnh báo cháy hoạt động ổn định

### 3.7 Ràng buộc

#### 3.7.1 Môi trường hoạt động

Nhiệt độ và độ ẩm: cảm biến đặt trong nhà, chịu được biến độ từ -10°C đến 50°C, độ ẩm từ 0% đến 95% (không ngưng tụ), cần đảm bảo vi điều khiển và pin không bị quá nhiệt.

Nhiều điện tử (EMI): Các thiết bị công suất lớn (động cơ, biến tần) có thể gây nhiễu, ảnh hưởng đến hoạt động của vi điều khiển.

Sóng vô tuyến (Wi-Fi): Cần khảo sát vị trí lắp đặt. Tường bê tông cốt thép, nhiều vách ngăn kim loại sẽ làm suy giảm tín hiệu.

Đây là hệ thống an toàn, không thể phụ thuộc 100% vào điện lưới. Bắt buộc phải có nguồn dự phòng (pin).

### **3.7.2 Ràng buộc pháp lý**

#### **(1) Đáp ứng các yêu cầu kỹ thuật theo tiêu chuẩn**

Hệ thống cần đáp ứng yêu cầu kỹ thuật trong các tiêu chuẩn về hệ thống báo cháy tự động (ví dụ: độ nhạy của cảm biến, âm lượng còi báo động >85dB ở khoảng cách 3m, thời gian phát hiện, v.v.).

Đầu báo khói: Phải phát hiện được sự cháy trong môi trường thử nghiệm tiêu chuẩn không quá 30 giây.

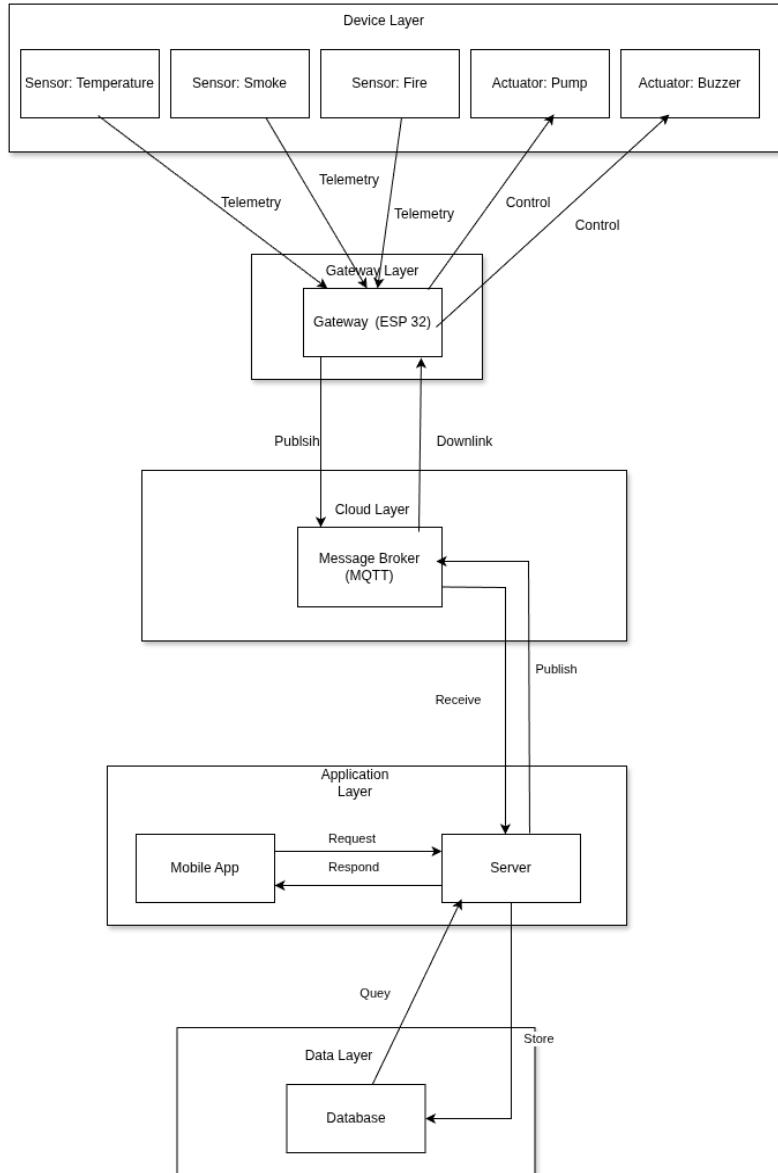
Đầu báo nhiệt: Phải kích hoạt khi nhiệt độ môi trường đạt đến ngưỡng quy định hoặc tốc độ gia tăng nhiệt vượt mức cho phép. Ví dụ: Kích hoạt ở 57°C, 68°C, hoặc khi nhiệt độ tăng 8.4°C/phút.

#### **(2) Bảo mật dữ liệu và Quyền riêng tư:**

Hệ thống IoT thu thập dữ liệu về trạng thái của ngôi nhà/công trình. Dữ liệu này cần được bảo vệ theo Nghị định 13/2023/NĐ-CP về bảo vệ dữ liệu cá nhân.

Phải có cơ chế mã hóa dữ liệu truyền đi, bảo mật tài khoản người dùng và nêu rõ chính sách sử dụng dữ liệu.

### 3.8 Mô hình hóa yêu cầu



Hình 3.2: Mô hình 4 lớp IoT

### 3.9 Xác minh và phê duyệt

#### 3.9.1 Xem xét lại yêu cầu với tất cả bên liên quan:

- (1) Người dùng cuối: Chủ nhà, ban quản lý tòa nhà, nhân viên an toàn lao động. Họ quan tâm đến: Dễ sử dụng, cảnh báo nhanh, chính xác, không phiền nhiễu, pin lâu.
- (2) Nhà đầu tư/Ban lãnh đạo: Quan tâm đến chi phí, tiềm năng thị trường, tiến độ dự án, lợi nhuận.
- (3) Đội ngũ kỹ thuật (phát triển): Quan tâm đến yêu cầu kỹ thuật có rõ ràng, khả thi không.
- (4) Cơ quan quản lý (PCCC): Quan tâm đến việc tuân thủ quy chuẩn, tiêu chuẩn.
- (5) Kỹ thuật viên lắp đặt/bảo trì: Quan tâm đến việc lắp đặt có dễ không, bảo trì có thuận tiện không.

### **3.9.2 Ưu tiên hóa yêu cầu:**

(1) Must-have:

- Phát hiện khói/nhiệt độ vượt ngưỡng một cách chính xác.
- Kích hoạt còi báo động tại chỗ với âm lượng lớn.
- Hoạt động được bằng pin khi mất điện.
- Gửi cảnh báo sự cố đến điện thoại người dùng.

(2) Should-have:

- Khả năng tắt chuông báo (silence) từ ứng dụng (ví dụ khi đang nấu ăn và có khói).
- Chức năng tự kiểm tra định kỳ và báo cáo trạng thái.

(3) Nice-to-have:

- Phân tích và hiển thị biểu đồ chất lượng không khí, nhiệt độ theo thời gian.
- Tích hợp với trợ lý ảo

Sau cùng là chốt bộ yêu cầu chính thức trước khi chuyển sang thiết kế logic

## CHƯƠNG 4: TRIỂN KHAI KẾ HOẠCH

### 4.1 Các công việc chính cần thực hiện

- (1) Tìm hiểu về lập trình Arduino
- (2) Tìm hiểu về HiveMQ
- (3) Mua và lắp đặt thiết bị
- (4) Lập trình Arduino với ESP32
- (5) Xây dựng server backend và cơ sở dữ liệu
- (6) Xây dựng ứng dụng người dùng
- (7) Viết báo cáo

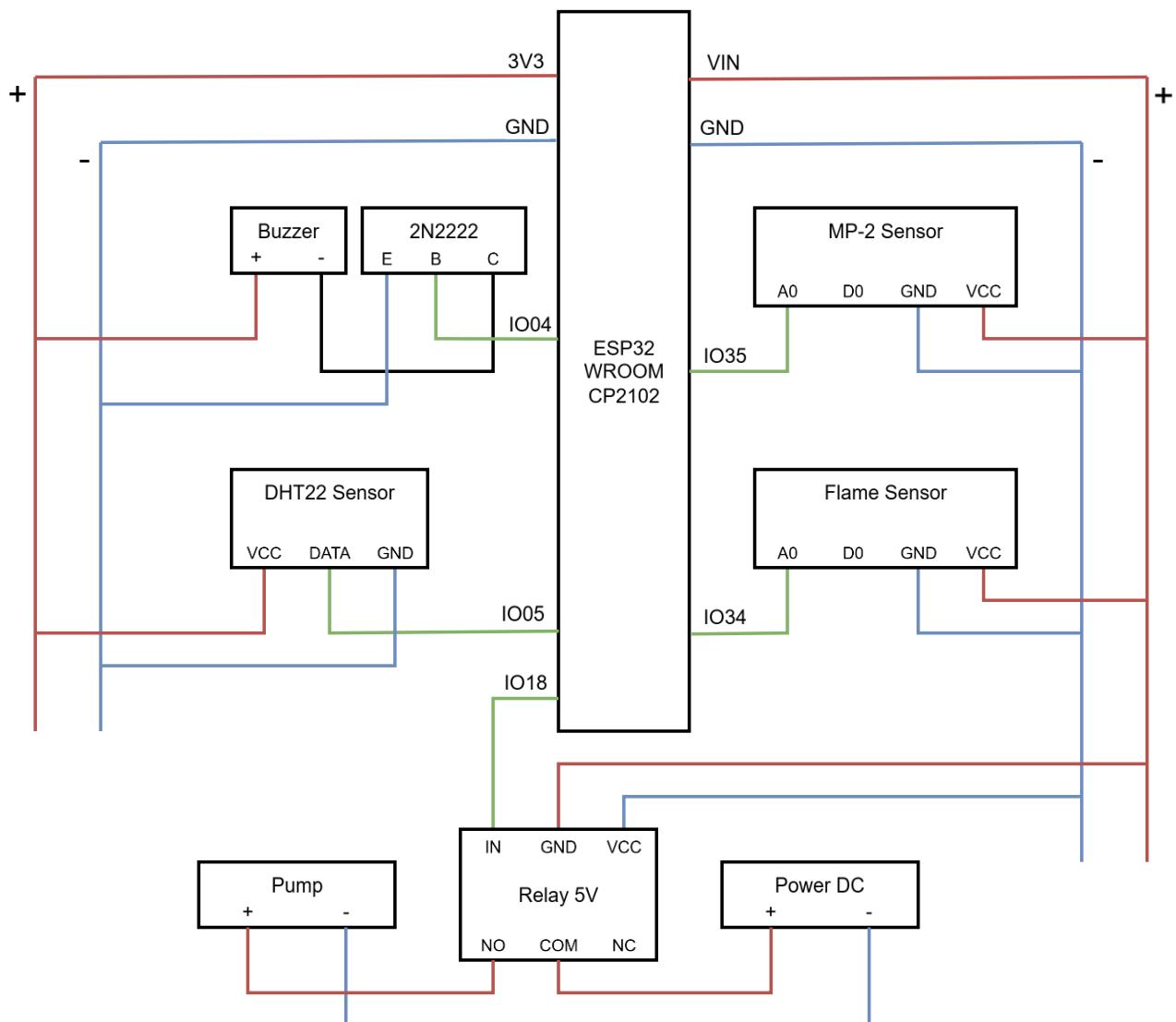
### 4.2 Bảng phân chia công việc

Sinh viên	Mã sinh viên	Công việc
Đào Đức Duy	B22DCCN145	<ul style="list-style-type: none"><li>- Tìm hiểu về MQTT Broker</li><li>- Thực hiện kết nối giữa thiết bị và HiveMQ.</li><li>- Thực hiện cài đặt chức năng cập nhật firmware từ xa.</li></ul>
Ong Thé Minh	B22DCCN541	<ul style="list-style-type: none"><li>- Viết báo cáo.</li><li>- Lắp đặt thiết bị</li><li>- Lập trình Arduino, kết nối ESP32 với HiveMQ</li></ul>
Đỗ Gia Phong	B22DCCN613	<ul style="list-style-type: none"><li>- Mua thiết bị</li><li>- Xây dựng backend và cơ sở dữ liệu</li><li>- Thực hiện kết nối giữa server và HiveMQ</li></ul>
Phan Thị Hồng Thắm	B22DCCN806	<ul style="list-style-type: none"><li>- Tìm hiểu về WebSocket</li><li>- Lập trình ứng dụng người dùng (Android).</li><li>- Xử lý dữ liệu nhận được từ server và gửi đi từ phía Client.</li></ul>

Bảng 4.1: Bảng phân công công việc cho thành viên nhóm

## 4.3 Thiết kế hệ thống

### 4.3.1 Sơ đồ mạch thiết bị



Hình 4.1: Sơ đồ mạch nối thiết bị

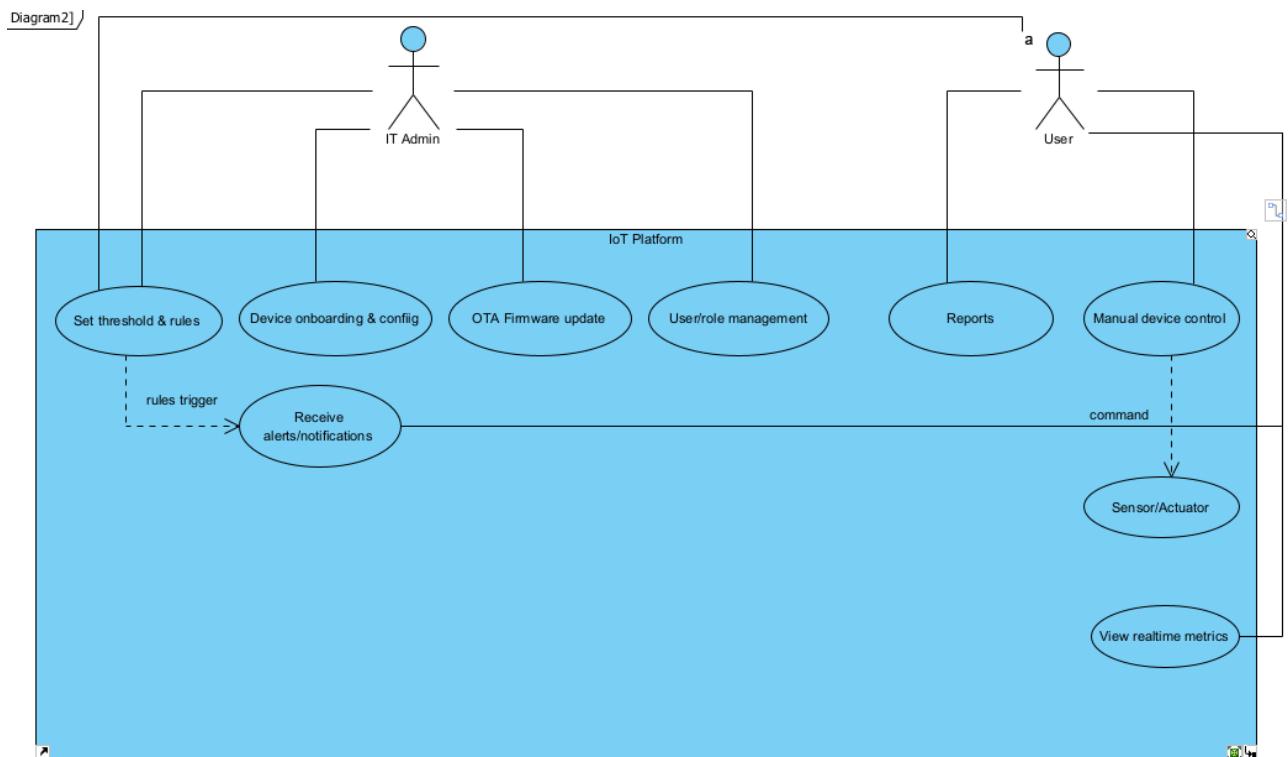
Giải thích chi tiết cách nối mạch giữa các cảm biến và tải:

Thiết bị	Chân nguồn	Chân đích
Còi (Buzzer)	+	3V3
	-	Chân C của Transistor 2N2222
Transistor 2N2222	Chân E	GND của ESP32
	Chân B	IO04
DHT22	VCC	3V3

	GND	GND của ESP32
	DATA	IO05
Cảm biến lửa	VCC	VIN của ESP32
	GND	GND của ESP32
	A0	IO34
Cảm biến khói MP2	VCC	VIN của ESP32
	GND	GND của ESP32
	A0	IO35
Relay 5V	VCC	VIN của ESP32
	GND	GND của ESP32
	IN	IO18
	COM	Chân + của nguồn pin
	NO	Chân + của máy bơm
Máy bơm	-	Chân – của máy bơm

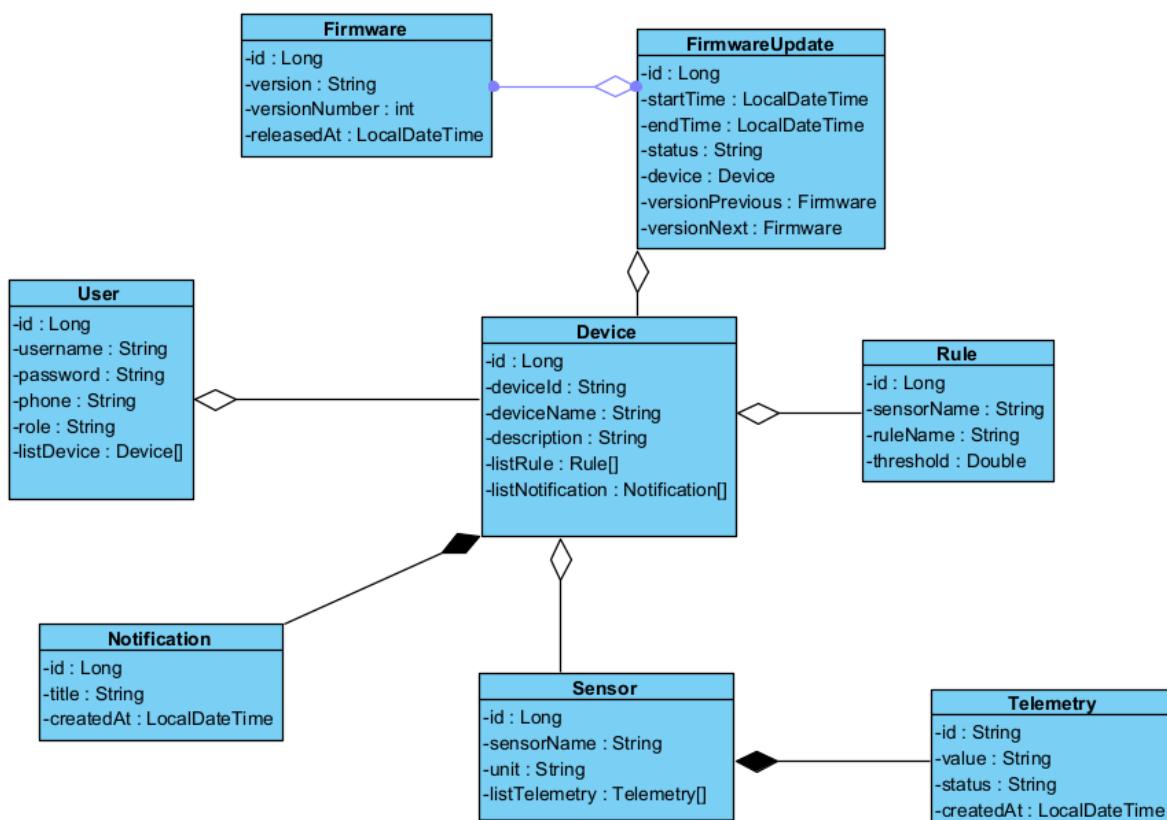
Bảng 4.2: Bảng chân nối chi tiết giữa các cảm biến, tải và ESP32

### 4.3.2 Sơ đồ Use Case hệ thống



Hình 4.2: Sơ đồ Use case toàn hệ thống

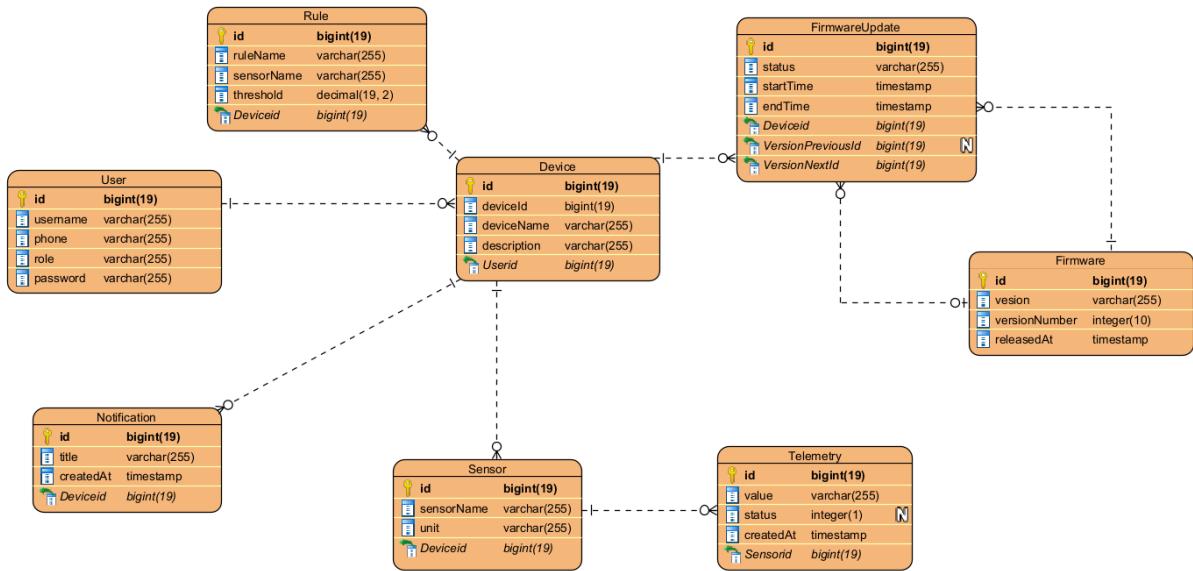
### 4.3.3 Sơ đồ lớp thực thể toàn hệ thống



Hình 4.3: Sơ đồ lớp thực thể toàn hệ thống

- User: thông tin người dùng (id, username, password, phone, role) và danh sách thiết bị mà họ sở hữu (listDevice – quan hệ với Device).
- Device: trung tâm hệ thống, mô tả thiết bị (id, deviceId, deviceName, description) và các danh sách liên quan:
  - ListRule – các Rule gắn với thiết bị.
  - ListNotification – các Notification gửi từ thiết bị.
  - Liên kết với Sensor (thiết bị có nhiều cảm biến).
  - Liên kết với FirmwareUpdate (các lần cập nhật firmware cho thiết bị).
- Sensor: cảm biến thuộc về một thiết bị (id, sensorName, unit) và có danh sách Telemetry (listTelemetry) ghi lại dữ liệu đo.
- Telemetry: bản ghi dữ liệu từ cảm biến (id, value, status, createdAt).
- Rule: luật/điều kiện áp dụng cho thiết bị hoặc cảm biến (id, sensorName, ruleName, threshold) dùng để kiểm tra ngưỡng và sinh cảnh báo.
- Notification: thông báo được tạo ra (thường do vi phạm Rule) (id, title, createdAt), liên kết với Device.
- Firmware: thông tin phiên bản firmware (id, version, versionNumber, releasedAt).
- FirmwareUpdate: lần cập nhật firmware cho một thiết bị (id, startTime, endTime, status) và tham chiếu tới:
  - Thiết bị được cập nhật (device – liên kết với Device),
  - Phiên bản firmware trước (versionPrevious) và sau (versionNext) – liên kết với Firmware.

#### 4.3.4 Sơ đồ cơ sở dữ liệu toàn hệ thống



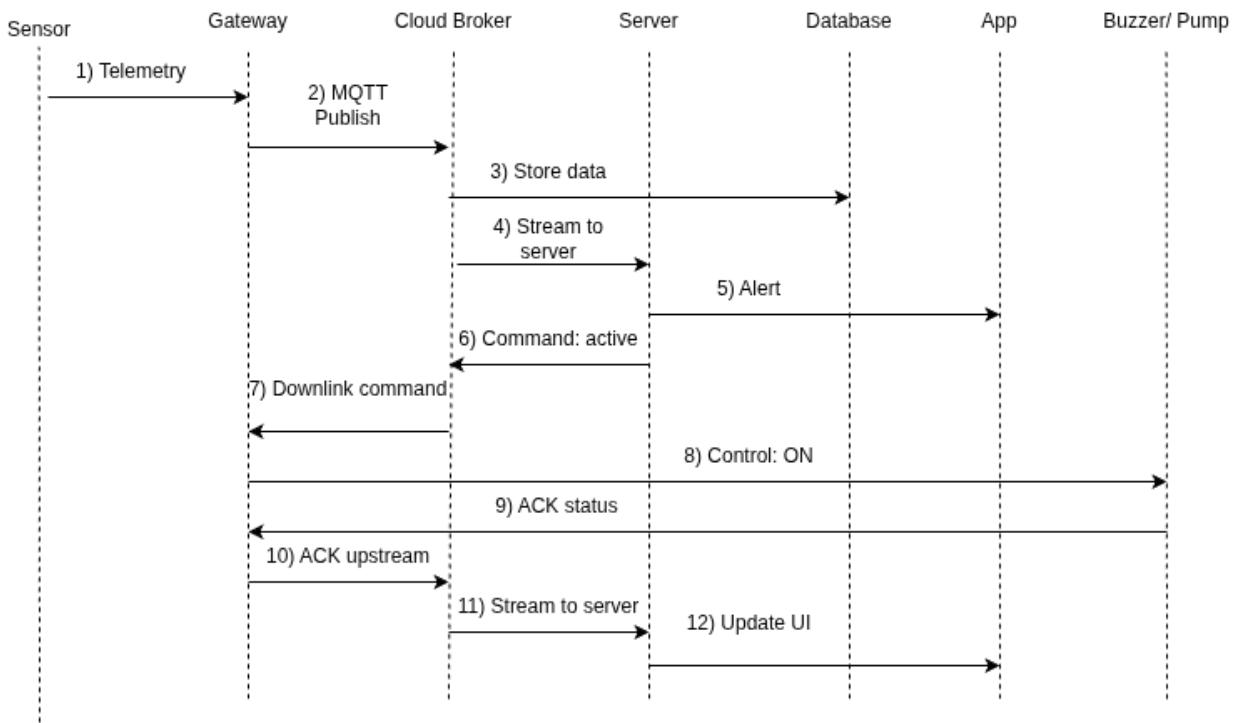
Hình 4.4: Thiết kế cơ sở dữ liệu của hệ thống

Thực thể	Thuộc tính	Quan hệ
User	id, username, phone, role, password	<ul style="list-style-type: none"> <li>1 User – N Device</li> </ul>
Device	id, deviceId, deviceName, description, UserId	<ul style="list-style-type: none"> <li>N Device – 1 User</li> <li>1 Device – N Sensor</li> <li>1 Device – N Rule</li> <li>1 Device – N Notification</li> <li>1 Device – N FirmwareUpdate</li> </ul>
Sensor	id, sensorName, unit, DeviceId	<ul style="list-style-type: none"> <li>N Sensor – 1 Device</li> <li>1 Sensor – N Telemetry</li> </ul>
Telemetry	id, value, status, createdAt, SensorId	<ul style="list-style-type: none"> <li>N Telemetry – 1 Sensor</li> </ul>
Rule	id, ruleName, sensorName, threshold, DeviceId	<ul style="list-style-type: none"> <li>N Rule – 1 Device</li> </ul>
Notification	id, title, createdAt, DeviceId	<ul style="list-style-type: none"> <li>N Notification – 1 Device</li> </ul>

Firmware	id, version, versionNumber, releasedAt	<ul style="list-style-type: none"> <li>1 Firmware – N FirmwareUpdate (versionPrevious)</li> <li>1 Firmware – N FirmwareUpdate (versionNext)</li> </ul>
----------	----------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng 4.3: Bảng mô tả chi tiết quan hệ giữa các bảng trong cơ sở dữ liệu

#### 4.3.5 Sơ đồ luồng dữ liệu (DFD)



Hình 4.5: Sơ đồ luồng dữ liệu của hệ thống

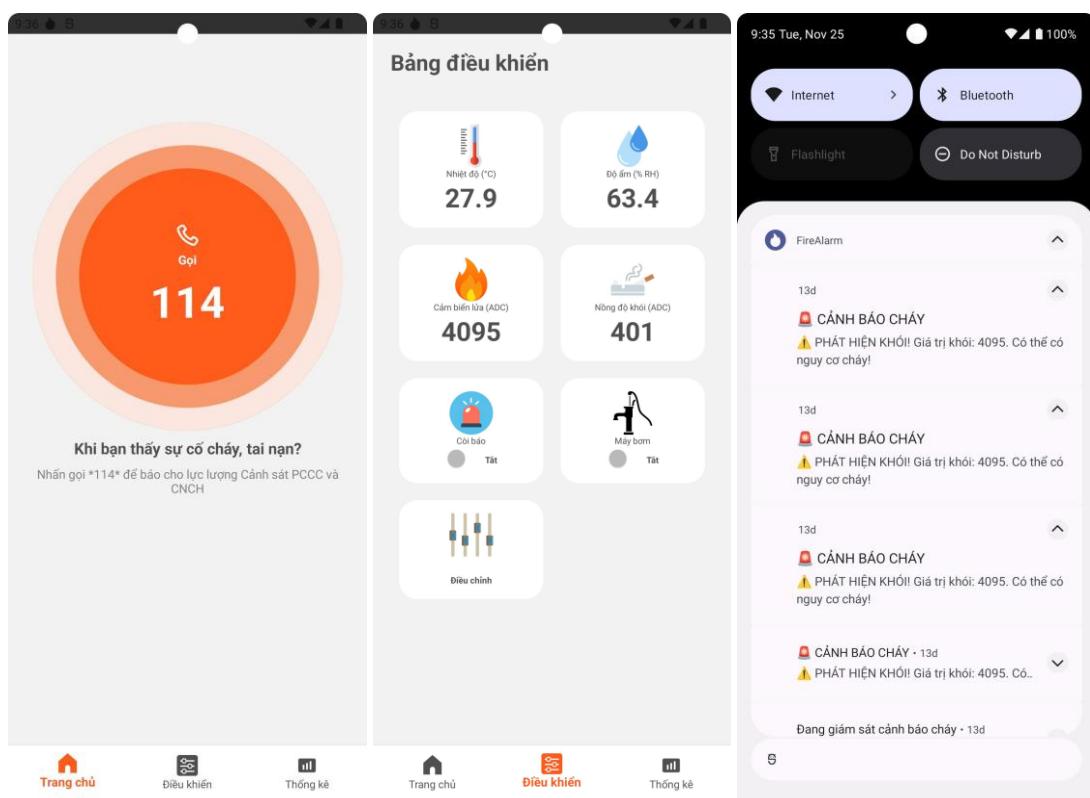
- (1) Sensor thu thập dữ liệu Telemetry (ví dụ: nhiệt độ, độ ẩm, mức nước) và gửi dữ liệu thô đến Gateway.
- (2) Gateway nhận dữ liệu và publish lên Cloud Broker thông qua giao thức MQTT, đảm bảo dữ liệu được đưa lên nền tảng đám mây theo dạng message.
- (3) Cloud Broker tiếp nhận và lưu dữ liệu vào Database để phục vụ việc truy xuất và phân tích sau này.
- (4) Đồng thời, Cloud Broker stream dữ liệu thời gian thực sang Server để xử lý logic hệ thống.
- (5) Server đánh giá dữ liệu theo các rule đã thiết lập (ví dụ vượt ngưỡng) rồi phát sinh cảnh báo (Alert) và gửi thông báo đến App cho người dùng.
- (6) Khi cần điều khiển thiết bị, Server gửi lệnh (Command: active) xuống Cloud Broker để kích hoạt thiết bị ở phía người dùng.

- (7) Cloud Broker gửi lệnh downlink từ đám mây về Gateway.
- (8) Gateway nhận lệnh và điều khiển thiết bị thực thi (ví dụ bật bom, bật còi báo động...).
- (9) Thiết bị phản hồi lại trạng thái thực thi (ACK status) về Cloud Broker.
- (10) Cloud Broker gửi ACK upstream ngược về Gateway để xác nhận lệnh đã được thực hiện.
- (11) Cloud Broker cũng stream trạng thái này lên Server để đồng bộ tình trạng thiết bị.
- (12) Server gửi dữ liệu cập nhật về App, giúp App cập nhật giao diện (UI) theo trạng thái mới nhất (thiết bị đã bật/tắt, cảnh báo đã xử lý, v.v.).

#### 4.4 Kết quả đạt được

Hệ thống đạt được những kết quả như sau:

- Thiết bị phát còi báo khi có khói và có cháy, đồng thời tự động bật máy bơm khi có cháy.
- Người dùng nhận được dữ liệu các cảm biến, gửi được cảnh báo cháy, cảnh báo khói đến ứng dụng
- Người dùng điều khiển được còi và máy bơm trực tiếp từ ứng dụng
- Người dùng có thể điều chỉnh được ngưỡng cảnh báo cho cảm biến lửa và cảm biến khói.
- Ứng dụng có cả năng phân quyền người dùng.



Hình 4.6: Giao diện hiển thị của ứng dụng cảnh báo cháy

#### **4.5 Kết luận và phương hướng mở rộng**

Trong tương lai, hệ thống có thể được mở rộng để hỗ trợ nhiều loại cảm biến hơn, bổ sung khả năng tự động xử lý tình huống và ứng dụng trí tuệ nhân tạo nhằm dự đoán cũng như ngăn chặn nguy cơ cháy ngay từ giai đoạn sớm. Việc nâng cấp này giúp tăng mức độ chủ động và cải thiện toàn diện khả năng bảo vệ an toàn cho không gian sử dụng.

Công nghệ học máy có thể được áp dụng để phân tích dữ liệu cảm biến theo thời gian thực, giúp phân biệt chính xác giữa khói thật, hơi nước, bụi hoặc các dạng nhiễu môi trường. Nhờ đó, hệ thống giám sát sẽ nhận biết chính xác và nhanh chóng khi có cảnh báo. Tính năng này hỗ trợ người quản lý kiểm soát sự cố từ xa và đưa ra quyết định kịp thời.

Việc kết hợp camera sẽ cho phép theo dõi hiện trạng trực tiếp trên điện thoại, cung cấp góc nhìn trực quan giúp xác thực nhanh tình huống khi có cảnh báo. Tính năng này hỗ trợ người quản lý kiểm soát sự cố từ xa và đưa ra quyết định kịp thời.

Hệ thống cũng có thể được thiết kế để phối hợp linh hoạt với các thiết bị và hệ thống an toàn khác như damper gió, nguồn điện từng tầng hoặc các biển báo thoát hiểm thông minh. Sự liên kết này giúp hình thành một “hệ sinh thái an toàn tòa nhà”, nơi mọi thành phần hoạt động đồng bộ nhằm giảm thiểu rủi ro và tăng hiệu quả ứng phó trong trường hợp khẩn cấp.