

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1



BÁO CÁO CUỐI KÌ
HỌC PHẦN IOT VÀ ỨNG DỤNG

ĐỀ TÀI:
HỆ THỐNG BÁO CHÁY THÔNG MINH

NHÓM 17 – LỚP 06

Giảng viên : Kim Ngọc Bách

Nhóm sinh viên thực hiện:

B22DCCN529

Hà Quang Minh

B22DCCN302

Dương Văn Hiếu

B22DCCN829

Bùi Tiến Thịnh

HÀ NỘI 2025

Mục Lục

| | |
|--|----|
| 1. Giới thiệu đề tài..... | 3 |
| 2. Mô tả tổng quan | 4 |
| 2.1. Mô tả hệ thống..... | 4 |
| 2.2. Môi trường hoạt động..... | 4 |
| 2.3. Giả định hệ thống..... | 4 |
| 3. Công nghệ và phần cứng sử dụng..... | 5 |
| 3.1. Phần cứng | 5 |
| 3.2. Phần mềm | 10 |
| 4. Yêu cầu chức năng..... | 11 |
| 4.1. Giám sát khói , lửa và nhiệt độ liên tục | 11 |
| 4.2. Cảnh báo tại chỗ bằng còi , đèn LED và hệ thống quạt,vòi phun tự động hoạt động | 11 |
| 4.3. Gửi thông báo từ xa (ứng dụng di động) | 12 |
| 4.4. Hiển thị trạng thái hệ thống..... | 12 |
| 4.5. Tùy chọn tắt/bật cảnh báo..... | 13 |
| 5. Yêu cầu phi chức năng..... | 13 |
| 6. Cấu trúc hệ thống | 14 |
| 6.1. Lớp Cảm biến và Thiết bị..... | 14 |
| 6.2. Lớp mạng | 14 |
| 6.3 Lớp xử lý dữ liệu..... | 15 |
| 6.4. Lớp ứng dụng | 15 |
| 7. Sơ đồ tuần tự | 15 |
| 7.1 Luồng dữ liệu | 16 |
| 8. Kết quả đạt được | 17 |
| 8.1. Kết quả phần cứng | 17 |
| 8.2. Kết quả phần mềm và giao diện người dùng..... | 18 |
| 9. Kết luận và hướng mở rộng | 18 |
| 9.1. Kết luận | 18 |
| 9.2. Hướng phát triển và mở rộng | 18 |

1. Giới thiệu đề tài

Trong thời đại công nghệ 4.0, vấn đề an toàn cháy nổ trong nhà ở, văn phòng và các khu sản xuất ngày càng được quan tâm đặc biệt. Các vụ hỏa hoạn thường xảy ra bất ngờ, gây ra thiệt hại nghiêm trọng về người và tài sản, mà nguyên nhân chủ yếu là không có hệ thống giám sát và cảnh báo sớm. Vì vậy, việc ứng dụng công nghệ IoT (Internet of Things) vào lĩnh vực phòng cháy chữa cháy trở thành một hướng đi thiết thực và có ý nghĩa lớn trong đời sống hiện nay.

Từ nhu cầu thực tiễn đó, nhóm chúng em thực hiện đề tài “Hệ thống giám sát và cảnh báo cháy tự động”. Đề tài hướng tới việc xây dựng một mô hình thông minh, hoạt động ổn định, chi phí thấp nhưng có khả năng phát hiện và cảnh báo sớm khi xuất hiện khói hoặc lửa trong môi trường từ đó kích hoạt vòi phun và quạt máy để xử lý.

Hệ thống sử dụng cảm biến khói (MQ-2) để phát hiện khí cháy hoặc khói, cảm biến nhiệt độ DHT22 để đo nhiệt độ môi trường theo thời gian thực, cảm biến lửa (Flame Sensor) để phát hiện ngọn lửa và vi điều khiển ESP32 làm trung tâm xử lý dữ liệu. Dữ liệu sẽ được hiển thị trực tuyến trên ứng dụng điện thoại, giúp người dùng có thể theo dõi trạng thái hệ thống mọi lúc, mọi nơi thông qua điện thoại thông minh. Khi phát hiện nguy cơ cháy, hệ thống sẽ kích hoạt còi, đèn LED tại chỗ, hệ thống quạt khói và vòi phun sẽ tự hoạt động đồng thời gửi thông báo cảnh báo tức thời lên ứng dụng trên điện thoại.

Mục tiêu cụ thể :

- Thiết kế và xây dựng một hệ thống giám sát khói và nhiệt độ thông minh, hoạt động liên tục và ổn định trong thời gian dài.
- Phát hiện và cảnh báo sớm khi có khói hoặc nhiệt độ vượt ngưỡng cho phép, giúp người dùng xử lý kịp thời.
- Hiển thị dữ liệu cảm biến theo thời gian thực và gửi thông báo cảnh báo tự động đến ứng dụng điện thoại.
- Kích hoạt quạt hút khói , kích hoạt vòi phun nước khi phát hiện khói.
 - Hỗ trợ bật/tắt cảnh báo thủ công ngay trên thiết bị hoặc qua ứng dụng.
 - Tạo nền tảng cho việc phát triển hệ thống IoT giám sát an toàn quy mô lớn trong tương lai.

Ý nghĩa :

- Hệ thống có thể ứng dụng hiệu quả trong nhà ở, văn phòng, kho hàng, nhà xưởng hoặc phòng thí nghiệm, nơi có nguy cơ cháy nổ cao.
- Giúp người dùng giám sát môi trường từ xa, phản ứng nhanh khi có sự cố, và giảm thiểu thiệt hại về người và tài sản.
- Góp phần thúc đẩy ứng dụng IoT vào đời sống thực tế, đặc biệt trong lĩnh vực an toàn – bảo mật – giám sát thông minh.
- Đề tài còn mang giá trị học tập cao: giúp sinh viên rèn luyện kỹ năng lập trình nhúng, kết nối IoT, xử lý dữ liệu cảm biến, và làm việc nhóm trong môi trường kỹ thuật.
- Trong tương lai, mô hình có thể mở rộng thêm cảm biến khí gas (MQ-5), cảm biến CO, hoặc camera AI, giúp hệ thống trở nên đa năng và tự động hóa cao hơn, tiến tới ứng dụng trong nhà thông minh (Smart Home).

2. Mô tả tổng quan

2.1. Mô tả hệ thống

Hệ thống báo cháy tự động sử dụng Arduino Nano được thiết kế để phát hiện sớm các dấu hiệu cháy nổ thông qua cảm biến khói (MQ-2), cảm biến lửa (Flame Sensor) và cảm biến nhiệt độ (DHT 22). Khi phát hiện nồng độ khói hoặc nhiệt độ vượt ngưỡng cho phép, hệ thống sẽ kích hoạt còi báo động, nhấp đèn LED cảnh báo,

2.2. Môi trường hoạt động

- **Địa điểm triển khai:** Nhà ở, văn phòng, phòng thí nghiệm, kho hàng nhỏ.
- **Điều kiện môi trường:**
- Nhiệt độ: 0°C – 100°C
- Độ ẩm: 20% – 80%
- Nguồn điện: 5V DC

2.3. Giả định hệ thống

- Cảm biến được đặt tại vị trí có thể phát hiện khói/nhiệt độ/lửa hiệu quả (gần trần nhà, khu vực dễ cháy).
- Người sử dụng có thể nhận cảnh báo qua âm thanh hoặc tin nhắn.
- Hệ thống luôn được cấp điện và hoạt động liên tục.

3. Công nghệ và phần cứng sử dụng

3.1. Phần cứng

- **ESP32**



ESP32 là vi điều khiển mạnh mẽ, tích hợp Wi-Fi và Bluetooth, phù hợp cho các ứng dụng IoT. Trong hệ thống này, ESP32 đóng vai trò trung tâm, nhận dữ liệu từ các cảm biến, xử lý và điều khiển các thiết bị ngoại vi như còi, quạt, và máy bơm, đồng thời gửi dữ liệu lên ứng dụng di động thông qua mạng Internet.

- **Cảm biến lửa (Flame Sensor)**



Cảm biến lửa được sử dụng để phát hiện nguồn sáng có bước sóng trong khoảng 760 nm – 1100 nm, tương ứng với ngọn lửa. Khi phát hiện lửa, cảm biến sẽ gửi tín hiệu điện đến ESP32 để kích hoạt hệ thống cảnh báo.

- **Cảm biến khói và khí gas (MQ2)**



Cảm biến MQ2 có khả năng phát hiện các loại khí cháy như LPG, Propane, Methane và khói. MQ2 cung cấp dữ liệu liên tục về nồng độ khí trong không khí, giúp hệ thống phát hiện sớm nguy cơ cháy nổ và gửi cảnh báo.

- **Cảm biến nhiệt độ và độ ẩm (DHT22)**



DHT22 đo chính xác nhiệt độ và độ ẩm trong môi trường. Nhiệt độ tăng bất thường hoặc độ ẩm thay đổi đột ngột sẽ được ESP32 phân tích để xác định khả năng xảy ra hỏa hoạn, hỗ trợ cảnh báo sớm.

- **Còi báo động (Buzzer)**



Buzzer được sử dụng để phát ra âm thanh cảnh báo tại chỗ khi hệ thống phát hiện khói hoặc lửa. Còi được điều khiển trực tiếp bởi ESP32.

- **Đèn led**



Hệ thống sử dụng các đèn LED để hiển thị trạng thái hoạt động của hệ thống và hỗ trợ cảnh báo tại chỗ. LED đỏ được dùng để báo hiệu khi phát hiện cháy hoặc nguy cơ cháy. LED xanh thể hiện hệ thống đang hoạt động bình thường.

- **Relay**



Relay là thiết bị chuyển mạch điện, cho phép ESP32 điều khiển các thiết bị công suất cao như quạt hoặc máy bơm. Relay giúp hệ thống bật/tắt các thiết bị chữa cháy tự động an toàn và hiệu quả.

- **Màn hình LCD**



Màn hình LCD được tích hợp để hiển thị các thông tin theo thời gian thực của hệ thống. Các thông số được hiển thị bao gồm nhiệt độ và độ ẩm thu thập từ cảm biến DHT22, giá trị khói do cảm biến MQ2 đo được, trạng thái phát hiện lửa, trạng thái bật tắt cảnh báo.

- **Quạt và máy bơm**



Quạt: Khi phát hiện cháy, quạt có thể được bật để điều hòa môi trường, giảm khói hoặc hướng khói ra khỏi khu vực nguy hiểm.



Máy bơm: Máy bơm được sử dụng để phun nước tự động nhằm dập tắt ngọn lửa khi xảy ra cháy. Hoạt động của máy bơm được điều khiển thông qua relay bởi ESP32.

- **Nguồn 5 V**



Hệ thống sử dụng nguồn DC 5V ổn định để cấp cho ESP32 và các cảm biến, đồng thời có nguồn riêng cho relay và máy bơm để đảm bảo an toàn khi hoạt động.

3.2. Phần mềm

Môi trường phát triển:

- **Arduino IDE:** Được sử dụng để lập trình và nạp chương trình điều khiển và vi điều khiển (ESP32/Arduino). Đây là công cụ chính để viết mã, biên dịch và quản lý kết nối với phần cứng IoT.
- **Visual Studio Code:** Trình soạn thảo mã nguồn hiện đại, hỗ trợ nhiều ngôn ngữ lập trình và có nhiều tiện ích (extension). VS Code được sử dụng để phát triển cả frontend(ReactJS) và backend(NodeJS).

Frontend:

- **ReactJS:** Dùng để xây dựng giao diện người dùng (UI), cho phép hiển thị trực quan trạng thái của các thiết bị IoT, đồng thời cung cấp các nút điều khiển để tương tác trực tiếp với hệ thống.
- Ngôn ngữ: **JavaScript** - Ngôn ngữ chính để viết Logic cho giao diện, xử lý sự kiện và gọi API đến Backend Server.

Backend:

- **NodeJS với ExpressJS:** NodeJS cung cấp môi trường chạy JavaScript phía Server. ExpressJS là framework trên NodeJS, được sử dụng để xây dựng các API RESTful, giúp kết nối frontend với cơ sở dữ liệu và các thiết bị.
- Ngôn ngữ: **JavaScript:** Dùng để viết các dịch vụ Backend, xử lý yêu cầu từ Frontend, giao tiếp với cơ sở dữ liệu, và quản lý luồng dữ liệu từ các thiết bị IoT.

Database:

- **MongoDB Compass:** Công cụ giao diện trực quan để quản lý cơ sở dữ liệu MongoDB; lưu trữ dữ liệu cảm biến, lịch sử báo cháy, thời điểm kích hoạt cảnh báo.

4. Yêu cầu chức năng

Trong quá trình nghiên cứu và phát triển hệ thống, nhóm xác định các tính năng cơ bản sau nhằm đảm bảo hệ thống hoạt động hiệu quả, dễ sử dụng và đáng tin cậy:

4.1. Giám sát khói , lửa và nhiệt độ liên tục

- **Mô tả:** Hệ thống sử dụng cảm biến khói (MQ-2) , cảm biến lửa (Flame Sensor) cảm biến nhiệt độ (DHT22) để đo liên tục nồng độ khói, nhiệt độ và kiểm tra môi trường.
- **Cách hoạt động:**

Vi điều khiển ESP32 đọc giá trị analog/digital từ các cảm biến thông qua các chân ADC hoặc GPIO, sau đó so sánh với ngưỡng an toàn đã được lập trình sẵn (ví dụ: cảm biến lửa phát hiện lửa và nồng độ khói vượt mức 300ppm).

Khi phát hiện giá trị vượt ngưỡng, ESP32 sẽ chuyển hệ thống sang trạng thái cảnh báo.
- **Mục tiêu:** Phát hiện kịp thời các dấu hiệu cháy, tránh tình trạng cháy lan trước khi cảnh báo.

4.2. Cảnh báo tại chỗ bằng còi , đèn LED và hệ thống quạt, vòi phun tự động hoạt động

- **Mô tả:** Khi cảm biến phát hiện khói hoặc nhiệt độ vượt ngưỡng, ESP32 sẽ kích hoạt buzzer (còi) , đèn LED cảnh báo để báo hiệu trực tiếp tại khu vực xảy ra sự cố và từ đó quạt , vòi phun nước tự động hoạt động.
- **Cách hoạt động:**
 - ESP32 xuất tín hiệu digital HIGH ra các chân điều khiển (GPIO) kết nối với buzzer và LED.
 - Còi kêu liên tục, đèn LED nhấp nháy để thu hút sự chú ý.
 - Quạt gió sẽ hoạt động để hút khí độc , vòi phun sẽ tự phun nước để dập lửa.
 - Khi người dùng nhấn nút “tắt cảnh báo”, tín hiệu cảnh báo sẽ được tắt.

- **Mục tiêu:** Tạo cảnh báo trực quan (LED) và âm thanh (buzzer) giúp người xung quanh nhận biết nguy cơ cháy ngay lập tức, tăng khả năng phản ứng kịp thời.

4.3. Gửi thông báo từ xa (ứng dụng di động)

- **Mô tả:**
Gửi thông báo cháy từ ESP32 đến ứng dụng điện thoại của người dùng thông qua Internet.
Khi phát hiện nhiệt độ hoặc nồng độ khói vượt ngưỡng, ESP32 sẽ tự động gửi cảnh báo thời gian thực đến ứng dụng .
- **Cách hoạt động:**
 - ESP32 được kết nối Wi-Fi và truyền dữ liệu cảm biến trực tiếp đến sever của hệ thống qua giao thức MQTT
 - Khi cảm biến phát hiện sự cố:
Máy chủ xử lý và:
 - Lưu dữ liệu (nhiệt độ, nồng độ khói, trạng thái cảnh báo).
 - Gửi thông báo đẩy (push notification) đến ứng dụng trên điện thoại

Nếu người dùng đang mở ứng dụng, họ sẽ thấy:

Thông báo đẩy (push notification) với nội dung như: “ Cảnh báo cháy: Nhiệt độ 68°C, nồng độ khói cao!”

- **Mục tiêu:**
 - Cho phép người quản lý hoặc chủ nhà nhận cảnh báo ngay lập tức dù không có mặt tại hiện trường.
 - Giúp theo dõi tình trạng môi trường liên tục qua Internet.
 - Tăng tính tự động hóa và kết nối IoT cho hệ thống cảnh báo cháy.

4.4. Hiện thị trạng thái hệ thống

- **Mô tả:**
Hệ thống sẽ hiển thị tình trạng hiện tại (bình thường, nguy cơ cháy, cảnh báo) qua LED hoặc màn hình LCD nhỏ.
- **Cách hoạt động :**
ESP32 cập nhật trạng thái hệ thống dựa trên dữ liệu cảm biến trên giao diện hiển thị.
- **Mục tiêu:**

Giúp người dùng theo dõi trực quan và kiểm tra tình trạng hệ thống mà không cần thiết bị phụ trợ.

4.5. Tùy chọn tắt/bật cảnh báo

- **Mô tả:**

Hệ thống được trang bị nút nhấn để bật/tắt cảnh báo.

- **Cách hoạt động:**

- Khi nhấn nút bật/tắt cảnh báo, ESP32 sẽ tắt còi, LED ,relay và đưa hệ thống về trạng thái giám sát bình thường.
- Khi nhấn nút lại lần nữa, hệ thống bật lại chế độ cảnh báo ,LED,còi relay hoạt động lại bình thường khi phát hiện nguy cơ cháy.

- **Mục tiêu:**

Cho phép kiểm soát linh hoạt, tránh báo động sai hoặc gây phiền phức.

5. Yêu cầu phi chức năng

| Tiêu chí | Mô tả chi tiết |
|---|--|
| Hiệu năng (Performance) | Hệ thống phản hồi trong vòng < 1 giây kể từ khi cảm biến phát hiện khói hoặc lửa. ESP32 xử lý và kích hoạt cảnh báo tức thời. Hệ thống được thiết kế để hoạt động ổn định liên tục 24/7, không bị treo hoặc quá tải. |
| Độ tin cậy (Reliability) | Các cảm biến MQ-2 , Flame Sensor và DHT22 được hiệu chuẩn với sai số thấp đảm bảo dữ liệu đo chính xác. Hệ thống vẫn duy trì hoạt động ổn định khi nguồn điện ổn định (5V–9V). |
| Khả năng mở rộng (Scalability) | Hệ thống dễ dàng mở rộng bằng cách tích hợp thêm cảm biến khí gas, CO, CO ₂ , hoặc module khác như GPS, camera, relay điều khiển tự động. Có thể mở rộng sang các nền tảng IoT khác (MQTT, Firebase) mà không cần thay đổi phần cứng chính. |
| Ràng buộc kỹ thuật (Technical) | Sử dụng vi điều khiển ESP32 làm trung tâm xử lý. Nguồn hoạt động 5V DC. Tương thích với các cảm biến MQ-2 (khói),Flame |

| | |
|---|--|
| Constraints) | Sensor(Lửa) , DHT22 (nhiệt độ/độ ẩm), LED, buzzer, nút nhấn tắt cảnh báo. |
| Môi trường hoạt động (Operating Environment) | Cảm biến cần được đặt ở vị trí thoáng khí, không bị bụi bẩn hoặc độ ẩm cao để tránh sai lệch giá trị đo. Tránh gió mạnh hoặc nơi có khói nhân tạo (nhà bếp, bếp gas) gây báo động giả. Hệ thống hoạt động tốt trong môi trường b |

6. Cấu trúc hệ thống

6.1. Lớp Cảm biến và Thiết bị

- **Chức năng:**

Thu thập dữ liệu môi trường như nhiệt độ, độ ẩm, khói, khí gas và gửi về bộ xử lý trung tâm (ESP32). Đây là lớp đầu tiên, chịu trách nhiệm cảm nhận các biến động vật lý trong môi trường.

- **Thành phần chính:**

- **Thiết bị Cảm biến (Sensors):**

- **Cảm biến khói MQ-2:** Thu thập dữ liệu về nồng độ khói/gas.
- **Cảm biến nhiệt độ DHT22:** Thu thập dữ liệu nhiệt độ và độ ẩm.
- **Cảm biến lửa Flame Sensors:** Thu thập dữ liệu về sự hiện diện của lửa.

- **Thiết bị Hiện thực hóa (Actuators/Output Devices):**

- **Còi báo động (Buzzer):** Thực hiện hành động phát ra âm thanh cảnh báo.
- **Đèn LED cảnh báo:** Thực hiện hành động phát sáng cảnh báo.
- **Quạt hút khói và vòi phun nước:** Thực hiện các hành động vật lý để giảm thiểu nguy hiểm.

- **Thiết bị Điều khiển/Giao tiếp (Control/Interfacing):**

- **Module relay 5V 2 kênh:** Đóng vai trò là bộ điều khiển/chuyển mạch trung gian, dùng để bật/tắt các Actuator công suất lớn hơn như quạt hay vòi phun nước.

6.2. Lớp mạng

Lớp này bao gồm các giao thức và kênh truyền dữ liệu.

- **Vi xử lý ESP32** cũng là một phần của lớp này, vì nó tích hợp Module Wi-Fi để truyền dữ liệu đi (ví dụ: gửi cảnh báo qua MQTT/HTTP lên Cloud hoặc ứng dụng điện thoại).

6.3 Lớp xử lý dữ liệu

Lớp này chứa các thiết bị có khả năng tính toán, xử lý cục bộ, và làm cầu nối truyền thông.

- **Vi xử lý ESP32:** Đây là thiết bị trung tâm và là thành phần quan trọng nhất của lớp xử lý.
 - **Chức năng:** Nhận dữ liệu từ các cảm biến, thực hiện các thuật toán phân tích (ví dụ: nếu nhiệt độ cao và có khói), đưa ra quyết định điều khiển (bật còi, bật quạt), và gửi dữ liệu lên Cloud qua Wi-Fi.

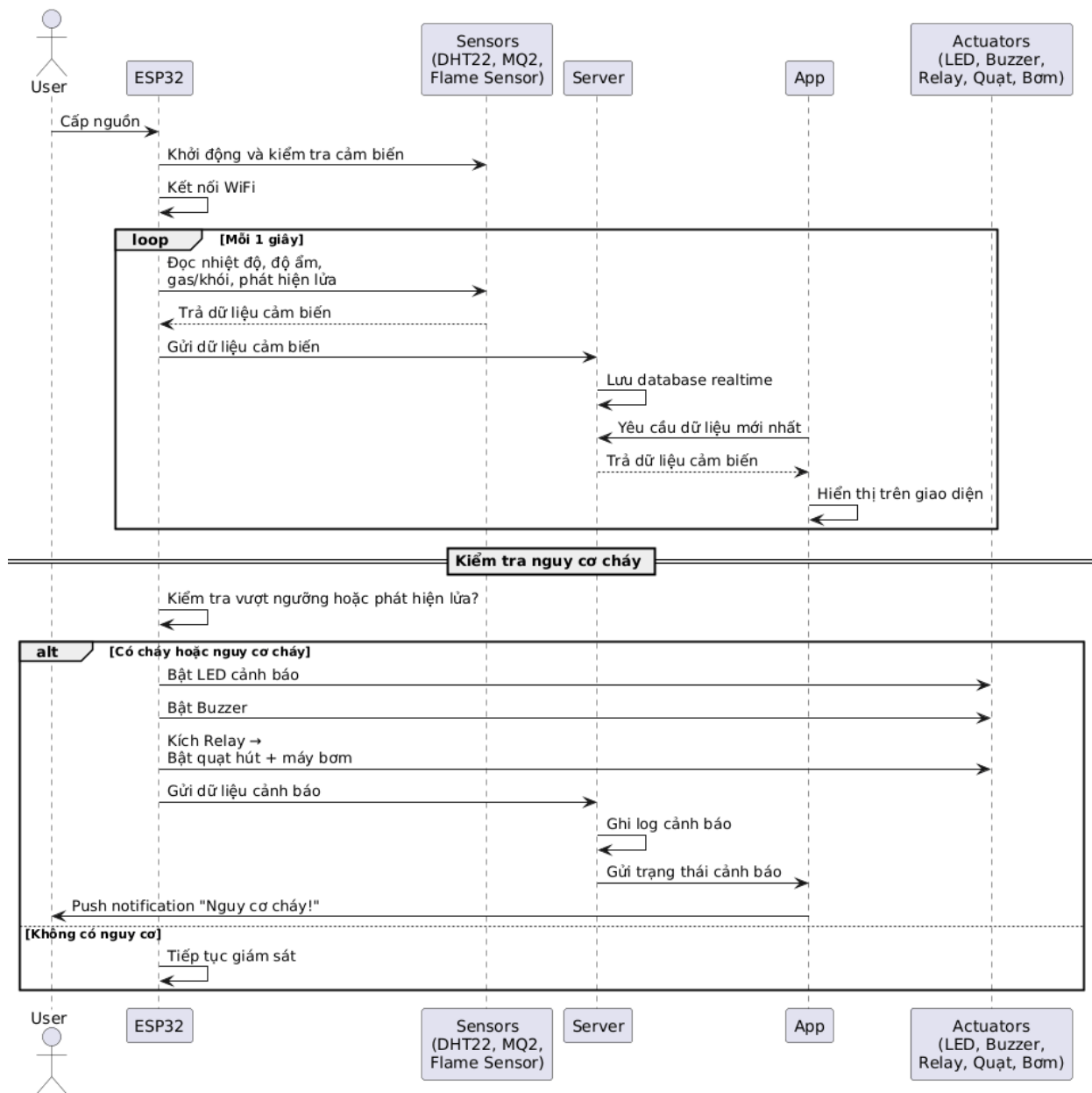
6.4. Lớp ứng dụng

Lớp này là nơi người dùng tương tác, và không có thiết bị vật lý nào trong danh sách của bạn nằm ở lớp này.

- **Các thành phần thuộc lớp này (không có trong danh sách):**
 - Ứng dụng di động để nhận cảnh báo.
 - Dashboard web để giám sát tình trạng hệ thống.
 - Máy chủ Cloud (Server) lưu trữ dữ liệu.

7. Sơ đồ tuần tự

Sequence Diagram - Hệ Thống ESP32 Giám Sát Cháy



7.1 Luồng dữ liệu

1. Cấp nguồn: Người dùng (User) cấp nguồn cho ESP32.
2. Khởi động Cảm biến: ESP32 khởi động và kiểm tra các Cảm biến (DHT22, MQ-2, Flame Sensor).
3. Kết nối Mạng: ESP32 thiết lập kết nối WiFi để sẵn sàng gửi/nhận dữ liệu.

4. Thu thập Dữ liệu: Cảm biến đọc và gửi dữ liệu môi trường (Nhiệt độ, Độ ẩm, Khói/Gas, Lửa) về ESP32.
5. Gửi lên Server: ESP32 truyền dữ liệu cảm biến đến Server.
6. Lưu trữ Dữ liệu: Server nhận dữ liệu và lưu trữ vào Database Realtime.
7. Yêu cầu Hiển thị: App yêu cầu dữ liệu mới nhất từ Server.
8. Cập nhật Giao diện: Server trả dữ liệu về App, và App hiển thị thông tin này trên giao diện người dùng.
9. Phân tích Nguy cơ: ESP32 tiến hành kiểm tra nội bộ: so sánh dữ liệu cảm biến với các Ngưỡng nguy hiểm hoặc phát hiện Phát hiện lửa.
- 10.(Trường hợp có Nguy cơ): Kích hoạt Cục bộ: ESP32 gửi lệnh kích hoạt ngay lập tức đến các Actuators (Bật LED cảnh báo và Bật Buzzer).
- 11.(Trường hợp có Nguy cơ): Kích hoạt Hành động: ESP32 kích hoạt Relay để điều khiển các thiết bị phản ứng (Bật Quạt hút và Bật Máy bơm).
- 12.(Trường hợp có Nguy cơ): Thông báo Server: ESP32 gửi thông tin Cảnh báo về Server.
- 13.(Trường hợp có Nguy cơ): Ghi Log: Server nhận dữ liệu và Ghi Log Cảnh báo vào Database.
- 14.(Trường hợp có Nguy cơ): Push Notification: Server gửi Push Notification thông báo "Nguy cơ cháy!" tới App (và người dùng).
- 15.(Kết thúc): Tiếp tục Giám sát: Hệ thống quay lại bước 4 để tiếp tục vòng lặp giám sát.

8. Kết quả đạt được

Sau quá trình nghiên cứu, thiết kế và thi công, nhóm đã hoàn thành mô hình "Hệ thống báo cháy thông minh" với các kết quả cụ thể về phần cứng và phần mềm như sau:

8.1. Kết quả phần cứng

Nhóm đã xây dựng thành công mô hình vật lý hoàn chỉnh, đảm bảo tính thẩm mỹ và độ bền cơ học.

- **Mạch điện:** Các module (ESP32, MQ-2, DHT22, Flame Sensor, Relay)

được kết nối chắc chắn, đi dây gọn gàng và hoạt động ổn định với nguồn 5V.

- **Cơ cấu chấp hành:** Hệ thống quạt hút khói và máy bơm mô phỏng hoạt động chính xác theo tín hiệu điều khiển từ Relay.
- **Đóng gói:** Mô hình được bố trí hợp lý trên sa bàn/hộp kỹ thuật, mô phỏng không gian phòng cần giám sát.

8.2. Kết quả phần mềm và giao diện người dùng

Hệ thống giám sát từ xa hoạt động mượt mà trên nền tảng Web App/Mobile App (ReactJS):

- **Dashboard giám sát:** Hiện thị trực quan các thông số theo thời gian thực: Nhiệt độ (độ C) , Độ ẩm (%), Nồng độ khói (PPM) và Trạng thái lửa (Có/Không).
- **Độ trễ:** Tốc độ cập nhật dữ liệu từ cảm biến lên ứng dụng đạt mức thấp (khoảng 1-2 giây), đảm bảo tính tức thời.
- **Lịch sử:** Ghi nhận và lưu trữ thành công dữ liệu vào MongoDB để phục vụ tra cứu.

9. Kết luận và hướng mở rộng

9.1. Kết luận

Đề tài "Hệ thống báo cháy thông minh" đã hoàn thành được các mục tiêu đề ra ban đầu.

- **Về mặt kỹ thuật:** Nhóm đã làm chủ được công nghệ vi điều khiển ESP32, giao thức truyền nhận dữ liệu qua Wi-Fi/MQTT và xây dựng thành công ứng dụng giám sát Fullstack (NodeJS + ReactJS).
- **Về mặt thực tiễn:** Hệ thống hoạt động ổn định, có khả năng phát hiện sớm các nguy cơ cháy nổ thông qua đa cảm biến (Khói, Nhiệt, Lửa) và thực hiện các biện pháp xử lý tức thời (Báo động, Chữa cháy tự động).
- **Hạn chế:** Do giới hạn về kinh phí và thiết bị mô phỏng, độ chính xác của cảm biến MQ-2 và Flame Sensor có thể bị ảnh hưởng bởi nhiều môi trường (ánh sáng mạnh, luồng gió tự nhiên).

9.2. Hướng phát triển và mở rộng

Để hệ thống có thể ứng dụng rộng rãi và tin cậy hơn trong thực tế, nhóm đề xuất các hướng phát triển trong tương lai:

1. **Tích hợp Camera AI (Computer Vision):** Sử dụng ESP32-CAM hoặc tích hợp Camera IP để xác thực đám cháy bằng hình ảnh. Áp dụng AI để phân biệt giữa khói thuốc lá, hơi nước và khói cháy thực sự, giúp giảm thiểu báo động giả.
2. **Nâng cấp hạ tầng kết nối:** Trong trường hợp hỏa hoạn gây mất điện và mất Wi-Fi, hệ thống cần tích hợp thêm module SIM (GSM/4G) để gửi tin nhắn SMS hoặc thực hiện cuộc gọi khẩn cấp đến chủ nhà và cơ quan chức năng. Sử dụng công nghệ LoRa nếu triển khai trong các khu xưởng rộng lớn.
3. **Nguồn điện dự phòng:** Tích hợp mạch sạc và pin dự phòng (UPS mini) để đảm bảo hệ thống vẫn hoạt động liên tục ngay cả khi lưới điện bị ngắt do cháy.
4. **Tích hợp Nhà thông minh (Smart Home):** Kết nối hệ thống với các nền tảng như Google Home hoặc Apple HomeKit, cho phép điều khiển bằng giọng nói và kết hợp với các kịch bản tự động hóa khác (ví dụ: Tự động mở cửa cuốn, bật đèn thoát hiểm khi có báo cháy).