

POST AND TELECOMMUNICATION INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY 1



IOT PROJECT

Class:	E22HTTT - 05
Group:	06
Topic:	Hệ thống nhà thông minh
Members:	Nguyễn Hải Quang – B22DCVT307 Nguyễn Phi Nhật – B22DCCN580 Trần Xuân Kiên – B22DCVT269 Nguyễn Hoàng Nguyên – B22DCVT383
Mentors:	Kim Ngọc Bách

MỤC LỤC

LỜI MỞ ĐẦU.....	4
I. Tổng quan đề tài.....	5
1. Giới thiệu đề tài.....	5
2. Lý do chọn đề tài:.....	5
3. Mục tiêu của đề tài:.....	6
4. Phạm vi triển khai:.....	6
II. Xác định yêu cầu chức năng (Function Requirements).....	6
III. Xác định yêu cầu phi chức năng (Non - Functional Requirements).....	7
IV. Phân tích ràng buộc (Constraints Analysis).....	8
V. Mô hình hóa yêu cầu.....	9
VI. Công nghệ sử dụng (Technology).....	9
1. Giao thức truyền tin (MQTT).....	9
2. Nền tảng ứng dụng: Native Mobile App (Android Studio).....	11
VII. Thiết bị sử dụng (Equipment).....	12
1. Vi điều khiển(Microcontroller) : ESP32.....	12
2. Bộ chấp hành(Actuator): Servo Motor.....	13
VIII. Phân công Công việc và Kế hoạch.....	14

LỜI MỞ ĐẦU

Trong thời đại của Internet, của kỹ thuật số như hiện nay, dù ít dù nhiều mỗi cá nhân đều phải tiếp xúc với công nghệ trong cuộc sống hằng ngày để phục vụ cho nhu cầu của bản thân. Và với sự phát triển của công nghệ, Internet of Things (viết tắt là IoT) được coi là một trong những xu hướng nổi bật nhất trong cuộc cách mạng công nghệ 4.0 cùng điện toán đám mây, AI, tự động hóa,... Học viện toàn cầu McKinsey dự đoán tăng trưởng giá trị lớn cho Internet of Things (11,1 nghìn tỷ đô la hàng năm vào năm 2025). Trong đó, Nhà thông minh, một môi trường gia đình được kết nối đầy đủ cung cấp cho cư dân của mình một mức độ kiểm soát và sự thoải mái chưa từng có, sẽ chiếm một phần lớn của sự gia tăng này.

Sự xuất hiện các Nhà thông minh (Smarthome) sẽ góp phần tạo nên một diện mạo đô thị hiện đại hoàn chỉnh. Về bản chất, mỗi căn hộ thông minh sẽ có vai trò giúp đô thị thông minh được vận hành trơn tru, đáp ứng đa dạng nhu cầu sống của cư dân. Do đó, trong tổng thể một khu đô thị thông minh không thể thiếu những “mắt xích” thông minh.

Với ưu thế và lợi ích của nhà ở thông minh so với những nhà ở bình thường, đây sẽ là xu thế phát triển tất yếu của thị trường nhà ở trong tương lai.

I. Tổng quan đề tài

1. Giới thiệu đề tài

Nhà thông minh (Smart Home) là một hệ thống sử dụng công nghệ để điều khiển và giám sát các thiết bị trong ngôi nhà một cách tự động hoặc từ xa. Đề tài này hướng tới việc xây dựng một hệ thống nhà thông minh có khả năng:

- Điều khiển thiết bị điện (đèn, ...) từ xa qua smartphone hoặc web.
- Giám sát trạng thái thiết bị, cảm biến theo thời gian thực.

Firestore được chọn làm nền tảng lưu trữ và đồng bộ dữ liệu theo thời gian thực, giúp đảm bảo khả năng kết nối và cập nhật trạng thái thiết bị một cách nhanh chóng, hiệu quả.

2. Lý do chọn đề tài:

- **Sự phát triển của công nghệ IoT:**
Trong thời đại công nghệ 4.0, IoT (Internet of Things) đang trở thành một xu hướng tất yếu, góp phần thay đổi cách con người sinh sống và làm việc. Việc áp dụng IoT vào mô hình nhà ở giúp tăng tính tiện nghi, an toàn và tiết kiệm năng lượng.
- **Nhu cầu thực tế của người dùng:**
Cuộc sống hiện đại đòi hỏi sự tiện lợi và an toàn cao hơn. Việc điều khiển, giám sát các thiết bị trong nhà từ xa qua smartphone hoặc tự động hóa hoạt động hàng ngày giúp người dùng tiết kiệm thời gian, công sức và đảm bảo an ninh cho ngôi nhà.
- **Tính ứng dụng cao:**
Hệ thống nhà thông minh có thể được áp dụng thực tế trong các hộ gia đình, văn phòng hoặc khu dân cư, giúp nâng cao chất lượng cuộc sống. Đây cũng là lĩnh vực có tiềm năng phát triển mạnh trong tương lai.
- **Nâng cao kiến thức và kỹ năng:**
Việc thực hiện đề tài giúp nhóm sinh viên có cơ hội tiếp cận, thực hành và tích lũy kinh nghiệm về các công nghệ hiện đại như IoT, cảm biến, vi điều khiển, lập trình phần mềm và cơ sở dữ liệu thời gian thực (Firestore).
- **Phù hợp với xu hướng chuyển đổi số:**
Đề tài phù hợp với định hướng phát triển công nghệ thông minh và đô thị thông minh, góp phần vào xu thế chuyển đổi số quốc gia.

3. Mục tiêu của đề tài:

- Xây dựng hệ thống nhà thông minh (Smarthome) giúp người dùng có thể điều khiển và giám sát các thiết bị trong nhà từ xa qua Internet hoặc qua Web.
- Tăng tính tiện nghi, an toàn và tiết kiệm năng lượng cho người dùng.

4. Phạm vi triển khai:

- Điều khiển và giám sát các thiết bị điện cơ bản: đèn, cửa ra vào.
- Theo dõi các thông số môi trường như: ánh sáng.
- Ứng dụng chạy trên web hoặc điện thoại di động (qua Wifi/Internet).

II. Xác định yêu cầu chức năng (Functional Requirements)

No	Chức năng	Mô tả chi tiết
1	Cảnh báo rò rỉ khí gas	Hệ thống sử dụng cảm biến Gas MQ-series để phát hiện rò rỉ khí gas. Khi nồng độ vượt ngưỡng, ESP32 kích hoạt còi Buzzer và đèn LED cảnh báo.
2	Điều khiển đèn tự động	Khi giá trị ánh sáng thu được từ cảm biến LDR thấp hơn ngưỡng cài đặt, ESP32 sẽ tự động bật đèn LED tương ứng. Khi đủ sáng, hệ thống tự động tắt đèn.
3	Mở cửa bằng nhận diện khuôn mặt	Khi hệ thống nhận diện khuôn mặt hợp lệ, ESP32 gửi tín hiệu điều khiển servo Face Servo để mở cửa tự động
4	Mở cửa bằng thẻ RFID	Khi người dùng quét thẻ RFID hợp lệ , ESP32 xác thực và điều khiển servo Face Servo để mở cửa.

III. Xác định yêu cầu phi chức năng (Non - Functional Requirements)

Chất lượng	Yêu cầu đầu ra	Công nghệ giải quyết
Hiệu năng	Độ trễ giao diện phải dưới 3s.	Giao thức MQTT nhẹ
Độ tin cậy	Hệ thống phải hoạt động ổn định, duy trì kết nối khi mạng yếu.	MQTT có khả năng reconnect, Firebase Realtime Database
Mở rộng	Dễ dàng thêm thiết bị mới (thêm ESP32 mới) mà không cần thay đổi cấu trúc hệ thống..	Cấu trúc hệ thống sử dụng MQTT Broker trung tâm, hỗ trợ nhiều client kết nối song song
Giao diện	Giao diện dễ sử dụng, hiển thị dữ liệu theo thời gian thực.	HTML, CSS, JavaScript (Web Dashboard)
Chi phí	Chi phí linh kiện trong tầm giá rẻ(4)	Sử dụng ESP32,cảm biến phổ thông

IV. Phân tích ràng buộc (Constraints Analysis)

Ràng buộc	Chi tiết	Tác động đến dự án
Môi trường	Hệ thống hoạt động trong môi trường nhà ở, văn phòng hoặc mô hình thu nhỏ trong phòng thí nghiệm. Cần đảm bảo nguồn điện ổn định, tránh nơi ẩm ướt, bụi bẩn hoặc nhiễu sóng.	Nếu môi trường không ổn định (mất điện, Wifi yếu, ẩm mốc) có thể gây lỗi truyền dữ liệu, giảm tuổi thọ thiết bị và ảnh hưởng đến độ chính xác của cảm biến.

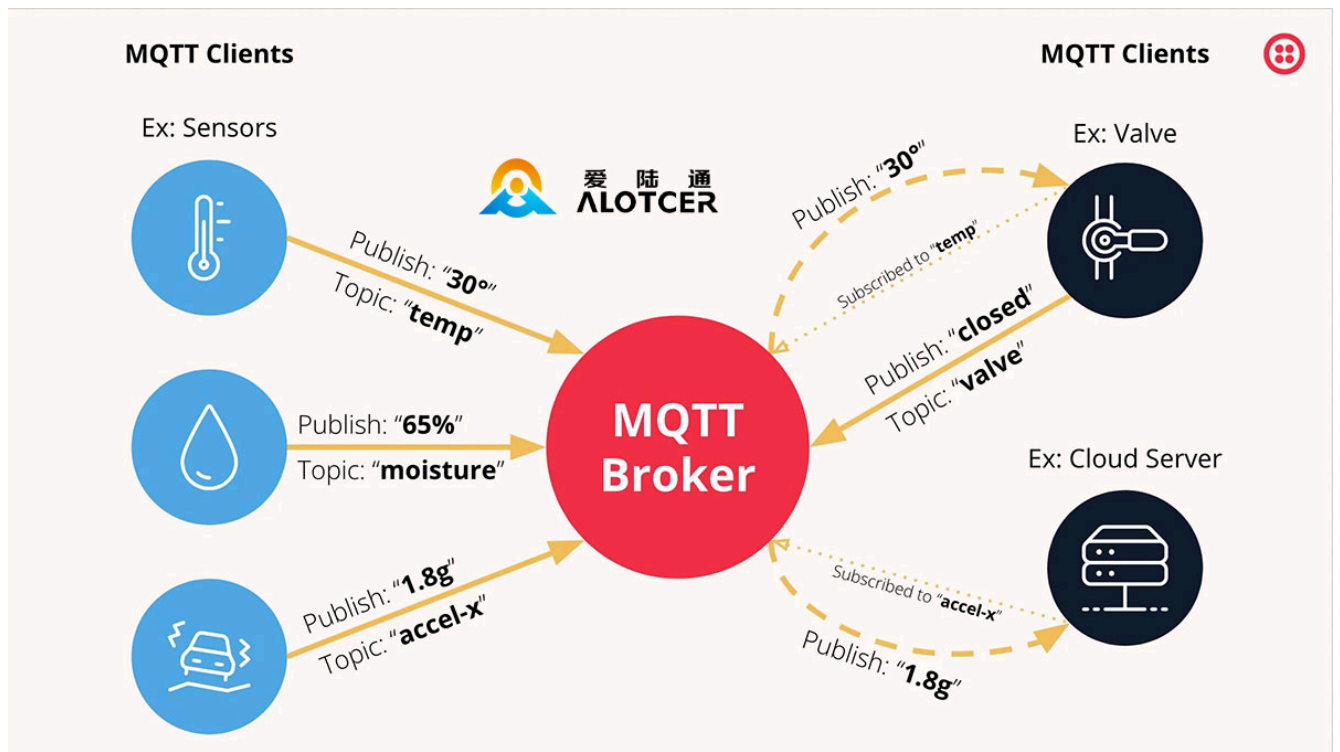
Cấu hình mạng	Các thiết bị IoT kết nối với mạng Wifi nội bộ và server qua giao thức MQTT/HTTP. Cần có địa chỉ IP ổn định và bảo mật truy cập.	Nếu mạng yếu hoặc mất kết nối Internet, hệ thống không thể điều khiển hoặc giám sát từ xa. Việc cấu hình sai địa chỉ IP hoặc lỗi bảo mật có thể khiến hệ thống dễ bị tấn công.
Thời gian	Dự án được thực hiện trong khoảng thời gian dài, bao gồm giai đoạn thiết kế, lắp ráp phần cứng, lập trình và kiểm thử. Thời gian phản hồi của hệ thống phải ngắn để đảm bảo tính “thời gian thực”.	Thời gian phát triển ngắn yêu cầu chia nhỏ công việc hợp lý, ưu tiên tính năng chính. Nếu không đảm bảo tiến độ, hệ thống có thể chưa hoàn thiện hoặc thiếu tính ổn định khi nộp báo cáo.

V. Mô hình hóa yêu cầu

	Thành phần	Mô tả chi tiết
Cảm nhận	ESP32, LDR, Servo, Relay	- Thu thập dữ liệu LDR và thực thi lệnh Relay(Output)
Mạng	Wi-fi, HiveMQ Broker	- Truyền dữ liệu cảm biến từ ESP32 đến server - Nhận lệnh điều khiển từ ứng dụng về ESP32.
Application	MQTT Client Library, Giao diện web(HTML, CSS)	- Truyền dữ liệu cảm biến từ ESP32 đến server - Nhận lệnh điều khiển từ ứng dụng về ESP32. - Hiển thị dữ liệu cảm biến theo thời gian thực.

VI. Công nghệ sử dụng (Technology)

1. Giao thức truyền tin (MQTT)



Lý do sử dụng:

1. Nhẹ và tiết kiệm tài nguyên:

- MQTT được thiết kế dành riêng cho các thiết bị IoT nên sử dụng rất ít băng thông mạng.
- Phù hợp với các vi điều khiển có tài nguyên hạn chế như **ESP32**, **ESP8266**.

2. Tốc độ truyền nhanh & độ trễ thấp:

- Hoạt động theo mô hình **Publish – Subscribe**, giúp dữ liệu được gửi và nhận gần như tức thời.
- Tốc độ phản hồi nhanh hơn nhiều so với các giao thức HTTP truyền thống.

3. Dễ triển khai và mở rộng:

- Chỉ cần một **MQTT Broker** (như Mosquitto), có thể kết nối nhiều thiết bị cùng lúc.
- Phù hợp với các hệ thống có quy mô từ nhỏ (1–2 thiết bị) đến lớn (nhiều cảm biến, thiết bị điều khiển).

4. Hoạt động ổn định trong mạng yếu:

- MQTT có khả năng giữ kết nối và tự động gửi lại gói tin nếu mạng bị gián đoạn.
- Giúp hệ thống nhà thông minh hoạt động ổn định hơn trong điều kiện Wi-Fi không mạnh.

5. Tích hợp dễ dàng với nhiều nền tảng:

- Có thư viện hỗ trợ cho hầu hết các ngôn ngữ lập trình (Python, C/C++, JavaScript, v.v.)
- Tương thích tốt với **Arduino**, **ESP32**, và nhiều dịch vụ đám mây IoT.

6. Hỗ trợ gửi thông báo thời gian thực:

- Khi có sự kiện (ví dụ rò rỉ khí gas), MQTT giúp gửi cảnh báo đến người dùng ngay lập tức.

2. Giao diện Web (HTML, CSS, JavaScript)



Lý do sử dụng:

1. Dễ xây dựng và triển khai:

- HTML, CSS và JavaScript là những công nghệ phổ biến, dễ học và dễ triển khai cho các dự án IoT nhỏ.

- Không cần cài đặt phần mềm phức tạp, chỉ cần trình duyệt web là có thể sử dụng hệ thống.

2. Giao diện trực quan, thân thiện:

- Cho phép thiết kế giao diện điều khiển thiết bị một cách trực quan.
- Người dùng có thể bật/tắt thiết bị, theo dõi cảm biến thông qua các nút bấm, biểu đồ, hoặc dashboard.

3. Kết nối thời gian thực:

- JavaScript có thể kết hợp với WebSocket hoặc MQTT qua Web để cập nhật dữ liệu cảm biến liên tục mà không cần tải lại trang.
- Phù hợp cho các ứng dụng giám sát nhà thông minh.

4. Khả năng hoạt động đa nền tảng:

- Chạy được trên mọi thiết bị có trình duyệt (máy tính, điện thoại, máy tính bảng).
- Giúp người dùng điều khiển thiết bị từ bất kỳ đâu.

5. Dễ mở rộng và tùy chỉnh:

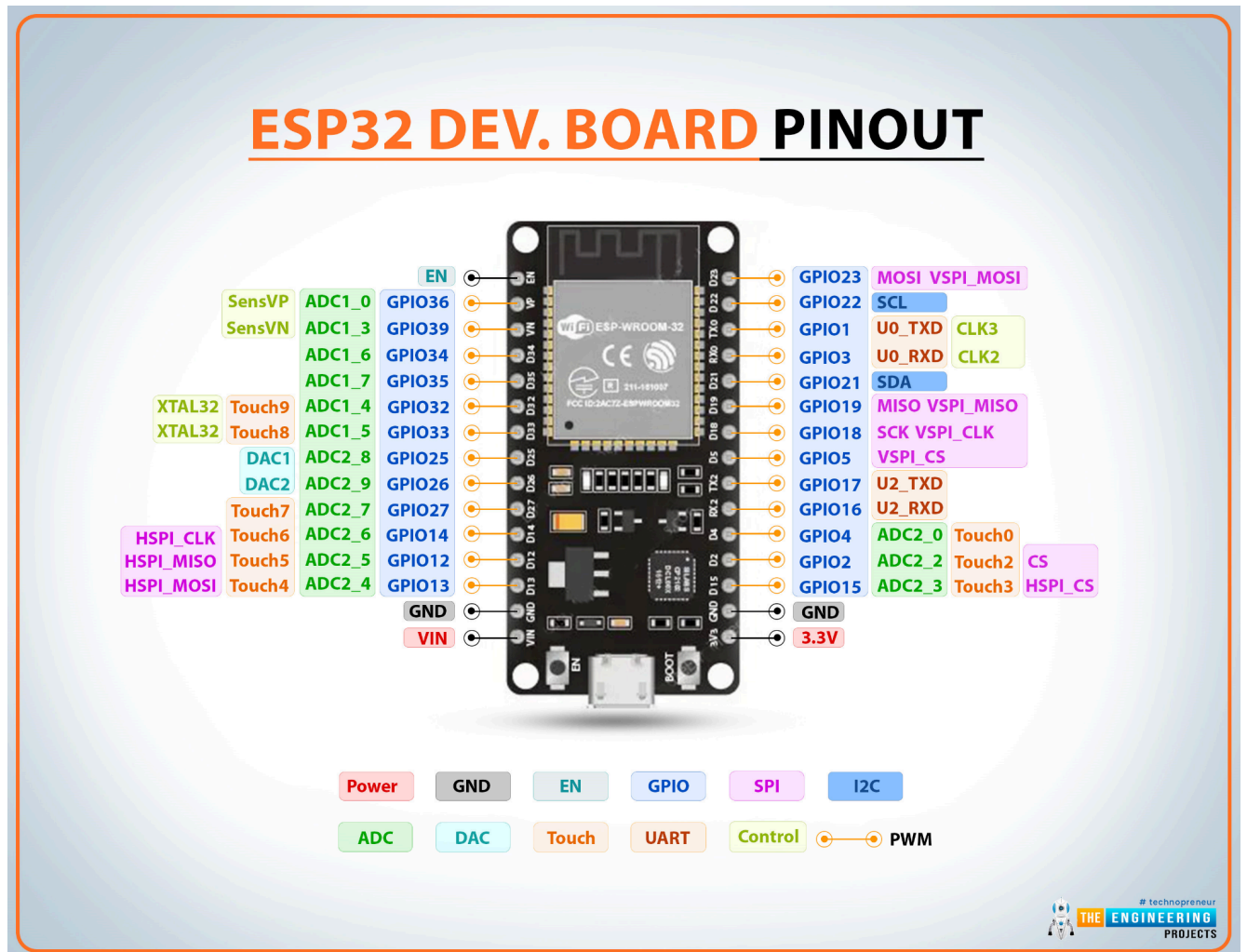
- Có thể dễ dàng tích hợp thêm biểu đồ (Chart.js), bản đồ, hoặc kết nối với các dịch vụ khác.
- Phù hợp với việc phát triển thêm các tính năng nâng cao trong tương lai.

6. Tách biệt giao diện và điều khiển:

- Giúp hệ thống rõ ràng: ESP32 tập trung vào thu thập – điều khiển thiết bị, còn giao diện Web chỉ đảm nhận việc hiển thị và gửi lệnh.

VII. Thiết bị sử dụng (Equipment)

1. Vi điều khiển(Microcontroller) : ESP32



Lý do sử dụng:

1. Chi phí thấp và thông dụng mà hiệu năng cao.
2. Tích hợp Wifi và Bluetooth không cần phải mở rộng.
3. Có nhiều GPIO để kết nối cảm biến.
4. Hỗ trợ lập trình dễ dàng qua môi trường Arduino, phù hợp cho người bắt đầu.

2. Bộ chấp hành(Actuator): Servo Motor



Lý do sử dụng:

1. Điều khiển chính xác

- Servo có thể quay đến một góc xác định (0–180° hoặc 360° tùy loại).
- Rất phù hợp để mở/đóng cửa, rèm hoặc điều chỉnh góc thiết bị theo yêu cầu.

2. Phản hồi nhanh – hoạt động ổn định

- Servo phản hồi nhanh với tín hiệu điều khiển từ ESP32, giúp hệ thống hoạt động gần như theo thời gian thực.

3. Tương thích dễ dàng với ESP32

- Servo có thể điều khiển trực tiếp bằng xung PWM, mà ESP32 có sẵn các chân PWM → dễ lập trình, không cần mạch điều khiển phức tạp.

4. Tiêu thụ điện năng thấp

- Phù hợp cho các thiết bị IoT chạy lâu dài, tiết kiệm năng lượng hơn so với motor DC thông thường.

5. Kích thước nhỏ gọn – dễ lắp đặt

- Servo thường nhỏ, nhẹ, dễ tích hợp vào hệ thống cửa thông minh, rèm, cửa sổ... mà không chiếm nhiều không gian.

6. Chi phí thấp – phổ biến

- Dễ mua, giá rẻ, nhiều tài liệu hỗ trợ → thuận tiện cho việc triển khai thử nghiệm và thực tế.

VIII. Phân công Công việc và Kế hoạch

Kế Hoạch Thực Hiện Dự Án Nhà Thông Minh (5 Tuần)

Mục tiêu: Hoàn thành hệ thống Smarthome cốt lõi với khả năng Điều khiển từ xa qua Web Dashboard, Giám sát môi trường (LDR, Khí Gas) và Kiểm soát truy cập (RFID, Nhận diện Khuôn mặt).

0.0	Tuần				
	Tuần 1:	Tuần 2:	Tuần 3:	Tuần 4:	Tuần 5:
Mục tiêu của tuần →	Thiết lập Nền tảng & Kết nối LDR/Firebase	Tích hợp MQTT, Điều khiển Web & Cảnh báo Gas	Kiểm soát Truy cập (RFID) & Tự động hóa	Nhận diện Khuôn mặt & Tích hợp Toàn diện	Kiểm thử, Sửa lỗi & Báo cáo
Nguyễn Hải Quang (Trưởng nhóm, Kiến trúc & Tích hợp)	<div>- Thiết kế Kiến trúc Hệ thống (Data Flow, Schema Firebase)</div> <div>- Phân tích yêu cầu tích hợp MQTT giữa Web và ESP32.</div>	<div>- Kiểm thử hiệu năng MQTT end-to-end (Độ trễ lệnh).</div> <div>- Đảm bảo đồng bộ trạng thái thiết bị giữa Web Firebase.</div>	<div>- Tài liệu hóa quy trình Mở/Đóng cửa tự động.</div> <div>- Kiểm tra tính ổn định của Servo khi nhận lệnh từ RFID/Web.</div>	<div>- Tích hợp & Kiểm thử Toàn diện (End-to-end): Từ Nhận diện → MQTT → Servo.</div> <div>- Ghi nhận và Phân loại lỗi của toàn bộ hệ thống.</div>	<div>- Tổ chức Buổi Kiểm thử Cuối cùng (Stress Test các kịch bản).</div> <div>- Tổng hợp Báo cáo Kỹ thuật và Kết quả. - Chuẩn bị Bài Thuyết trình và Demo.</div>

<p>Nguyễn Phi Nhật (Phần cứng & ESP32)</p>	<ul style="list-style-type: none"> - Thiết lập môi trường Arduino IDE/PlatformIO cho ESP32. - Đấu nối, lập trình LDR/Relay/LED và kiểm tra hoạt động. - Tích hợp Firebase: Gửi dữ liệu LDR lên DB. 	<ul style="list-style-type: none"> - Tích hợp MQTT Client trên ESP32. - Lập trình ESP32 nhận lệnh điều khiển Relay qua MQTT (ON/OFF). - Tích hợp Cảm biến Gas và kích hoạt Buzzer/LED tại chỗ. 	<ul style="list-style-type: none"> - Tích hợp RFID Reader vào ESP32 (Đọc thẻ). - Lập trình Servo Motor điều khiển cửa. - Xây dựng logic Tự động hóa (Ví dụ: Tự động bật đèn theo LDR). 	<ul style="list-style-type: none"> - Phát triển API/MQTT Logic trên ESP32 để nhận lệnh mở cửa từ hệ thống Python Nhận diện Khuôn mặt. 	<ul style="list-style-type: none"> - Tối ưu hóa code ESP32 (MQTT Reconnect, Stability). - Đảm bảo phần cứng hoạt động ổn định trong mọi tình huống.
<p>Trần Xuân Kiên (Phát triển Web Dashboard)</p>	<ul style="list-style-type: none"> - Thiết lập Web Dev Environment (Front-end Framework/Library). - Thiết kế Wireframe và Khởi tạo UI cho Dashboard (Màn hình chính). 	<ul style="list-style-type: none"> - Tích hợp MQTT Web Client (WebSocket) và Firebase SDK. - Phát triển chức năng Điều khiển ON/OFF từ Dashboard (Publish MQTT). - Phát triển giao diện hiển thị dữ liệu Khí Gas. 	<ul style="list-style-type: none"> - Phát triển màn hình Cảnh báo (pop-up khi Gas/Truy cập trái phép). - Phát triển Màn hình Cài đặt cho các Ngưỡng Tự động hóa (LDR). 	<ul style="list-style-type: none"> - Hoàn thiện UI/UX và tính Responsive cho tất cả các màn hình Dashboard. - Phát triển màn hình Quản lý Người dùng/Thẻ (Admin View). 	<ul style="list-style-type: none"> - Sửa lỗi giao diện, tối ưu tốc độ tải và phản hồi của Dashboard. - Đảm bảo tính năng Cảnh báo hoạt động chính xác.
<p>Nguyễn Hoàng Nguyên (Nhận diện & CSDL)</p>	<ul style="list-style-type: none"> - Thiết lập Firebase Project (Authentication, Realtime DB). 	<ul style="list-style-type: none"> - Định nghĩa các Topic MQTT tiêu chuẩn. - Thiết lập Logic Cảnh báo 	<ul style="list-style-type: none"> - Thiết lập CSDL Thẻ RFID trên Firebase. - Phát triển module Python 	<ul style="list-style-type: none"> - Phát triển module Nhận diện Khuôn mặt (Python/Open CV). 	<ul style="list-style-type: none"> - Sửa lỗi module Nhận diện (tăng tốc độ xử lý). - Tối ưu hóa Firebase

Firebase)	- Xây dựng cấu trúc dữ liệu (schema) cho Cảm biến/Thiết bị.	trên Firebase (kiểm tra ngưỡng Gas).	để xử lý dữ liệu Thẻ RFID (nếu cần xử lý phức tạp).	- Lập trình Logic: Nhận diện thành công → Publish lệnh mở cửa qua MQTT.	Security Rules.
-----------	---	--------------------------------------	---	---	-----------------

Module

1. Giám sát & Điều khiển (Monitoring & Control)

- Cho phép người dùng xem trạng thái môi trường và điều khiển thiết bị tức thời.
- Hiển thị trạng thái Gas/LDR. Điều khiển ON/OFF (Đèn, Quạt).

2. An ninh & Kiểm soát truy cập (Security & Access)

- Quản lý người dùng, thẻ RFID và theo dõi các sự kiện mở cửa, cảnh báo.
- Quản lý thẻ RFID (Thêm/Xóa). Nhật ký mở cửa (RFID, Khuôn mặt).

3. Thiết lập Tự động hóa (Automation Settings)

- Cho phép người dùng tùy chỉnh các ngưỡng cảm biến và kịch bản tự động.
- Cài đặt ngưỡng LDR (Tự động bật/tắt đèn). Cài đặt ngưỡng Gas (Kích hoạt cảnh báo).

USE CASE DIAGRAM - HỆ THỐNG GIÁM SÁT, ĐIỀU KHIỂN & TỰ ĐỘNG HÓA IOT (GÓC NHÌN NGƯỜI DÙNG)

