

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**BÁO CÁO
THỰC TẬP ĐỒ ÁN MÔN HỌC ĐA NGÀNH (CO3109)
ĐỀ TÀI
HỆ THỐNG BÁO ĐỘNG AN TOÀN CHO GIA ĐÌNH**

Nhóm (Tổ) : 82
Học kỳ : 242 (Năm học 2024-2025)
Giảng viên hướng dẫn : Mai Đức Trung
Sinh viên thực hiện : Trương Minh Thông (L01) 2153005
Nguyễn Anh Duy (L02) 2233163
Nguyễn Thành Đạt (L02) 2012938
Nguyễn Ngọc Phú (L02) 2212588
Nguyễn Viết Ký (L03) 2151217

TP. Hồ Chí Minh, 05/06/2025

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	1
1.1. Hệ thống báo động an toàn cho gia đình	1
1.2. Yêu cầu của hệ thống	1
1.2.1. Yêu cầu chức năng	1
1.2.2. Yêu cầu phi chức năng	1
1.2.3. Phạm vi và điều kiện hoạt động	2
1.3. Phân công nhiệm vụ trong nhóm	2
CHƯƠNG 2. THIẾT KẾ HỆ THỐNG	3
2.1. Thiết bị sử dụng	3
2.2. Mô tả hoạt động hệ thống	4
2.3. Công nghệ sử dụng	6
2.4. Sử dụng Design Pattern	7
CHƯƠNG 3. USE CASE CHI TIẾT VÀ SƠ ĐỒ HOẠT ĐỘNG, SƠ ĐỒ LỚP CỦA HỆ THỐNG	8
3.1. Sơ đồ Use Case của hệ thống	8
3.2. Đặc tả Use Case của hệ thống	8
3.3. Sơ đồ hoạt động của hệ thống	11
3.4. Sơ đồ lớp của hệ thống	12
CHƯƠNG 4. GIAO DIỆN NGƯỜI DÙNG VÀ DEMO SẢN PHẨM	14
4.1. Demo sản phẩm	14
4.2. Chạy hệ thống	14
4.3. Giao diện người dùng	15
CHƯƠNG 5. KẾT LUẬN	17
CHƯƠNG 6. TÀI LIỆU THAM KHẢO	18

Danh sách hình vẽ

1	Thiết bị sử dụng trong hệ thống	3
2	Sơ đồ khối kiến trúc hệ thống	5
3	Cơ sở dữ liệu tài khoản người dùng trên Firebase	6
4	Lưu trữ dữ liệu và cài đặt hệ thống trên Firebase	6
5	Sơ đồ Use Case của hệ thống	8
6	Sơ đồ hoạt động cho cập nhật dữ liệu từ cảm biến và điều khiển thiết bị . .	11
7	Sơ đồ hoạt động cho đăng nhập và đăng xuất, quên mật khẩu	11
8	Sơ đồ hoạt động cho báo động có đột nhập	12
9	Sơ đồ hoạt động cho tạo-xóa-cập nhật thông tin tài khoản	12
10	Sơ đồ lớp của hệ thống	13

11	Giao diện trang đăng nhập	15
12	Giao diện trang chủ - dữ liệu môi trường, và các tab điều hướng	15
13	Giao diện khi có cảnh báo đột nhập	15
14	Giao diện xem dữ liệu cảm biến theo thời gian	15
15	Giao diện trang điều khiển thiết bị thủ công	16
16	Giao diện cài đặt các ngưỡng cho điều khiển thiết bị tự động	16
17	Giao diện trang chỉnh sửa thông tin tài khoản	16
18	Giao diện đăng ký tài khoản mới (Chức năng cho Admin)	16

Danh sách bảng

1	Phân công nhiệm vụ trong nhóm	2
2	Các thiết bị và cổng kết nối trên Yolo:Bit	4
3	Use Case 1: Thiết lập ngưỡng tự động	9
4	Use Case 2: Điều khiển thủ công thiết bị	9
5	Use Case 3: Kích hoạt báo động	9
6	Use Case 4: Xem Dashboard dữ liệu cảm biến	10
7	Use Case 5: Quản lý tài khoản người dùng	10

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Hệ thống báo động an toàn cho gia đình

Hệ thống nhà thông minh (SmartHome) có lẽ là từ khóa quá quen thuộc với mọi người trong cuộc sống hiện đại ngày nay. Nhà thông minh (Smart Home) là một hệ thống tích hợp các thiết bị điện tử và công nghệ tiên tiến nhằm tạo ra một môi trường sống tiện nghi, an toàn và tiết kiệm năng lượng. Các thiết bị trong nhà thông minh có thể bao gồm đèn chiếu sáng, điều hòa nhiệt độ, hệ thống an ninh, và các thiết bị gia dụng khác, tất cả đều được kết nối và điều khiển thông qua mạng internet. Nhờ vào công nghệ Internet of Things (IoT), các thiết bị này có thể giao tiếp với nhau và với người dùng, cho phép điều khiển từ xa và tự động hóa các hoạt động hàng ngày. Hệ thống nhà thông minh không chỉ mang lại sự tiện lợi mà còn giúp tối ưu hóa việc sử dụng năng lượng, nâng cao mức độ an toàn và cải thiện chất lượng cuộc sống.

Trong đề án đa ngành này, nhóm chúng em sẽ thực hiện một hệ thống báo động an toàn cho gia đình (một mô hình SmartHome đơn giản) được thiết kế để giám sát môi trường trong nhà và cảnh báo khi phát hiện có đột nhập lạ. Hệ thống sử dụng vi điều khiển Yolo:Bit cùng với các cảm biến để theo dõi nhiệt độ, độ ẩm, ánh sáng và chuyển động. Khi phát hiện đột nhập, hệ thống sẽ kích hoạt cảnh báo bằng đèn LED RGB, tự động đóng cửa phòng, cảnh báo đến người dùng thông qua trang web trên điện thoại/máy tính bảng có kết nối Internet.

1.2. Yêu cầu của hệ thống

1.2.1. Yêu cầu chức năng

- Thu thập dữ liệu từ cảm biến nhiệt độ & độ ẩm, ánh sáng và khoảng cách; xử lý thông tin trong thời gian thực.
- Giám sát điều kiện môi trường (nhiệt độ, độ ẩm, ánh sáng) và tự động điều khiển các thiết bị quạt, đèn LED RGB, cửa trong phòng dựa trên ngưỡng cài đặt hoặc điều khiển thủ công.
- Phát hiện nguy hiểm khi có đột nhập và kích hoạt cảnh báo (đèn LED RGB chớp nháy, đóng cửa, thông báo trên web).
- Giao diện điều khiển trực quan trên web: hiển thị dữ liệu, điều khiển thiết bị từ xa, có dashboard để theo dõi trạng thái hệ thống.
- Lưu trữ lịch sử dữ liệu của hệ thống để truy xuất khi cần.
- Đồng bộ dữ liệu cảm biến và trạng thái thiết bị giữa nhiều người dùng đăng nhập cùng lúc.

1.2.2. Yêu cầu phi chức năng

- **Độ chính xác:** Cảm biến nhiệt độ sai số $\pm 2\%$, độ ẩm $\pm 5\%$.
- **Thời gian phản hồi:** Phản hồi dưới 1 giây sau khi nhận tín hiệu (real-time).

- **Độ bền:** Thiết bị hoạt động ổn định trong khoảng 0°C đến 50°C .
- **Tiêu thụ điện năng:** Nguồn 5V, tiêu thụ năng lượng thấp.
- **Tính khả dụng:** Hệ thống luôn sẵn sàng 24/7, không gián đoạn quá 60 phút.
- **Bảo mật:** Dữ liệu và mật khẩu được mã hóa; yêu cầu nhập mật khẩu để truy cập thông tin quan trọng.
- **Dễ sử dụng:** Giao diện người dùng đơn giản, đẹp, học sử dụng trong vòng 3 phút; tương thích với tất cả các trình duyệt web.
- **Tin cậy:** Dữ liệu thu được đảm bảo chính xác, ổn định và đáng tin cậy.
- **Hiệu suất:** Hệ thống xử lý nhanh, hiệu quả, đảm bảo không trễ dữ liệu.

1.2.3. Phạm vi và điều kiện hoạt động

- Phù hợp với hộ gia đình, căn hộ nhỏ, văn phòng hoặc lớp học quy mô nhỏ.
- Hỗ trợ các thiết bị cơ bản như quạt mini, đèn LED RGB, servo đóng/mở cửa.
- Kết nối và điều khiển từ xa qua mạng Internet trên nền tảng web.

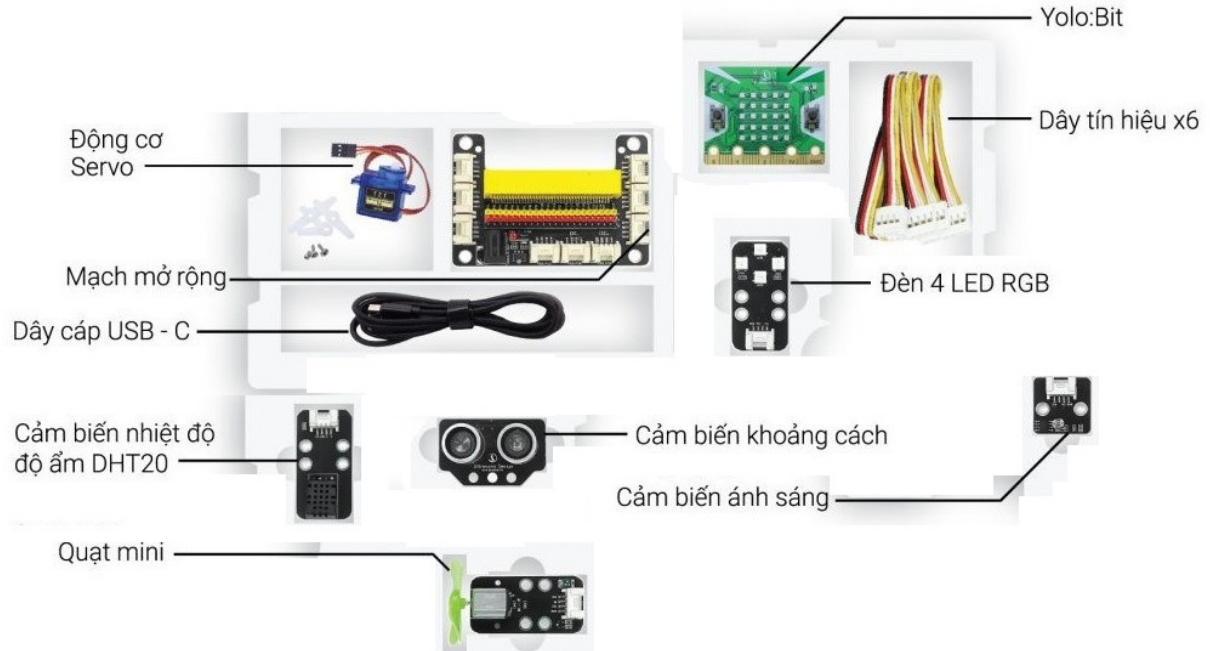
1.3. Phân công nhiệm vụ trong nhóm

STT	Sinh viên thực hiện	Mã số sinh viên	Nhiệm vụ
1	Nguyễn Anh Duy	2233163	Front-End cho Web App
2	Nguyễn Thành Đạt	2012938	Code cho mạch YoloBit
3	Nguyễn Ngọc Phú	2212588	Code IoT Gateway
4	Trương Minh Thông	2153005	Viết báo cáo L ^A T _E X
5	Nguyễn Viết Ký	2151217	Back-End cho Web App

Bảng 1: Phân công nhiệm vụ trong nhóm

CHƯƠNG 2. THIẾT KẾ HỆ THỐNG

2.1. Thiết bị sử dụng



Hình 1: Thiết bị sử dụng trong hệ thống

Vi điều khiển và linh kiện kết nối

- **Yolo:Bit:** Vi điều khiển trung tâm, thực hiện thu thập dữ liệu từ các cảm biến và điều khiển các thiết bị đầu ra.
- **Mạch mở rộng:** Giúp mở rộng cổng kết nối, cấp nguồn và giao tiếp giữa Yolo:Bit và các thiết bị ngoại vi.
- **Dây tín hiệu (x6):** Dây nối tín hiệu giữa Yolo:Bit và các cảm biến hoặc thiết bị điều khiển.
- **Dây cáp USB-C:** Dùng để cấp nguồn và lập trình cho Yolo:Bit từ máy tính.

Thiết bị đầu vào

- **Cảm biến nhiệt độ độ ẩm DHT20:** đo nhiệt độ và độ ẩm môi trường; tự động kích hoạt quạt mini khi nhiệt độ vượt ngưỡng.
- **Cảm biến khoảng cách:** phát hiện vật thể ở khoảng cách nhất định, dùng để kích hoạt cảnh báo (như đèn LED RGB).
- **Cảm biến ánh sáng:** đo cường độ ánh sáng môi trường; tự động bật/tắt đèn LED RGB tùy vào điều kiện sáng/tối.

Thiết bị đầu ra

- **Động cơ Servo:** điều khiển cơ cấu đóng/mở cửa hoặc chốt an toàn.
- **Đèn 4 LED RGB:** thể hiện trạng thái hệ thống (bình thường, cảnh báo, báo động).
- **Quạt mini:** kích hoạt khi nhiệt độ môi trường vượt ngưỡng để làm mát.

Thiết bị	Cổng kết nối
Cảm biến ánh sáng	P0
Cảm biến khoảng cách	P3 - P6
Cảm biến DHT20	I2C 1
Quạt mini	P10
Đèn LED RGB	P14
Động cơ Servo	P4

Bảng 2: Các thiết bị và cổng kết nối trên Yolo:Bit

2.2. Mô tả hoạt động hệ thống

1. Luồng dữ liệu cảm biến

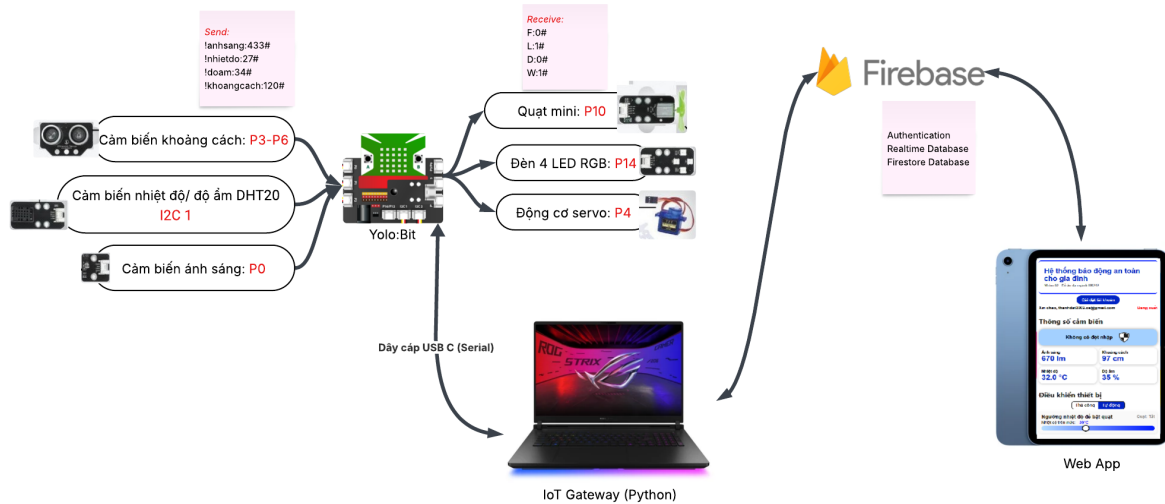
- **Cảm biến khoảng cách** kết nối qua các chân P3 đến P6 trên Yolo:Bit, dùng để phát hiện vật thể chuyển động hoặc người xâm nhập.
- **Cảm biến nhiệt độ/độ ẩm DHT20** sử dụng giao tiếp I2C (I2C 1) để đo các thông số môi trường.
- **Cảm biến ánh sáng** kết nối qua chân P0 để theo dõi cường độ ánh sáng môi trường.
- Yolo:Bit đọc dữ liệu từ các cảm biến, xử lý và gửi thông tin qua cổng USB-C tới máy tính qua giao tiếp Serial.
- Trên máy tính, một script Python hoạt động như một **IoT Gateway**, tiếp nhận dữ liệu từ Yolo:Bit, sau đó gửi lên Firebase (Realtime Database hoặc Firestore).

2. Luồng điều khiển thiết bị

- Người dùng điều khiển hệ thống thông qua giao diện Web App, gửi lệnh (bật quạt, bật đèn, mở servo...) thông qua Firebase.
- Máy tính trung gian (IoT Gateway) đọc các dữ liệu từ Firebase, sau đó chuyển lệnh điều khiển đến Yolo:Bit thông qua cổng Serial.
- Yolo:Bit nhận lệnh và kích hoạt các thiết bị đầu ra tương ứng:
 - **Quạt mini (P10):** Bật khi nhiệt độ cao hoặc do người dùng điều khiển.
 - **Đèn LED RGB (P14):** Hiển thị trạng thái cảnh báo, báo động.
 - **Động cơ Servo (P4):** Điều khiển đóng/mở cửa hoặc khóa an toàn.

Sơ đồ khối kiến trúc hệ thống

Hệ thống báo động an toàn gia đình sử dụng vi điều khiển **Yolo:Bit** để thu thập dữ liệu từ các cảm biến và điều khiển các thiết bị ngoại vi. Dữ liệu được xử lý cục bộ và đồng bộ hóa với giao diện web thông qua một máy tính trung gian (IoT Gateway) và dịch vụ **Firestore**.



Hình 2: Sơ đồ khối kiến trúc hệ thống

- **Vi điều khiển trung tâm: Yolo:Bit**
 - Đọc dữ liệu từ cảm biến
 - Nhận/gửi dữ liệu qua Serial USB-C
 - Kích hoạt thiết bị điều khiển theo lệnh từ Web
- **Máy tính trung gian (IoT Gateway):**
 - Nhận dữ liệu từ Yolo:Bit (Serial)
 - Upload dữ liệu lên Firebase (Realtime DB hoặc Firestore)
 - Nhận lệnh từ Firebase và chuyển về cho Yolo:Bit
- **Firestore:**
 - Authentication: xác thực người dùng

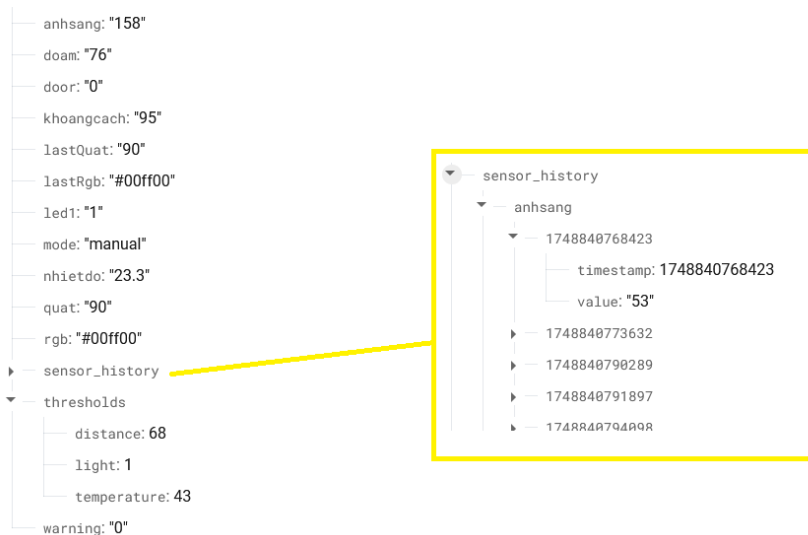
(default)	users	VVcLtQKigCSOZkYgSgZkCBjgHgQ2
+ Start collection	+ Add document	+ Start collection
users >	VVcLtQKigCSOZkYgSgZkCBjgHgQ2 >	+ Add field
	e9irJQ0R5YN10605...	<p>createdAt: May 31, 2025 at 10:57:54 PM UTC+7</p> <p>email: "user2DADN@gmail.com"</p> <p>name: "Người dùng 2"</p> <p>phone: "000223334"</p> <p>role: "admin"</p>

Identifier	Providers	Created ↓	Signed in	User UID
admin1234@hcmut.ed...	📧	May 31, 2025	Jun 5, 2025	e9irJQ0R5YN10605JnbPOY5...
user2dadn@gmail.com	📧	May 31, 2025	May 31, 2025	VVcLtQKigCSOZkYgSgZkCBjgHgQ2...

Rows per page: 50 1 - 2 of 2

Hình 3: Cơ sở dữ liệu tài khoản người dùng trên Firebase

– Realtime Database / Firestore: lưu trữ dữ liệu cảm biến và điều khiển



Hình 4: Lưu trữ dữ liệu và cài đặt hệ thống trên Firebase

• Web App:

- Hiển thị thông tin cảm biến theo thời gian thực
- Gửi lệnh điều khiển thiết bị
- Cho phép người dùng đăng nhập và quản lý thiết bị, quản lý tài khoản

2.3. Công nghệ sử dụng

- **Firebase:** Xác thực người dùng và lưu trữ dữ liệu tài khoản, dữ liệu cài đặt và lịch sử dữ liệu môi trường.
- **Python (IoT Gateway):** Quản lý giao tiếp với Yolo:bit và Firebase.
- **Công nghệ Web (HTML, CSS, JavaScript):** Tạo giao diện người dùng để giám sát và điều khiển từ xa.

- **GitHub, GitHub Pages:** Lưu trữ mã nguồn và triển khai trang web.

Các công cụ khác:

- **Figma, Photoshop:** Thiết kế giao diện cho Web App.
- **Lucidchart:** Vẽ các sơ đồ trong báo cáo.
- **Overleaf:** Soạn thảo và trình bày báo cáo.

2.4. Sử dụng Design Pattern

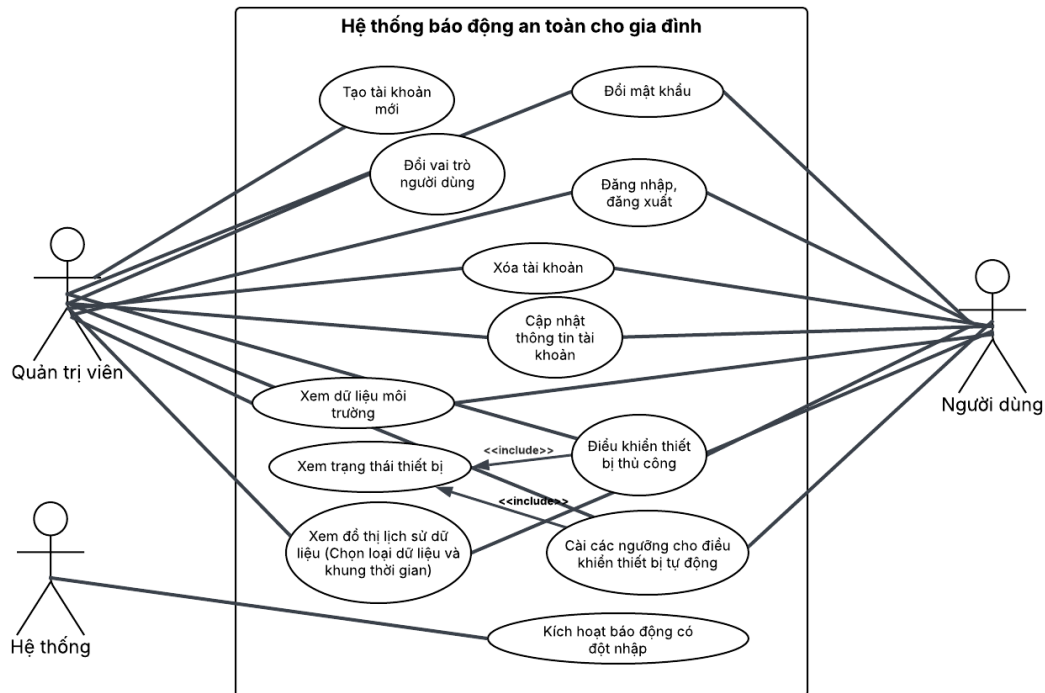
Trong quá trình xây dựng trang web điều khiển cho Hệ thống báo động an toàn, nhóm đã áp dụng một số mẫu thiết kế (*Design Patterns*) để cấu trúc của hệ thống rõ ràng, dễ bảo trì hơn:

- **Observer Pattern (Mẫu Quan sát viên):** Ứng dụng trong việc cập nhật dữ liệu cảm biến và phản hồi hành động. Khi một cảm biến thay đổi giá trị (ví dụ: nhiệt độ tăng vượt ngưỡng), hệ thống sẽ tự động thông báo đến các thành phần liên quan (bật quạt). Giúp giảm sự phụ thuộc trực tiếp giữa các thành phần và tăng tính linh hoạt của hệ thống.
- **MVC Pattern (Model - View - Controller):** Cấu trúc hệ thống web app và ứng dụng di động được thiết kế theo mô hình MVC. **Model** đại diện cho dữ liệu (cảm biến, thiết bị điều khiển), **View** là giao diện người dùng (giao diện web), và **Controller** xử lý logic trung gian như gửi lệnh từ người dùng đến thiết bị hoặc hiển thị dữ liệu cảm biến lên màn hình. Việc tách biệt ba thành phần này giúp dễ dàng thay đổi hoặc nâng cấp một phần mà không ảnh hưởng đến toàn bộ hệ thống.
- **Mediator Pattern (Mẫu Trung gian):** Được sử dụng trong giao tiếp giữa các thiết bị thông qua máy tính trung gian (chạy Python script). Thay vì các cảm biến và thiết bị đầu ra giao tiếp trực tiếp với nhau, chúng đều kết nối và gửi nhận qua **Python script**, đóng vai trò trung gian quản lý luồng dữ liệu và lệnh điều khiển giữa các thành phần (Yolo:bit, Firebase, Web App). Giúp giảm độ phức tạp và tăng khả năng kiểm soát trung tâm.
- **Singleton Pattern:** Dùng Firebase như là **singleton service** để quản lý người dùng, quyền truy cập và cấu hình thiết bị. Trong toàn hệ thống, Firebase đóng vai trò độc nhất cho việc xác thực và lưu trữ cài đặt hệ thống, đảm bảo không có xung đột xảy ra giữa các phiên truy cập.

Dùng các Design Pattern giúp hệ thống có cấu trúc rõ ràng, dễ kiểm soát dòng dữ liệu và tăng cường khả năng mở rộng trong tương lai (tích hợp thêm các công nghệ AI, học máy hoặc hệ thống điều khiển phức tạp hơn).

CHƯƠNG 3. USE CASE CHI TIẾT VÀ SƠ ĐỒ HOẠT ĐỘNG, SƠ ĐỒ LỚP CỦA HỆ THỐNG

3.1. Sơ đồ Use Case của hệ thống



Hình 5: Sơ đồ Use Case của hệ thống

3.2. Đặc tả Use Case của hệ thống

Use Case	Thiết lập ngưỡng cho điều khiển tự động
Actors	Người dùng, admin
Descriptions	Cho phép người dùng thiết lập ngưỡng (slider) cho các cảm biến như nhiệt độ, ánh sáng, khoảng cách để hệ thống điều khiển thiết bị tự động.
Precondition	- Người dùng đã đăng nhập thành công. - Cảm biến đang hoạt động bình thường.
Normal flow	1. Người dùng mở giao diện cài đặt tự động. 2. Hệ thống hiển thị các slider ngưỡng cho từng cảm biến. 3. Người dùng kéo slider để điều chỉnh ngưỡng phù hợp. 4. Hệ thống lưu các ngưỡng vào cơ sở dữ liệu. 5. Thông báo thiết lập thành công.
Alternative flow	Alternative 1: - Nếu người dùng nhập giá trị vượt mức cho phép, hệ thống hiển thị cảnh báo và không cho lưu.

Exceptions	Exception 1: - Lỗi kết nối cơ sở dữ liệu khiến không thể lưu thiết lập.
-------------------	---

Bảng 3: Use Case 1: Thiết lập ngưỡng tự động

Use Case	Điều khiển thủ công thiết bị
Actors	Người dùng, admin
Descriptions	Người dùng có thể điều khiển bật/tắt thiết bị như quạt, đèn, khóa cửa thủ công qua giao diện.
Precondition	- Người dùng đã đăng nhập. - Thiết bị được thêm vào hệ thống và đang hoạt động.
Normal flow	1. Người dùng chọn phòng và thiết bị muốn điều khiển. 2. Giao diện hiển thị trạng thái hiện tại của thiết bị. 3. Người dùng chọn hành động (bật/tắt). 4. Hệ thống gửi lệnh điều khiển đến phần cứng. 5. Giao diện cập nhật trạng thái mới.
Alternative flow	Alternative 1: - Nếu thiết bị không phản hồi, hệ thống hiển thị thông báo lỗi.
Exceptions	Exception 1: - Lỗi giao tiếp phần cứng hoặc thiết bị ngoại tuyến.

Bảng 4: Use Case 2: Điều khiển thủ công thiết bị

Use Case	Kích hoạt báo động
Actors	Hệ thống
Descriptions	Khi giá trị cảm biến vượt ngưỡng cho phép (khoảng cách gần, nhiệt độ cao), hệ thống tự động bật báo động trên web và phần cứng.
Precondition	- Thiết lập ngưỡng đã được cấu hình. - Thiết bị và cảm biến đang hoạt động.
Normal flow	1. Cảm biến gửi dữ liệu mới. 2. Hệ thống kiểm tra ngưỡng. 3. Nếu vượt ngưỡng, hệ thống: - Phát âm thanh cảnh báo trên web. - Chớp đèn LED cảnh báo phần cứng. - Khóa cửa (nếu là cảnh báo xâm nhập). 4. Giao diện web hiển thị chỉ số màu đỏ và in đậm.
Alternative flow	Alternative 1: - Nếu mức chỉ vượt nhẹ, chỉ hiển thị cảnh báo nhẹ, không kích hoạt toàn bộ hệ thống.
Exceptions	Exception 1: - Nếu không thể điều khiển thiết bị vật lý, hệ thống chỉ cảnh báo trên web.

Bảng 5: Use Case 3: Kích hoạt báo động

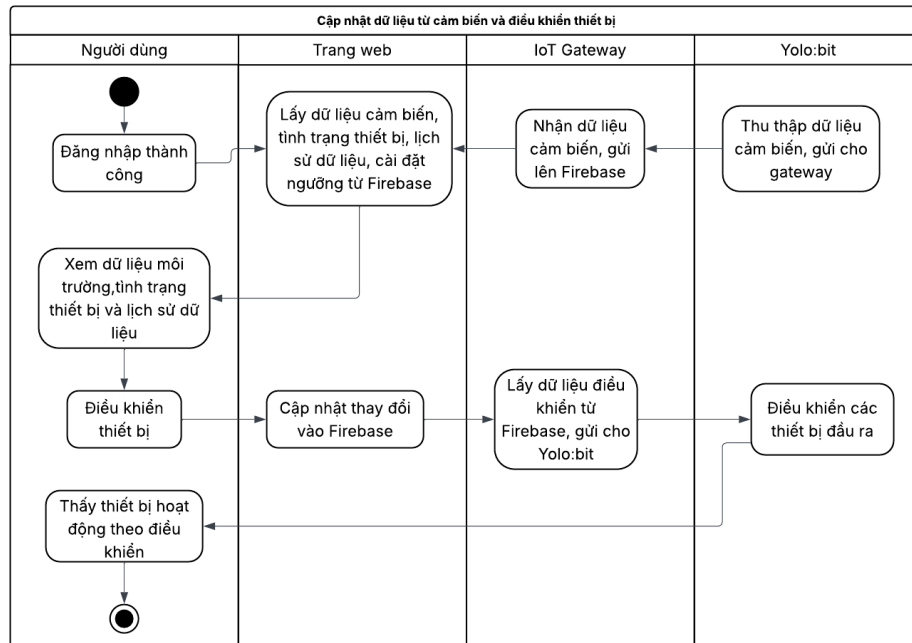
Use Case	Xem Dashboard dữ liệu cảm biến
Actors	Người dùng
Descriptions	Hệ thống hiển thị biểu đồ dữ liệu cảm biến theo thời gian thực hoặc theo khoảng thời gian chọn lọc.
Precondition	- Dữ liệu cảm biến đang được ghi nhận và lưu. - Người dùng đã đăng nhập.
Normal flow	1. Người dùng vào giao diện dashboard. 2. Chọn loại dữ liệu muốn xem (nhiệt độ, độ ẩm...). 3. Chọn thời gian: 1 phút, 5 phút, hoặc tùy chỉnh. 4. Biểu đồ cập nhật theo thời gian thực (cập nhật mỗi 2 giây). 5. Hiển thị thông tin rõ ràng, các ngưỡng vượt mức được in đậm.
Alternative flow	Alternative 1: - Nếu không có dữ liệu trong khoảng thời gian chọn, hiển thị “Không có dữ liệu”.
Exceptions	Exception 1: - Lỗi truy vấn dữ liệu từ cơ sở dữ liệu.

Bảng 6: Use Case 4: Xem Dashboard dữ liệu cảm biến

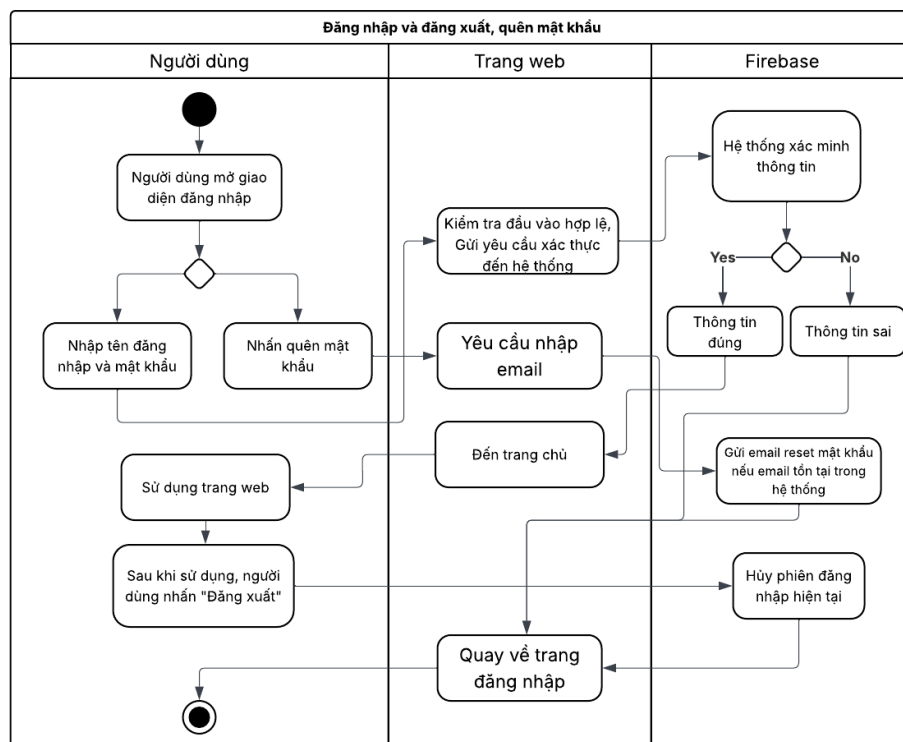
Use Case	Quản lý tài khoản người dùng
Actors	Admin
Descriptions	Admin có thể chỉnh sửa, xóa tài khoản người dùng, hoặc thêm mới tài khoản nếu cần.
Precondition	- Admin đã đăng nhập hệ thống.
Normal flow	1. Admin truy cập giao diện quản lý tài khoản. 2. Xem danh sách tài khoản hiện tại. 3. Chọn thao tác: sửa, xóa, hoặc thêm mới. 4. Thực hiện thao tác và lưu thay đổi. 5. Hệ thống hiển thị thông báo thành công.
Alternative flow	Alternative 1: - Nếu người dùng không có quyền admin, giao diện không hiển thị tính năng này.
Exceptions	Exception 1: - Lỗi kết nối cơ sở dữ liệu gây thất bại thao tác.

Bảng 7: Use Case 5: Quản lý tài khoản người dùng

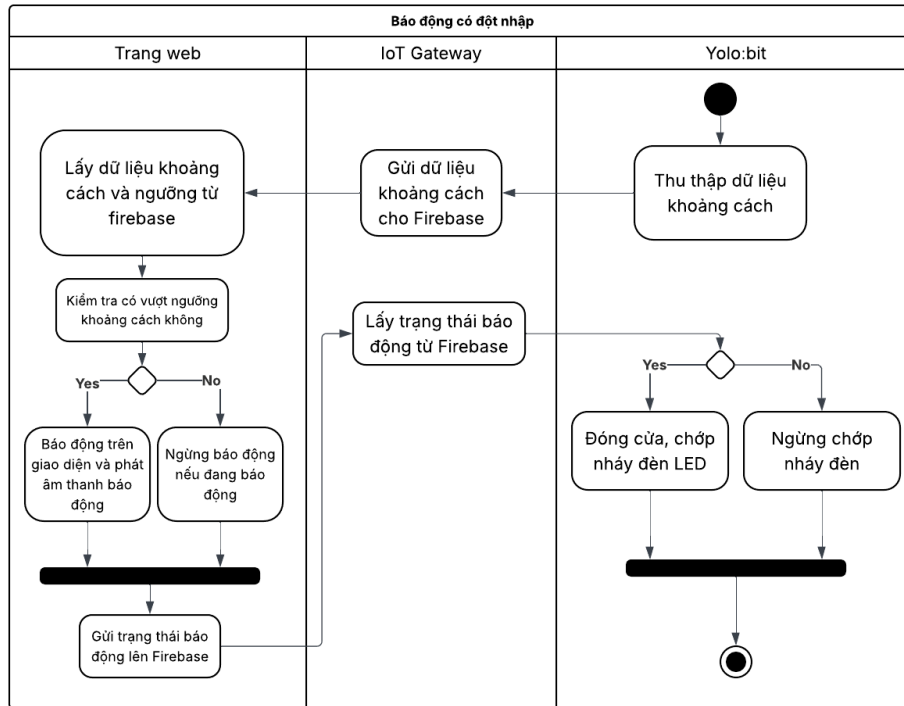
3.3. Sơ đồ hoạt động của hệ thống



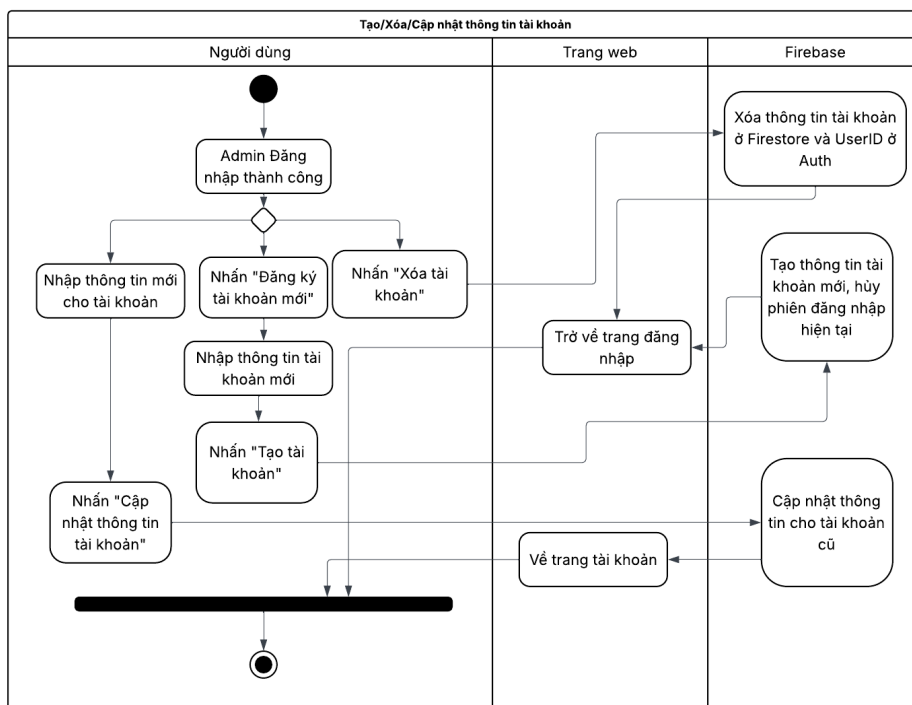
Hình 6: Sơ đồ hoạt động cho cập nhật dữ liệu từ cảm biến và điều khiển thiết bị



Hình 7: Sơ đồ hoạt động cho đăng nhập và đăng xuất, quên mật khẩu



Hình 8: Sơ đồ hoạt động cho báo động có đột nhập



Hình 9: Sơ đồ hoạt động cho tạo-xóa-cập nhật thông tin tài khoản

3.4. Sơ đồ lớp của hệ thống

Các lớp quan trọng bao gồm: **Người dùng, Cảm biến, Ngưỡng, Dữ liệu cảm biến, Cảnh báo.** Trong đó, **Người dùng** là đối tượng trung tâm, có thể quản lý các cảm biến,

thực hiện thay đổi ngưỡng, theo dõi dữ liệu.

Mỗi **Cảm biến** gắn với một giá trị **Ngưỡng**, được sử dụng để xác định khi nào dữ liệu vượt giới hạn an toàn. Khi cảm biến hoạt động sẽ liên tục tạo ra **Dữ liệu cảm biến**, bao gồm giá trị của cảm biến đó và thời gian ghi nhận. Nếu dữ liệu gần đây nhất vượt ngưỡng, hệ thống sẽ tạo một **Cảnh báo** và gửi đến người dùng để xử lý kịp thời. Mỗi cảm biến có kiểu, vị trí và trạng thái. Phương thức `readData()` trả về dữ liệu cảm biến, trong khi `checkThreshold()` kiểm tra ngưỡng và `triggerAlert()` tạo cảnh báo nếu vượt quá ngưỡng.

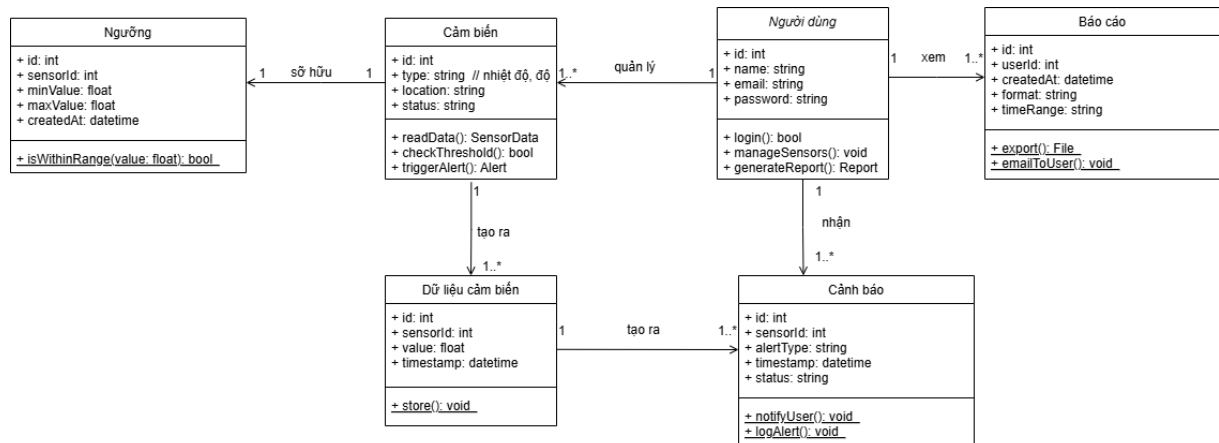
Lớp **Người dùng** đại diện cho cá nhân quản lý hệ thống, có các thuộc tính như `name`, `email` và `password`.

Lớp **Ngưỡng** định nghĩa giới hạn cho từng cảm biến. Bao gồm giá trị tối thiểu và tối đa, và phương thức `isWithinRange()` để kiểm tra một giá trị có nằm trong khoảng cho phép hay không.

Lớp **Dữ liệu cảm biến** lưu lại thông tin được thu thập, bao gồm giá trị, thời gian và trạng thái. Mỗi bản ghi được tạo ra bởi cảm biến và được lưu trữ bằng phương thức `store()`.

Lớp **Cảnh báo** được tạo ra khi dữ liệu cảm biến vượt quá ngưỡng. Nó ghi nhận loại cảnh báo, thời gian và trạng thái, đồng thời có thể gửi thông báo cho người dùng (`notifyUser()`) và ghi nhật ký (`logAlert()`).

Mối quan hệ giữa các lớp chủ yếu là quan hệ một-nhiều (*one-to-many*), chẳng hạn như một người dùng quản lý nhiều cảm biến, một cảm biến có thể tạo ra nhiều dữ liệu hoặc cảnh báo.



Hình 10: Sơ đồ lớp của hệ thống

CHƯƠNG 4. GIAO DIỆN NGƯỜI DÙNG VÀ DEMO SẢN PHẨM

4.1. Demo sản phẩm

Link video Demo sản phẩm trên Google Drive: https://drive.google.com/file/d/1Nr93MCFbsciDFKg_xInyUwdJTMvygbvV/view?usp=sharing

Mã nguồn báo cáo L^AT_EX, Mã nguồn Web App, Mã nguồn Yolo:bit và IoT Gateway: https://github.com/yymt242/DADN242_Group82

4.2. Chạy hệ thống

Bước 1. Truy cập giao diện điều khiển web

Mở trình duyệt và truy cập trang web điều khiển được host tại địa chỉ:

https://yymt242.github.io/DADN242_Group82/

Đăng nhập bằng tài khoản quản trị viên:

1	Email:	admin1234@hcmut.edu.vn
2	Mật khẩu:	admin1234

Bước 2. Nạp chương trình cho thiết bị Yolo:bit

- Sử dụng trình duyệt Google Chrome.
- Truy cập trang lập trình Yolo:bit: <https://app.ohstem.vn/>
- Tiến hành:
 - (a) Cập nhật firmware cho thiết bị.
 - (b) Cài đặt thư viện mở rộng cần thiết.
 - (c) Reset thiết bị bằng cách: Nhấn giữ nút A trong 3 giây, sau đó nhấn nút Reset trong 1 giây (vẫn giữ nút A), rồi thả nút A.
 - (d) Nạp chương trình đã được chuẩn bị.

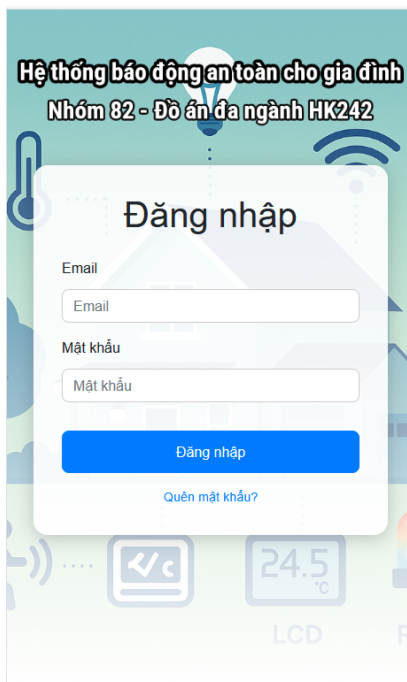
Bước 3. Khởi động chương trình IoT Gateway trên máy tính (Windows)

Mở terminal (CMD hoặc PowerShell), chạy lần lượt các lệnh sau để cài đặt và chạy:

1	py -m pip install --upgrade pip
2	pip install requests pyserial
3	py .\gateway.py

Để dừng chương trình, nhấn tổ hợp phím Ctrl + C.

4.3. Giao diện người dùng



Hình 11: Giao diện trang đăng nhập



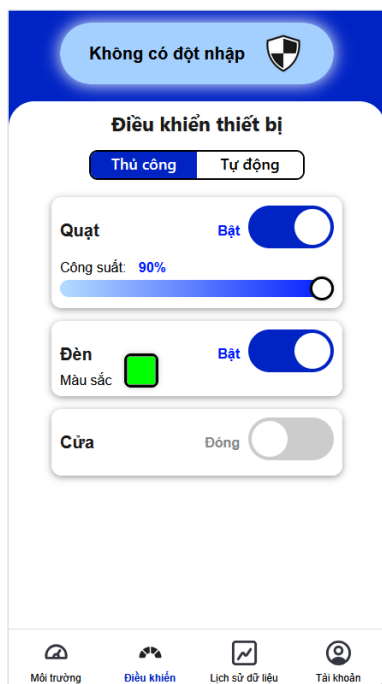
Hình 12: Giao diện trang chủ - dữ liệu môi trường, và các tab điều hướng



Hình 13: Giao diện khi có cảnh báo đột nhập



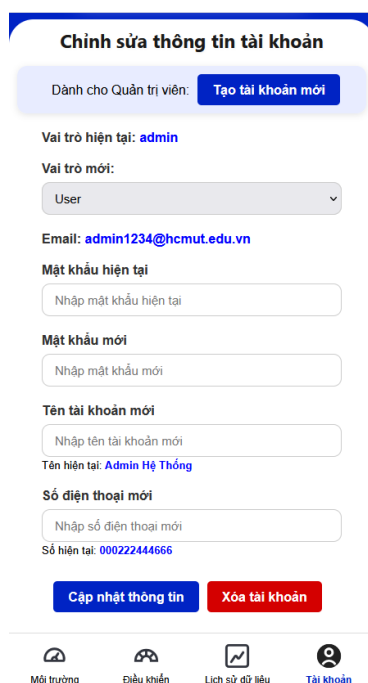
Hình 14: Giao diện xem dữ liệu cảm biến theo thời gian



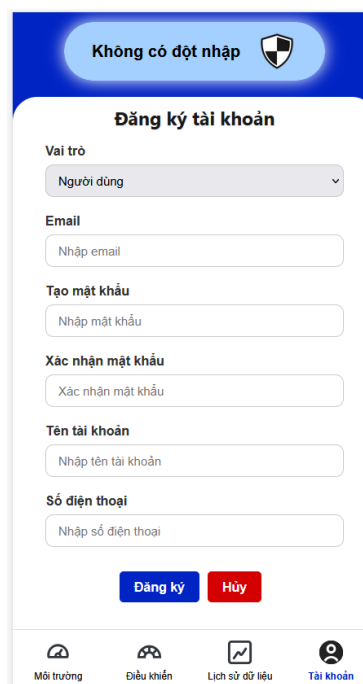
Hình 15: Giao diện trang điều khiển thiết bị thủ công



Hình 16: Giao diện cài đặt các ngưỡng cho điều khiển thiết bị tự động



Hình 17: Giao diện trang chỉnh sửa thông tin tài khoản



Hình 18: Giao diện đăng ký tài khoản mới (Chức năng cho Admin)

CHƯƠNG 5. KẾT LUẬN

Trong đề án này, nhóm đã xây dựng thành công một hệ thống cảnh báo an toàn cho gia đình với đầy đủ các tính năng quan trọng:

- Theo dõi các chỉ số môi trường như nhiệt độ, độ ẩm, ánh sáng và khoảng cách; đồng thời tự động đưa ra các phản hồi phù hợp như bật quạt, bật đèn LED RGB, đóng cửa.
- Điều khiển thiết bị tự động theo ngưỡng giá trị cảm biến với giao diện tùy chỉnh mức hoạt động thông qua thanh trượt.
- Cảnh báo nguy hiểm: chớp đèn, âm thanh cảnh báo trên web, khoá cửa phần cứng trong trường hợp có đột nhập.
- Giao diện dashboard hiển thị dữ liệu thời gian thực với biểu đồ trực quan, cập nhật nhanh, dễ theo dõi.
- Hệ thống tài khoản có phân quyền: chỉ quản trị viên mới được tạo tài khoản mới; người dùng có thể đăng nhập, chỉnh sửa và xóa tài khoản.
- Đăng nhập/đăng ký mượt mà với xử lý lỗi đầy đủ và tương tác người dùng tối ưu (nhấn Enter sau khi nhập thông tin để đăng nhập, hiển thị hộp thoại thay vì alert thô).

Trong tương lai, hệ thống của nhóm có thể được mở rộng và cải tiến theo các hướng sau:

- **Áp dụng trí tuệ nhân tạo (AI):** để học ngưỡng hoạt động tối ưu từ dữ liệu lịch sử, giảm cảnh báo giả.
- **Cải tiến giao diện và trải nghiệm người dùng:** bổ sung chế độ tối/sáng, hỗ trợ đa ngôn ngữ.
- **Tích hợp nhiều kênh cảnh báo khác:** ngoài cảnh báo qua web, thì nên tích hợp thêm các kênh cảnh báo khác như email, Zalo, Telegram hoặc ứng dụng di động để tăng khả năng phản hồi trong thực tế.
- **Mở rộng kết nối thiết bị:** để hỗ trợ thêm các phần cứng mới như camera AI, robot tuần tra hoặc thiết bị khóa thông minh.

CHƯƠNG 6. TÀI LIỆU THAM KHẢO

- [1] Firebase Documentation. *Cloud Firestore*. Retrieved from <https://firebase.google.com/docs/firestore>
- [2] OhStem Education. *YOLO:HOME – AI và IoT cho Nhà Thông Minh*.
- [3] Lê Trọng Nhân, Nguyễn Trần Hữu Nguyên, Võ Tấn Phương. *Hướng dẫn lập trình Python cơ bản*.
- [4] Nguyễn Thiên Ân, Phạm Thanh Danh, Huỳnh Nhữ Hùng, Trần Nhật Quang. *Phát triển Gateway IoT bằng Python*.
- [5] GitHub Docs. *Creating a GitHub Pages site*. Retrieved from <https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site>
- [6] Chart.js Documentation. *Line Chart Example*. Retrieved from <https://www.chartjs.org/docs/latest/samples/line/line.html>
- [7] SweetAlert2. *Usage*. Retrieved from <https://sweetalert2.github.io/#usage>