

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

---



**BÁO CÁO GIỮA KỲ MÔN IOT**  
**HỆ THỐNG ĐIỆN THÔNG MINH**

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

**Thành viên thực hiện:** Nguyễn Việt Quang - B22DCCN650  
Phạm Công Minh - B22DCCN542  
Phạm Hải Dương - B22DCCN169  
Phùng Trung Kiên - B22DCCN433

***HÀ NỘI, 2025***

## MỤC LỤC

<b><i>PHÂN TÍCH YÊU CẦU HỆ THỐNG ĐÈN THÔNG MINH.....</i></b>	<b><i>1</i></b>
<b><i>(DÙNG ESP32).....</i></b>	<b><i>1</i></b>
1. Mục tiêu & phạm vi hệ thống.....	1
2. Thu thập yêu cầu từ các bên liên quan.....	2
3. Yêu cầu chức năng (tính năng dự kiến triển khai).....	2
4. Yêu cầu phi chức năng.....	3
5. Phân tích ràng buộc kỹ thuật và môi trường.....	3
6. Mô hình yêu cầu.....	3
<b><i>Phân chia công việc &amp; Kế hoạch triển khai.....</i></b>	<b><i>12</i></b>

# PHÂN TÍCH YÊU CẦU HỆ THỐNG ĐÈN THÔNG MINH (DÙNG ESP32)

## 1. Mục tiêu & phạm vi hệ thống

*Vấn đề thực tế:* Người dùng thường quên tắt đèn khi ra khỏi phòng, hoặc bật đèn không phù hợp với mức ánh sáng môi trường, gây lãng phí điện năng và giảm tuổi thọ thiết bị.

*Mục tiêu hệ thống IoT:*

- Tự động bật/tắt đèn dựa trên cảm biến chuyển động.
- Điều chỉnh độ sáng thủ công phù hợp với điều kiện môi trường và lịch trình bật tắt.
- Cho phép người dùng điều khiển, giám sát đèn qua ứng dụng di động hoặc web.
- Tiết kiệm điện năng, nâng cao tiện nghi sinh hoạt.

*Phạm vi triển khai:*

- Môi trường: nhà ở, văn phòng, lớp học.
- Quy mô: 10–20 đèn thông minh, mỗi đèn gắn 1 cảm biến (ánh sáng/chuyển động).
- Bộ xử lý trung tâm: ESP32 kết nối Wi-Fi, giao tiếp trực tiếp với server/cloud.
- Hệ thống thử nghiệm trong 1 căn hộ hoặc phòng học mẫu.

*Tiêu chí thành công (KPIs):*

Tiêu chí	Mục tiêu
Tiết kiệm điện năng	$\geq 25\%$ so với hệ thống truyền thống
Độ trễ điều khiển	$< 2$ giây
Tỷ lệ truyền dữ liệu thành công	$\geq 98\%$
Độ ổn định hoạt động	$\geq 99\%$ uptime/tháng
Chi phí	$\leq 500k$ 1 thiết bị

*Kết quả mong đợi:*

- Tiết kiệm điện
- Người dùng có thể điều khiển bật tắt đèn hoặc điều chỉnh độ sáng tối của đèn từ xa

- Người dùng có thể nhìn thấy xem đèn nào đang bật, đang tắt => tránh đi ra ngoài quên tắt đèn
- Người dùng có hẹn lịch bật tắt đèn.
- Đèn có thể cảm biến khi có chuyển động qua lại.

## 2. Thu thập yêu cầu từ các bên liên quan

Nhóm liên quan	Yêu cầu chính
Người dùng cuối (chủ nhà, giáo viên, nhân viên)	<ul style="list-style-type: none"> <li>- Bật/tắt đèn thủ công hoặc tự động.</li> <li>- Điều chỉnh độ sáng theo nhu cầu.</li> <li>- Giám sát trạng thái đèn qua điện thoại.</li> <li>- Nhận cảnh báo khi đèn không hoạt động hoặc mất kết nối.</li> </ul>
Doanh nghiệp / Nhà phát triển	<ul style="list-style-type: none"> <li>- Hệ thống dễ mở rộng thêm đèn.</li> <li>- Giao diện thân thiện, dễ bảo trì.</li> <li>- Thu thập dữ liệu để tối ưu hóa thuật toán tự động.</li> </ul>
Kỹ thuật / IT	<ul style="list-style-type: none"> <li>- Kết nối qua Wifi chuẩn 2.4 GHz.</li> <li>- Giao thức MQTT hoặc HTTP để trao đổi dữ liệu.</li> <li>- Mã hóa dữ liệu</li> <li>- Hỗ trợ cập nhật firmware từ xa (OTA).</li> </ul>

Phương pháp thu thập: Phỏng vấn người dùng, khảo sát hành vi sử dụng điện, quan sát thực tế trong môi trường học/làm việc.

## 3. Yêu cầu chức năng (tính năng dự kiến triển khai)

1. Thu thập dữ liệu cảm biến:
  - ESP32 đọc dữ liệu từ cảm biến chuyển động.
2. Xử lý và ra quyết định tự động:
  - Bật đèn khi có người.
  - Tắt đèn khi không có người trong 2 phút.
3. Điều khiển từ xa:

- Người dùng bật/tắt hoặc đặt lịch qua app (qua MQTT/HTTP).
4. Cảnh báo và hiển thị:
- Báo lỗi đèn, hiển thị trạng thái hoạt động (On/Off).
5. Cập nhật từ xa:
- OTA firmware update qua Wi-Fi.

#### 4. Yêu cầu phi chức năng

Nhóm yêu cầu	Mô tả
Hiệu năng	Độ trễ điều khiển $\leq 2$ giây; cập nhật dữ liệu mỗi 5 giây.
Bảo mật	Xác thực thiết bị bằng token; phân quyền người dùng (admin/user).
Độ tin cậy	ESP32 tự phục hồi sau khi mất kết nối; lưu trạng thái đèn cục bộ khi không có Internet.
Khả năng mở rộng	Có thể thêm tới 50 thiết bị ESP32 trong cùng mạng Wi-Fi.
Chi phí & năng lượng	Tránh lãng phí điện khi người dùng quên tắt đèn và tăng tính tiện nghi trong sinh hoạt.

#### 5. Phân tích ràng buộc kỹ thuật và môi trường

- Môi trường hoạt động: trong nhà ( $10-40^{\circ}\text{C}$ , độ ẩm  $\leq 85\%$ ).
- Kết nối: Wi-Fi 2.4GHz, hỗ trợ MQTT/HTTP với cloud.
- Nguồn cấp: điện lưới 220V AC (đèn), ESP32 dùng bộ nguồn 5V DC.
- Giới hạn phần cứng: RAM 520KB, Flash 4MB  $\rightarrow$  xử lý đơn giản, gửi dữ liệu lên cloud để phân tích.

#### 6. Mô hình yêu cầu

##### *Các actor:*

Người dùng (user)

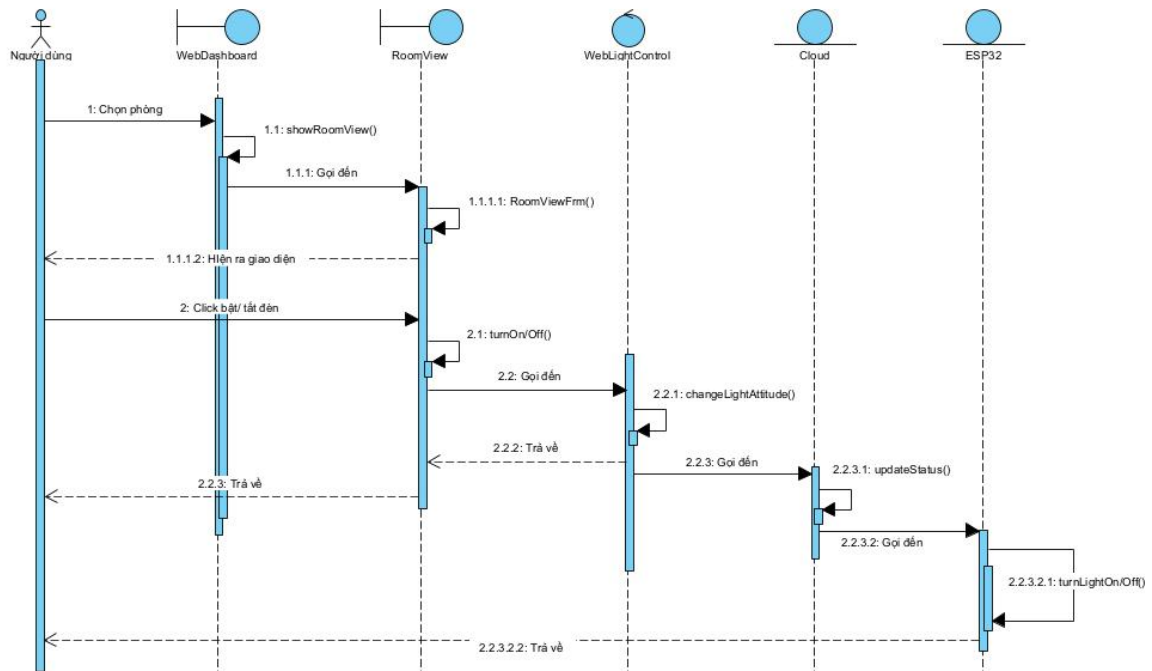
Cloud/ server

Esp32/ thiết bị

##### *Use Case chính + Sequence Diagram tương ứng:*

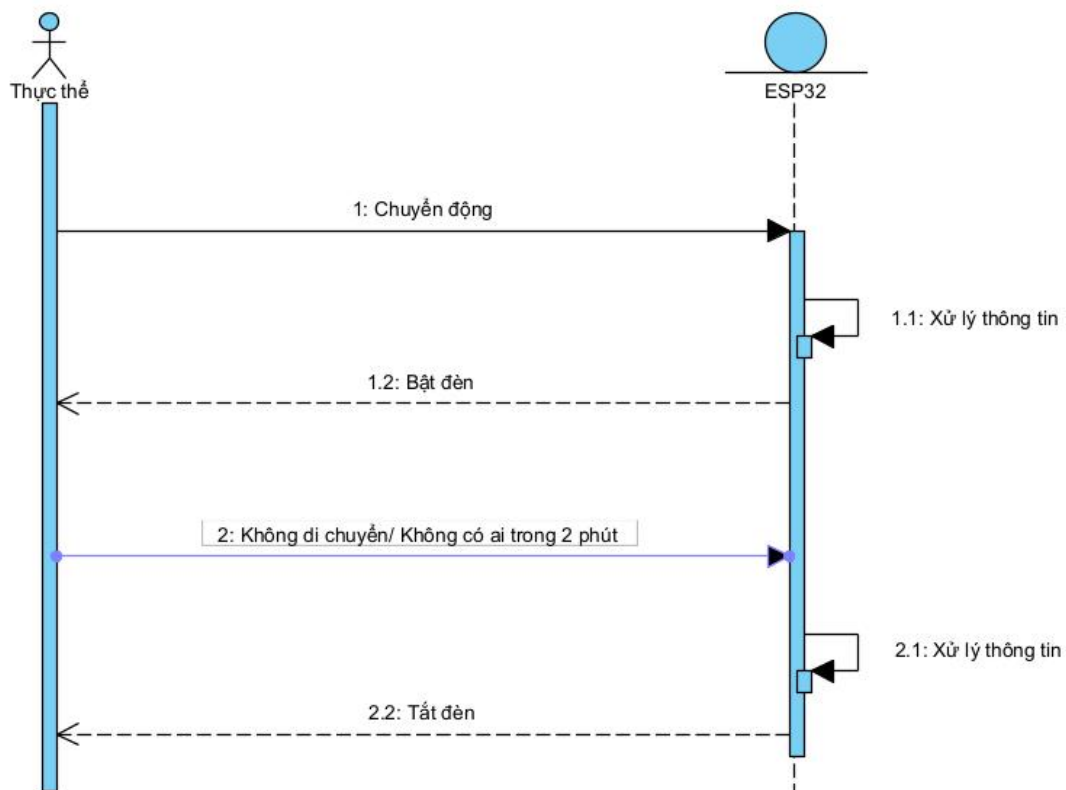
UC1 – Bật/Tắt đèn thủ công qua Internet

- Người dùng: gửi lệnh bật/tắt đèn từ app/web.
- Cloud: nhận lệnh của người dùng và gửi xuống đúng ESP32 tương ứng.
- ESP32: nhận lệnh và điều khiển relay bật/tắt đèn.



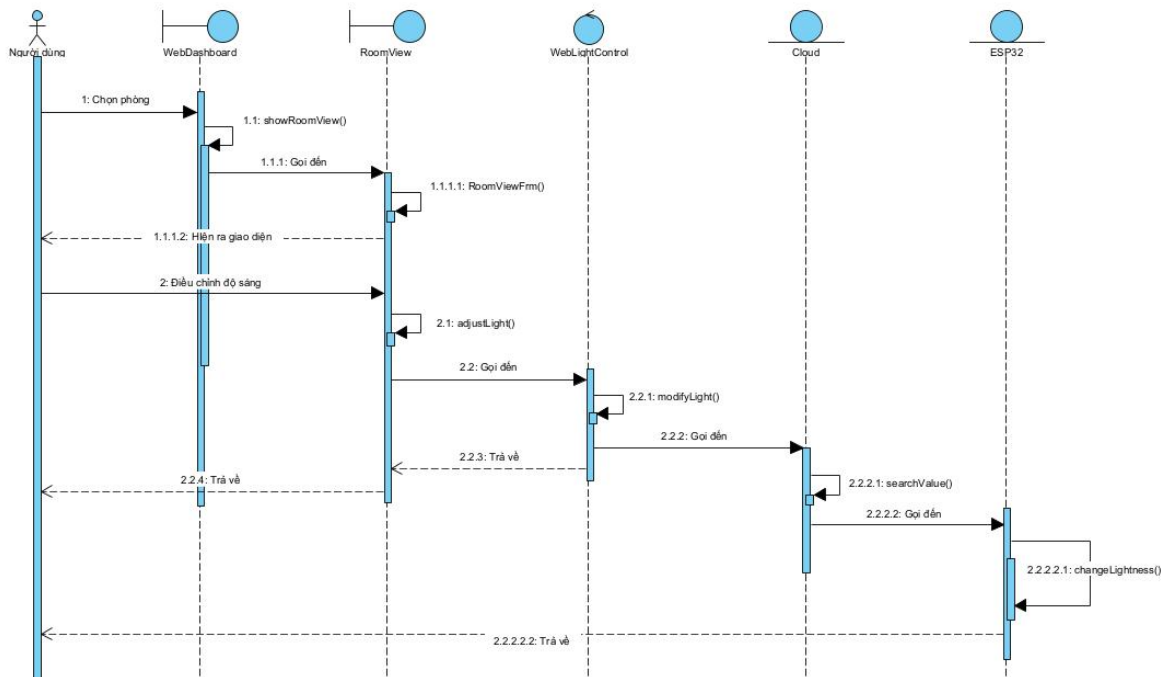
## UC2 – Bật/Tắt tự động (theo cảm biến)

- ESP32: đọc dữ liệu từ cảm biến chuyển động (PIR).
- Khi phát hiện có người → tự động bật đèn, ngược lại thì tắt.  
→ Chỉ ESP32 tham gia trực tiếp, vì nó tự động xử lý tại chỗ mà không cần người dùng hoặc cloud.



### UC3 – Điều chỉnh độ sáng

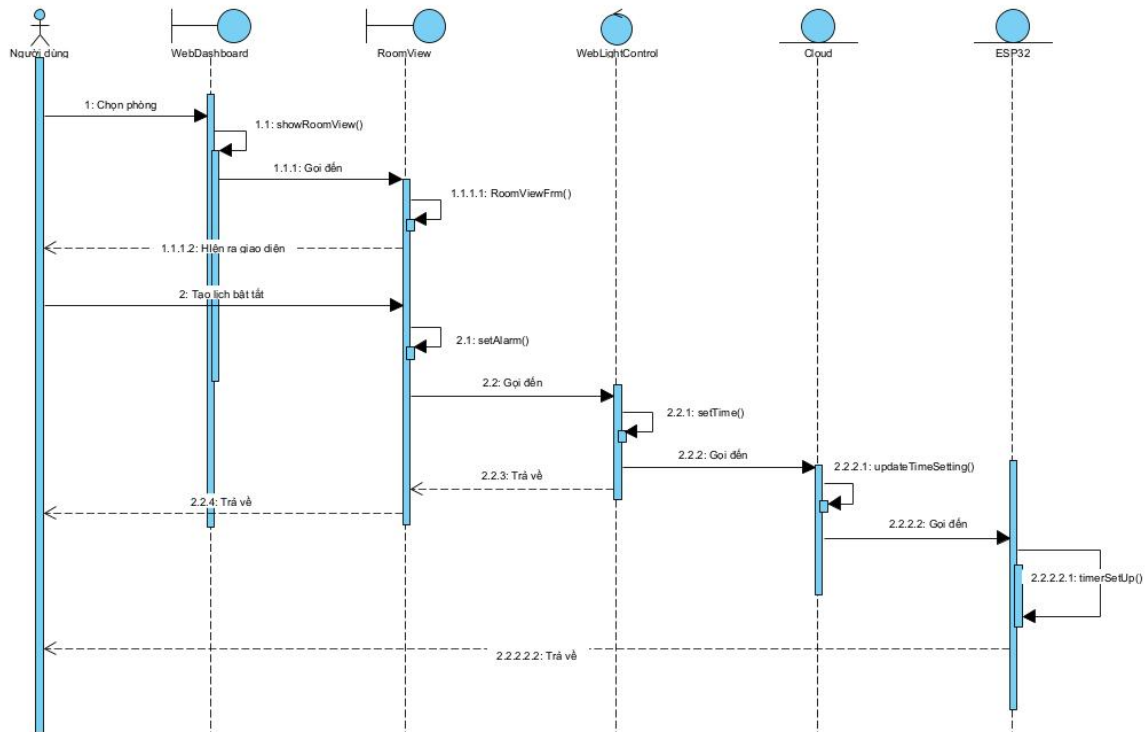
- Người dùng: có thể chỉnh độ sáng mong muốn trên app (ví dụ 50%, 100%).
- ESP32: nhận giá trị từ cloud và điều khiển mạch PWM để thay đổi độ sáng đèn.



#### UC4 – Đặt lịch bật/tắt

- Người dùng: tạo lịch (ví dụ: bật đèn lúc 18h, tắt lúc 22h) trên ứng dụng.
  - Cloud: lưu lịch và gửi lệnh đến ESP32 đúng thời điểm.
  - ESP32: thực hiện bật/tắt theo lịch được gửi xuống.
- Ba actor cùng tham gia: *user* cấu hình → *cloud* lưu & theo dõi → *ESP32* thực hiện.

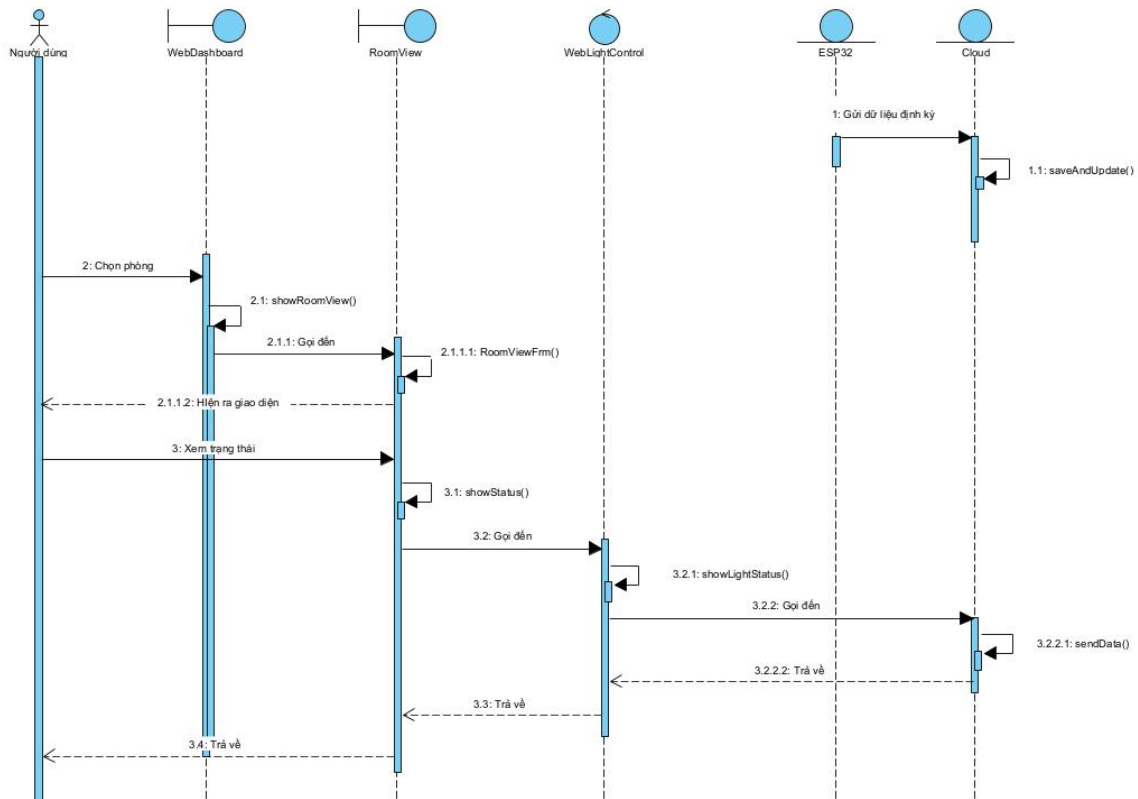




#### UC5 – Xem trạng thái đèn online

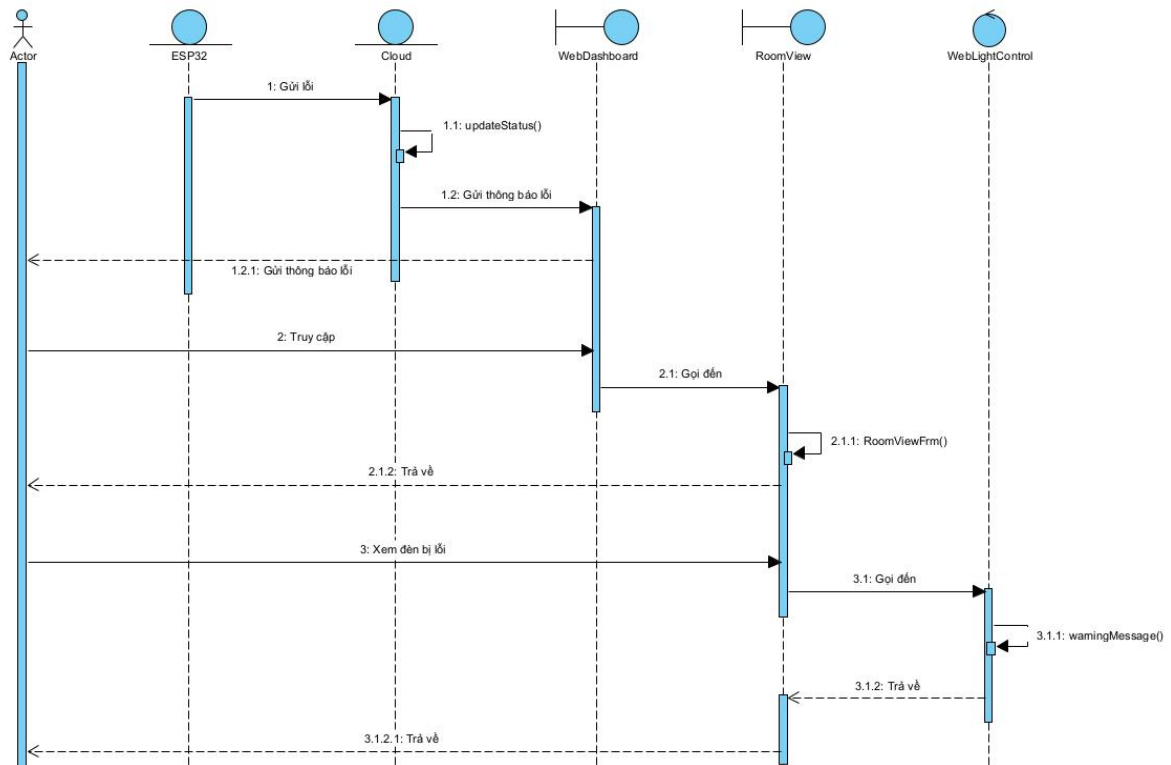
- Người dùng: xem trạng thái hiện tại của đèn (đang bật/tắt, độ sáng bao nhiêu %).
- ESP32: gửi dữ liệu trạng thái đèn định kỳ lên cloud.
- Cloud: lưu và hiển thị thông tin trạng thái cho user qua app/web.

→ Ba actor liên quan: *ESP32* cung cấp dữ liệu → *cloud* lưu trữ → *user* xem.

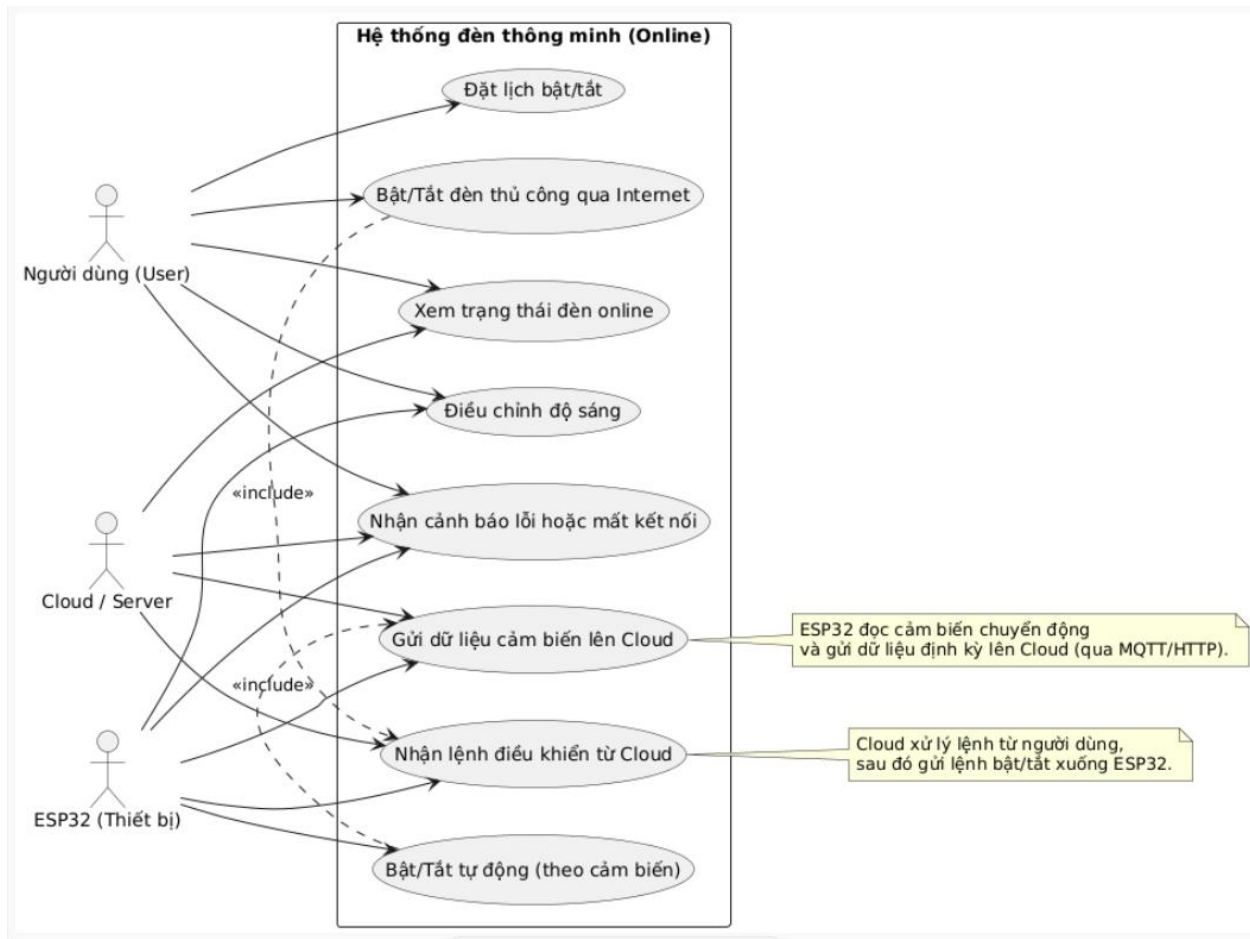


#### UC6 – Nhận cảnh báo lỗi hoặc mất kết nối

- ESP32: gửi cảnh báo khi gặp lỗi (mất Wi-Fi, hỏng cảm biến, relay không phản hồi).
  - Cloud: tiếp nhận cảnh báo và thông báo qua app/web.
  - Người dùng: nhận cảnh báo để kiểm tra hoặc khắc phục.
- Ba actor liên quan: *ESP32 phát hiện lỗi → cloud trung chuyển → user nhận thông báo.*



Sơ đồ usecase tổng quát:



### • Luồng dữ liệu:



### Công nghệ và lý thuyết áp dụng

Phần cứng:

- ESP32: Vi điều khiển tích hợp WiFi, đóng vai trò trung tâm điều khiển và kết nối Internet.
- LED: Mô phỏng đèn trong mô hình, điều khiển bật/tắt thông qua GPIO của ESP32.
- Điện trở ( $220\Omega - 330\Omega$ ): Giới hạn dòng điện để bảo vệ LED, đảm bảo hoạt động ổn định.
- Cảm biến chuyển động
- Relay module

#### Phần mềm:

- Ngôn ngữ lập trình: Arduino C/C++ để lập trình cho ESP32.
- Giao tiếp: MQTT và HTTP để truyền dữ liệu giữa ESP32 và server.
- Server / Backend: Flask
- Cơ sở dữ liệu: Postgres để lưu thông tin thiết bị và trạng thái đèn.

#### Lý thuyết liên quan:

- Internet of Things (IoT): Kết nối và điều khiển thiết bị qua mạng Internet.
- Giao thức MQTT / REST API: Truyền dữ liệu giữa thiết bị và máy chủ.
- Tự động hóa dựa trên cảm biến: Đưa ra quyết định bật/tắt dựa trên dữ liệu môi trường.

## Phân chia công việc & Kế hoạch triển khai

**Thành viên nhóm (4 người):**

Thành viên	Nhiệm vụ chính
<b>Phùng Trung Kiên</b>	Thiết kế và lập trình vi điều khiển ESP32, điều khiển LED, tích hợp cảm biến chuyển động
<b>Nguyễn Việt Quang</b>	Xây dựng hệ thống Backend (API, database, MQTT/HTTP giao tiếp với ESP32)
<b>Phạm Hải Dương</b>	Phát triển giao diện người dùng (Web/App) để điều khiển và giám sát hệ thống
<b>Phạm Công Minh</b>	Thiết kế tài liệu, mô hình hệ thống, hỗ trợ tích hợp và kiểm thử tổng thể

**Kế hoạch triển khai:**

Thời gian	Công việc	Người phụ trách
Tuần 1–2	Nghiên cứu công nghệ, lựa chọn phần cứng	Cả nhóm
Tuần 3–4	Thiết kế mạch, lập trình ESP32 (LED + cảm biến)	Phùng Trung Kiên
Tuần 3–5	Xây dựng Backend, cơ sở dữ liệu	Nguyễn Việt Quang
Tuần 4–6	Phát triển giao diện Web/App, kết nối API	Phạm Hải Dương
Tuần 5–6	Tích hợp hệ thống, kiểm thử, ghi nhận lỗi	Cả nhóm
Tuần 7	Hoàn thiện báo cáo, chuẩn bị demo	Cả nhóm