

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO ĐỒ ÁN III

**Đề tài: Thiết kế và triển khai hệ thống rèm cửa
thông minh**

Sinh viên thực hiện: ĐÀO ANH QUÂN

Mã sinh viên: 20215631

Mã lớp: 744286

Giảng viên hướng dẫn: PGS.TS. NGÔ QUỲNH THU

Hà Nội, 12/2024

LỜI NÓI ĐẦU

Trong thời đại công nghệ 4.0, việc ứng dụng các giải pháp IoT (Internet of Things) đã trở thành xu hướng tất yếu để nâng cao chất lượng cuộc sống. Một trong những ứng dụng tiêu biểu của IoT là hệ thống rèm cửa thông minh, giúp tự động hóa việc điều chỉnh ánh sáng, nhiệt độ trong không gian sống và làm việc.

Dự án này tập trung phát triển một hệ thống rèm cửa thông minh, kết hợp giữa phần cứng (ESP32), nền tảng Firebase và ứng dụng di động phát triển bằng Kotlin. Hệ thống cho phép người dùng dễ dàng theo dõi, quản lý trạng thái rèm cửa và cảm biến thông qua ứng dụng, đồng thời hỗ trợ chế độ tự động dựa trên dữ liệu cảm biến thực tế.

Thông qua dự án, em mong muốn trình bày một giải pháp hiệu quả, tiện dụng và dễ triển khai cho các hộ gia đình và văn phòng, đồng thời khám phá tiềm năng của IoT trong việc tối ưu hóa không gian sống.

Mặc dù đã cố gắng hoàn thiện sản phẩm nhưng không thể tránh khỏi những thiếu hụt về kiến thức và sai sót trong kiểm thử. Em rất mong nhận được những đóng góp cũng như nhận xét thẳng thắn, chi tiết đến từ cô để tiếp tục hoàn thiện hơn nữa. Cuối cùng, em xin được gửi lời cảm ơn đến **PGS.TS. Ngô Quỳnh Thu** đã hướng dẫn em trong suốt quá trình hoàn thiện đồ án này. Em xin chân thành cảm ơn cô!

MỤC LỤC

DANH MỤC HÌNH ẢNH	i
DANH MỤC BẢNG BIỂU	ii
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài	1
1.2.1 Mục tiêu đề tài	1
1.2.2 Phạm vi đề tài	1
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1 IoT (Internet of Things)	3
2.2 Vi điều khiển ESP32	4
2.2.1 Giới thiệu về ESP32	4
2.2.2 Tính năng nổi bật	4
2.3 Giao thức HTTPS	4
2.3.1 Kiến trúc và cơ chế hoạt động	4
2.3.2 Thành phần của HTTPS	5
CHƯƠNG 3. THIẾT KẾ VÀ TRIỂN KHAI	6
3.1 Phân tích yêu cầu bài toán	6
3.1.1 Yêu cầu chức năng	6
3.1.2 Yêu cầu phi chức năng	6
3.2 Phân tích yêu cầu chức năng	7
3.2.1 Biểu đồ Use Case tổng quan	7
3.2.2 Biểu đồ Use Case phân rã và đặc tả chi tiết	7
3.3 Yêu cầu hệ thống	19
3.3.1 Yêu cầu phần cứng	19

3.3.2	Yêu cầu phần mềm	24
3.4	Kiến trúc hệ thống	26
3.5	Triển khai phần cứng	27
3.6	Triển khai phần mềm	28
3.6.1	Lập trình ESP32	28
3.6.2	Triển khai Firebase	29
3.6.3	Phát triển ứng dụng Kotlin trên Android Studio	30
CHƯƠNG 4. KẾT LUẬN		37
TÀI LIỆU THAM KHẢO		38

DANH MỤC HÌNH ẢNH

Hình 2.1	Những ứng dụng phong phú của IoT	3
Hình 3.1	Biểu đồ Use Case tổng quan của hệ thống	7
Hình 3.2	Biểu đồ Use Case phân rã “Quản lý nhà”	7
Hình 3.3	Biểu đồ Use Case phân rã “Quản lý phòng”	11
Hình 3.4	Biểu đồ Use Case phân rã “Quản lý thiết bị”	15
Hình 3.5	Các chân/cổng của ESP32 CH340	21
Hình 3.6	Module điều khiển động cơ L298N	22
Hình 3.7	Module công tắc hành trình	22
Hình 3.8	Động cơ giảm tốc 6-12VDC 8kg.cm	23
Hình 3.9	Cảm biến nhiệt độ, độ ẩm DHT11	24
Hình 3.10	Sơ đồ kiến trúc hệ thống	26
Hình 3.11	Kết nối phần cứng thực tế	27
Hình 3.12	Sơ đồ kết nối động cơ với rèm	28
Hình 3.13	Lập trình ESP32 trên Arduino IDE	29
Hình 3.14	Quản lý dữ liệu trên firebase	30
Hình 3.15	Giao diện danh sách nhà, phòng, thiết bị	31
Hình 3.16	Giao diện thêm nhà, phòng	32
Hình 3.17	Kết nối với thiết bị mới	33
Hình 3.18	Giao diện chỉnh sửa tên nhà, phòng, thiết bị	34
Hình 3.19	Giao diện xóa nhà, phòng, thiết bị	35
Hình 3.20	Điều khiển thiết bị rèm	36

DANH MỤC BẢNG BIỂU

Bảng 3.1	Đặc tả Use Case “Thêm nhà mới”	8
Bảng 3.2	Đặc tả Use Case “Đổi tên nhà”	9
Bảng 3.3	Đặc tả Use Case “Xóa nhà”	10
Bảng 3.4	Đặc tả Use Case “Thêm phòng mới”	12
Bảng 3.5	Đặc tả Use Case “Đổi tên phòng”	13
Bảng 3.6	Đặc tả Use Case “Xóa phòng”	14
Bảng 3.7	Đặc tả Use Case “Thêm thiết bị mới”	16
Bảng 3.8	Đặc tả Use Case “Đổi tên thiết bị”	17
Bảng 3.9	Đặc tả Use Case “Xóa thiết bị”	18
Bảng 3.10	Đặc tả Use Case “Điều khiển thiết bị”	19
Bảng 3.11	Thông số kỹ thuật của ESP32 CH340	20
Bảng 3.12	Thông số kỹ thuật của động cơ giảm tốc	23
Bảng 3.13	Kết nối tín hiệu giữa các thành phần và ESP32	28

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại công nghệ phát triển nhanh chóng, việc ứng dụng các hệ thống thông minh trong gia đình ngày càng trở nên phổ biến. Một trong những nhu cầu thiết yếu là tự động hóa các thiết bị gia dụng nhằm tiết kiệm thời gian và năng lượng. Hệ thống rèm cửa thông minh không chỉ đáp ứng nhu cầu tiện nghi mà còn góp phần tối ưu hóa việc sử dụng ánh sáng tự nhiên.

Xuất phát từ nhu cầu thực tế, em đã quyết định thực hiện đề tài "Thiết kế hệ thống rèm cửa thông minh" với mục tiêu xây dựng một hệ thống có khả năng tự động điều chỉnh dựa trên cường độ ánh sáng môi trường và hỗ trợ điều khiển từ xa qua Internet.

Hệ thống được kỳ vọng sẽ mang lại lợi ích thiết thực trong việc tiết kiệm năng lượng, nâng cao chất lượng cuộc sống và mở rộng ứng dụng IoT vào các thiết bị gia dụng khác.

1.2 Mục tiêu và phạm vi đề tài

1.2.1 Mục tiêu đề tài

Đề tài tập trung vào việc xây dựng và triển khai hệ thống rèm cửa thông minh dựa trên nền tảng IoT, với các mục tiêu chính như sau:

- Tự động hóa: Hệ thống có khả năng tự động đóng/mở rèm dựa trên điều kiện môi trường như ánh sáng và nhiệt độ, độ ẩm.
- Điều khiển từ xa: Người dùng có thể điều khiển rèm thông qua ứng dụng di động được tích hợp với Firebase, hỗ trợ kết nối từ mọi nơi qua Internet.
- Tối ưu chi phí: Sử dụng các linh kiện phần cứng dễ tiếp cận như ESP32, cảm biến ánh sáng và động cơ DC, cùng giao thức HTTPS để tối ưu hóa chi phí và hiệu quả.
- Tăng tính tiện lợi và tiết kiệm năng lượng: Giúp người dùng giảm thiểu thao tác thủ công, đồng thời tối ưu hóa ánh sáng tự nhiên trong không gian sống.

1.2.2 Phạm vi đề tài

1.2.2.1 Phạm vi công nghệ

- Sử dụng phần cứng ESP32 và các cảm biến môi trường (cảm biến ánh sáng, động cơ DC).
- Firebase được chọn làm nền tảng lưu trữ và quản lý dữ liệu, hỗ trợ giao tiếp thời

gian thực.

- Ứng dụng di động viết bằng Kotlin Android, giao tiếp với hệ thống thông qua Firebase và giao thức HTTPS.

1.2.2.2 Phạm vi chức năng

- Điều khiển đóng/mở rèm thủ công hoặc tự động dựa trên ánh sáng môi trường.
- Hiển thị trạng thái rèm (đóng/mở) và các thông số môi trường trên ứng dụng di động.
- Hỗ trợ điều khiển từ xa qua kết nối Internet.

1.2.2.3 Phạm vi ứng dụng

- Đề tài tập trung triển khai mô hình thử nghiệm phù hợp với không gian gia đình hoặc văn phòng nhỏ.
- Giới hạn trong phạm vi hệ thống đơn lẻ, chưa mở rộng thành hệ thống IoT quy mô lớn tích hợp nhiều thiết bị.

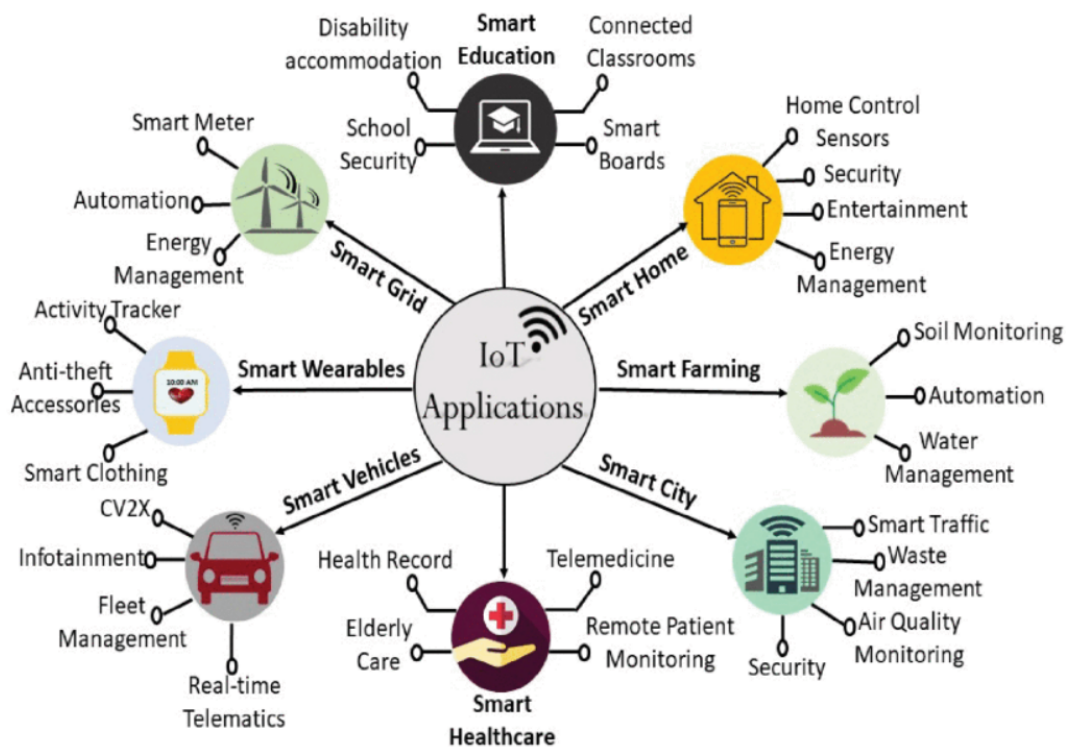
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này giới thiệu về những khái niệm cơ bản, cơ sở lý thuyết cho việc thiết kế và triển khai hệ thống rèm cửa thông minh, bao gồm phần cứng, phần mềm và các công nghệ cần thiết để tạo nên một hệ thống hoàn chỉnh.

2.1 IoT (Internet of Things)

Internet of Things (IoT) là một lĩnh vực mới nổi tập hợp và chuyển giao dữ liệu mà không cần can thiệp của con người. Nó được gọi là một hệ thống các đối tượng kết nối được tích hợp cảm biến, phần mềm và hệ thống điều khiển. [1] Các ứng dụng IoT ngày càng thể hiện sự hiện diện của mình trong hầu hết mọi lĩnh vực, ví dụ như nhà thông minh, y tế, nông nghiệp, giao thông...

IoT cho phép các thiết bị trong nhà giao tiếp với nhau và với con người, mang lại sự tiện lợi, an toàn, và tiết kiệm năng lượng. Trong hệ thống rèm cửa thông minh, IoT đóng vai trò trung gian để kết nối các thiết bị như cảm biến ánh sáng, mô-tơ điều khiển và ứng dụng di động.



Hình 2.1 Những ứng dụng phong phú của IoT

2.2 Vi điều khiển ESP32

2.2.1 Giới thiệu về ESP32

ESP32 là một vi điều khiển mạnh mẽ và linh hoạt do Espressif Systems phát triển, được thiết kế dành cho các ứng dụng Internet of Things (IoT). Nó tích hợp các tính năng vượt trội như kết nối Wi-Fi, Bluetooth và khả năng xử lý hiệu năng cao, phù hợp với nhiều loại dự án thông minh, bao gồm hệ thống điều khiển từ xa, giám sát, và tự động hóa [2].

- **Kiến trúc:** ESP32 sử dụng bộ vi xử lý Xtensa LX6, hỗ trợ hai lõi xử lý (dual-core) hoặc một lõi (single-core).
- **Tích hợp:** Tích hợp Wi-Fi 802.11 b/g/n, Bluetooth 4.2 và Bluetooth Low Energy (BLE)
- **Khả năng mở rộng:** Hỗ trợ GPIO, PWM, ADC, DAC, I2C, SPI, UART, và CAN

2.2.2 Tính năng nổi bật

- **Wi-Fi:** ESP32 có khả năng kết nối Wi-Fi chuẩn 802.11 với băng tần 2.4 GHz, cung cấp giao thức TCP/IP, UDP, HTTP/HTTPS, MQTT, và WebSocket [2].
- **Bluetooth:** Hỗ trợ Bluetooth Classic và BLE, phù hợp với giao tiếp cự ly ngắn và tiêu thụ năng lượng thấp.
- **Hiệu năng:** Tần số xung nhịp lên đến 240 MHz, RAM 520 KB, và bộ nhớ flash lên đến 4 MB (tùy loại).
- **Tiêu thụ năng lượng thấp:** Hỗ trợ các chế độ tiết kiệm năng lượng (Deep Sleep, Light Sleep, Modem Sleep), lý tưởng cho các ứng dụng chạy bằng pin.

2.3 Giao thức HTTPS

HTTPS (HyperText Transfer Protocol Secure) là phiên bản bảo mật của giao thức HTTP, được sử dụng để truyền dữ liệu an toàn qua mạng. HTTPS sử dụng giao thức TLS (Transport Layer Security) hoặc SSL (Secure Sockets Layer) để mã hóa thông tin giữa client (thiết bị người dùng) và server (máy chủ), nhằm bảo vệ tính bảo mật, tính toàn vẹn và khả năng xác thực [3].

2.3.1 Kiến trúc và cơ chế hoạt động

2.3.1.1 Mã hóa dữ liệu (Encryption)

Dữ liệu được mã hóa trước khi gửi qua mạng để ngăn chặn các cuộc tấn công nghe lén (eavesdropping) [3]. HTTPS sử dụng thuật toán mã hóa như AES (Advanced Encryption Standard) để bảo vệ dữ liệu.

2.3.1.2 *Xác thực server (Server Authentication)*

Server sử dụng chứng chỉ SSL/TLS được cấp bởi các tổ chức chứng nhận số (Certificate Authority - CA) để đảm bảo danh tính của mình.

2.3.1.3 *Tính toàn vẹn (Integrity)*

HTTPS sử dụng mã kiểm tra toàn vẹn (Message Authentication Code - MAC) để đảm bảo dữ liệu không bị thay đổi trong quá trình truyền.

2.3.1.4 *Quy trình kết nối HTTPS*

- **Bước 1:** Trình duyệt gửi yêu cầu kết nối HTTPS tới server.
- **Bước 2:** Server phản hồi với chứng chỉ SSL.
- **Bước 3:** Trình duyệt kiểm tra tính hợp lệ của chứng chỉ.
- **Bước 4:** Trình duyệt và server thiết lập khóa mã hóa phiên (Session Key).
- **Bước 5:** Truyền dữ liệu đã mã hóa giữa client và server.

2.3.2 *Thành phần của HTTPS*

2.3.2.1 *Chứng chỉ SSL/TLS:* Đây là chứng chỉ được cấp bởi CA, bao gồm thông tin xác thực của server. Chứng chỉ này được dùng để xác nhận rằng server là hợp pháp và tin cậy

2.3.2.2 *Khóa mã hóa:* Bao gồm khóa công khai (Public Key) và khóa bí mật (Private Key) để mã hóa và giải mã dữ liệu.

2.3.2.3 *Thuật toán mã hóa:* Sử dụng các thuật toán mã hóa bất đối xứng (RSA) và đối xứng (AES) để bảo mật dữ liệu.

CHƯƠNG 3. THIẾT KẾ VÀ TRIỂN KHAI

3.1 Phân tích yêu cầu bài toán

3.1.1 Yêu cầu chức năng

3.1.1.1 Quản lý thiết bị

Người dùng có thể quản lý Nhà, Phòng và thiết bị (rèm) trong từng phòng. Người dùng có thể thêm mới, sửa, xóa các Nhà hay Phòng, đồng thời thêm thiết bị mới.

3.1.1.2 Điều khiển thủ công qua ứng dụng

Ở chức năng này, người dùng sẽ điều khiển rèm cửa thông qua ứng dụng trên điện thoại. Cách điều khiển đơn giản là ấn nút Mở hoặc Đóng, khi ấn nút thì dữ liệu trạng thái rèm sẽ được cập nhật lên Firebase, điều khiển mô-tơ và gửi thông báo về thiết bị di động của người dùng.

3.1.1.3 Chế độ tự động

Đây là chức năng mà rèm cửa sẽ tự động cập nhật trạng thái của nó (Mở hay Đóng) tùy thuộc vào điều kiện môi trường. Khi ánh sáng vượt mức cho phép, rèm cửa sẽ đóng và ngược lại. Trạng thái của rèm cửa cũng sẽ luôn được cập nhật lên ứng dụng di động.

3.1.2 Yêu cầu phi chức năng

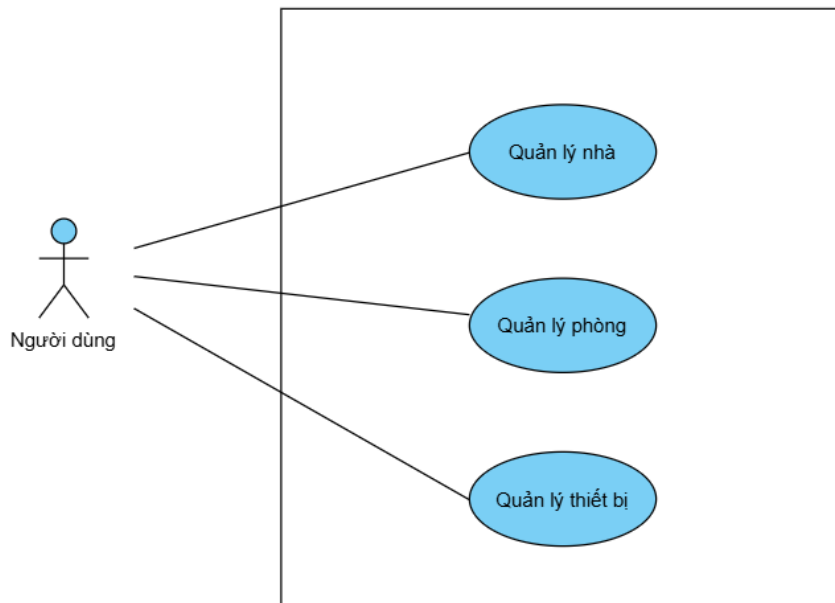
3.1.2.1 *Tốc độ phản hồi:* Thời gian từ khi gửi lệnh đến khi thực hiện yêu cầu không vượt quá 2 giây.

3.1.2.2 *Độ tin cậy:* Hệ thống có thể hoạt động được trong điều kiện được truyền kém.

3.1.2.3 *Bảo mật:* Chỉ người dùng được xác thực mới có quyền điều khiển rèm.

3.2 Phân tích yêu cầu chức năng

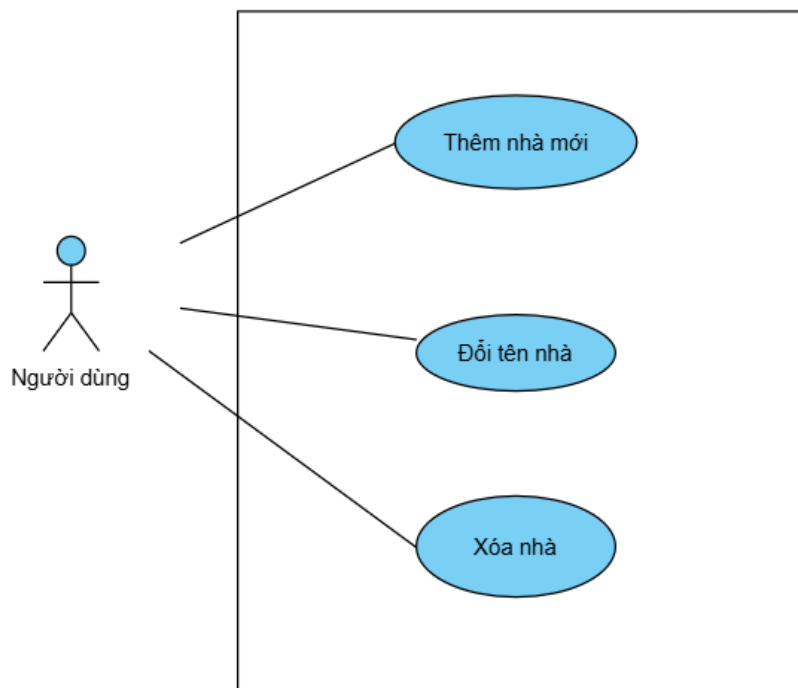
3.2.1 Biểu đồ Use Case tổng quan



Hình 3.1 Biểu đồ Use Case tổng quan của hệ thống

3.2.2 Biểu đồ Use Case phân rã và đặc tả chi tiết

3.2.2.1 Use Case "Quản lý nhà"



Hình 3.2 Biểu đồ Use Case phân rã “Quản lý nhà”

Bảng 3.1 Đặc tả Use Case “Thêm nhà mới”

Mã Use Case	UC001	Tên Use Case	Thêm nhà mới
Mục đích	Người dùng thêm nhà mới		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào hệ thống		
Điều kiện tiên quyết	Không có		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn nút "Thêm nhà"
	2	Hệ thống	Hiển thị hộp thoại thêm nhà
	3	Người dùng	Nhập tên nhà mới cần thêm
	4	Hệ thống	Thông báo thêm nhà mới thành công và cập nhật lại danh sách nhà
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Người dùng	Nhấn thêm khi chưa nhập tên nhà
	4a	Hệ thống	Hiển thị thông báo tên nhà không được để trống
Hậu điều kiện	Không có		

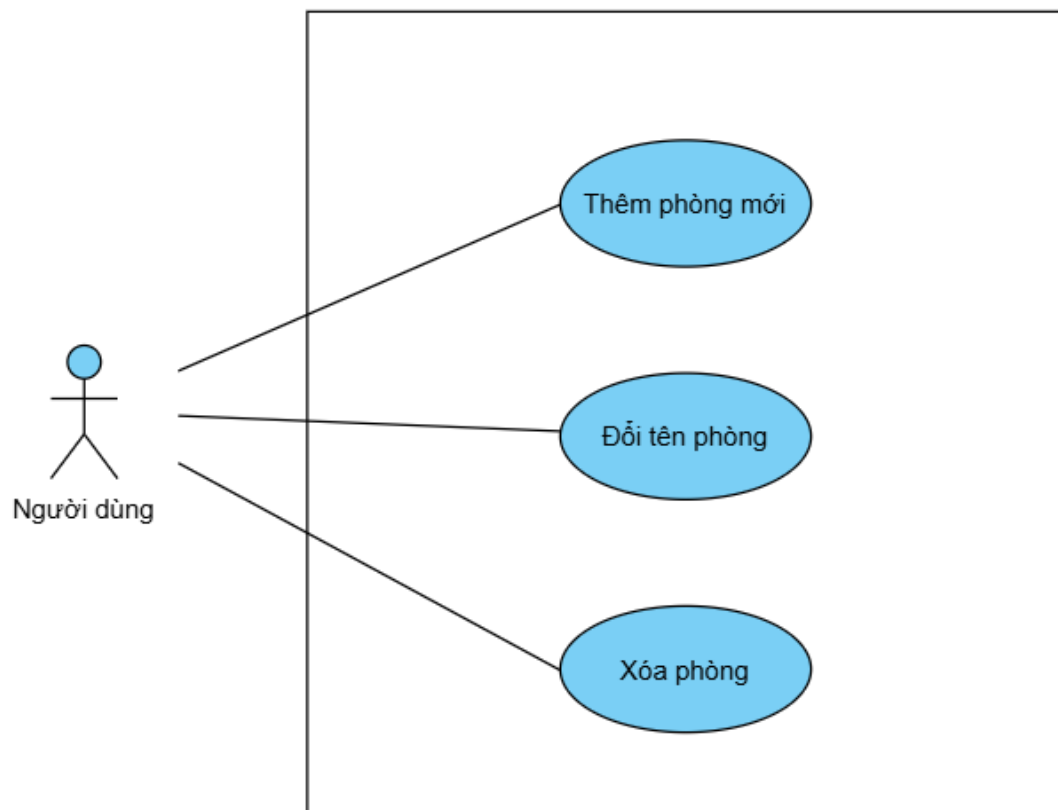
Bảng 3.2 Đặc tả Use Case “Đổi tên nhà”

Mã Use Case	UC002	Tên Use Case	Đổi tên nhà
Mục đích	Người dùng đổi tên nhà		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào hệ thống		
Điều kiện tiên quyết	Không có		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn giữ vào nhà muốn đổi tên"
	2	Hệ thống	Hiển thị lựa chọn đổi tên hoặc xóa nhà
	3	Người dùng	Chọn "Đổi tên"
	4	Hệ thống	Hiển thị hộp thoại đổi tên nhà
	5	Người dùng	Nhập tên muốn thay đổi và ấn lưu
	6	Hệ thống	Đóng hộp thoại và cập nhật lại thông tin danh sách nhà
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Người dùng	Nhấn đóng hộp thoại
	6a	Hệ thống	Đóng hộp thoại đổi tên
Hậu điều kiện	Không có		

Bảng 3.3 Đặc tả Use Case “Xóa nhà”

Mã Use Case	UC003	Tên Use Case	Xóa nhà
Mục đích	Người dùng xóa nhà		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào hệ thống		
Điều kiện tiên quyết	Không có		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn giữ vào nhà muốn xóa"
	2	Hệ thống	Hiển thị lựa chọn đổi tên hoặc xóa nhà
	3	Người dùng	Chọn "Xóa"
	4	Hệ thống	Hiển thị hộp thoại xác nhận xóa
	5	Người dùng	Xác nhận xóa nhà
	6	Hệ thống	Thông báo xóa thành công, đóng hộp thoại và cập nhật lại thông tin danh sách nhà
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Người dùng	Nhấn đóng hộp thoại
	6a	Hệ thống	Đóng hộp thoại xác nhận xóa
Hậu điều kiện	Không có		

3.2.2.2 Use Case "Quản lý phòng"



Hình 3.3 Biểu đồ Use Case phân rã “Quản lý phòng”

Bảng 3.4 Đặc tả Use Case “Thêm phòng mới”

Mã Use Case	UC004	Tên Use Case	Thêm phòng mới
Mục đích	Người dùng thêm phòng mới		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào nhà		
Điều kiện tiên quyết	Đã chọn nhà		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn nút "Thêm phòng"
	2	Hệ thống	Hiển thị hộp thoại thêm phòng
	3	Người dùng	Nhập tên phòng mới cần thêm
	4	Hệ thống	Thông báo thêm phòng mới thành công và cập nhật lại danh sách phòng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Người dùng	Nhấn thêm khi chưa nhập tên phòng
	4a	Hệ thống	Hiển thị thông báo tên phòng không được để trống
Hậu điều kiện	Không có		

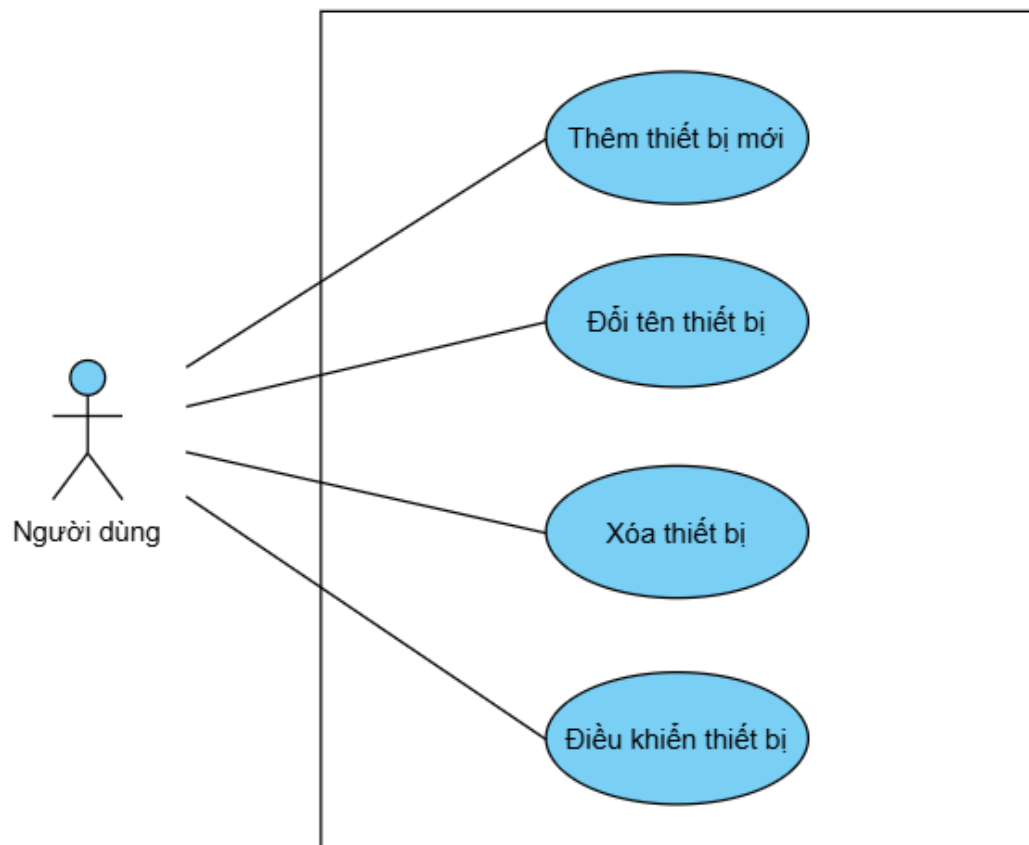
Bảng 3.5 Đặc tả Use Case “Đổi tên phòng”

Mã Use Case	UC005	Tên Use Case	Đổi tên phòng
Mục đích	Người dùng đổi tên phòng		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập nhà		
Điều kiện tiên quyết	Đã chọn nhà		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn giữ vào phòng muốn đổi tên"
	2	Hệ thống	Hiển thị lựa chọn đổi tên hoặc xóa phòng
	3	Người dùng	Chọn "Đổi tên"
	4	Hệ thống	Hiển thị hộp thoại đổi tên phòng
	5	Người dùng	Nhập tên muốn thay đổi và ấn lưu
	6	Hệ thống	Đóng hộp thoại và cập nhật lại thông tin danh sách phòng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Người dùng	Nhấn đóng hộp thoại
	6a	Hệ thống	Đóng hộp thoại đổi tên
Hậu điều kiện	Không có		

Bảng 3.6 Đặc tả Use Case “Xóa phòng”

Mã Use Case	UC006	Tên Use Case	Xóa phòng
Mục đích	Người dùng xóa phòng		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào nhà		
Điều kiện tiên quyết	Đã chọn nhà		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn giữ vào phòng muốn xóa"
	2	Hệ thống	Hiển thị lựa chọn đổi tên hoặc xóa phòng
	3	Người dùng	Chọn "Xóa"
	4	Hệ thống	Hiển thị hộp thoại xác nhận xóa
	5	Người dùng	Xác nhận xóa phòng
	6	Hệ thống	Thông báo xóa thành công, đóng hộp thoại và cập nhật lại thông tin danh sách phòng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Người dùng	Nhấn đóng hộp thoại
	6a	Hệ thống	Đóng hộp thoại xác nhận xóa
Hậu điều kiện	Không có		

3.2.2.3 Use Case "Quản lý thiết bị"



Hình 3.4 Biểu đồ Use Case phân rã “Quản lý thiết bị”

Bảng 3.7 Đặc tả Use Case “Thêm thiết bị mới”

Mã Use Case	UC007	Tên Use Case	Thêm thiết bị mới
Mục đích	Người dùng thêm thiết bị mới		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào phòng cần thêm thiết bị		
Điều kiện tiên quyết	Đã chọn phòng		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn nút "Thêm thiết bị"
	2	Hệ thống	Hiển thị hộp thoại chờ kết nối thiết bị
	3	Người dùng	Kích hoạt thiết bị phần cứng khởi động
	4	Hệ thống	Thông báo thêm thiết bị mới thành công và cập nhật lại danh sách thiết bị trong phòng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Người dùng	Chưa kích hoạt thiết bị trong vòng 2 phút
	4a	Hệ thống	Hiển thị thông báo hết thời gian chờ và tắt hộp thoại
Hậu điều kiện	Không có		

Bảng 3.8 Đặc tả Use Case “Đổi tên thiết bị”

Mã Use Case	UC008	Tên Use Case	Đổi tên thiết bị
Mục đích	Người dùng đổi tên thiết bị		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy phòng		
Điều kiện tiên quyết	Đã chọn phòng		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn giữ vào thiết bị muốn đổi tên trên ứng dụng"
	2	Hệ thống	Hiển thị lựa chọn đổi tên hoặc xóa thiết bị
	3	Người dùng	Chọn "Đổi tên"
	4	Hệ thống	Hiển thị hộp thoại đổi tên thiết bị
	5	Người dùng	Nhập tên muốn thay đổi và ấn lưu
	6	Hệ thống	Đóng hộp thoại và cập nhật lại thông tin danh sách thiết bị
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Người dùng	Nhấn hủy hoặc ấn ra ngoài hộp thoại
	6a	Hệ thống	Đóng hộp thoại đổi tên
Hậu điều kiện	Không có		

Bảng 3.9 Đặc tả Use Case “Xóa thiết bị”

Mã Use Case	UC009	Tên Use Case	Xóa thiết bị
Mục đích	Người dùng xóa thiết bị		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào phòng		
Điều kiện tiên quyết	Đã chọn phòng		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn giữ vào thiết bị muốn xóa"
	2	Hệ thống	Hiển thị lựa chọn đổi tên hoặc xóa thiết bị
	3	Người dùng	Chọn "Xóa"
	4	Hệ thống	Hiển thị hộp thoại xác nhận xóa
	5	Người dùng	Xác nhận xóa thiết bị
	6	Hệ thống	Thông báo xóa thành công, đóng hộp thoại và cập nhật lại thông tin danh sách thiết bị
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Người dùng	Nhấn đóng hộp thoại
	6a	Hệ thống	Đóng hộp thoại xác nhận xóa
Hậu điều kiện	Không có		

Bảng 3.10 Đặc tả Use Case “Điều khiển thiết bị”

Mã Use Case	UC010	Tên Use Case	Điều khiển thiết bị
Mục đích	Người dùng điều khiển thiết bị		
Tác nhân	Người dùng		
Sự kiện kích hoạt	Người dùng truy cập vào phòng chứa thiết bị cần điều khiển		
Điều kiện tiên quyết	Đã chọn phòng		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Điều khiển rèm bằng cách chuyển đổi chế độ "Thủ công" và "Tự động"
	2	Hệ thống	Thông báo và chuyển sang chế độ tương ứng
	3	Người dùng	Nếu chọn chế độ "Thủ công", người dùng điều khiển rèm đóng mở bằng nút ấn trên ứng dụng
	4	Hệ thống	Hiển thị thông báo đang đóng/mở rèm, và điều khiển thiết bị tương ứng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Người dùng	Nhấn mở khi rèm đã mở, hoặc đóng khi rèm đã đóng
	4a	Hệ thống	Thông báo rèm đã mở/đóng, thiết bị không thay đổi gì
Hậu điều kiện	Không có		

3.3 Yêu cầu hệ thống

3.3.1 Yêu cầu phần cứng

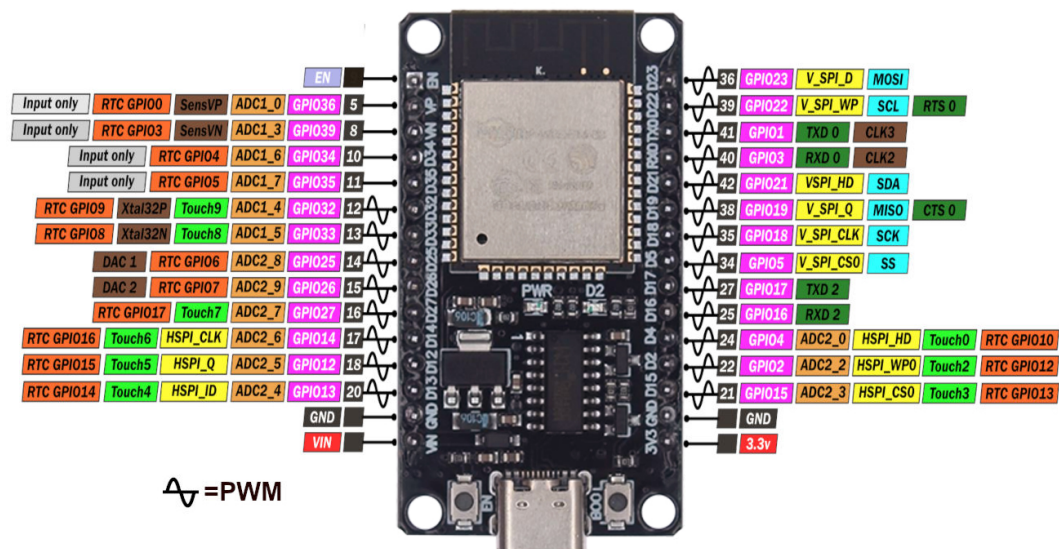
3.3.1.1 ESP32 CH340

ESP32 CH340 là phiên bản ESP32 sử dụng chip CH340 làm giao tiếp USB-to-Serial, cho phép dễ dàng kết nối bộ vi điều khiển với máy tính qua cổng USB. Module này có khả năng tương thích cao, hỗ trợ hầu hết các hệ điều hành, bao gồm Windows, macOS và Linux. Chi phí rẻ, tích hợp cổng USB trực tiếp, không cần module chuyển đổi

USE-to-TTL bên ngoài. Bảng dưới đây là thông số của ESP32 CH340.

Bảng 3.11 Thông số kỹ thuật của ESP32 CH340

Thông số	Chi tiết
Module chính	ESP-WROOM-32
Vi xử lý	32-bit, tốc độ 240 MHz
Bộ nhớ Flash	4 MByte (32 Mbit) SPI Flash
Bộ nhớ ROM	448 KB
Bộ nhớ SRAM	520 KB
SRAM trong RTC	16 KB
Điện áp cung cấp	5V
Kết nối không dây	- Wi-Fi: 2.4 GHz
	- Bluetooth: 4.2
Ăng-ten	PCB Antenna
Kết nối USB	USB Type-C
Nút chức năng	- Reset
	- User/Firmware
Đèn LED	- LED báo trạng thái người dùng
	- LED nguồn
Giao tiếp USB-to-Serial	Chip CH340 (Cung cấp cổng USB ảo để gỡ lỗi và nâng cấp firmware)
Kích thước	51.5 x 28.5 mm
Hình thức cấp nguồn	- Qua cổng USB Type-C
	- Qua chân header



Hình 3.5 Các chân/cổng của ESP32 CH340

3.3.1.2 Cảm biến ánh sáng LDR

Cảm biến ánh sáng hoạt động dựa trên nguyên tắc phát hiện và chuyển đổi ánh sáng thành tín hiệu điện. Các phần tử nhạy sáng trong cảm biến, như photodiode hoặc photoresistor, sẽ thay đổi đặc tính điện của chúng khi ánh sáng chiếu vào. LDR (Light Dependent Resistor) là một loại điện trở đặc biệt, có giá trị điện trở thay đổi theo cường độ ánh sáng chiếu vào. Khi ánh sáng tăng, điện trở giảm và ngược lại. Giá trị điện trở này sẽ được chuyển đổi thành giá trị ánh sáng. Trong dự án này, LDR được sử dụng trong chế độ Tự động. Khi ánh sáng cao hơn mức cho phép thì sẽ làm động cơ quay theo chiều thuận (mở rèm) và ngược lại (đóng rèm).

Ưu điểm:

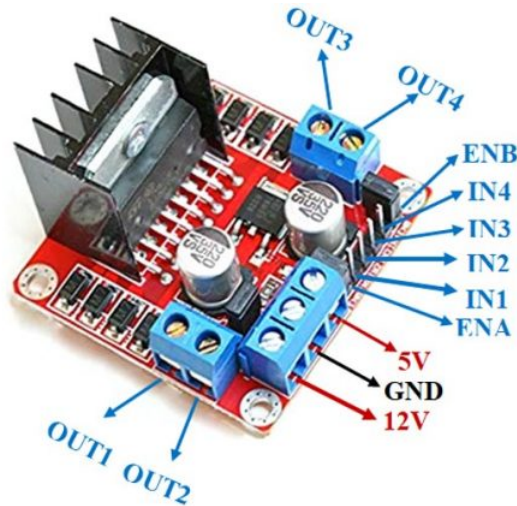
- Giá thành rẻ.
- Dễ sử dụng và kết nối.
- Không cần nguồn cung cấp riêng (lấy nguồn trực tiếp từ mạch).

Nhược điểm:

- Độ chính xác không cao so với cảm biến ánh sáng kỹ thuật số (như BH1750, TSL2561).
- Chịu ảnh hưởng bởi nhiều ánh sáng môi trường.
- Không nhạy với ánh sáng hồng ngoại.

3.3.1.3 Module điều khiển động cơ L298N

Module L298N là một bộ điều khiển cầu H (H-Bridge) được sử dụng để điều khiển động cơ DC hoặc động cơ bước. Nó dựa trên IC L298N, cho phép điều khiển tốc độ và hướng quay của động cơ.



Hình 3.6 Module điều khiển động cơ L298N [4]

3.3.1.4 Module công tắc hành trình

Module công tắc hành trình (Limit Switch) là một thiết bị cơ học được sử dụng để phát hiện vị trí hoặc chuyển động của một đối tượng. Nó thường được tích hợp trong các hệ thống tự động hóa, robot, hoặc cơ cấu cơ khí để giới hạn phạm vi chuyển động hoặc xác định trạng thái. Trong dự án này, công tắc hành trình dùng để xác định thời điểm mà rèm đã thực hiện hết hành trình đóng/mở của nó. Khi hết hành trình, công tắc hành trình sẽ được kích hoạt để dừng động cơ điều khiển lại.



Hình 3.7 Module công tắc hành trình

3.3.1.5 Động cơ giảm tốc

Động cơ giảm tốc là sự kết hợp giữa động cơ điện và hộp số giảm tốc (gearbox). Hộp số giảm tốc được thiết kế để giảm tốc độ quay (RPM) và tăng mô-men xoắn (torque) của động cơ. Đây là thành phần quan trọng trong các hệ thống cơ đi và robot, giúp tối ưu hóa hiệu suất và điều khiển chuyển động. Động cơ được dùng trong dự án này có thông số kỹ thuật như sau:

Bảng 3.12 Thông số kỹ thuật của động cơ giảm tốc

Thông số	Chi tiết
Điện áp	6-12 VDC
Dòng không tải	25mA
Tốc độ không tải	24 vòng/phút
Mô-men xoắn	8kg.cm
Đường kính trục đầu ra	6mm
Chiều dài trục đầu ra	19.5mm
Đường kính ngoài động cơ	27.8mm
Đường kính ngoài hộp số	39.8mm
Chiều dài động cơ bánh răng	77m

Động cơ trên phù hợp với dự án điều khiển rèm bởi vì có tốc độ quay thấp, mô-men xoắn lớn, giúp cho chuyển động được trơn chu, và có thể điều khiển được rèm nặng do lực mô-men xoắn lớn. Ngoài ra động cơ có độ bền cao, tiết kiệm năng lượng, giá thành rẻ, tối ưu chi phí vận hành.



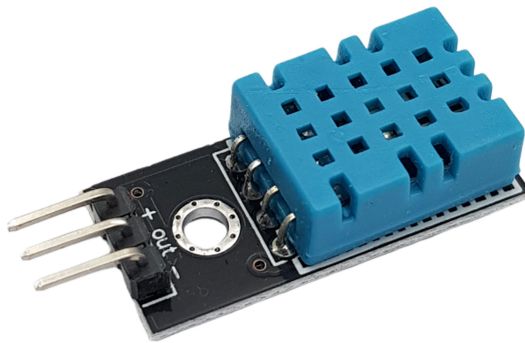
Hình 3.8 Động cơ giảm tốc 6-12VDC 8kg.cm [5]

3.3.1.6 Cảm biến DHT11

Cảm biến DHT11 là một module cảm biến giá rẻ, dễ sử dụng, được thiết kế để đo nhiệt độ và độ ẩm trong không gian. Nó được sử dụng phổ biến trong các hệ thống IoT, tự động hóa, và các dự án DIY. DHT11 bao gồm:

- Cảm biến nhiệt độ sử dụng một nhiệt điện trở (Thermistor) để đo nhiệt độ môi trường. Dải nhiệt độ đo từ 0°C đến 50°C, độ chính xác $\pm 2^\circ\text{C}$.
- Cảm biến độ ẩm sử dụng cảm biến điện dung để đo độ ẩm tương đối (Relative Humidity). Dải đo từ 20% đến 90% RH, độ chính xác $\pm 5\%$ RH.

Cảm biến này không trực tiếp liên quan đến việc điều khiển rèm, nhưng ta có thể sử dụng thông tin về nhiệt độ và độ ẩm để làm điều kiện điều khiển rèm. Ngoài ra đây cũng là một thông tin hữu ích để người dùng biết được điều kiện môi trường trong phòng.



Hình 3.9 Cảm biến nhiệt độ, độ ẩm DHT11 [6]

3.3.2 Yêu cầu phần mềm

3.3.2.1 Arduino IDE

Arduino IDE là một phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino. Có rất nhiều các module Arduino như Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều module khác.

Mỗi module chứa một bộ vi điều khiển trên bo mạch được lập trình và chấp nhận thông tin dưới dạng mã. Mã chính, còn được gọi là sketch, được tạo trên nền tảng IDE sẽ tạo ra một file Hex, sau đó được chuyển và tải lên trong bộ điều khiển trên bo.

Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình chỉnh sửa và Trình biên dịch,

phần đầu sử dụng để viết mã được yêu cầu và phần sau được sử dụng để biên dịch và tải mã lên module Arduino. Môi trường này hỗ trợ cả ngôn ngữ C và C ++.

3.3.2.2 *Firebase*

Firebase là một nền tảng được Google phát triển nhằm hỗ trợ các nhà phát triển xây dựng, cải thiện và mở rộng ứng dụng một cách nhanh chóng. Ra mắt lần đầu tiên vào năm 2011 và được Google mua lại vào năm 2014, Firebase đã trở thành một công cụ phổ biến trong cộng đồng lập trình nhờ tích hợp nhiều dịch vụ mạnh mẽ và dễ sử dụng.

Firebase cung cấp một hệ sinh thái đa dạng bao gồm cơ sở dữ liệu thời gian thực, lưu trữ tệp, thông báo đẩy, xác thực người dùng, phân tích hành vi và nhiều tính năng khác. Điểm đặc biệt của Firebase là khả năng đồng bộ hóa thời gian thực giữa các thiết bị, giúp tạo ra trải nghiệm liền mạch cho người dùng. Ngoài ra, nền tảng này còn hỗ trợ triển khai nhanh chóng trên cả Android, iOS và ứng dụng web. Một số dịch vụ được sử dụng trong dự án:

Firebase Authentication: giúp xác thực người dùng dễ dàng thông qua các phương thức như email/mật khẩu, số điện thoại, hoặc các nhà cung cấp bên thứ ba như Google, Facebook, và Apple. Dịch vụ này hỗ trợ cả xác thực ẩn danh và tích hợp chặt chẽ với các dịch vụ khác để đảm bảo bảo mật [7].

Firebase Realtime Database: Đây là cơ sở dữ liệu NoSQL thời gian thực, lưu trữ dữ liệu dưới dạng JSON. Realtime Database cho phép đồng bộ dữ liệu tức thời giữa các thiết bị, rất phù hợp cho các ứng dụng như chat, theo dõi GPS, và quản lý trạng thái trực tuyến [7].

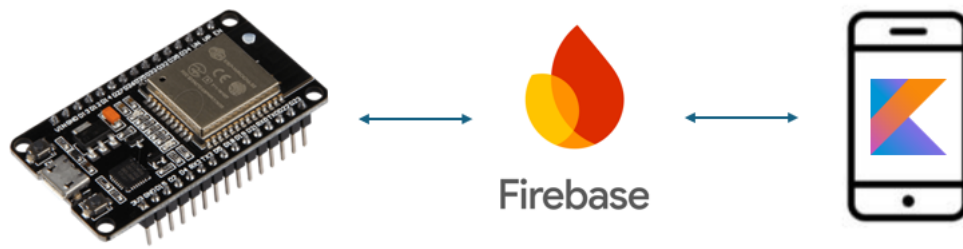
3.3.2.3 *Android Studio*

Android Studio là môi trường phát triển tích hợp (IDE) chính thức do Google cung cấp để phát triển ứng dụng Android. Ra mắt lần đầu vào năm 2013, Android Studio được xây dựng dựa trên nền tảng IntelliJ IDEA, cung cấp một bộ công cụ mạnh mẽ giúp lập trình viên thiết kế, lập trình, kiểm tra và triển khai ứng dụng Android hiệu quả.

Android Studio nổi bật với giao diện thân thiện, tích hợp sẵn các tính năng như trình chỉnh sửa mã thông minh, hỗ trợ thiết kế giao diện trực quan, và các công cụ phân tích hiệu năng ứng dụng. Trình giả lập (emulator) tích hợp trong Android Studio cho phép lập trình viên kiểm tra ứng dụng trên nhiều thiết bị ảo với cấu hình đa dạng, từ điện thoại, máy tính bảng đến thiết bị đeo thông minh (wearables) hoặc TV Android.

Kotlin và Java là hai ngôn ngữ chính thức để phát triển ứng dụng Android trên Android Studio. Ngoài ra, IDE này cũng hỗ trợ C++ thông qua Android NDK, giúp tạo các ứng dụng yêu cầu hiệu năng cao hoặc xử lý ở mức hệ thống.

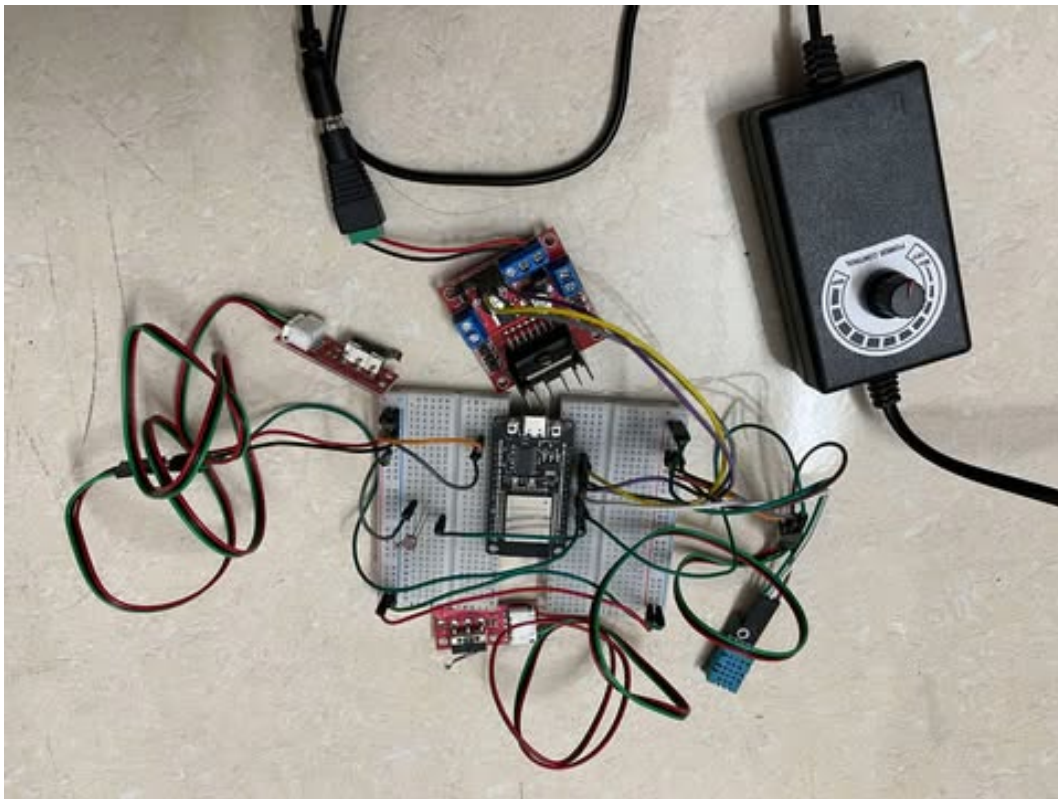
3.4 Kiến trúc hệ thống



Hình 3.10 Sơ đồ kiến trúc hệ thống

Hình 3.10 biểu diễn sơ đồ kiến trúc hệ thống gồm 3 thành phần chính là ESP32, Firebase và Ứng dụng Android. Hệ thống được thiết kế theo mô hình Client-Server, trong đó ESP32 đóng vai trò là thiết bị phần cứng thực hiện các thao tác điều khiển và thu thập dữ liệu từ các cảm biến. Firebase là trung gian lưu trữ và đồng bộ dữ liệu thời gian thực giữa ESP32 và ứng dụng Kotlin. Ứng dụng Android được viết bằng ngôn ngữ Kotlin, là giao diện người dùng (UI), cho phép điều khiển rèm và hiển thị trạng thái hệ thống.

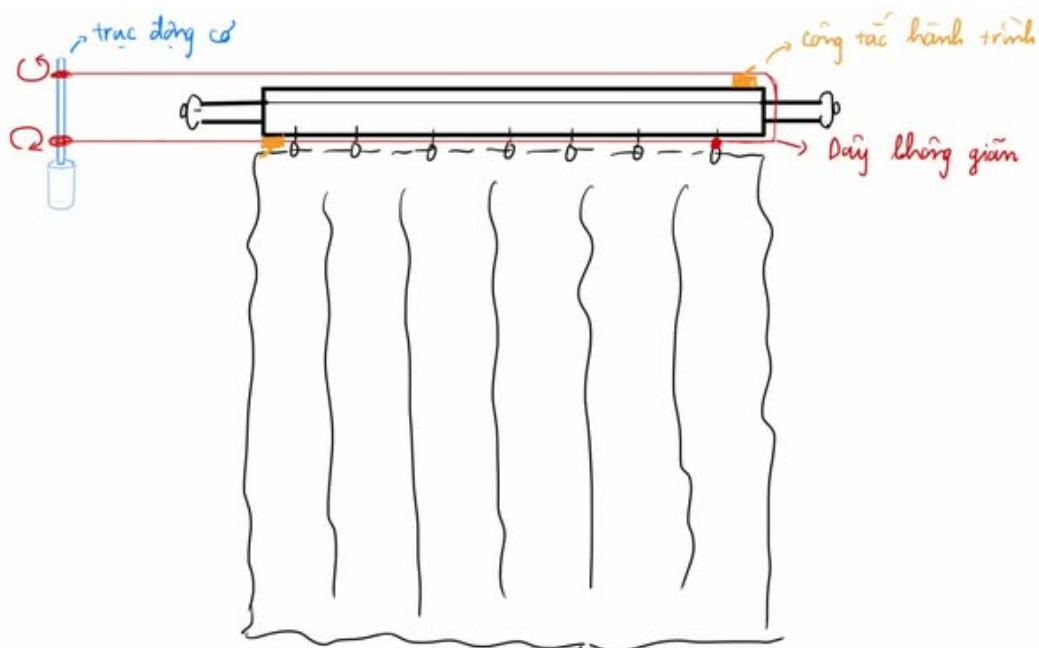
3.5 Triển khai phần cứng



Hình 3.11 Kết nối phần cứng thực tế

Hình 3.11 là sơ đồ mạch của hệ thống trong thực tế (chưa nối với động cơ). Ý tưởng của hệ thống là dùng ESP32 để thu thập dữ liệu về nhiệt độ, độ ẩm phòng thông qua cảm biến DHT11, đồng thời quản lý trạng thái đóng mở của rèm để điều khiển thông qua dữ liệu trên Firebase. Động cơ giảm tốc được kết nối với module L298N để điều khiển tốc độ và hướng quay. Ngoài ra, hệ thống còn gồm hai công tắc hành trình, một công tắc điều khiển dừng khi động cơ đang quay cùng chiều kim đồng hồ (rèm mở), công tắc còn lại điều khiển khi động cơ đang quay ngược chiều kim đồng hồ (rèm đóng).

Về việc lắp đặt động cơ để điều khiển rèm, ý tưởng ở đây là sẽ dùng dây không giãn. Hai đầu dây được vòng qua khung rèm và cuộn thành vòng ngược chiều nhau tại trục rèm. Một điểm tại mép rèm sẽ được cố định với dây. Mô hình thực tế như hình 3.12. Khi động cơ quay cùng chiều kim đồng hồ, dây phía trên sẽ di chuyển về bên phải, dây phía dưới di chuyển về bên trái (giống như một vòng dây kín), làm rèm mở. Khi rèm chạy đến công tắc hành trình, rèm sẽ kích hoạt công tắc làm động cơ dừng, quá trình mở rèm kết thúc. Tương tự như quá trình đóng, chỉ khác động cơ sẽ quay ngược chiều kim đồng hồ.



Hình 3.12 Sơ đồ kết nối động cơ với rèm

Bảng dưới đây là bảng kết nối giữa các chân tín hiệu của các thành phần để phù hợp với mã nguồn.

Bảng 3.13 Kết nối tín hiệu giữa các thành phần và ESP32

Chân các thiết bị	Số hiệu cổng GPIO
DHT11	GPIO 27
IN1 của L298N	GPIO 12
IN2 của L298N	GPIO 13
EN của L298N	GPIO 14
Cảm biến ánh sáng	GPIO 34
Công tắc hành trình mở	GPIO 33
Công tắc hành trình đóng	GPIO 32

3.6 Triển khai phần mềm

Hướng tới sự thuận tiện trong việc quản lý và điều khiển thiết bị rèm, đồ án này xây dựng ứng dụng Android trên điện thoại bằng ngôn ngữ Kotlin. Quy trình triển khai bao gồm lập trình ESP32, triển khai Firebase và phát triển ứng dụng Kotlin trên Android Studio.

3.6.1 Lập trình ESP32

Mục tiêu của việc lập trình ESP32 là kết nối ESP32 với Firebase qua giao thức HTTPS, thu thập dữ liệu từ cảm biến (ánh sáng, nhiệt độ, độ ẩm, công tắc hành trình) để

phục vụ cho việc lưu trữ và quản lý trạng thái của động cơ.

Các bước triển khai bao gồm:

- Cài đặt thư viện Firebase và Wi-Fi trên Arduino IDE
- Viết mã kết nối ESP32 với Firebase
- Viết mã điều khiển module L298N để thực hiện đóng mở rèm, theo dõi trạng thái cảm biến trên Firebase
- Gửi trạng thái và dữ liệu cảm biến về Firebase



Hình 3.13 Lập trình ESP32 trên Arduino IDE

3.6.2 Triển khai Firebase

Quá trình này bao gồm việc tạo cơ sở dữ liệu thời gian thực (Realtime Database) trên Firebase, định nghĩa cấu trúc dữ liệu và kết nối với ESP32, ứng dụng Android. Mục đích là để lưu trữ và đồng bộ lệnh, trạng thái, dữ liệu của cảm biến, đồng thời thiết lập xác thực người dùng.

Dữ liệu trên firebase được lưu trữ theo từng thành phần riêng biệt như Nhà, Phòng, Cảm biến, Rèm. Các thành phần này sẽ bao gồm id để định danh, đồng thời chứa các thuộc tính, các định danh của các thành phần khác. Việc tách biệt các thành phần giúp dễ dàng trong việc thêm, sửa, xóa từng thành phần và hỗ trợ tối ưu cho việc truy vấn.



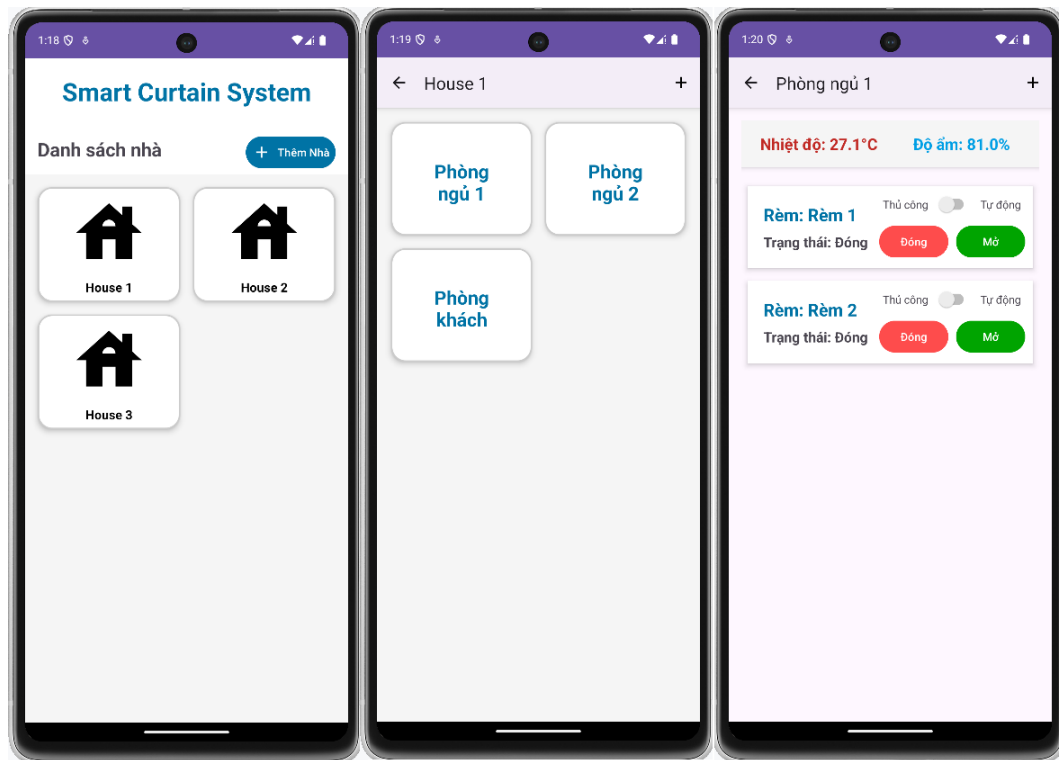
Hình 3.14 Quản lý dữ liệu trên firebase

3.6.3 Phát triển ứng dụng Kotlin trên Android Studio

Đây là thành phần giúp người dùng có thể tương tác, quản lý hệ thống của mình thông qua điện thoại thông minh. Chỉ cần kết nối với Internet, người dùng có thể theo dõi trạng thái các thiết bị mọi lúc, mọi nơi. Các bước triển khai ứng dụng Android:

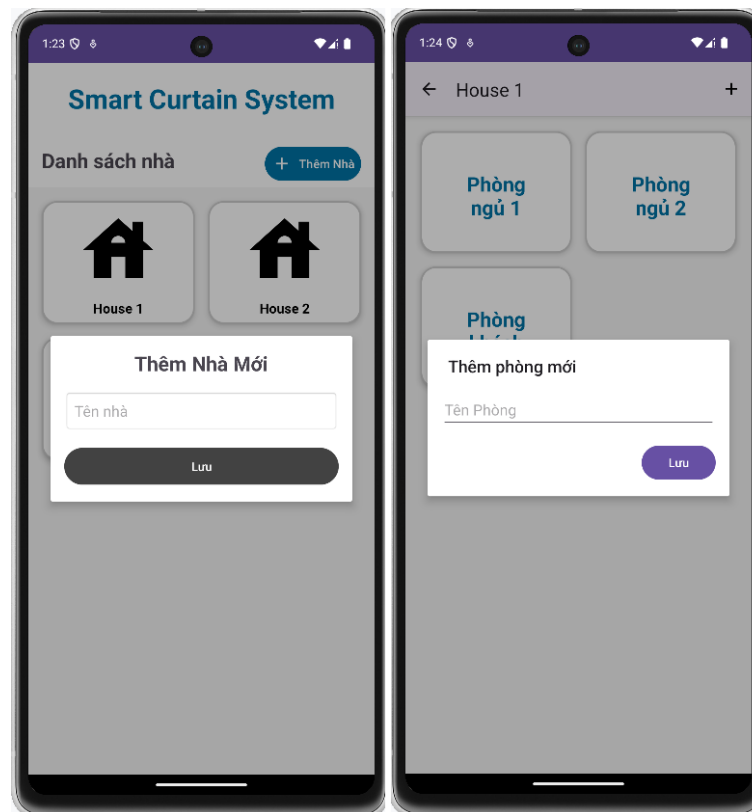
- **Thiết kế giao diện người dùng (UI):** Sử dụng XML để tạo màn hình điều khiển, kết hợp các thành phần chính như nút bấm (Button), Công tắc (Switch)...
- **Tích hợp Firebase vào ứng dụng Android:** Sử dụng Firebase SDK để kết nối Firebase với ứng dụng
- **Lập trình logic điều khiển:** Bao gồm việc lập trình lấy dữ liệu từ Firebase để hiển thị ra màn hình, chuyển đổi giữa các giao diện với nhau, lập trình các chức năng thêm, sửa, xóa, thêm thiết bị mới, điều khiển rèm...
- **Kiểm thử ứng dụng:** Kiểm tra việc gửi lệnh và đồng bộ trạng thái với Firebase, đảm bảo dữ liệu được cập nhật chính xác.

Kết quả thu được:



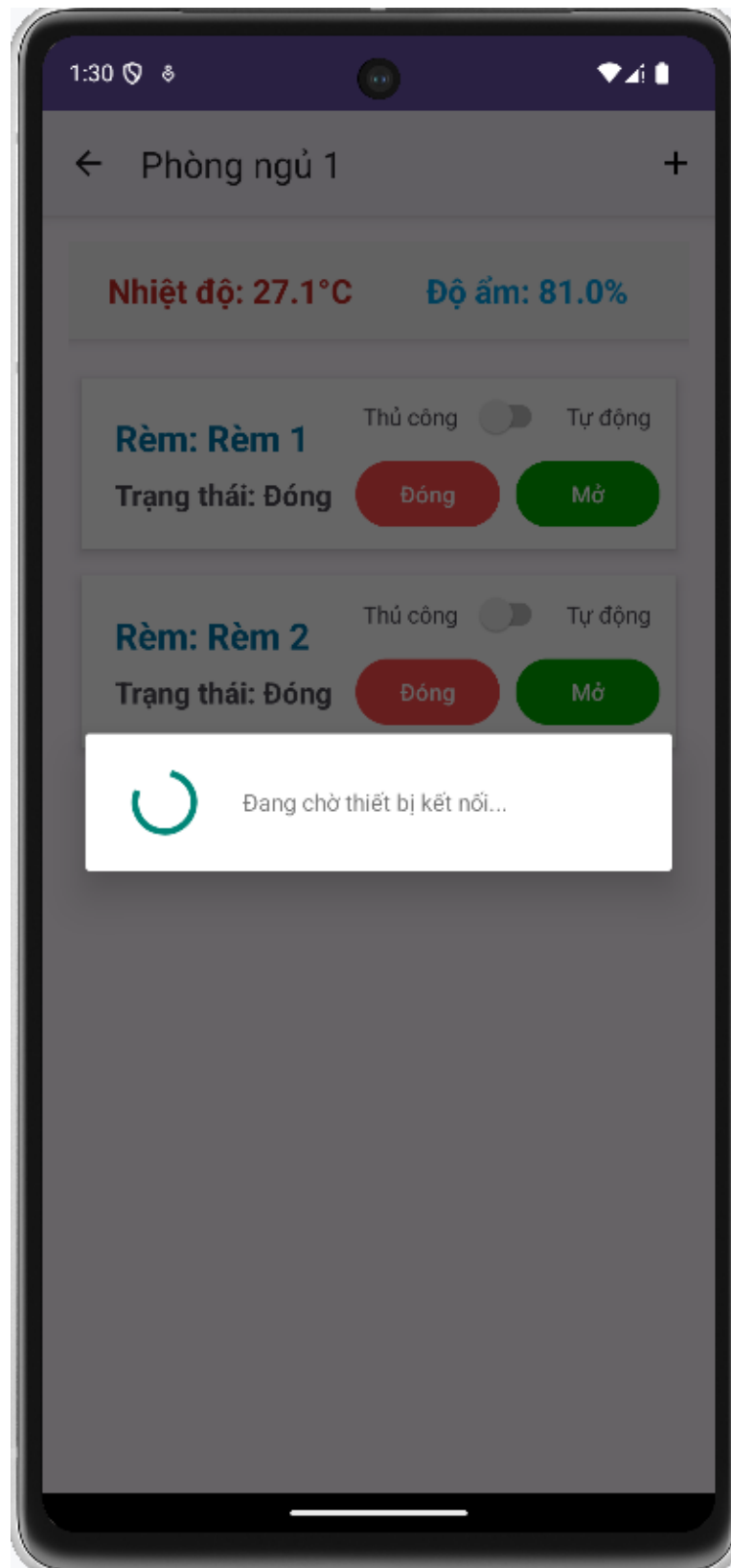
Hình 3.15 Giao diện danh sách nhà, phòng, thiết bị

Hình 3.15 là các giao diện danh sách nhà, phòng trong nhà và các thiết bị trong phòng. Dữ liệu được lấy từ Firebase và sẽ được cập nhật thời gian thực khi có bất kỳ thay đổi nào ở cơ sở dữ liệu.



Hình 3.16 Giao diện thêm nhà, phòng

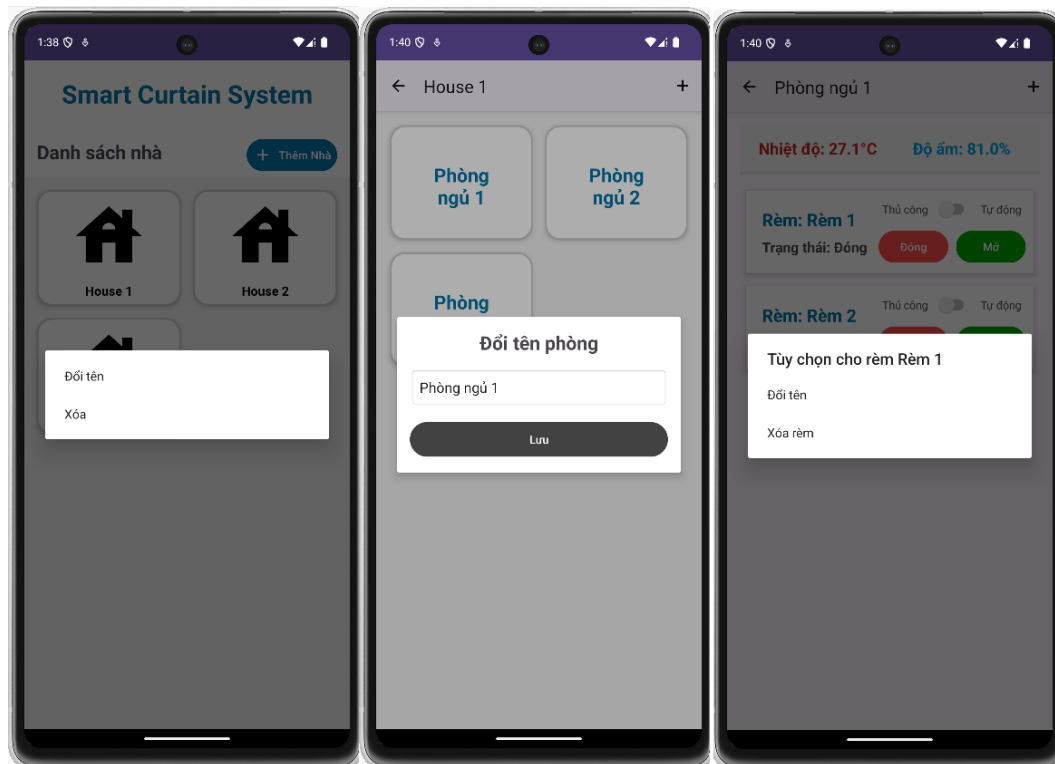
Hình 3.16 biểu diễn chức năng thêm mới nhà và phòng. Thông tin cần nhập là tên của nhà/phòng. Khi thêm mới thì cơ sở dữ liệu sẽ tạo thêm 1 nhà/phòng tương ứng với ID định danh riêng và cập nhật ngay lập tức giao diện hiển thị danh sách.



Hình 3.17 Kết nối với thiết bị mới

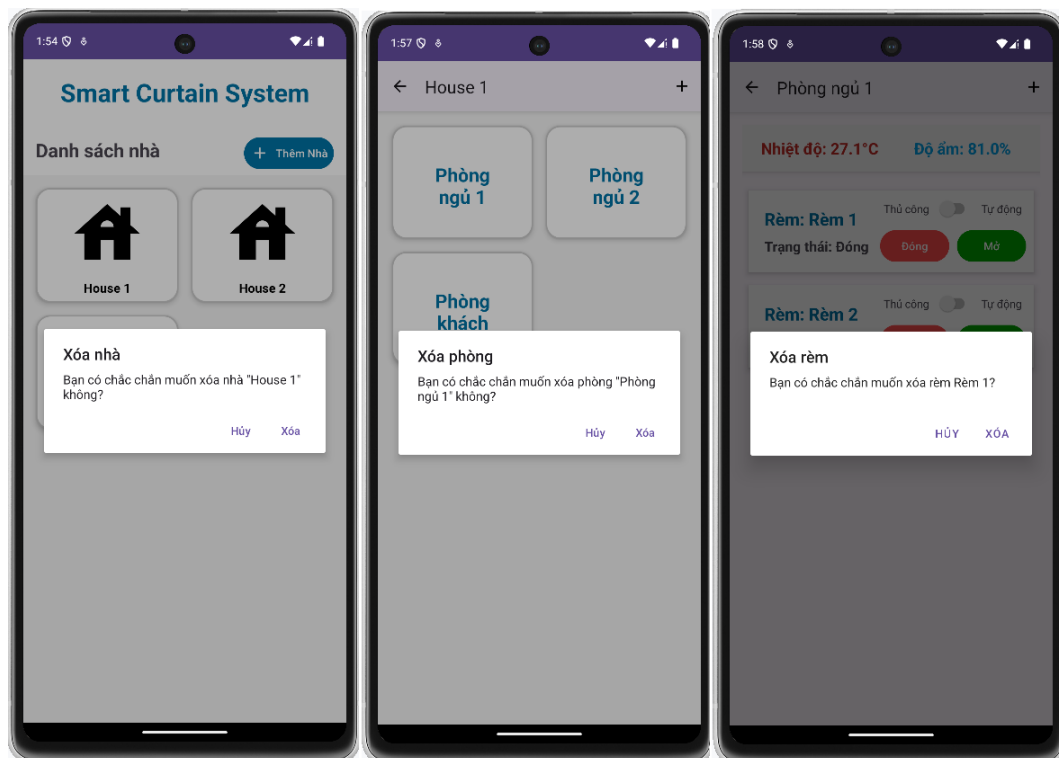
Hình 3.17 là giao diện khi thêm thiết bị mới. Để bắt đầu thêm, nhấn nút "+" ở góc trên bên phải của giao diện chi tiết phòng. Khi đó, hệ thống sẽ lắng nghe sự kiện thêm

mới thiết bị phần cứng với thời gian chờ là 2 phút. Trong thời gian này, kích hoạt thiết bị phần cứng, Firebase sẽ thêm mới một thiết bị và cảm biến gắn với phòng là phòng hiện tại. Khi thêm mới thành công, hiển thị pop-up thông báo và cập nhật lại giao diện danh sách thiết bị trong phòng. Nếu trong 2 phút không có thiết bị mới được kích hoạt thì ứng dụng sẽ tắt hộp thoại chờ kết nối và gửi thông báo cho người dùng.



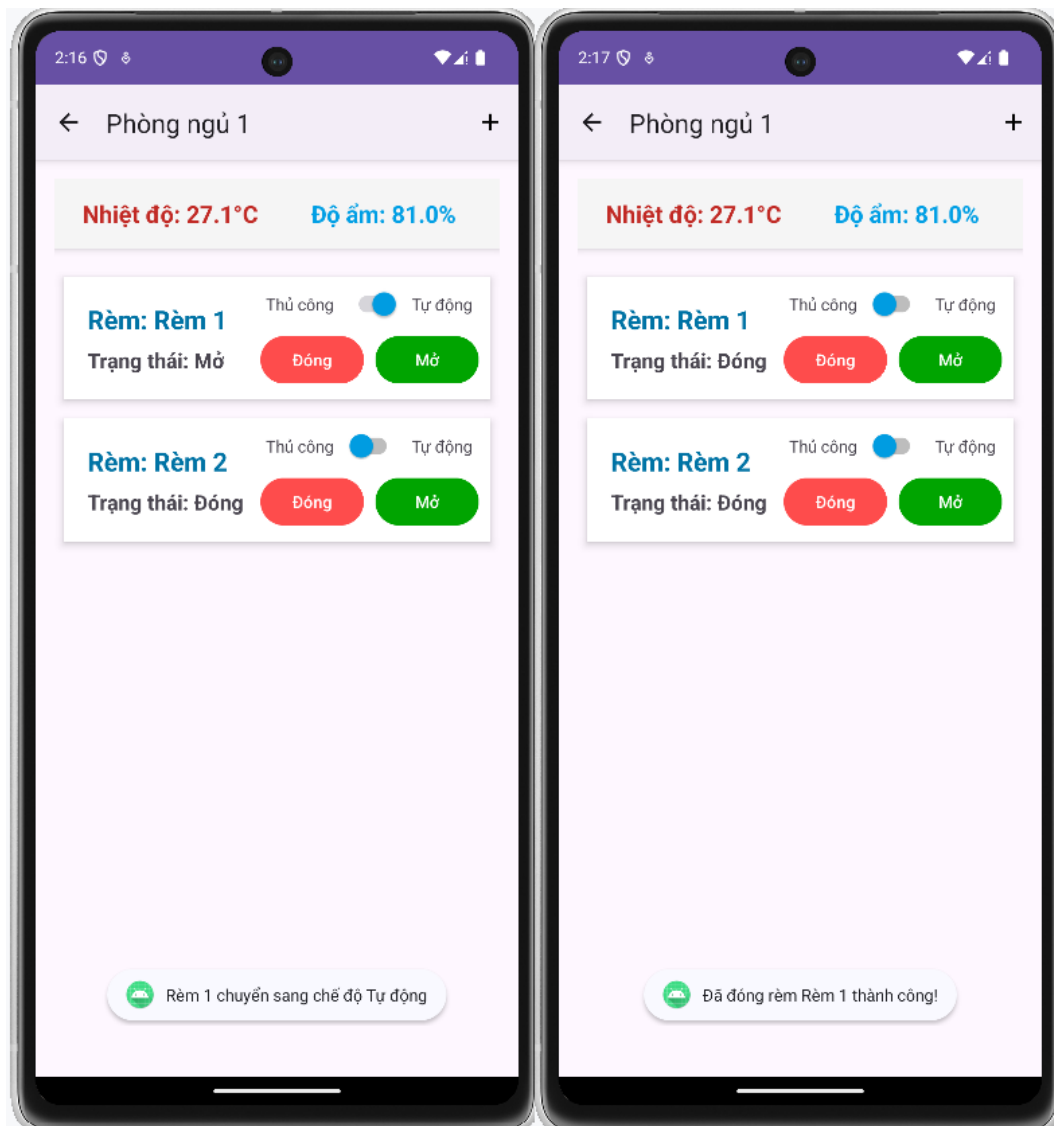
Hình 3.18 Giao diện chỉnh sửa tên nhà, phòng, thiết bị

Hình 3.18 biểu diễn chức năng đổi tên nhà, phòng, thiết bị. Khi nhấn giữ vào 1 trong các nhà, phòng, hoặc thiết bị, một menu gồm 2 lựa chọn "Đổi tên" và "Xóa" sẽ hiện lên, chọn "Đổi tên" để cập nhật lại tên. Khi đó, ứng dụng sẽ hiển thị một hộp thoại đổi tên với tên hiện tại của nhà/phòng/thiết bị tương ứng. Người dùng thực hiện đổi tên và chọn "Lưu", nếu tên để trống thì hệ thống sẽ thông báo và tắt hộp thoại, ngược lại thông báo thành công đồng thời cập nhật lại thông tin danh sách.



Hình 3.19 Giao diện xóa nhà, phòng, thiết bị

Chức năng xóa ở hình 3.19 cũng tương tự như hình 3.18. Thay vì chọn "Đổi tên" thì chọn "Xóa", khi đó một hộp thoại xác nhận xóa sẽ hiện lên, chọn "Xóa" để thực hiện xóa nhà, phòng, thiết bị tương ứng. Sau khi xóa ứng dụng sẽ hiển thị một thông báo xóa thành công và cập nhật lại giao diện danh sách.



Hình 3.20 Điều khiển thiết bị rèm

Hình 3.20 là chức năng điều khiển rèm. Để điều khiển thiết bị rèm, cần chọn một trong hai chế độ: Tự động và Thủ công. Ở chế độ tự động, rèm sẽ thay đổi trạng thái Đóng/Mở theo giá trị của cảm biến ánh sáng. Khi ánh sáng vượt mức cho phép, rèm sẽ đóng và ngược lại. Ở chế độ thủ công, người dùng sẽ dùng hai nút "Đóng" và "Mở" để điều khiển rèm. Khi người dùng điều khiển thì trạng thái sẽ được cập nhật ngay lập tức và hiển thị thông báo.

CHƯƠNG 4. KẾT LUẬN

Trong báo cáo này, em đã trình bày toàn bộ quá trình thiết kế và triển khai hệ thống rèm cửa thông minh sử dụng Firebase và ứng dụng Android viết bằng Kotlin. Dự án đã hoàn thành mục tiêu ban đầu là tạo ra một hệ thống tự động, giúp người dùng điều khiển rèm cửa từ xa thông qua ứng dụng và tích hợp cảm biến để tối ưu hóa việc sử dụng rèm cửa trong các điều kiện môi trường khác nhau.

Những kết quả mà đề án đã hoàn thành bao gồm:

- Xây dựng thành công kiến trúc hệ thống IoT với Firebase làm nền tảng lưu trữ và đồng bộ dữ liệu.
- Thiết kế ứng dụng Android với giao diện thân thiện, hỗ trợ điều khiển rèm cửa một cách tiện lợi.
- Tích hợp cảm biến nhiệt độ, độ ẩm, và ánh sáng, giúp rèm cửa hoạt động tự động dựa trên môi trường.

Tuy nhiên, vẫn còn một số hạn chế như hệ thống chưa hỗ trợ báo cáo chi tiết hoặc thống kê dữ liệu. Trong tương lai, em dự án sẽ tập trung vào tích hợp các tính năng tiên tiến như điều khiển bằng giọng nói, tối ưu hóa thuật toán, và tăng cường bảo mật dữ liệu.

Em xin gửi lời cảm ơn đến giảng viên hướng dẫn đồng hành và hỗ trợ em để hoàn thành dự án này. Đây là bước đệm quan trọng để phát triển các hệ thống nhà thông minh trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] N. Mishra and S. Pandya, “Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review,” *IEEE Access*, vol. 9, pp. 59 353–59 377, 2021.
- [2] ESPRESSIF. (2024) Esp32 series datasheet. Accessed: 2024-12-30. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [3] Cloudflare. (2024) What is https? Accessed: 2024-12-30. [Online]. Available: <https://www.cloudflare.com/learning/ssl/what-is-https/>
- [4] Mescu. (2023) Module điều khiển động cơ 1298n. Accessed: 2024-12-30. [Online]. Available: <https://mecsu.vn/ho-tro-ky-thuat/module-dieu-khien-dong-co-1298n.lGG>
- [5] NSHOPVN. (2024) Động cơ motor giảm tốc 12-24 rpm 12-24vdc. Accessed: 2024-12-30. [Online]. Available: <https://nshopvn.com/product/dong-co-giam-toc-20-rpm-12-24vdc/>
- [6] ——. (2024) Module cảm biến Độ Ẩm, nhiệt Độ dht11. Accessed: 2024-12-30. [Online]. Available: <https://nshopvn.com/product/module-cam-bien-do-am-nhiet-do-dht11/>
- [7] K. A. Licon. (2015) Introduction of firebase. Accessed: 2024-12-30. [Online]. Available: <https://viblo.asia/p/introduction-of-firebase-w5WQvzb1vk3E>