

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



ĐỒ ÁN ĐA NGÀNH HƯỚNG CÔNG NGHỆ PHẦN MỀM

HỆ THỐNG HỖ TRỢ NHÀ THÔNG MINH

Giảng viên phụ trách: Trần Thanh Bình

Sinh viên thực hiện: Nguyễn Tuấn Anh - 2012608
Nguyễn Phước Đắc Hùng - 2013369
Dương Văn Nghĩa - 2013864
Phạm Anh Minh - 2013782
Trương Mạnh Khôi - 2013536

TP. Hồ Chí Minh, Ngày 15 tháng 12 năm 2023

Mục lục

1	Giới thiệu đề tài	3
2	Những yêu cầu chức năng và phi chức năng.	4
2.1	Yêu cầu chức năng	4
2.2	Yêu cầu phi chức năng.	4
3	Tìm hiểu công nghệ	5
3.1	Giao thức MQTT	5
3.1.1	MQTT là gì ?	5
3.1.2	Kiến trúc mức cao của MQTT	5
3.1.3	Topic	6
3.1.4	MQTT Bridge:	6
3.2	IoT gateway	7
3.2.1	IoT gateway là gì?	7
3.2.2	Các chức năng của Iot gateway	7
3.3	ReactJs.	9
3.3.1	Giới thiệu	9
3.3.2	React là gì?	9
3.3.3	Tại sao sử dụng React?	10
3.4	Python	11
3.4.1	Python là gì?	11
3.4.2	Ưu điểm	11
3.4.3	Nhược điểm	12

4	Use Case	13
4.1	Usecase Diagram: Whole System	13
4.2	Use case 1: Đăng nhập.	13
4.3	Use case 2: Đăng xuất	14
4.4	Use case 3: Đăng ký người dùng.	15
4.5	Use case 4: Thêm thiết bị	15
4.6	Use case 5: Xóa thiết bị.	16
4.7	Use case 6: Xóa phòng.	16
4.8	Use case 7: Xem, sửa thông tin người dùng.	17
4.9	Use case 8: Xem lịch sử hoạt động	17
4.10	Use case 9: Điều khiển đèn bằng tay	18
4.11	Use case 10: Điều khiển đèn bằng ánh sáng.	18
4.12	Use case 11: Điều khiển đèn bằng giao diện.	19
4.13	Use case 12: Điều khiển quạt bằng tay	19
4.14	Use case 13: Điều khiển quạt bằng nhiệt độ	20
4.15	Use case 14: Điều khiển quạt bằng giao diện	20
4.16	Use case 15: Mở cửa bằng tay	20
4.17	Use case 16: Mở cửa bằng giao diện	21
4.18	Use case 17: Mở cửa bằng phát hiện người	21
4.19	Use case 18: Cảnh báo độ ẩm trong phòng	22
4.20	Use case 19: Yêu cầu hỗ trợ	22
5	Sơ đồ khối.	23
6	Sequence Diagram	24
6.1	Điều khiển ứng dụng thông qua Sensor.	24
6.2	Điều khiển ứng dụng thông qua App	24
7	Giao diện	25
7.1	Giao diện trên Website	25
7.2	Giao diện trên mobile	28
8	Tổng kết	34
8.1	Ưu điểm	34
8.2	Nhược điểm	34
8.3	Hướng phát triển của dự án.	34

1 Giới thiệu đề tài

Ngày nay, con người chúng ta đang sống trong thời đại mà khoa học - công nghệ phát triển chóng mặt. Khoa học - công nghệ giúp ích rất nhiều cho chúng ta, từ giải trí, học tập cho tới y học. Do đó, nhóm chúng tôi quyết định phát triển một hệ thống nhà thông minh để giúp con người có khả năng điều khiển các thiết bị trong nhà từ xa. Hệ thống này cũng có khả năng tự động điều khiển các thiết bị dựa trên dữ liệu môi trường mà nó thu thập được từ các cảm biến. Trong thời đại hiện nay, chúng ta không thể tránh khỏi sự lan truyền nhanh chóng của khoa học - công nghệ. Hệ thống nhà thông minh là một ví dụ tiêu biểu cho sự tiến bộ đáng kể trong lĩnh vực này. Được xem như một sự kết hợp hoàn hảo giữa khoa học và công nghệ, nó đã mang lại nhiều lợi ích cho cuộc sống hàng ngày của chúng ta.

Hệ thống nhà thông minh không chỉ giúp tiết kiệm năng lượng và tạo ra môi trường sống hiệu quả hơn mà còn cung cấp tiện ích và an ninh cho người sử dụng. Nó cho phép chúng ta kiểm soát ánh sáng, nhiệt độ, âm thanh và nhiều thiết bị khác trong ngôi nhà của mình thông qua điện thoại di động hoặc máy tính. Điều này không chỉ tạo ra sự thuận tiện mà còn giúp tiết kiệm thời gian và tạo ra môi trường sống thông minh, tiên tiến.

Với sự phát triển liên tục của khoa học - công nghệ, hệ thống nhà thông minh hứa hẹn sẽ tiếp tục cung cấp nhiều giải pháp thông minh và tiện ích cho cuộc sống của con người trong tương lai.



Hình 1: Ví dụ đơn giản về hệ thống nhà thông minh

2 Những yêu cầu chức năng và phi chức năng

2.1 Yêu cầu chức năng

- Điều khiển mở/đóng cửa: Hệ thống nhà thông minh IoT có thể được cài đặt để mở/đóng các cửa thông qua ứng dụng
- Hệ thống hiển thị nhiệt độ: Hệ thống nhà thông minh IoT có thể tự động hiển thị nhiệt độ của phòng thông qua cảm biến nhiệt độ.
- Điều khiển bật/tắt đèn: Hệ thống nhà thông minh IoT có thể được cài đặt để bật/tắt đèn thông qua ứng dụng, tự động thông qua các cảm biến khoảng cách và ánh sáng.
- Điều khiển bật/tắt quạt: Hệ thống nhà thông minh IoT có thể bật/tắt quạt thông qua ứng dụng
- Hiển thị thông tin độ ẩm và ánh sáng: Hệ thống nhà thông minh sẽ hiển thị thông tin độ ẩm và ánh sáng của căn phòng thông qua các cảm biến
- Điều khiển bật/tắt thiết bị theo thời gian đặt trước

2.2 Yêu cầu phi chức năng

- Hoạt động ổn định app trên android/ios
- Bảo mật thông tin người dùng
- Đầy đủ cơ chế đảm bảo quyền truy cập
- Đáp ứng được số lượng thiết bị kết nối > 100 thiết bị đồng thời và đáp ứng được lưu lượng dữ liệu >100mb từ các thiết bị này
- Thời gian phản hồi <50ms
- Đơn vị đo nhiệt độ là oC, đơn vị đo độ ẩm là %, đơn vị đo cường độ ánh sáng là %

3 Tìm hiểu công nghệ

3.1 Giao thức MQTT

3.1.1 MQTT là gì ?

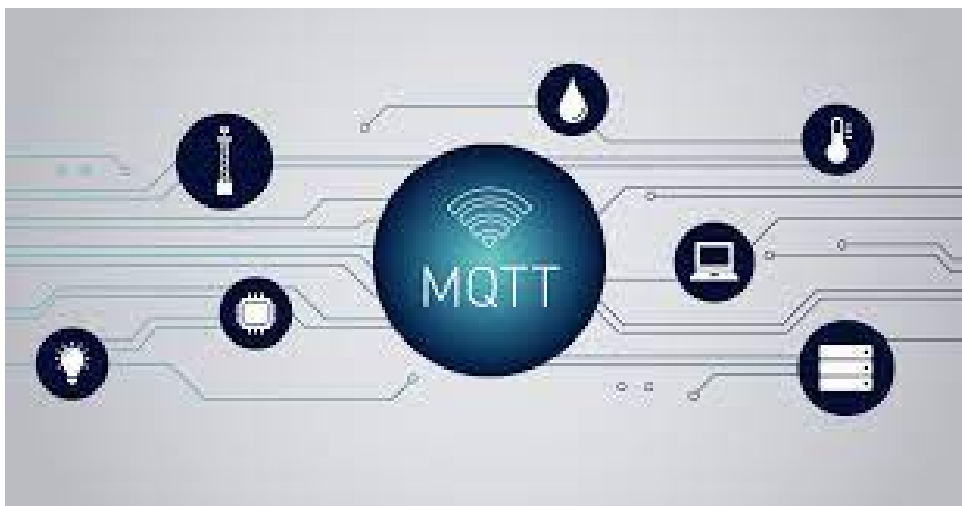
MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông điệp (message) theo mô hình publish/subscribe, sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

3.1.2 Kiến trúc mức cao của MQTT

Kiến trúc mức cao (high-level) của MQTT gồm 2 phần chính là Broker (MQTT server) và Clients (devices/software components):

- Broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ client. Nhiệm vụ chính của broker là nhận message từ publisher, xếp các message theo hàng đợi rồi chuyển chúng tới một địa chỉ cụ thể.
- Client thì được chia thành 2 nhóm là publisher và subscriber. Client chỉ làm ít nhất một trong 2 việc là xuất (publish) các message lên một topic cụ thể hoặc đăng ký (subscribe) một topic nào đó để nhận message từ topic này

Mỗi client sẽ đăng ký theo dõi các kênh thông tin (topic) hoặc gửi dữ liệu lên kênh thông tin đó. Quá trình đăng ký này gọi là “subscribe” và hành động một client gửi dữ liệu lên kênh thông tin được gọi là “publish”. Mỗi khi kênh thông tin đó được cập nhật dữ liệu (dữ liệu này có thể đến từ các client khác) thì những client nào đã đăng ký theo dõi kênh này sẽ nhận được dữ liệu cập nhật đó



Hình 2: Ví dụ đơn giản về hệ thống nhà thông minh

3.1.3 Topic

Topic có thể coi như một "đường truyền" logic giữa 2 điểm là publisher và subscriber. Về cơ bản, khi message được publish vào một topic thì tất cả những subscriber của topic đó sẽ nhận được message này.

Giao thức MQTT cho phép khai báo các topic kiểu phân cấp.

Giả sử chúng ta có một hệ thống cảm biến đo thông tin môi trường trong khu vườn của chúng ta. Như vậy, các topic phục vụ truyền tải thông tin môi trường cho ngôi nhà của chúng ta có thể được khai báo như sau:

- home/temperatre: topic thông tin nhiệt độ khu vườn
- home/humidity: topic thông tin độ ẩm khu vườn
- home/light: topic thông tin độ sáng khu vườn

3.1.4 MQTT Bridge:

MQTT Bridge là một tính năng của MQTT Broker cho phép các MQTT Broker có thể kết nối và trao đổi dữ liệu với nhau. Để sử dụng tính năng này, ta cần tối thiểu 2 Broker, trong đó, một Broker bất kỳ sẽ được cấu hình thành Bridge. Khi cấu hình MQTT bridge, ta cần lưu ý tới các thông số sau:

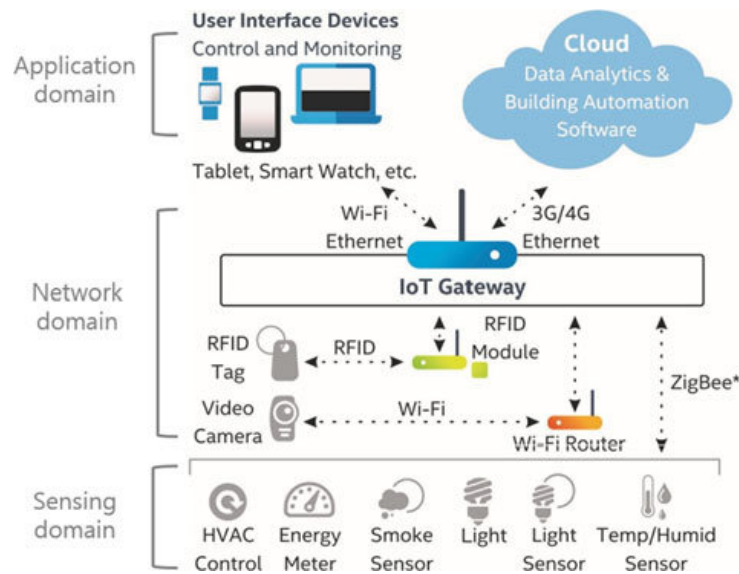
- address: địa chỉ của broker cần kết nối
- bridge-protocol-version: phiên bản của giao thức MQTT đang sử dụng chung cho 2 broker.
- topic: phần này định nghĩa 3 thông số: tên topic được trao đổi giữa 2 broker, chiều trao đổi (1 chiều hay 2 chiều) và topic mapping giữa 2 broker.

3.2 IoT gateway

3.2.1 IoT gateway là gì?

IoT gateway là một máy tính nhúng làm cầu nối kết nối giữa các cảm biến (sensors), tác nhân (actors) tới mạng diện rộng Internet hoặc mạng nội bộ Intranet.

Một số công nghệ truyền thông được sử dụng bởi các cảm biến: Bluetooth, WiFi, BLE, Zigbee, Z-Wave, 6LoWPAN, NFC, WiFi Direct, GSM, LTE, LoRa, NB-IoT, LTE-M, ...



Hình 3: Ví dụ đơn giản về hệ thống nhà thông minh

Một số cảm biến có thể tạo ra hàng chục ngàn điểm dữ liệu mỗi giây. IoT gateway cung cấp một nơi để xử lý trước dữ liệu cục bộ ở vùng biên trước khi nó được gửi lên đám mây. Khi dữ liệu được tổng hợp, thu gọn và phân tích một cách khôn ngoan ở vùng biên, nó sẽ giảm thiểu khối lượng dữ liệu cần chuyển tiếp lên đám mây, điều này có thể tác động lớn đến thời gian hồi đáp và chi phí đường truyền mạng.

Một lợi ích khác của cổng IoT là nó có thể cung cấp cơ chế bảo mật bổ sung cho mạng IoT và dữ liệu mà nó vận chuyển. Vì gateway quản lý thông tin di chuyển theo cả hai chiều, nó có thể bảo vệ dữ liệu khi di chuyển lên đám mây khỏi các rò rỉ, hạn chế các thiết bị IoT bị xâm phạm bởi nguồn tấn công bên ngoài với các tính năng như phát hiện giả mạo, mã hóa, tạo số ngẫu nhiên bằng phần cứng và công cụ mã hóa.

3.2.2 Các chức năng của Iot gateway

Một IoT Gateway đa năng có thể thực hiện rất nhiều các nhiệm vụ, một số ví dụ như:

- Tạo điều kiện giao tiếp với các thiết bị cũ hoặc không có kết nối internet.
- Bộ nhớ đệm dữ liệu, caching và media streaming.
- Xử lý trước dữ liệu (data pre-processing), làm sạch, lọc và tối ưu hóa.
- Tổng hợp dữ liệu thô.

- Liên lạc Device-to-Device / Machine-to-Machine.
- Tính năng kết nối mạng và lưu trữ dữ liệu trực tiếp.
- Trực quan hóa dữ liệu và phân tích dữ liệu cơ bản thông qua các ứng dụng IoT Gateway.
- Tính năng lịch sử dữ liệu ngắn hạn.
- Bảo mật: quản lý truy cập người dùng và các tính năng bảo mật mạng.
- Quản lý cấu hình thiết bị.
- Chẩn đoán hệ thống.

3.3 ReactJs

3.3.1 Giới thiệu

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javascript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn

3.3.2 React là gì?

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và www.instagram.com được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM. Trước khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

- **Virtual DOM:** công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật. React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.
- **JSX:** JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng

- **Components:** React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.
- **Props và State:**
 - Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.
 - State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.

3.3.3 Tại sao sử dụng React?

Dễ sử dụng

React là một thư viện GUI nguồn mở JavaScript tập trung vào một điều cụ thể; hoàn thành nhiệm vụ UI hiệu quả. Nó được phân loại thành kiểu “V” trong mô hình MVC (Model-View-Controller).

Là lập trình viên JavaScript, bạn sẽ dễ dàng hiểu được những điều cơ bản về React. Bạn thậm chí có thể bắt đầu phát triển các ứng dụng dựa trên web bằng cách sử dụng react chỉ trong vài ngày.

Để củng cố hiểu biết của mình, bạn hãy thử khám phá thêm nhiều hướng dẫn về React. Chúng mang đến nhiều thông tin về cách sử dụng công cụ: videos, hướng dẫn và dữ liệu làm phong phú góc nhìn của bạn.

Nó hỗ trợ Reusable Component trong Java

React cho phép bạn sử dụng lại components đã được phát triển thành các ứng dụng khác có cùng chức năng. Tính năng tái sử dụng component là một lợi thế khác biệt cho các lập trình viên.

Viết component dễ dàng hơn

React component dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép bạn kết hợp HTML với JavaScript.

JSX là một sự pha trộn tuyệt vời của JavaScript và HTML. Nó làm rõ toàn bộ quá trình viết cấu trúc trang web. Ngoài ra, phần mở rộng cũng giúp render nhiều lựa chọn dễ dàng hơn.

SX có thể không là phần mở rộng cú pháp phổ biến nhất, nhưng nó được chứng minh là hiệu quả trong việc phát triển components đặc biệt hoặc các ứng dụng có khối lượng lớn.

Hiệu suất tốt hơn với Virtual DOM

React sẽ cập nhật hiệu quả quá trình DOM (Document Object Model – Mô hình đối tượng tài liệu). Như bạn có thể biết, quá trình này có thể gây ra nhiều thất vọng trong các dự án ứng dụng dựa trên web. May mắn là React sử dụng virtual DOMs, vì vậy bạn có thể tránh được vấn đề này.

Công cụ cho phép bạn xây dựng các virtual DOMs và host chúng trong bộ nhớ. Nhờ vậy, mỗi khi có sự thay đổi trong DOM thực tế, thì virtual sẽ thay đổi ngay lập tức

Hệ thống này sẽ ngăn DOM thực tế để buộc các bản cập nhật được liên tục. Do đó, tốc độ của ứng dụng sẽ không bị gián đoạn.

Thân thiện với SEO

React cho phép bạn tạo giao diện người dùng có thể được truy cập trên các công cụ tìm kiếm khác nhau. Tính năng này là một lợi thế rất lớn vì không phải tất cả các khung JavaScript đều thân thiện với SEO.

Ngoài ra, vì React có thể tăng tốc quá trình của ứng dụng nên có thể cải thiện kết quả SEO. Cuối cùng tốc độ web đóng một vai trò quan trọng trong tối ưu hóa SEO.

Tuy nhiên, bạn cần lưu ý rằng React chỉ là một thư viện JavaScript. Nghĩa là nó không thể tự làm mọi thứ. Sử dụng các thư viện bổ sung có thể cần thiết cho các mục tiêu quản lý, định tuyến và tương tác.

3.4 Python

3.4.1 Python là gì?

Là một ngôn ngữ lập trình mã nguồn mở đa mục đích, một loại ngôn ngữ bậc cao, thông dịch và hướng đối tượng. Đây là một ngôn ngữ lập trình thông dịch được tạo ra bởi Guido van Rossum vào năm 1990.

Điểm đặc biệt của ngôn ngữ Lập trình Python có cú pháp khá dễ hiểu, dễ đọc và dễ học. Trong việc phát triển ứng dụng thì ngôn ngữ này cũng rất linh hoạt. Python hỗ trợ mẫu đa lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và mệnh lệnh hoặc là các phong cách lập trình theo thủ tục. Python là ngôn ngữ lập trình động nên không cần sử dụng các kiểu dữ liệu khai báo.

3.4.2 Ưu điểm

- **Đơn giản:** Là một ngôn ngữ có hình thức sáng sủa, cấu trúc rõ ràng, cú pháp ngắn gọn giúp người lập trình dễ dàng đọc và tìm hiểu.

- **Tốc độ xử lý khá nhanh:** được đánh giá nhanh hơn so với ngôn ngữ PHP. Với tốc độ xử lý cực nhanh, Python có thể tạo ra những chương trình từ những script siêu nhỏ tới những phần mềm cực lớn như Blender 3D. ... - **Chất lượng:** Thư viện có tiêu chuẩn cao, Python có khối cơ sở dữ liệu khá lớn nhằm cung cấp giao diện cho tất cả các CSDL thương mại.

- **Thuận tiện:** Python được biên dịch và chạy trên tất cả các nền tảng lớn. Nó có trên tất cả các nền tảng hệ điều hành từ UNIX, MS – DOS, Mac OS, Windows và Linux và các OS khác thuộc họ Unix. Tương thích mạnh mẽ với Unix, hardware, third-party software với số lượng thư viện khổng lồ (400 triệu người sử dụng)

- **Mở rộng:** Với tính năng này, Python cho phép người lập trình có thể thêm hoặc tùy chỉnh các công cụ nhằm tối đa hiệu quả có thể đạt được trong công việc.

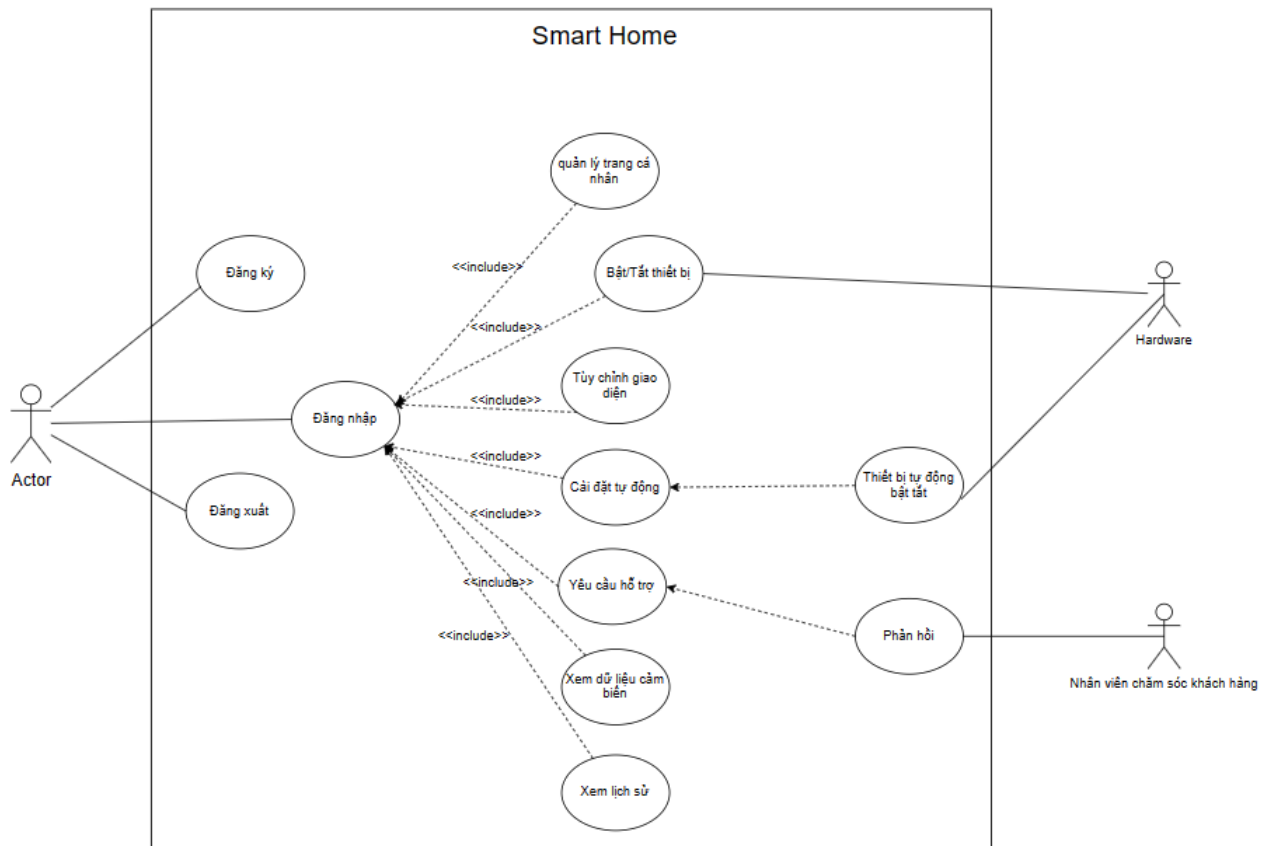
- **GUI Programming:** Giúp cho việc thực hiện ảnh minh họa di động một cách tự nhiên và sống động hơn.

3.4.3 Nhược điểm

- Python không có các thuộc tính như :protected,private hay public, không có vòng lặp do ... while và switch ... case.
- Mặc dù tốc độ xử lý của Python nhanh hơn PHP nhưng không bằng Java và C++.

4 Use Case

4.1 Usecase Diagram: Whole System



Hình 4: Whole System

4.2 Use case 1: Đăng nhập

Usecase name	Đăng nhập
Actor	Người dùng
Description	Người dùng đăng nhập tài khoản cá nhân
Pre-conditions	Người dùng có tài khoản
Normal flow	<ol style="list-style-type: none"> 1. Người dùng truy cập trang đăng nhập tài khoản trên ứng dụng. 2. Nhập tài khoản, mật khẩu cá nhân vào các trường thông tin hợp lý trong trang đăng nhập. 3. Hệ thống xác nhận thông tin đăng nhập. 4. Đăng nhập thành công, người dùng truy cập vào giao diện trang chủ
Alternative Flow	Không.

Exception	Exception 1: Tại bước 3: 3.1 Nếu người dùng đăng nhập sai tài khoản, mật khẩu thì hệ thống thông báo “Tài khoản hoặc mật khẩu của bạn không chính xác” và quay trở lại bước 2.
------------------	---

4.3 Use case 2: Đăng xuất

Usecase name	Đăng xuất
Actor	Người dùng
Description	Người dùng đăng xuất tài khoản cá nhân ra khỏi hệ thống.
Pre-conditions	Người dùng đã đăng nhập tài khoản trên hệ thống
Normal flow	1. Người dùng truy cập vào tranh chủ của ứng dụng. 2. Người dùng nhấn vào nút đăng xuất. 3. Hệ thống đăng xuất người dùng ra khỏi hệ thống và chuyển người dùng đến lại trang đăng nhập ứng dụng.
Alternative Flow	Không.
Exception	Không.

4.4 Use case 3: Đăng ký người dùng

Usecase name	Đăng ký người dùng
Actor	Người dùng
Description	Người dùng đăng ký tài khoản.
Pre-conditions	Người dùng có tài khoản email hay số điện thoại.
Normal flow	<ol style="list-style-type: none">1. Người dùng truy cập đến trang đăng nhập.2. Người dùng chọn vào đăng ký tài khoản.3. Người dùng nhập tên, mật khẩu, nhập lại mật khẩu và nhấn đăng ký.4. Hệ thống sẽ kiểm tra tính hợp lệ của thông tin đăng ký.5. Hệ thống chuyển người dùng đến trang thêm thông tin.6. Người dùng nhập thông tin theo trường tương ứng.7. Hệ thống gửi mail chứa mã xác nhận đến mail mà người dùng cung cấp.8. Người dùng nhập mã xác nhận đã nhận được và tích chọn đăng ký.9. Hệ thống chuyển người dùng trở về lại trang đăng nhập.
Alternative Flow	Không.
Exception	<p>Exception 1: tại bước 4</p> <ol style="list-style-type: none">4.1. Nếu mail đã có rồi hay mật khẩu nhập sai thì hệ thống hiện lên thông báo “Email này đã được đăng ký rồi hoặc thông tin đăng ký không đúng yêu cầu”.4.2. Hệ thống chuyển người dùng lại về trang đăng ký. <p>Exception 2: tại bước 6</p> <ol style="list-style-type: none">6.1. Nếu người dùng nhập thiếu thông tin thì hệ thống sẽ hiển thị thông báo “Bạn chưa nhập đủ thông tin yêu cầu”.

4.5 Use case 4: Thêm thiết bị

Usecase name	Thêm thiết bị
Actor	Người dùng
Description	Người dùng có thể thêm 1 thiết bị thông minh vào phòng ở mà mình muốn.
Pre-conditions	Người dùng đã đăng nhập vào hệ thống.
Normal flow	<ol style="list-style-type: none">1. Người dùng chọn phòng ở thanh nav bar bên trái.2. Người dùng dấu cộng để có thể thêm thiết bị.3. Người dùng chọn loại thiết bị muốn thêm vào.4. Người dùng thêm thông tin cho thiết bị mới thêm.5. Hệ thống sẽ cập nhật thiết bị mới thêm vào trong ứng dụng của người dùng.
Alternative Flow	Alternative 1: Tại bước 1:

	<p>1.1. Nếu người dùng chưa có phòng muốn thêm thì người dùng có thể nhấn dấu cộng ở bên thanh nav bar để thêm phòng.</p> <p>1.2. Người dùng cập nhật thông tin của căn phòng mình mới tạo.</p> <p>1.3. Hệ thống sẽ cập nhật phòng mà người dùng mới tạo và trở lại trang chủ.</p> <p>1.4. Quay lại bước 1</p>
Exception	<p>Exception 1: Tại bước 4</p> <p>4.1. Nếu người dùng thêm thông tin trùng hoàn toàn với 1 thiết bị khác đã thêm trước đó thì hệ thống thông báo “Thiết bị này đã được thêm trước đó”.</p>

4.6 Use case 5: Xóa thiết bị

Usecase name	Xóa thiết bị
Actor	Người dùng
Description	Người dùng xóa thiết bị trên hệ thống.
Pre-conditions	Người dùng đã truy cập vào hệ thống và hệ thống cá nhân của người dùng đã có thiết bị được thêm trước đó rồi.
Normal flow	<p>1. Người dùng chọn phòng có chứa thiết bị mình muốn xóa.</p> <p>2. Người dùng nhấn vào thiết bị.</p> <p>3. Chọn vào nút delete để xóa thiết bị.</p>
Alternative Flow	Không.
Exception	Không.

4.7 Use case 6: Xóa phòng

Usecase name	Xóa phòng
Actor	Người dùng
Description	Người dùng muốn xóa phòng trên hệ thống.
Pre-conditions	Người dùng đã truy cập vào hệ thống và đã có phòng được thêm trước đó rồi.
Normal flow	<p>1. Người dùng chọn vào phòng mà mình muốn xóa.</p> <p>2. Người dùng chọn delete để xóa phòng.</p> <p>3. Hệ thống sẽ xóa dữ liệu của toàn bộ phòng đó.</p>
Alternative Flow	Không.
Exception	Exception 1: tại bước 2

	<p>2.1. Nếu phòng đó vẫn còn thiết bị bên trong thì hệ thống sẽ thông báo “Phòng này có thiết bị bên trong, bạn có muốn xóa luôn không?”.</p> <p>2.2. Chọn “Có”, quay lại bước 3, Chọn “Không” thì hệ thống tắt thông báo và quay về trang phòng đó. Không.</p>
--	---

4.8 Use case 7: Xem, sửa thông tin người dùng

Usecase name	Xem, sửa thông tin người dùng
Actor	Người dùng
Description	Người dùng xem, sửa thông tin cá nhân của mình.
Pre-conditions	Người dùng đã đăng nhập vào hệ thống.
Normal flow	<ol style="list-style-type: none">1. Người dùng truy cập vào trang chủ của hệ thống.2. Người dùng chọn logo cá nhân bên phải góc trên.3. Hệ thống hiển thị thông tin của người dùng.4. Người dùng sửa thông tin cá nhân của mình tại các trường thông tin tương ứng muốn sửa.5. Người dùng nhấn chọn “Save” để lưu thông tin cá nhân mới của mình.6. Hệ thống sẽ cập nhật lại thông tin cá nhân mới của người dùng và hiển thị lại trang thông tin cá nhân.
Alternative Flow	<p>Alternative 1: tại bước 3</p> <p>3.1. Nếu người dùng chỉ muốn xem thông tin cá nhân thì sau khi xem xong có thể nhấn chọn logo của hệ thống để thoát khỏi trang thông tin cá nhân.</p>
Exception	<p>Exception 1: tại bước 5</p> <p>5.1. Nếu người dùng xóa dữ liệu tại 1 trường thông tin và để trống (Không cập nhật gì mới) thì hệ thống hiện lên thông báo “Bạn chưa điền đủ thông tin”.</p> <p>5.2. Người dùng chọn “Oke” để quay lại điền thông tin còn thiếu.</p> <p>5.3. Quay lại bước 5.</p>

4.9 Use case 8: Xem lịch sử hoạt động

Usecase name	Xem lịch sử hoạt động
Actor	Người dùng
Description	Người dùng xem lịch sử hoạt động của các thiết bị trong khoảng thời gian. Bao gồm thời gian và trạng thái hoạt động.

Pre-conditions	Người dùng đã đăng nhập vào ứng dụng và có quyền truy cập các thiết bị.
Normal flow	<ol style="list-style-type: none">1. Người dùng truy cập vào trang chủ của hệ thống.2. Người dùng nhấn chọn xem lịch sử hoạt động.3. Hệ thống sẽ hiển thị lựa chọn xem lịch sử thiết bị hoặc người dùng.4. Người dùng chọn xem lịch sử thiết bị.5. Hệ thống hiển thị danh sách hoạt động của các thiết bị.
Alternative Flow	Alternative 1: tại bước 3 <ol style="list-style-type: none">3.1. Người dùng chọn xem lịch sử của người dùng.3.2. Hệ thống sẽ hiển thị danh sách hoạt động của người dùng.
Exception	Exception 1: tại bước 5 <ol style="list-style-type: none">5.1 Nếu người dùng chưa thêm thiết bị nào vào thì hệ thống sẽ hiển thị không hiển thị danh sách hoạt động mà là dòng chữ “Không có thông tin”

4.10 Use case 9: Điều khiển đèn bằng tay

Usecase name	Điều khiển đèn bằng tay
Actor	Phần cứng
Description	Người dùng điều khiển đèn bằng tay và hệ thống sẽ cập nhật trạng thái hoạt động của nó.
Pre-conditions	Bóng đèn đã được thêm vào danh sách thiết bị trên hệ thống.
Normal flow	<ol style="list-style-type: none">1. Người dùng bật hoặc tắt bóng đèn bằng công tắc bằng tay.2. Hệ thống sẽ cập nhật lại trạng thái bật tắt của bóng đèn.
Alternative Flow	Không.
Exception	Không.

4.11 Use case 10: Điều khiển đèn bằng ánh sáng

Usecase name	Điều khiển đèn bằng ánh sáng
Actor	Phần cứng
Description	Cảm biến ánh sáng sẽ đo ánh sáng ngoài trời để biết ngày hay đêm để bật tắt đèn trên hệ thống.
Pre-conditions	Bóng đèn đã được thêm vào danh sách thiết bị trên hệ thống.
Normal flow	<ol style="list-style-type: none">1. Cảm biến ánh sáng đo ánh sáng bên ngoài để nhận biết ngày hay đêm.

	2. Nếu ngày thì toàn bộ đèn sẽ được tắt đi, ngược lại nếu đêm thì đèn sẽ được bật lên. 3. Hệ thống cập nhật trạng thái của bật/tắt của bóng đèn.
Alternative Flow	Không.
Exception	Không.

4.12 Use case 11: Điều khiển đèn bằng giao diện

Usecase name	Điều khiển đèn bằng giao diện
Actor	Người dùng
Description	Người dùng điều khiển đèn thông qua giao diện.
Pre-conditions	Người dùng đã đăng nhập vào hệ thống.
Normal flow	1. Người dùng truy cập vào trang chủ. 2. Người dùng chọn phòng chứa đèn mình muốn bật. 3. Người dùng bật/tắt trạng thái của bóng đèn. 4. Hệ thống sẽ cập nhật lại trạng thái của bóng đèn.
Alternative Flow	Không.
Exception	Không.

4.13 Use case 12: Điều khiển quạt bằng tay

Usecase name	Điều khiển quạt bằng tay
Actor	Phần cứng
Description	Người dùng điều khiển quạt bằng tay.
Pre-conditions	Chiếc quạt đó đã được thêm vào trong hệ thống.
Normal flow	1. Người dùng bật/tắt quạt bằng tay. 2. Hệ thống sẽ cập nhật lại trạng thái của quạt.
Alternative Flow	Không.
Exception	Không.

4.14 Use case 13: Điều khiển quạt bằng nhiệt độ

Usecase name	Điều khiển quạt bằng nhiệt độ
Actor	Phần cứng
Description	Hệ thống sẽ bật/tắt quạt tùy vào bảng nhiệt động mà người dùng đã cài đặt trước.
Pre-conditions	Chiết quạt đó đã được thêm vào hệ thống và có thiết bị đo nhiệt độ căn phòng.
Normal flow	1. Thiết bị đo nhiệt độ căn phòng. 2. Nếu nhiệt độ lớn hơn mức cài đặt trước thì quạt sẽ bật để làm giảm nhiệt độ trong phòng, ngược lại nếu thấp hơn hoặc bằng thì quạt sẽ tắt. 3. Hệ thống sẽ cập nhật lại trạng thái của quạt.
Alternative Flow	Không.
Exception	Không.

4.15 Use case 14: Điều khiển quạt bằng giao diện

Usecase name	Điều khiển quạt bằng giao diện
Actor	Người dùng
Description	Người dùng bật/tắt quạt thông qua giao diện người dùng.
Pre-conditions	Người dùng đã đăng nhập vào hệ thống và có thiết bị quạt muốn bật/tắt.
Normal flow	1. Người dùng truy cập vào trang chủ hệ thống. 2. Người dùng chọn phòng có chiếc quạt mà mình muốn bật/tắt. 3. Người dùng có thể bật/tắt thiết bị. 4. Hệ thống cập nhật lại trạng thái của thiết bị.
Alternative Flow	Không.
Exception	Không.

4.16 Use case 15: Mở cửa bằng tay

Usecase name	Mở cửa bằng tay
Actor	Phần cứng
Description	Người dùng mở cửa bằng tay
Pre-conditions	Cánh cửa đã được thêm vào hệ thống.
Normal flow	1. Người dùng có thể mở/đóng cửa bằng tay. 2. Hệ thống cập nhật trạng thái của cánh cửa.

Alternative Flow	Không.
Exception	Không.

4.17 Use case 16: Mở cửa bằng giao diện

Usecase name	Mở cửa bằng giao diện
Actor	Người dùng
Description	Người dùng mở cửa bằng giao diện người dùng.
Pre-conditions	Người dùng đã đăng nhập vào hệ thống và cánh cửa đó cũng đã được thêm vào hệ thống.
Normal flow	<ol style="list-style-type: none">1. Người dùng truy cập trang chủ.2. Người dùng chọn phòng có cánh cửa mình muốn mở.3. Người dùng thay đổi trạng thái đóng/mở của cánh cửa.4. Hệ thống cập nhật trạng thái của cánh cửa.
Alternative Flow	Không.
Exception	Không.

4.18 Use case 17: Mở cửa bằng phát hiện người

Usecase name	Mở cửa bằng phát hiện người
Actor	Phần cứng
Description	Cảm biến phát hiện người dùng để phát hiện người đứng trước cửa và tự động mở nó ra.
Pre-conditions	Cánh cửa và cảm biến đã được thêm vào hệ thống.
Normal flow	<ol style="list-style-type: none">1. Nếu có người đứng trước cửa (thông qua cảm biến phát hiện người) thì cửa sẽ tự động mở.2. Hệ thống sẽ cập nhật trạng thái của cánh cửa.
Alternative Flow	Không.
Exception	Exception 1: tại bước 1 <ol style="list-style-type: none">1.1. Nếu cảm biến không còn phát hiện ai đứng trước cửa nữa thì sẽ đóng cửa lại.1.2. Quay lại bước 2.

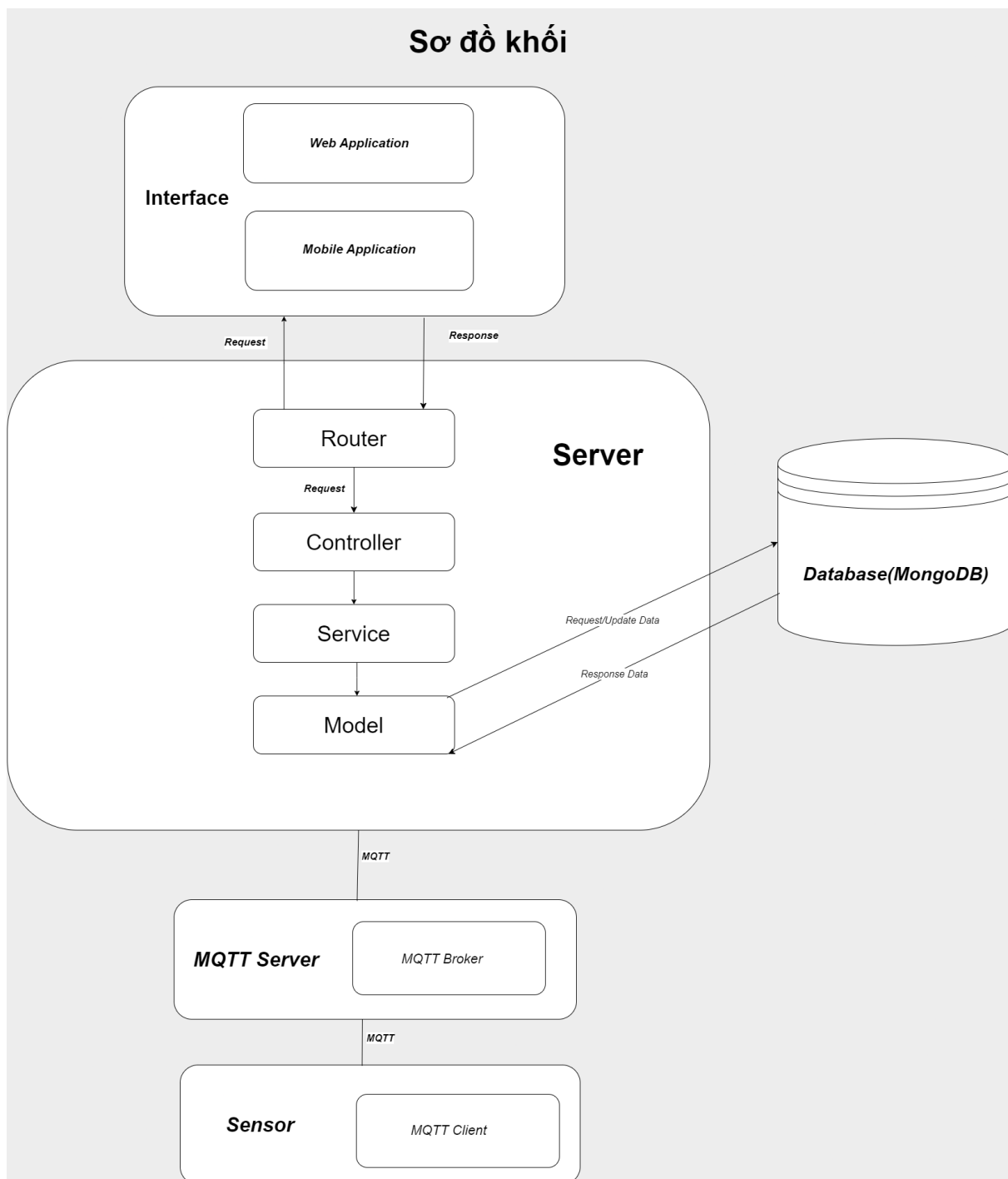
4.19 Use case 18: Cảnh báo độ ẩm trong phòng

Usecase name	Cảnh báo độ ẩm trong phòng
Actor	Phần cứng
Description	Hệ thống đo độ ẩm trong phòng, nếu nó lớn hơn mức đã cài đặt trước thì sẽ thông báo cho người dùng.
Pre-conditions	Thiết bị đo độ ẩm đã được thêm vào hệ thống.
Normal flow	1. Thiết bị đo độ ẩm trong phòng. 2. Nếu độ ẩm đo được lớn hơn mức cài đặt trước thì hệ thống sẽ gửi thông báo đến người dùng rằng độ ẩm tăng quá mức cho phép.
Alternative Flow	Không.
Exception	Exception 1: tại bước 2 2.1 Nếu độ ẩm giảm xuống mức cho phép thì hệ thống sẽ gửi thông báo cho người dùng rằng độ ẩm đã giảm xuống mức cho phép.

4.20 Use case 19: Yêu cầu hỗ trợ

Usecase name	Yêu cầu hỗ trợ
Actor	Nhân viên chăm sóc khách hàng
Description	Nhân viên hỗ trợ sẽ hỗ trợ cho người dùng.
Pre-conditions	_____
Normal flow	1. _____ 2. _____
Alternative Flow	Không.
Exception	Không.

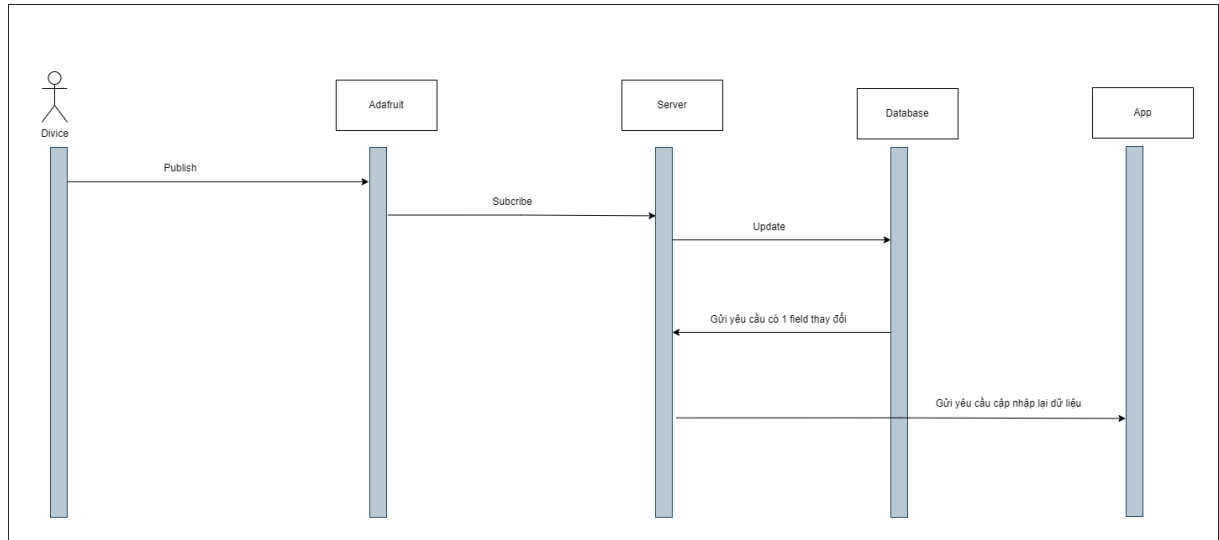
5 Sơ đồ khối



Hình 5: Sơ đồ khối

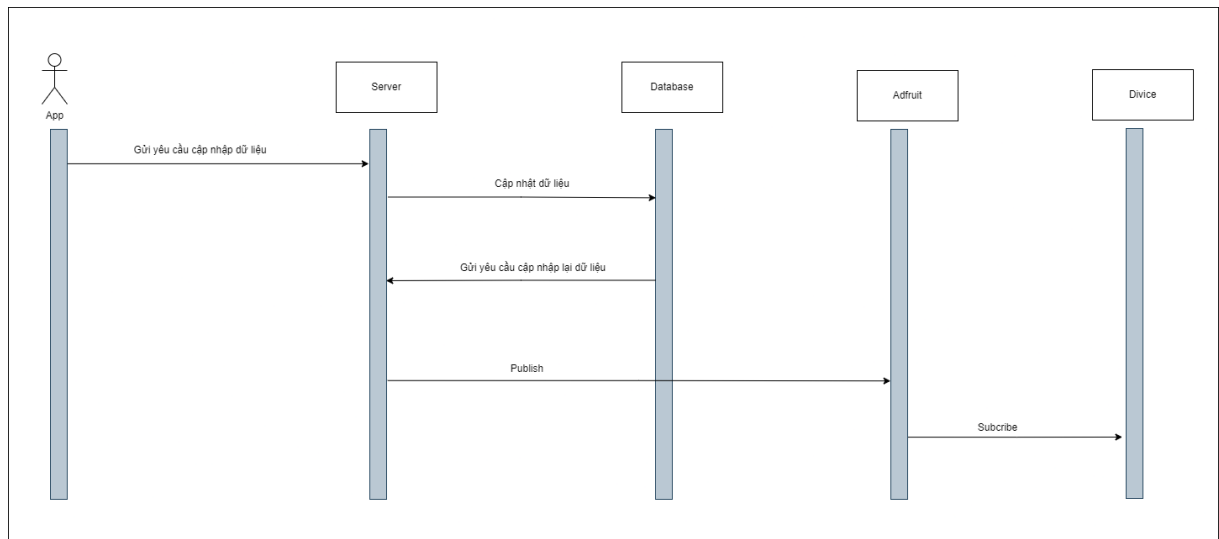
6 Sequence Diagram

6.1 Điều khiển ứng dụng thông qua Sensor



Hình 6: Sequence diagram: Điều khiển ứng dụng thông qua Sensor

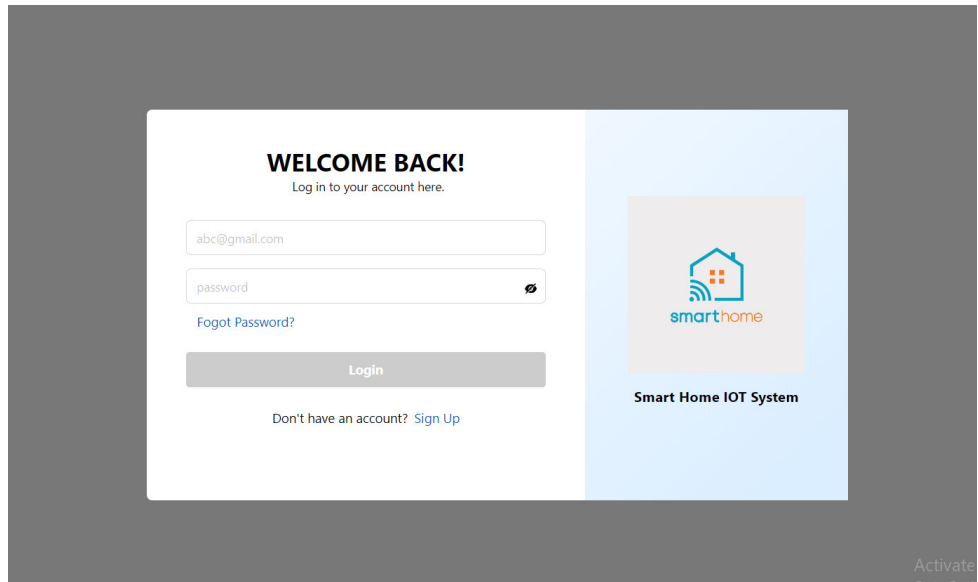
6.2 Điều khiển ứng dụng thông qua App



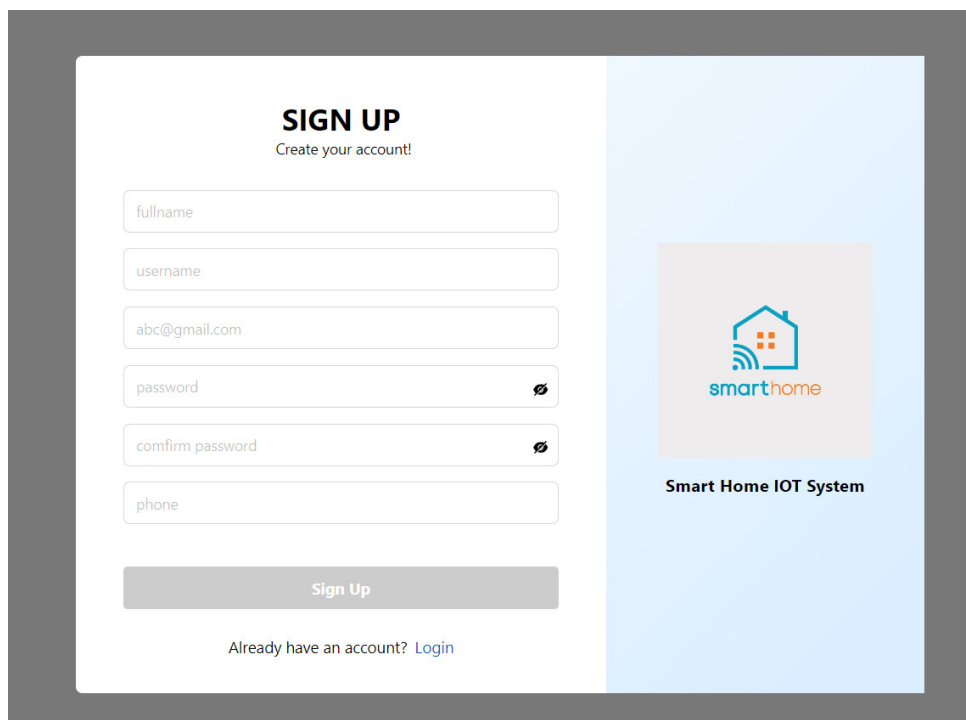
Hình 7: Sequence diagram: Điều khiển ứng dụng thông qua App

7 Giao diện

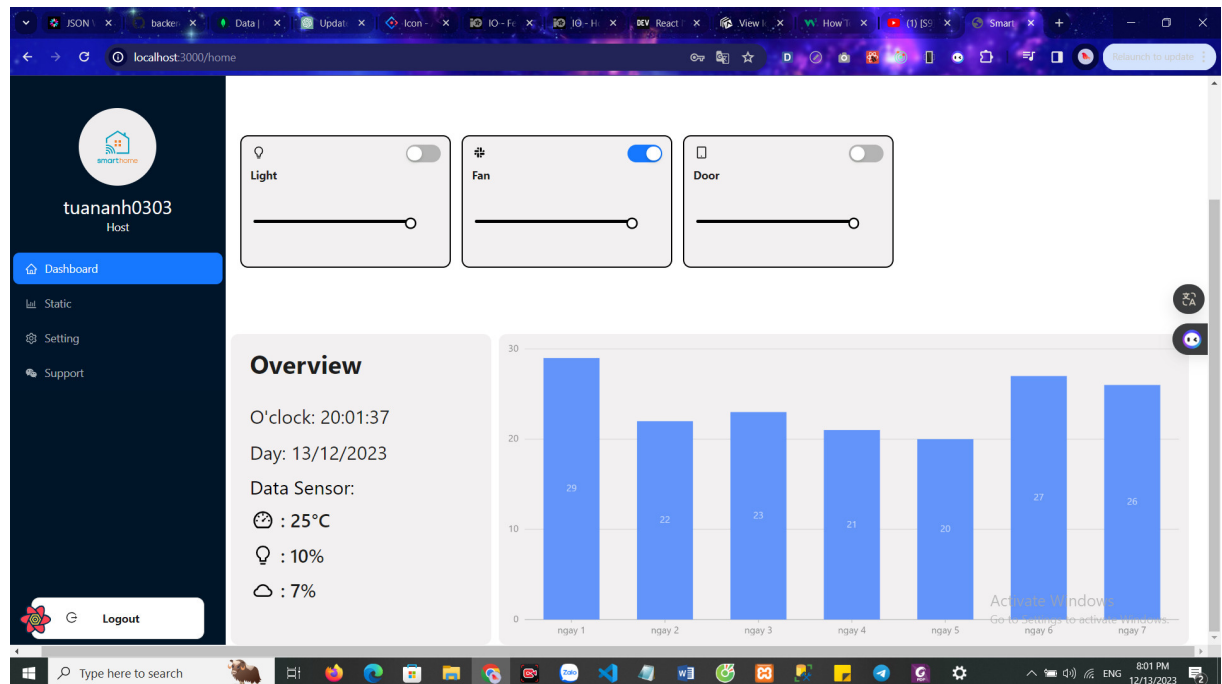
7.1 Giao diện trên Website



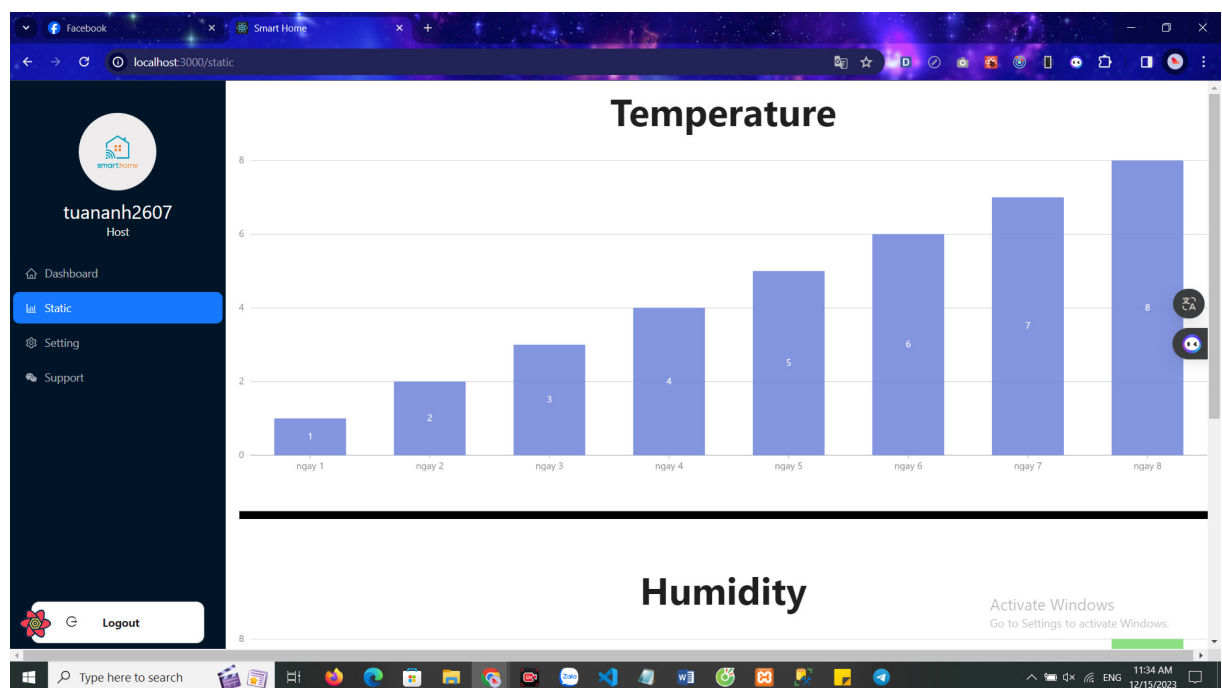
Hình 8: Giao diện trang đăng nhập



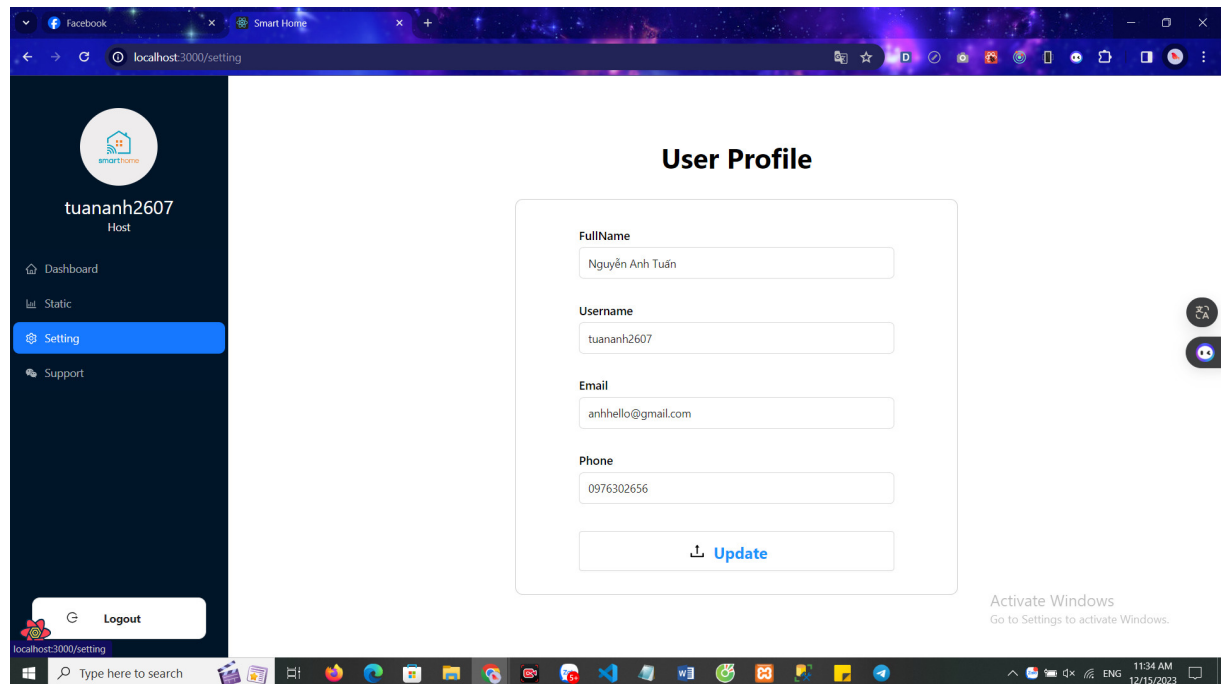
Hình 9: Giao diện trang đăng ký



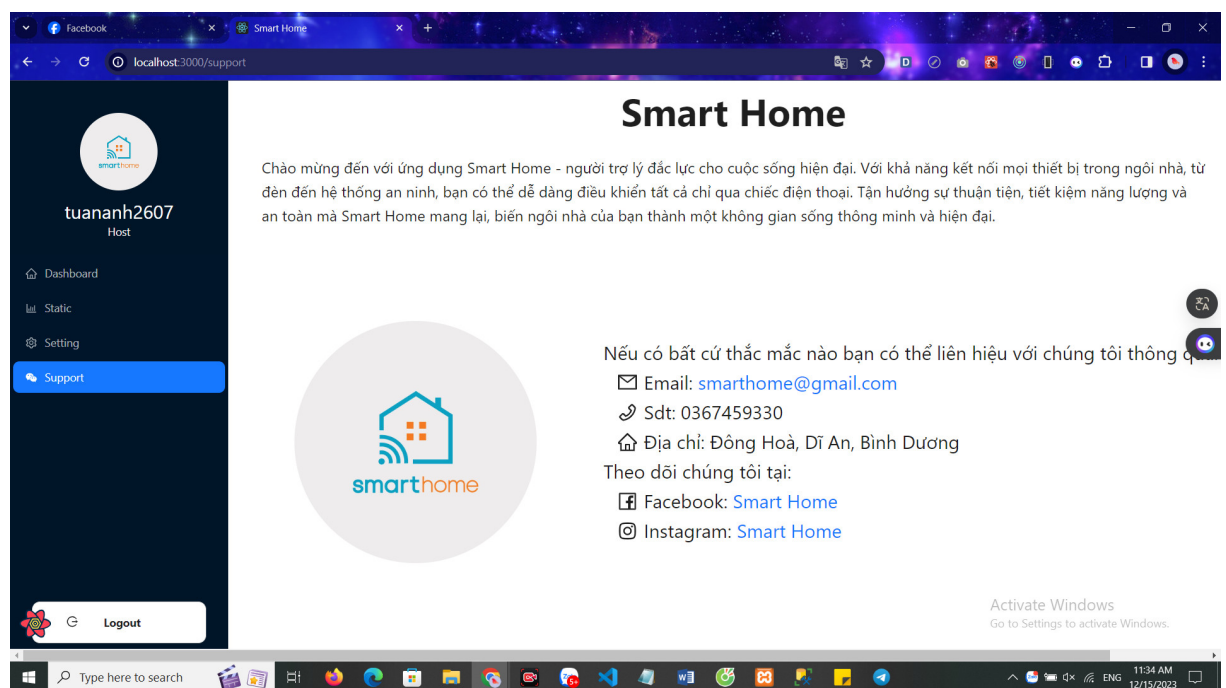
Hình 10: Giao diện điều khiển thiết bị



Hình 11: Giao diện Static

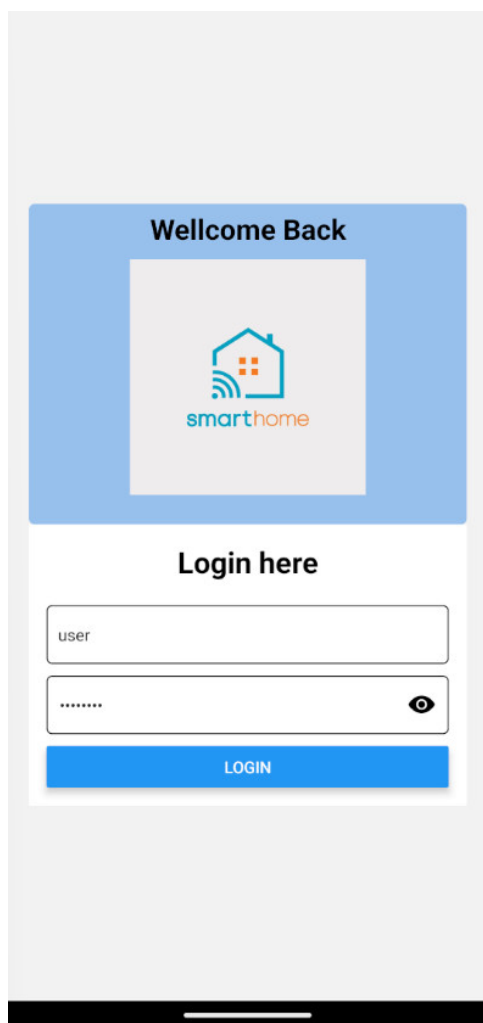


Hình 12: Giao diện Settings

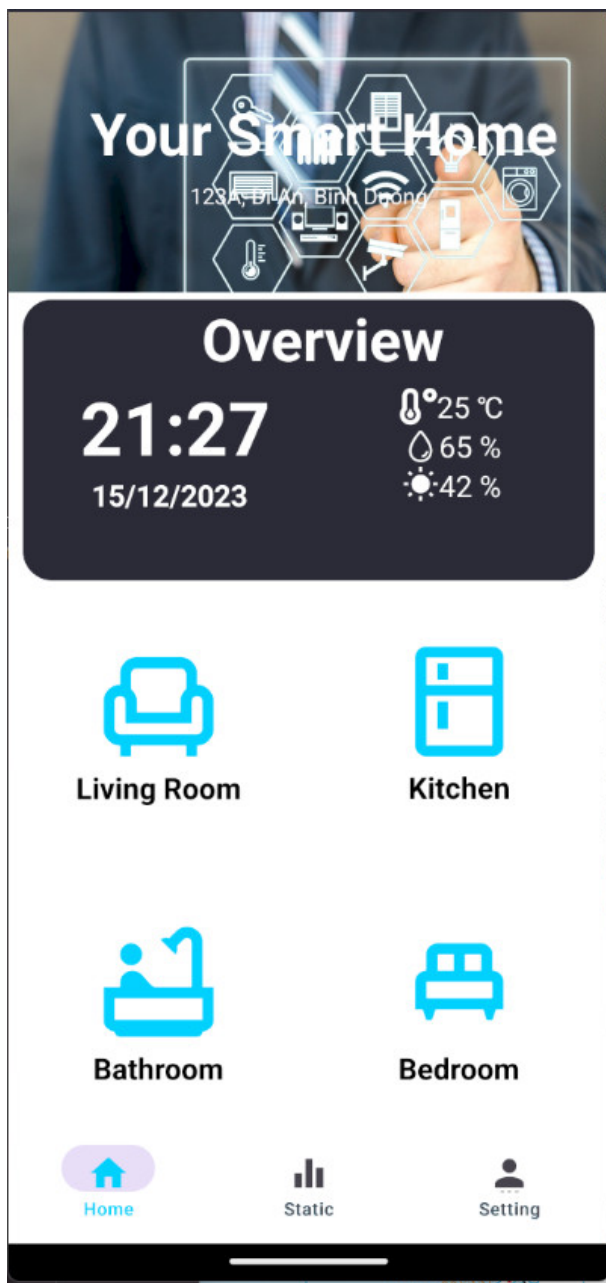


Hình 13: Giao diện Support

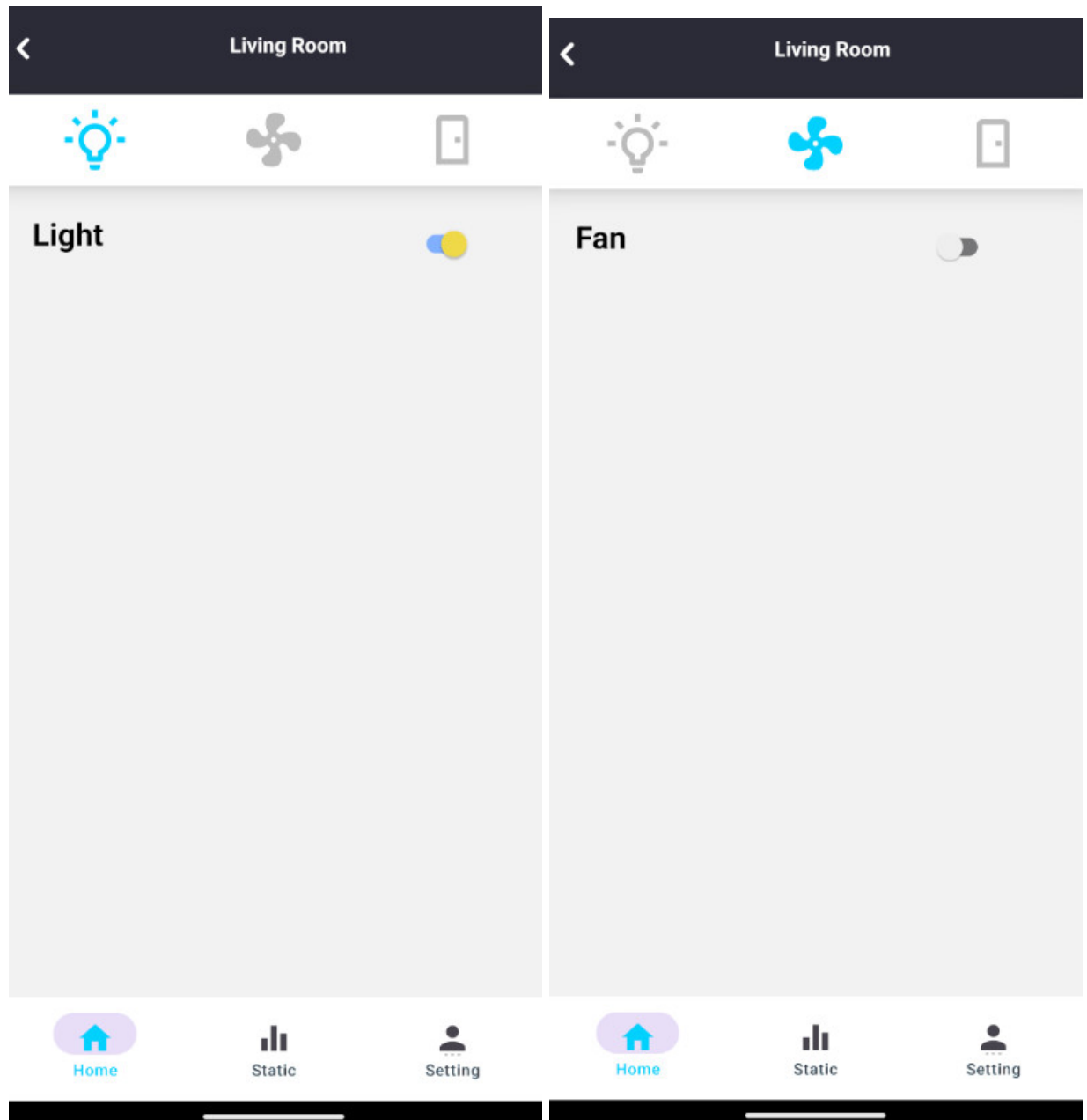
7.2 Giao diện trên mobile



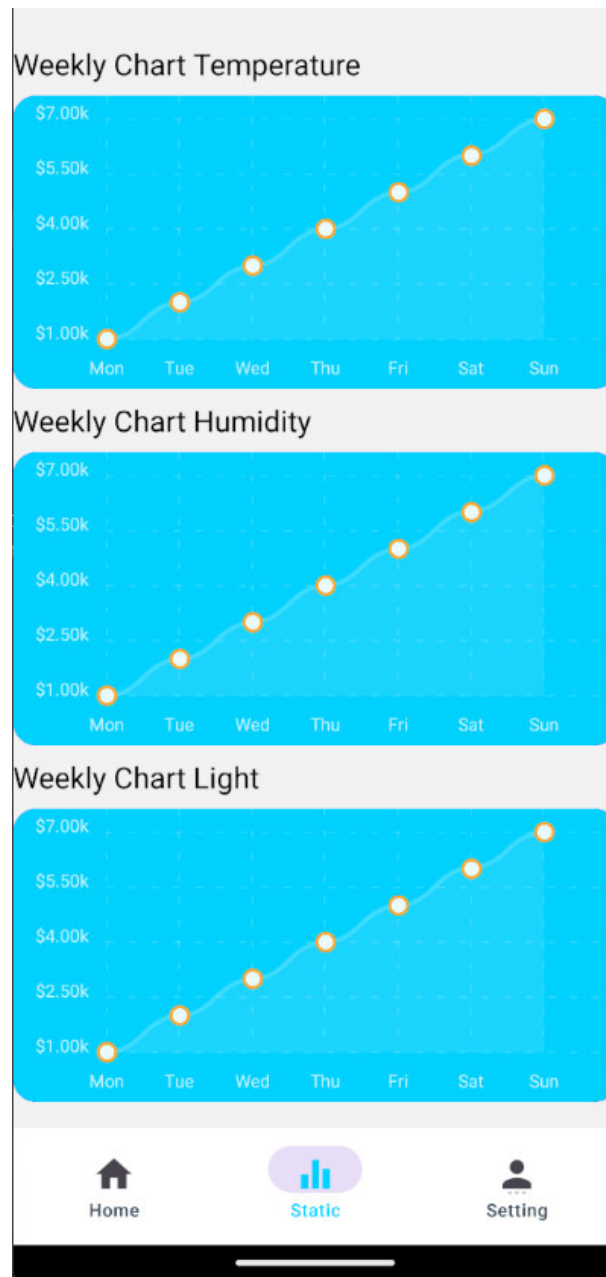
Hình 14: Giao diện đăng nhập trên mobile



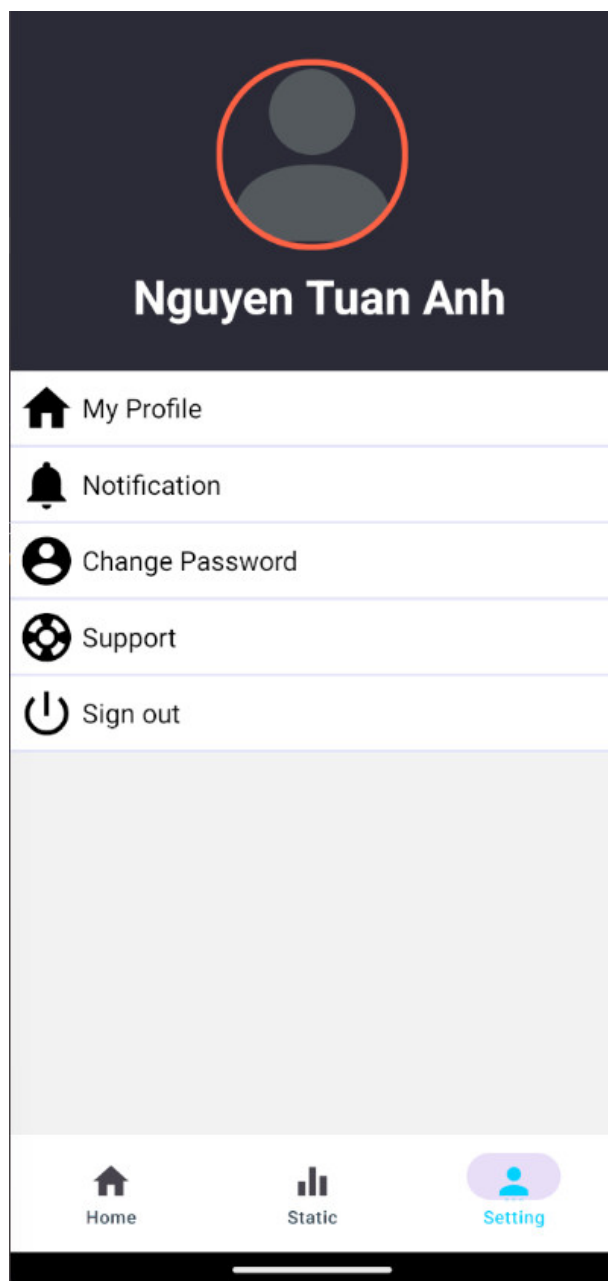
Hình 15: Giao diện Home



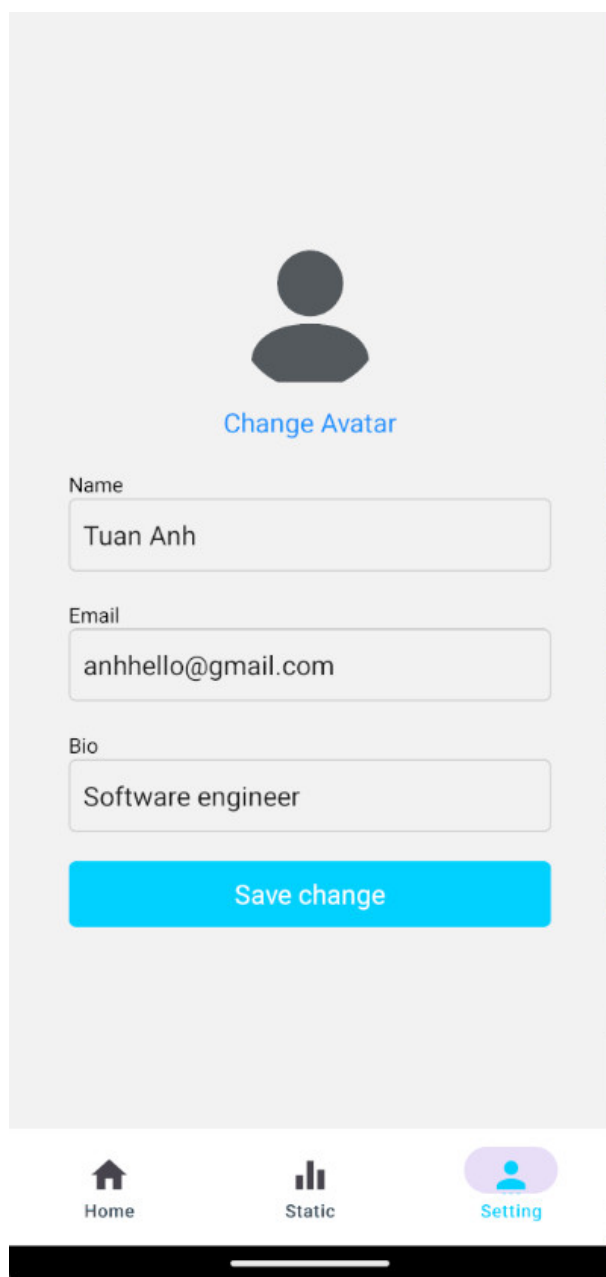
Hình 16: Giao diện điều khiển thiết bị



Hình 17: Giao diện Static trên mobile



Hình 18: Giao diện Settings trên mobile



The image shows a mobile application interface for editing a user's profile. At the top, there is a dark grey silhouette of a person's head and shoulders. Below this is a blue text link that says "Change Avatar". The form consists of three text input fields: "Name" with the value "Tuan Anh", "Email" with the value "anhhello@gmail.com", and "Bio" with the value "Software engineer". Below these fields is a prominent blue button labeled "Save change". At the bottom of the screen is a navigation bar with three icons: a house icon labeled "Home", a bar chart icon labeled "Static", and a person icon labeled "Setting" (which is highlighted with a purple background).

Hình 19: *Giao diện thay đổi thông tin*

8 Tổng kết

8.1 Ưu điểm

Nhóm đã làm được những điểm dưới đây:

- Hoàn thiện được gần như đầy đủ các UI đã tạo
- Hiện thực được chức năng bật/tắt đèn thông qua ứng dụng
- Hiện thực được chức năng bật/tắt quạt thông qua ứng dụng
- Hiển thị được các thông tin nhiệt độ, độ ẩm, ánh sáng trên ứng dụng
- Điều khiển được các thiết bị theo 2 chiều từ hệ thống đến thiết bị và ngược lại.

hông ti

8.2 Nhược điểm

Nhưng có một số điểm mà nhóm em vẫn chưa làm được:

- Chưa hiện thực được chức năng bật/tắt đèn tự động dựa trên độ sáng và khi không sử dụng
- Chưa hiện thực được chức năng đặt thời gian bật tắt thiết bị
- Hệ thống vẫn chưa đáp ứng được một số yêu cầu phi chức năng, trong đó có thời gian phản hồi, gây khó chịu cho người dùng

8.3 Hướng phát triển của dự án

Dưới đây là những đề xuất để nhóm có thể tiếp tục hoàn thiện hơn nữa hệ thống:

- Thêm một số loại phòng, thiết bị, cảm biến
- Thiết kế lại UI-UX để cải thiện trải nghiệm người dùng
- Hoàn thiện tính năng tự động theo thời gian để tăng thêm trải nghiệm người dùng

Tài liệu tham khảo

- [1] Những điều cần biết về giao thức MQTT.

URL: <https://techmaster.vn/posts/34394/iot-giao-thuc-mqtt-va-ung-dung-trong-iot>

Truy cập ngày 26/7/2021.

- [2] Khái niệm cơ bản về IoT Gateway - Cấu trúc của IoT Gateway - Chức năng của một IoT Gateway.

URL: <https://www.maytinh365.com/2019/07/IoT-Gateway-la-gi.html>

Truy cập ngày 26/7/2021.

- [3] IOT Gateway là gì? Ý nghĩa và chức năng vận hành của thiết bị này?

URL: <https://www.thegioimaychu.vn/blog/tong-hop/iot-gateway>

Truy cập ngày 25/7/2021.

- [4] Bước đầu tìm hiểu về ReactJs.

URL: <https://viblo.asia/p/buoc-dau-tim-hieu-ve-reactjs-1VgZvwWYlAw>

Truy cập ngày 25/7/2021.

- [5] Tìm hiểu về python cơ bản.

URL: <https://viblo.asia/p/tim-hieu-ve-python-co-ban-1-LzD5djkeZjY>

Truy cập ngày 27/7/2021.