

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 Công Nghệ Phần Mềm

ĐỀ TÀI

Smart Home

GVHD: Nguyễn Thiên Ân

SV thực hiện: Trần Minh Tân - 2012018
Nguyễn Quốc Thịnh - 2014601
Nguyễn Văn Trọng - 2012299
Mai Lê Cường - 2012764

TP Hồ Chí Minh, Tháng 05/2023



Mục lục

1 Giới thiệu đề tài	4
2 Phân tích yêu cầu	4
2.1 Functional Requirement	5
2.1.1 Thông số kỹ thuật	5
2.1.2 Giao diện	5
2.1.3 Xác thực người dùng	5
2.1.4 Trường hợp sử dụng	5
2.2 Non-Functional Requirement	5
2.2.1 Tốc độ	5
2.2.2 Ứng dụng	5
2.2.3 Độ tin cậy	5
3 Thiết bị	5
3.1 Cảm biến DHT11	5
3.2 Cảm biến đo độ ẩm đất	6
3.3 Yolobit	6
3.4 Quạt Mini	6
3.5 Màn hình LCD	6
3.6 Đèn Led	6
3.7 Module đóng ngắt 2 kênh	6
3.8 Động cơ bơm nước	6
3.9 Cảm biến khoảng cách	6
4 Use-case diagram các tính năng	7
4.1 Use-case 1: Use-case hiển thị	7
4.1.1 Manage Room device	7
4.1.2 Manage Room information	7
4.2 Use-case 2: Đăng nhập, Đăng xuất	9
4.2.1 Đăng nhập	9
4.3 Use-case 3: Thêm, xoá và sửa thiết bị	9
4.3.1 Thêm thiết bị	10
4.3.2 Xoá thông tin thiết bị	10
4.4 Use-case tổng	11
5 Screen Flow	12
5.1 Trang chủ	12
5.2 Login	12
5.3 Màn hình chính	13
5.4 Quản lý Thiết Bị	13
5.5 Xem thông số thiết bị	14
5.6 Quản lý danh sách thiết bị IoT	15
6 Phân chia công việc	17
7 Thiết kế tổng quát	18
7.1 Các thành phần của mô hình MVC:	18
7.2 Luồng xử lí cơ bản	18
7.3 Ưu điểm của mô hình MVC	18
7.4 Ưu điểm của mô hình MVC	19
7.5 Ứng dụng trong bài tập lớn	19
7.5.1 Cấu trúc file	19
7.5.1.a Controller	19
7.5.1.b Model	20
7.5.1.c View	21



8	Tổng quát database	22
8.1	Database Diagram	22
8.2	Mô tả các bảng dữ liệu	22
9	Thiết kế Yolobit	23
9.1	Các tính năng chính của Smart Home được hiện thực trên Yolobit	23
9.1.1	Nhập mật khẩu vào cửa	23
9.1.2	Hiển thị thông số nhiệt độ, độ ẩm	23
9.1.3	Giám Sát Độ Ẩm Đất	23
9.1.4	Điều khiển quạt từ xa	23
9.1.5	Rửa tay tự động	23
9.2	Một số tính năng phụ	23
9.2.1	Bật quạt tự động	23
10	Thiết kế Adafruit.io	24
10.1	Thiết kế trên kênh dữ liệu Feeds	24
10.2	Thiết kế trên kênh dữ liệu Dashboards	24
10.3	Thông tin chung về Dashboards	24
10.4	Các loại dữ liệu được hiện thực trên Dashboards	25
11	Thành phẩm	26



DOCUMENT HISTORY

Date	Version	Change	Person in Charge
15/02/2023	1.0.0	Tạo ra báo cáo	All
17/02/2023	1.0.1	Thêm các lược đồ Use-case và Use-case table	All
17/02/2023	1.0.2	Thêm Mô tả về yêu cầu chức năng/phi chức năng, cách các thiết bị được sử dụng trong project này	Lê Cường
29/03/2023	1.0.3	Add Figma UIs of features	Minh Tân, Quốc Thịnh
07/04/2023	1.0.4	Kết nối thành công thiết bị với Adafruit	Cường
16/04/2023	1.0.5	Demo các thiết bị trực tiếp với thầy và nhận feedback	All
26/04/2023	1.0.6	Kết nối thành công cơ sở dữ liệu với Adafruit	Trọng
04/05/2023	2.0.1	Xây dựng Sidebar và IoT Device Manager	Tân
06/05/2023	2.0.2	House Manager được hoàn thành sơ bộ	Trọng, Thịnh
11/05/2023	2.0.3	Database được chỉnh sửa lại so với Adafruit server	Trọng, Cường
12/05/2023	2.0.4	House Manager được chỉnh sửa sao cho phù hợp với Back-end	Thịnh
13/05/2023	2.0.5	Chỉnh sửa, bổ sung biểu đồ vào chức năng Statistic	Trọng
14/05/2023	2.0.6	IoT Device Manager được chỉnh sửa sao cho phù hợp với Back-end	Tân
15/05/2023	2.0.7	IoT Device Manager được chỉnh sửa sao cho phù hợp với Back-end	Tân
16/05/2023	2.0.8	Sửa lỗi IoT House Manager	Cường, Trọng, Thịnh
17/05/2023	2.0.9	Sửa lỗi Login, IoT Manager, Device Manager và Merge các phần lại với nhau	ALL

1 Giới thiệu đề tài

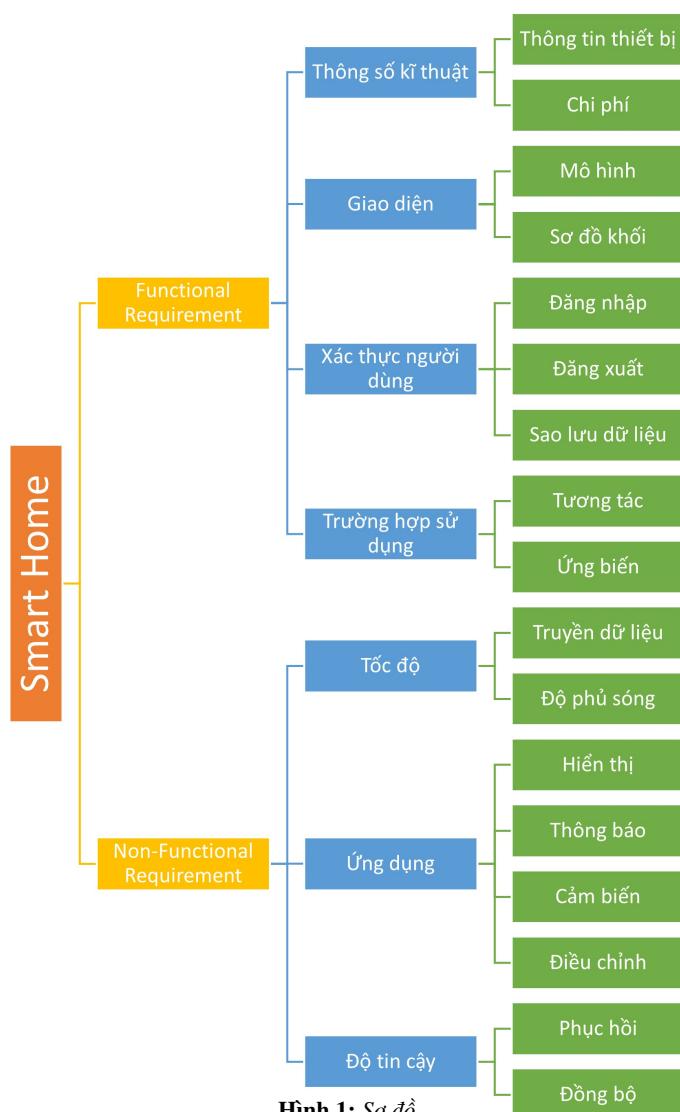
Internet of Things (IoT) là một cuộc cách mạng mới của Internet đã nỗi lên như một mô hình mới để kết nối các đối tượng thông qua Internet. IoT không chỉ là một mạng máy tính mà còn là một mạng lưới các đối tượng vật lý. Thuật ngữ IoT hay Internet vạn vật đề cập đến mạng lưới tập hợp các thiết bị thông minh và công nghệ tạo điều kiện thuận lợi cho hoạt động giao tiếp giữa thiết bị và đám mây cũng như giữa các thiết bị với nhau.

Nhờ sự ra đời của chip máy tính giá rẻ và công nghệ viễn thông băng thông cao, ngày nay, chúng ta có hàng tỷ thiết bị được kết nối với internet. Điều này nghĩa là các thiết bị hàng ngày như bàn chải đánh răng, máy hút bụi, ô tô và máy móc có thể sử dụng cảm biến để thu thập dữ liệu và phản hồi lại người dùng một cách thông minh.

IoT nâng cao khả năng kết nối mọi lúc, mọi nơi với bất kỳ ai, lý tưởng nhất là sử dụng mọi đường dẫn/mạng và mọi dịch vụ. Nó nhằm mục đích liên kết các thiết bị tự báo cáo trong thời gian thực, nâng cao hiệu quả và đưa thông tin quan trọng lên bề mặt nhanh hơn một hệ thống phụ thuộc vào sự can thiệp của con người.

Hệ thống Smart Home được coi là một trong những ứng dụng nổi bật trong kỷ nguyên Internet of Things (IoT), nơi có thể điều khiển các thiết bị trong nhà để đạt được mục đích sử dụng tốt hơn về mặt chi phí và sự tiện lợi. IoT cung cấp kết nối internet và quản lý tự động hóa gia đình từ xa, được tích hợp với nhiều cảm biến. Smart Home là hình thức nhà ở sáng tạo và phổ biến trong các thành phố hiện đại, do đó các nhà thiết kế cần phải phù hợp với sở thích và yêu cầu của người sử dụng.

2 Phân tích yêu cầu





2.1 Functional Requirement

2.1.1 Thông số kỹ thuật

- Thông tin thiết bị có thể được hiển thị trực quan trong IoT Device Manager
- Người dùng có thể chỉnh sửa các thông số để điều khiển các thiết bị trong House Manager.
- Người dùng có thể xem thống kê về độ ẩm và nhiệt độ.

2.1.2 Giao diện

- Có thể xem và điều khiển được các thiết bị IoT trong nhà.
- Thân thiện, dễ sử dụng cho người dùng.

2.1.3 Xác thực người dùng

- Người dùng có thể sử dụng tài khoản được cấp để truy cập và sử dụng các tính năng của hệ thống.

2.1.4 Trường hợp sử dụng

- Người dùng có thể thêm và xoá các thiết bị trong hệ thống.
- Người dùng có thể xem được thông tin người dùng của mình.
- Người dùng có thể điều chỉnh thông số của các thiết bị trong nhà thông qua House Manager.
- Người dùng không thể truy cập các tính năng khi chưa đăng nhập.

2.2 Non-Functional Requirement

2.2.1 Tốc độ

- Độ trễ của việc truyền dữ liệu Dữ liệu phải đảm bảo trong ít nhất 0.5s.
- Việc đăng nhập của người dùng phải diễn ra trong nhiều nhất 1s.

2.2.2 Ứng dụng

- Hiển thị : Thông tin nhiệt độ, độ ẩm, độ sáng trong phòng phải được cập nhật mỗi 0.5s
- Thông kê: Thông kê về nhiệt độ, độ ẩm phải được cập nhật sau một ngày.

2.2.3 Độ tin cậy

- Ứng dụng phải đảm bảo được khôi phục sau tấn công DDoS sau nhiều nhất 2 tiếng.
- Server có thể xử lý đăng nhập cho 500 tài khoản trong cùng một thời điểm.

3 Thiết bị

Danh sách các thiết bị sử dụng trong mô hình Smart Home

3.1 Cảm biến DHT11

- Ứng dụng : đo nhiệt độ và độ ẩm không khí rồi gửi về hệ thống.
- Input : nhiệt độ và độ ẩm, cảm biến được đặt trên mặt đất.
- Output : giá trị nhiệt độ, độ ẩm thu được gửi về hệ thống.



3.2 Cảm biến đo độ ẩm đất

- Ứng dụng : đo độ ẩm trong đất rồi gửi về hệ thống.
- Input : độ ẩm đất, cảm biến được đặt dưới đất.
- Output : giá trị độ ẩm đất thu được gửi về hệ thống.

3.3 Yolobit

- Ứng dụng : mạch trung tâm đóng vai trò trung gian nối các thiết bị và kết nối với nguồn.
- Input : các thiết bị đo đặc và màn hình kết nối.
- Output : các kết quả thu được, thông tin gửi về hệ thống.

3.4 Quạt Mini

- Ứng dụng : Quạt xoay để làm mát không khí
- Input : Lệnh điều khiển từ User hoặc quạt xoay tự động
- Output : Quạt xoay theo mức độ điều khiển

3.5 Màn hình LCD

- Ứng dụng : Hiển thị lời chào, thông báo và thông số đo được.
- Input : Tín hiệu từ hệ thống.
- Output : Hiển thị theo tín hiệu nhận được.

3.6 Đèn Led

- Ứng dụng : Phát sáng cho ngôi nhà theo điều khiển của người dùng.
- Input : Lệnh từ trung tâm điều khiển.
- Output : Bật/Tắt tùy theo tín hiệu nhận được.

3.7 Module đóng ngắt 2 kênh

- Ứng dụng : Module trung gian có 2 cổng kết nối, 1 cổng kết nối với máy bơm.
- Input : Lệnh từ trung tâm điều khiển
- Output : Truyền dữ liệu để bật/tắt máy bơm.

3.8 Động cơ bơm nước

- Ứng dụng : Cung cấp nước cho các hoạt động trong và ngoài Smart Home.
- Input : Lệnh truyền từ Module đóng ngắt 2 kênh.
- Output : Bật/tắt theo tín hiệu nhận được.

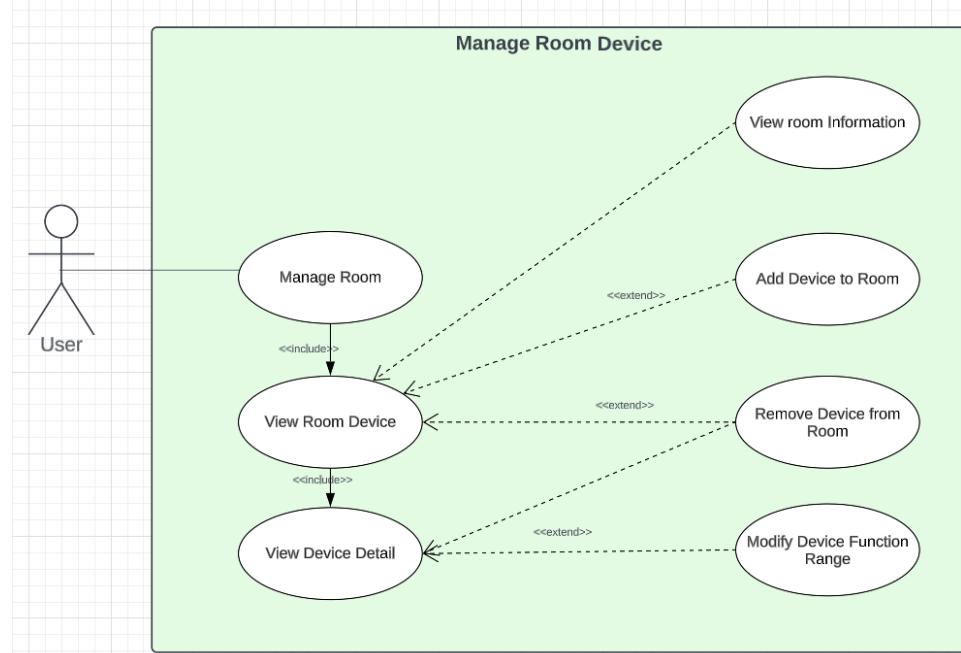
3.9 Cảm biến khoảng cách

- Ứng dụng : Khi có người lại gần sẽ chuyển tín hiệu về hệ thống
- Input : Khoảng cách từ người đến cảm biến gần hay xa.
- Output : Tín hiệu gửi về hệ thống khi khoảng cách gần nhất định.

4 Use-case diagram các tính năng

4.1 Use-case 1: Use-case hiển thị

4.1.1 Manage Room device



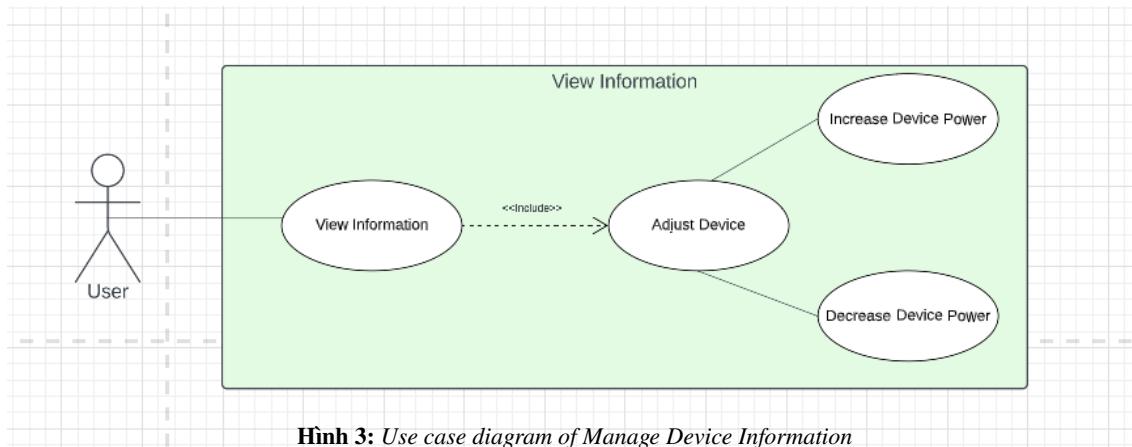
Hình 2: Use case diagram of Manage Device Room

- Use-case: Manage Room Device

Use case	Manage Room Device
Actor	User
Description	Chức năng quản lý các thiết bị có trong từng phòng.
Trigger	Người dùng muốn xem các thiết bị có trong từng phòng
Preconditions	Hệ thống đã đăng nhập thành công.
Postconditions	Hệ thống hoàn tất hiển thị các thiết bị
Main Flow	1. Hệ thống hiển thị các loại thiết bị đã gán vào phòng 2. Người dùng chọn 1 loại thiết bị để xem thông tin chi tiết về các thiết bị trong phòng đó. 3. Hệ thống hiển thị danh sách hoặc chỉ một thiết bị duy nhất. 4. Người dùng có thể thay đổi các phạm vi hoạt động của thiết bị 5. Người dùng xác nhận hoàn tất để quay lại màn hình hiển thị thiết bị
Alternative Flows	Tại bước 5 (2): 5.1. Người dùng chọn thiết bị muốn thay đổi phạm vi hoạt động 5.2. Hệ thống hiển thị các mức hoạt động của thiết bị. 5.3. Người dùng chọn các mức cần thiết. 5.4. Hệ thống hiển thị thông báo yêu cầu xác nhận thay đổi thiết bị . 5.5. Người dùng xác nhận. 5.6. Hệ thống thành công thay đổi thiết bị và quay lại hiển thị thông tin chi tiết của thiết bị
Exception Flow	Tại các bước 2a.3, 2b.3, 3a.5, 3b.5, 5.5 nếu người dùng thoát khỏi ứng dụng thì sẽ dừng use case

4.1.2 Manage Room information

- Use-case: Manage Room Information



Hình 3: Use case diagram of Manage Device Information

Use case	Manage Room Information
Actor	User
Description	Chức năng quản lý các thông tin trong phòng và điều chỉnh gián tiếp nó thông qua các thiết bị.
Trigger	Người dùng muốn xem thông tin phòng hoặc muốn điều chỉnh thiết bị
Preconditions	Hệ thống đang trong giao diện xem các phòng.
Postconditions	Hệ thống hoàn tất hiển thị thông tin trong phòng hoặc điều chỉnh thiết bị
Main Flow	<ol style="list-style-type: none"> 1. Hệ thống hiển thị các thông số trong phòng 2. Người dùng chọn 1 thông số muốn thay đổi 3. Hệ thống hiển thị các thiết bị liên quan tới thông số cần thay đổi 4. Người dùng thay đổi thông số của thiết bị trong phạm vi cho phép 5. Hệ thống xác nhận thay đổi thông số thiết bị 6. Người dùng xác nhận 7. Hệ thống hoàn tất thay đổi thông số thiết bị và quay trở về màn hình hiển thị các thiết bị có thể thay đổi thông số
Alternative Flows	
Exception Flow	tại bước 6 nếu người dùng thoát ứng dụng không xác nhận thì dừng use case

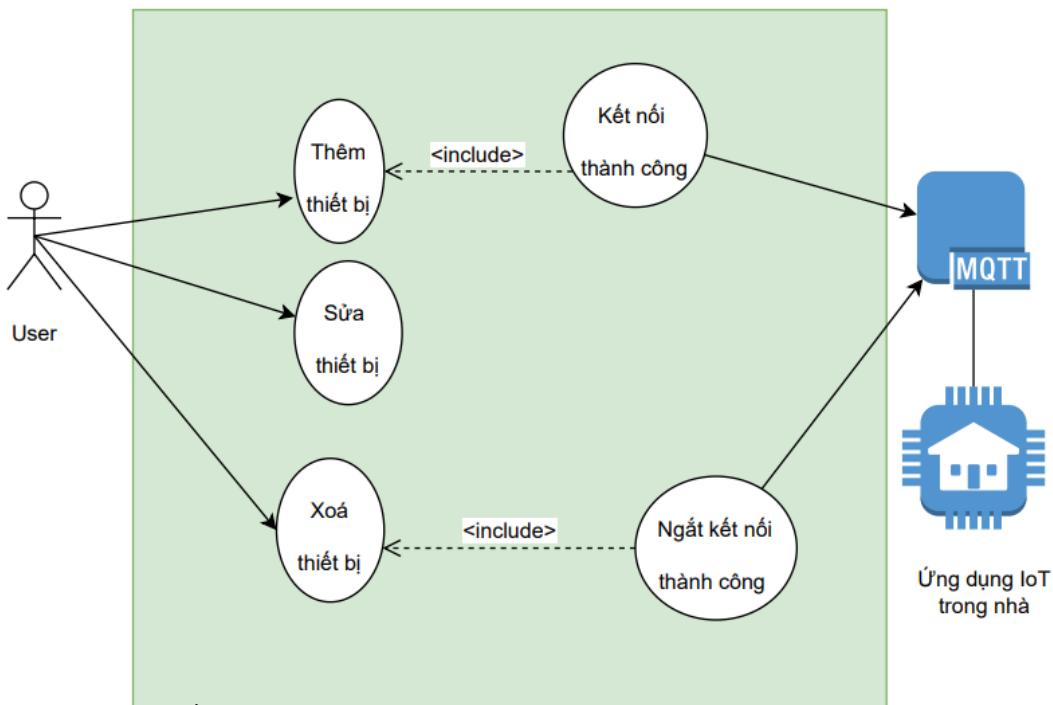
4.2 Use-case 2: Đăng nhập, Đăng xuất

4.2.1 Đăng nhập

Use case	Đăng nhập
Actor	User
Description	Người dùng đăng nhập để sử dụng hệ thống.
Trigger	Người dùng nhấn "Login".
Preconditions	Tài khoản đã có trong hệ thống.
Postconditions	Người dùng đăng nhập thành công.
Main Flow	1. Người dùng truy cập vào hệ thống. 2. Người dùng nhập tài khoản và mật khẩu. 3. Người dùng nhấn nút "Login" 4. Hệ thống xác thực thông tin đăng nhập của người dùng 5. Xác thực thành công, người dùng đăng nhập với vai trò User
Alternative Flows	
Exception Flow	2a. Trên giao diện "Login", người dùng truy cập vào một trang khác . 2b. Hệ thống đưa người dùng về Trang chủ. Kết thúc UC. 5a. Xác thực thất bại, hệ thống thông báo lỗi, người dùng đăng nhập lại.

4.3 Use-case 3: Thêm, xoá và sửa thiết bị

THÊM/SỬA/XOÁ THIẾT BỊ



Hình 4: Use case diagram of add, change infomation and delete IoT devices



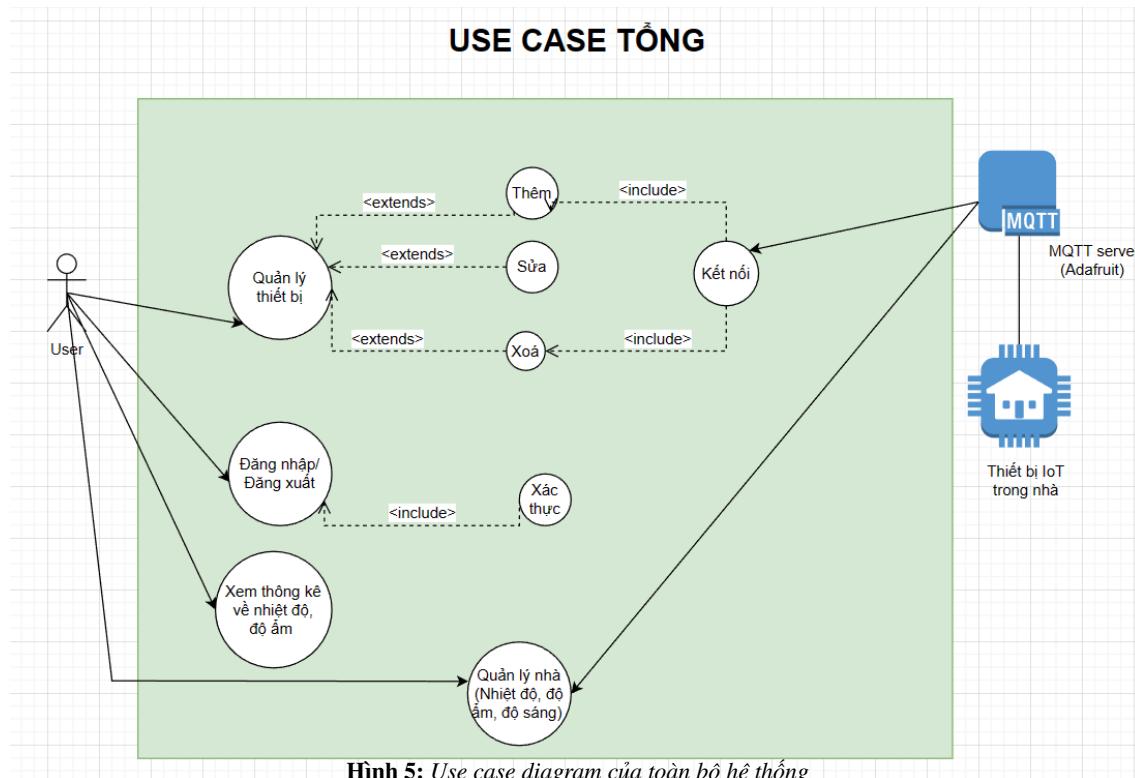
4.3.1 Thêm thiết bị

Use case	THÊM THIẾT BỊ
Actor	User
Description	Chức năng thêm thiết bị IoT trong danh mục quản lý.
Trigger	Người dùng muốn thêm thiết bị “thông minh” trong ngôi nhà của mình vào hệ thống để quản lý từ xa tốt hơn
Preconditions	Người dùng đã đăng nhập vào hệ thống
Postconditions	Người dùng thêm được thiết bị mình mong muốn vào hệ thống quản lý
Main Flow	<ol style="list-style-type: none">Từ Trang chủ, người dùng chọn mục “Device manager” ở thanh dọc bên trái (mục thứ 3)Tại trang quản lý thiết bị , người dùng chọn “Add devices” ngay trên thanh công cụNgười dùng điền các thông tin cần thiết của thiết bị vào, trong đó có thông tin cần để kết nối với MQTT.Người dùng bấm “Submit” để thực hiện việc lưu vào databaseHệ thống web sẽ hiển thị form hỏi lại người dùng có điền đúng thông tin không. Người dùng chọn YesThiết bị được thêm vào thành công
Alternative Flows	NGƯỜI DÙNG MUỐN SỬA THÔNG TIN THIẾT BỊ CẦN THÊM, người dùng chọn "No", sau đó tiến hành quay lại bước 3 để kiểm tra việc điền thông tin cần để kết nối với MQTT. Sau đó tiếp tục thực hiện từ bước 4 của Main Flow.

4.3.2 Xoá thông tin thiết bị

Use case	Xoá thông tin thiết bị
Actor	User
Description	Chức năng xoá thông tin các thiết bị đã cũ trong nhà.
Trigger	Thiết bị không còn hoạt động được nữa hoặc thay thế thiết bị mới
Preconditions	Người dùng ngắt và xoá được kết nối với thiết bị thành công.
Postconditions	Người dùng sửa được thông thiết bị mình mong muốn vào hệ thống quản lý, trừ những vùng liên quan đến kết nối IoT.
Main Flow	<ol style="list-style-type: none">Từ Trang chủ, người dùng chọn mục “Device manager” ở thanh dọc bên trái (mục thứ 3) .Trang quản lý thiết bị sẽ hiện ra danh sách các thiết bị. Người dùng chọn một thiết bị cần xoá trong đó.Người dùng nhấn icon "thùng rác", nằm ở ngoài cùng ứng với thiết bị cần xoá.Hệ thống hiện ra thông báo , yêu cầu người dùng xác nhận yêu cầu xoá của mình. Người dùng xác nhận yêu cầu xoáNgười dùng xoá thành công thiết bị
Alternative Flows	
Exception Flow	NGƯỜI DÙNG MUỐN DÙNG VIỆC XOÁ Tại bước 4, người dùng bấm “No” hoặc bấm dấu “X” trên hộp thoại, sau đó trở về với giao diện thông tin thiết bị.

4.4 Use-case tổng

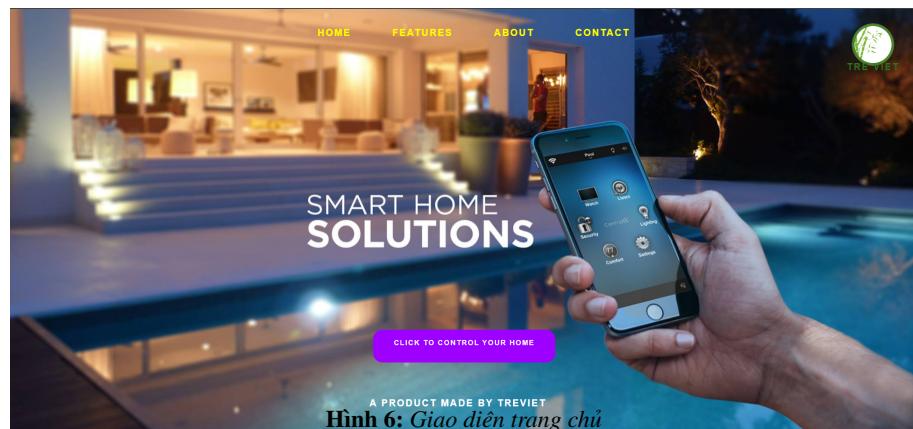




5 Screen Flow

5.1 Trang chủ

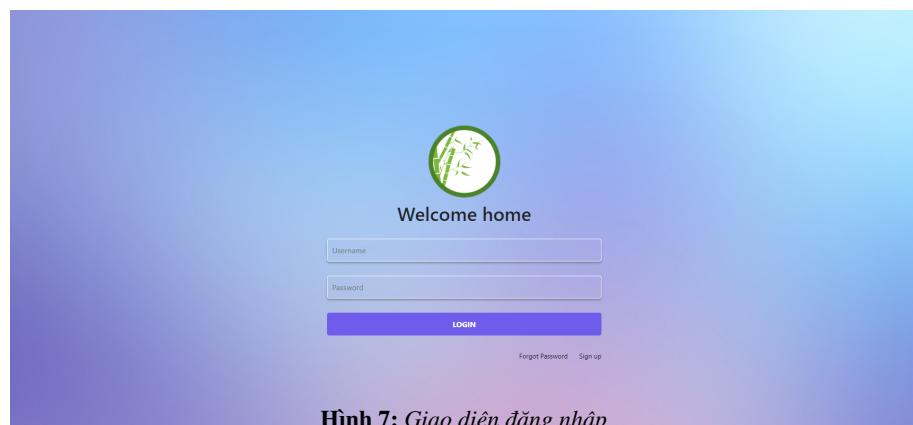
Khi người dùng truy cập vào hệ thống, giao diện Homepage sẽ hiển thị thông tin cơ bản về Web app Smart Home. Người dùng click vào "Click to control your home" để vào giao diện Đăng nhập.



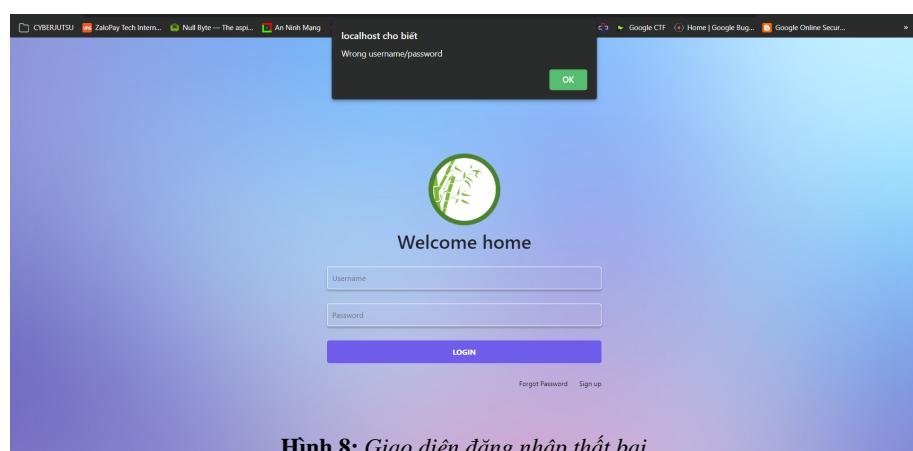
Hình 6: Giao diện trang chủ

5.2 Login

Tại giao diện login, người dùng đăng nhập theo thông tin đã được cung cấp sẵn trong hệ thống để sử dụng hệ thống. Nếu người dùng đăng nhập sai thông tin, hệ thống sẽ pop-up thông báo "Wrong username/password".



Hình 7: Giao diện đăng nhập

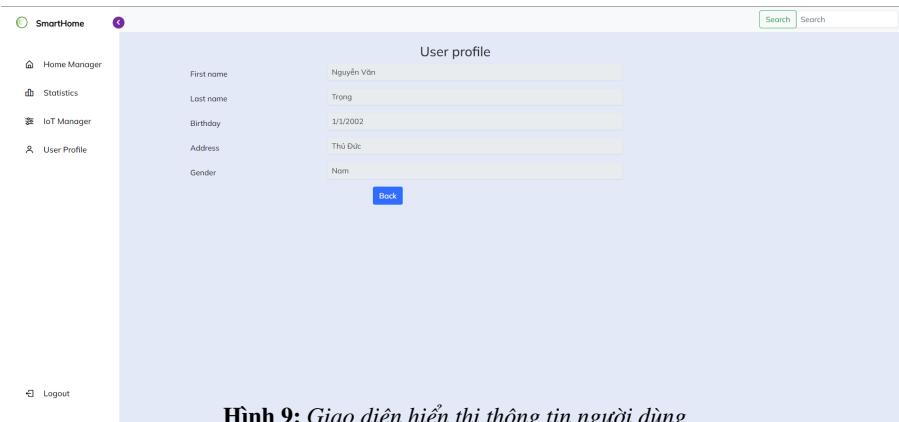


Hình 8: Giao diện đăng nhập thất bại



5.3 Màn hình chính

Người dùng sau khi đăng nhập thành công, hệ thống sẽ hiển thị giao diện chính của hệ thống, đồng thời hiện thông tin của người dùng.



Hình 9: Giao diện hiển thị thông tin người dùng

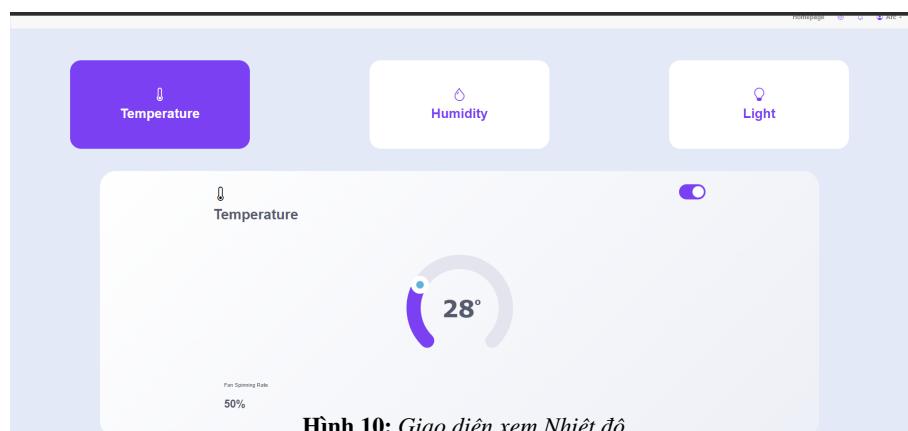
Tại giao diện, người dùng có thể sử dụng các chức năng của hệ thống thông qua Sidebar, bao gồm:

- **Home Manager:** Điều khiển các thiết bị trong nhà như Quạt, Đèn, v.v...
- **Statistic:** Xem các thông số trong ngôi nhà như Độ ẩm, Nhiệt độ, v.v... ở thời gian thực.
- **IoT Manager:** Xem và quản lý các thiết bị hiện có, bao gồm thêm, xóa thiết bị.
- **User Profile:** Xem thông tin của người dùng. Sau khi đăng nhập thành công, người dùng được đưa đến giao diện này.

5.4 Quản lý Thiết Bị

Tại đây, người dùng có thể xem thông tin của nhiệt độ, kèm theo đó, người dùng có thể điều khiển quạt thông qua thanh xoay trên màn hình.

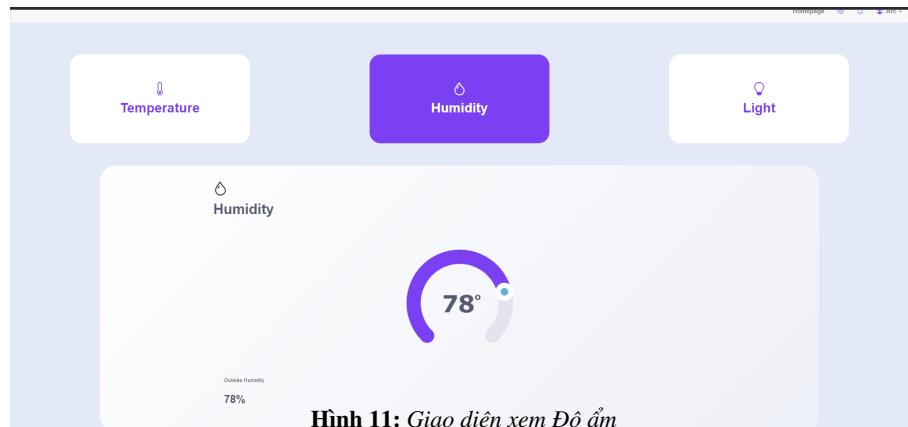
Nếu nhiệt độ được đo đạc trên 35 độ, hệ thống sẽ tự động bật quạt ở mức 50



Hình 10: Giao diện xem Nhiệt độ

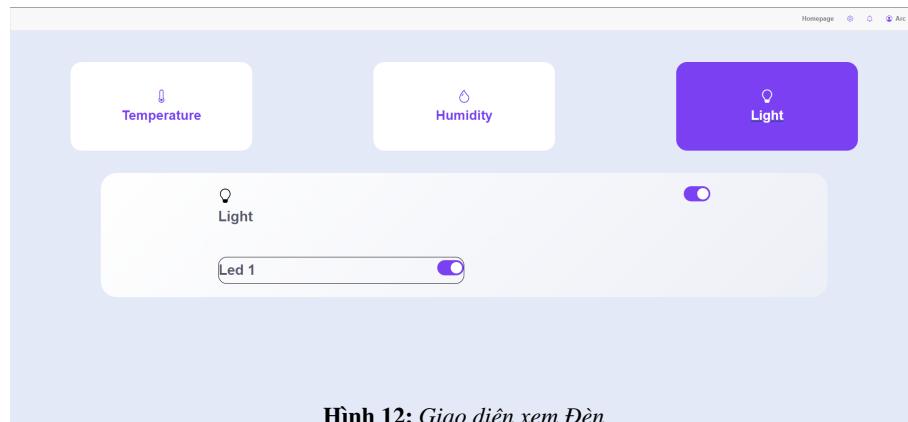


Tại giao diện này, người dùng có thể theo dõi độ ẩm của căn phòng chứa thiết bị.



Hình 11: Giao diện xem Độ ẩm

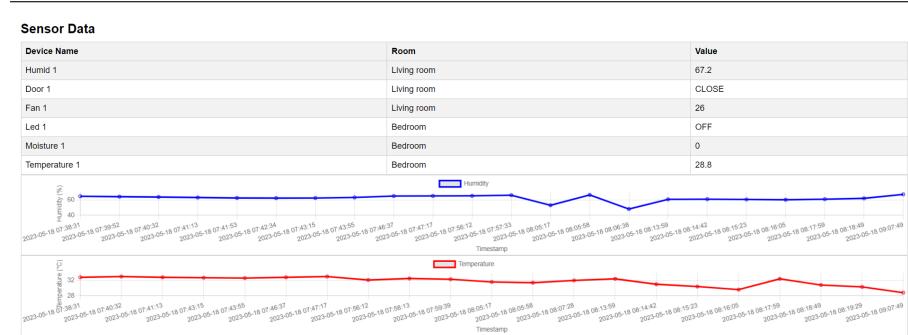
Tại giao diện này, người dùng có thể xem trạng thái của đèn. Đồng thời có thể tắt/bật đèn thông qua hệ thống.



Hình 12: Giao diện xem Đèn

5.5 Xem thông số thiết bị

Tại đây, người dùng có thể xem tất cả thông số của các thiết bị. Đồng thời có các đồ thị để theo dõi các thông số trong ngày.



Hình 13: Giao diện xem thông số tất cả thiết bị



5.6 Quản lý danh sách thiết bị IoT

Khi vào giao diện, người dùng sẽ xem được tất cả các thiết bị hiện có. Nếu người dùng muốn thêm thiết bị, nhấn vào "Add device".

The screenshot shows a web-based IoT management interface titled 'Manage IoT devices'. On the left is a sidebar with links: 'SmartHome', 'Home Manager', 'Statistics', 'IoT Manager', and 'User Profile'. The main area has a search bar at the top right. Below it is a table titled 'Device list' with columns: No, Name, Feed name, House, Room, and Sensor. The table contains six entries:

No	Name	Feed name	House	Room	Sensor
1	Humid 1	dht20-humi	KTX	Living room	Multi-value
2	Door 1	yolo-door	KTX	Living room	Two-mode text
3	Fan 1	yolo-fan	KTX	Living room	Multi-value
4	Led 1	yolo-led	KTX	Bedroom	Two-mode number
5	Moisture 1	yolo-moisture	KTX	Bedroom	Multi-value
6	Temperature 1	yolo-temp	KTX	Bedroom	Multi-value

At the bottom left is a 'Logout' link.

Hình 14: Danh sách các thiết bị IoT đang được sử dụng trong nhà

Nếu người dùng muốn thêm thiết bị, nhấn vào "Add device". Giao diện Thêm thiết bị sẽ hiện ra.

The screenshot shows a 'New IoT devices' form. It includes fields for 'Device Name' (set to 'New_Device'), 'Feed Name' (set to 'New_Feed_name'), 'Room' (set to 'KTX - Living room'), and a dropdown menu for 'Sensor' which lists 'Two-mode number', 'Two-mode number', 'Two-mode text', and 'Multi-value'. The 'Two-mode text' option is currently selected.

Hình 15: Giao diện thêm thiết bị IoT

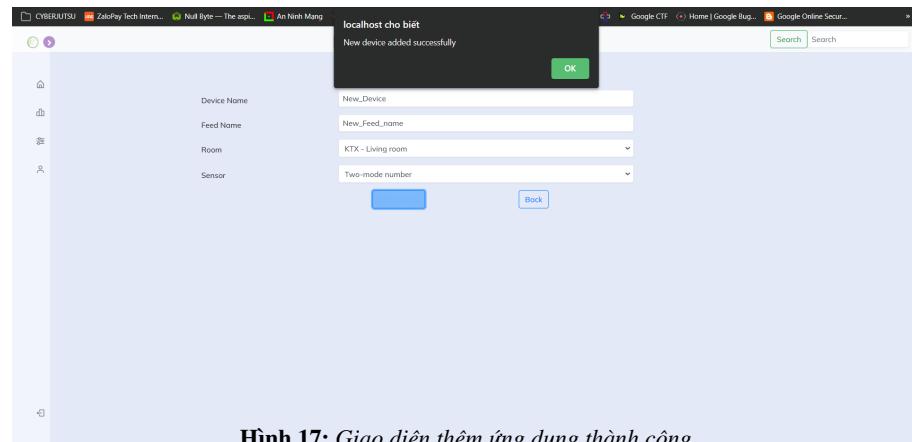
Tại đây, người dùng điền các thông tin như Tên thiết bị, Tên feed-name liên kết với adafruit, chọn phòng và loại thiết bị. Sau khi điền đầy đủ thông tin, nhấn Add device. Hệ thống sẽ hiện pop-up xác nhận.

The screenshot shows a confirmation dialog box titled 'localhost cho biet' with the question 'Are you sure want to add this device?'. It has two buttons: 'OK' and 'Hủy'. Below the dialog is the same 'New IoT devices' form as in Figure 15, with the 'Sensor' dropdown still showing 'Two-mode text'. At the bottom of the form are 'Add Device' and 'Cancel' buttons.

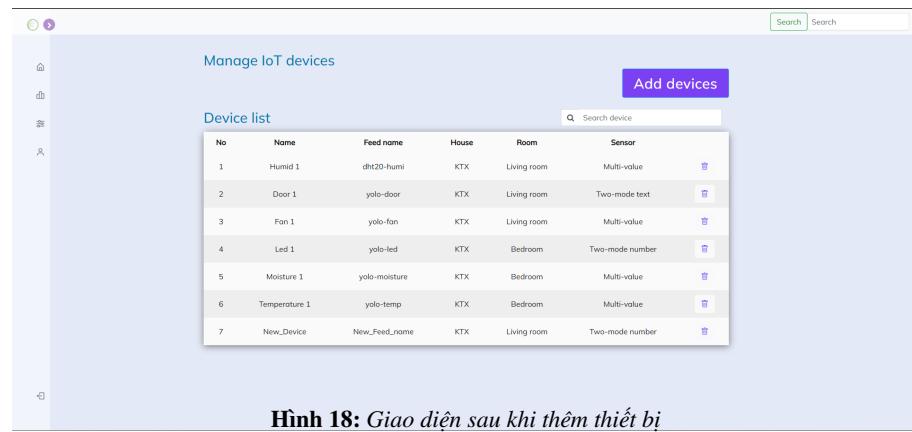
Hình 16: Giao diện xác nhận thêm thiết bị IoT



Người dùng nhấn "Xác nhận", hệ thống sẽ thêm vào dữ liệu và hiện ra màn hình danh sách thiết bị mới.

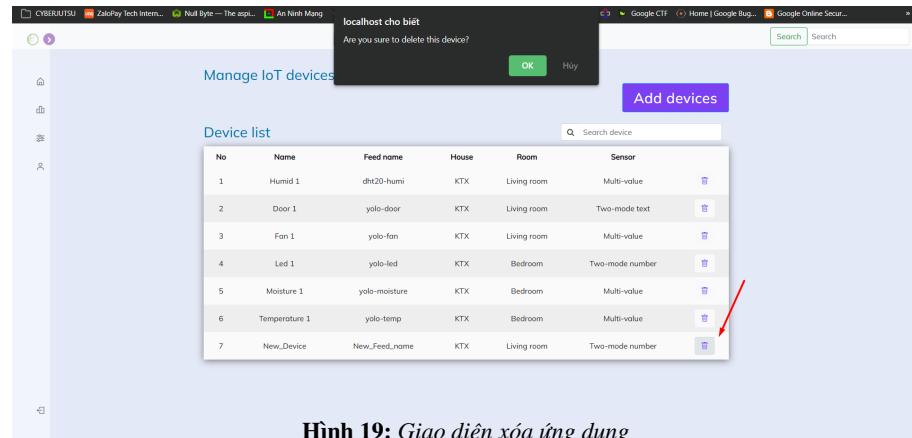


Hình 17: Giao diện thêm ứng dụng thành công



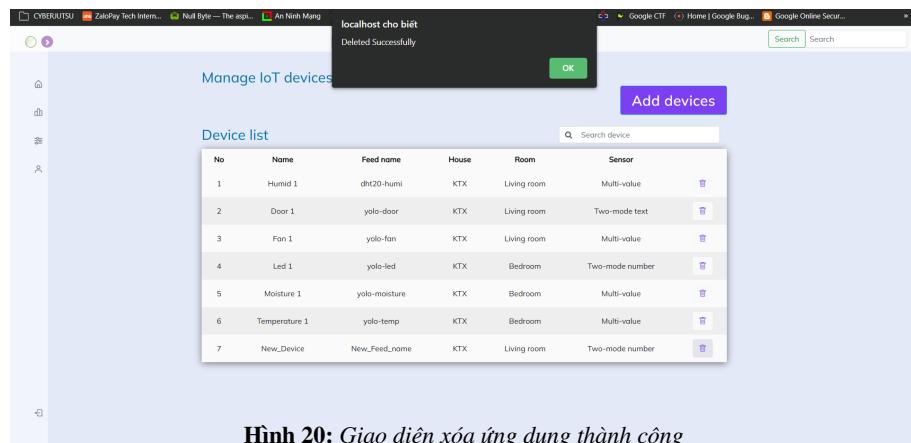
Hình 18: Giao diện sau khi thêm thiết bị

Tại giao diện, người dùng có thể xóa thiết bị thông qua nút "Thùng rác" ở bên cạnh thiết bị. Nhấn vào xóa, hệ thống sẽ pop-up xác nhận.



Hình 19: Giao diện xóa ứng dụng

Người dùng nhấn "Xác nhận", hệ thống sẽ xóa dữ liệu của thiết bị và hiện ra màn hình danh sách thiết bị mới.



Hình 20: Giao diện xóa ứng dụng thành công

6 Phân chia công việc

Thành viên	Nhiệm vụ
Nguyễn Quốc Thịnh	<ul style="list-style-type: none"> Thiết kế UI tổng thể cho hệ thống. Thiết kế UI cho phần House Manager Hiện thực UI các chức năng thuộc House Manager Hiện thực UI Top-bar
Trần Minh Tân	<ul style="list-style-type: none"> Thiết kế UI cho phần Device Manager Hiện thực UI Side-bar. Hiện thực UI cho các chức năng thuộc phần Device Manager.
Mai Lê Cường	<ul style="list-style-type: none"> Viết mô tả, yêu cầu chức năng và phi chức năng. Viết use-case cho các chức năng cần thiết trong hệ thống. Kết nối, kiểm thử các thiết bị với MQTT và Adafruit.
Nguyễn Văn Trọng	<ul style="list-style-type: none"> Vẽ lược đồ cơ sở dữ liệu. Kết nối database với Adafruit server. Hiện thực UI phần Homepage, Login.



7 Thiết kế tổng quát

Trong đề tài này nhóm sẽ xây dựng trang web theo mô hình MVC.

MVC là viết tắt của cụm từ "Model - View - Controller". Đây là mô hình được sử dụng phổ biến trong kỹ thuật phần mềm, đặc biệt là lập trình web. MVC chia làm 3 thành phần khác nhau, mỗi thành phần đều có một nhiệm vụ của nó và độc lập với các thành phần khác.

7.1 Các thành phần của mô hình MVC:

- Model (dữ liệu): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản.
- View (giao diện): là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
- Controller(Bộ điều khiển): Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.

Ví dụ :một Controller có thể cập nhật một Model bằng cách thay đổi các thuộc tính của nhân vật trong game. Và nó có thể sửa đổi view bằng cách hiển thị nhân vật được cập nhật trong game đó.

7.2 Luồng xử lí cơ bản

- Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
- Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
- Model chuẩn bị data và gửi lại cho Controller.
- Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

Ở đây, **View không giao tiếp trực tiếp với Model**. Sự tương tác giữa **View và Model** sẽ chỉ được xử lý bởi **Controller**.

7.3 Ưu điểm của mô hình MVC

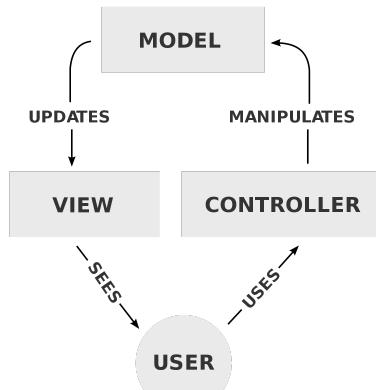
- **Tiết kiệm băng thông:** Vì không sử dụng viewstate nên MVC rất nhẹ và tiết kiệm được diện tích của băng thông. Khi cần tương tác gửi và nhận dữ liệu liên tục, người dùng có thể sử dụng các ứng dụng trên web. Điều này giúp website có thể hoạt động ổn định và tốt hơn.
- **Dễ dàng kiểm tra:** Nhờ có MVC, bạn sẽ dễ dàng hơn trong việc kiểm tra, rà soát lỗi, đảm bảo được chất lượng và độ uy tín cho phần mềm trước khi tới tay người dùng. Đồng thời cũng Hỗ trợ TDD (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won test case.
- **Dễ dàng duy trì và chia việc:** Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau. Ta cũng có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
- **Tính kết hợp:** Có thể thoải mái viết code trên nền tảng web khi tích hợp ở mô hình MVC để giảm tải dữ liệu server.
- **Tính đơn giản:** Kết cấu của mô hình MVC tương đối đơn giản và dễ dàng sử dụng ngay cả khi bạn không có chuyên môn cao.

7.4 Ưu điểm của mô hình MVC

Ngoài các ưu điểm nổi bật bên trên, mô hình MVC cũng có những hạn chế nhất định, cụ thể như:

- Cần nhiều thời gian để tìm hiểu nếu muốn áp dụng đúng đắn
- Tốn kém thời gian và công sức để quản lý tổ chức file.
- . Xây dựng quy trình tương đối phức tạp, bởi vậy không cần thiết áp dụng mô hình này cho các dự án nhỏ.

7.5 Ứng dụng trong bài tập lớn



Hình 21: Mô tả mô hình MVC

7.5.1 Cấu trúc file

7.5.1.a Controller

add_device.php	somewhat	40 minutes ago
auth_check.php	somewhat	40 minutes ago
call.php	a	1 minute ago
connect.php	somewhat	40 minutes ago
connect2.php	a	1 minute ago
control.php	somewhat	40 minutes ago
control2.php	a	1 minute ago
delete_device.php	somewhat	40 minutes ago
display.php	somewhat	40 minutes ago
function.php	somewhat	40 minutes ago
realtime_display.css	somewhat	40 minutes ago
realtime_display.php	somewhat	40 minutes ago
script.php	somewhat	40 minutes ago
searchDevices.php	somewhat	40 minutes ago

Hình 22: List file trong controller

- add_device.php
- auth_check.php



- call.php
- connect.php
- connect2.php
- control.php
- control2.php
- delete_device.php
- display.php
- function.php
- realtime_display.php
- script.php.php
- searchDevices.php

7.5.1.b Model

Bảng	Hành động	Hàng	Kiểu	Bảng mã đổi chiều	Kích thước	Tổng chi phí
devices	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	6	InnoDB	utf8mb4_general_ci	48,0 Kib	-
houses	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	1	InnoDB	utf8mb4_general_ci	16,0 Kib	-
rooms	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	2	InnoDB	utf8mb4_general_ci	32,0 Kib	-
sensors	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	3	InnoDB	utf8mb4_general_ci	16,0 Kib	-
sensor_readings	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	100	InnoDB	utf8mb4_general_ci	32,0 Kib	-
users	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	4	InnoDB	utf8mb4_general_ci	16,0 Kib	-
user_house	★ Duyệt Cấu trúc Tim kiếm Chèn Rỗng Xóa	1	InnoDB	utf8mb4_general_ci	32,0 Kib	-
7 bảng	Tổng	117	InnoDB	utf8mb4_general_ci	192,0 Kib	0 B

Hình 23: List các bảng trong model

- devices
- houses
- rooms
- sensors
- sensor_readings
- users
- user_house



7.5.1.c View

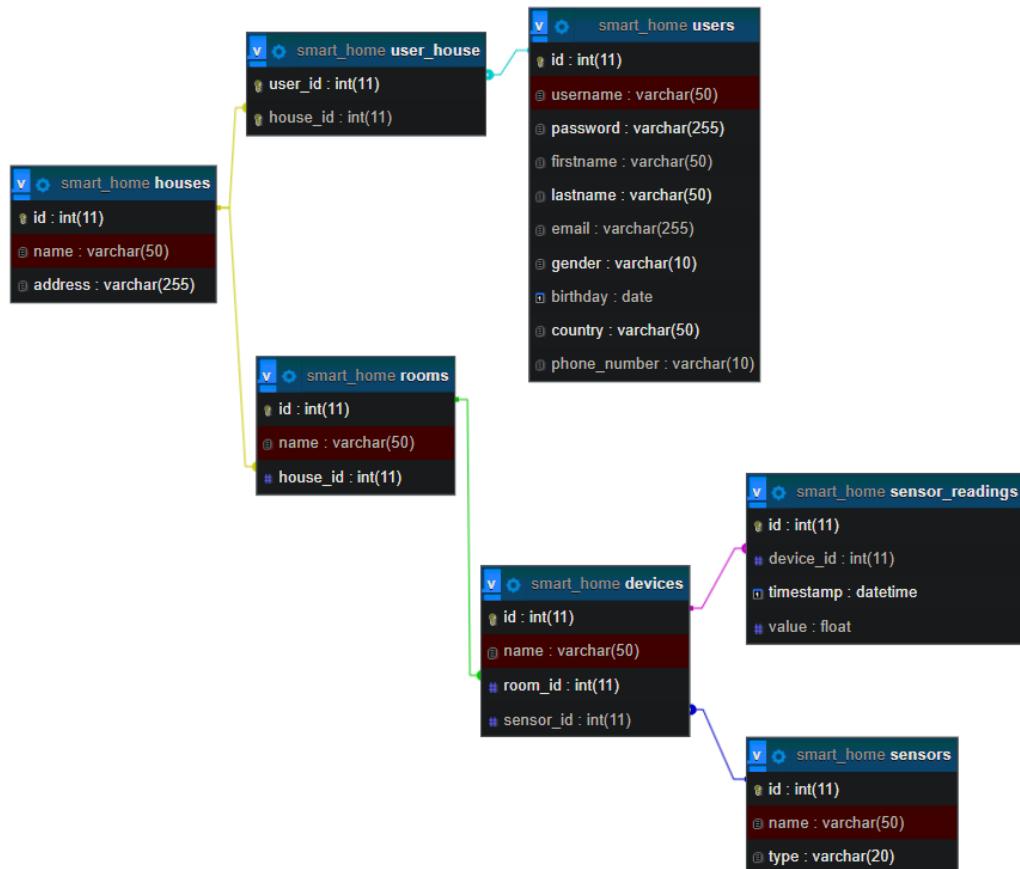
Name	Last commit message	Last commit date
...		
Sidebar.php	somewhat	39 minutes ago
UserDashboard.php	somewhat	39 minutes ago
addIoTDevices.php	somewhat	39 minutes ago
header.php	somewhat	39 minutes ago
homepage.php	somewhat	39 minutes ago
housestate.php	a	now
index.php	somewhat	39 minutes ago
login.php	somewhat	39 minutes ago
topbar.php	somewhat	39 minutes ago
user.php	somewhat	39 minutes ago
userProfile.php	somewhat	39 minutes ago
viewDevices.php	somewhat	39 minutes ago

Hình 24: List các file trong View

- Sidebar.php
- UserDashboard.php
- addIoTDevices.php
- header.php
- homepage.php
- housestate.php
- index.php
- login.php
- topbar.php
- user.php
- userProfile.php
- viewDevices.php

8 Tổng quát database

8.1 Database Diagram



Hình 25: Database diagram cho SmartHouse

8.2 Mô tả các bảng dữ liệu

STT	Tên bảng	Mô tả
1	Houses	Lưu trữ tất cả các nhà
2	User house	Lưu trữ thông tin nhà ứng với mỗi user
3	Users	Lưu trữ thông tin user
4	Rooms	Lưu trữ thông tin từng phòng trong nhà
5	Devices	Lưu trữ thông tin tất cả các thiết bị
6	Sensor readings	Lưu trữ thông tin đọc được từ cảm biến
7	Sensors	Lưu trữ loại thiết bị



9 Thiết kế Yolobit

9.1 Các tính năng chính của Smart Home được hiện thực trên Yolobit

9.1.1 Nhập mật khẩu vào cửa

Hệ thống sẽ yêu cầu mật khẩu vào cửa, khi nhập đúng hệ thống sẽ hiển thị thông báo "Xin Mời Vào" và phát nhạc "Power Up", sau đó gửi thông tin "OPEN" lên Dashboard trên Adafruit.io. Nếu nhập sai hệ thống sẽ hiển thị "Sai Mật Khẩu", phát nhạc "Power Down" và gửi thông tin "CLOSE" lên Dashboard trên Adafruit.io.

9.1.2 Hiển thị thông số nhiệt độ, độ ẩm

Hệ thống sử dụng cảm biến DHT20 để đo nhiệt độ và độ ẩm không khí, thông tin đo được sẽ hiển thị lên màn hình và gửi về Dashboard trên Adafruit.io.

9.1.3 Giám Sát Độ Ẩm Đất

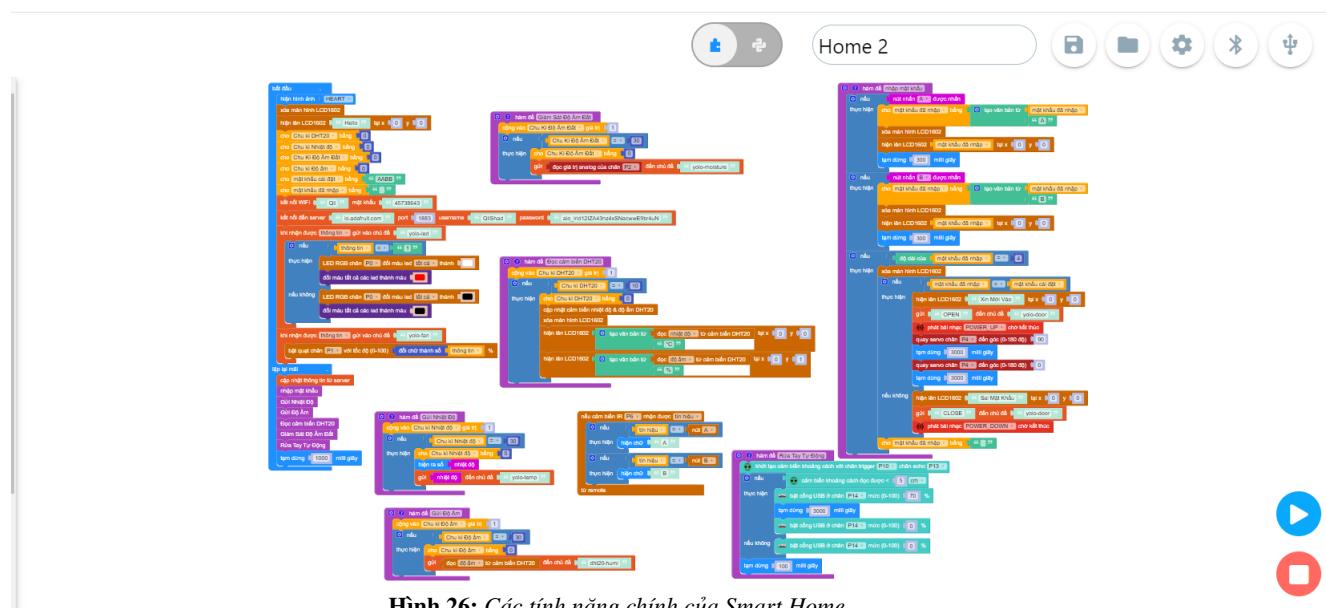
Hệ thống sử dụng cảm biến độ ẩm đất để đo độ ẩm đất, thông tin đo được sẽ được gửi về Dashboard trên Adafruit.io.

9.1.4 Điều khiển quạt từ xa

Người dùng có thể dùng nút nhấn từ Dashboard để có thể điều khiển tốc độ quạt, tốc độ dao động từ 0-100

9.1.5 Rửa tay tự động

Hệ thống sử dụng cảm biến PIR để hiện thực được tính năng này, khi cảm biến nhận thấy có người tới gần sẽ báo về cổng USB và bật máy bơm nước.

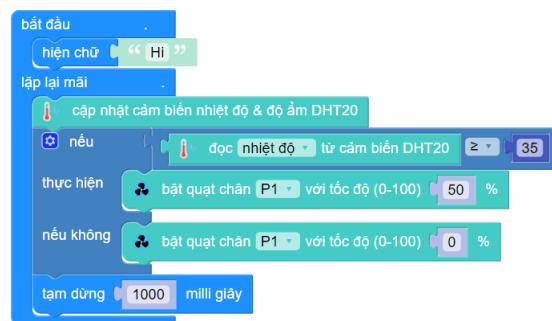


Hình 26: Các tính năng chính của Smart Home

9.2 Một số tính năng phụ

9.2.1 Bật quạt tự động

Khi nhiệt độ vượt ngưỡng được thiết lập sẵn, quạt sẽ tự động quay.



Hình 27: Tính năng bật quạt tự động

10 Thiết kế Adafruit.io

10.1 Thiết kế trên kênh dữ liệu Feeds

- DHT20-HUMI : nhận thông tin về độ ẩm không khí đo được từ DHT20.
- YOLO-DOOR : nhận thông tin từ hệ thống nhập mật khẩu.
- YOLO-FAN : thông tin để điều khiển tốc độ quạt.
- YOLO-LED : thông tin để điều chỉnh đèn bật/tắt.
- YOLO-MOISTURE : nhận thông tin về độ ẩm đất.
- YOLO-TEMP : nhận thông tin về nhiệt độ.

<input type="checkbox"/> DHT20_HUMI	dht20-humi	56.0	1 day ago
<input type="checkbox"/> YOLO_DOOR	yolo-door	CLOSE	1 day ago
<input type="checkbox"/> YOLO_FAN	yolo-fan	0	1 day ago
<input type="checkbox"/> YOLO_LED	yolo-led	0	1 day ago
<input type="checkbox"/> YOLO_MOISTURE	yolo-moisture	0	1 day ago
<input type="checkbox"/> YOLO_TEMP	yolo-temp	28.1	1 day ago

Hình 28: Các kênh được tạo trên Feeds

10.2 Thiết kế trên kênh dữ liệu Dashboards

10.3 Thông tin chung về Dashboards

Dashboards sẽ kết nối với Feeds và nhận/gửi tín hiệu từ mạch để tương tác hoặc nhận thông tin từ thiết bị gửi về. Data sẽ được cập nhật liên tục giữa những khoảng thời gian nhất định.

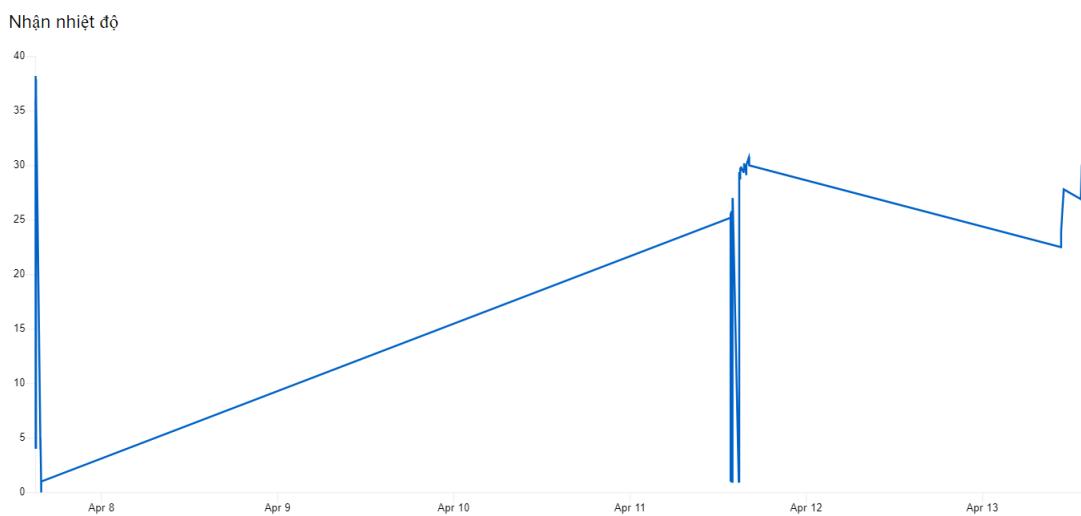
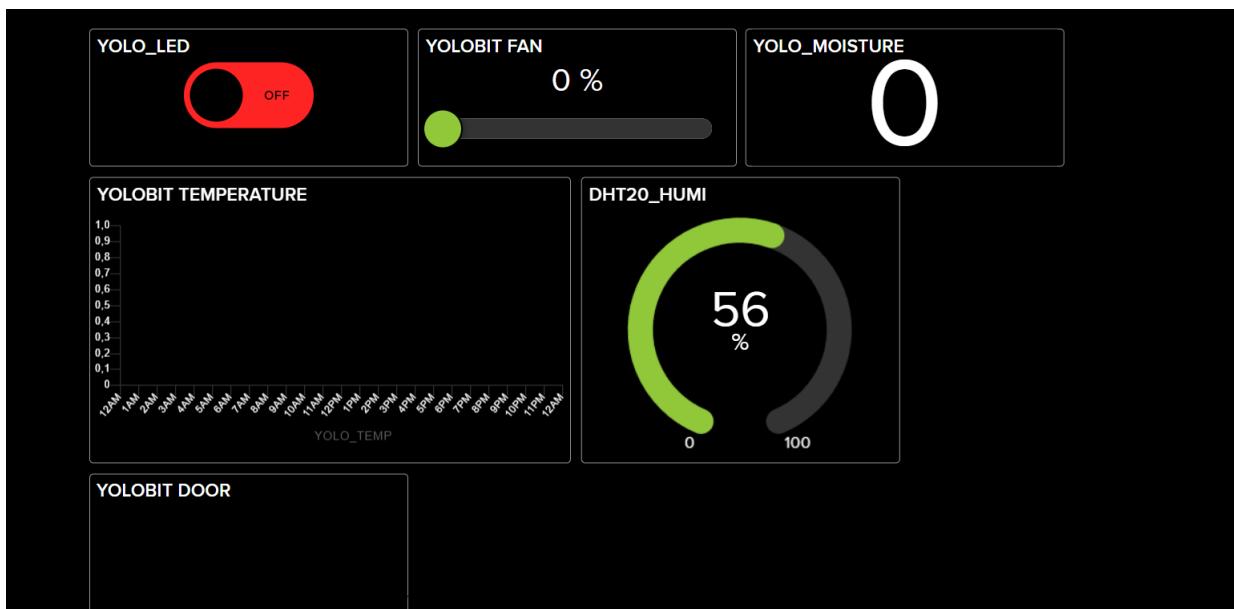
The screenshot shows a table with one row of data:

Name	Key	Created At
YOLO_LED	yolo-led	April 7, 2023

Hình 29: Tên kênh Dashboards của dự án

10.4 Các loại dữ liệu được hiện thực trên Dashboards

- DHT20-HUMI : hiển thị thông tin độ ẩm không khí.
- YOLO-DOOR : khi nhập mật khẩu đúng sẽ hiển thị chữ "OPEN", nếu không sẽ hiển thị "CLOSE", đi kèm là thời gian nhập mật khẩu, ngoài ra sẽ hiện thị ALARM khi có cảnh báo cháy.
- YOLO-FAN : thanh từ 0-100 để điều khiển tốc độ quạt.
- YOLO-LED : 2 mode ON/OFF để điều khiển đèn bật/tắt.
- YOLO-MOISTURE : hiển thị thông tin về độ ẩm đất.
- YOLO-TEMP : hiển thị thông tin về nhiệt độ.



Hình 31: Một ví dụ về nhận dữ liệu bên Feeds và hiển thị lên Dashboards



11 Thành phẩm

Mã nguồn của hệ thống đã hiện thực có thể được truy cập tại: <https://github.com/Cuong251/DADN>