

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



## THỰC TẬP ĐỒ ÁN ĐA NGÀNH

---

Mobile App

# WAREHOUSE MANAGEMENT

---

TA: Trần Khải Thiện  
SV: Nguyễn Đình Thuần - 1713375  
SV: Huỳnh Ngọc Tú - 1713835  
SV: Mai Huỳnh Hữu Tín - 1713505  
SV: Nguyễn Phước Thăng - 1713209  
SV: Phạm Hồng Tiến - 1713490

TP. HỒ CHÍ MINH, THÁNG 7/2020



## Mục lục

<b>1 Giới thiệu đề tài</b>	<b>3</b>
<b>2 Phân công công việc</b>	<b>4</b>
<b>3 Phân tích chức năng</b>	<b>7</b>
3.1 Xem thông tin trạng thái hiện tại của đèn . . . . .	7
3.2 Thiết lập thông số cho thiết bị chiếu sáng . . . . .	8
<b>4 Tìm hiểu công nghệ</b>	<b>9</b>
4.1 Front-end và Back-end: python, html/CSS, JavaScript . . . . .	9
4.2 Database: MongoDB . . . . .	9
4.3 Mô hình IOT . . . . .	10
4.4 Thuật toán mã hoá Advanced Encryption Standard (Dùng mã hoá mật khẩu tài khoản người dùng) . . . . .	10
<b>5 Thiết kế hệ thống</b>	<b>12</b>
5.1 Người dùng . . . . .	12
5.2 Trang web . . . . .	12
5.3 hệ thống đèn . . . . .	12
5.4 Server . . . . .	12
5.5 Database . . . . .	12
<b>6 Mô tả chi tiết</b>	<b>13</b>
6.1 Ứng dụng . . . . .	13
6.1.1 Đăng nhập và đăng ký . . . . .	13
6.1.2 Trang chủ . . . . .	18
6.1.3 Tab Overview . . . . .	19
6.1.4 Tab Controller . . . . .	22
6.1.5 Xem tab Report . . . . .	24
6.1.6 Xem tab Status . . . . .	26
6.2 Server, input và output . . . . .	28
6.2.1 Kết nối server . . . . .	28
6.2.2 Input . . . . .	30
6.2.3 Output . . . . .	30
<b>7 Tổng kết</b>	<b>31</b>
7.1 Kết quả . . . . .	31
7.1.1 Hạn chế . . . . .	31
<b>8 Hướng phát triển</b>	<b>32</b>



## Danh sách hình vẽ

1	Cơ chế tổng quát	3
2	Nhiệt độ và độ ẩm được hiển thị ra màn hình ứng dụng	7
3	Flowchart auto fan	8
4	Sơ đồ triển khai mô hình IoT	10
5	Mô tả các bước của quy trình mã hoá	11
6	Giao diện đăng nhập	15
7	Giao diện đăng ký	16
8	Database tài khoản người dùng	17
9	Giao diện trang chủ	18
10	Giao diện tab Overview	20
11	Tab Controller	23
12	Giao diện xem tab Report	25
13	Giao diện xem tab Status	27

## Danh sách bảng

1	Bảng phân công tuần 3	4
2	Bảng phân công tuần 4	5
3	Bảng phân công tuần 5 - 7	5
4	Bảng phân công tuần 9 - 10	6
5	Use-case chức năng đăng nhập	13
6	Use-case chức năng đăng ký	14
7	Use-case trang chủ	18
8	Use-case xem Overview	19
9	Use-case xem Controller	22
10	Use-case xem tab Report	24
11	Use-case xem tab Status	26

## 1 Giới thiệu đề tài

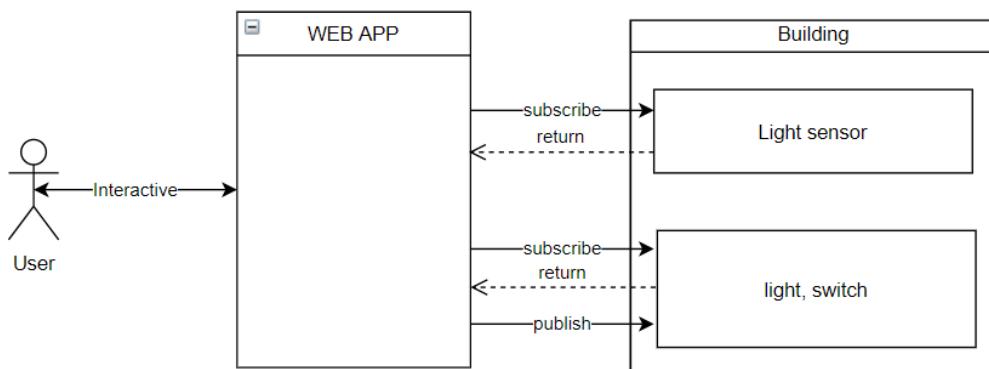
Xu hướng nhà thông minh không dây đang dần trở nên phổ biến trên toàn thế giới. Ở Việt Nam, WESMART là một trong số ít hệ thống nhà thông minh đầu tiên sử dụng kết nối không dây hoàn toàn. Thông qua bộ điều khiển trung tâm WESMART, công nghệ nhà thông minh Internet of Things (IoT) cho độ bảo mật và tốc độ giao tiếp cao. Những thiết bị nhà thông minh lắp đặt được kết nối đồng bộ để tạo nên hệ thống nhà thông minh tiên tiến.

Vì nhu cầu của cuộc sống hiện đại ngày càng cao, việc điều khiển hệ thống đèn chiếu sáng một cách tự động sẽ giảm bớt được các chi phí thuê người quản lý, cũng như tăng tính hiệu quả sử dụng các đèn. Nó cũng giúp ta giảm bớt được điện năng hao phí do những bóng đèn bật không đúng lúc hay do ta quên tắt đèn. Nó hoạt động dựa trên điều kiện ánh sáng tự nhiên, sẽ tự động điều chỉnh độ sáng dựa vào ánh sáng hiện tại và thời gian trong ngày.

Thiết bị input của hệ thống đèn gồm 1 cảm biến là cảm biến ánh sáng . Thiết bị output là một motor công tắt đèn và các bóng đèn.

Người dùng sẽ tương tác với một ứng dụng web đã kết nối với các thiết bị trong tòa nhà thông qua internet. Ứng dụng giúp người dùng xem thông số các thiết bị input theo thời gian thực và chủ động điều khiển thiết bị output theo cơ chế định sẵn hoặc theo cách thủ công.

Cơ chế tổng quát được trình bày trong hình dưới đây:



Hình 1: Cơ chế tổng quát



## 2 Phân công công việc

### Tuần 1:

- Tìm nhóm và họp nhóm.
- Suy nghĩ đề tài.

### Tuần 2:

- Thông nhất đề tài.
- Thông nhất môi trường phát triển và chọn thiết bị.
- Lên ý tưởng cho dự án.

### Tuần 3:

Tên	MSSV	Công việc
Phạm Hồng Tiên	1713490	Viết UseCase "xem mô hình kho" và "điều khiển thông số quạt tự động"
Nguyễn Đình Thuần	1713375	Viết UseCase "đăng nhập" và "đăng ký", viết báo cáo tuần
Mai Huỳnh Hữu Tín	1713505	Viết UseCase "xem thông số gợi ý các loại nông sản" và "xem thời tiết"
Huỳnh Ngọc Tú	1713835	Viết UseCase "đổi thông tin tài khoản" và "đổi mật khẩu"
Nguyễn Phước Thắng	1713209	Viết UseCase "lưu dữ liệu vào lịch sử" và "xem lịch sử"

Bảng 1: Bảng phân công tuần 3



**Tuần 4:**

Tên	MSSV	Công việc
Phạm Hồng Tiến	1713490	Thiết kế giao diện "Environment"
Mai Huỳnh Hữu Tín	1713505	Thiết kế giao diện "Login" và "Register"
Nguyễn Đình Thuần	1713375	Thiết kế giao diện "Paramenter" và "Weather"
Huỳnh Ngọc Tú	1713835	Thiết kế giao diện "Account"
Nguyễn Phước Thăng	1713209	Thiết kế Giao diện "History"

Bảng 2: Bảng phân công tuần 4

**Tuần 5 - 7:**

Tên	MSSV	Công việc
Phạm Hồng Tiến	1713490	Thiết kế giao diện "Environment"
Mai Huỳnh Hữu Tín	1713505	Thiết kế giao diện "Login" và "Register"
Nguyễn Đình Thuần	1713375	Thiết kế giao diện "Paramenter" và "Weather"
Huỳnh Ngọc Tú	1713835	Thiết kế giao diện "Account"
Nguyễn Phước Thăng	1713209	Thiết kế Giao diện "History"

Bảng 3: Bảng phân công tuần 5 - 7

**Tuần 8:**

- Demo giữa kỳ.



**Tuần 9 - 10:**

Tên	MSSV	Công việc
Phạm Hồng Tiên	1713490	Thiết kế giao diện "Environment"
Mai Huỳnh Hữu Tín	1713505	Thiết kế giao diện "Login" và "Register"
Nguyễn Đình Thuần	1713375	Thiết kế giao diện "Paramenter" và "Weather"
Huỳnh Ngọc Tú	1713835	Thiết kế giao diện "Account"
Nguyễn Phước Thăng	1713209	Thiết kế Giao diện "History"

Bảng 4: Bảng phân công tuần 9 - 10

**Tuần 11:**

- Tổng kết dự án.
- Viết báo cáo.

**Tuần 12:**

- Nhóm trưởng nghiệm thu và báo cáo tổng kết: Mai Huỳnh Hữu Tín.

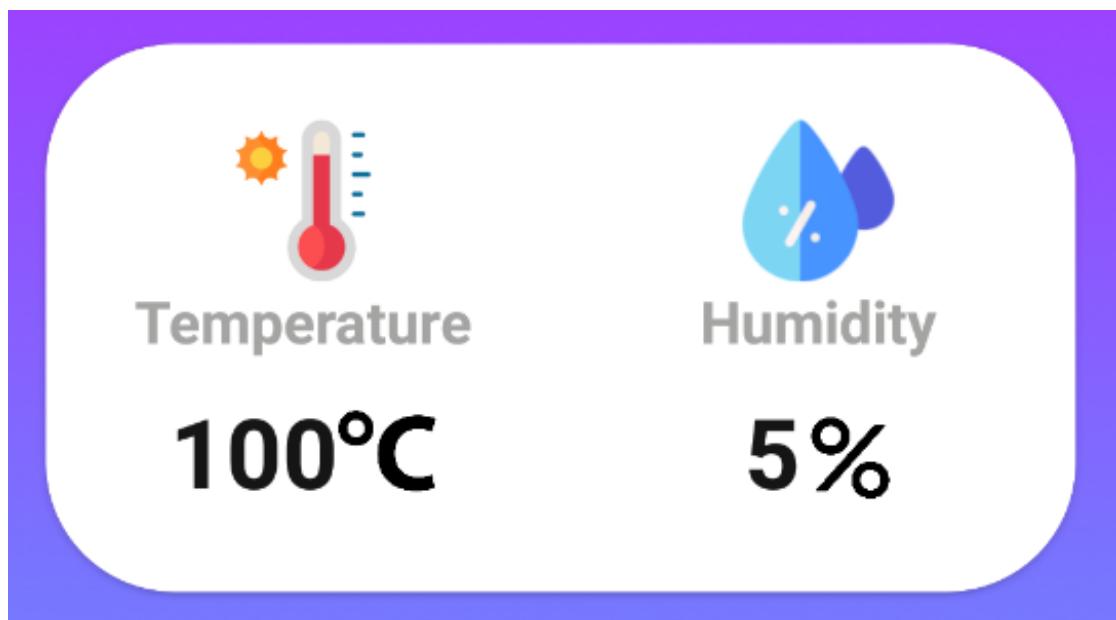
### 3 Phân tích chức năng

#### 3.1 Xem thông tin trạng thái hiện tại của đèn

Với đề tài quản lý hệ thống đèn trong tòa nhà, nhóm lựa chọn thiết bị input là cảm biến cường độ ánh sáng.

Thiết bị input này gồm thông số cường độ ánh sáng với mức sáng từ [0-255] (với mức 0 được xem là trạng thái tắt).

Ứng dụng sẽ lấy thông số từ cảm biến theo thời gian thực và hiển thị ra màn hình dưới dạng một biểu đồ:

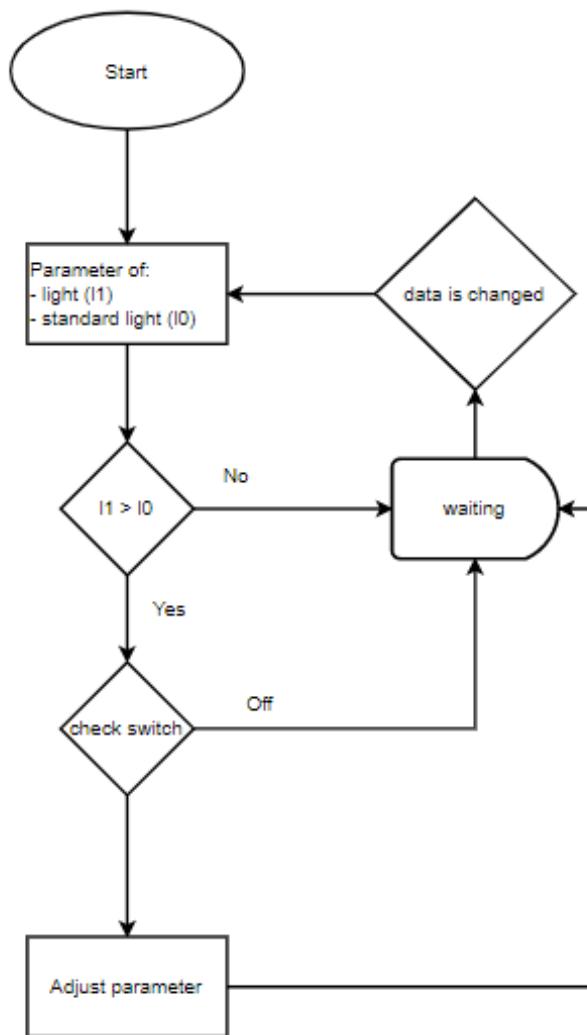


Hình 2: Nhiệt độ và độ ẩm được hiển thị ra màn hình ứng dụng

### 3.2 Thiết lập thông số cho thiết bị chiếu sáng

Ứng dụng điều khiển thiết bị output theo cơ chế:

- Đầu tiên, người dùng có thể vào tab điều chỉnh độ sáng, tại đây trang web sẽ hiện thị trạng thái của đèn hiện tại, kèm theo một thanh điều chỉnh độ sáng bên dưới. Ta có thể điều chỉnh độ sáng bằng cách kéo thả thanh này. Mức sáng sẽ nằm trong giá trị từ 0 - 255, ngoài ra độ sáng của đèn còn có thể thay đổi một cách tự động dựa theo thông số từ cảm biến ánh sáng. Thông số này được cảm biến đo từ ánh sáng môi trường ở thời điểm hiện tại, sau đó gửi dữ liệu về server. Hệ thống sẽ tự động điều chỉnh độ sáng dựa vào thông số này.



Hình 3: Flowchart auto fan



## 4 Tìm hiểu công nghệ

### 4.1 Front-end và Back-end: python, html/CSS, JavaScript

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.[24] Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.

HTML, HyperText Markup Language, cung cấp cấu trúc nội dung và ý nghĩa bằng cách xác định nội dung đó, ví dụ như tiêu đề, đoạn văn hoặc hình ảnh. CSS, hay Cascading Style Sheets, là ngôn ngữ trình bày được dùng để tạo kiểu cho sự xuất hiện của nội dung sử dụng, ví dụ như phông chữ hoặc màu sắc.

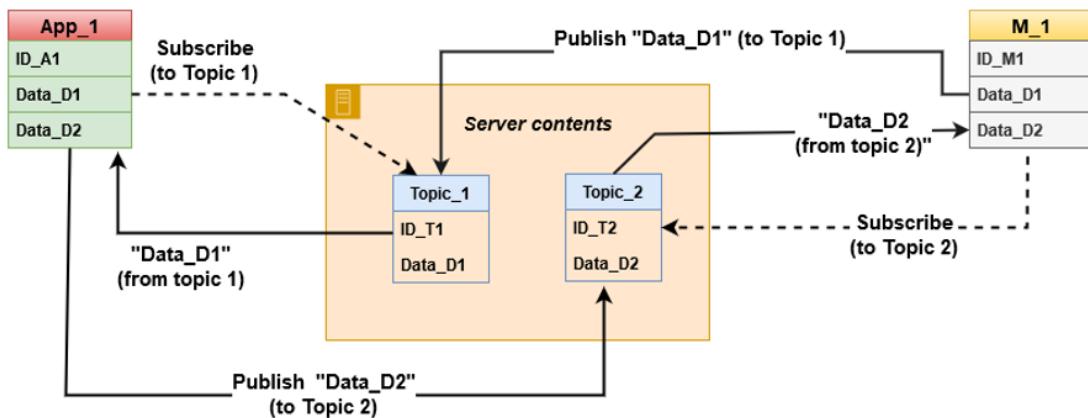
Hai ngôn ngữ HTML và CSS độc lập với nhau và vẫn giữ nguyên như vậy. CSS không nên được viết bên trong một tài liệu HTML và ngược lại. Theo quy định, HTML sẽ luôn đại diện cho nội dung và CSS sẽ luôn thể hiện sự xuất hiện của nội dung đó.

JavaScript (viết tắt là "js") là một ngôn ngữ lập trình mang đầy đủ tính năng của một ngôn ngữ lập trình động mà khi nó được áp dụng vào một tài liệu HTML, nó có thể đem lại khả năng tương tác động trên các trang web. Cha đẻ của ngôn ngữ này là Brendan Eich, đồng sáng lập dự án Mozilla, quỹ Mozilla, và tập đoàn Mozilla.

### 4.2 Database: MongoDB

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL(\*) hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++. Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng. (\*) NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng Transact-SQL để truy vấn thông tin. NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL. CSDL này được phát triển trên Javascript Framework với kiểu dữ liệu JSON. (Cú pháp của JSON là “key:value”) NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,...

### 4.3 Mô hình IOT



Hình 4: Sơ đồ triển khai mô hình IoT

Mô tả tổng quan

Server contents: dùng để truyền các thông điệp (messages) giữa nhiều Client (Publisher và Subscriber). Trong đó nó là điểm giao của tất cả các kết nối đến từ client. Nhiệm vụ của Server Contents là nhận message từ publisher và xếp các message theo hàng đợi rồi chuyển chúng tới một địa chỉ cụ thể.

Publisher và Subscriber làm một trong 2 việc là publish các message lên một topic cụ thể hoặc subscribe một topic nào đó để nhận message từ topic này.

### 4.4 Thuật toán mã hoá Advanced Encryption Standard (Dùng mã hoá mật khẩu tài khoản người dùng)

Advanced Encryption Standard (Tiêu chuẩn mã hóa tiên tiến) là một thuật toán mã hóa khối làm việc với các khối dữ liệu (đầu vào và đầu ra) 128 bit và khóa có độ dài 128, 192 hoặc 256 bit.

Mỗi khối dữ liệu 128 bit đầu vào được chia thành 16 byte (mỗi byte 8 bit), có thể xếp thành 4 cột, mỗi cột 4 phần tử hay là một ma trận 4x4 của các byte, nó được gọi là ma trận trạng thái.

Trong quá trình thực hiện thuật toán các toán tử tác động để biến đổi ma trận trạng thái này dựa vào key cũng là một ma trận 4x4. Bao gồm các bước:

Khởi động vòng lặp:

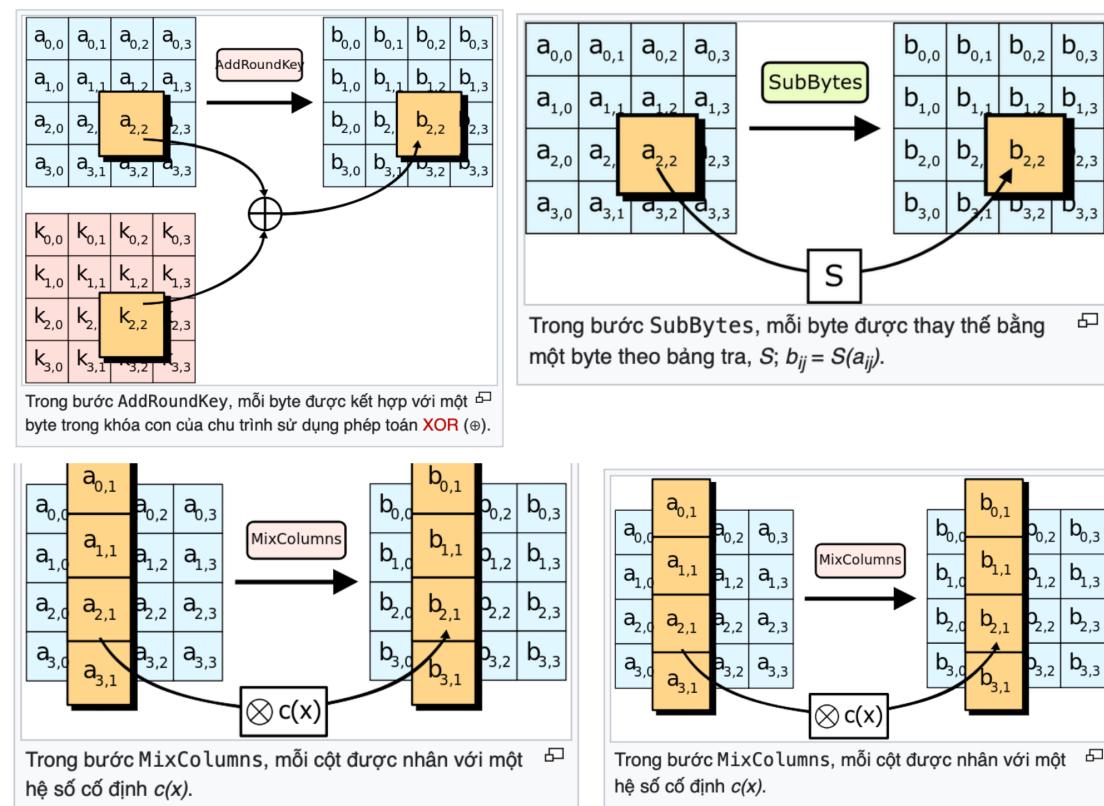
- AddRoundKey — Mỗi cột của trạng thái đầu tiên lần lượt được kết hợp với một khóa con theo thứ tự từ đầu dãy khóa.

Vòng lặp:

- SubBytes — đây là phép thay thế (phi tuyến) trong đó mỗi byte trong trạng thái sẽ được thay bằng một byte khác theo bảng tra (Rijndael S-box).
- ShiftRows — dịch chuyển, các hàng trong trạng thái được dịch vòng theo số bước khác nhau.
- MixColumns — quá trình trộn làm việc theo các cột trong khối theo một phép biến đổi tuyến tính.
- AddRoundKey

Vòng lặp cuối:

- SubBytes
  - ShiftRows
  - AddRoundKey
- (Tại chu trình cuối thì bước MixColumns không thực hiện)



Hình 5: Mô tả các bước của quy trình mã hóa



## 5 Thiết kế hệ thống

Hệ thống dự án gồm có 5 thành phần sau:

- Người dùng
- Trang web
- Server
- Hệ thống đèn chiếu sáng
- Database (MongoDB)

Các thành phần sẽ kết hợp với nhau như hình dưới đây để tạo thành một hệ thống hoàn chỉnh.

### 5.1 Người dùng

Người dùng đóng vai trò quan trọng trong hệ thống, là nhân tố đầu cuối mà hệ thống phục vụ, theo dõi và điều khiển toàn bộ hệ thống thông qua việc tương tác với ứng dụng.

### 5.2 Trang web

Đây là thành phần chủ chốt của hệ thống, với các vai trò:

- Tương tác với người dùng, show các thông số và nhận lệnh từ người dùng.
- Thực hiện subscribe đến các thiết bị để lấy thông số (Thông qua server).
- Xử lý số liệu cũng như nhận lệnh để publish đến thiết bị output (Thông qua server).
- Lấy dữ liệu từ Database về xử lý và lưu dữ liệu mới lên Database.

### 5.3 Hệ thống đèn

Đây là thành phần thực hiện chính của hệ thống. Nó bao gồm hệ thống các đèn chiếu sáng và công tắc.

Nó là Output của hệ thống.

### 5.4 Server

Đây là thành phần xử lý các yêu cầu: subscribe và publish giữa hệ thống đèn và giao diện người dùng.

### 5.5 Database

Database nhóm sử dụng là MongoDB.

Đây là nơi lưu trữ thông tin dữ liệu của trạng thái đèn chiếu sáng tại các thời điểm khác nhau.



## 6 Mô tả chi tiết

### 6.1 Ứng dụng

#### 6.1.1 Đăng nhập và đăng ký

Use-case	Dăng nhập
Actor	Người dùng
Description	Người dùng thực hiện đăng nhập để vào app
Trigger	Người dùng mở app
Pre-condition	Người dùng phải có tài khoản đúng
Post-condition	Người dùng đăng nhập vào app thành công
Normal Flow	<ol style="list-style-type: none"><li>Người dùng nhập username và password</li><li>Người dùng nhấn nút "Login"</li><li>Hệ thống kiểm tra và thực hiện đăng nhập</li></ol>

Bảng 5: Use-case chức năng đăng nhập



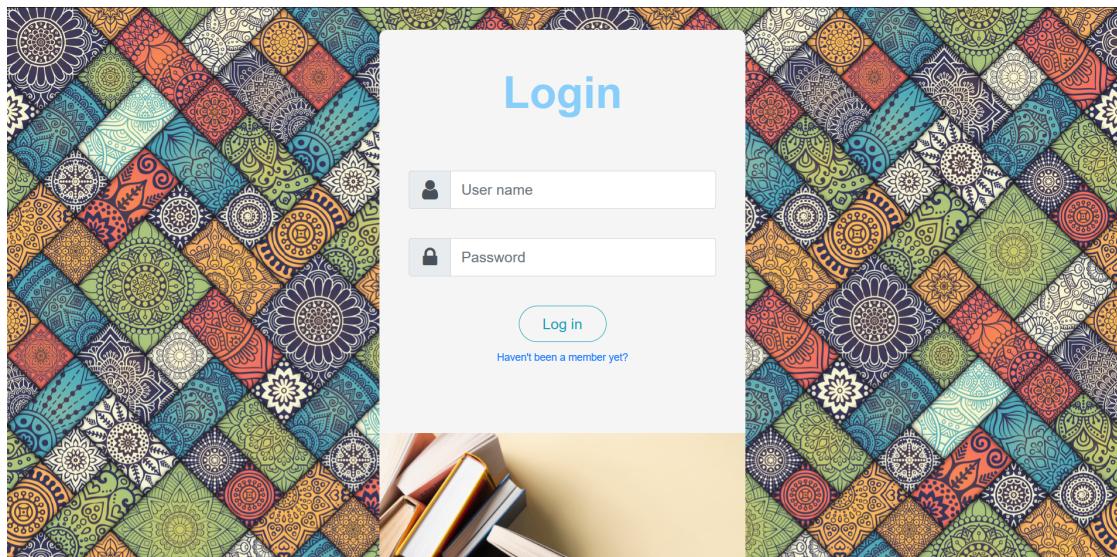
Use-case	Dăng ký
Description	Người dùng thực hiện đăng ký để tạo tài khoản
Trigger	Người dùng nhấn vào nút "Haven't a member yet?" ở màn hình đăng nhập
Pre-condition	Không có
Post- condition	Người dùng có được tài khoản mới
Normal Flow	<ol style="list-style-type: none"><li>Người dùng nhập thông tin tài khoản mong muốn</li><li>Người dùng nhấn nút "Haven't a member yet?"</li><li>Hệ thống kiểm tra và thực hiện đăng ký</li></ol>

Bảng 6: Use-case chức năng đăng ký



Ngay khi mở ứng dụng, ta sẽ được đưa tới màn hình đăng nhập. Nếu đã có tài khoản, ta sẽ nhập tài khoản và nhấn đăng nhập.

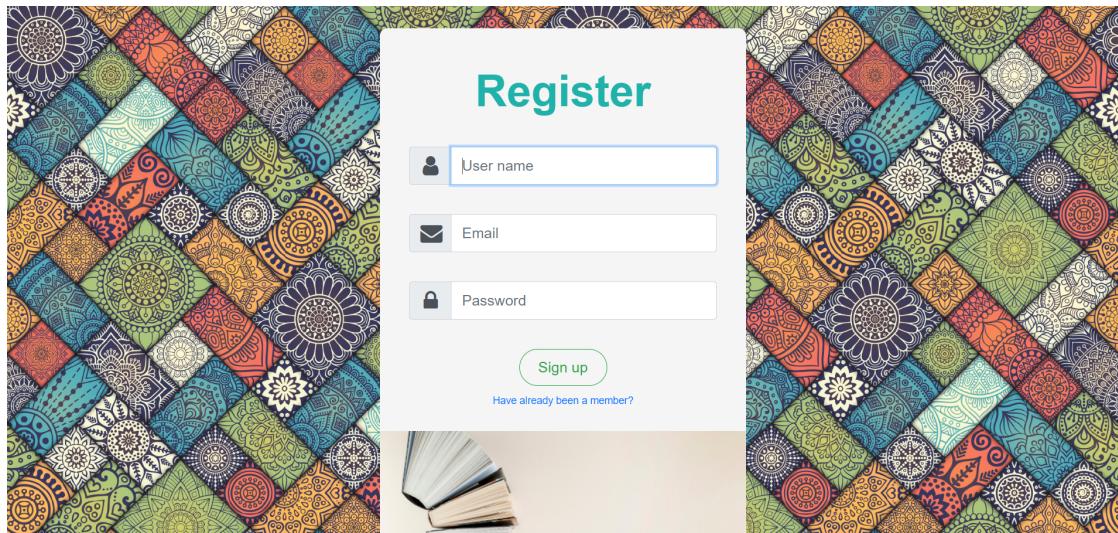
Ứng dụng sẽ kiểm tra xem ta đã nhập đúng yêu cầu chưa. Nếu đúng ứng dụng sẽ truy cập vào Database và kiểm tra xem tài khoản và mật khẩu có đúng hay không. Nếu đúng, đăng nhập thành công. Nếu sai, ứng dụng sẽ báo lỗi cụ thể để người dùng kiểm tra.



Hình 6: Giao diện đăng nhập



Nếu chưa có tài khoản, ta sẽ bấm nút đăng kí và ứng dụng sẽ chuyển đến màn hình đăng kí.



Hình 7: Giao diện đăng kí

Sau khi được đưa đến màn hình đăng kí, ta thực hiện nhập các thông tin và nhấn đăng kí.

Ứng dụng sẽ kiểm tra xem ta đã nhập đúng yêu cầu hay chưa. Nếu đúng ứng dụng sẽ truy cập Database và kiểm tra xem Username yêu cầu đăng kí đã tồn tại hay chưa. Nếu đã tồn tại, ứng dụng sẽ báo lỗi và yêu cầu người dùng chọn Username khác. Nếu chưa tồn tại ứng dụng sẽ tạo tài khoản với thông tin tương ứng và lưu vào Database.



Database tài khoản người dùng sẽ có dạng như sau:

```
_id: ObjectId("5edda699fc4b6180547aef7e")
name: "admin"
pwd: "admin"
email: "a@b.c"
access: 1
```

```
_id: ObjectId("5ede09b6badf8443d6a5d717")
name: "a"
pwd: "a"
email: ""
access: 0
```

Hình 8: Database tài khoản người dùng

Khi vừa tạo tài khoản, tài khoản sẽ được lưu vào phần name và pwd. Riêng phần email, đây chỉ là một phần không bắt buộc có đối với mỗi tài khoản. Về thuộc tính access dùng để cài đặt quyền admin, tuy nhiên nhóm chưa phát triển.



### 6.1.2 Trang chủ

Use-case	Trang chủ
Actor	Người dùng
Description	Người dùng sau khi đăng nhập thành công sẽ vào trang chủ
Pre-condition	Người dùng đăng nhập thành công
Post-condition	Hiển thị trang chủ
Normal Flow	<ol style="list-style-type: none"><li>Người dùng đăng nhập thành công từ trang Login</li><li>Hệ thống hiển thị giao diện trang chủ với tab Overview mặc định</li><li>Người dùng có thể chọn các tab khác hoặc tiến hành Logout</li></ol>

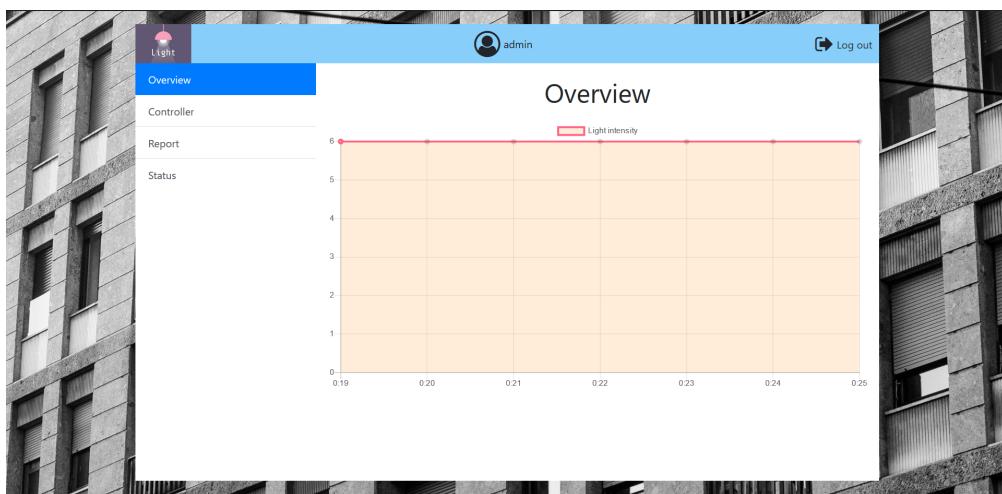
Bảng 7: Use-case trang chủ

Sau khi đăng nhập thành công, người dùng sẽ được đưa đến giao diện trang chủ. Tab mặc định xuất hiện là Overview.

Trang chủ với thanh trên cùng gồm biểu tượng sensor được điều khiển, người dùng và chức năng Log out.

Các tab như Overview, Controller, Report và Status ở bên trái của trang chủ.

Trung tâm của trang chủ sẽ hiển thị dữ liệu, thông tin cũng như tiện ích cho từng tab.



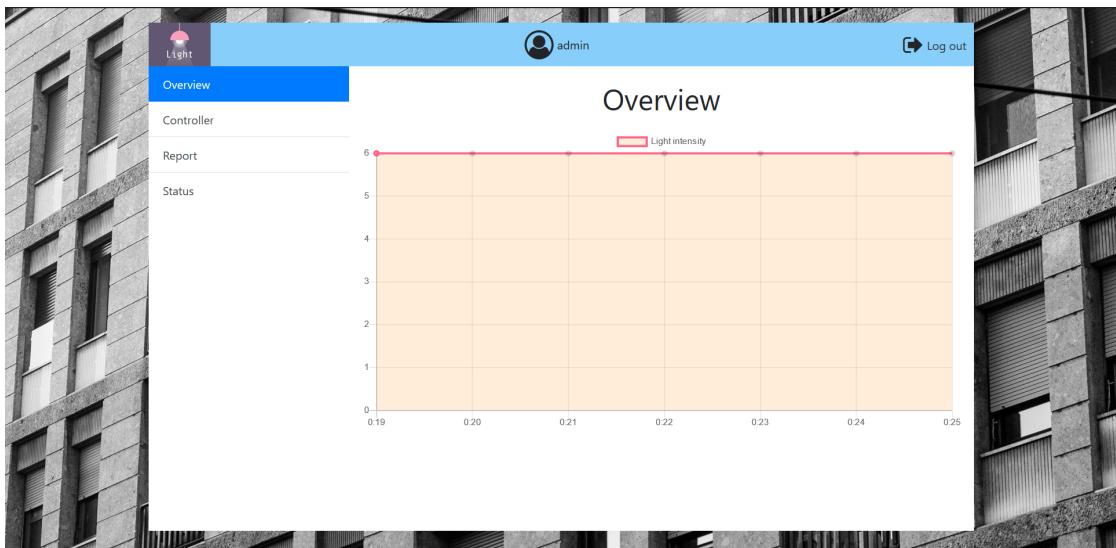
Hình 9: Giao diện trang chủ



### 6.1.3 Tab Overview

Use-case	Xem Overview
Actor	Người dùng
Description	Người dùng theo dõi các thông kê về thông số môi trường lấy từ thiết bị trong kho theo thời gian thực
Trigger	Người dùng nhấn vào tab Overview ở trang chủ
Pre-condition	Không có
Post-condition	Tab Overview được hiển thị, các biểu đồ thống kê của thiết bị được hiển thị
Normal Flow	<ol style="list-style-type: none"><li>Người dùng vào chức năng Overview</li><li>Ứng dụng lấy thông tin từ server và hiển thị các thông số lên màn hình</li><li>Ứng dụng dùng các thông số đã lấy để vẽ biểu đồ và hiển thị</li></ol>

Bảng 8: Use-case xem Overview



Hình 10: Giao diện tab Overview

Tại đây người dùng sẽ theo dõi thông số từ các thiết bị theo thời gian thực.

Ở hình ảnh trên, cường độ ánh sáng là 6 duy trì từ giây 19 đến giây 25 dựa vào đồ thị.



Để thực hiện tự động lấy dữ liệu, mã nguồn dưới đây dùng để lấy thông số cảm biến sau mỗi 5 phút và thực hiện tính toán để lấy thông số cảm biến và thực hiện tính toán).

```
broker_address = "52.187.125.59"
print("creating new instance")
client = mqtt.Client("P1") # create new instance
client.username_pw_set(username='BKvm', password='Hcmut_CSE_2020')
client.on_message = on_message # attach function to callback
print("connecting to broker")
client.connect(broker_address, port=1883, keepalive=500)
client.on_connect = on_connect
client.on_disconnect = on_disconnect # connect to broker
print("connected")
client.loop_start() # start the loop
print("Subscribing to topic", "Topic/Light")
client.subscribe("Topic/Light")
time.sleep(1)
client.loop_stop()

# Thread loop getting data every 5 minutes
loop_time = 19

def loop_get_data():
    # client.connect(broker_address, port=1883) # connect to broker
    client.loop_start()
    client.subscribe("Topic/Light")
    time.sleep(1)
    client.loop_stop()
    threading.Timer(loop_time, loop_get_data).start()

threading.Timer(loop_time, loop_get_data).start()
```



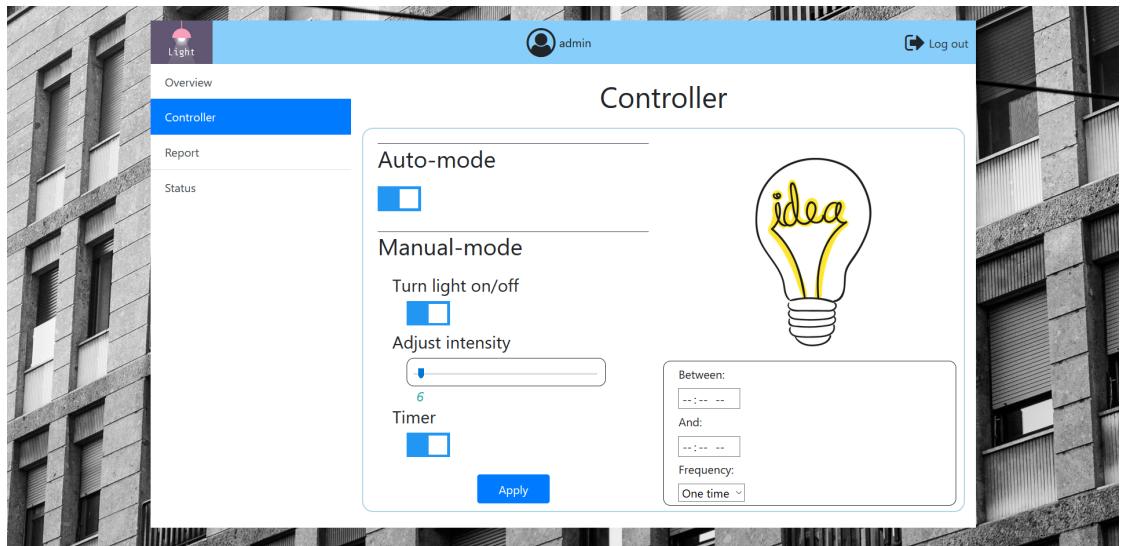
#### 6.1.4 Tab Controller

Use-case	Xem Controller
Actor	Người dùng
Description	Người dùng muốn thao tác, điều khiển các sensor
Trigger	Người dùng chọn tab Controller
Pre-condition	Không
Post-condition	Các chức năng của sensor được tùy chỉnh
Normal Flow	<ol style="list-style-type: none"><li>Người dùng chọn tab Controller</li><li>Ứng dụng hiển thị tình trạng hiện tại của sensor</li><li>Người dùng tùy chỉnh, điều khiển các sensor này</li><li>Người dùng chọn "Apply" để lưu tùy chọn</li></ol>
Exceptions	Không có

Bảng 9: Use-case xem Controller



Dưới đây là giao diện của tab Controller:



Hình 11: Tab Controller

Các chức năng chính bao gồm: Auto-mode và Manual-mode.

- Auto-mode: là chế độ mặc định. Tính năng này sẽ tự động chỉnh độ sáng khi độ sáng dưới mức cần thiết.
- Manual-mode: là chế độ người dùng tự chỉnh khi chế độ Auto-mode được tắt đi.
  - + Turn light on/off: người dùng có thể bật tắt thiết bị.
  - + Adjust intensity: người dùng có thể thay đổi cường độ sáng của đèn.
  - + Timer: Người dùng có thể chỉnh thời gian ở bên phải giao diện với khoảng thời gian từ bắt đầu (between) đến kết thúc (and) và với lịch thường xuyên (frequency) như một lần, lặp lại mỗi ngày, mỗi tuần,...



### 6.1.5 Xem tab Report

Use-case	Xem tab Report
Actor	Người dùng
Description	Người dùng xem lịch sử các thiết bị
Trigger	Người dùng chọn tab Report
Pre-condition	Không có
Post-condition	Tab Report được hiển thị
Normal Flow	<ol style="list-style-type: none"><li>1. Người dùng vào chức năng Report</li><li>2. Ứng dụng hiển thị lịch sử các thiết bị</li></ol>
Exceptions	Không có

Bảng 10: Use-case xem tab Report



Tại giao diện bên dưới, có hai tab nhỏ là Sensor và Controller ứng với lần lượt là dữ liệu được nhận về từ sensor và người dùng gửi đi qua controller.

Người dùng có thể chọn ngày ở bên dưới hai tab nhỏ này để lọc ra ngày muốn truy vấn.

Phía bên phải là lịch sử dữ liệu được gửi hoặc người dùng tùy chỉnh controller. Dựa vào hình vẽ, bảng lịch sử này gồm tên thiết bị và thông số được người dùng tùy chỉnh cũng như là lúc thiết lập và đặt lịch cụ thể cho thiết bị đó.

#	Device	Value	Setting time	Schedule
1	LightD	1,128	01:58:43	{"start":"02:00","duration":2,"freq":0}
2	LightD	1,128	01:59:58	{"start":"02:05","duration":5,"freq":0}
3	LightD	1,128	02:01:50	{"start":"02:05","duration":5,"freq":0}
4	LightD	1,128	02:02:33	{"start":"02:05","duration":5,"freq":0}
5	LightD	1,128	02:03:11	{"start":"02:05","duration":5,"freq":0}
6	LightD	1,128	02:03:44	{"start":"02:08","duration":2,"freq":0}
7	LightD	1,128	02:03:57	{"start":"02:08","duration":2,"freq":0}
8	LightD	1,128	02:05:14	{"start":"02:08","duration":2,"freq":0}
9	LightD	1,128	02:06:40	{"start":"02:08","duration":2,"freq":0}
10	LightD	1,128	02:07:24	{"start":"02:10","duration":5,"freq":0}
11	LightD	1,128	02:09:54	{"start":"02:10","duration":5,"freq":0}
12	LightD	1,128	02:11:07	{"start":"02:10","duration":5,"freq":0}
13	LightD	1,128	02:11:38	{"start":"02:10","duration":5,"freq":0}

Hình 12: Giao diện xem tab Report



#### 6.1.6 Xem tab Status

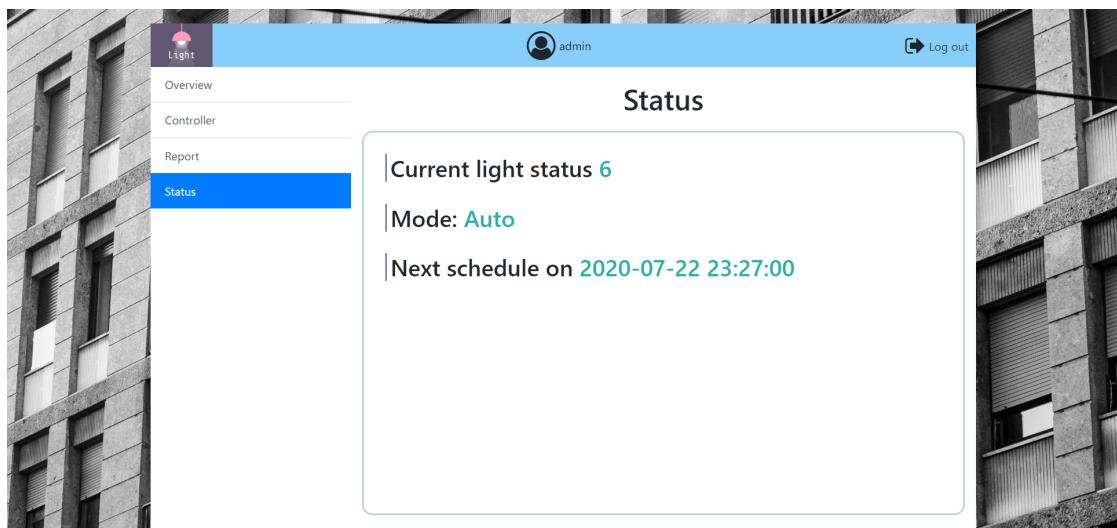
Use-case	Xem tab Status
Actor	Người dùng
Description	Người dùng xem trạng thái hiện tại của thiết bị
Trigger	Người dùng chọn tab Status
Post-condition	Không có
Normal Flow	<ol style="list-style-type: none"><li>Người dùng vào chức năng Status</li><li>Ứng dụng hiển thị trạng thái các thiết bị</li></ol>
Exceptions	Không có

Bảng 11: Use-case xem tab Status



Tại giao diện Status, người dùng thể biết các thông tin về độ sáng hiện tại, chức năng hiện tại là auto hay manual và có lịch hoạt động của nó hay không.

Üng với hình bên dưới, ta thấy thiết bị hiện tại với độ sáng là 6, với chức năng auto và lịch chạy kế tiếp là vào ngày 22/7.



Hình 13: Giao diện xem tab Status



## 6.2 Server, input và output

### 6.2.1 Kết nối server

Dầu tiên, chúng ta sử dụng file *connect.py* để kết nối với server giúp thực hiện các chức năng Publisher và Subscriber.

Listing 1: “connect.py”

```
import paho.mqtt.client as mqtt # import the client
import time
import threading
import db
from datetime import timedelta, datetime
import insertdata
import json

#####
recv_data = 0
save_time = 0
flag_connected = 0

def on_message(client, userdata, message):
    print("message received ", str(message.payload.decode("utf-8")))
    print("message topic=", message.topic)
    print("message qos=", message.qos)
    print("message retain flag=", message.retain)
    global recv_data
    global save_time
    save_time += 1
    recv_data = str(message.payload.decode("utf-8")).replace("\n", "").replace(" ", "")
    print("reodata", recv_data)
    print(save_time)
    # Insert data to database
    if save_time >= 5:
        # print(threading.active_count())
        insertdata.insert_data(recv_data)
        save_time -= 5
        # print(a)

def on_publish(client, userdata, mid):
    print("publish message result ", mid)

def on_connect(client, userdata, flags, rc):
    global flag_connected
    flag_connected = 1

def on_disconnect(client, userdata, rc):
    global flag_connected
    flag_connected = 0
```



```
#####
broker_address = "52.187.125.59"
print("creating new instance")
client = mqtt.Client("P1") # create new instance
client.username_pw_set(username='BKvm', password='Hcmut_CSE_2020')
client.on_message = on_message # attach function to callback
print("connecting to broker")
client.connect(broker_address, port=1883, keepalive=500)
client.on_connect = on_connect
client.on_disconnect = on_disconnect # connect to broker
print("connected")
client.loop_start() # start the loop
print("Subscribing to topic", "Topic/Light")
client.subscribe("Topic/Light")
time.sleep(1)
client.loop_stop()
# client.disconnect()

# Thread loop getting data every 5 minutes
loop_time = 19

def loop_get_data():
    # client.connect(broker_address, port=1883) # connect to broker
    client.loop_start()
    client.subscribe("Topic/Light")
    time.sleep(1)
    client.loop_stop()
    threading.Timer(loop_time, loop_get_data).start()

threading.Timer(loop_time, loop_get_data).start()
```



### 6.2.2 Input

Thông tin input nhận được sẽ có dạng JSON như sau.

Listing 2: “input”

```
[  
  { "device_id": "<device-name>", "values": ["<status>", "<value>"] }  
]
```

Trong đó, <device-name> là tên thiết bị <status> là giá trị trạng thái bật hay tắt của thiết bị với giá trị "0" là tắt và "1" là bật, <value> là giá trị mà ở đây nhóm triển khai thiết bị Light với giá trị từ "0" đến "255" tương đương với cường độ sáng.

### 6.2.3 Output

Thông tin output nhận được sẽ có dạng JSON như sau.

Listing 3: “output”

```
[  
  { "device_id": "<Device-name>", "values": ["<status>", "<value>"] }  
]
```

Các giá trị tương tự như input.

Khi thực hiện Publish ta gửi lên server file JSON có định dạng tương tự đã được chuyển về String.



## 7 Tổng kết

### 7.1 Kết quả

Sau 12 tuần làm việc cùng nhau, nhóm đã đạt được những kết quả đúng như mong đợi, cụ thể như sau:

- Nhóm đã hoàn thành hoàn chỉnh ứng dụng, giống tới 95% mục tiêu đề ra lúc đầu.
- Thực hiện demo thành công, kết nối và điều khiển được các thiết bị.
- Có thêm kiến thức chuyên môn về IOT, thiết kế ứng dụng,... và nhiều kỹ năng mềm như làm việc nhóm, tổ chức phát triển dự án.

#### 7.1.1 Hạn chế

Tuy nhiên vẫn còn tồn tại một vài hạn chế đến từ dự án và hoạt động nhóm chưa được khắc phục:

- Một số ít tính năng chưa được như mong muốn vì giới hạn năng lực và sự đầu tư chưa được tối đa.
- Việc demo chỉ mang tính tương đối vì giới hạn về thiết bị.
- Nhóm đôi lúc trì trệ công việc và phân bổ thời gian chưa thực sự tốt.



## 8 Hướng phát triển

Sau khi hoàn thành dự án, nhóm nhận thấy đây là dự án rất có tiềm năng, mang nhiều lợi ích thực tế và thị trường rất rộng lớn.

Vì vậy, nhóm đã đề ra rất nhiều hướng phát triển và hoàn thiện dự án. Cụ thể như sau:

- Trước hết phát triển ứng dụng trên đa nền tảng chứ không chỉ web để mọi người dùng đều có thể sử dụng.
- Tăng cường thêm nhiều tính năng, nhiều tùy chọn tự động hóa.
- Liên kết với công ty cung cấp thiết bị để tối ưu hóa hệ thống IOT cũng như dễ dàng lắp đặt cho người dùng cuối.
- Tích hợp các công nghệ mới như AI để phân tích số liệu, định hướng phát triển, phát triển tính năng dựa trên thói quen người dùng hoặc dựa trên hướng phát triển chung của ngành.



## Tài liệu

- [1] Tài liệu được cung cấp từ môn học
  - Manual-for-Android-device.docx
  - Sơ đồ triển khai mô hình IoT.docx
  - WorkflowData-format.docx
- [2] Hệ thống quản lý kho  
[https://vi.wikipedia.org/wiki/Hệ\\_thống\\_quản\\_lý\\_kho](https://vi.wikipedia.org/wiki/Hệ_thống_quản_lý_kho)
- [3] Internet of things  
[https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)
- [4] Advanced Encryption Standard  
[https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)  
<https://howtодоinjava.com/security/aes-256-encryption-decryption>
- [5] Lập trình Android  
<https://www.youtube.com/playlist?list=PLzrVYRai0riSRJ3M3bifVWWRq5eJMu6tv>  
[https://www.youtube.com/watch?v=bVXT\\_AkHIEQ&t=645s](https://www.youtube.com/watch?v=bVXT_AkHIEQ&t=645s)
- [6] Design UI  
<https://www.youtube.com/watch?v=LD2zsCAAVXw>  
<https://www.youtube.com/watch?v=L0cD1evBcSA>  
<https://www.youtube.com/watch?v=2pirZvqXza0>  
<https://www.youtube.com/watch?v=Uur6-64KqxI>  
<https://www.youtube.com/watch?v=fqZDL8XsWHc>  
<https://www.flaticon.com>
- [7] Firebase on Android  
<https://firebase.google.com/docs/database/android/start>  
[https://www.youtube.com/playlist?list=PLzrVYRai0riTLPLclyGuByHvZ8\\_tDZZIr](https://www.youtube.com/playlist?list=PLzrVYRai0riTLPLclyGuByHvZ8_tDZZIr)  
[https://www.youtube.com/watch?v=djW-f\\_8QIWc](https://www.youtube.com/watch?v=djW-f_8QIWc)
- [8] Tiêu chuẩn nhiệt độ các loại thực phẩm  
<https://kholanhnambac.com/nhung-tieu-chuan-ve-nhiet-do-bao-quan-trong-kho-lanh>
- [9] Thời tiết  
<https://www.accuweather.com>