

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN THIẾT BỊ ĐIỆN



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG SMART HOME  
SỬ DỤNG MẠNG KHÔNG DÂY ZIGBEE

SVTH: DƯƠNG QUỐC HOÀNG

MSSV: 1711377

GVHD: TS. TRỊNH HOÀNG HƠN

HỆ THỐNG SMART HOME SỬ DỤNG MẠNG KHÔNG DÂY ZIGBEE

2023

TP. HỒ CHÍ MINH, THÁNG 06 NĂM 2023

## *Lời cảm ơn*

Đầu tiên, Em xin chân thành cảm ơn các thầy, cô giáo trong Trường ĐHBK Tp.HCM nói chung và các thầy, cô giáo Khoa Điện – Điện tử nói riêng đã giảng dạy em trong thời gian những năm em học ở trường ĐHBK. Giúp em có kiến thức nền tảng để phát triển và học tập tốt trong suốt quá trình học tập.

Em chân thành biết ơn đến Thầy Trịnh Hoàng Hợp, người đã quan tâm giúp đỡ và tận tình hướng dẫn em hoàn thành đồ án này trong thời gian qua, người đã đưa ra đề nghị em đến với đề tài lần này, người đã luôn kiên nhẫn giải đáp những thắc mắc của em trong quá trình làm đồ án này.

Em chân thành biết ơn đến gia đình của em, những người đã không ngừng hỗ trợ, khuyến khích và thúc đẩy em cố gắng trong quá trình học tập để em có được những kiến thức như ngày hôm nay.

Em xin chân thành cảm ơn đến những người bạn đã hỗ trợ em hết mình trong quá trình làm đồ án môn học của mình, giúp em có thêm những kiến thức khác từ trong quá trình học tập và cả trong quá trình làm việc.

Em chân thành cảm ơn tới những người đã hiểu và quan tâm đến em !

## *Tóm tắt đề tài*

Đề tài này trình bày về các vấn đề chính sau:

- Giới thiệu về nhà thông minh.
- Giới thiệu về các chuẩn giao tiếp.
- Thiết kế tổng quan về mô hình nhà thông minh.
- Thiết kế phần cứng, phần mềm cho các thiết bị.
- Thiết kế trang web tương tác với người dùng.
- Thiết kế giao tiếp giữa trang web và các thiết bị.
- Thiết kế ứng dụng cho một nhà cấp 4 ở Việt Nam

## Mục lục

<b>Chương 1. Giới thiệu đề tài.....</b>	<b>1</b>
<b>1.1. Tổng quan về Nhà thông minh .....</b>	<b>1</b>
<b>1.2. Một số tính năng của Nhà thông minh.....</b>	<b>1</b>
1.2.1. Tiết kiệm năng lượng .....	1
1.2.2. Ánh sáng thông minh.....	1
1.2.3. Đảm bảo an ninh cho ngôi nhà .....	2
<b>Chương 2. Thiết kế tổng quan của hệ thống .....</b>	<b>3</b>
<b>2.1. Lựa chọn giao tiếp các thiết bị trong nhà .....</b>	<b>3</b>
<b>2.2. Thiết kế về giao diện giao tiếp với người dùng.....</b>	<b>4</b>
<b>2.3. Thiết kế tổng quan của hệ thống.....</b>	<b>5</b>
<b>Chương 3. Thiết kế các thiết bị trong nhà .....</b>	<b>7</b>
<b>3.1. Giao tiếp Zigbee.....</b>	<b>7</b>
3.1.1. Cơ sở lý thuyết .....	7
3.1.2. Thiết kế giao tiếp giữa các thiết bị.....	9
<b>3.2. Thiết kế các thiết bị.....</b>	<b>12</b>
3.2.1. Thiết kế các khối dùng chung .....	12
3.2.2. Thiết kế bộ điều khiển trung tâm Hub .....	13
3.2.3. Thiết kế Node Sensor.....	17
3.2.4. Thiết kế Node Button.....	19
3.2.5. Thiết kế Node Relay .....	20
3.2.6. Thiết kế Node Relay ADE .....	22
<b>Chương 4. Thiết kế Website .....</b>	<b>28</b>
<b>4.1. Front-end .....</b>	<b>28</b>
4.1.1. Giới thiệu .....	28
4.1.2. Các ngôn ngữ cơ bản .....	29
4.1.3. Các thư viện sử dụng .....	30
<b>4.2. Back-end.....</b>	<b>34</b>
4.2.1. Giới thiệu .....	34
4.2.2. Vai trò .....	34
4.2.3. NodeJs.....	35
4.2.4. ExpressJs.....	36
<b>4.3. Database .....</b>	<b>37</b>
4.3.1. Giới thiệu .....	37

4.3.2. MongoDB .....	38
4.3.3. Thiết kế .....	39
<b>4.4. Trang web thiết kế .....</b>	<b>40</b>
4.4.1. Giao diện chung của Trang Web .....	40
4.4.2. Trang Đăng nhập, đăng ký.....	43
4.4.3. Trang chính Home .....	43
4.4.4. Trang thành viên trong nhà.....	44
4.4.5. Trang tất cả các phòng .....	45
4.4.6. Trang tất cả các thiết bị.....	47
4.4.7. Trang các Node.....	47
4.4.8. Trang hiển thị các chart .....	48
<b>4.5. Chức năng phân quyền người dùng .....</b>	<b>49</b>
<b>Chương 5. Thiết kế kết nối giữa thiết bị và Website .....</b>	<b>50</b>
<b>5.1. Giới thiệu về giao thức MQTT.....</b>	<b>50</b>
5.1.1. Khái niệm.....	50
5.1.2. Cách hoạt động .....	51
<b>5.2. Thiết kế dữ liệu truyền thông qua MQTT .....</b>	<b>52</b>
5.2.1. Thiết kế MQTT Broker, MQTT Client .....	52
5.2.2. Thiết kế cấu trúc gói tin .....	55
<b>Chương 6. Thiết kế nhà thông minh cho nhà cấp 4 ở Việt Nam .....</b>	<b>59</b>
<b>6.1. Cấu trúc của ngôi nhà thiết kế.....</b>	<b>59</b>
<b>6.2. Thiết kế vị trí lắp đặt các Node .....</b>	<b>60</b>
6.2.1. Phòng khách.....	60
6.2.2. Phòng Ngủ .....	61
6.2.3. Phòng ăn .....	62
6.2.4. Phòng tắm .....	63
<b>6.3. Các bước khởi tạo ban đầu trên trang web .....</b>	<b>63</b>
6.3.1. Truy cập trang web, đăng ký và đăng nhập .....	64
6.3.2. Tạo home .....	64
6.3.3. Thêm tất cả các Node .....	65
6.3.4. Thêm phòng và thêm thiết bị.....	65
6.3.5. Liên kết thiết bị với relay, và liên kết với nút nhấn.....	66
<b>6.4. So sánh thiết kế giữa truyền thống và smart home.....</b>	<b>67</b>
6.3.1. Tính an toàn .....	68
6.3.2. Tính kỹ thuật.....	68

6.3.3. Tính kinh tế.....	69
6.3.4. Tính thẩm mỹ.....	70
<b>Chương 7. Kết quả thực hiện.....</b>	<b>72</b>
<b>7.1. Luồng dữ liệu đi trong hệ thống .....</b>	<b>72</b>
<b>7.2. Panel thử nghiệm.....</b>	<b>74</b>
<b>Chương 8. Kết luận và hướng phát triển .....</b>	<b>76</b>
<b>8.1. Kết quả đạt được .....</b>	<b>76</b>
<b>8.2. Hạn chế.....</b>	<b>76</b>
8.2.1. Thời gian .....	76
8.2.2. Vật chất .....	76
8.2.3. Kiến thức.....	76
<b>8.3. Hướng phát triển đề tài .....</b>	<b>77</b>
8.3.1. Tối ưu các Node có sẵn.....	77
8.3.2. Thiết kế thêm các Node mới.....	77
8.3.3. Thiết kế các tính năng mới .....	77
<b>Tài liệu tham khảo .....</b>	<b>79</b>

## Danh mục hình ảnh

Hình 1.1. Smart home .....	1
Hình 2.1. Thiết kế giao tiếp giữa các node.....	3
Hình 2.2. Thiết kế trang web.....	4
Hình 2.3. Tổng quan giao tiếp của hệ thống.....	6
Hình 3.1. Các dạng mô hình mạng trong Zigbee .....	8
Hình 3.2. Giao tiếp giữa VĐK và Module Zigbee.....	9
Hình 3.3. Cấu trúc Frame truyền giao tiếp với Module Zigbee .....	9
Hình 3.4. Miền MT Data trong frame truyền.....	9
Hình 3.5. Frame truyền lệnh SYS_PING.....	10
Hình 3.6. Frame phản hồi từ lệnh SYS_PING.....	10
Hình 3.7. Frame truyền dữ liệu giữa G-Hub và Nodes.....	10
Hình 3.8. Giao tiếp giữa G-Hub và Nodes.....	11
Hình 3.9. Module Zigbee sz1v5.....	12
Hình 3.10. IC AMS1117 3.3V.....	12
Hình 3.11. ESP32.....	13
Hình 3.12. Sơ đồ khối chức năng của G-Hub .....	13
Hình 3.13. RTOS .....	14
Hình 3.14. Các Task chính của ESP32 .....	15
Hình 3.15. Task Zigbee .....	15
Hình 3.16. Task RTC và LCD .....	16
Hình 3.17. Task Wi-Fi .....	16
Hình 3.18. G-Hub Thủ nghiệm .....	16
Hình 3.19. Pic16F628a .....	17
Hình 3.20. Sơ đồ khối chức năng của Node Sensor .....	17
Hình 3.21. MQ-135 .....	17
Hình 3.22. DHT11.....	18
Hình 3.23. Flow chart của Node Sensor.....	18
Hình 3.24. Node Sensor thử nghiệm.....	18
Hình 3.25. Sơ đồ khối chức năng của Node Button .....	19
Hình 3.26. Nút nhấn .....	19
Hình 3.27. Flow chart của Node Button.....	19
Hình 3.28. Node Button thử nghiệm.....	20
Hình 3.29. Sơ đồ khối chức năng của Node Relay .....	20
Hình 3.30. Relay.....	20

<i>Hình 3.31. Flow chart của Node Relay</i> .....	21
<i>Hình 3.32. Node Relay thử nghiệm</i> .....	21
<i>Hình 3.33. Sơ đồ khối chức năng của Node Relay ADE</i> .....	22
<i>Hình 3.34. ADE7753</i> .....	22
<i>Hình 3.35. ade7753 pinout</i> .....	23
<i>Hình 3.36. Khối chuyển dòng điện</i> .....	24
<i>Hình 3.37. Biến dòng ZMCT103C</i> .....	24
<i>Hình 3.38. Biến áp ZN28120BA1</i> .....	25
<i>Hình 3.39. Khối chuyển điện áp</i> .....	25
<i>Hình 3.40. Flow chart của Node Relay ADE</i> .....	25
<i>Hình 3.41. Node Relay ADE thử nghiệm</i> .....	26
<i>Hình 4.1. Front-end gồm các yếu tố tạo nên giao diện của các trang web (Nguồn: CodeSchool)</i> .....	28
<i>Hình 4.2. Javascript and ReactJs</i> .....	31
<i>Hình 4.3. DOM (Nguồn: <a href="https://www.w3schools.com/js/js_dom.asp">https://www.w3schools.com/js/js_dom.asp</a>)</i> .....	32
<i>Hình 4.4. Real DOM and Virtual DOM</i> .....	33
<i>Hình 4.5. MUI</i> .....	33
<i>Hình 4.6. NodeJs</i> .....	35
<i>Hình 4.7. Cấu trúc file</i> .....	36
<i>Hình 4.8. MongoDB</i> .....	38
<i>Hình 4.9. Quan hệ giữa các bảng</i> .....	40
<i>Hình 4.10. Giao diện chung của Trang Web</i> .....	41
<i>Hình 4.11. Phần header</i> .....	41
<i>Hình 4.12. Home selection</i> .....	42
<i>Hình 4.13. Sidebar</i> .....	42
<i>Hình 4.14. Trang Đăng Nhập</i> .....	43
<i>Hình 4.15. Trang Đăng Ký</i> .....	43
<i>Hình 4.16. Trang Home DASHBOARD (1)</i> .....	43
<i>Hình 4.17. Trang Home DASHBOARD (2)</i> .....	44
<i>Hình 4.18. Trang Home ROOMS</i> .....	44
<i>Hình 4.19. Trang các thành viên trong nhà</i> .....	45
<i>Hình 4.20. Trang thêm thành viên</i> .....	45
<i>Hình 4.21. Trang hiển thị tất cả các phòng</i> .....	45
<i>Hình 4.22. Trang của một phòng (2)</i> .....	46
<i>Hình 4.23. Trang của một phòng (1)</i> .....	46
<i>Hình 4.24. Trang của từng thiết bị</i> .....	47

<i>Hình 4.25. Trang tất cả thiết bị</i> .....	47
<i>Hình 4.26. Trang hiển thị tất cả các Node</i> .....	48
<i>Hình 4.27. Trang hiển thị tất cả các chart</i> .....	48
<i>Hình 4.28. View trang web của Host và Admin</i> .....	49
<i>Hình 4.29. View trang web của Manager</i> .....	49
<i>Hình 4.30. View trang Web của User</i> .....	49
<i>Hình 5.1. Cách thức hoạt động của MQTT</i> .....	51
<i>Hình 6.1. Cấu trúc ngôi nhà</i> .....	59
<i>Hình 6.2. Vị trí các thiết bị trong phòng khách</i> .....	60
<i>Hình 6.3. Vị trí lắp đặt các Node trong phòng khách</i> .....	60
<i>Hình 6.4. Vị trí các thiết bị trong phòng ngủ</i> .....	61
<i>Hình 6.5. Vị trí lắp đặt các Node trong phòng ngủ</i> .....	62
<i>Hình 6.6. Vị Trí thiết bị và các Node trong phòng ăn</i> .....	63
<i>Hình 6.7. Trang đăng nhập</i> .....	64
<i>Hình 6.8. Bảng chọn home</i> .....	64
<i>Hình 6.9. Bảng điền thông tin ngôi nhà</i> .....	64
<i>Hình 6.10. Trang Nodes</i> .....	65
<i>Hình 6.11. Thêm phòng</i> .....	65
<i>Hình 6.12. Thêm thiết bị</i> .....	65
<i>Hình 6.13. Link thiết bị với relay</i> .....	66
<i>Hình 6.14. Bảng chọn relay</i> .....	66
<i>Hình 6.15. Danh sách nút nhấn</i> .....	66
<i>Hình 6.16. Trang Thiết bị</i> .....	66
<i>Hình 6.17. thiết kế truyền thống</i> .....	67
<i>Hình 6.18. Thiết kế smart home</i> .....	67
<i>Hình 6.19. Cách thiết kế truyền thống 2 nơi điều khiển [14]</i> .....	69
<i>Hình 6.21. Ví dụ đi dây trong nhà truyền thống</i> .....	70
<i>Hình 6.21. Ví dụ đi dây trong smart home</i> .....	70
<i>Hình 7.1. Luồng dữ liệu đi trong hệ thống khi người dùng tương tác trên trang Web</i> .....	72
<i>Hình 7.2. Ví dụ về dữ liệu truyền đi trong hệ thống khi thêm Node</i> .....	72
<i>Hình 7.3. Trang liên kết relay với button</i> .....	73
<i>Hình 7.4. Dữ liệu gửi đi khi người dùng nhấn nút mà có liên kết</i> .....	73
<i>Hình 7.5. Hình ảnh panel được dùng để kiểm tra</i> .....	74
<i>Hình 7.6. Vị trí lắp đặt các Node trên Panel</i> .....	74

## **Danh mục bảng**

<i>Bảng 2.1. Bảng đánh giá các mạng không dây trong smart home .....</i>	<i>4</i>
<i>Bảng 3.1. Thông số của Pic16F628a .....</i>	<i>17</i>
<i>Bảng 3.2. Bảng thông số của Pic16F887.....</i>	<i>22</i>
<i>Bảng 3.3. Kết quả đo điện áp sánh với đồng hồ đo đa năng Selec MFM383A [1].....</i>	<i>26</i>
<i>Bảng 3.4. Kết quả đo dòng điện so sánh với đồng hồ đo đa năng Selec MFM383A [1].....</i>	<i>27</i>
<i>Bảng 3.5. Kết quả đo công suất so sánh với đồng hồ đo đa năng Selec MFM383A [1] .....</i>	<i>27</i>
<i>Bảng 6.1. Bảng danh sách thiết bị và nơi điều khiển.....</i>	<i>69</i>

# Chương 1. Giới thiệu đề tài

## 1.1. Tổng quan về Nhà thông minh

Ngày nay với sự phát triển rất nhanh về công nghệ với cuộc cách mạng 4.0, Hệ thống IoT đã và đang phát triển rất mạnh mẽ. Dựa theo hệ thống IoT phát triển thì Smart home hay nhà thông minh cũng đang trên đà phát triển rất mạnh mẽ.

Smart home hay Nhà thông minh là kiểu nhà được lắp đặt các thiết bị điện tử có thể được điều khiển



Hình 1.1. Smart home

hoặc tự động hóa hoặc bán tự động hóa. Thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển. Hệ thống điện tử ngày nay giao tiếp với người dùng thông qua bảng điện tử trong nhà, ứng dụng trên điện thoại di động, máy tính bảng hoặc một giao diện web.

Trong một căn nhà thông minh hay Smart home, mọi nơi sẽ được kiểm soát bằng thiết bị điện tử. Chúng sẽ sử dụng các cách giao tiếp riêng để hiểu nhau như: Bluetooth, Zigbee, Z-wave, wifi, KNX, ...

## 1.2. Một số tính năng của Nhà thông minh

### 1.2.1. Tiết kiệm năng lượng

Nhà thông minh giúp tiết kiệm năng lượng vì bây giờ không cần phải đốt tiền bằng cách quên tắt máy lạnh khi bạn đi. Bằng cách theo dõi hệ thống sưởi, nước, điện của bạn, sẽ rất có ít khả năng chi trả hóa đơn tiền điện quá mức.

### 1.2.2. Ánh sáng thông minh

Hệ thống ánh sáng thông minh luôn là điểm nhấn đặc biệt và phổ biến nhất khi dùng Nhà thông minh. Ánh sáng thông minh sẽ giúp bạn rất nhiều trong cuộc sống, ví dụ như:

- Bật/ tắt điện ngay từ điện thoại của bạn dù bạn đang ở bất cứ đâu.
- Hẹn giờ lên lịch hoạt động cho các bóng đèn trong nhà. Và bạn sẽ không phải đi bật/tắt bóng đèn khi trời tối hay tắt chúng vào mỗi sáng.
- Kết hợp cùng các thiết bị điện tử khác ví dụ như các cảm biến. Đây là điểm mà chỉ có các thiết bị chiếu sáng thông minh có thể làm.

### **1.2.3. Đảm bảo an ninh cho ngôi nhà**

Sử dụng các camera thông minh và thiết bị an ninh để bảo vệ căn nhà của bạn. Những camera chất lượng cao thường thiết kết nối trực tiếp với điện thoại của bạn. Và có thể kết hợp với các cảnh báo để thông báo khi có người lạ xuất hiện hay các vấn đề cảnh báo khác nếu có.

## Chương 2.Thiết kế tổng quan của hệ thống

### 2.1. Lựa chọn giao tiếp các thiết bị trong nhà

Trong hệ thống nhà thông minh, các thiết bị điện tử giao tiếp với nhau thông qua một giao tiếp chung, một số giao tiếp như: Z-wave, Bluetooth, Zigbee, wifi...

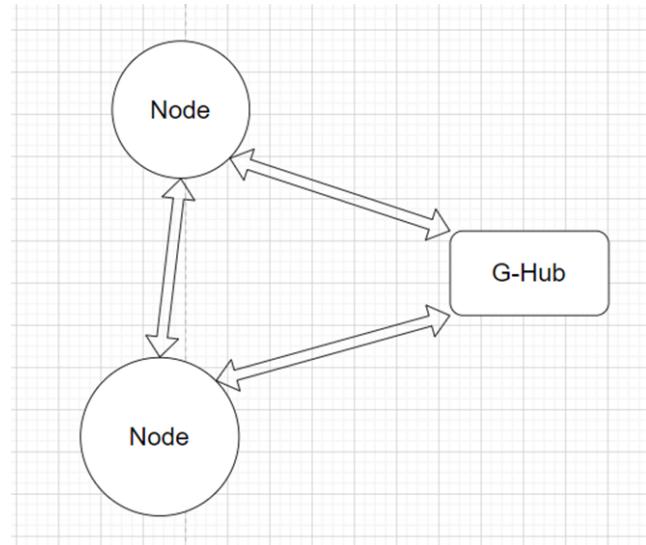
Giới thiệu về các giao tiếp:

- Z-wave: Giao thức kết nối này rất nhanh, sử dụng băng tần 908.42MHz, thấp hơn rất nhiều so với các thiết bị không dây khác trong nhà (sử dụng tần số 2.4Ghz), giúp giảm nhiễu và cải thiện hiệu quả giao tiếp cũng như tốc độ. Tương thích với hơn 1500 thiết bị. Z-wave cũng giúp tiết kiệm năng lượng khi sử dụng.

- Wifi: Các thiết bị tương thích sử dụng băng tần 2.4Ghz và 5Ghz, cho khoảng cách khả dụng thực tế lên đến 20 mét, bị ảnh hưởng bởi các yếu tố ngoại cảnh, độ nhiễu, nguồn phát và chất lượng ăng ten. Có thể kết nối tối đa 256 thiết bị cùng lúc, nhưng có khả năng nghẽn mạng xảy ra. Điểm tuyệt vời là tốc độ truyền phát, khoảng cách nhưng cũng có khả năng bị nhiễu tín hiệu vì có quá nhiều thiết bị và vấn đề tiêu hao năng lượng.

- Zigbee: Giao thức này cho khả năng giao tiếp nhanh, sử dụng tần số vô tuyến tiêu chuẩn IEEE 802.15.14, hoạt động trên các băng tần 2.4GHz, 900MHz và 868MHz. Một trong những lợi ích lớn nhất của giao thức kết nối nhà thông minh này chính là khả năng tiêu thụ điện năng siêu thấp, mang lại hiệu quả cực cao. ZigBee cũng giúp cải thiện bảo mật, là một trong những giao thức an toàn nhất tính đến thời điểm hiện tại, sử dụng mã hóa 128Bit, được áp dụng cho các hệ thống tài chính, ngân hàng.

- Bluetooth: Tính tương thích của giao thức kết nối này cực kỳ tuyệt vời, người dùng có thể kết nối bất kỳ thiết bị Bluetooth nào với thiết bị khác rất dễ dàng. Một trong những điều tuyệt vời nhất của kết nối Bluetooth chính là tiết kiệm năng lượng hơn rất nhiều so



Hình 2.1. Thiết kế giao tiếp giữa các node

với các giải pháp khác, lý tưởng cho những ai quan tâm đến tiêu hao điện năng và môi trường. Tuy nhiên, giao thức kết nối này gặp vấn đề về khoảng cách, thiết bị sẽ không làm việc nếu di chuyển ra khỏi tầm khả dụng, mặc dù các nghiên cứu mới đang được tiến hành, cho phép tiết kiệm năng lượng trong khi mở rộng phạm vi sử dụng.

Dựa theo tiêu chí: Mở rộng sau này, tiết kiệm năng lượng, tính bảo mật, khoảng cách giao tiếp giữa các thiết bị.

Tiêu chí	Z-Wave	Wifi	Zigbee	Bluetooth
Mở rộng	232 thiết bị	256 thiết bị	56554 thiết bị	Số lượng lớn
Năng lượng	Tốt	Tiêu hao lớn	Tốt	Rất Tốt
Bảo mật	Tương đối	Tương đối	cao	Tương đối
Khoảng cách	30m	20m	50 – 100m	Dưới 10m

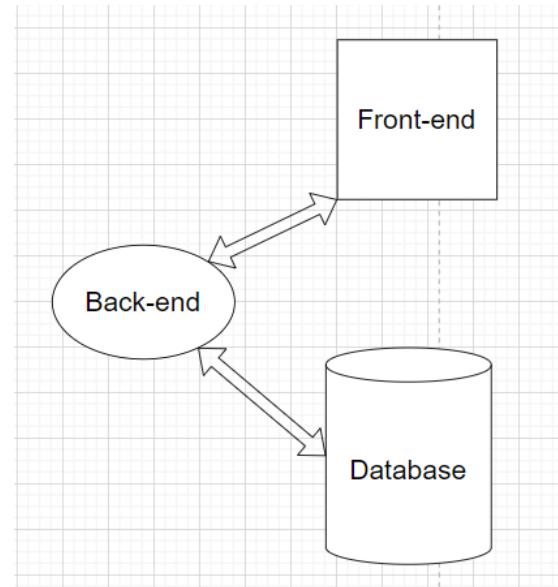
Bảng 2.1. Bảng đánh giá các mạng không dây trong smart home

Dựa theo những tiêu chí trên ta có thể thấy được với số lượng thiết bị kết nối lên tới hơn 56 000 thiết bị và có bảo mật tốt, khoảng cách cao thì lựa chọn mạng Zigbee là hợp lý nhất.

## 2.2. Thiết kế về giao diện giao tiếp với người dùng

Để người dùng có thể giao tiếp với các thiết bị trong nhà thì người dùng phải thông qua một thiết bị khác để giao tiếp với các thiết bị, ví dụ như: Bảng điện tử trong nhà, thông qua một trang web, ứng dụng có sẵn trên điện thoại, ...

Để có tính bảo mật cao thì sẽ loại bỏ các ứng dụng có sẵn trên các thiết bị vì có khả năng rò rỉ thông tin của người dùng rất cao. Sử dụng một bảng điện tử trong nhà thì sẽ mất đi khả năng linh hoạt để điều khiển thiết bị



Hình 2.2. Thiết kế trang web

→ Cho nên lựa chọn thiết kế một trang web để có tính bảo mật cho người dùng, còn có thể giao tiếp linh hoạt ở bất cứ đâu khi có điện thoại hoặc một cái máy tính.

Web sẽ bao gồm 3 phần chính:

- + Front-end là phần giao tiếp được với người dùng.
- + Back-end (server): là phần điều hướng, xử lý dữ liệu.
- + Database: là nơi lưu trữ dữ liệu của ngôi nhà

### **2.3. Thiết kế tổng quan của hệ thống**

Để giao tiếp được giữa các node (các thiết bị trong nhà) và web thì phải thông qua một giao thức khác để hai bên hiểu nhau và giao tiếp với nhau.

Data protocols - Giao thức truyền dữ liệu IoT (Lớp ứng dụng): Hiểu nôm na là hình thức đóng gói dữ liệu và truyền gói dữ liệu đó qua lại giữa các thiết bị trong mạng lưới với nhau. Một số Data protocols phổ biến ứng dụng trong IoT hiện nay:

**MQTT (Message Queue Telemetry Transport):** MQTT là một giao thức theo cơ chế xuất bản/đăng ký, ở đó máy client có thể xuất bản hay nhận bản tin. Nó giúp giao tiếp dễ dàng giữa nhiều thiết bị. Nó là một giao thức nhắn tin đơn giản được thiết kế cho các thiết bị bị hạn chế và có băng thông thấp, vì vậy nó là một giải pháp hoàn hảo cho các ứng dụng internet vạn vật.

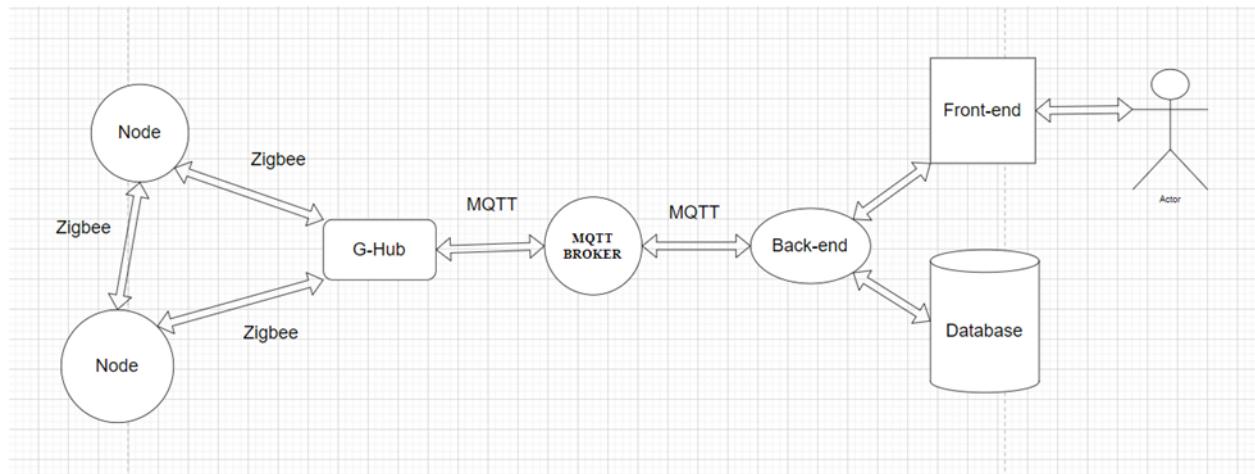
**HTTP (HyperText Transfer Protocol):** cho phép trao đổi thông tin (chủ yếu ở dạng siêu văn bản) qua Internet. HTTP hiện là nền tảng truyền dữ liệu của ứng dụng duyệt web ngày nay và được sử dụng rộng rãi trong hệ thống Internet of Things. Mặc dù giao thức http có nhiều nhược điểm trong việc truyền dữ liệu và không phù hợp bằng các giao thức tối ưu khác như MQTT, CoAP, AMQP áp dụng cho IoT, nhưng giao thức này vẫn phổ biến trong lĩnh vực nhà thông minh cũng như việc sử dụng nhiều trong bộ vi điều khiển và vi xử lý tiên tiến.

**CoAP (Constrained Applications Protocol):** CoAP là một giao thức đơn giản chi phí thấp được thiết kế riêng cho các thiết bị hiệu năng thấp (chẳng hạn như vi điều khiển) và nơi mạng có băng thông thấp. Giao thức này được sử dụng để trao đổi dữ liệu M2M và

rất giống với HTTP. Do đó, CoAP có thể được coi là sự lựa chọn giao thức tốt nhất cho mạng truyền thông gia đình.

**AMQP (Advanced Message Queue Protocol)** là một giao thức làm trung gian cho các gói tin trên lớp ứng dụng với mục đích thay thế các hệ thống truyền tin độc quyền và không tương thích. AMQP là một giao thức có dây (wire-protocol), có khả năng diễn tả các message phù hợp với định dạng dữ liệu, có thể triển khai với rất nhiều loại ngôn ngữ lập trình.

➔ Dựa trên điểm dễ dàng trao đổi dữ liệu, và hoạt động được trên các thiết bị có băng thông thấp nên lựa chọn MQTT.



Hình 2.3. Tổng quan giao tiếp của hệ thống

Như thiết kế ở trên ta có thể tách hệ thống thành 2 phần chính: Là các node trong nhà và phần giao tiếp với người dùng (website). Và kết hợp hai phần đó với nhau thông qua giao thức MQTT để giao tiếp giữa hai thành phần với nhau.

## Chương 3.Thiết kế các thiết bị trong nhà

### 3.1. Giao tiếp Zigbee

#### 3.1.1. Cơ sở lý thuyết

Zigbee [11] là một giao thức truyền thông được phát triển dựa trên chuẩn truyền thông không dây IEEE 802.15.4. Giao thức này được tạo ra nhằm phục vụ cho những ứng dụng yêu cầu giá thành và công suất thấp nhưng phải có khả năng linh động trong phạm vi rộng. Chuẩn Zigbee được phát triển và xúc tiến bởi hãng Zigbee Alliance với sự hỗ trợ hơn 200 công ty trên thế giới.

Tính năng của Zigbee:

- Tính ổn định: Mạng Zigbee hình lưới có đặc điểm tự thích nghi, tức là chúng có khả năng tự xây dựng lại và hoạt động bình thường ngay cả khi có một vài nút bị hỏng, hoặc tìm đường đi khác khi đường đi thông thường bị chặn.
- Tính bảo mật: Chuẩn Zigbee hỗ trợ bảo mật trên nhiều tầng, gồm có tầng xác thực cơ bản, mã hóa AES 128-bits, bảo mật trong cơ cấu hình thành và sát nhập nút.
- Khả năng mở rộng: Cơ chế định địa chỉ 64-bits có thể mở rộng đến hơn 65000 nút mạng, có khả năng bao quát toàn bộ nhà máy.
- Giá thành rẻ: Bao gồm chi phí mua thiết bị, chi phí lắp đặt và chi phí bảo trì. Các thiết bị Zigbee có thể hoạt động bằng pin chính trong vài năm mà không cần bộ sạc, đồng thời các nút có thể hoạt động ở chế độ nghỉ (sleep mode) giúp tiết kiệm đáng kể năng lượng.
- Khả năng hỗ trợ: Chuẩn mở với nhiều nhà cung cấp, hỗ trợ nhiều ứng dụng và ngày càng được cải tiến, phát triển rộng rãi.

Giải tần của Zigbee:

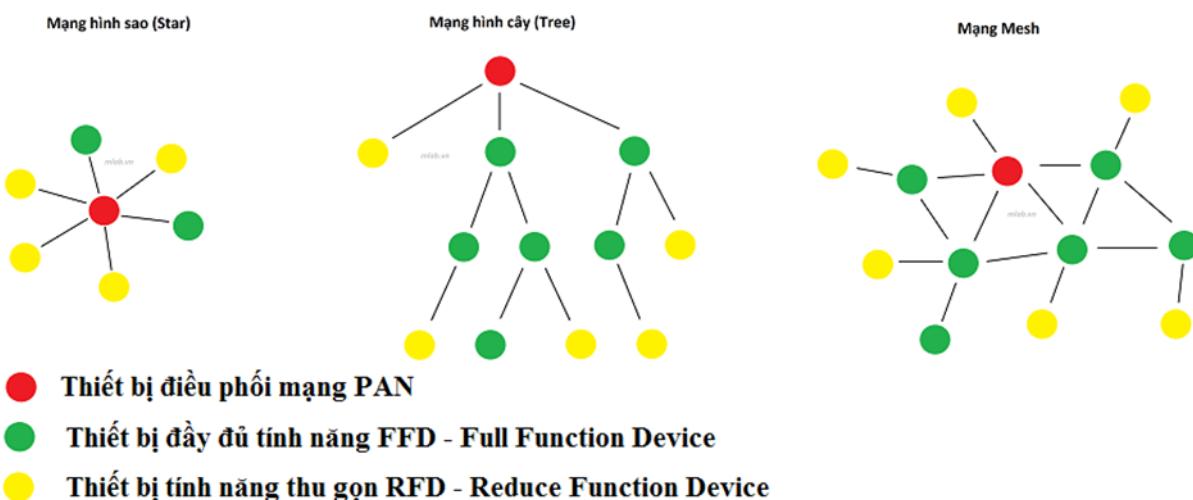
- Khu vực Bắc Mỹ: 915MHz
- Khu vực Châu Âu, Nhật: 868MHz
- Khu vực khác: 2.4GHz

Thành phần trong mạng Zigbee:

- Zigbee Coordinator (ZC): Đây là thiết bị gốc có nhiệm vụ quyết định cấu hình mạng, quy định cách đánh địa chỉ và lưu trữ bảng địa chỉ. Mỗi mạng chỉ có duy nhất một ZC và nó cũng là thiết bị duy nhất nói chuyện được với các mạng khác.
- Zigbee Router (ZR): Thiết bị này có nhiệm vụ định tuyến trung gian trong việc truyền dữ liệu, nó sẽ tự phát hiện và lập bản đồ các nút xung quanh cũng như là theo dõi các điều khiển hoạt động bình thường.
- Zigbee End Device (ZED): Gọi là thiết bị điểm cuối, nó sẽ giao tiếp với ZC, ZR gần nó nhất. Chúng có nhiệm vụ đọc thông tin từ các thành phần vật lý, chúng thường ở trạng thái nghỉ và làm việc khi chuyên nhận thông điệp nào đó.

Mô hình mạng Zigbee:

- Mạng hình sao (Star network): Gồm 1 nút trung tâm ZC, tất cả các nút khác đều được kết nối với nút trung tâm này, mạng hình sao bị hạn chế khoảng cách và sự mở rộng.
- Mạng hình cây (Cluster Tree network): Gồm 1 nút ZC, các nút khác được liên kết với nhau theo mô hình giống một cái rẽ cây, dạng này có khả năng mở rộng cao, tăng khoảng cách và quy mô hệ thống.
- Mạng hình lưới (Mesh network): Gồm 1 nút ZC, các thiết bị trong mạng đều có thể kết nối với nhau (trừ ZED). Khi một đường truyền bị lỗi, sẽ tự động tìm đường truyền khác, tăng tính tin cậy và kết nối trong mạng.



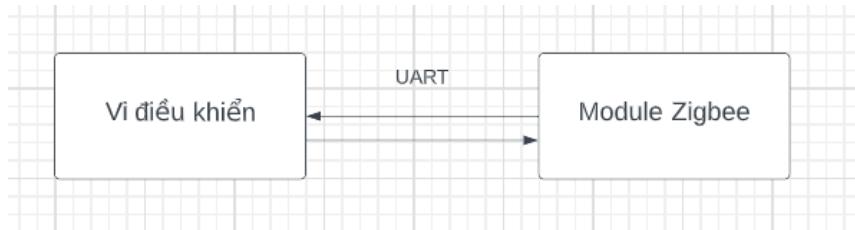
Hình 3.1. Các dạng mô hình mạng trong Zigbee

Ứng dụng của Zigbee: Hướng ưu tiên của Zigbee là mạng cảm biến không dây, ứng dụng giám sát và điều khiển không dây.

### 3.1.2. Thiết kế giao tiếp giữa các thiết bị

#### 3.1.2.1. Frame truyền gói tin giao tiếp với module Zigbee

Để giao tiếp được giữa Vi điều khiển và Module Zigbee thì ta sẽ sử dụng giao tiếp uart để truyền và nhận dữ liệu giữa các thiết bị.



Hình 3.2. Giao tiếp giữa VDK và Module Zigbee

SOF	MT CMD	FCS
Byte: 1	3-256	1

Hình 3.3. Cấu trúc Frame truyền giao tiếp với Module Zigbee

Cấu trúc một Frame truyền gói tin giao tiếp với module Zigbee sẽ gồm 3 thành phần chính:

- SOF (Start of Frame): đây là 1-byte có giá trị 0xFE dùng để xác định việc bắt đầu của mỗi frame truyền.
- MT CMD (Monitor Test Command): là thành phần chứa dữ liệu và lệnh để điều khiển với module Zigbee.
- FCS (Frame Check Sequence): đây là 1-byte dùng để đảm bảo tính toàn vẹn của gói tin. Giá trị của byte này sẽ được tính bằng cách XOR tất cả các byte của MT CMD. Thiết bị khi nhận sẽ dùng byte này so sánh với XOR của các byte trong MT CMD để quyết định lấy dữ liệu này xử lý hay không.

LEN	CMD	DATA
Byte: 1	2	0-250

Hình 3.4. Miền MT Data trong frame truyền

Cụ thể nội dung của MT CMD sẽ bao gồm 3 thành phần chính:

- LEN (Length): đây là 1-byte có giá trị biểu diễn độ dài byte của miền DATA
- CMD (Command): đây là miền 2-byte dùng để xác định loại gói tin khi giao tiếp giữa vi điều khiển và module Zigbee. Các lệnh điều khiển trong gói tin có nhiều loại ví dụ như lệnh cấu hình module, lệnh gửi gói tin, ...
- DATA: đây là miền chứa thành phần dữ liệu mà mình muốn gửi kèm với lệnh ở miền CMD. Đối với từng gói tin thì miền này sẽ có thể có các độ dài khác nhau từ 0~250-byte.

Ví dụ của Frame truyền gói tin với module Zigbee:

<b>SOF</b>	<b>LEN</b>	<b>CMD0</b>	<b>CMD1</b>	<b>DATA</b>	<b>FCS</b>
<b>Byte: 1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b>0xFE</b>	<b>0x00</b>	<b>0x21</b>	<b>0x01</b>	<b>N/A</b>	<b>0x20</b>

Hình 3.5. Frame truyền lệnh SYS\_PING

Lệnh SYS\_PING khi gửi sẽ có dạng 0xFE 0x00 0x21 0x01 0x20

Và phản hồi của lệnh SYS\_PING từ module sẽ có dạng 0xFE 0x02 0x61 0x01 0x11 0x00 0x73 hoặc tương tự như vậy.

<b>SOF</b>	<b>LEN</b>	<b>CMD0</b>	<b>CMD1</b>	<b>DATA0</b>	<b>DATA1</b>	<b>FCS</b>
<b>Byte: 1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>0xFE</b>	<b>0x02</b>	<b>0x61</b>	<b>0x01</b>	<b>0x11</b>	<b>0x00</b>	<b>0x73</b>

Hình 3.6. Frame phản hồi từ lệnh SYS\_PING

Module Zigbee khi nhận gói tin từ các thiết bị Zigbee khác trong mạng sẽ gửi lại gói tin về cho vi điều khiển thông qua giao tiếp UART với cấu trúc Frame truyền đã nêu trên. Khi nhận gói tin đó, vi điều khiển sẽ kiểm tra cấu trúc Frame để xác thực gói tin và trích xuất dữ liệu phần miền DATA ra để xử lý.

### 3.1.2.2. Frame truyền dữ liệu giao tiếp giữa các G-Hub và Nodes

<b>Destination Address</b>	<b>Source Address</b>	<b>Endpoint</b>	<b>Cluster ID</b>	<b>Data</b>
2 bytes	2 bytes	1 byte	2 bytes	N bytes

Hình 3.7. Frame truyền dữ liệu giữa G-Hub và Nodes

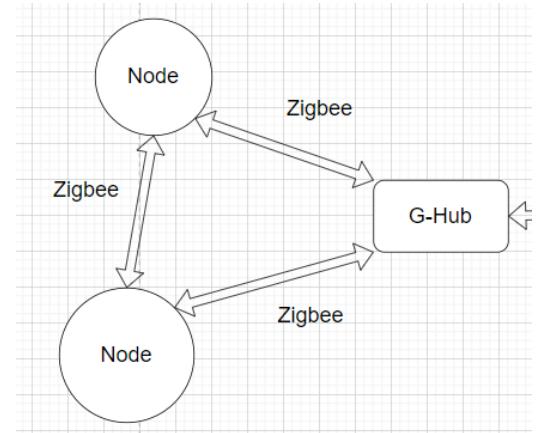
Frame truyền dữ liệu giữa các thiết bị với nhau [2] phía trên là phần dữ liệu trong miền DATA của MT CMD với lệnh CMD tương ứng là truyền nhận dữ liệu. Trong đó sẽ bao gồm các thành phần chính sau:

- Destination Address: Chứa địa chỉ đích của gói tin.
- Source Address: Chứa địa chỉ nguồn của gói tin.
- Endpoint: Là thành phần phân biệt loại gói tin muốn gửi giữa các thiết bị. Với loại gói tin khác nhau thì sẽ có giá trị endpoint riêng.
- Cluster ID: Phân biệt loại chức năng thiết bị. Các loại thiết bị có chức năng khác nhau khi gửi gói tin đến HUB sẽ có các giá trị Cluster ID riêng.
- Data: Phần dữ liệu được truyền đi.

### 3.1.2.3. Phân loại gói tin giao tiếp.

Gói tin truyền dữ liệu giao tiếp giữa Hub và Node sẽ được chia thành 3 loại với các giá trị Endpoint khác nhau:

- Provision: Là loại gói tin của Node dùng để cung cấp dữ liệu thông tin về chức năng cũng như loại thiết bị để gửi đến cho Hub. Gói tin này sẽ được gửi khi thiết bị tham gia vào mạng Zigbee thành công. Giá trị Endpoint: 0x01.
- Telemetry: Là loại gói tin của Node dùng để gửi dữ liệu thông số và trạng thái thiết bị đến Hub nhằm mục đích giám sát. Gói tin này sẽ được gửi khi có dữ liệu cập nhật mới. Giá trị Endpoint: 0x02.
- Command: Là loại gói tin của Hub dùng để gửi các lệnh, yêu cầu điều khiển đến thiết bị Node. Gói tin này được gửi đi khi người dùng thực hiện thao tác điều khiển thiết bị thông qua giao diện. Giá trị Endpoint: 0x03.



Hình 3.8. Giao tiếp giữa G-Hub và Nodes

## 3.2. Thiết kế các thiết bị

### 3.2.1. Thiết kế các khối dùng chung

#### a. Khối giao tiếp Zigbee

Module ZigBee sz1v5 CC2530 2.4G:

- 2.4Ghz IEEE802.15.4 tương thích thu phát RF
- Tốc độ truyền dữ liệu 250kbps
- Bộ nhớ Flash 256KB
- RAM 8KB
- Hỗ trợ giao tiếp UART
- Cấp nguồn từ 2V đến 3.6V



Hình 3.9. Module Zigbee sz1v5

Module Zigbee được sử dụng để giao tiếp giữa bộ Hub với các Node

#### b. Khối nguồn

IC AMS1117 - 3.3V là IC ổn áp có độ ổn định cao, Là IC chuyên dụng để ổn định điện áp tuyến tính với sụt áp thấp. IC có các đặc điểm sau:



Hình 3.10. IC AMS1117 3.3V

- Điện áp ngõ ra: 3.3V
- Dòng ra tối đa: 800mA
- Áp ngõ vào:  $1.5 < V_{in} - V_{out} < 12V$  (4.8-8.7VDC)
- Nhiệt độ hoạt động: âm 40 đến 125 độ C

Sử dụng nguồn Adapter biến đổi điện áp xoay chiều 220V thành 5VDC, sau đó áp 5VDC sẽ được đi vào mạch điện có dùng IC ASM1117 để hạ điện áp xuống còn 3.3VDC. Sau đó sẽ cấp nguồn cho toàn mạch.

### 3.2.2. Thiết kế bộ điều khiển trung tâm Hub

#### 3.2.2.1. Sơ đồ khối chức năng

ESP32-DevKitC V4 (ESP32) là một mạch phát triển dựa trên ESP32 có kích thước nhỏ được sản xuất bởi Espressif. Hầu hết các chân được chia ra thành các đầu ghim ở cả hai bên để dễ dàng giao tiếp. Một số đặc điểm của ESP32-DevKitC V4 như sau:

- Bộ nhớ có 520 KByte SRAM trên chip cho dữ liệu và lệnh.
- Hầu hết các chân trên module ESP đều được đưa ra các đầu ghim trên bo mạch để thuận tiện lập trình ESP32 để kích hoạt nhiều chức năng như PWM, ADC, DAC, I2C, I2S, SPI, v.v.
- Có cổng chuyển đổi USB sang UART nhằm giao tiếp và nạp code dễ dàng.
- Có Webserver riêng, có thể làm trạm phát Wifi hoặc bắt Wifi.
- Cấp nguồn thông qua USB hoặc nguồn ngoài 5V / 3.3V.

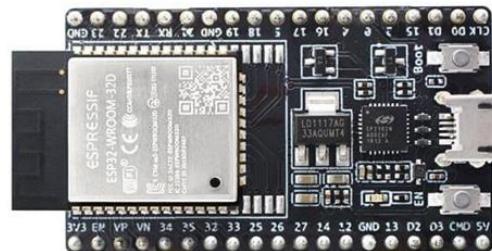
ESP32 được sử dụng để:

- Kết nối WiFi
- Gửi thông số thiết bị đo được thông qua WiFi để đến giao diện người dùng.
- Làm vi điều khiển chính thực hiện các chức năng sau:

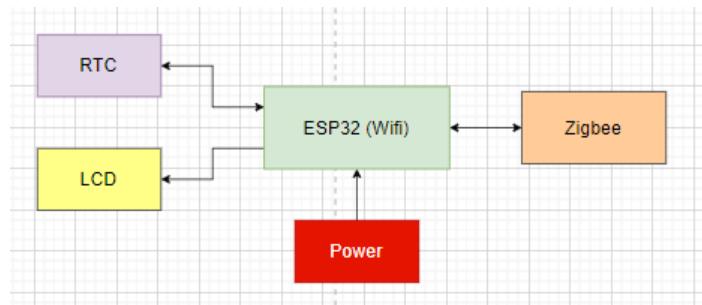
- + Giao tiếp I2C với module DS1307 để cài đặt và lấy thông số về thời gian thực.
- + Giao tiếp UART với Module Zigbee để điều khiển và nhận dữ liệu các node
- + Giao tiếp SPI với màn hình LCD5110 để hiển thị các thông số lên màn hình

#### 3.2.2.2. Thiết kế phần mềm

##### a. Giới thiệu về RTOS



Hình 3.11. ESP32



Hình 3.12. Sơ đồ khối chức năng của G-Hub

RTOS [3] (Real-Time Operating System) là một hệ thống điều khiển thời gian thực, được thiết kế để xử lý các ứng dụng yêu cầu khắt khe đối với độ chính xác và độ tin cậy của thời gian đáp ứng. Những ứng dụng này thường được sử dụng trong các hệ thống nhúng, như điều khiển trình duyệt của máy bay, bộ điều khiển của hệ thống treo tường thông minh, hệ thống giám sát và điều khiển giao thông, các cảm biến và hệ thống điều khiển của robot, và nhiều hệ thống khác.



Hình 3.13. RTOS

Về cơ bản, RTOS giống như một hệ thống điều khiển đa nhiệm trên một máy tính, nhưng nó có những tính năng và đặc tính riêng để đảm bảo việc xử lý được các tác vụ yêu cầu chính xác và đáp ứng thời gian thực. Một điểm đáng chú ý của RTOS là tính đối xứng, nghĩa là các tác vụ sẽ được ưu tiên theo thời gian đáp ứng của chúng, và chúng đều được giải quyết đồng thời. Tuy nhiên, tính đối xứng này có thể được điều chỉnh để đáp ứng các yêu cầu đặc biệt của một ứng dụng cụ thể.

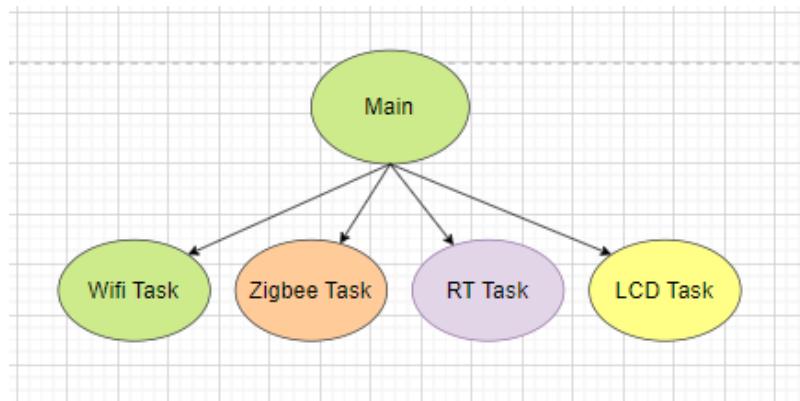
Để giải quyết các tác vụ yêu cầu chính xác, RTOS sử dụng một số cơ chế quản lý tác vụ, bao gồm các bộ lập lịch và thời gian đặt tác vụ. Các tác vụ RTOS được quản lý bởi một nhân viên điều hành, có thể được điều chỉnh để đáp ứng các yêu cầu không đồng thời và theo thời gian thực.

Ứng dụng của RTOS rất đa dạng và phong phú, từ điều khiển các ứng dụng IoT nhỏ gọn cho đến được sử dụng trên các hệ thống phức tạp, phân tán. Một số ví dụ về ứng dụng RTOS bao gồm:

- Điều khiển hệ thống giao thông để đảm bảo an toàn và tiết kiệm thời gian cho các phương tiện giao thông.
- Điều khiển các cảm biến và hệ thống điều khiển trong robot hỗ trợ con người trong các tác vụ khác nhau.
- Điều khiển các thiết bị điện tử trong các hệ thống nhà thông minh, bao gồm cả các hệ thống treo tường thông minh và hệ thống điều khiển năng lượng.
- Điều khiển các thiết bị tại bệnh viện hoặc trong các ứng dụng y tế để đảm bảo tính an toàn và đáp ứng thời gian thực.

## b. Phần mềm

Esp32 [4] được sử dụng kết hợp với RTOS [5] để có thể chia chương trình thành các task khác nhau để hoạt động riêng biệt với nhau. Tạo khả năng hoạt động của chương trình tốt hơn và ổn định hơn.

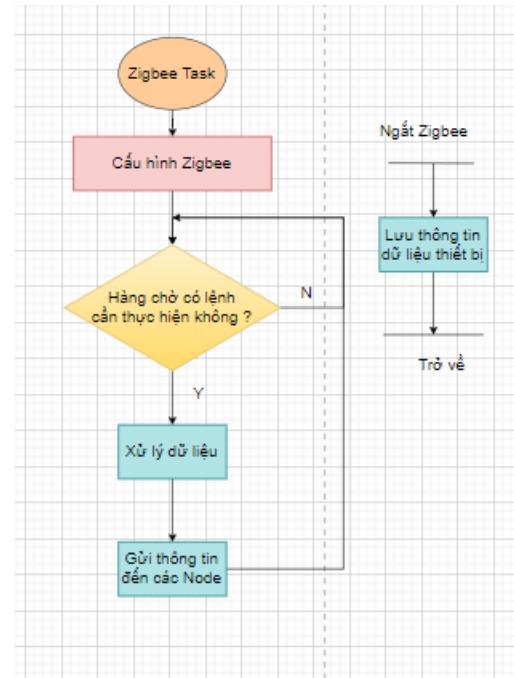


Hình 3.14. Các Task chính của ESP32

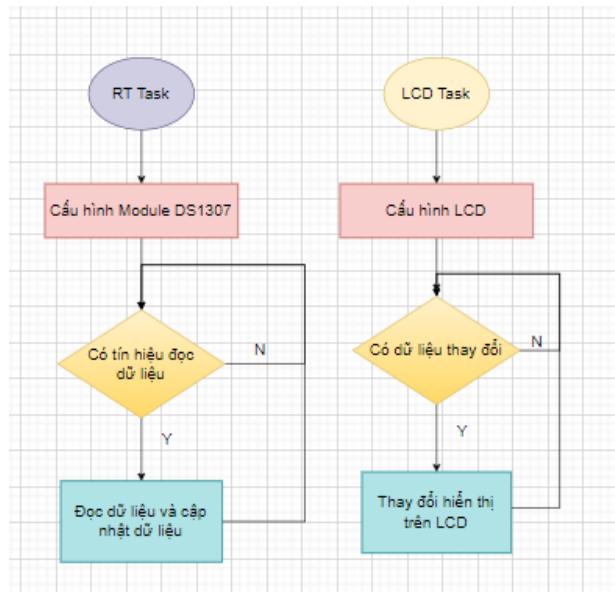
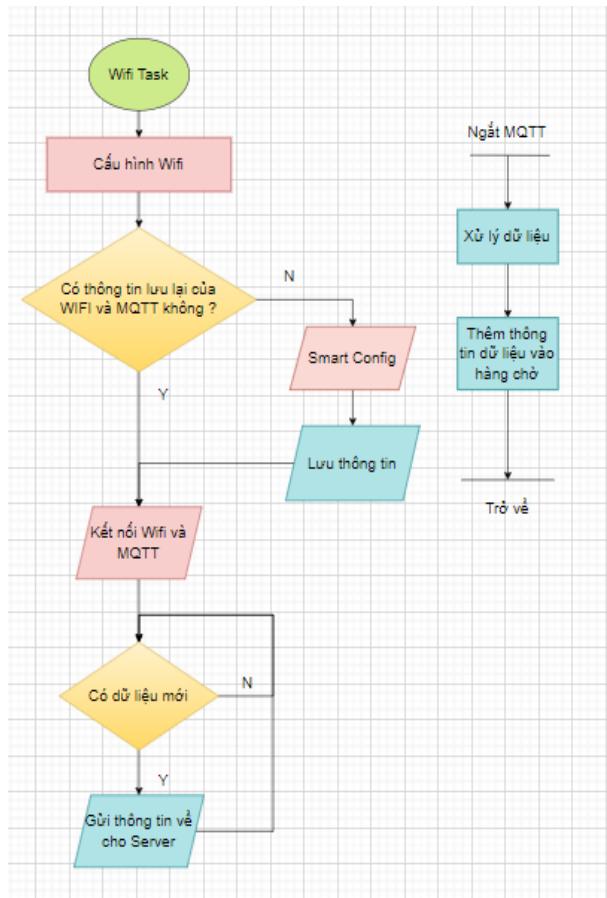
Task Zigbee (Hình 3.15) đảm nhận nhiệm vụ liên quan đến Zigbee, mới đầu sẽ cấu hình Zigbee sau đó chờ nhận lệnh và gửi đi khi có yêu cầu từ các task khác. Nếu có thông tin từ Zigbee thì dữ liệu sẽ được lưu vào và xử lý phụ thuộc vào dữ liệu nhận đó là gì.

Trong hình 3.17 là hiển thị hoạt động của Task Wi-Fi, Task này đảm nhận liên quan đến giao tiếp với bên ngoài thông qua mạng internet. Kết nối wifi sau đó kết nối MQTT để giao tiếp với back-end, trao đổi dữ liệu và nhận lệnh từ phía đó.

Còn trong hình 5.16 là 2 Task, Task RTC và Task LCD. Task TRC giao tiếp với module DS1307



Hình 3.15. Task Zigbee



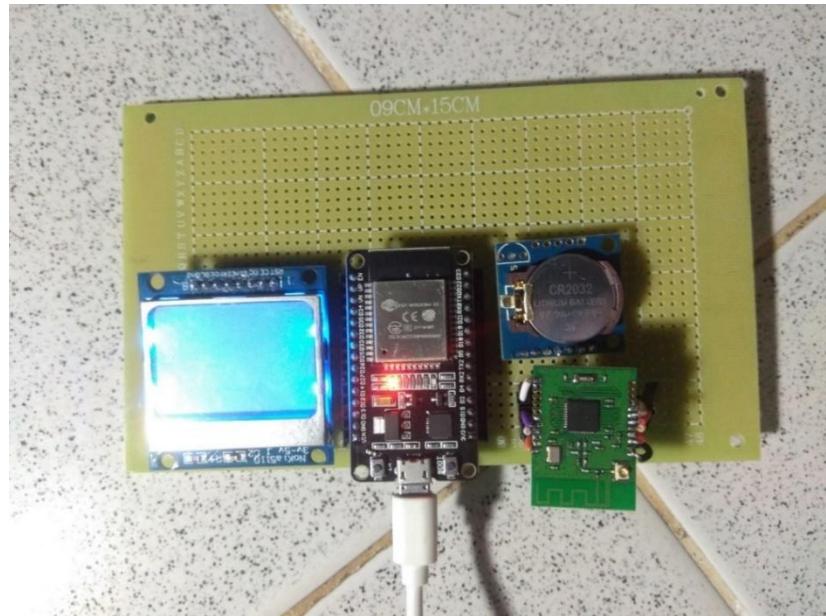
Hình 3.16. Task RTC và LCD

để đảm bảo việc quản lý thời gian và cập nhật thời gian cho G-Hub. Task LCD giao tiếp với LCD để hiển thị thông tin ra LCD tùy thuộc vào trường hợp có thể là hiển thị kết nối Wifi hoặc là hiển thị trạng thái ngày giờ.

Hình 3.17. Task Wi-Fi

### 3.2.2.3. Kết quả

G-Hub được thử nghiệm đưa ra kết quả hoạt động như mong đợi. Kết nối được wifi, giao tiếp mạng Zigbee với các Node khác và còn cả giao tiếp MQTT để giao tiếp với bên ngoài. LCD có thể hiển thị thông tin của G-Hub và một số trạng thái. RTC đưa ra được thời gian đúng.



Hình 3.18. G-Hub Thủ nghiệm

### 3.2.3. Thiết kế Node Sensor

#### 3.2.3.1. Sơ đồ khối chức năng

Node Sensor là thiết bị kết nối Zigbee có chức năng đọc cảm biến về thông số môi trường như nhiệt độ, độ ẩm và chất lượng không khí. Được sử dụng để giám sát điều kiện môi trường cũng như cảnh báo các chất khí dễ cháy như khí gas.

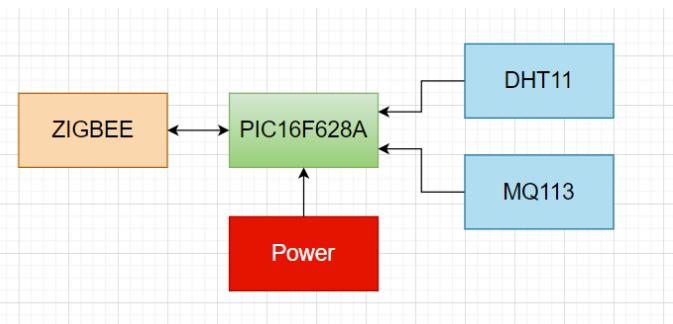
PIC16F628A [6] là một chíp vi điều khiển được sản xuất bởi hãng Microchip thuộc họ Pic, là một bộ vi điều khiển 8-bit dựa trên kiến trúc RISC với các thông số kỹ thuật sau:

Bộ nhớ Flash	2KB
Bộ nhớ EEPROM	128B
Bộ nhớ SRAM	224B
Internal Oscillator	4MHz $\pm 1\%$
IO	16
UART	1
Analog comparator	1
Capture, Compare, PWM (CCP)	1
Timer 8-bits/16-bits	2/1

Bảng 3.1. Thông số của Pic16F628a

Cảm biến chất lượng không khí MQ-135 [7] sử dụng để kiểm tra chất lượng không khí trong môi trường, thường được sử dụng trong các thiết bị kiểm tra chất lượng không khí bên trong cao ốc, văn phòng, thích hợp để phát hiện NH<sub>3</sub>, NOx, Ancol, Benzen, khói, CO<sub>2</sub>, ...

Khoảng phát hiện: 10 ~ 300 ppm NH<sub>3</sub>, 10 ~ 1000 ppm Benzene, 10 ~ 300 Alcol.



Hình 3.20. Sơ đồ khối chức năng của Node Sensor



Hình 3.19. Pic16F628a

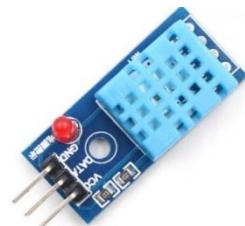


Hình 3.21. MQ-135

Cảm biến nhiệt độ, độ ẩm DHT11 [8] sử dụng để thu thập dữ liệu nhiệt độ và độ ẩm trong môi trường

Thông số kỹ thuật:

- Khoảng đo nhiệt độ:  $0 \sim 50^{\circ}\text{C} \pm 2^{\circ}\text{C}$
- Khoảng đo độ ẩm:  $20\% \sim 80\%\text{RH} \pm 5\%\text{RH}$
- Tần số lấy mẫu tối đa: 1Hz



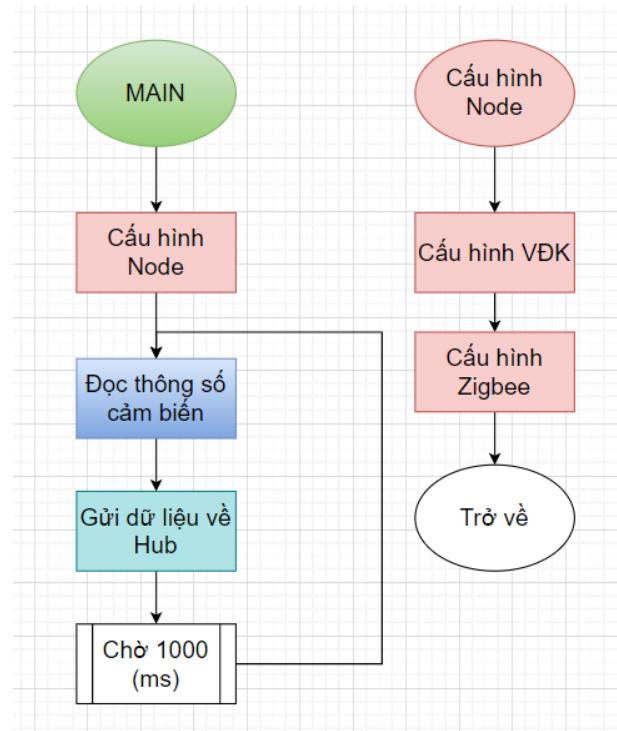
Hình 3.22. DHT11

### 3.2.3.2. Thiết kế phần mềm

Phần mềm được thiết kế để Pic16F628a hoạt động đáp ứng được yêu cầu của Node Sensor. Cho nên Pic16F628a phải thu thập dữ liệu từ 2 cảm biến DHT11 và MQ135 và sau đó gửi dữ liệu thu thập được về cho G-Hub để G-Hub cập nhật dữ liệu và hiển thị cho người dùng. Trước khi làm những điều đó Node Sensor phải cấu hình cho VĐK và cấu hình cho Zigbee để có thể kết nối vào mạng và giao tiếp được với các Node trong mạng hoặc là giao tiếp với G-Hub để gửi dữ liệu lên.

### 3.2.3.3. Kết quả

Sau khi thực hiện được Node Sensor, Node Sensor hoạt động ổn có thể giao tiếp được mạng Zigbee, đọc thông số môi trường được và gửi về cho G-Hub.



Hình 3.23. Flow chart của Node Sensor



Hình 3.24. Node Sensor thử nghiệm

### 3.2.4. Thiết kế Node Button

#### 3.2.3.1. Sơ đồ khối chức năng

Node Button bao gồm các nút nhấn, được sử dụng để điều khiển các relay đã được liên kết thông qua trang web.

Vị điều khiển chính là Pic16f628a.

Nó được dùng để lập trình hoạt động cho Node giống như Node Sensor.

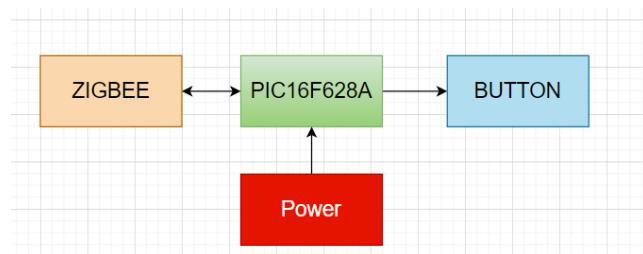
Nút nhấn dùng trong các mạch điện tử thông dụng, có thể dùng để làm nút reset hoặc nút nguồn, ...

Thông số kỹ thuật của nút nhấn sử dụng:

- Kích thước: 6x6x8mm
- Loại chân: cắm 4 chân.

#### 3.2.3.2. Thiết kế phần mềm

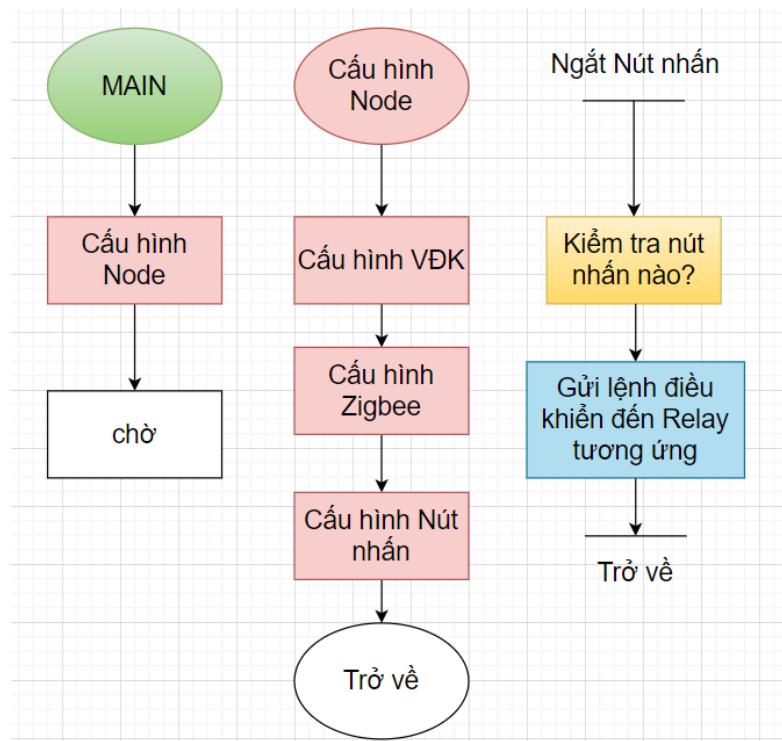
Ở Node Button, Vị điều khiển chính là Pic16F628a nên sẽ phải lập trình cho nó để Node có thể hoạt động tốt được. Pic16F628a được cấu hình lúc khởi động, cấu hình cho Zigbee để tham gia vào mạng, rồi sau đó sẽ cấu hình ngắt cho nút nhấn để khi người dùng nhấn nút để thì sẽ kích hoạt ngắt để VĐK biết người dùng đã nhấn nút và xử lý để gửi dữ liệu đến các Node liên quan khác.



Hình 3.25. Sơ đồ khối chức năng của Node Button



Hình 3.26. Nút nhấn

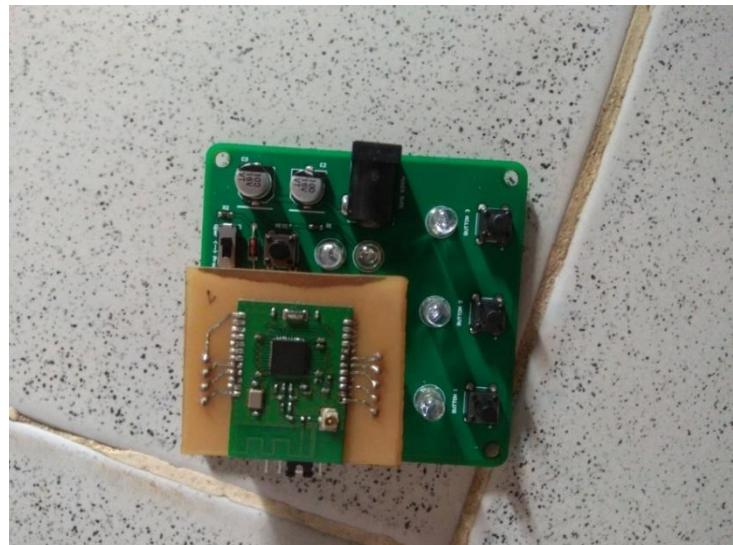


Hình 3.27. Flow chart của Node Button

### 3.2.3.3. Kết quả

Node Button hoạt động tốt, giao tiếp được trong mạng Zigbee, có thể liên kết với Node Relay để điều khiển.

Node Button sẽ được hoạt động theo 2 kiểu. Kiểu thứ nhất là sẽ cố định, được gắn trên tường hoặc ở các vị trí cố định để điều khiển. Kiểu thứ hai là hoạt động giống như remote có thể mang đi mang lại được mà vẫn có thể điều khiển relay bật tắt tốt.



Hình 3.28. Node Button thử nghiệm

### 3.2.5. Thiết kế Node Relay

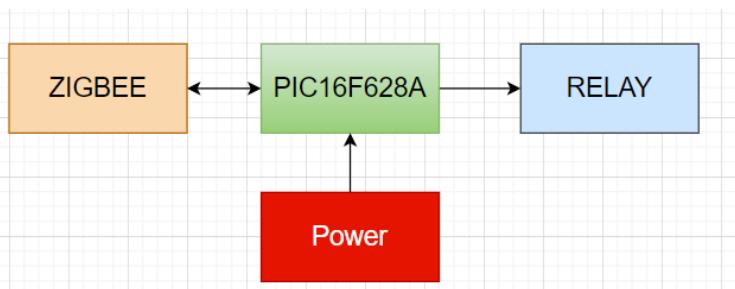
#### 3.2.3.1. Sơ đồ khối chức năng

Node Relay bao gồm các relay, được sử dụng để bật/ tắt thiết bị được điều khiển ở trong nhà. Node relay được điều khiển bằng cách gửi yêu cầu thông qua mạng Zigbee có thể từ G-Hub hoặc từ các Node Button.

Vị điều khiển chính là Pic16F628a tương tự như hai node trên kết hợp với các relay để điều khiển các thiết bị.

Thông số của Relay:

- Điện áp định mức 5VDC.
- Dòng tải 10A.
- cuộn kháng  $400\Omega \pm 10\%$ .
- Dòng điện hoạt động: 40mA.



Hình 3.29. Sơ đồ khối chức năng của Node Relay



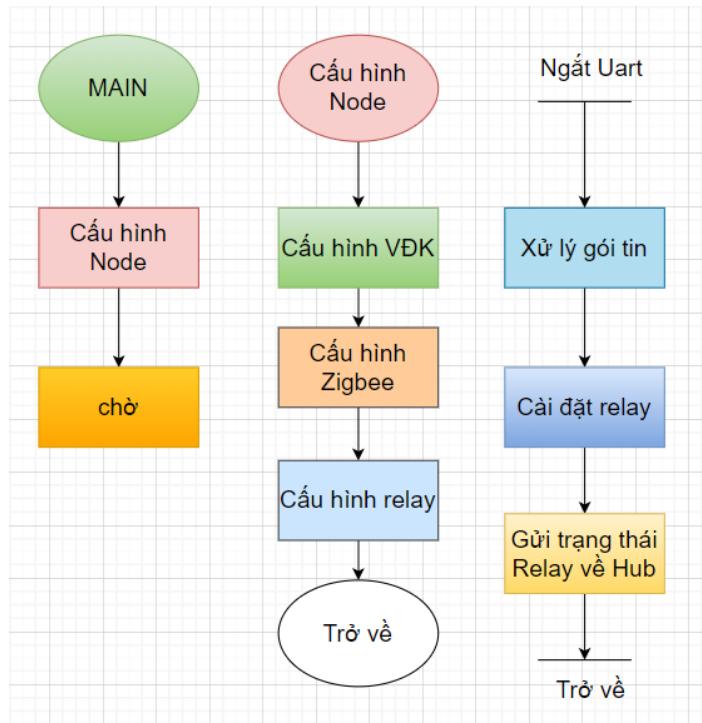
Hình 3.30. Relay

### 3.2.3.2. Thiết kế phần mềm

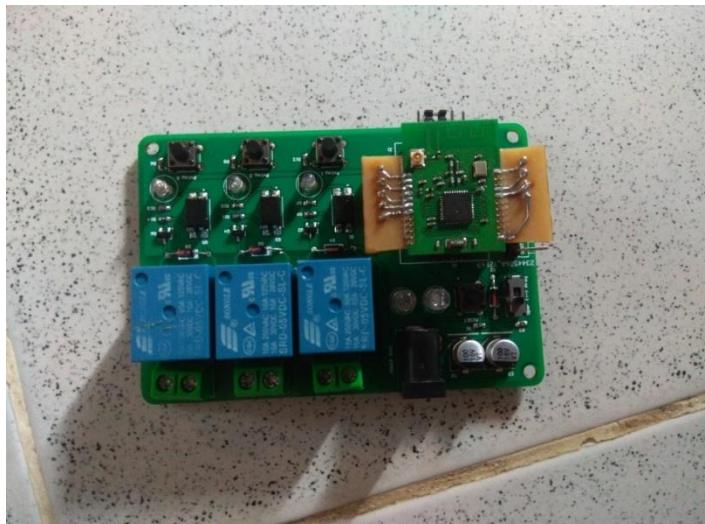
Khởi đầu VĐK được cấu hình để cấu hình các chân ra điều khiển bật/tắt relay từ đó có thể điều khiển được các thiết bị khác. VĐK được cấu hình để hoạt động điều khiển relay qua các cách điều khiển khác nhau và cấu hình Zigbee để Node Relay giao tiếp được với các Node khác. Khi đã cấu hình xong Node relay sẽ chờ lệnh từ các Node khác để có thể điều khiển cài đặt trạng thái cho Relay và gửi trạng thái về cho Hub nếu có sự thay đổi về trạng thái của Relay.

### 3.2.3.3. Kết quả

Hình 3.32 là sản phẩm thử nghiệm hoạt động của Node Relay. Node Relay bật/tắt được và có thể được điều khiển bằng các nút nhấn đã được liên kết và điều khiển thông qua trang Web có thể hoạt động ổn định. Sẽ có nhiều Node Relay tùy theo nhiều loại có thể thuộc vào số Relay hoạt của Node, sẽ có tăng hoặc giảm số lượng Relay để phù hợp với các trường hợp khác nhau.



Hình 3.31. Flow chart của Node Relay



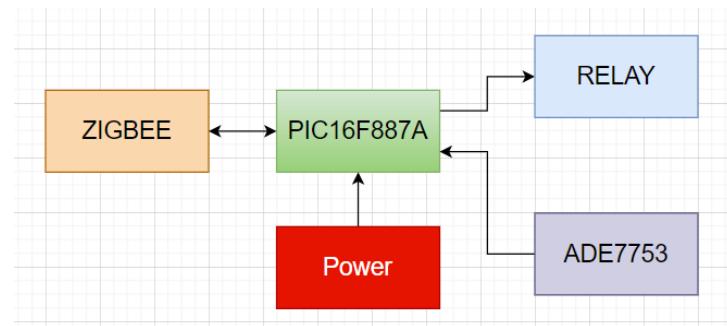
Hình 3.32. Node Relay thử nghiệm

### 3.2.6. Thiết kế Node Relay ADE

#### 3.2.3.1. Sơ đồ khối chức năng

Node Relay ADE có chức năng điều khiển bật/ tắt relay nhưng ngoài ra có thêm tính năng đọc và giám sát và đo lường các thông số điện năng như Vrms, Irms, công suất và năng lượng tiêu thụ của thiết bị được lắp đặt với Node này.

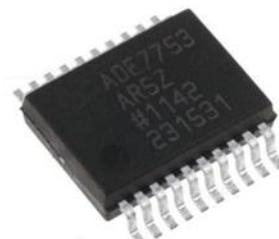
VĐK chính được sử dụng ở đây là Pic16F887 [6], là một chíp vi điều khiển được sản xuất bởi hãng Microchip thuộc họ Pic, là một bộ vi điều khiển 8-bit dựa trên kiến trúc RISC với các thông số kỹ thuật sau:



Bộ nhớ Flash	8KB
Bộ nhớ EEPROM	256B
Bộ nhớ SRAM	368B
Internal Oscillator	31kHz ~ 8MHz ±1%
IO	35
EUSART	1
Analog comparator	2
ECCP/CCP	1/1
Timer 8-bits/16-bits	2/1
MSSP	1
10-bit ADC	14

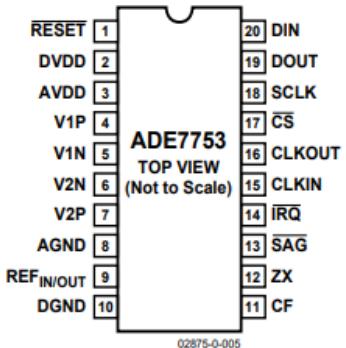
Bảng 3.2. Bảng thông số của Pic16F887

Node Relay ADE ngoài sử dụng relay còn sử dụng thêm ic ADE7753 [9] dùng để tính toán thông số điện. ADE7753 (ADE) là IC (Intergated Circuit) do hãng Analog Device sản xuất. ADE7753 có những tính năng cơ bản sau:



Hình 3.34. ADE7753

- Độ chính xác cao, tuân theo chuẩn ICE 61036/60687/61268, IEC62053-21, IEC 62053-22, và 62053-23.
- Tích hợp bộ tích phân số cho phép kết nối trực tiếp tới cảm biến dòng điện đầu ra tỉ lệ với  $di/dt$ .
- Một bộ PGA trong kênh dòng điện cho phép giao diện trực tiếp tới shunt và bộ biến dòng điện.
- Tính năng lượng hoạt động và năng lượng biểu kiến, dạng sóng và giá trị hiệu dụng của dòng điện và điện áp với sai số nhỏ hơn 0.1%.
- Chế độ tích lũy năng lượng dương.
- Cho phép người dùng đặt chương trình ngưỡng cho sự sụt áp, quá điện áp.
- Hiệu chuẩn số cho nguồn, pha và bù đầu vào.
- Cảm biến đo nhiệt độ trên chip ( $\pm 3^\circ\text{C}$ ).
- Truyền thông nối tiếp SPI.
- Lập trình tần số xung ngõ ra.
- Yêu cầu ngắn ở chân IRQ và thanh ghi trạng thái.
- Điện áp chuẩn 2.4V, cho đưa từ ngoài.
- Nguồn nuôi 5V, công suất thấp (25 mW).



Hình 3.35. ade7753 pinout

IC ADE7753 được sử dụng để đo các thông số như sau:

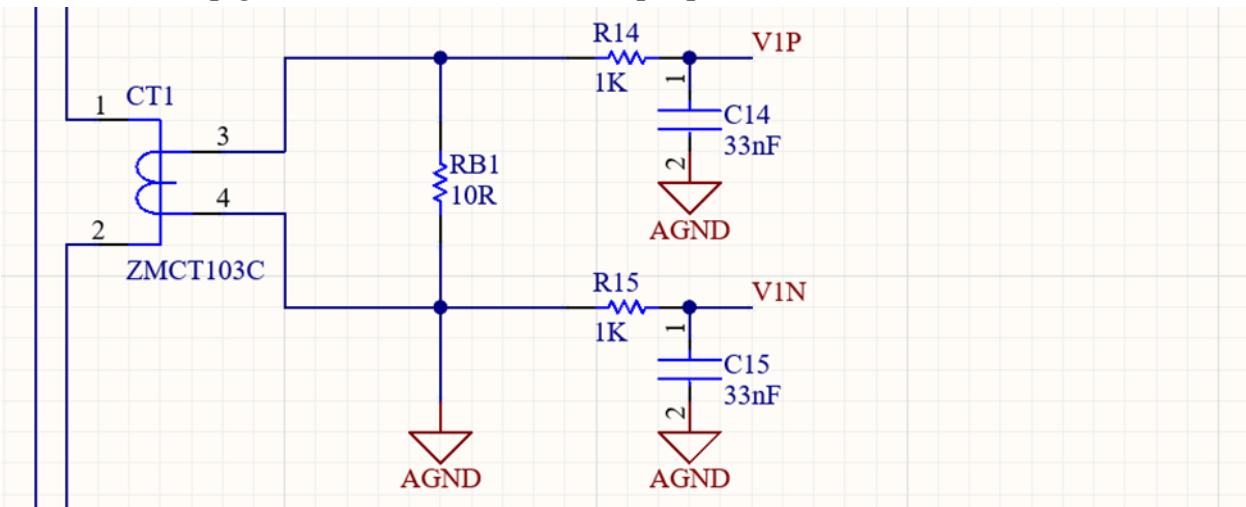
- Tín hiệu điện: cường độ dòng điện xoay chiều, hiệu điện thế xoay chiều.
- Các loại công suất như công suất có ích.

Để ic ADE7753 có thể đo được thì ta cần phải có khói chuyển dòng điện, chuyển điện áp sao cho phù hợp với yêu cầu của Ic.

- Khối chuyển dòng điện

Vì dòng tải yêu cầu đo có thể rất lớn (lên đến vài Ampere) nên ta thiết sử dụng cuộn biến dòng và điện trở burden để đo dòng đi vào IC ADE7753.

Theo tài liệu kỹ thuật của IC ADE7753, kênh V1P/V1N sẽ dùng để đo tần hiệu dòng điện. Độ lệch áp giữa V1P và V1N tối đa cho phép là  $\pm 0.5$  V.



Hình 3.36. Khối chuyển dòng điện

Ta chọn hệ số an toàn là 2, chọn biến dòng 1000:1, dòng hiệu dụng tối đa là 10A. ta có công thức hiệu điện thế U1 đi vào hai chân V1P và V1N:

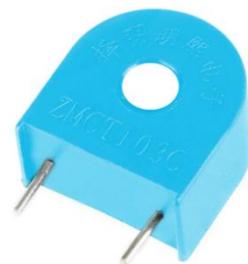
$$U1 = \frac{10\sqrt{2}}{1000} * RB1 < \frac{0.5}{2}$$

$$\Rightarrow RB1 < 17.68 \Omega$$

$$\Rightarrow \text{Chọn } RB1 = 10 \Omega$$

Thông số kỹ thuật của biến dòng zmct103c:

- Tỷ lệ: 1000:1
- Phạm vi tuyến tính: 0~15A
- Sai số: 0.2%
- Dải nhiệt độ: -40 ~ +70 °C
- Khối chuyển điện áp



Hình 3.37. Biến dòng  
ZMCT103C

Theo tài liệu kỹ thuật của IC ADE7753, kênh V2P/V2N sẽ dùng để đo tần hiệu điện áp. Độ lệch áp giữa V2P và V2N tối đa cho phép là  $\pm 0.5$  V.

Ta chọn biến áp 220/12, chọn điện áp đỉnh  $V_{max} = 250\sqrt{2}$ , hệ số an toàn là 2 ta có công thức hiệu điện thế  $U_2$  trên  $R_{19}$  ( $1k\Omega$ ), đồng thời cũng là hiệu điện thế đi vào hai chân  $V_{2P}$  và  $V_{2N}$ :

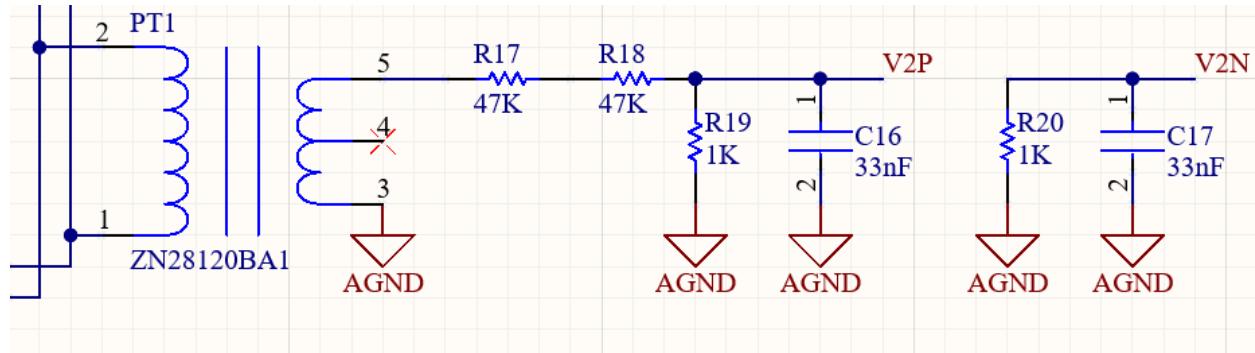
$$U_2 = \frac{250\sqrt{2} * \frac{12}{220}}{R_{17} + R_{18} + R_{19}} * R_{19} < \frac{0.5}{2}$$

$$\Rightarrow R_{17} + R_{18} > 77,14 \text{ k}\Omega$$



Hình 3.38. Biến áp ZN28120BA1

=> Chọn  $R_{17} = R_{18} = 47 \text{ k}\Omega$



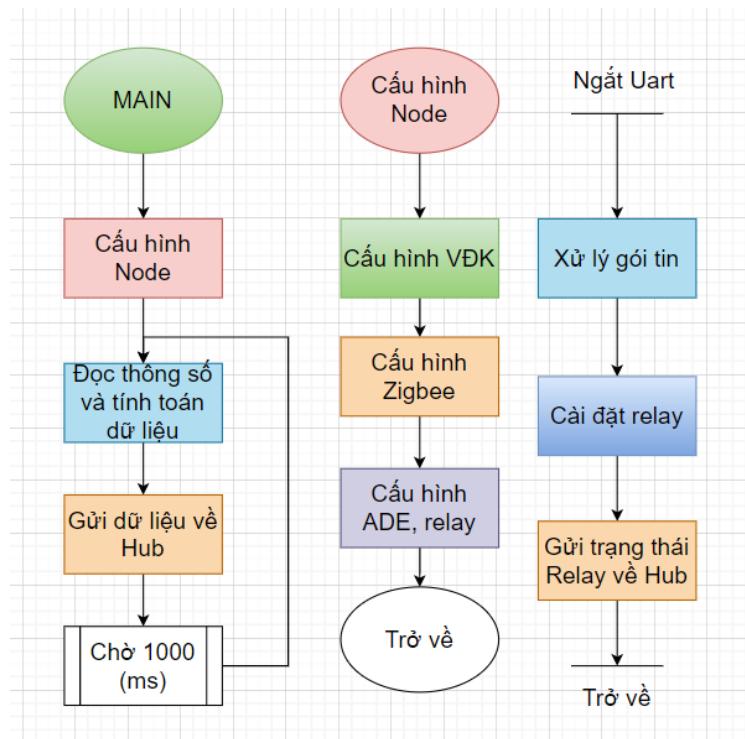
Hình 3.39. Khối chuyển điện áp

Thông số kỹ thuật của biến áp:

- Điện áp vào: 220VAC
- Điện áp ra: 12VAC
- Công suất 1VA

### 3.2.3.2. Thiết kế phần mềm

VĐK được cấu hình đầu tiên để giao tiếp với các Module khác như Zigbee hay ADE7753. Sau đó sẽ là cấu hình cho Module Zigbee và Ic Ade7753 để có thể đo được giá trị chính xác nhất, và còn cả cấu hình relay để bật tắt trạng thái. Khi đã cấu hình xong, VĐK sẽ đọc giá trị thông số điện và gửi về cho G-Hub



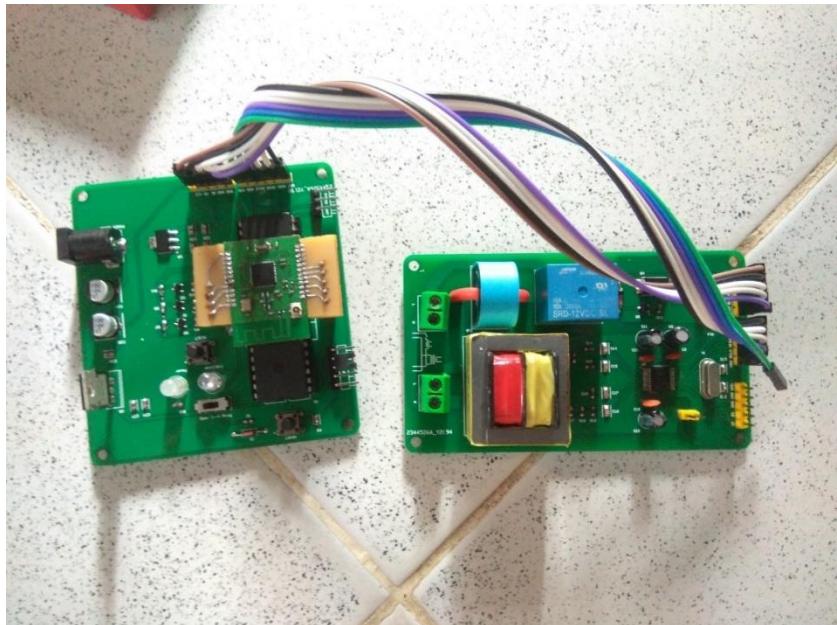
Hình 3.40. Flow chart của Node Relay ADE

để hiển thị cho người dùng sau mỗi giây. Và cũng đợi ngắt Uart để điều khiển trạng thái relay thông qua Module Zigbee.

### 3.2.3.3. Kết quả

Sau khi thử nghiệm Node Relay Ade [2], ta thấy Node hoạt động tốt, có thể giao tiếp Zigbee với các Node khác, đo được thông số điện của thiết bị.

Còn về độ chính xác của thiết bị ta sẽ so sánh với một đồng hồ đo khác. Kết quả đo được thực hiện tính toán và so sánh với Đồng hồ đo đa năng Selec MFM383A.



Hình 3.41. Node Relay ADE thử nghiệm

Node Relay Ade	MFM383A	Sai số (%)
99.67462	99.9	0.225604
120.0036	119.9	0.08644
140.5072	140.3	0.147655
160.087	159.9	0.116952
179.9276	180.2	0.151182
200.1653	200.4	0.117137
211.1576	211.1	0.027283
220.1046	219.8	0.138571
230.6364	230.5	0.059188
241.3361	241.6	0.109214

Bảng 3.3. Kết quả đo điện áp so sánh với đồng hồ đo đa năng Selec MFM383A [1]

Nhận xét: Với kết quả đo điện áp như trên, điện áp giữa hai thiết bị sai số với nhau < 0,5%.

Node Relay Ade	MFM383A	Sai số (%)
0.388162967	0.39	0.471034
0.764894899	0.764	0.117133
1.134394829	1.136	0.1413
1.503665778	1.503	0.044297
2.059652117	2.06	0.016888
2.400033224	2.395	0.210156
2.786993	2.786	0.035642
3.167884763	3.171	0.098241
3.553966776	3.552	0.055371
3.928351647	3.931	0.067371

Bảng 3.4. Kết quả đo dòng điện so sánh với đồng hồ đo đa năng Selec MFM383A [1]

Nhận xét: Sai số đo dòng điện so sánh với đồng hồ đo đa năng Selec MFM383A < 0,5%.

Node Power Meter	MFM383A	Sai số (%)
34.10063	34	0.295978
65.76475	66	0.356443
96.78227	97	0.224464
127.3578	127	0.281763
190.6912	191	0.161661
220.3141	220	0.142775
255.3617	255	0.141825
288.7188	289	0.09731
322.0475	322	0.014758
354.8612	355	0.03909

Bảng 3.5. Kết quả đo công suất so sánh với đồng hồ đo đa năng Selec MFM383A [1]

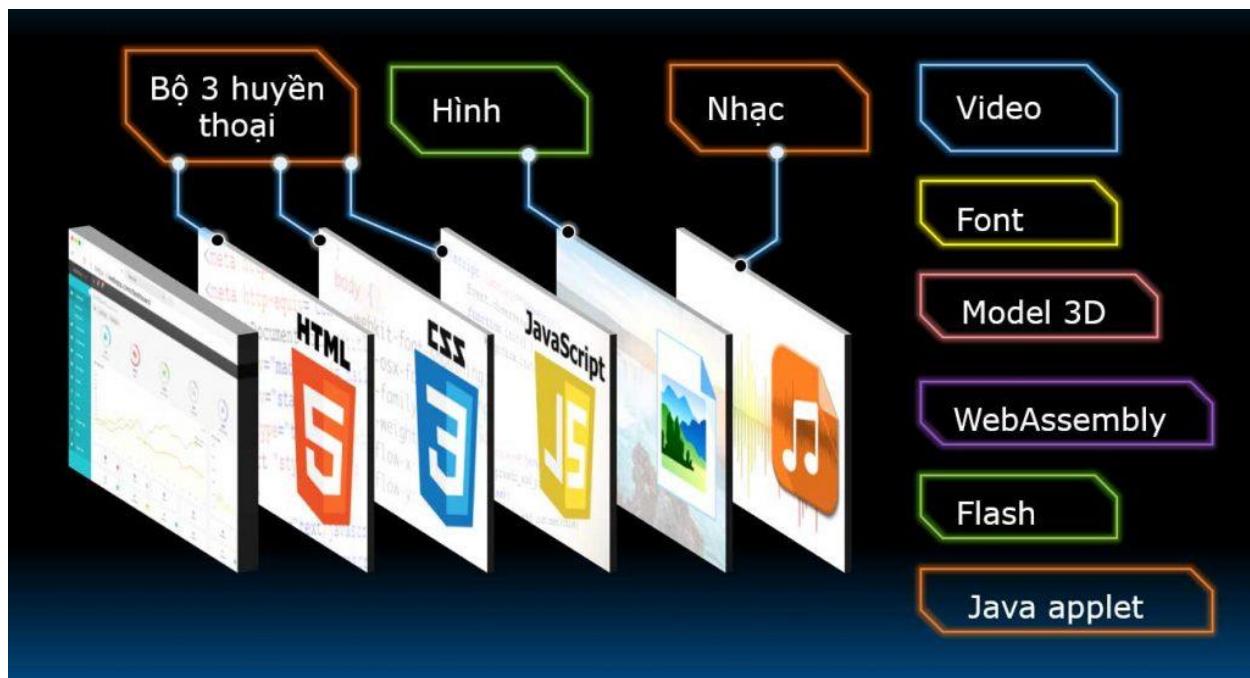
Nhận xét: Sai số khi so sánh với đồng hồ đo đa năng Selec MFM383A < 0,5%.

## Chương 4.Thiết kế Website

### 4.1. Front-end

#### 4.1.1. Giới thiệu

Front-end [10] là những gì mà người dùng nhìn thấy khi truy cập vào các trang web hay ứng dụng web. Hiểu một cách đơn giản, Front-end chính là giao diện của một website. Nó mang lại cho người sử dụng các trải nghiệm tương tác (nghe, nhìn) trên trang web đó. Vì thế, Front-end còn được gọi là “Client-side”.



Hình 4.1. Front-end gồm các yếu tố tạo nên giao diện của các trang web (Nguồn: CodeSchool)

Trong thiết kế web, Front-end được tạo ra bởi 3 loại ngôn ngữ cơ bản là HTML, CSS và JavaScript. Các lập trình viên Front-end sẽ cần đảm bảo nội dung hiển thị tốt trên mọi nền tảng khác nhau.

Thông thường, Front-end của một trang web sẽ được người dùng tương tác trực tiếp theo các khía cạnh như:

- Nhận biết màu sắc chủ đạo, logo
- Khai thác thông tin trên web (văn bản, hình ảnh, âm thanh...)
- Sử dụng các button, toolbars...

#### **4.1.2. Các ngôn ngữ cơ bản**

##### **a. HTML**

HyperText Markup Language (HTML) là một trong 3 ngôn ngữ chính giúp phát triển Front-end cho trang web. Nó được dùng để mô tả cấu trúc của một trang web trên trình duyệt.

Ngôn ngữ HTML có một số đặc điểm cơ bản sau:

- Là ngôn ngữ đơn giản, dễ sử dụng với nhiều thẻ định dạng
- Sử dụng linh hoạt để thiết kế web với văn bản
- Dễ dàng thêm hình ảnh, âm thanh vào các trang web
- Có thể liên kết đến nhiều trang web khác

##### **b. CSS**

Cascading Style Sheets (CSS) cũng là một yếu tố “xương sống” trong xây dựng Front-end. Ngôn ngữ này được dùng để mô tả giao diện và định dạng hiển thị của một trang web. Nhờ có CSS, trang web sẽ trở nên trực quan và hấp dẫn hơn.

Trong lập trình web, CSS sẽ mô tả định dạng theo tài liệu được viết bằng ngôn ngữ HTML. Nó cung cấp tính năng bổ sung cho HTML. Ngoài HTML, CSS còn có thể áp dụng cho nhiều loại tài liệu khác như XML, XUL và SVG.

Về cơ bản, ngôn ngữ CSS có các đặc điểm chính như:

- CSS bổ sung các thuộc tính chi tiết hơn HTML
- Là ngôn ngữ “làm đẹp” cho trang web
- Có thể áp dụng cho nhiều dạng tài liệu khác nhau, bao gồm cả HTML, XUL, SVG...

##### **c. Javascript**

Đây là một ngôn ngữ lập trình cao cấp mà các lập trình viên Front-end luôn phải nắm rõ. Cùng với HTML và CSS, JavaScript là một trong 3 ngôn ngữ thiết yếu trong xây dựng giao diện trang web.

Đối với lập trình viên Front-end, JavaScript có một số ưu điểm như:

- JavaScript hỗ trợ tất cả các trình duyệt mà không cần plug-in

- Hỗ trợ thiết kế trang web động và các hiệu ứng hình ảnh qua DOM
- Có thể thực hiện được một số tác vụ như tự động thay đổi hình ảnh, kiểm tra thông tin nhập vào...

#### 4.1.3. Các thư viện sử dụng

Hiện nay, nhờ sự hỗ trợ của các framework, lập trình viên Front-end có thể đảm bảo nội dung hiển thị tốt trên các nền tảng khác nhau.

Các công ty công nghệ đã cho ra đời rất nhiều framework hỗ trợ phát triển Front-end như:

- Google: Material, AngularJS...
- Facebook: React native, React, ...
- Twitter: Bootstrap, hogan.js...

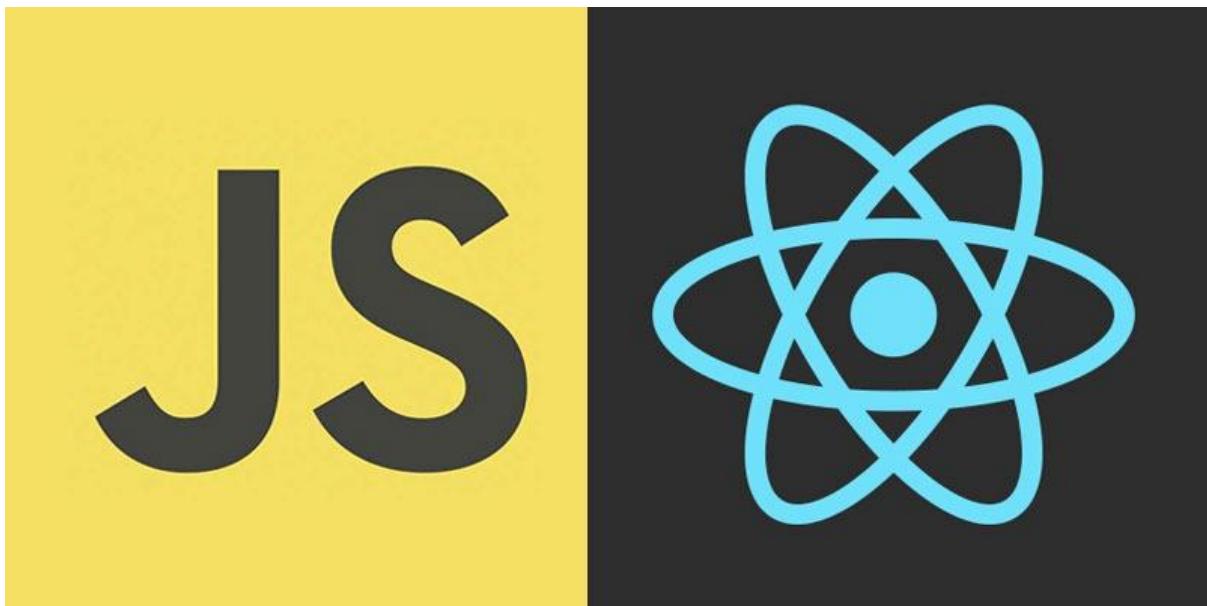
Ở đây chúng ta sẽ sử dụng ReactJs và MUI để thiết kế trang web tương tác với người dùng.

##### 4.1.3.1. ReactJs

ReactJS [11] là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,... Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript” (JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

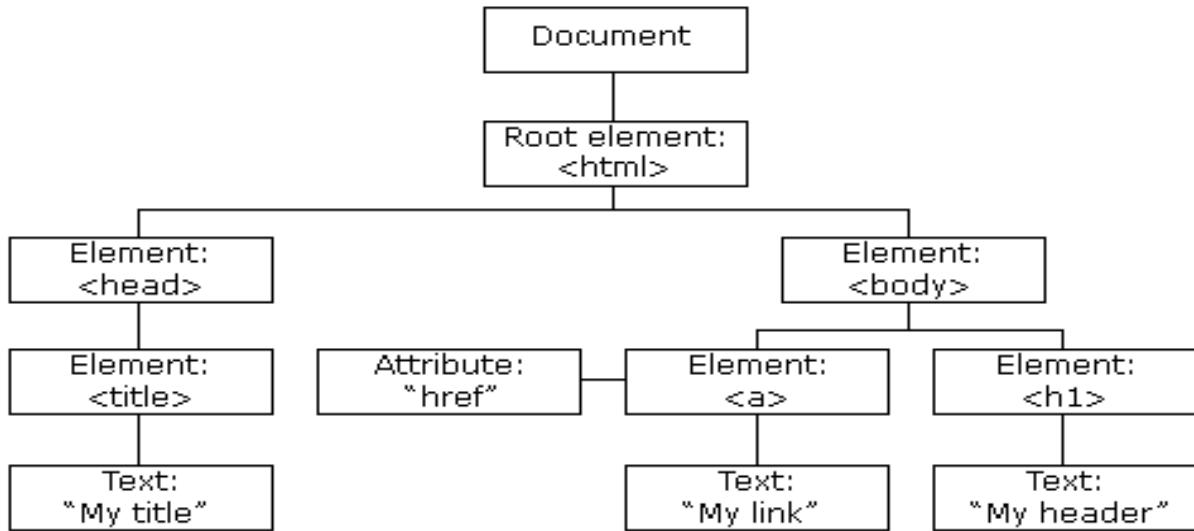


Hình 4.2. Javascript and ReactJs

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:

- JSX.
  - Virtual DOM.
- a. JSX

Trọng tâm chính của bất kỳ website cơ bản nào đó là những HTML documents. Trình duyệt Web đọc những document này để hiển thị nội dung của website trên máy tính, tablet, điện thoại của bạn. Trong suốt quá trình đó, trình duyệt sẽ tạo ra một thứ gọi là Document Object Model (DOM) – một tree đại diện cho cấu trúc website được hiển thị như thế nào. Lập trình viên có thể thêm bất kỳ dynamic content nào vào những dự án của họ bằng cách sử dụng ngôn ngữ JavaScript để thay đổi cây DOM.



Hình 4.3. DOM (Nguồn: [https://www.w3schools.com/js/js\\_htmldom.asp](https://www.w3schools.com/js/js_htmldom.asp))

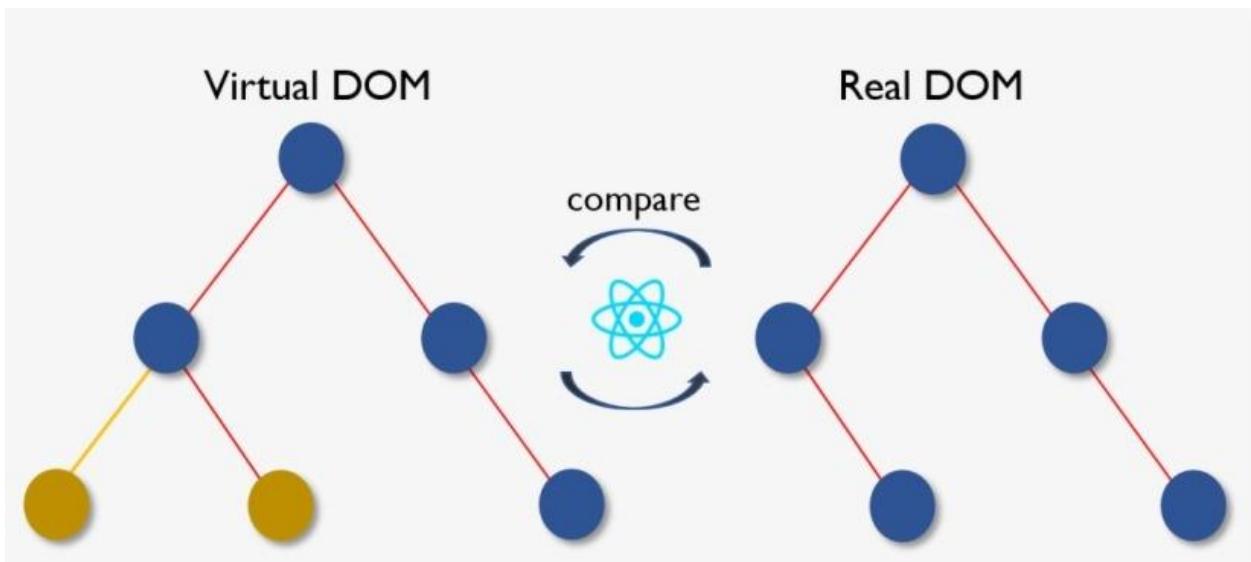
JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào mà bạn đang làm việc.

### b. Virtual DOM

Nếu bạn không sử dụng ReactJS (và JSX), website của bạn sẽ sử dụng HTML để cập nhật lại cây DOM cho chính bản nó (quá trình thay đổi diễn ra tự nhiên trên trang mà người dùng không cần phải tải lại trang), cách làm này sẽ ổn cho các website nhỏ, đơn giản, static website. Nhưng đối với các website lớn, đặc biệt là những website thiên về xử lý các tương tác của người dùng nhiều, điều này sẽ làm ảnh hưởng performance website cực kỳ nghiêm trọng bởi vì toàn bộ cây DOM phải reload lại mỗi lần người dùng nhấp vào tính năng yêu cầu phải tải lại trang).

Tuy nhiên, nếu bạn sử dụng JSX thì bạn sẽ giúp cây DOM cập nhật cho chính DOM đó, ReactJS đã khởi tạo một thứ gọi là Virtual DOM (DOM ảo). Virtual DOM (bản chất của nó theo đúng tên gọi) là bản copy của DOM thật trên trang đó, và ReactJS sử dụng bản copy đó để tìm kiếm đúng phần mà DOM thật cần cập nhật khi bất kỳ một sự kiện nào đó khiến thành phần trong nó thay đổi (chẳng hạn như user nhấp vào một nút bất kỳ).

Ví dụ, khi người dùng bình luận vào khung comment vào bất kỳ bài Blog nào trên website của bạn và nhấn “Enter”. Dĩ nhiên, người dùng của bạn sẽ cần phải thấy được bình luận của mình đã được thêm vào danh sách bình luận. Giả sử trong trường hợp không sử dụng ReactJS, toàn bộ cây DOM sẽ phải cập nhật để báo hiệu sự thay đổi mới này. Còn khi bạn sử dụng React, nó sẽ giúp bạn scan qua Virtual DOM để xem những gì đã thay đổi sau khi người dùng thực hiện hành động trên (trong trường hợp này, thêm mới bình luận) và lựa chọn đúng nơi đúng chỗ cần cập nhật sự thay đổi mà thôi.



Hình 4.4. Real DOM and Virtual DOM

Với việc cập nhật đúng chỗ như vậy, khỏi phải nói nó tiết kiệm cho chúng ta rất nhiều tài nguyên cũng như thời gian xử lý. Ở các website lớn và phức tạp như thương mại điện tử, đặt món ăn, v.v bạn sẽ thấy việc này là vô cùng cần thiết và quan trọng để làm tăng trải nghiệm của khách hàng và performance được cải thiện đáng kể.

#### 4.1.3.2. MUI

Material UI [12] là một thư viện thành phần React mã nguồn mở triển khai bởi Google's Material Design. Nó bao gồm một bộ sưu tập toàn diện các thành phần dựng sẵn sẵn sàng để sử dụng trong sản xuất ngay khi xuất xưởng.



Hình 4.5. MUI

Material UI có thiết kế đẹp mắt và có một bộ tùy chọn tùy chỉnh giúp dễ dàng triển khai hệ thống thiết kế tùy chỉnh của riêng bạn trên các thành phần.

## 4.2. Back-end

### 4.2.1. Giới thiệu

Back-End [10] là thuật ngữ được dùng để chỉ những phần không thể nhìn thấy và không thể tương tác được của một trang web. Những thành phần này sẽ chạy trong nền để cung cấp các chức năng cần thiết cho người dùng. Có thể hiểu rằng Back-End chính là máy chủ của một trang web, có nhiệm vụ lưu trữ và sắp xếp dữ liệu sao cho website hoạt động hiệu quả nhất.

Theo đó, lập trình viên Back-End là người có nhiệm vụ xử lý những vấn đề liên quan đến dữ liệu và những nghiệp vụ chức năng để hệ thống hoạt động trơn tru. Cụ thể, Back-End sẽ thực hiện việc xây dựng mã và ngôn ngữ cho chạy trên máy chủ của website. Công việc của một lập trình viên Back-End thường không được thể hiện ra bên ngoài bởi vì họ thường thao tác với server và dữ liệu, mà những thành phần này có yêu cầu rất cao về bảo mật nên thường được mã hóa.

### 4.2.2. Vai trò

Trong lĩnh vực lập trình web, các lập trình viên Back-End có vai trò tạo dựng môi trường với những thuật toán logic phù hợp để ứng dụng hay website hoạt động chính xác. Việc này được thực hiện bằng cách sử dụng các ngôn ngữ lập trình như Java, .NET, PHP hay Ruby.

Back-End cũng đồng thời tạo ra một giải pháp lưu trữ dữ liệu tối ưu với database và đảm bảo tối ưu hóa tốc độ cùng hiệu quả của ứng dụng, website. Đối với các ứng dụng, website thì database là thành phần rất quan trọng, bởi vì đó là nơi lưu trữ toàn bộ thông tin về người dùng, bài đăng, bình luận và những tài nguyên dữ liệu khác.

Bên cạnh đó, Back-End cũng giữ vai trò phát triển hệ thống xử lý dữ liệu và đảm bảo lưu trữ dữ liệu an toàn. Họ cũng chịu trách nhiệm quản lý tài nguyên API trên các thiết bị, tham gia vào việc xây dựng khung hoặc kết cấu để việc lập trình được thuận lợi hơn.

Thông thường một Back-End sẽ đảm nhận những nhiệm vụ chính sau:

- Xác thực người dùng: đảm bảo tính chính xác các chi tiết tài khoản của người dùng và thiết lập quyền của người dùng.
- Kiểm soát trình tự hoạt động: đảm bảo tính logic các hoạt động được thực hiện trên trang web, không để xảy ra sai sót.
- Tối ưu hóa hiệu quả hoạt động: đảm bảo các chức năng của trang web hoạt động nhanh nhất có thể.

#### 4.2.3. NodeJS

NodeJS là một trình thông dịch JavaScript, được xây dựng trên JavaScript V8 Engine của Google Chrome, và được nhiều công ty nổi tiếng trên thế giới sử dụng nó. Thông thường, NodeJS được sử dụng để chạy JavaScript trên server, và bạn có thể sử dụng nó là để phát triển các trang phát video trực tuyến, hay là các ứng dụng chat online. Bạn hoàn toàn có thể sử dụng nó miễn phí và chỉnh sửa nó thoải mái bởi vì đây là một dự án mã nguồn mở.



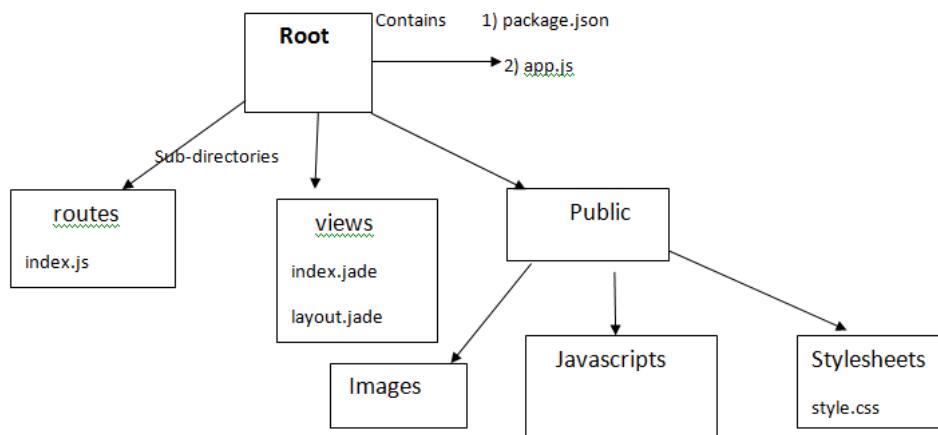
Hình 4.6. NodeJs

Bạn có thể sử dụng NodeJS trên đa số các nền tảng hệ điều hành máy tính hiện nay, chẳng hạn như Windows, Linux hay là macOS. Ngoài ra, nhờ nguồn JavaScript Module phong phú, vậy nên các lập trình viên có thể tiết kiệm khá thời gian, đồng thời đơn giản hóa quy trình lập trình của mình. Một số ứng dụng thực tế của NodeJS mà bạn nên biết đó là Ad Server, Real-time Data Application, Cloud Services, WebSocket server, RESTful API hay là Fast File Upload Client.

Những đặc điểm của NodeJS:

- Các APIs nằm bên trong thư viện NodeJS sẽ không được đồng bộ bởi vì NodeJS không cần đợi API trả dữ liệu về.
- NodeJS hỗ trợ lập trình viên phát triển web một cách độc lập, nhưng cũng rất nhanh chóng nhờ vào việc đây là một platform chứ không phải là framework.
- Một số nền tảng hệ điều hành máy tính hỗ trợ NodeJS đó là Windows, Linux hay là macOS.
- NodeJS không thể hỗ trợ đa luồng bởi vì NodeJS được xem như là một máy chủ đơn luồng.
- Lập trình viên cần nắm vững các kiến thức lập trình cơ bản như là các giao thức hay là JavaScript, bởi vì NodeJS không được xem như là một ngôn ngữ lập trình.
- Người mới bắt đầu sử dụng NodeJS không cần quá lo lắng bởi vì cộng đồng hỗ trợ cho NodeJS thường rất lớn, nên bạn hoàn toàn có thể tìm thấy sự giúp đỡ của họ khi gặp khó khăn.
- Tốc độ xử lý tương đối cao nhờ vào việc được viết bằng ngôn ngữ C++, do đó nó đáp ứng được thời gian chạy trên đa thiết bị, đa nền tảng.

#### 4.2.4. ExpressJs



Hình 4.7. Cấu trúc file

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Tổng hợp một số chức năng chính của Expressjs như sau:

- Thiết lập các lớp trung gian để trả về các HTTP request.
- Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

## 4.3. Database

### 4.3.1. Giới thiệu

Database là một tập hợp dữ liệu có hệ thống hay còn được gọi là cơ sở dữ liệu. Database được hình dung như một kho sách ngăn nắp có chia để mục và phân loại rõ ràng. Chỉ khác là Database hoạt động trên nền tảng điện tử có phần trùu tượng hơn. Tuy nhiên nó cũng sở hữu đủ các tính năng như quản lý, điều phối, trích xuất, tìm kiếm hay chỉnh sửa... dữ liệu.

Database có thể được lưu trữ trên các phần mềm online, offline dưới dạng một tệp tin hoặc một hệ thống dữ liệu. Database cũng thường được lưu trên các thiết bị có chức năng ghi nhớ như ổ cứng, thẻ nhớ, USB...

Thành phần chính của database:

- **Phần cứng:** Phần cứng được hiểu như các thiết bị vật lý giúp kết nối người dùng và hệ thống dữ liệu. Phần cứng thường là máy tính, điện thoại di động, các thiết bị I/O hay USB, thẻ nhớ... Ví dụ, nếu cơ sở dữ liệu của bạn nằm trên máy tính cá nhân thì máy tính này sẽ là phần cứng. Vẫn là cơ sở dữ liệu đó nhưng bạn lại tải về và bật nó trên điện thoại thì phần cứng sẽ là điện thoại.
- **Phần mềm:** Ngược lại với phần cứng, phần mềm là các chương trình trùu tượng không thể có tiếp xúc vật lý với người dùng. Phần mềm được dùng để quản lý và điều khiển các Database. Có rất nhiều loại phần mềm khác nhau, có thể là phần mềm điều hành, phần mềm chia sẻ dữ liệu giữa nhiều người dùng khác nhau hoặc bản thân Database cũng được hiểu như một dạng phần mềm.

- **Dữ liệu:** Dữ liệu còn được gọi là Data theo ngôn ngữ công nghệ. Các dữ liệu này đến từ nhiều nguồn khác nhau, được lưu trữ bằng nhiều cách khác nhau. Dữ liệu cũng có thể tồn tại dưới rất nhiều hình thức, ví dụ như hình ảnh, âm thanh, văn bản... và thậm chí là hệ ngôn ngữ nhị phân. Tuy nhiên Data vẫn chưa thể được sử dụng ngay lập tức, nó vẫn tồn tại dưới dạng “nguyên liệu thô” và đang chờ để được Database xử lý.
- **Quy trình:** Một Database có thể được sử dụng bởi nhiều người dùng khác nhau, nhất là trong một team phát triển sản phẩm. Vì thế thật bất tiện nếu những thành viên trong nhóm cứ phải liên tục hỏi nhau về cách thao tác trong Database phải không nào? Để giải quyết vấn đề này, người tạo Database sẽ soạn một hướng dẫn chi tiết về cách mà Database vận hành cũng như cách sử dụng nó. Thường thì quy trình này sẽ được tài liệu hóa để người đọc có thể dễ hiểu hơn.
- **Ngôn ngữ truy cập:** Ngôn ngữ truy cập được dùng để tham gia vào một Database. Nó cho phép người dùng truy cập vào dữ liệu cũ, cập nhật dữ liệu mới hoặc trích xuất và thực hiện nhiều thao tác khác trong Database. Ngôn ngữ truy cập thường sẽ được quy định bởi người khởi tạo ra Database. Nếu người dùng không biết cách sử dụng ngôn ngữ có sẵn của Database thì có thể dùng các trình biên dịch.

#### 4.3.2. MongoDB

MongoDB [13] là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có thể



Hình 4.8. MongoDB

lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

NoSQL còn được xem là mảnh vá cho những hạn chế, khiếm khuyết của mô hình Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS). Đó là cải tiến về tốc độ, tính năng cũng như khả năng mở rộng.

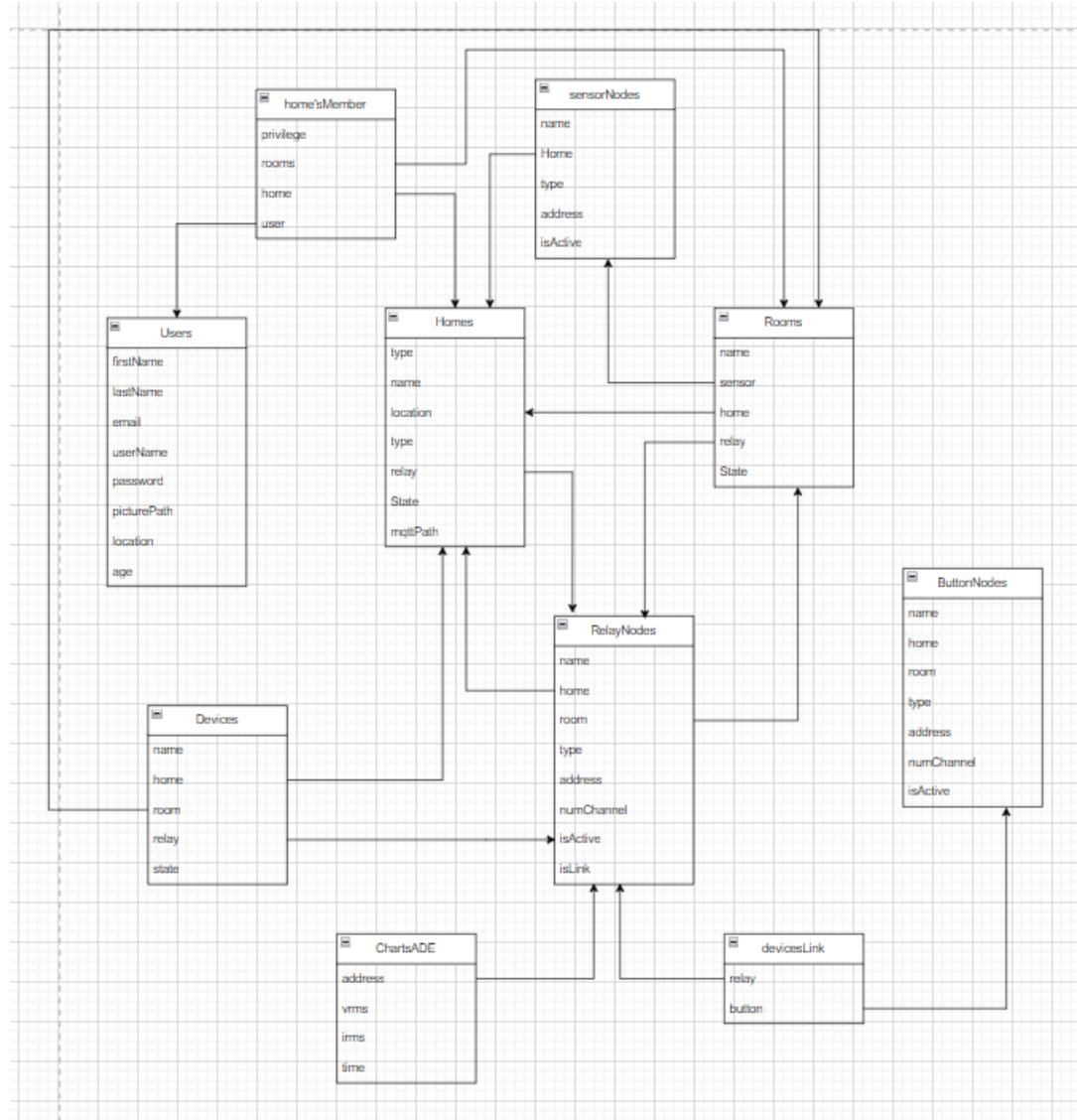
NoSQL cho phép mở rộng dữ liệu mà không cần tạo khóa ngoại, khóa chính hay kiểm tra tính ràng buộc... Có thể thấy, dạng cơ sở dữ liệu này có thể linh động, sẵn sàng bỏ qua tính toàn vẹn cứng nhắc của transaction hay dữ liệu nhằm đổi lấy hiệu suất, cùng với khả năng mở rộng tốt. Chính vì ưu điểm này mà nhiều Tập đoàn lớn rót vốn cho NoSQL như Facebook, Google, ...

Hoạt động của MongoDB:

- Nguyên tắc hoạt động của MongoDB là dưới một tiến trình dịch vụ ngầm và mở một cổng (mặc định là cổng 27017), để có thể tiếp nhận các yêu cầu truy vấn, thao tác; sau đó tiến hành xử lý.
- Mỗi bản ghi của MongoDB (document) được gắn một trường có tên “\_id” nhằm xác định tính duy nhất của bản ghi. Có thể hiểu id này như tên gọi của một bản ghi và dùng phân biệt chúng với các bản ghi khác. Đồng thời, nó còn được sử dụng cho mục đích truy vấn hoặc tìm kiếm thông tin. Trường dữ liệu “\_id” được tự động đánh chỉ mục (index) để đảm bảo tốc độ truy vấn đạt hiệu suất tối ưu.
- Mỗi truy vấn dữ liệu đều được ghi đệm lên bộ nhớ RAM nên các truy vấn sau đó sẽ diễn ra nhanh hơn. Bởi nó không cần đọc dữ liệu từ ổ cứng.
- Khi thực hiện thêm, xóa hay sửa bản ghi thì MongoDB đều mất 60s để ghi các dữ liệu được thay đổi từ RAM xuống ổ cứng. Điều này nhằm mục đích đảm bảo hiệu suất mặc định của chương trình.

#### 4.3.3. Thiết kế

Một Database [14] sẽ bao gồm rất nhiều bảng, mỗi bảng sẽ gồm rất nhiều thành phần của bảng đó. Mọi thành phần sẽ gồm các thuộc tính của thành phần.



Hình 4.9. Quan hệ giữa các bảng

Mỗi bảng só có quan hệ với các bảng khác như: quan hệ 1-1, quan hệ 1-n hay quan hệ n-n. Tùy theo quan hệ như thế nào mà sẽ có liên kết giữa các bảng là khác nhau. Thiết kế làm sao để phù hợp nhất để dễ dàng truy xuất dữ liệu.

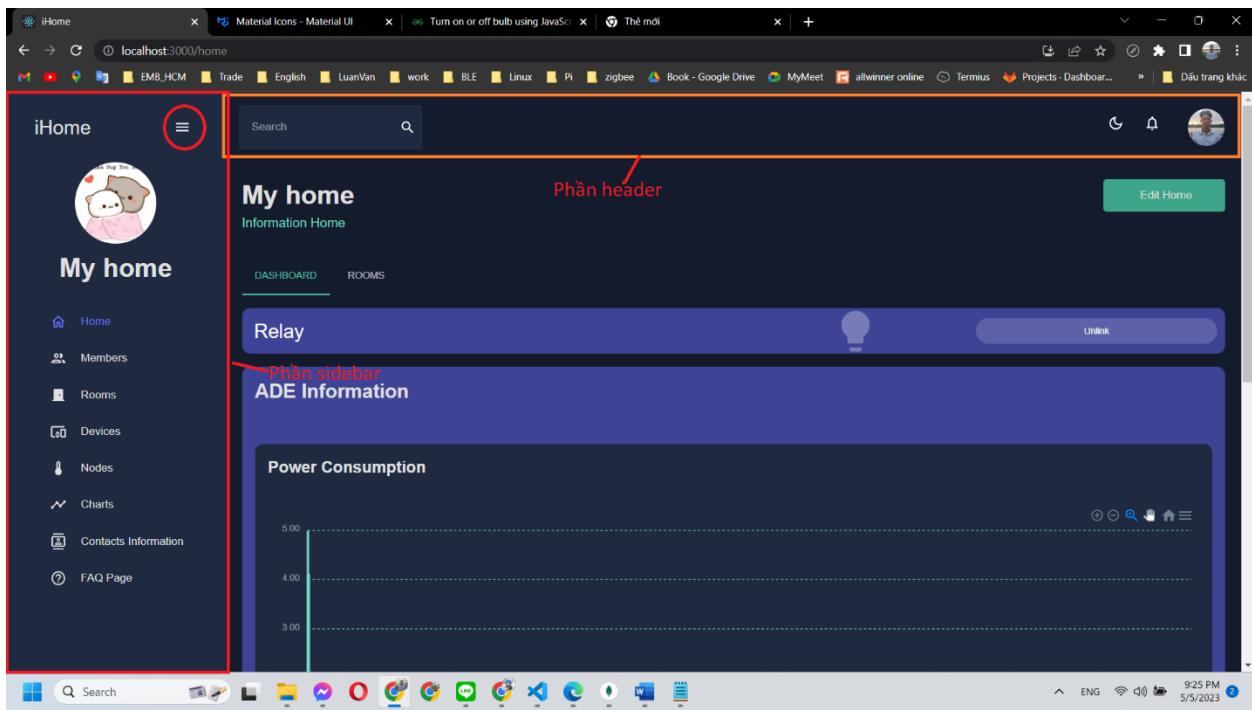
#### **4.4. Trang web thiết kế**

#### **4.4.1. Giao diện chung của Trang Web**

Giao diện chung của các trang sẽ gồm 3 phần chính:

- Phần header
  - Phần sidebar

- Phần nội dung chính của trang đó



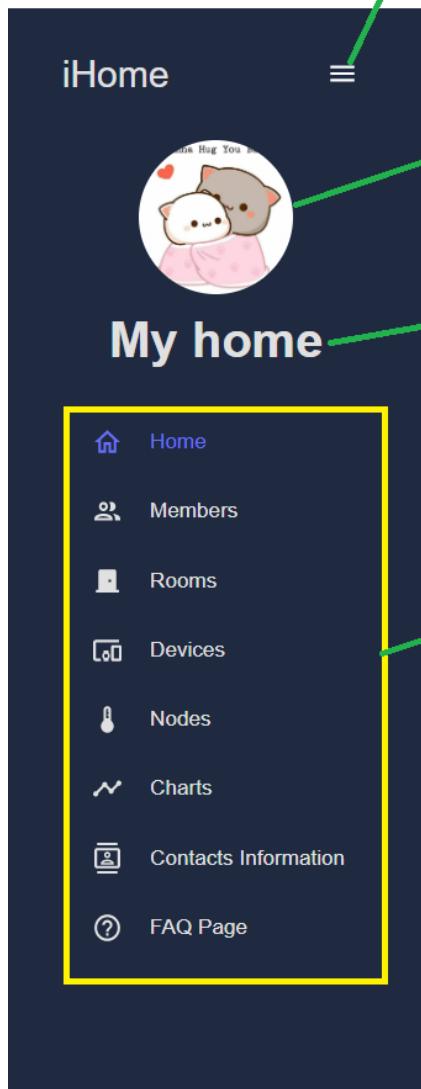
Hình 4.10. Giao diện chung của Trang Web

#### a. Phần header



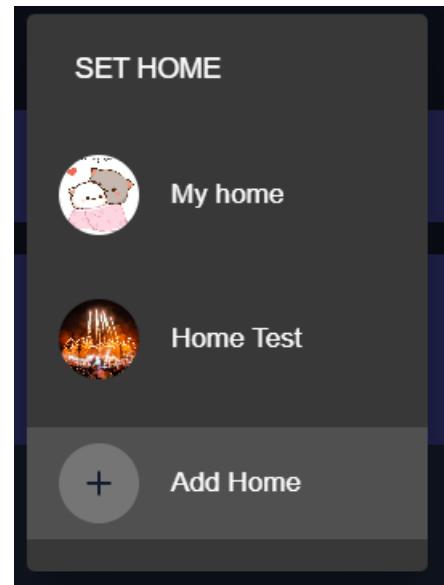
Hình 4.11. Phần header

- Icon mode: chuyển đổi giữa 2 chế độ sáng, tối cho trang web.
- Icon thông báo: hiển thị thông báo cho người dùng.
- Avatar user: click vào để chuyển đến trang user hoặc log out.
- Ô tìm kiếm: tìm kiếm thông tin.
- Phần sidebar



Hình 4.13. Sidebar

nhà của mình.



Hình 4.12. Home selection

#### b. Phần sidebar

Avatar home: Án vào để chuyển sang home khác hoặc tạo thêm home mới

Thanh chuyển trang sẽ chuyển đổi sang trang khác nhau tùy thuộc vào nhấn vào thanh nào. Từ đó người dùng sẽ tiện lợi khi chuyển qua chuyển lại các trang để điều khiển và xem thông tin ngôi

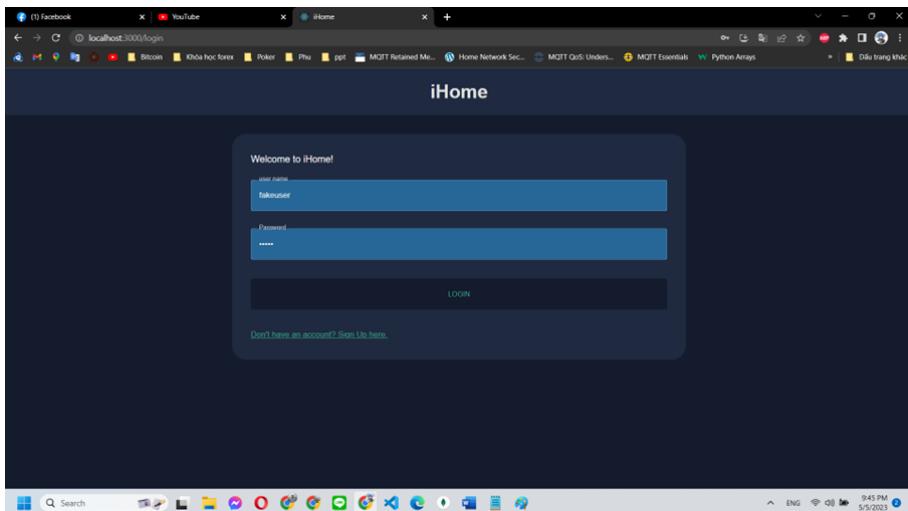
#### 4.4.2. Trang Đăng nhập, đăng ký

Trang đăng nhập để người dùng đập nhập vào tài khoản của họ. Khi người dùng chưa có tài khoản, người dùng có thể nhấn tạo tài khoản mới để có thể đưa qua trang tạo tài khoản và tạo một tài khoản mới để có thể truy cập vào trang.

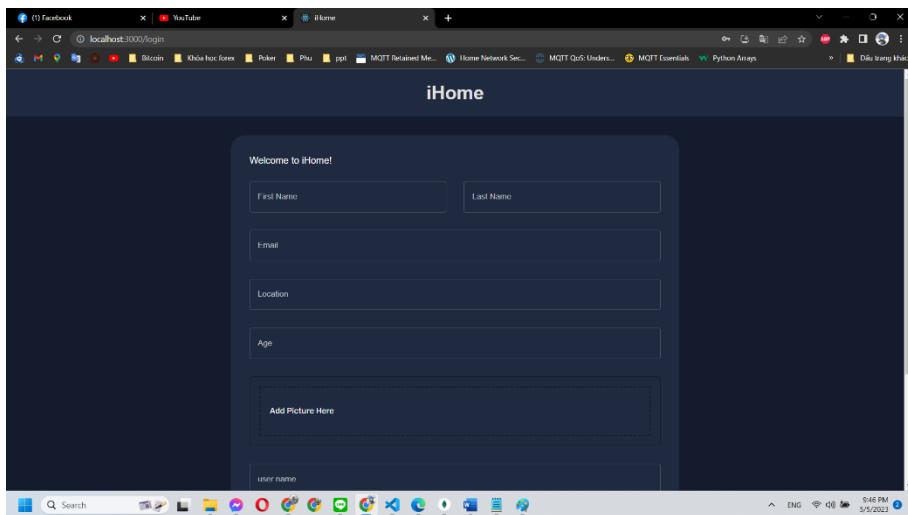
Nếu người dùng chưa có tài khoản thì không thể đăng nhập vào trang và xem được các thông tin của ngôi nhà hay thông tin các nhân.

#### 4.4.3. Trang chính Home

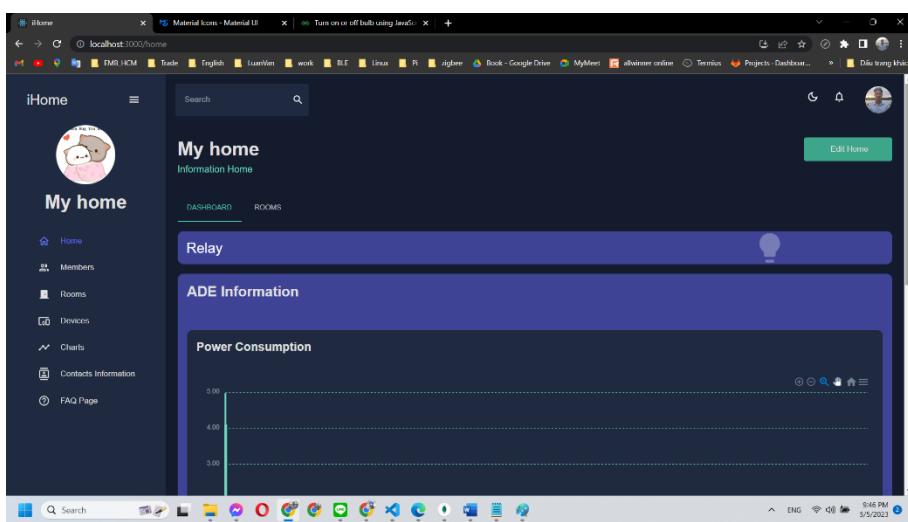
Trang Home được hiển thị thông tin relay tổng của ngôi nhà, hiển thị trạng thái và thông tin điện.



Hình 4.14. Trang Đăng Nhập



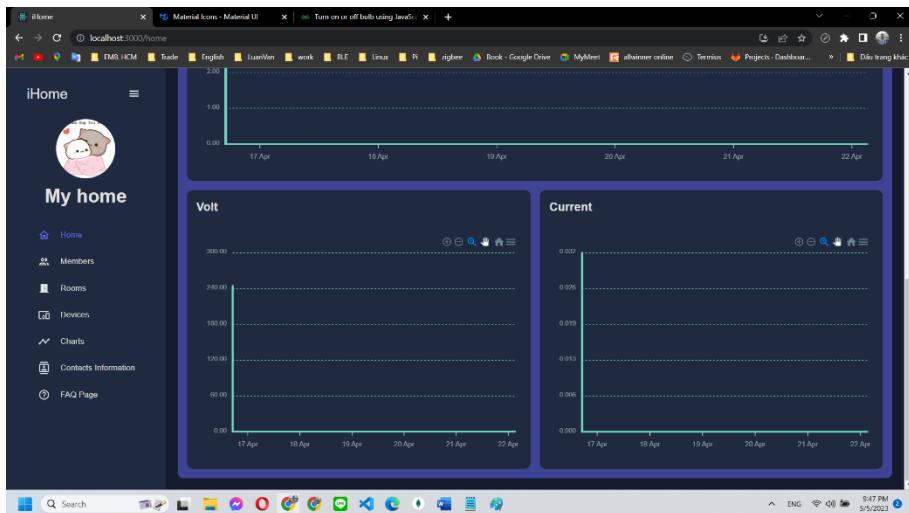
Hình 4.15. Trang Đăng Ký



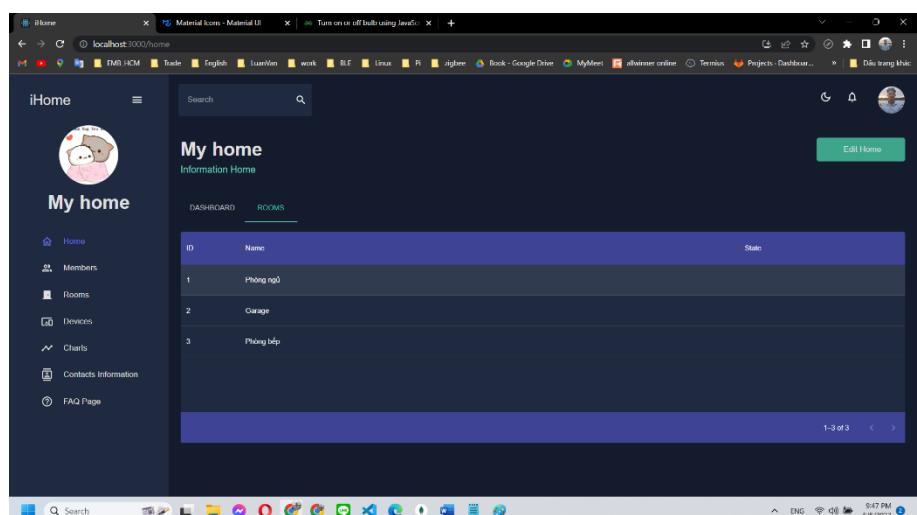
Hình 4.16. Trang Home DASHBOARD (1)

Trong trang Dashboard này sẽ hiển thị trạng thái của relay cũng như hiển thị tổng quan dòng điện, điện áp và cả công suất mà cả ngôi nhà đang sử dụng tại thời điểm hiện tại. Từ đó người dùng có một cái nhìn tổng quan tiêu thụ điện của ngôi nhà.

Trang này hiển thị tất cả các phòng của ngôi nhà và hiển thị trạng thái hoạt động relay của phòng đó. Người dùng có thể nhấp vào tên của từng phòng để chuyển qua các phòng đó, hiển thị các thông tin của phòng đầy đủ và rõ ràng hơn.



Hình 4.17. Trang Home DASHBOARD (2)



Hình 4.18. Trang Home ROOMS

#### 4.4.4. Trang thành viên trong nhà

Trang Members là trang hiển thị tất cả các thành viên trong nhà gồm cả một số thông tin cơ bản của một người dùng và quyền truy cập của người đó vào ngôi nhà. Cũng có thể thêm thành viên mới khi nhấn vào nút “Add member” ở phía trên để chuyển qua trang thêm thành viên mới.

Người dùng có thể tìm kiếm thành viên mới bằng cách nhấn bằng email của người dùng đó và nhấn nút “find” để tìm kiếm người dùng. Khi tìm được người dùng đó sẽ hiển thị ra và có thể nhấn vào người dùng đó và thêm thành viên đó vào.

ID	Name	Age	Email	Access Level
1	fake name	23	fakeemail@gmail.com	admin
2	Quoc Hoang	23	choquo135@gmail.com	USER
3	Quoc Hoang	24	choquo135@gmail.com	USER
4	Quoc Hoang	24	huong.duong2008@hcmut.edu.vn	manager

Hình 4.19. Trang các thành viên trong nhà

Enter Email: quoc.hoang2008@gmail.com

Hình 4.20. Trang thêm thành viên

#### 4.4.5. Trang tất cả các phòng

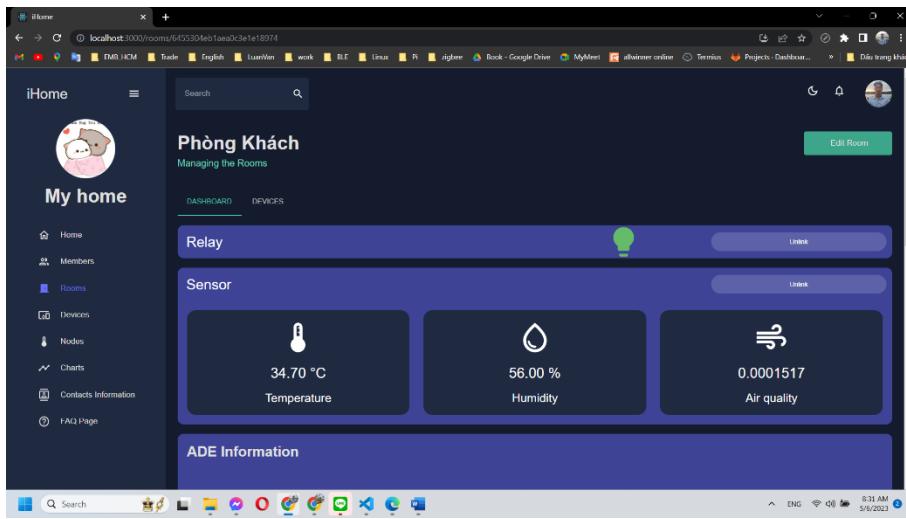
Khi vào trang Rooms thì sẽ hiển thị danh sách tất cả các phòng trong nhà có cả trạng thái relay của phòng đó. Có thể thêm hoặc xóa room tùy theo

ID	Name	Status	Delete
1	Phòng ngủ		
2	Garage		
3	Phòng bếp		

Hình 4.21. Trang hiển thị tất cả các phòng

nhu cầu của người dùng. Khi nhấn vào tên của bất kì phòng nào trang sẽ chuyển qua trang mới là trang của phòng đó.

Trong phòng thì có thể chỉnh sửa phòng và hiển thị trạng thái relay của phòng và hiển thị thông tin điện của phòng đó. Còn có cả một khung hiển thị trạng thái, thông tin của môi trường trong phòng khi liên kết với Node Sensor có khả năng thu thập thông tin của môi trường. Đó là trang mặc định khi đi vào một phòng. Người dùng có thể chuyển sang trang Devices để hiển thị danh sách các thiết bị trong phòng có thể điều khiển được.



Hình 4.23. Trang của một phòng (1)

ID	Name	Room	Status
1	Quạt	Phòng Khách	
2	Đèn 123	Phòng Khách	
3	Đèn	Phòng Khách	

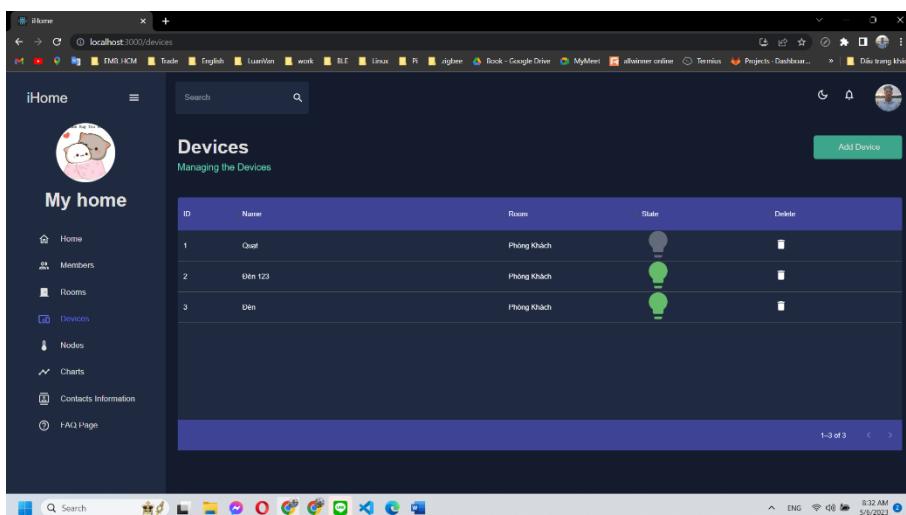
Hình 4.22. Trang của một phòng (2)

Khi vào trang devices này, người dùng có thể thấy được hiển thị danh sách thiết bị được điều khiển trong phòng và có thể điều khiển các thiết bị đó bằng cách nhấn vào trạng thái của nó để nó có thể chuyển đổi trạng thái. Không những thế khi người dùng nhấn vào tên thiết bị thì sẽ được chuyển sang trang của thiết bị đó.

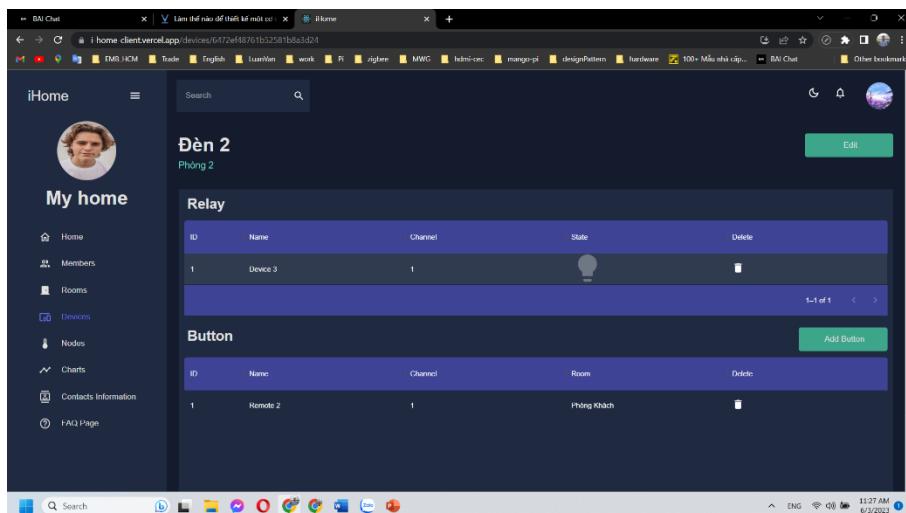
#### 4.4.6. Trang tất cả các thiết bị

Trang này được dùng để hiển thị tất cả các thiết bị và có thể điều khiển trạng thái của thiết bị này bằng cách nhấn vào trạng thái của nó để có thể chuyển đổi trạng thái thiết bị. Ở đây người dùng có thể thêm hoặc xóa thiết bị tùy theo nhu cầu của người sử dụng. Và khi nhấn vào tên của thiết bị thì người dùng sẽ được chuyển sang trang của thiết bị đó.

Khi ở trong trang của thiết bị đó, người dùng có thể sửa thiết bị, hoặc thêm liên kết với các nút nhấn để điều khiển và còn hiển thị trạng thái của thiết bị.



Hình 4.25. Trang tất cả thiết bị



Hình 4.24. Trang của từng thiết bị

#### 4.4.7. Trang các Node

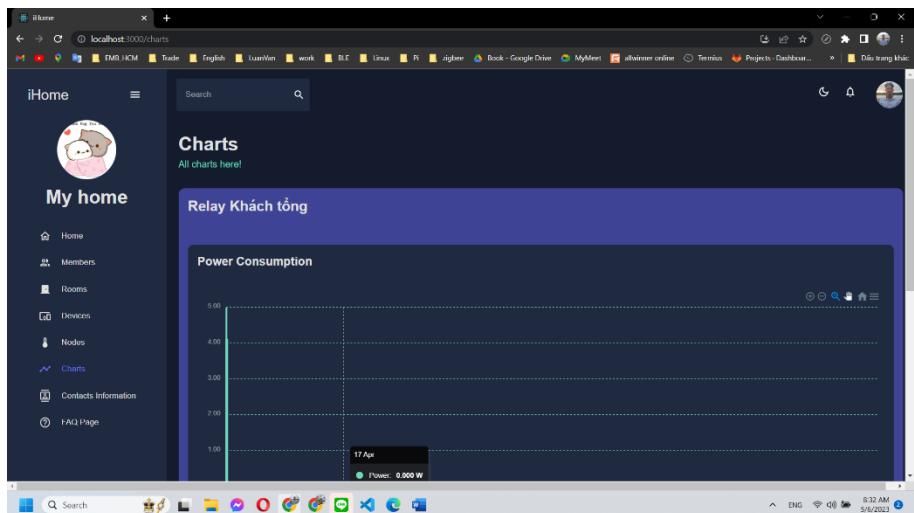
Trang này hiển thị tất cả các Node trong nhà kể cả Node relay, Node Button, Node Sensor, Node Relay Ade. Ở trang này người dùng có thể yêu cầu G-Hub cho thêm thiết bị mới bằng cách nhấn “Add Node” để yêu cầu G-Hub thêm thiết bị mới. Hoặc có thể remove thiết bị bằng cách nhấn icon delete ở mỗi node.

ID	Name	Room	Type Node	Number Channel	Delete
1	Relay Khách tổng	Phòng Khách	Relay	1	
2	Sensor Phòng Khách	Phòng Khách	Sensor		
3	Remote Phòng Khách	Phòng Khách	Button	3	
4	Sensor Phòng bếp	Phòng bếp	Sensor		
5	Relay Phòng Khách	Phòng Khách	Relay	3	
6	Relay Phòng bếp	Phòng bếp	Relay	3	

Hình 4.26. Trang hiển thị tất cả các Node

#### 4.4.8. Trang hiển thị các chart

Ở trong trang này, người dùng có thể nhìn thấy tất cả các biểu đồ thông số điện mà các thiết bị có trong nhà.

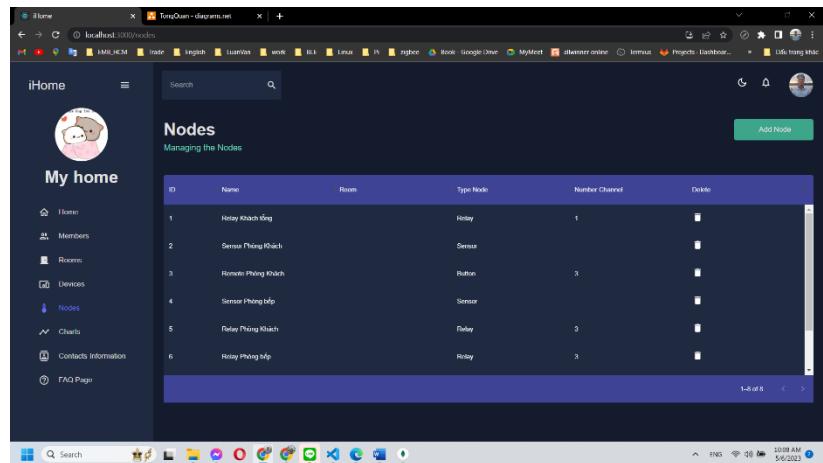


Hình 4.27. Trang hiển thị tất cả các chart

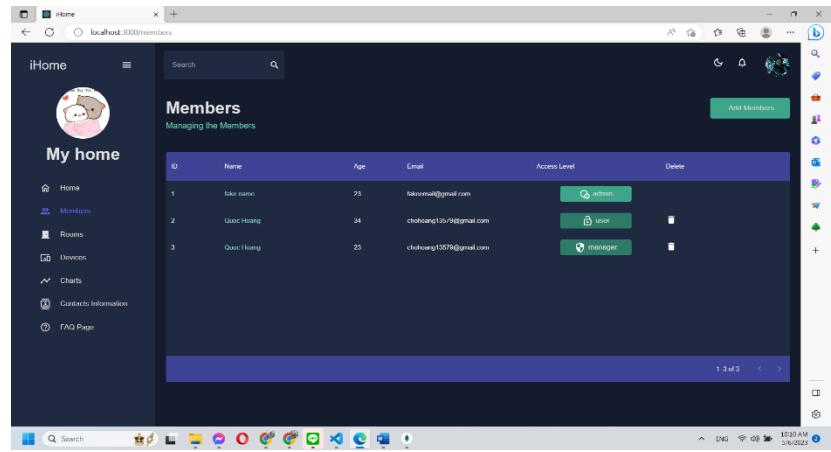
## 4.5. Chức năng phân quyền người dùng

Người dùng được phân theo 4 cấp bậc đối với một ngôi nhà:

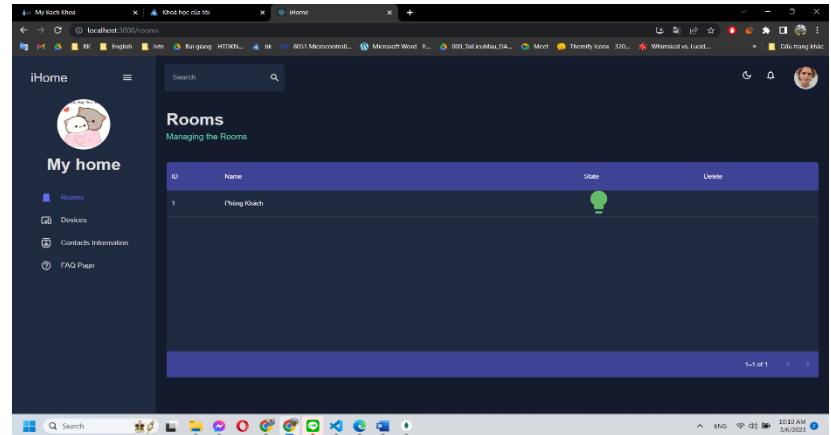
1. Host: Là người tạo ra ngôi nhà đó và có toàn quyền quyết định.
2. Admin: Những người được Host thêm vào và có được quyền thêm, sửa hay xóa các Node trong nhà. Và các quyền truy cập vào điều khiển các thiết bị hay là thêm, xóa thành viên.
3. Manager: Những người được thêm vào mà có thể được điều khiển các thiết bị hay truy cập vào trang các thành viên trong nhà.
4. User: Những người được thêm vào có thể điều khiển được các thiết bị trong một số phòng nhất định mà khi thêm vào đã được quyết định.



Hình 4.28. View trang web của Host và Admin



Hình 4.29. View trang web của Manager



Hình 4.30. View trang Web của User

## Chương 5.Thiết kế kết nối giữa thiết bị và Website

### 5.1. Giới thiệu về giao thức MQTT

#### 5.1.1. Khái niệm

MQTT [15] (Message Queue Telemetry Transport) là một giao thức truyền thông đơn giản và nhẹ, được sử dụng rộng rãi trong các ứng dụng IoT (Internet of Things) để kết nối và truyền tải dữ liệu giữa các thiết bị khác nhau. MQTT được phát triển bởi Eurotech và IBM vào năm 1999 và đã trở thành một chuẩn công nghiệp từ đó.

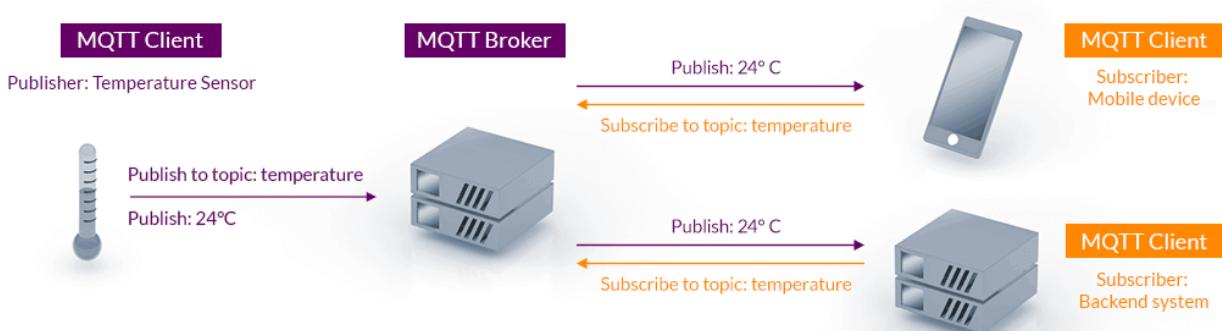
Để hiểu chi tiết hơn về MQTT, chúng ta cần tìm hiểu một số thuật ngữ theo thứ tự từ tổng quát đến chi tiết như sau:

- MQTT Broker: Là một máy chủ trung tâm hoạt động như một cầu nối giữa các thiết bị MQTT. MQTT Broker nhận và phân phối các thông điệp giữa các thiết bị. Đồng thời, nó quản lý các chủ đề (topic) MQTT, cho phép đăng ký và phân phối các thông điệp đến các khách hàng được đăng ký.
- MQTT Client: Là một thiết bị hoặc ứng dụng sử dụng giao thức MQTT để kết nối, truyền và nhận dữ liệu từ MQTT Broker. MQTT Client gửi các thông điệp đến MQTT Broker bằng cách đăng ký các chủ đề MQTT để nhận thông điệp quan tâm tới hoặc gửi các thông điệp tới MQTT Broker bằng cách đăng ký các chủ đề có sẵn trên MQTT Broker.
- MQTT Topic: Là một chuỗi định danh unique identify các thông điệp, giúp MQTT Broker theo dõi và phân phối chúng. MQTT Topic có thể đại diện cho trạng thái của một thiết bị như nhiệt độ, độ ẩm hoặc được sử dụng để kích hoạt một hành động nào đó.
- MQTT Message: Là dữ liệu được truyền tải giữa MQTT Broker và MQTT Client. Mỗi thông điệp được gửi đến một chủ đề (topic) cụ thể và có thể chứa bất kỳ dữ liệu nào.
- QoS (Quality of Service): Là chất lượng dịch vụ, là một cách để giải quyết các vấn đề về độ chính xác và độ tin cậy của dữ liệu được truyền qua MQTT. MQTT hỗ trợ ba

cấp độ QoS, từ QoS 0 cho đến QoS 2, giúp đảm bảo tính đúng đắn và đáp ứng thời gian thực của dữ liệu.

- MQTT Publish/Subscribe: Là hai phương thức cơ bản trong giao thức MQTT. Phương thức Publish được sử dụng khi một MQTT Client gửi một thông điệp lên MQTT Broker thông qua một chủ đề. Trong khi đó, phương thức Subscribe được sử dụng khi một MQTT Client đăng ký nhận các thông điệp từ một chủ đề cụ thể trên MQTT Broker.
- Retained Message: Là một kiểu thông điệp MQTT đặc biệt, được sử dụng để gửi thông điệp đến MQTT Broker và lưu giữ cho đến khi một MQTT Client kết nối và yêu cầu thông điệp đó. Khi một MQTT Client đăng ký lấy dữ liệu từ một chủ đề, niềm tin (retain) được gửi tới client với thông tin cuối cùng được lưu trữ trên broker.

### 5.1.2. Cách hoạt động



Hình 5.1. Cách thức hoạt động của MQTT

Cách hoạt động của MQTT được mô tả như sau:

Trong hệ thống MQTT, các thiết bị sử dụng giao thức này được xác định dưới dạng MQTT Client. Một MQTT Client có thể gửi hoặc nhận các thông điệp MQTT được gửi giữa các thiết bị thông qua MQTT Broker.

Broker hoạt động như một máy chủ trung tâm, là nơi các thông điệp MQTT được gửi và nhận. MQTT Broker đảm bảo rằng các thông điệp được gửi đến đúng địa chỉ đích thông qua các chủ đề (topics) tương ứng.

MQTT Client đăng ký chủ đề (subscribe) mà nó muốn lắng nghe thông điệp MQTT. Khi một MQTT Client gửi thông điệp (publish) đến một chủ đề (topic), MQTT Broker gửi lại thông điệp được gửi đến tất cả các MQTT Client đã đăng ký chủ đề đó.

MQTT Client sử dụng các gói tin MQTT để kết nối và truyền tải dữ liệu giữa các thiết bị. Các gói tin MQTT bao gồm các trường cho các thông số như chủ đề (topic), nội dung thông điệp (message payload), mức độ quan tâm (QoS - Quality of Service), ...vv.

MQTT hỗ trợ ba cấp độ QoS để đảm bảo tính đúng đắn và đáp ứng thời gian thực của dữ liệu. Các cấp độ QoS bao gồm:

- QoS 0 (at most once): Dữ liệu được gửi một lần, không đảm bảo tính đúng đắn hay đáp ứng thời gian thực.
- QoS 1 (at least once): Dữ liệu được gửi ít nhất một lần và đảm bảo tính đúng đắn, nhưng có thể được gửi nhiều hơn một lần.
- QoS 2 (exactly once): Dữ liệu đảm bảo được gửi chính xác một lần duy nhất, bằng cách sử dụng một quy trình xác nhận hai bước.

MQTT cũng hỗ trợ Retained Message - một kiểu thông điệp MQTT đặc biệt, được sử dụng để gửi thông điệp đến MQTT Broker và lưu giữ cho tới khi được yêu cầu. Khi một MQTT Client đăng ký lấy dữ liệu từ một chủ đề, tin này (retain) được gửi tới Client với thông tin cuối cùng được lưu trữ trên broker.

Tóm lại, MQTT hoạt động theo mô hình Publish/Subscribe, cho phép các thiết bị IoT truyền tải và nhận dữ liệu từ MQTT Broker thông qua các chủ đề (topics) tương ứng. MQTT hỗ trợ ba cấp độ QoS để đảm bảo tính đúng đắn và đáp ứng thời gian thực cho dữ liệu truyền tải.

## 5.2. Thiết kế dữ liệu truyền thông qua MQTT

### 5.2.1. Thiết kế MQTT Broker, MQTT Client

#### a. MQTT Broker

Có nhiều cách để tạo một MQTT broker, tùy thuộc vào mục đích và yêu cầu của bạn. Dưới đây là một số phương pháp phổ biến:

- Cài đặt một MQTT broker trên máy chủ của bạn: Bạn có thể cài đặt các giải pháp broker như Mosquitto hoặc EMQ X trực tiếp trên máy chủ của bạn. Để làm điều này, bạn cần có kiến thức về quản trị hệ thống và mạng, để có thể cấu hình và bảo trì broker.
- Sử dụng các dịch vụ cloud MQTT: Nhiều nhà cung cấp dịch vụ đám mây như AWS, Google Cloud và Microsoft Azure cung cấp các dịch vụ MQTT broker, cho phép bạn triển khai các broker mà không cần quản lý cơ sở hạ tầng. Điều này có thể là lựa chọn tốt nếu bạn không muốn tự quản lý cơ sở hạ tầng của mình hoặc muốn tích hợp với các dịch vụ đám mây khác.
- Sử dụng các dịch vụ third-party MQTT: Ngoài các dịch vụ đám mây, có nhiều nhà cung cấp dịch vụ MQTT broker khác nhau, cho phép bạn sử dụng broker của họ thông qua API hoặc giao thức MQTT. Điều này có thể là một lựa chọn tốt nếu bạn muốn triển khai nhanh chóng và đơn giản mà không cần quan tâm đến việc quản lý cơ sở hạ tầng.

Để có thể tăng cường bảo mật và sử dụng hợp lý một số hoạt động của MQTT Broker thì sẽ cấu hình MQTT Broker. Và mosquito là một lựa chọn hợp lý để cài đặt và sử dụng. Mosquitto là một MQTT broker mã nguồn mở, được phát triển bởi Eclipse Foundation. Nó cung cấp một giải pháp broker nhỏ gọn và hiệu quả cho các hệ thống IoT và các ứng dụng máy tính đám mây.

Một số tính năng của Mosquitto bao gồm:

- Hỗ trợ các phiên bản MQTT 3.1 và MQTT 3.1.1.
- Có khả năng kết nối với hàng ngàn clients MQTT cùng lúc.
- Có tính năng xác thực và mã hóa thông tin qua SSL/TLS.
- Có tính năng duy trì trạng thái kết nối khi client thoát ra và tiếp tục khi client kết nối lại.
- Có thể chạy trên các hệ điều hành khác nhau như Windows, Linux và macOS.

Mosquitto có thể được cài đặt và sử dụng miễn phí. Ngoài ra, Mosquitto có một số thư viện API cho phép các lập trình viên tạo các ứng dụng MQTT theo ý muốn.

Với các tính năng mạnh mẽ và dễ dàng sử dụng, Mosquitto đã được sử dụng rộng rãi trong các ứng dụng IoT, các hệ thống máy tính đám mây và các ứng dụng nhiều client.

Với mục đích thử nghiệm hoạt động ở đây sẽ sử dụng broker: test.mosquitto.org để thử nghiệm hoạt động của hệ thống. test.mosquitto.org là một MQTT broker công cộng miễn phí cho phép các lập trình viên và kỹ sư IoT thử nghiệm ứng dụng của họ với giao thức MQTT. Nó được quản lý bởi nhóm phát triển Mosquitto và được sử dụng rộng rãi để kiểm tra các ứng dụng MQTT trước khi chúng được triển khai trên các broker MQTT riêng.

Thông tin kết nối đến test.mosquitto.org:

- Địa chỉ: test.mosquitto.org
- Cổng: 1883 (MQTT) hoặc 8883 (MQTTS)

#### b. MQTT Client

Trong hệ thống sẽ có hai client sẽ giao tiếp với nhau đó là phía back-end và G-Hub.

Đối với phía web thì back-end sẽ là một mqtt-client để kết nối tới mqtt-broker và giao tiếp với phía các thiết bị.

Đối với phía các thiết bị thì G-Hub sẽ là một mqtt-client để kết nối với mqtt-broker và giao tiếp với phía trang web.

#### c. SSL

SSL (Secure Sockets Layer) là một giao thức bảo mật mạng được sử dụng để bảo vệ dữ liệu truyền tải trên mạng, bao gồm cả dữ liệu do người dùng nhập vào trên các trang web, email, thông tin tài khoản ngân hàng và thẻ tín dụng.

SSL tạo ra một kết nối bảo mật giữa một máy tính khách và một máy chủ web bằng cách sử dụng chứng chỉ SSL. SSL sử dụng một phương thức gọi là mã hóa để mã hóa dữ liệu và đảm bảo rằng dữ liệu chỉ được giải mã khi nó đến đúng đích.

Một chứng chỉ SSL bao gồm các thông tin như tên miền hoặc địa chỉ IP của máy chủ web, tên của tổ chức phát hành chứng chỉ, thời gian hiệu lực và mã hóa thông tin. Các trình duyệt web và ứng dụng đã được cấu hình để tự động xác thực các chứng chỉ SSL, giúp bảo vệ dữ liệu của người dùng khi sử dụng các dịch vụ trực tuyến.

Một số ứng dụng của SSL bao gồm:

- Bảo vệ thông tin cá nhân của người dùng khi sử dụng các dịch vụ trực tuyến, bao gồm thông tin tài khoản ngân hàng, thẻ tín dụng, số điện thoại và địa chỉ email.
- Bảo vệ dữ liệu và thông tin bí mật của doanh nghiệp khi chuyển tải qua mạng internet, bao gồm các thông tin về khách hàng, đối tác và nội bộ.
- Giúp bảo vệ các trang web khỏi các cuộc tấn công và tấn công giả mạo hoặc tràn đổ dữ liệu.
- Tạo một môi trường truyền tải an toàn cho các ứng dụng web như email, dịch vụ lưu trữ đám mây, giao dịch trực tuyến, và các ứng dụng khác.

### **5.2.2. Thiết kế cấu trúc gói tin**

Thiết kế cấu trúc của topic: Mã Client + “/” + command + “/” + Thông tin thêm.

- Mã Client: Mỗi G-Hub sẽ có một mã riêng biệt để nhận dạng G-Hub đó
- Command: yêu cầu hoặc là thông tin, ví dụ: permit\_join: yêu cầu mở mạng để thêm thiết bị.
- Thông tin thêm: (có thể có hoặc không) địa chỉ của thiết bị.

#### a. Các gói tin từ G-Hub gửi lên Backend

G-Hub sẽ được cài đặt một gói tin lastwill để khi G-Hub mất kết nối thông báo cho Backend biết để hiển thị thông tin cho người dùng biết.

PAYLOAD: {

"home": "mã của G-Hub",

"status": "offline"

}

retainFlag: 1

Gói tin khi mới online và kết nối được với broker thông báo cho Backend biết để hiển thị thông tin cho người dùng biết.

PAYLOAD: {

"home": "mã của G-Hub",

"status": "online"

}

retainFlag: 1

Hai gói tin trên được gửi lên topic: “mã Backend”/status

Gói tin thông báo có thiết bị mới đã được thêm vào:

TOPIC: “mã Backend”/provision

PAYLOAD: {

```
"dev_addr": "địa chỉ của node",
"type": "Relay", "Button", "Sensor"
"numChannel": 3
```

}

retainFlag: 0

Gói tin thông báo trạng thái thiết bị:

TOPIC: “mã Backend”/device\_status/ + addr

PAYLOAD: {

```
"status": "online", "offline"
```

}

retainFlag: 1

Gói tin gửi thông tin của thiết bị:

TOPIC: “mã Backend”/telemetry/ + dev\_addr

Về phần PAYLOAD sẽ phụ thuộc vào loại thiết bị nào mà gói tin sẽ khác nhau:

- Với Node Sensor

PAYLOAD: {

```
"temperature": 25,
"humidity": 40,
"air_quality" : 0.2
```

}

- Với Node Relay:

PAYLOAD: {

```
"status1": "ON"/"OFF",
```

```
        "status2": "ON"/"OFF",
        "status...": "ON"/"OFF"
    }
```

- Với Node Relay Ade:

```
PAYLOAD: {
    "status": true,
    "irms": 25,
    "vrms": 40,
    "power": 0.2
}
```

Với các gói tin này thì retainFlag: 1

#### b. Các gói tin Backend gửi xuống G-Hub

Gói tin thêm node mới để yêu cầu G-Hub mở mạng để thêm thiết bị mới.

TOPIC: “mã của G-Hub”/permit\_join

```
PAYLOAD: {
    "command": "permit_join"
}
```

retainFlag: 0

Gói tin yêu cầu một Node nào đó rời khỏi mạng.

TOPIC: “mã của G-Hub”/leave\_req

```
PAYLOAD: {
    "dev_addr": ".." (địa chỉ của thiết bị rời đi)
}
```

retainFlag: 0

Gói tin điều khiển thiết bị

TOPIC: “mã của G-Hub”/control

```
PAYLOAD: {
```

```
        "dev_addr": "00124B000418FD81", (Địa chỉ node relay)
        "channel": 1,2,3, (kênh trên node cần điều khiển)
    }
```

retainFlag: 0

Gói tin liên kết Relay và Nút nhấn

TOPIC: “mã của G-Hub”/link\_dev

PAYLOAD: {

```
        "btn_addr": "00124B000418FD81", (Địa chỉ node button)
        "btn_channel": 1,2,3, (channel button liên kết)
        "relay_addr": "00124B000418FD81", (Địa chỉ node relay)
        "relay_channel": 1,2,3, (channel relay liên kết)
    }
```

retainFlag: 0

Gói tin hủy liên kết giữa Relay và Nút nhấn

TOPIC: “mã của G-Hub”/disconnect\_dev

PAYLOAD: {

```
        "btn_addr": "00124B000418FD81", (Địa chỉ node button)
        "btn_channel": 1,2,3, (channel button liên kết)
        "relay_addr": "00124B000418FD81", (Địa chỉ node relay)
        "relay_channel": 1,2,3, (channel relay liên kết)
    }
```

retainFlag: 0

## Chương 6. Thiết kế nhà thông minh cho nhà cấp 4 ở Việt Nam

### 6.1. Cấu trúc của ngôi nhà thiết kế

Ngôi nhà thiết kế lắp đặt các Node sẽ có diện tích là 7,78m ngang và 11,28m dọc. Độ cao là 3m tính từ dưới nền lên đến trần nhà.

Ngôi nhà sẽ bao gồm 5 phòng, 1 phòng khác, 2 phòng ngủ, 1 phòng ăn, phòng tắm và nhà vệ sinh ở chung một phòng. Và có diện tích của mỗi phòng như sau:

- Phòng khách: 4,28x6,8 m
- Phòng ngủ: 3,5x4,5 m
- Phòng ăn: 4,28x4,48m
- Phòng tắm: 3,5x2,28m



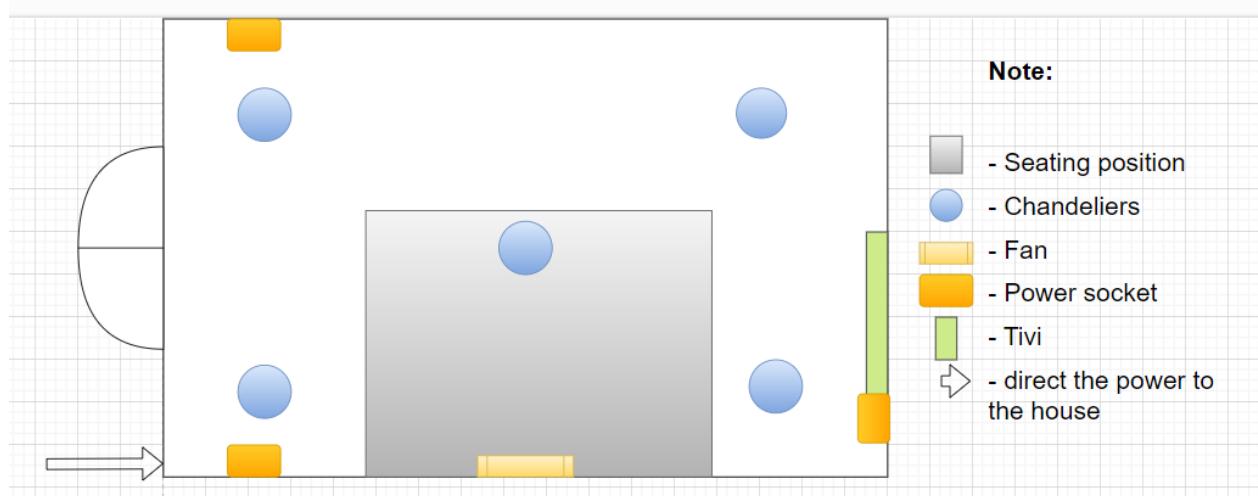
Hình 6.1. Cấu trúc ngôi nhà

Cấu trúc đặt các Node relay và Button của các phòng như sau:

- Mỗi phòng sẽ có một node Relay chính để từ các kênh trên relay đi đến vị trí của các thiết bị.
- Mỗi thiết bị sẽ có một nút nhấn điều khiển thiết bị đó ngay gần thiết bị.
- Sẽ có một vị trí điều khiển trong phòng (Bedside control cho phòng ngủ, Dashboard cho phòng khách) chỉ điều khiển các thiết bị trong phòng đó và được đặt cố định tại một vị trí. Có thể điều khiển đèn của phòng tắm hoặc phòng vệ sinh.
- Remote: điều khiển tất cả các thiết bị trong phòng nhưng có thể mang đi, di chuyển không cố định như thiết bị điều khiển cố định.
- Để điều khiển thiết bị của các phòng khác người dùng phải truy cập trang web.

## 6.2. Thiết kế vị trí lắp đặt các Node

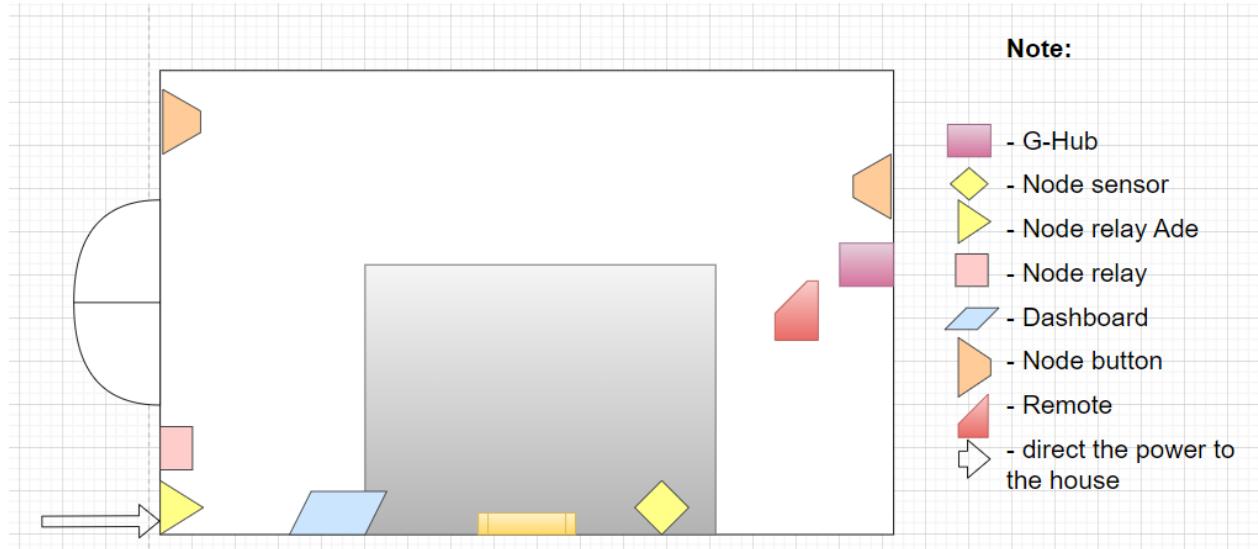
### 6.2.1. Phòng khách



Hình 6.2. Vị trí các thiết bị trong phòng khách

Trong phòng khách sẽ có 5 thiết bị cần điều khiển: Bộ đèn chum, 3 cái ổ cắm điện, quạt điện. -> sẽ cần một relay 5 kênh để điều khiển hết thiết bị trong phòng khách.

Bộ đèn chum sẽ được gắn từ trên trần nhà xuống. Các ổ cắm sẽ được đặt ở các vị trí cao 1m5 từ dưới nền nhà lên còn quạt thì sẽ là cao 2m từ dưới nền nhà tính lên.



Hình 6.3. Vị trí lắp đặt các Node trong phòng khách

G-Hub được đặt ngay bên tivi để có thể là vị trí trung tâm nhất của ngôi nhà.

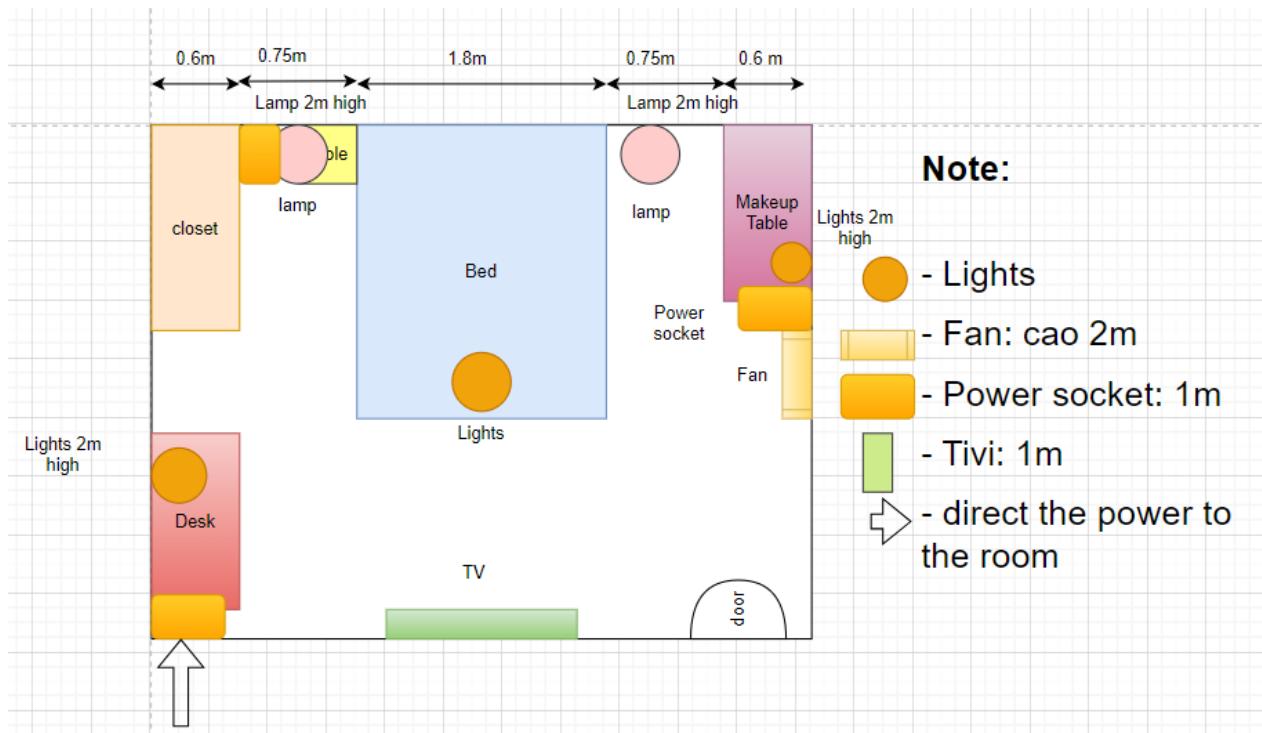
Node sensor được đặt tại vị trí gần ghế ngồi để có thể lấy được dữ liệu tại vị trí người dùng đang ngồi là tốt nhất.

Node Relay Ade được đặt ngay trên trần và tại vị trí đường dây điện đi vào nhà. Và sau đó dẫn đi qua các phòng.

Ngay bên cạnh Node Relay Ade là một node Relay Ade 5 kênh để có thể điều khiển được 5 thiết bị trong phòng.

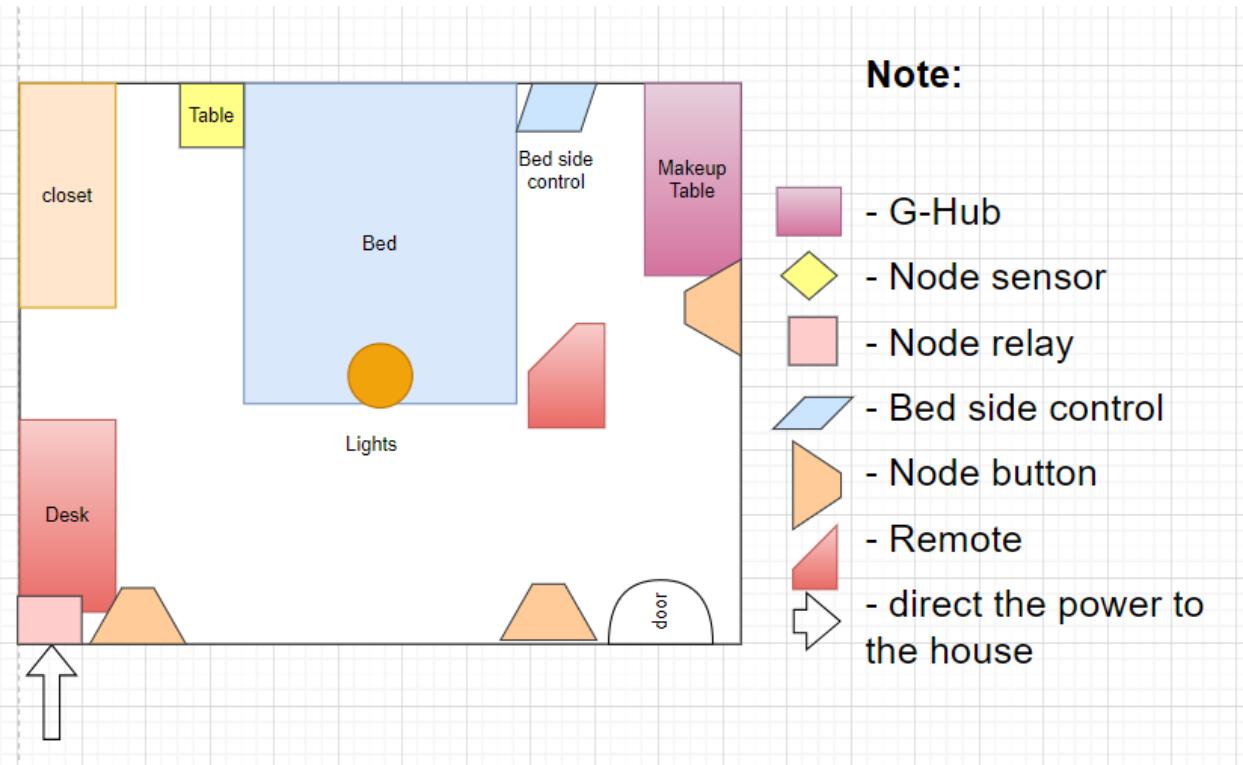
Một remote trong phòng để điều khiển các thiết bị trong phòng. Một Dashboard điều khiển các thiết bị trong phòng. Và có 2 Node Button được gán tường, một ở ngay bên cạnh cửa ra vào để điều khiển đèn và một còn lại ở gần tivi để điều khiển bóng đèn và ô cắm ở gần đó.

### 6.2.2. Phòng Ngủ



Hình 6.4. Vị trí các thiết bị trong phòng ngủ

Trong phòng ngủ sẽ có bóng chiếu sáng cho toàn phòng, một cặp bóng ngủ, một bóng đèn ở bàn làm việc, một bóng đèn ở bàn makeup, một cái quạt và có 3 ô cắm. Chiều cao của thiết bị được hiển thị trên hình 6.4 (chiều cao so với mặt đất).



Hình 6.5. Vị trí lắp đặt các Node trong phòng ngủ

Node Relay được đặt ngay trên trần và ngay lối vào của đường dây điện vào phòng sau đó phân tán ra các thiết bị.

Node Button được đặt ở 3 vị trí(cao 1m so với mặt đất):

- Node Button gần cửa: Điều khiển bóng đèn chiếu sáng
- Node Button gần bàn làm việc: Điều khiển bóng đèn ở bàn và cả ố cắm
- Node Button ở bàn Makeup: Điều khiển bóng đèn ở bàn makeup, ố cắm và cả quạt.

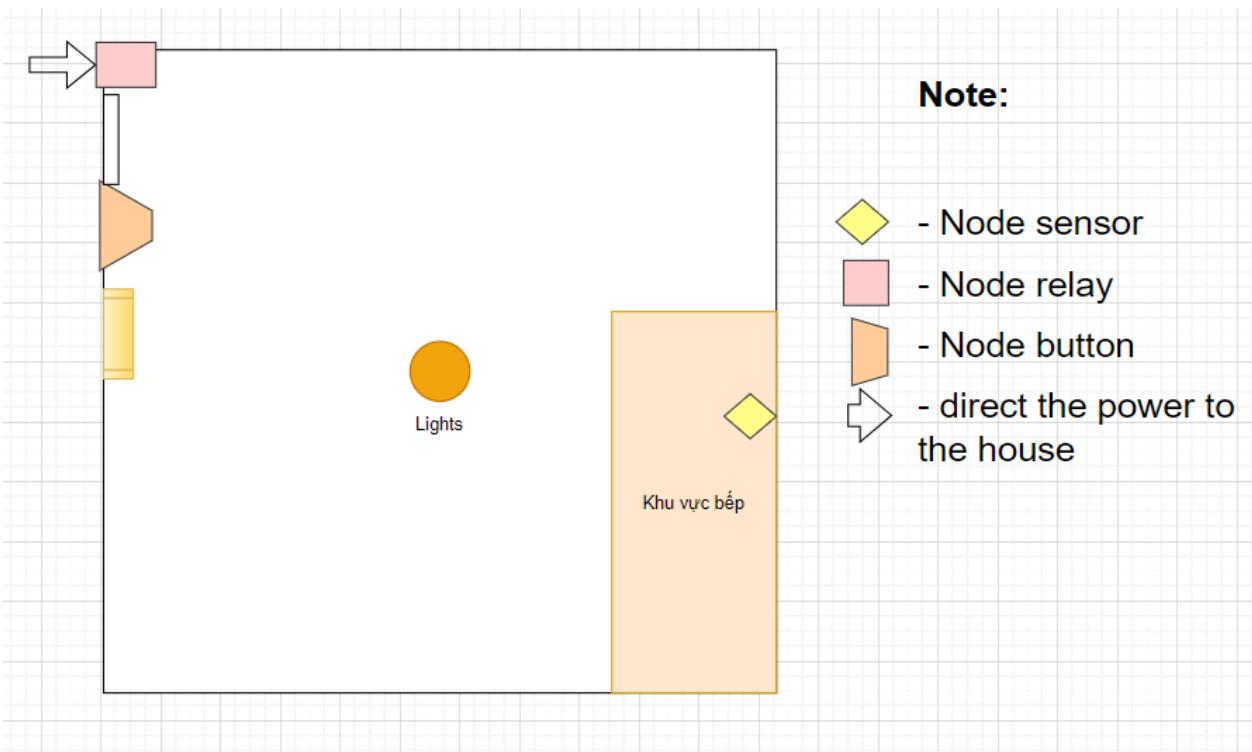
Bedside Control điều khiển tất cả các thiết bị trong phòng và bóng đèn của phòng vệ sinh và được đặt một bên cạnh giường. Remote sẽ điều khiển tất cả các thiết bị trong phòng nhưng có thể mang đi được.

Node sensor được đặt ngay bàn gần giường ngủ.

### 6.2.3. Phòng ăn

Trong phòng ăn sẽ có 2 thiết bị đó là quạt và bóng đèn chiếu sáng.

Sẽ cần Node Relay được đặt trên trần ngay vị trí dây điện vào phòng và một node Button ngay gần lối ra vào để người dùng thuận tiện bật / tắt thiết bị.



Hình 6.6. Vị Trí thiết bị và các Node trong phòng ăn

Node Sensor được đặt ở gần bếp.

#### 6.2.4. Phòng tắm

Phòng tắm chỉ cần một cái đèn chiếu sáng ở giữa phòng cho nên chỉ cần relay một channel điều khiển ngày thiết bị và đặt Node Button phía ngoài phòng tắm để tiện lợi điều khiển. Bóng đèn phòng tắm còn được điều khiển ở các phòng khác và cả trang web.

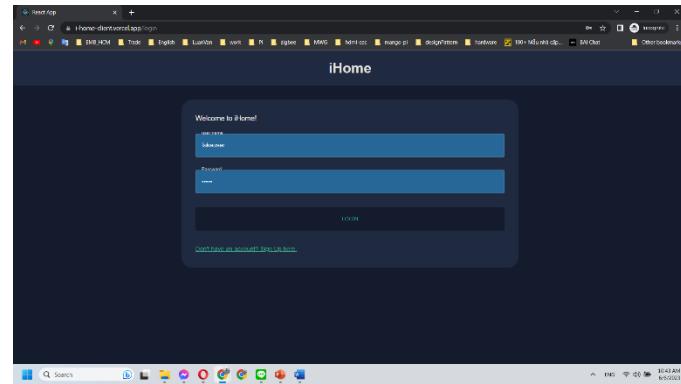
### 6.3. Các bước khởi tạo ban đầu trên trang web

Sau khi kết nối dây điện qua relay và đèn thiết bị. Chúng ta sẽ đến với bước tiếp theo liên kết relay với thiết bị và liên kết nút nhấn để điều khiển thiết bị không thông qua trang web.

### 6.3.1. Truy cập trang web, đăng ký và đăng nhập

Đầu tiên phải truy cập vào trang web để khởi tạo. Link trang web tham khảo: <https://i-home-client.vercel.app/>

Sau khi truy cập trang sẽ được chuyển đến phần đăng nhập. Nếu chưa có tài khoản người dùng phải đăng ký tài khoản và đăng nhập vào.



Hình 6.7. Trang đăng nhập

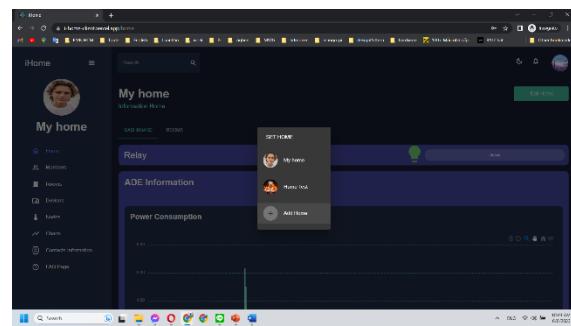
### 6.3.2. Tạo home

Sau khi đã đăng nhập xong, ở bên phần sidebar nhấp vào phần avatar của home thì sẽ xuất hiện một bảng chọn home (Hiển thị tất cả các Home mà người dùng là thành viên). Lúc này sẽ nhấp vào “Add home” để tạo thêm home mới.

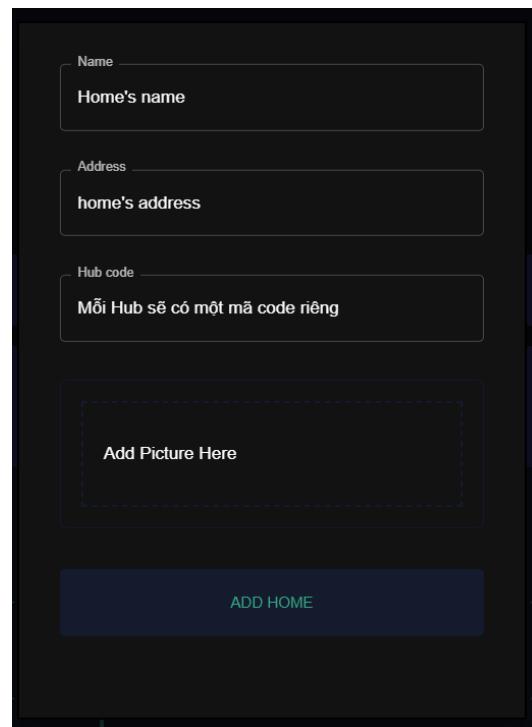
Sau khi nhấn xong, một bảng điền thông tin home sẽ hiện ra. Thông tin của home:

- Name: tên của ngôi nhà
- Address: địa chỉ của ngôi nhà
- Hub code: Mỗi G-Hub sẽ có một mã riêng dành cho nó. Chúng ta sẽ lấy thông tin đó để điền vào đây.
- Thêm 1 hình ảnh để nhận diện ngôi nhà.

Sau khi nhấn “ADD HOME” thì người dùng đã tạo được một ngôi nhà mới.



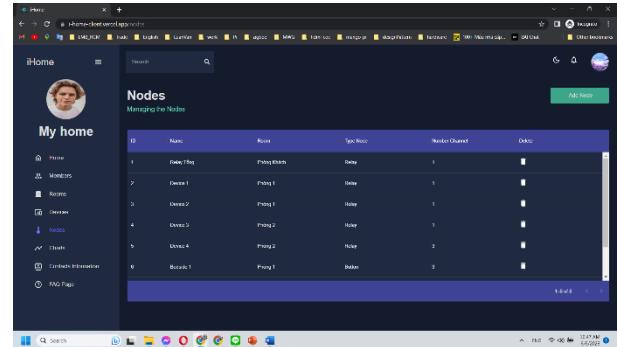
Hình 6.8. Bảng chọn home



Hình 6.9. Bảng điền thông tin ngôi nhà

### 6.3.3. Thêm tất cả các Node

Để thêm Node, G-Hub phải được kết nối mạng và kết nối MQTT thành công. Khi kết nối đã được hoàn thành. Chuyển trang web vào trang Nodes, khởi động Node cần thêm vào. Nhấn nút “Add Node” trên trang web và chờ. Nếu thêm Node thành công sẽ hiển thị Node mới đó, còn nếu không thành công một bảng thông báo sẽ xuất hiện. Sau khi Node đã được thêm vào thì thay đổi thông tin của Node đó. Làm tương tự với tất cả các Node còn lại và thêm tất cả các Node vào mạng.

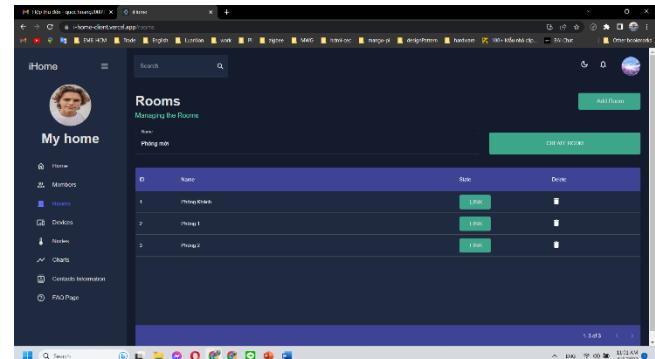


Hình 6.10. Trang Nodes

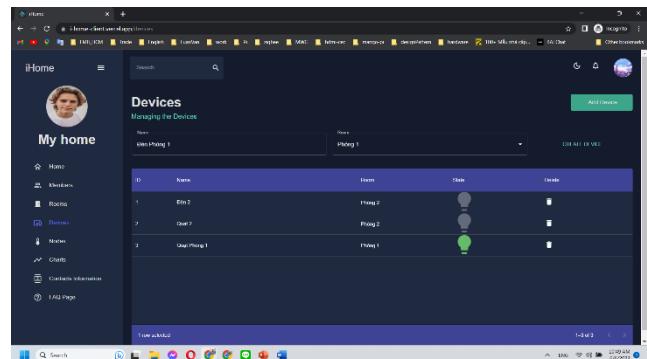
### 6.3.4. Thêm phòng và thêm thiết bị

Để thêm phòng, vào trang rooms và nhấn vào nút “Add Room” để tạo thêm một phòng mới. Thêm tất cả các phòng vào trang.

Tương tự với thêm phòng, để người dùng thêm thiết bị thì vào trang devices và nhấn nút “Add device” để thêm tất cả các thiết bị. Lựa chọn phòng và đặt tên cho thiết bị. Thêm tất cả các thiết bị của từng phòng.



Hình 6.11. Thêm phòng



Hình 6.12. Thêm thiết bị

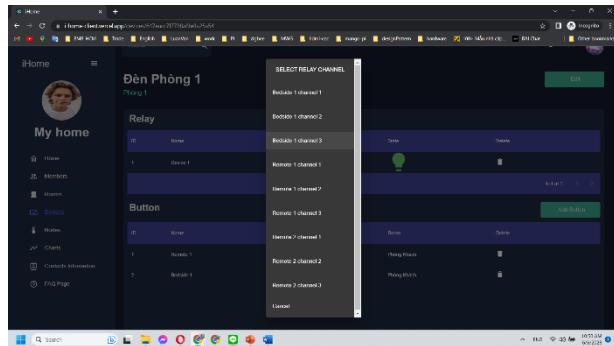
### 6.3.5. Liên kết thiết bị với relay, và liên kết với nút nhấn

Khi mới tạo thiết bị xong, thiết bị chưa được liên kết với relay nào hết cho nên sẽ có một nút “LINK” ở thiết bị đó. Người dùng sẽ nhấn vào nút “LINK” này để hiển thị ra tất cả các relay chưa liên kết với thiết bị sẽ hiển thị ra.

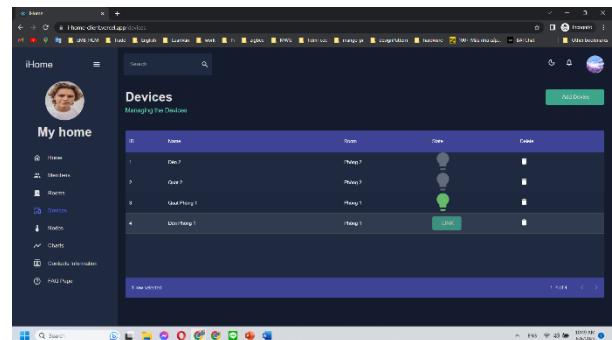
Bảng chọn relay (Hình 6.14) sẽ hiển thị ra tất cả các relay mà chưa liên kết với thiết bị. Lúc này, người dùng sẽ phải chọn relay mà đã nối trước khi tới thiết bị đó.

Sau khi đã chọn xong relay để liên kết, nhấn chọn vào tên thiết bị, để chuyển sang trang thiết bị (Hình 6.16).

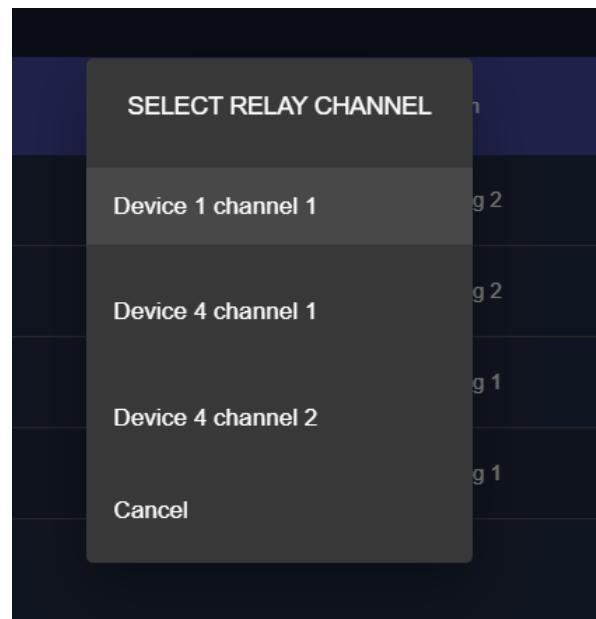
Ở trang thiết bị sẽ chia làm hai phần, phần relay và phần button. Relay đã được liên kết và Button hiển thị danh sách các nút nhấn đã liên kết với thiết bị. Nhấn “Add button” để hiện ra danh sách các nút nhấn.



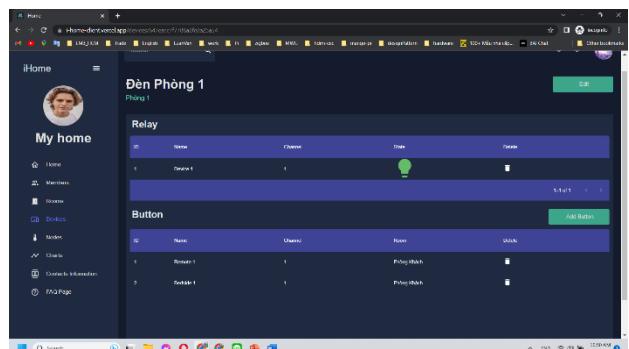
Hình 6.15. Danh sách nút nhấn



Hình 6.13. Link thiết bị với relay



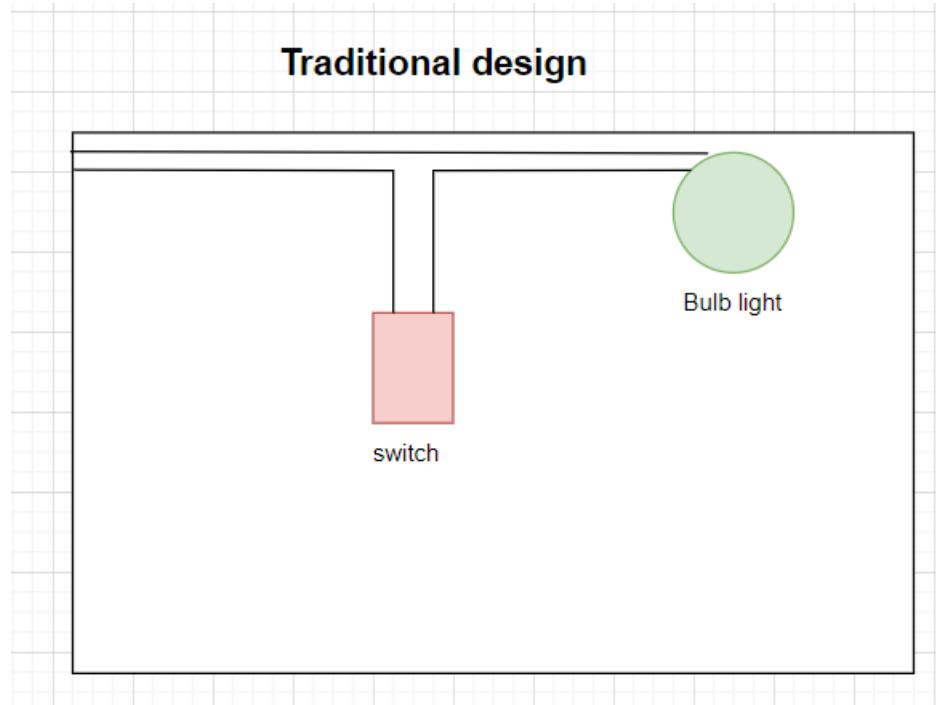
Hình 6.14. Bảng chọn relay



Hình 6.16. Trang Thiết bị

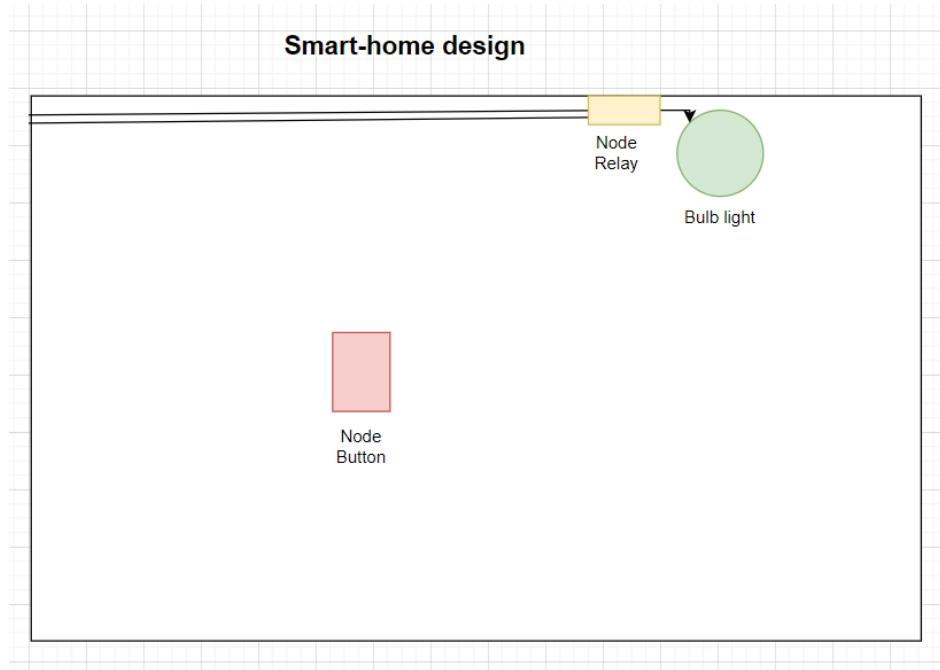
Lựa chọn nút nhấn nào muốn liên kết với thiết bị.

#### 6.4. So sánh thiết kế giữa truyền thống và smart home



Hình 6.17. thiết kế truyền thống

Theo như thiết kế truyền thống, để kết nối từ nguồn cho tới thiết bị thì cần phải có một dây đi xuống công tắc và đi lên lại thiết bị.



Hình 6.18. Thiết kế smart home

Thiết kế smart home, nối từ nguồn vào đi qua node relay và tới thiết bị. Một Node Button được đặt một vị trí khác để tương tác với người dùng.

### 6.3.1. Tính an toàn

Thiết kế truyền thông cho các công tắc điện trong gia đình gặp phải nguy cơ đáng kể đối với người dùng do tiếp xúc trực tiếp với nguồn cung cấp điện nguy hiểm 220V khi vận hành công tắc. Phương pháp truyền thông này có thể dẫn đến tai nạn điện ngẫu nhiên, gây nguy hiểm nghiêm trọng đến tính mạng con người. Tuy nhiên, thiết kế smart home đề xuất giảm thiểu rủi ro này bằng cách giới hạn tiếp xúc của người dùng chỉ với nguồn cung cấp điện không độc hại 5V, từ đó nâng cao tổng thể an toàn của hệ thống.

Việc áp dụng các công nghệ tiên tiến để tạo ra các giải pháp thông minh cho ngôi nhà đã mở ra cơ hội mới để làm cho cuộc sống của chúng ta trở nên an toàn và thuận tiện hơn. Giải pháp thông minh cho hệ thống điện trong nhà có thể được thiết kế để giảm thiểu những nguy cơ liên quan đến điện, đặc biệt là tai nạn điện ngẫu nhiên có thể xảy ra do sự tiếp xúc của người dùng với nguồn điện nguy hiểm.

Thiết kế smart home đề xuất cho hệ thống điện trong nhà giúp giảm thiểu nguy cơ tai nạn điện cho người dùng bằng cách sử dụng công nghệ tiên tiến như các công tắc điện không tiếp xúc trực tiếp với nguồn điện, giúp người dùng giữ khoảng cách an toàn với 5V điện không độc hại. Công nghệ này giúp người dùng cảm thấy yên tâm và an toàn hơn khi sử dụng các thiết bị điện trong gia đình.

### 6.3.2. Tính kỹ thuật

Việc điều khiển các thiết bị ở nhiều địa điểm khác nhau đem lại nhiều thách thức với thiết kế truyền thông cho các công tắc điện trong gia đình. Thiết kế smart home đề xuất cung cấp một giải pháp hiệu quả bằng cách cho phép dễ dàng kiểm soát các thiết bị trong các môi trường khác nhau. Sự tiện lợi trong việc kiểm soát này là một ưu điểm đáng chú ý của thiết kế đề xuất so với phương pháp truyền thông và được thực hiện thông qua việc sử dụng các công nghệ tiên tiến như giao thức truyền thông không dây và kết nối dựa trên internet. Kết quả là, thiết kế nhà thông minh đề xuất cung cấp cho người dùng tính linh hoạt

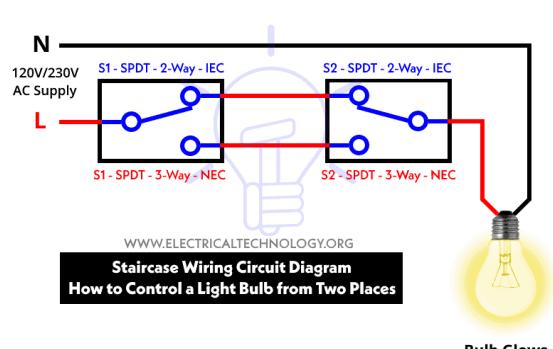
và tiện lợi hơn trong việc quản lý các thiết bị điện của họ, từ đó nâng cao trải nghiệm người dùng tổng thể.

Với sự phát triển của các công nghệ hiện đại, đặc biệt là công nghệ kết nối không dây và kết nối internet, việc điều khiển các thiết bị điện trong nhà thông minh trở nên dễ dàng hơn bao giờ hết. Thiết kế smart home đề xuất cung cấp cho người dùng khả năng điều khiển các thiết bị điện trong nhiều môi trường khác nhau, từ xa và đồng thời. Điều này mang lại tính tiện lợi và linh hoạt cho người dùng, giúp họ kiểm soát các thiết bị điện của mình một cách dễ dàng và thuận tiện hơn.

Thiết kế smart home đề xuất có thể được kết nối với internet, cho phép người dùng quản lý các thiết bị điện của mình từ bất kỳ đâu trên thế giới, chỉ cần có kết nối internet. Ngoài ra, các công tắc điện không dây giúp người dùng dễ dàng chuyển đổi các thiết bị từ xa mà không cần tiếp xúc trực tiếp với nguồn điện. Tính linh hoạt và tiện lợi của thiết kế nhà thông minh đề xuất giúp mang lại sự thoải mái và an toàn hơn cho người dùng, nhất là trong việc quản lý các thiết bị điện ở nhiều địa điểm khác nhau.

### 6.3.3. Tính kinh tế

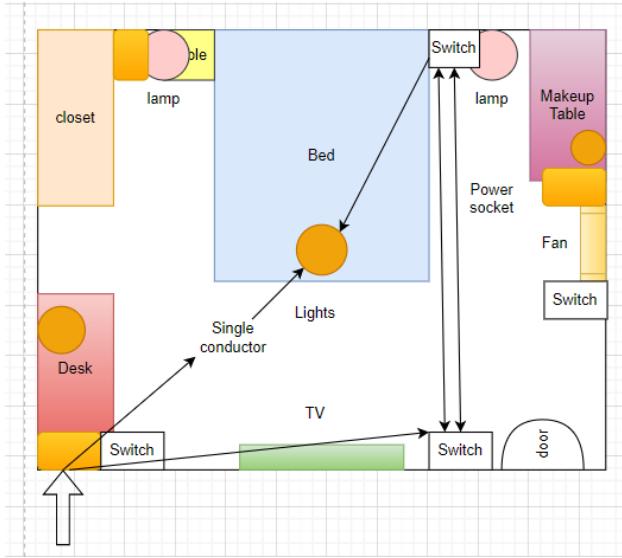
Với mục đích so sánh kinh tế, nghiên cứu này sẽ giới thiệu hai hệ thống điều khiển từ xa để vận hành cả hai lựa chọn thiết kế. Phòng ngủ sẽ được sử dụng làm cơ sở cho thiết kế.



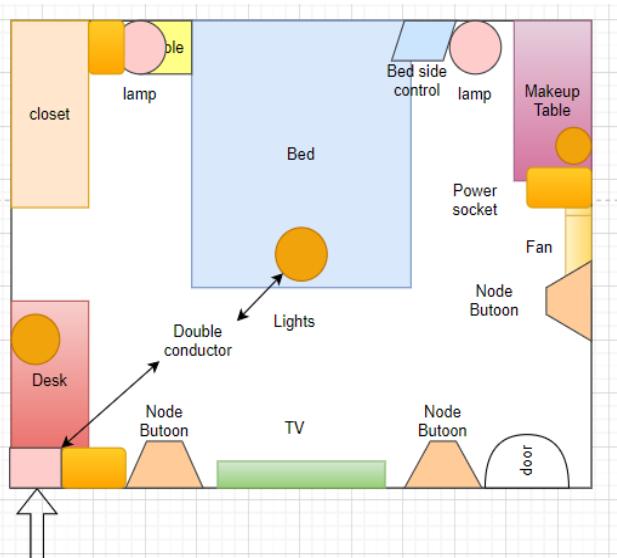
Hình 6.19. Cách thiết kế truyền thống 2 nơi điều khiển [16].

Device	Place 1	Place 2
Lights	Near door	Bedside
Lights Desk	In Desk	Bedside
Lamp	(No use)	Bedside
Lights Makeup Table	In Makeup Table	Bedside

Bảng 6.1. Bảng danh sách thiết bị và nơi điều khiển



Hình 6.21. Ví dụ đi dây trong nhà truyền thống



Hình 6.21. Ví dụ đi dây trong smart home

Chọn loại dây [17] có giá 4280 VND cho cả hai thiết kế. Theo tính toán của chúng tôi, số lượng dây cần thiết trong thiết kế truyền thống là 94,5m và trong thiết kế nhà thông minh là 37,3m.

Phương pháp truyền thống yêu cầu chi phí dây là 404.375 VND, 3 công tắc ba cực có giá 175.000 VND và ba hộp công tắc có giá 30.000 đồng, dẫn đến tổng chi phí là 610.000 VND. Thiết kế nhà thông minh yêu cầu chi phí dây là 160.000 VND, một bộ khuếch đại nút có giá 175.000 VND và hai nút nút có giá 300.000 VND, dẫn đến tổng chi phí cao hơn một chút là 635.000 VND. Tuy nhiên, các lợi ích của thiết kế nhà thông minh có thể vượt qua chi phí bổ sung này. Nhà thông minh cung cấp tiện ích điều khiển thiết bị từ điện thoại thông minh hoặc laptop thông qua một trang web.

Mặc dù có khác biệt về chi phí này, nhưng các lợi ích của sự tiện lợi và tính linh hoạt có sẵn trong thiết kế nhà thông minh cung cấp các ứng dụng tiềm năng hứa hẹn trong tương lai.

#### 6.3.4. Tính thẩm mỹ

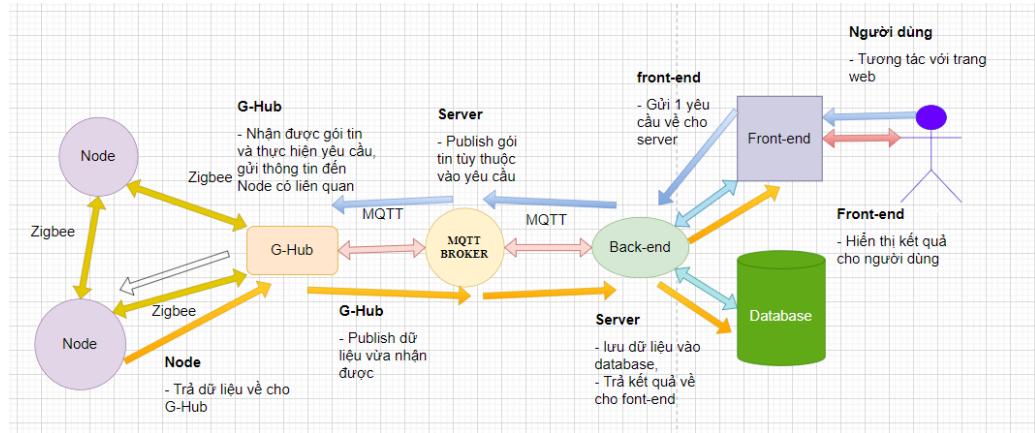
Phương pháp kết nối dây điện truyền thống yêu cầu các dây điện phải thông qua công tắc trước khi đến bóng đèn, phải thực hiện các biện pháp như đục tường hay giấu dây. Tuy nhiên, thiết kế được đề xuất chỉ yêu cầu một nút relay ở bất kỳ vị trí nào giữa nguồn và thiết

bị. Do đó, các biện pháp kết nối dây điện cần thiết cho phương pháp truyền thống bị loại bỏ, mang lại lợi ích bổ sung cho thiết kế được đề xuất.

**Kết luận:** Kết quả nghiên cứu của chúng tôi cho thấy rằng thiết kế nhà thông minh được đề xuất mang lại những lợi ích về an toàn, tính kỹ thuật và thẩm. Do đó, việc triển khai thiết kế nhà thông minh sẽ là một lựa chọn tốt nhất cho các gia đình muốn tận dụng tiềm năng của công nghệ trong cuộc sống hàng ngày.

# Chương 7. Kết quả thực hiện

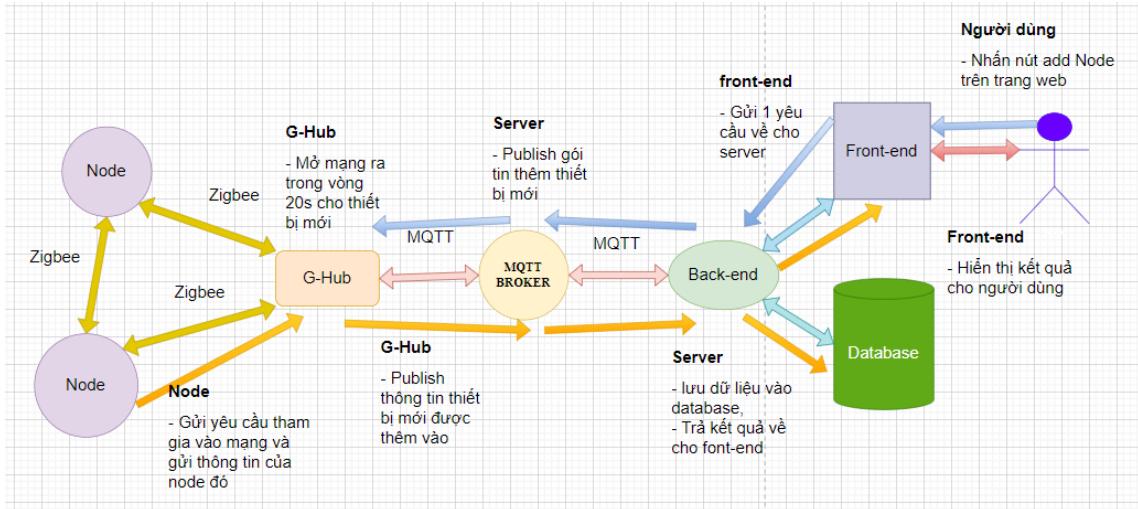
## 7.1. Luồng dữ liệu đi trong hệ thống



Hình 7.1. Luồng dữ liệu đi trong hệ thống khi người dùng tương tác trên trang Web

Khi người dùng thao tác trên trang web, bên phía front-end sẽ nhận hành động của người dùng và gửi yêu cầu về cho back-end. Từ những dữ liệu mà mình nhận được, back-end publish một gói tin và MQTT-Broker nhận gói tin và phân tán gói tin đó cho tất cả các client đã subscribe topic mà back-end đã publish. G-Hub nhận được gói tin và xử lý dữ liệu, gửi thông tin đến các Node liên quan. Sau đó Node sẽ trả dữ liệu về cho G-Hub và G-Hub tiếp tục Publish lên để Back-end nhận được dữ liệu để lưu vào database và hiển thị lên cho người dùng xem.

### a. Thêm node mới



Hình 7.2. Ví dụ về dữ liệu truyền đi trong hệ thống khi thêm Node

Người dùng nhấn vào nút “Add Node” trong trang tất cả các Node thì lúc đó front-end sẽ gửi một yêu cầu về phía back-end. Back-end publish một gói tin thêm thiết bị mới. G-Hub nhận được yêu cầu và mở mạng ra. Khi đó Node mới sẽ gửi một yêu cầu tham gia vào mạng, G-Hub chấp nhận và gửi thông tin nhận được đó lên phía Back-end thông qua MQTT-Broker. Back-end lưu dữ liệu mới đó vào database và hiển thị lên cho người dùng.

### b. Liên kết nút nhấn và relay

Để liên kết giữa nút nhấn và thiết bị người dùng phải vào trang của thiết bị đó và nhấn vào nút “Add button” lúc này sẽ hiển thị ra một danh sách các nút nhấn. Người dùng sẽ lựa chọn nút nhấn mà mình muốn liên kết với thiết bị.

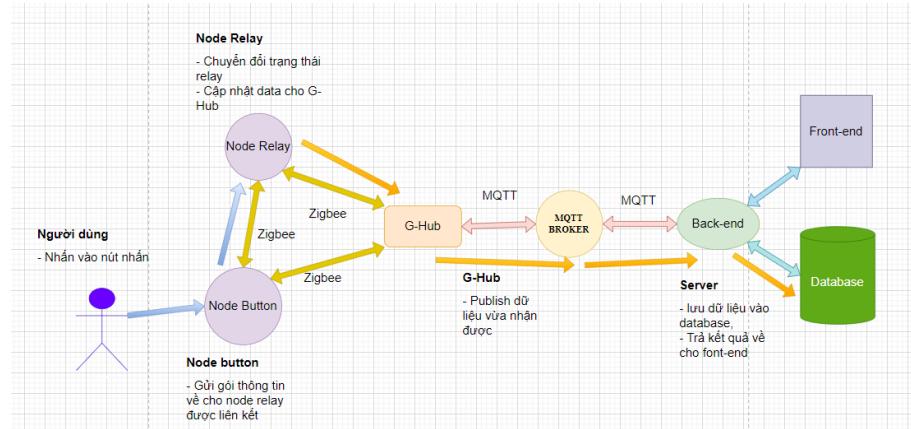
ID	Name	Channel	State	Delete
1	Relay Phòng Khách	1		

ID	Name	Channel	Room	Delete
1	Remote Phòng Khách	1	Phòng Khách	
2	Remote Phòng Khách	2	Phòng Khách	

Hình 7.3. Trang liên kết relay với button

Yêu cầu này sẽ được chuyển về cho back-end để back-end gửi yêu cầu xuống G-Hub, sau đó G-Hub sẽ liên kết relay điều khiển thiết bị và button đó lại với nhau.



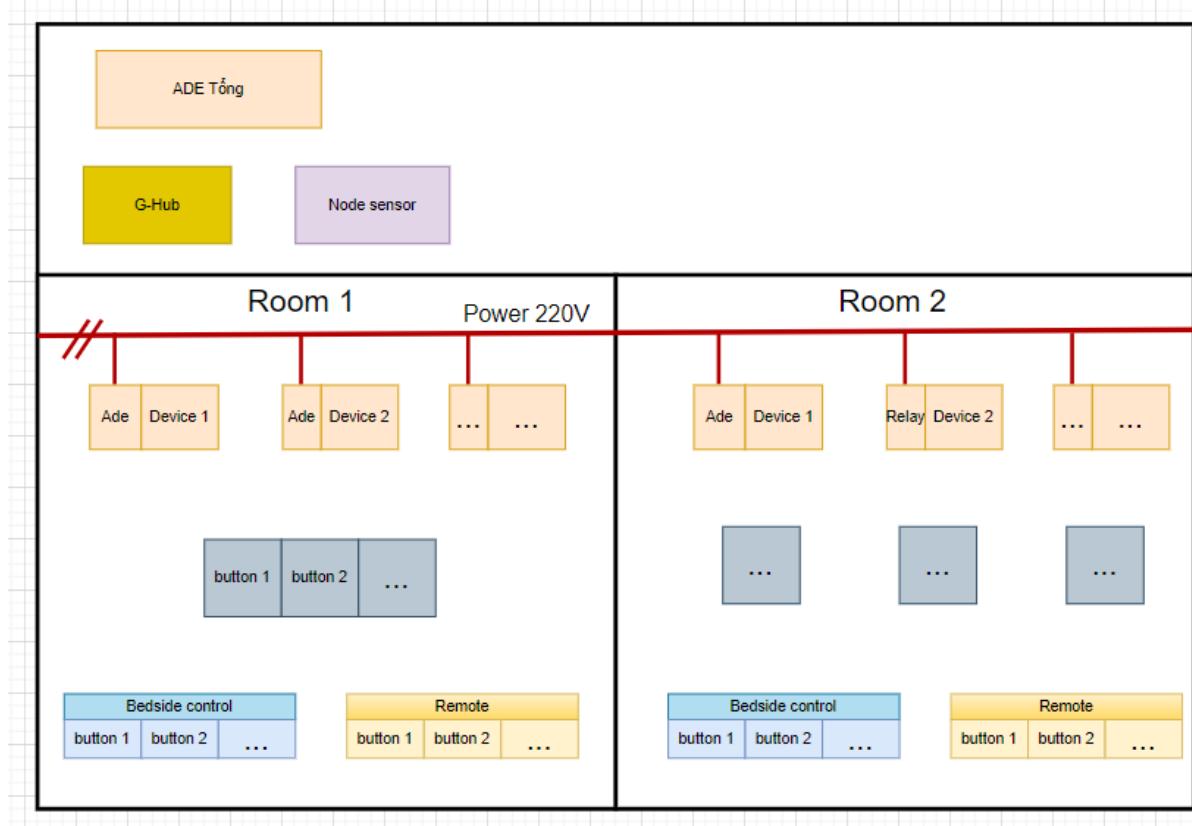
Hình 7.4. Dữ liệu gửi đi khi người dùng nhấn nút mà có liên kết

Sau khi liên kết người dùng có thể điều khiển thiết bị thông qua các nút nhấn. Và dữ liệu sẽ được truyền đi như hình 7.4.

## 7.2. Panel thử nghiệm



Hình 7.5. Hình ảnh panel được dùng để kiểm tra



Hình 7.6. Vị trí lắp đặt các Node trên Panel

Panel được thử nghiệm thấy được một relay có thể được điều khiển ở nhiều vị trí khác nhau, từ nút nhàn, bedside control hay là sử dụng remote. Relay còn có thể được điều khiển được thông qua trang web.

## **Chương 8. Kết luận và hướng phát triển**

### **8.1. Kết quả đạt được**

Mạng Zigbee có thể giao tiếp và hoạt động ổn định và có thể dễ dàng mở rộng hơn.

Thiết kế trang web: giao diện dễ nhìn và dễ sử dụng, lưu trữ dữ liệu vào database.

Các node hoạt động tốt và ổn định.

Năm tổng quát cách hoạt động của một hệ thống.

### **8.2. Hạn chế**

#### **8.2.1. Thời gian**

- Không thể kịp để thử nghiệm trong thực tế nhiều được.
- Không kịp thiết kế các tính năng thông minh hơn để dẫn tới hoạt động tự động trong hệ thống và tối ưu sử dụng điện.
- Không kịp để thiết kế lại và tối ưu các node, để các node có thể tối ưu về mặt điện năng để sử dụng pin có thể lâu hơn và hoạt động dài hạn hơn.

#### **8.2.2. Vật chất**

- Không đủ các trang thiết bị để phát triển, tối ưu các node.
- Không thể thử nghiệm được những phần cứng có chi phí cao dẫn tới sự lựa chọn ít đi.

#### **8.2.3. Kiến thức**

- Để phát triển một hệ thống hoạt động bền vững thì cần có rất nhiều các kiến thức từ cách vận hành, cho tới thiết kế từng chi tiết nhỏ nhất. Cần có rất nhiều kiến thức cả trong và ngoài lề các kiến thức đã học trong trường.
- Cần phải học các kiến thức mới để theo kịp trong thời công nghệ 4.0 như ngày nay

## **8.3. Hướng phát triển đề tài**

### **8.3.1. Tối ưu các Node có sẵn**

Tối ưu về mặt điện áp và dòng điện sử dụng trên các node để các node có thể hoạt động lâu hơn với pin. Tránh tình trạng phải thay pin thường xuyên.

Phát triển thêm các tính năng trong các node kết hợp với việc phát triển các node mới như node relay kết hợp với node human detection.

Phát triển phát hiện giọng nói trong G-Hub để có thể điều khiển bằng giọng nói.

### **8.3.2. Thiết kế thêm các Node mới**

Có nhiều hướng để phát triển các node mới để có thể kết hợp với các node cũ để phát triển hay tạo thêm những node mới có tính riêng và độc đáo hơn.

- Node chuông cửa: Phát hiện ra người tới và thông báo đến cho người nhà biết có khách tới, có thể phát triển thêm camera để có thể hiển thị hình ảnh của người tới để cho người nhà biết đó là ai
- Node human detection: Dùng để phát hiện người trong một khu vực nào đó để phát triển và kết hợp với relay hoặc những node khác để bật/ tắt khi có người hoặc không có người.
- Node điều khiển điều hòa: Điều khiển điều hòa, cài đặt giờ cho điều hòa.

### **8.3.3. Thiết kế các tính năng mới**

Có rất nhiều tính năng trong nhà mà có thể phát triển.

- Home context: Có thể tạo ra các trường hợp trong nhà và cài đặt hẹn giờ ở trong mỗi trường hợp khác nhau mà các thiết bị trong ngôi nhà sẽ bật/ tắt tương ứng với trường hợp đó.
- Timer Relay: Cài đặt thời gian bật/ tắt cho các thiết bị
- Phát hiện dòng điện bất thường: Dựa trên node Relay ADE cung cấp cho các thông tin về điện của một thiết bị nào đó, khi gặp trạng thái bất thường của thiết bị thì sẽ phát hiện và thông báo cho người dùng.
- Phát hiện thời điểm bật/ tắt thường xuyên của thiết bị để dần dần trở nên tự động

- Từ những dữ liệu thu thập được trong quá trình sử dụng của người dùng, các thiết bị sẽ thường có những thời điểm bật/ tắt lặp đi lặp lại và từ những dữ liệu đó có thể dần dần trở nên tự động cho thiết bị.

## Tài liệu tham khảo

- [1] Sinem Coleri Ergen “ZigBee/IEEE 802.15.4 Summary” September 10, 2004.
- [2] Bạn Phạm Minh Trí và bạn Phạm Đức Ái “Thiết kế và thi công mô hình nhà thông minh sử dụng mạng không dây Zigbee”, Luận văn đại học, Đại học Bách Khoa tp.Hcm, 2022.
- [3] John A.Stankovic, R.Rajkumar “Realtime operating system”
- [4] espressif, “esp32 documentation” [online]. Available: <https://docs.espressif.com>
- [5] freertos, “freertos documentation” [online]. Available: <https://www.freertos.org>.
- [6] Microchip Technology, “Pic16f628a datasheet” “Pic16f887 datasheet” [online].
- [7] Zhengzhou Winsen Electronics Technology Co., “Air Quality Gas Sensor”, 10/03/2015
- [8] D-Robotics “DHT11Humidity & Temperature Sensor”, 7/30/2010
- [9] Analog Devices, “ade7753 datasheet” [online]. Available: <https://www.analog.com/en/index.html>
- [10] Palaksinghal9903 “Frontend vs Backend”, <https://www.geeksforgeeks.org/frontend-vs-backend>, access 26/05/2023
- [11] Reactjs “document”, [online]: <https://legacy.reactjs.org/>
- [12] MUI “document”, [online]: <https://mui.com/material-ui/getting-started/overview/>
- [13] Mongodb, <https://www.mongodb.com/> [online]. Available: <https://www.mongodb.com/docs/atlas/> [Accessed 25 04 2023].
- [14] Database, <https://viblo.asia/p/lam-the-nao-de-thiet-ke-mot-co-so-du-lieu-phan-1-rYvGwvKVw> [online]. Available: <https://viblo.asia/p/lam-the-nao-de-thiet-ke-mot-co-so-du-lieu-phan-1-rYvGwvKVw> [Accessed 14 01 2023].
- [15] hivemq, “series of articles about mqtt” [online]. Available: <https://www.hivemq.com/tags/mqtt-essentials>  
Available: <https://www.microchip.com>
- [16] Electrical Technology “How to Control One Light Bulb From Two Different Locations?”, [online]: <https://www.electricaltechnology.org/2022/06/control-light-bulb-from-two-locations.html>

[17] Hai au “Cadivi VCM-1 mm<sup>2</sup> flexible single wire (Green) – 300/500V”, [online]:  
<https://codienhaiau.com/product/day-dien-cadivi-vcm-1-1x32-0-2-300-500v/>